Safe yet Precise Soft Robots

Incorporating Physics into Learned Models for Control

Stölzle, Maximilian

DOI
10.4233/uuid:24c1f667-8fd6-431a-bb78-11d22f8cb3da

Publication date
2025

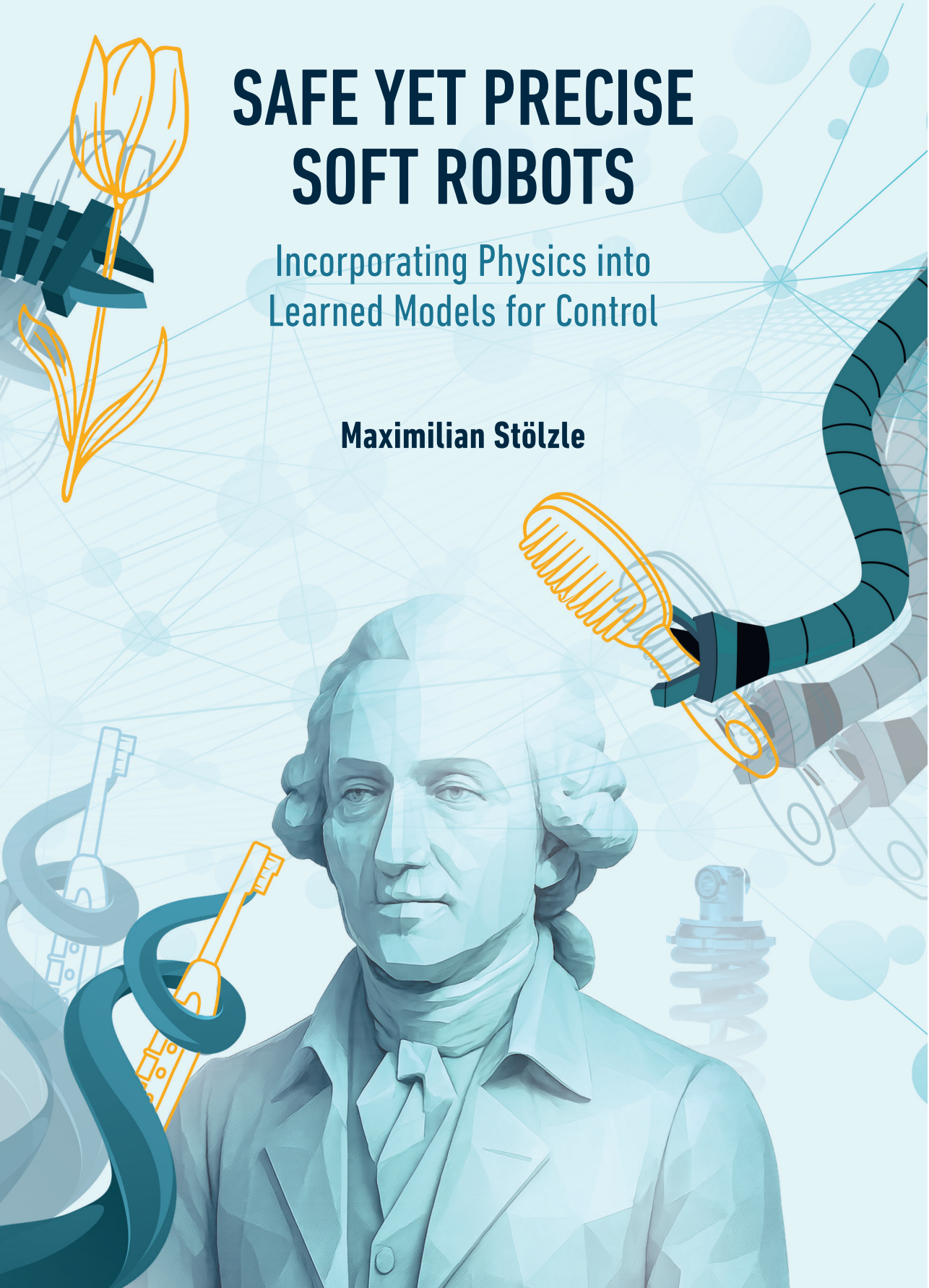
Document Version
Final published version

Citation (APA)
Stölzle, M. (2025). *Safe yet Precise Soft Robots: Incorporating Physics into Learned Models for Control*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:24c1f667-8fd6-431a-bb78-11d22f8cb3da

SAFE YET PRECISE SOFT ROBOTS

Incorporating Physics into Learned Models for Control

Maximilian Stölzle



About the Cover

The cover envisions continuum soft robotic manipulators — modeled after invertebrates such as octopus tentacles — operating in human-centric environments (e.g., selling tulips at the zaterdagmarkt) and assisting people with activities of daily living (e.g., hair brushing and toothbrushing) while ensuring safety via their inherent compliance. Controlling these systems is challenging due to their effectively infinite degrees of freedom and distributed actuation; therefore, this thesis focuses on dynamical models that predict the robot's future motion and shape, as symbolized by the translucent silhouettes. We target control-friendly dynamics with explicit physical structure—such as control-affine Euler–Lagrange equations—hence the portrait of Joseph-Louis Lagrange. Our approach combines physics-based and data-driven methods, incorporating priors such as damped harmonic oscillators (represented by spring–damper elements) into modern neural networks (illustrated by the neuron motif in the background), ultimately enabling precise and efficient control through closed-form energy-shaping feedforward terms.

Safe yet Precise Soft Robots

Incorporating Physics into Learned Models for Control

Dissertation

for the purpose of obtaining the degree of Doctor
at Delft University of Technology,
by the authority of the Rector Magnificus, Prof. dr. ir. T.H.J.J. van der Hagen,
Chair of the Board of Doctorates,
to be defended publicly on
Monday, 15 September 2025 at 17:30 o'clock

by

Maximilian STÖLZLE

Master of Science ETH in Mechanical Engineering,
Swiss Federal Institute of Technology, Switzerland,
born in Göppingen, Germany.

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus,	Delft University of Technology, *Chairperson*
Prof. Dr. R. Babuška,	Delft University of Technology, *Promotor*
Dr. C. Della Santina,	Delft University of Technology, *Copromotor*
Independent Members:	
Jun. Prof. Dr. G. Rizzello,	Saarland University, Germany
Dr. M. Bächer,	Disney Research, Switzerland
Prof. Dr. Ir. T. A. E. Oomen,	Eindhoven University of Technology, Netherlands
Prof. Dr. G. C. H. E. de Croon,	Delft University of Technology, Netherlands
Prof. Dr. G. Zardini,	Massachusetts Institute of Technology, USA
Prof. Dr. Ir. M. Wisse,	Delft University of Technology, *Reserve member*

This research was partially funded by the European Union's Horizon Europe Program from Project EMERGE - Grant Agreement No. 101070918 and by the Dutch Cultuurfonds Wetenschapsbeurzen 2024. The research visit to LIDS/Zardini Lab at MIT was supported by the Rudge (1948) and Nancy Allen Chair. A significant part of the research appearing in this thesis was developed in collaboration with Prof. Dr. Daniela Rus and Prof. Dr. Gioele Zardini from the Massachusetts Institute of Technology, USA.
Published and distributed by: Maximilian Stölzle. E-mail: maximilian@stoelzle.ch

Keywords:	Soft Robotics, Nonlinear Control, Machine Learning, AI
Printed by:	Ridderprint | `www.ridderprint.nl`
Cover:	Maximilian Stölzle (Concept) & Joey Impoza Roberts (Realization)
Style:	TU Delft House Style, with modifications by Moritz Beller

ISBN/EAN: 978-94-6384-836-7
DOI: 10.4233/uuid:24c1f667-8fd6-431a-bb78-11d22f8cb3da

An electronic version of this dissertation is available at
`http://repository.tudelft.nl/`.

If you can model it, you can control it.

W. Morrow and G. Strickland[1]

[1] https://edu.nl/u4ujn

Contents

Summary

As we increasingly strive to integrate robots into human-centric environments, safety is a top priority. Traditionally, rigid collaborative robots have relied on safety-aware computational control policies, which are susceptible to perception errors and often lead to overly cautious behavior that limits performance. In contrast, soft robotics offers a promising alternative by ensuring passive compliance throughout the robot's structure via material softness. This mechanical compliance inherently mitigates safety issues arising from perception or control errors, although this has been paid for with a substantial drop in precision. In recent years, significant advances have been seen in soft robotics, with exciting new developments in design, smart materials, actuators, sensors, models, and control strategies. However, the modeling and control of continuum soft robots continue to pose major challenges due to their infinite degrees of freedom, complex nonlinear dynamics, and time-dependent behaviors like hysteresis. As a result, soft robots often lack the necessary capability and motion precision, leading to a tradeoff where performance is sacrificed for safety. With this thesis, we argue that this tradeoff can be overcome by developing more advanced algorithms that can reason on the physics of the soft robot. More specifically, we propose combining powerful learned models with efficient and effective model-based control approaches that allow for interpretability into the actions and admit stability guarantees.

Currently, two main approaches exist for controlling soft robots. The first employs model-based control using approximated physics-based models derived from first principles. The second directly learns control policies, primarily through reinforcement learning. Both strategies face notable limitations. Existing model-based controllers are unable to fully manage and eventually exploit the dynamics of soft robots because their underlying models inadequately capture complex behaviors, particularly how actuation and external interactions affect the robot's deformation. Moreover, deriving these models requires extensive expert knowledge. Additionally, the combined complexity and uncertainty of the dynamics between a soft robot and its environment make it currently infeasible to develop comprehensive world models from first principles alone, thereby motivating the integration of machine learning approaches that can effectively leverage data-driven insights. Conversely, directly learning the controller — such as via reinforcement learning — lacks interpretability and stability guarantees while being highly sample inefficient, a significant drawback given the time-dependent material properties and limited lifespan of current soft robots.

In this thesis, we contend that combining learned models with model-based controllers presents a promising alternative that brings together the advantages of both approaches: expressive, data-driven models that require less expert knowledge paired with controllers that are both interpretable and provably stable. Although recent years have seen increased interest in leveraging learned models for control, most work in this area depends on computationally intensive optimal control methods, such as MPC, to optimize the actuation

sequence with the learned model. However, the high computational cost of solving these optimal control problems limits the maximum control frequency during deployment, preventing us from fully exploiting the dynamic capabilities of soft robots. Instead, this thesis pursues closed-form controllers that utilize the physical structure of learned models within an energy-shaping framework. The main challenge here is to develop approaches that integrate such physical structures—specifically, kinetic and potential energy terms—into the learning of dynamical models for soft robots. Before addressing this main challenge, we first had to advance physics-based models derived from first principles and identify novel techniques to leverage them for control. On one hand, this clarified which physical priors were available for learning, while on the other hand, it inspired new ways to integrate model-based controllers with learned models. The thesis addresses this topic through several interconnected key contributions.

First, we argue that quantifying the safety of soft robots is crucial for designing and controlling them to ensure that the closed-loop system meets the specific safety requirements of their intended applications. To this end, we present the first safety metric for continuum soft robots, which assesses the safety of an integrated soft robot design by accounting for both its embodied and computational intelligence.

Secondly, this thesis enhances shape sensing for soft robots by leveraging insights from kinematic models. We accomplish this by formulating and solving nonlinear optimization problems that align sensor measurements with the backbone shapes predicted by the kinematic model. We present two distinct approaches that integrate commercial sensors—namely, visual and magnetic—with SLAM algorithms and a learned sensor measurement model, respectively, to accurately estimate the soft robot's state, a key requirement for effective feedback control.

Thirdly, this thesis introduces advanced physics-based actuation models, including those for robots actuated by auxetic metamaterials - referred to as HSA robots—and models that capture the actuation dynamics of piston-driven pneumatic soft robots. We then leverage the acquired model insights to design provably stable nonlinear controllers—specifically, an integral-saturated PID combined with potential shaping and Cartesian-space impedance control for planar HSA robots, as well as a backstepping controller for pneumatic piston-driven soft robots. This contribution deepens our understanding of actuation, a critical aspect of soft robot behavior, and demonstrates how such insights can be incorporated into model-based control strategies. Moreover, experiments with HSA robots have highlighted the limitations of purely physics-based models in capturing complex phenomena like hysteresis, thereby motivating the exploration of learning-based approaches. In the future, the developed actuation models can serve as valuable physical priors for learned models.

Fourthly, the thesis presents techniques for learning soft robot models that incorporate physical structures while ensuring stability. We accomplish this by embedding physics-based dynamical models into the learning algorithm, which determines the free parameters of the dynamics and optionally optimizes a coordinate transformation—such as encoding into latent space. Two notable approaches are introduced: (1) an algorithm that extracts low-dimensional soft robot strain models from samples of the robot backbone's shape evolution, and (2) a network of coupled harmonic oscillators for learning latent dynamics from high-dimensional observations like images. The explicit inclusion of kinematic and potential energy terms in these models allows for stability analysis using standard nonlinear

system theory tools, such as Lyapunov methods. For instance, we prove that the coupled oscillator network is both globally asymptotically stable and input-to-state stable.

Fifthly, we exploit the physical structure of the learned models from contribution four to design closed-form setpoint regulators. The controller contains two key components: (1) a potential shaping feedforward term that positions the local/global minimum of the closed-loop potential energy at the setpoint by leveraging the learned model knowledge, and (2) an integral-saturated PID feedback term that rejects disturbances and compensates for modeling errors to prevent steady-state errors. The stability of the closed-loop system can then be analyzed using Lyapunov arguments.

Finally, the thesis explores methods for generating compliant motion behaviors in soft robots beyond low-level control. One approach focuses on assisting users, particularly elderly individuals, with activities of daily living by guiding the low-level controller with brain signals. This is achieved by combining motor imagery classification from wearable EEG devices with compliant impedance control in operational space. The second approach combines an orbitally stable dynamical system in latent space with a bijective neural network parametrized encoder to learn periodic motions from demonstrations. By avoiding reliance on time references, this learned motion policy enables natural and compliant tracking of demonstrated periodic motions. This contribution ensures that not just the robot structure and low-level controller are compliant, but also the high-level motion strategy.

In summary, this thesis aims to work towards safe and precise soft robot behavior in human-centric environments. In this context, safety is achieved through the soft body's mechanical compliance. To enhance precision while maintaining insight into decision-making, ensuring stability, and preserving computational efficiency, the core contribution of this work is the development of closed-form soft robot controller architectures and their connection to learned models. Several supporting contributions include a metric for quantifying the safety of soft robots, strategies for integrating kinematic models into shape sensing methods, advanced physics-based actuation models that could serve as future priors for learned models, and moving beyond low-level controllers by devising compliant motion strategies. All proposed models, sensing methods, control strategies, and motion approaches have been thoroughly verified through simulation or real-world testing on soft robots. The code and data from most chapters have been made publicly available on GitHub, thereby contributing to the broader research community.

Samenvatting

Aangezien we er in toenemende mate naar streven robots te integreren in omgevingen die op mensen gericht zijn, staat veiligheid voorop. Doorgaans baseren rigide collaboratieve robots zich op veiligheidsbewuste computationele regelalgorithmen, die gevoelig zijn voor waarnemingsfouten en vaak leiden tot een te voorzichtige werking die de prestaties beperkt. Zachte flexibele robotica (Soft Robotic) daarentegen biedt een veelbelovend alternatief door passieve compliantie in de gehele robotstructuur te waarborgen via materiaaleigenschappen. Deze mechanische compliantie beperkt inherent veiligheidsproblemen die voortkomen uit fouten in waarneming of regeling, al gaat dit ten koste van een aanzienlijke daling in precisie. In de afgelopen jaren zijn grote vooruitgangen geboekt in de zachte robotica, met veelbelovend nieuwe ontwikkelingen in ontwerp, slimme materialen, actuatoren, sensoren, modellen en regelstrategieën. Het modelleren en regelen van continuümzachte robots brengt echter grote uitdagingen met zich mee vanwege hun oneindige vrijheidsgraden, complexe niet-lineaire dynamica en tijdsafhankelijke verschijnselen zoals hysterese. Daardoor missen zachte robots vaak de benodigde capaciteit en bewegingsprecisie, wat leidt tot een compromis waarbij prestaties worden opgeofferd voor veiligheid. Met dit proefschrift stellen wij dat dit compromis kan worden overwonnen door geavanceerdere algoritmen te ontwikkelen die de fysica van de zachte robot in acht nemen. Meer specifiek stellen wij voor om krachtige geleerde modellen te combineren met efficiënte en effectieve modelgebaseerde regelmethoden die inzicht in de handelingen bieden en stabiliteitsgaranties toelaten.

Momenteel bestaan er twee primaire strategie voor het regelen van zachte robots. De eerste maakt gebruik van modelgebaseerde regeling met benaderde fysische modellen die zijn afgeleid uit eerste principes. De tweede leert regeltechnieken rechtstreeks, voornamelijk via reinforcement learning. Beide strategieën kennen aanzienlijke beperkingen. Bestaande modelgebaseerde regelaars zijn niet in staat de dynamica van zachte robots volledig te beheersen en uiteindelijk te benutten, omdat hun onderliggende modellen complexe gedragingen — met name hoe actuatie en externe interacties de vervorming van de robot beïnvloeden — ontoereikend vastleggen. Bovendien vergt het afleiden van deze modellen uitgebreide expertise. Daarnaast maken de gecombineerde complexiteit en onzekerheid van de dynamica tussen een zachte robot en haar omgeving het momenteel onhaalbaar om louter vanuit eerste principes volledige wereldmodellen te ontwikkelen, wat de integratie motiveert van machine-learning-methodes die datagedreven inzichten effectief kunnen benutten. Daarentegen biedt het rechtstreeks leren van de regelstrategie, bijvoorbeeld via reinforcement learning, geen interpretatie en stabiliteitsgaranties en gaat het gepaard met een hoge monsterinefficiëntie, wat problematisch is door de tijdsafhankelijke materiaaleigenschappen en beperkte levensduur van huidige zachte robots.

In dit proefschrift betogen wij dat het combineren van geleerde modellen met modelgebaseerde regelaars een veelbelovend alternatief vormt dat de voordelen van beide strategien samenbrengt: expressieve, datagedreven modellen die minder expertkennis

vereisen, gekoppeld aan regelaars die zowel interpreteerbaar als aantoonbaar stabiel zijn. Hoewel de laatste jaren toenemende interesse is ontstaan in het benutten van geleerde modellen voor de regeltechniek, is het merendeel van dit werk afhankelijk van computationeel intensieve optimale-regelmethoden, zoals MPC, om de actuatiesequentie te optimaliseren met behulp van het geleerde model. De hoge rekenkosten van het oplossen van deze optimale-regelproblemen beperken echter de maximale regelfrequentie tijdens inzet, waardoor we de dynamische mogelijkheden van zachte robots niet volledig kunnen benutten. Dit proefschrift richt zich daarom op closed-form-regelaars die de fysische structuur van geleerde modellen benutten binnen een energy-shaping-raamwerk. De belangrijkste uitdaging hierbij is het ontwikkelen van methoden die dergelijke fysische structuren — specifiek kinetische en potentiële energie-termen — integreren in het leren van dynamische modellen voor zachte robots.

Voordat we deze uitdaging aanpakten, moesten we eerst fysica-gebaseerde modellen afleiden uit eerste principes en nieuwe technieken identificeren om ze voor regeling te benutten. Enerzijds maakte dit duidelijk welke fysische voorkennis beschikbaar waren voor het leren, terwijl het anderzijds nieuwe manieren inspireerde om modelgebaseerde regelaars met geleerde modellen te integreren. Het proefschrift behandelt dit onderwerp via verschillende onderling verbonden kernbijdragen.

Ten eerste stellen wij dat het kwantificeren van de veiligheid van zachte robots essentieel is voor het ontwerpen en regelen ervan, zodat het gesloten-lus regelsysteem voldoet aan de specifieke veiligheidseisen van de beoogde toepassingen. Daartoe presenteren wij de eerste veiligheidsmaatstaf voor continuümzachte robots, die de veiligheid van een geïntegreerd ontwerp van een zachte robot beoordeelt door zowel de belichaamde als de computationele intelligentie in rekening te brengen.

Ten tweede verbetert dit proefschrift vormwaarneming voor zachte robots door inzichten uit kinematische modellen te benutten. Dit bereiken we door niet-lineaire optimalisatieproblemen te formuleren en op te lossen die sensormetingen uitlijnen met de ruggraadvormen die door het kinematische model worden voorspeld. We presenteren twee verschillende benaderingen die commerciële visuele en magnetische sensoren te integretren met integreren met respectievelijk SLAM-algoritmen en een geleerd sensormeetmodel om de toestand van de zachte robot nauwkeurig te schatten, een sleutelfactor voor doeltreffende feedbackregeling.

Ten derde introduceert dit proefschrift geavanceerde fysica-gebaseerde actuatiemodellen, waaronder modellen voor robots die worden aangedreven door auxetische metamaterialen — aangeduid als HSA-robots — en modellen die de actuatiedynamica van zuiger-gedreven pneumatische zachte robots vastleggen. Vervolgens benutten we de verkregen modelinzichten om aantoonbaar stabiele niet-lineaire regelaars te ontwerpen — met name een integraal-gesatureerde PID gecombineerd met potentiaalvormende en impedantieregeling in Cartesiaanse ruimte voor planaire HSA-robots, alsook een backstepping-regelaar voor pneumatische zuiger-gedreven zachte robots. Deze bijdrage verdiept ons begrip van actuatie, een cruciaal aspect van het gedrag van zachte robots, en toont aan hoe dergelijke inzichten kunnen worden geïntegreerd in modelgebaseerde regelstrategieën. Bovendien hebben experimenten met HSA-robots de beperkingen blootgelegd van puur fysica-gebaseerde modellen bij het vastleggen van complexe fenomenen zoals hysterese, waarmee de noodzaak van op leren gebaseerd benaderingen wordt onderstreept. In de toe-

komst kunnen de ontwikkelde actuatiemodellen dienen als waardevolle fysische voorkennis voor geleerde modellen.

Ten vierde presenteert het proefschrift technieken voor het leren van modellen van zachte robots die fysische structuren integreren en tegelijk stabiliteit waarborgen. Dit realiseren we door fysica-gebaseerde dynamische modellen op te nemen in het leeralgorithme, dat de vrije parameters van de dynamica bepaalt en optioneel een coördinatentransformatie — zoals een latente-ruimte-encodering — optimaliseert. Twee noemenswaardige benaderingen worden geïntroduceerd: (1) een algoritme dat laagdimensionale vervormingsmodellen van zachte robots afleidt uit voorbeelden van de vorm-evolutie van de ruggengraat van de robot, en (2) een netwerk van gekoppelde harmonische oscillatoren voor het leren van latente dynamica uit hoogdimensionale observaties zoals beelden. De expliciete opname van kinematische en potentiële-energie-termen in deze modellen maakt stabiliteitsanalyse mogelijk met tools uit de niet-lineare systeemtheorie, zoals Lyapunov-methoden. Zo bewijzen wij bijvoorbeeld dat het netwerk van gekoppelde oscillatoren zowel globaal asymptotisch stabiel als input-to-state-stabiel is.

Ten vijfde benutten wij de fysische structuur van de geleerde modellen uit bijdrage vier om closed-form-setpoint-regelaars te ontwerpen. De regelaar bevat twee kernelementen: (1) een potentiaalvormende-feedforwardterm die het lokale of globale minimum van de gesloten-lus-potentiële energie op het setpoint plaatst door de kennis uit het geleerde model te benutten, en (2) een integraal-gesatureerde PID-feedbackterm die verstoringen onderdrukt en modelleringsfouten compenseert om stationaire fouten te voorkomen. De stabiliteit van het gesloten-lus regelsysteem kan vervolgens met Lyapunov-argumenten worden geanalyseerd.

Ten slotte verkent het proefschrift methoden om conform methode in zachte robots te genereren, voorbij de laag-niveau-regeling. Eén bewegingen richt zich op het ondersteunen van gebruikers, met name ouderen, bij activiteiten van het dagelijks leven door de laag-niveau-regelaar te sturen met hersensignalen. Dit wordt bereikt door motor-beeld-classificatie van draagbare EEG-apparaten te combineren met compliant impedantieregeling in de operationele ruimte. De tweede methode combineert een orbitale-stabiel dynamisch systeem in latente ruimte met een bijectieve neurale-netwerk-geparametriseerde encoder om periodieke bewegingen uit demonstraties te leren. Doordat het bewegingsbeleid geen gebruikmaakt van tijdreferenties, kan het op een natuurlijke en meegevende manier periodieke gedemonstreerde bewegingen volgen. Deze bijdrage waarborgt dat niet alleen de robotstructuur en de laag-niveau-regelaar compliant zijn, maar ook de hoog-niveau-bewegingsstrategie.

Samengevat beoogt dit proefschrift bij te dragen aan veilig en nauwkeurig gedrag van zachte robots in mensgerichte omgevingen. Veiligheid wordt bereikt via de mechanische compliantie van het zachte lichaam. Om de precisie te verbeteren terwijl inzicht in besluitvorming, stabiliteit en computationele efficiëntie behouden blijven, is de kernbijdrage van dit werk de ontwikkeling van closed-form-regelarchitecturen voor zachte robots en hun koppeling aan geleerde modellen. Verscheidene ondersteunende bijdragen omvatten een maatstaf voor het kwantificeren van de veiligheid van zachte robots, strategieën voor het integreren van kinematische modellen in vormwaarnemingsmethoden, geavanceerde fysica-gebaseerde actuatiemodellen die als toekomstige voorkennis voor geleerde modellen kunnen dienen, en het overstijgen van laag-niveau-regelaars door compliant

bewegingsstrategieën te ontwikkelen. Alle voorgestelde modellen, waarnemingsmethoden, regelstrategieën en bewegingsbenaderingen zijn grondig gevalideerd via simulaties of experimenten in de echte wereld op zachte robots. De code en data van de meeste hoofdstukken zijn publiekelijk beschikbaar gesteld op GitHub en dragen zo bij aan de bredere onderzoeksgemeenschap.

Zusammenfassung

Da wir zunehmend bestrebt sind, Roboter in menschenzentrierte Umgebungen zu integrieren, hat Sicherheit höchste Priorität. Traditionell haben starre kollaborative Roboter auf sicherheitsbewusste, rechnergestützte Regelstrategien zurückgegriffen, die anfällig für Wahrnehmungsfehler sind und häufig zu übervorsichtigem Verhalten führen, das die Leistung begrenzt. Im Gegensatz dazu bietet die weiche Robotik (Soft Robotic) eine vielversprechende Alternative, indem sie durch Materialweichheit eine passive Nachgiebigkeit im gesamten Roboteraufbau sicherstellt. Diese mechanische Nachgiebigkeit mindert Sicherheitsprobleme, die aus Wahrnehmungs- oder Regelungsfehlern resultieren, geht jedoch mit einem erheblichen Präzisionsverlust einher. In den letzten Jahren wurden in der weichen Robotik bedeutende Fortschritte erzielt, mit interessanten neuen Entwicklungen in Design, intelligenten Materialien, Aktoren, Sensoren, Modellen und Regelungsstrategien. Dennoch stellen die Modellierung und Regelung von kontinuierlichen weichen Robotern nach wie vor große Herausforderungen dar, da sie unendlich viele Freiheitsgrade, komplexe nichtlineare Dynamiken und zeitabhängige Verhaltensweisen wie Hysterese aufweisen. Folglich mangelt es weichen Robotern häufig an der erforderlichen Leistungsfähigkeit und Bewegungspräzision, was zu einem Kompromiss führt, bei dem Leistung der Sicherheit geopfert wird. In dieser Arbeit argumentieren wir, dass dieser Kompromiss überwunden werden kann, indem fortschrittlichere Algorithmen entwickelt werden, die auf der Physik des weichen Roboters basieren. Genauer gesagt schlagen wir vor, leistungsstarke, gelernte Modelle mit effizienten und effektiven modellbasierten Regelungsansätzen zu kombinieren, die Interpretierbarkeit des Handelns ermöglichen und Stabilitätsgarantien zulassen.

Derzeit existieren zwei Hauptansätze zur Regelung weicher Roboter. Der erste verwendet modellbasierte Regelungen, die auf approximierten, physikbasierten Modellen beruhen, die aus ersten Prinzipien abgeleitet sind. Der zweite Ansatz erlernt Steuerungsrichtlinien direkt, überwiegend mittels Reinforcement Learning. Beide Strategien weisen erhebliche Einschränkungen auf. Bestehende modellbasierte Regler können die Dynamik weicher Roboter nicht vollständig beherrschen und schließlich ausnutzen, da ihre zugrunde liegenden Modelle komplexe Verhaltensweisen, insbesondere wie Aktuierung und externe Interaktionen die Verformung des Roboters beeinflussen, unzureichend erfassen. Darüber hinaus erfordert die Ableitung dieser Modelle umfangreiches Expertenwissen. Ferner machen die kombinierte Komplexität und Unsicherheit der Dynamik zwischen einem weichen Roboter und seiner Umgebung es derzeit unmöglich, umfassende Weltmodelle ausschließlich aus ersten Prinzipien zu entwickeln, was die Integration von Machine-Learning-Ansätzen motiviert, die datengetriebene Erkenntnisse effektiv nutzen können. Umgekehrt mangelt es beim direkten Erlernen des Reglers – etwa durch Reinforcement Learning – an Interpretierbarkeit und Stabilitätsgarantien, während es zudem hochgradig sample-ineffizient ist, ein signifikanter Nachteil angesichts der zeitabhängigen Materialeigenschaften und der begrenzten Lebensdauer heutiger weicher Roboter.

In dieser Arbeit vertreten wir die Auffassung, dass die Kombination gelernter Modelle mit modellbasierten Reglern eine vielversprechende Alternative darstellt, welche die Vorteile beider Ansätze vereint: ausdrucksstarke, datengetriebene Modelle, die weniger Expertenwissen erfordern, gepaart mit Reglern, die sowohl interpretierbar als auch nachweislich stabil sind. Obwohl in den letzten Jahren das Interesse gewachsen ist, gelernte Modelle für die Regelung zu nutzen, basiert der Großteil der Arbeiten in diesem Bereich auf rechenintensiven optimalen Regelungsmethoden wie MPC, um mit dem gelernten Modell die Antriebssequenz zu optimieren. Die hohen Rechenkosten bei der Lösung dieser Optimalsteuerungsprobleme begrenzen jedoch die maximale Regelungsfrequenz während des Einsatzes und verhindern so die volle Ausschöpfung der dynamischen Fähigkeiten weicher Roboter. Stattdessen verfolgt diese Arbeit geschlossene Regler in geschlossener Form, die die physikalische Struktur gelernter Modelle ausnutzen um deren Energieprofil durch Regelung zu verformen. Die Hauptschwierigkeit besteht darin, Ansätze zu entwickeln, die solche physikalischen Strukturen – insbesondere kinetische und potenzielle Energieanteile – in das Erlernen dynamischer Modelle für weiche Roboter integrieren.

Bevor wir uns dieser Hauptaufgabe widmeten, mussten wir zunächst physikbasierte Modelle aus ersten Prinzipien weiterentwickeln und neue Techniken identifizieren, um sie für die Regelung zu nutzen. Einerseits wurde dadurch geklärt, welche physikalischen Priors für das Lernen verfügbar sind, andererseits inspirierte es neue Wege, modellbasierte Regler mit gelernten Modellen zu integrieren. Die Arbeit behandelt dieses Thema durch mehrere miteinander verbundene Hauptbeiträge.

Erstens argumentieren wir, dass die Quantifizierung der Sicherheit weicher Roboter entscheidend ist, um sie so zu entwerfen und zu regeln, dass das geschlossene System die spezifischen Sicherheitsanforderungen seiner vorgesehenen Anwendungen erfüllt. Zu diesem Zweck präsentieren wir die erste Sicherheitsmetrik für kontinuierliche weiche Roboter, die die Sicherheit eines integrierten weichen Roboters bewertet, indem sie sowohl seine verkörperte als auch seine rechnerische Intelligenz berücksichtigt.

Zweitens verbessert diese Arbeit die Formwahrnehmung (Shape Sensing) für weiche Roboter, indem Erkenntnisse aus kinematischen Modellen genutzt werden. Dies erreichen wir, indem wir nichtlineare Optimierungsprobleme formulieren und lösen, die Sensormessungen mit den durch das kinematische Modell vorhergesagten Rückgratsformen abgleichen. Wir präsentieren zwei unterschiedliche Ansätze, die kommerzielle Sensoren – nämlich visuelle und magnetische – mit SLAM-Algorithmen bzw. mit einem gelernten Sensor-Messmodell kombinieren, um den Zustand des weichen Roboters genau zu schätzen, eine wichtige Voraussetzung für eine effektive Regelung mit Rückkopplung.

Drittens führt diese Arbeit fortschrittliche physikbasierte Aktuierungsmodelle ein, darunter solche für Roboter, die durch auxetische Metamaterialien – sogenannte HSA-Roboter – angetrieben werden, sowie Modelle, die die Antriebsdynamik von kolbengesteuerten pneumatischen weichen Robotern erfassen. Anschließend nutzen wir die gewonnenen Modellkenntnisse, um nachweislich stabile nichtlineare Regler zu entwerfen – konkret einen integralgesättigten PID-Regler kombiniert mit Umformung der potentiellen Energie und Impedanzregelung im kartesischen Raum für planare HSA-Roboter sowie einen Backstepping-Regler für pneumatische, kolbengesteuerte weiche Roboter. Dieser Beitrag vertieft unser Verständnis des Antriebs, eines entscheidenden Aspekts des Verhaltens weicher Roboter, und zeigt, wie solche Erkenntnisse in modellbasierte Regelungsstrategi-

en eingebunden werden können. Darüber hinaus haben Experimente mit HSA-Robotern die Grenzen rein physikbasierter Modelle bei der Erfassung komplexer Phänomene wie Hysterese aufgezeigt und motivieren so die Erforschung lernbasierter Ansätze. Zukünftig können die entwickelten Antriebsmodelle als wertvolle physikalische Vorwissen für gelernte Modelle dienen.

Viertens stellt die Arbeit Techniken zum Erlernen von Modellen weicher Roboter vor, die physikalische Strukturen integrieren und gleichzeitig Stabilität gewährleisten. Dies erreichen wir, indem wir physikbasierte dynamische Modelle in den Lernalgorithmus einbetten, der die freien Parameter der Dynamik bestimmt und optional eine Koordinatentransformation – etwa eine Kodierung in einen latenten Raum – optimiert. Zwei hervorzuhebende Ansätze werden vorgestellt: (1) ein Algorithmus, der aus Stichproben der Entwicklung der Rückgratform des Roboters niedrigdimensionale Deformationsmodelle extrahiert, und (2) ein Netzwerk gekoppelter harmonischer Oszillatoren zum Erlernen latenter Dynamik aus hochdimensionalen Beobachtungen wie Bildern. Die explizite Einbeziehung von kinematischen und potenziellen Energieanteilen in diese Modelle ermöglicht eine Stabilitätsanalyse mit Standardwerkzeugen der nichtlinearen Systemtheorie, wie z. B. Lyapunov-Methoden. So beweisen wir beispielsweise, dass das Netzwerk gekoppelter Oszillatoren sowohl global asymptotisch stabil als auch Eingangs-Zustands (Input-to-State) stabil ist.

Fünftens nutzen wir die physikalische Struktur der aus Beitrag vier gelernten Modelle, um Regler in geschlossener Form für Sollwertregelung zu entwerfen. Der Regler enthält zwei zentrale Komponenten: (1) einem Vorwärtsregelung-Teil basierend auf Umformung der potenziellen Energie, der das lokale/globale Minimum der geschlossenen Potenzialenergie mithilfe des gelernten Modellwissens am Sollwert positioniert, und (2) einen integralgesättigten PID-Feedback-Term, der Störungen unterdrückt und Modellierungsfehler kompensiert, um stationäre Fehler zu vermeiden. Die Stabilität des geschlossenen Systems kann anschließend mithilfe von Lyapunov-Argumenten analysiert werden.

Schließlich untersucht die Arbeit Methoden zur Erzeugung nachgiebiger Bewegungsverhalten in weichen Robotern über die Regelung auf niedriger Ebene (Low Level) hinaus. Ein Ansatz konzentriert sich darauf, Benutzer, insbesondere ältere Menschen, bei alltäglichen Aktivitäten zu unterstützen, indem der Low-Level-Regler durch Gehirnsignale gesteuert wird. Dies wird erreicht, indem die Klassifizierung der Bewegungsimagination von tragbaren EEG-Geräten mit einer nachgiebigen Impedanzregelung im Arbeitsraum kombiniert wird. Der zweite Ansatz verbindet ein orbital stabiles dynamisches System im latenten Raum mit einem bijektiven, durch ein neuronales Netzwerk parametrisierten Encoder, um periodische Bewegungen aus Demonstrationen zu erlernen. Durch den Verzicht auf Zeitreferenzen ermöglicht diese gelernte Bewegungsstrategie eine natürliche und nachgiebige Verfolgung der demonstrierten periodischen Bewegungen. Dieser Beitrag stellt sicher, dass nicht nur der Roboteraufbau und der Low-Level-Regler nachgiebig sind, sondern auch die Bewegungsstrategie auf hoher Ebene (High-Level).

Zusammenfassend zielt diese Arbeit darauf ab, sicheres und präzises Verhalten weicher Roboter in menschenzentrierten Umgebungen zu ermöglichen. In diesem Kontext wird Sicherheit durch die mechanische Nachgiebigkeit des weichen Körpers erreicht. Um die Präzision zu erhöhen und gleichzeitig Einblick in die Entscheidungsfindung zu behalten, Stabilität sicherzustellen und die Rechenaufwände gering zu halten, besteht der

Kernbeitrag dieser Arbeit in der Entwicklung von Reglerarchitekturen in geschlossener Form für weiche Roboter und ihrer Verknüpfung mit gelernten Modellen. Mehrere unterstützende Beiträge umfassen eine Metrik zur Quantifizierung der Sicherheit weicher Roboter, Strategien zur Integration kinematischer Modelle in Formwahrnehmungsmethoden, fortschrittliche physikbasierte Aktuierungsmodelle, die als zukünftiges Vorwissen für gelernte Modelle dienen können, sowie das Überschreiten von Low-Level-Reglern durch die Entwicklung nachgiebiger Bewegungsstrategien. Alle vorgeschlagenen Modelle, Sensormethoden, Regelungsstrategien und Bewegungsansätze wurden umfassend durch Simulation oder reale Tests an weichen Robotern verifiziert. Der Code und die Daten aus den meisten Kapiteln wurden auf GitHub öffentlich zugänglich gemacht und tragen so zur breiteren Forschungsgemeinschaft bei.

1

Introduction

This chapter situates the dissertation within the broader societal context, addressing the challenges of integrating robotics into human-centered environments and assisting with daily living activities. First, we explain how soft robots, with their inherent safety, can help overcome these challenges despite current performance limitations—particularly regarding precision. Next, we argue that employing modern machine learning methods, with their significant expressiveness, could enable the development of far more accurate dynamic models of the complex behaviors exhibited by soft robots. To maintain transparency in decision-making, ensure stability guarantees, and preserve computational efficiency in the controller, our approach integrates physical structure into the learned models for use in closed-form model-based controllers. We then present the research questions this thesis aims to tackle, along with the corresponding contributions. Finally, we outline the structure of the thesis, connecting each chapter to the relevant topics, contributions, and the strength of physical priors within the employed models.

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q} = \mathcal{Q}$$

1

1.1 Motivation

Over the past few decades, the domain of robotics has made remarkable strides, leading to the widespread adoption of high-performance, precise, and powerful rigid robots (Todd, 1996). While these systems have traditionally excelled in manufacturing environments with repetitive tasks—such as assembly lines—they are increasingly finding use in applications that require more autonomy, including warehouse logistics and automated inspections. Yet, in addressing contemporary societal needs, there is a heightened demand for robots specifically crafted for human-centered settings, e.g., homes and public spaces (Chibani et al., 2013; He et al., 2021; Nahavandi, 2019; Royakkers and Van Est, 2015). Fully realizing the potential of such robots calls for integrating inherent compliance, enabling safe interaction with humans in dynamic, unconstrained, and unpredictable contexts. This requirement aligns with Asimov's First Law (Asimov, 1941) — robots must never harm humans, making safety a paramount consideration in both their design and deployment (Villani et al., 2018).

Existing approaches to ensuring safety largely depend on computational intelligence (Ahn et al., 2024; Sermanet et al., 2025), employing advanced control strategies, e.g., safety filters (Ames et al., 2016) and impedance controllers (Khatib, 1987), real-time collision detection (Haddadin et al., 2017), and predefined safety zones (Zhao et al., 2024b). These methods leverage sophisticated sensing, algorithms, and extensive pre-programming to predict and mitigate potential hazards (Fragapane et al., 2021). For instance, collision detection improves safety by stopping or slowing the robot upon contact. Although this measure helps reduce the risk of serious injury, it is inherently reactive and cannot entirely eliminate harm. Moreover, any sensor malfunction or failure in the perception or decision-making pipeline undermines safety guarantees. Hardware-based solutions have also been developed to enhance safety. For example, collaborative robots (cobots) (El Zaatari et al., 2019), which are designed for safe human-robot interaction, typically incorporate series elastic actuators to isolate actuator dynamics from the robot's links (Pratt and Williamson, 1995) and minimize link inertia (Albu-Schäffer et al., 2007). Still, their rigid structures pose considerable risks to nearby individuals (Haddadin, 2013). To address these dangers, standards such as ISO/TS 15066:2016 (Standard, 2016) mandate that cobots move slowly enough to stop before colliding with a person, thereby constraining performance and effectiveness (Ajoudani et al., 2018; Lucci et al., 2020). Ultimately, current safety measures place heavy demands on perception systems and substantially reduce the efficiency and capabilities of robots and cobots in human-centered environments.

Soft robotics reimagines safety from the ground up by embedding it into the robot's fundamental mechanical design. Rather than treating safety as an add-on or relying solely on computational layers, soft robots incorporate safety through the choice of materials and structural configurations (Laschi et al., 2016; Rus and Tolley, 2015). Their inherent compliance facilitates safe interactions with humans[1] and makes them ideal for safety-critical settings like personal assistance, caregiving, and handling delicate items (Abidi and Cianchetti, 2017; Yumbla et al., 2021). Recent advances in the field have been substan-

[1]To the best of our knowledge, the inherent safety of soft robots has not yet been rigorously quantified, experimentally analyzed, or validated in literature—for instance, through controlled user studies or by comparing them to (collaborative) rigid manipulators. In Chapter 3, we introduce a safety metric as a first step toward quantifying the (added) safety of soft robots.

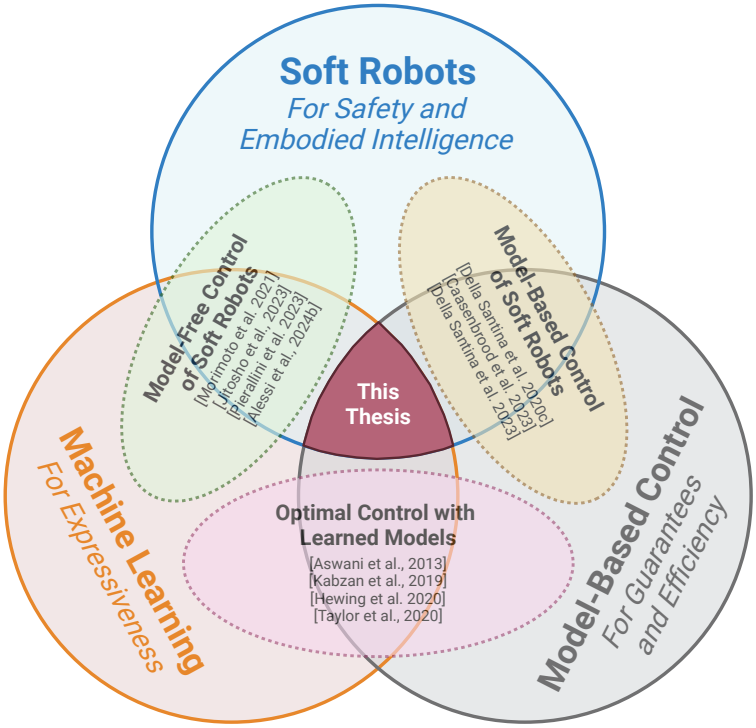


Figure 1.1: Topics covered in this thesis: leveraging learned models for the control of soft robots.

tial (Yasa et al., 2023), with innovations in soft robot designs (Guan et al., 2023; Hawkes et al., 2017; Katzschmann et al., 2018; Laschi et al., 2012; Tolley et al., 2014), smart materials (Mazzolai et al., 2022; Terryn et al., 2017), actuators (Aubin et al., 2022; Gravert et al., 2024; Lipton et al., 2018; Shepherd et al., 2013; Vasios et al., 2020; Wehner et al., 2016), sensors (Larson et al., 2016; Thuruthel et al., 2019; Truby et al., 2020), modeling methods (Boyer et al., 2020; Renda et al., 2020, 2018), and control techniques (Della Santina et al., 2020c; Jitosho et al., 2023; Pustina et al., 2024a; Thuruthel et al., 2018b). Despite this progress and the potential to transform human-robot interaction (Jørgensen et al., 2022), widespread practical adoption of soft robots remains a challenge (Hawkes et al., 2021). In particular, many soft robots still struggle with imprecise, often oscillatory motion (Hawkes et al., 2017; Majidi, 2014; Mazzolai et al., 2022). This is largely because modeling (Armanini et al., 2023) and controlling (Della Santina et al., 2023) continuum soft robots from first principles is inherently difficult; they exhibit infinite degrees of freedom, complex nonlinear dynamics, and time-dependent phenomena such as hysteresis (Armanini et al., 2023), and might make extensive large-area multi-point contact with the environment. In summary, our findings indicate that current soft robots lack sufficient capability—particularly in precision—and, much like safety-aware controllers for rigid robots that enforce cautious, slow motions, we are again sacrificing performance for safety. Therefore, we see a pressing need for novel motor control strategies for soft robots that combine the inherent compliance and

1

embodied intelligence of soft robots with the precise motion characteristics of rigid robotic manipulators.

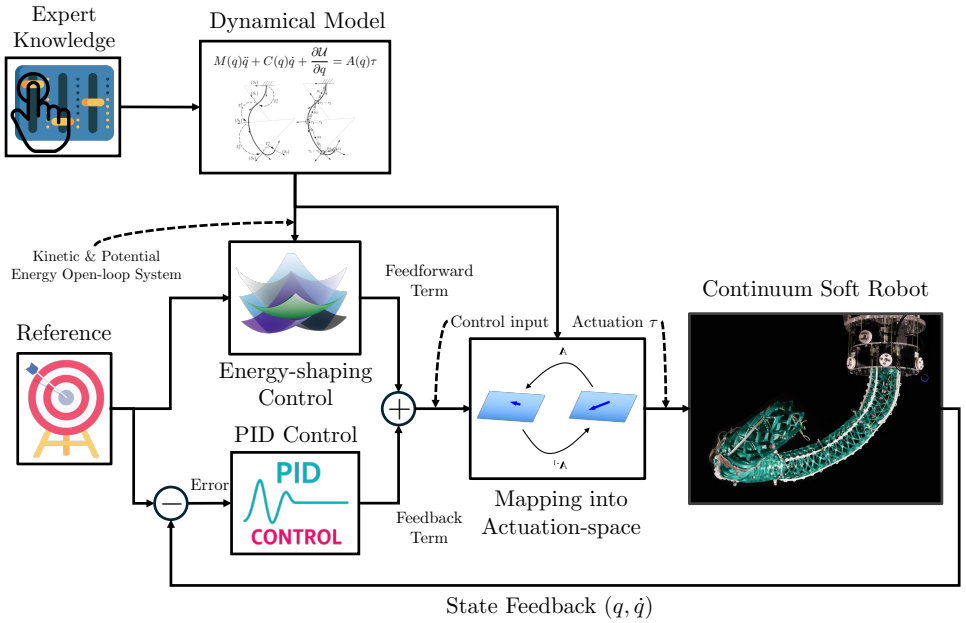


Figure 1.2: **Existing Approach: Model-Based Control with Physical Models.** Expert knowledge is required to derive kinematic and dynamic models for soft robots from first principles (Armanini et al., 2017). The dynamics can often be expressed in a control-oriented Euler-Lagrangian framework. This model knowledge is then integrated into closed-form controllers that combine an error-based feedback term (e.g., a PID-like controller) with a model-based term (e.g., an energy-shaping feedforward) to track a given reference (Caasenbrood and Nijmeijer, 2021; Della Santina et al., 2023, 2020c). Finally, the control input must be translated into an actuation that exerts forces and torques on the continuum soft robot in a closed loop.

At present, two main approaches can be distinguished for controlling the complex behavior of soft robots (Della Santina et al., 2023; Thuruthel et al., 2018a). The first (1), visualized in Fig. 1.2, relies on fully physics-based models (Armanini et al., 2023), for example using strain models (Alessi et al., 2024a), and typically involves nonlinear feedback, e.g., PD+ (Della Santina et al., 2020c), or error-based feedback+feedforward, e.g., PID+energy-shaping (Caasenbrood et al., 2023; Della Santina et al., 2023; Soleti et al., 2025), control schemes. The second (2) approach, visualized in Fig. 1.3, is to learn the controller directly in a model-free fashion (George Thuruthel et al., 2017), for example, through model-free RL (Alessi et al., 2024b; Jitosho et al., 2023; Morimoto et al., 2021) or ILC (Hofer et al., 2019; Pierallini et al., 2023). However, both of these strategies exhibit significant drawbacks.

Developing fully physics-based models, as used in (1), demands substantial expert knowledge (Stella and Hughes, 2023) and often hinges on strong simplifying assumptions, such as the continuum soft robot being slender, i.e., the backbone radius is much smaller relative to its length (Cosserat, 1909), allowing for a low-dimensional kinematic parametrization (Armanini et al., 2023), preserving a constant backbone cross-section (Gazzola et al., 2018), and assuming isotropic Hookean materials without time-dependent effects

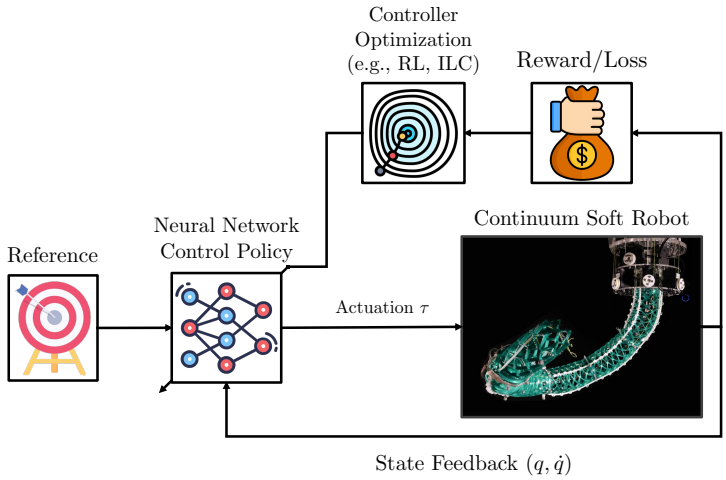


Figure 1.3: **Existing Approach: Direct Controller Learning.** Directly learning the controller via Reinforcement Learning (RL) (Alessi et al., 2024b; Jitosho et al., 2023; Morimoto et al., 2021) or Iterative Learning Control (ILC) (Hofer et al., 2019; Pierallini et al., 2023) by interacting with the soft robotic system and optimizing the control policy based on reward/loss/error signals.

like hysteresis. These assumptions fail to capture the intricate physics of many soft robots, undermining control performance. Furthermore, accurately modeling from physics using only first principles the joint dynamics of a soft robot that interacts intensively with the environment—whether by collaborating with humans, making extensive large-area multi-point contact with its surroundings, or predicting the environment dynamic evaluation in the spirit of world models (Ha and Schmidhuber, 2018)—appears currently infeasible, thereby motivating the need for learning-based solutions. Discussing approach (2): Model-free RL, meanwhile, tends to be sample inefficient and does not guarantee stability, whereas ILC can only be used for learning repetitive trajectories (Bristow et al., 2006). The sample inefficiency is particularly problematic for soft robots since their material properties change over time—rendering previously learned controllers less effective—and their limited lifespans (Yasa et al., 2023) risk the robot being damaged or failing altogether before a controller learned through RL is fully trained. The lack of insight into the decision-making process and the lack of stability guarantees can cause safety issues, particularly when operating in close proximity to humans.

To overcome the limitations of approaches (1) and (2), research in recent years has combined the strengths of ML and model-based control by first learning a model—often using a Neural Network (NN)—and then exploiting it for control (Alora et al., 2023a,b; Bruder et al., 2024, 2020; Chen et al., 2024; Gillespie et al., 2018; Thuruthel et al., 2018b). However, this enhanced model expressiveness through ML techniques comes at the expense of computational efficiency, as closed-form controllers are typically unavailable, and one must rely on optimal control/optimization-based strategies (see Fig. 1.4). For example, learning linear dynamical models—such as those based on a Koopman approach (Bruder et al., 2024, 2020) often proves unstable, suffers from the trade-off between model dimensionality and expressiveness, and cannot accurately represent non-linearizable systems (Cenedese

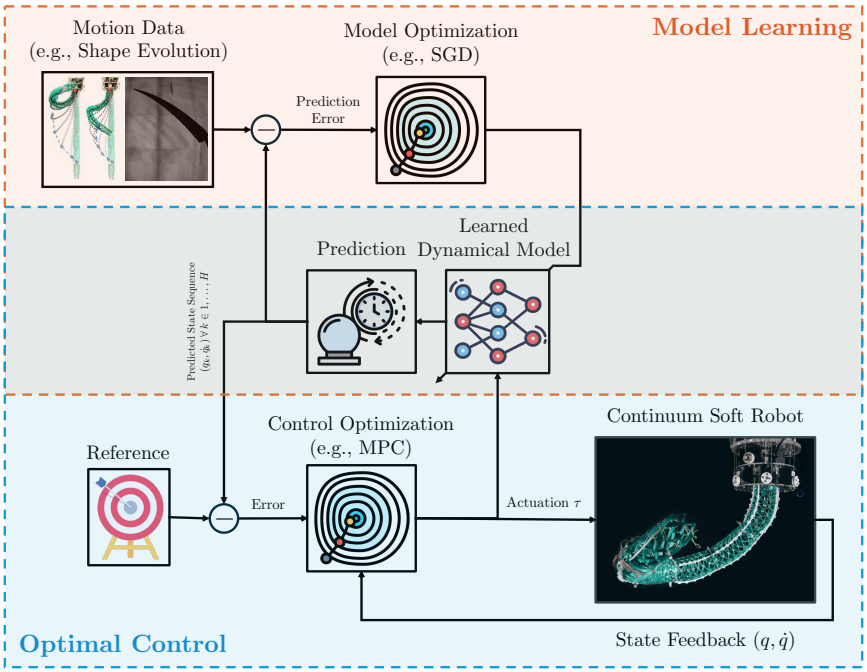


Figure 1.4: **Emerging Approach: Optimal Control with Learned Models.** First, the dynamic model is learned via Machine Learning (ML) approaches (Alora et al., 2023a,b; Chen et al., 2024; Gillespie et al., 2018; Kasaei et al., 2023; Liu et al., 2024a; Marques Monteiro et al., 2024; Xie et al., 2023), and subsequently, the control input is devised via optimal control techniques, such as Linear-Quadratic Regulator (LQR) (Bruder et al., 2024, 2020) or Model Predictive Control (MPC) (Alora et al., 2023a,b; Gillespie et al., 2018; Hewing et al., 2020), by iteratively predicting the evolution of the closed-loop system for a given control input sequence over a horizon H and using the error/loss between the predicted and desired behavior to optimize the control input.

et al., 2022). On the other hand, using nonlinear functions (notably Deep Neural Networks (DNNs), including Multilayer Perceptrons (MLPs), Recurrent Neural Networks (RNNs) (Sun et al., 2022; Thuruthel et al., 2018b), Long Short-Term Memorys (LSTMs) (Xie et al., 2023), or Neural ODEs (NODEs) (Kasaei et al., 2023) provides more expressive models but requires computationally heavy control techniques such as optimal control with MPC (Alora et al., 2023a,b; Aswani et al., 2013; Gillespie et al., 2018; Hewing et al., 2020; Kabzan et al., 2019) or Control Barrier Functions (CBFs) (Taylor et al., 2020), or model-based RL (Thuruthel et al., 2018b) as they typically lack a clear physical structure that would allow, for example, for energy shaping-based or feedback-linearization control (Khalil, 2002) and that would permit stability analyses of both the open-loop and closed-loop system by conventional means, such as Lyapunov arguments (Khalil, 2002). The high computational demand associated with MPC limits the maximum control frequency achievable in practice, thereby restricting the effective control of highly dynamic behaviors.

In this thesis, we seek to circumvent the aforementioned shortcomings by integrating learned models with closed-form control, as visualized in Fig. 1.1. By embedding learned models within established closed-form model-based control strategies, such as nonlinear

feedback (Della Santina et al., 2020c) or error-based feedback+energy shaping (Caasenbrood et al., 2023; Della Santina et al., 2023), as visualized in Fig. 1.5, we benefit from both paradigms: learned models demand less specialized expertise while capturing more complex dynamical behaviors, and the closed-form model-based controller remains computationally very efficient. However, to apply such approaches, it is necessary to inspect the kinetic and potential energies of the learned model—something that existing popular ML architectures like RNNs, MLPs, or NODEs generally do not allow. Consequently, it becomes clear that the learned models must exhibit a physical structure in order to employ common closed-form model-based control schemes and to secure closed-loop stability via Lyapunov arguments.

Recent advances in this direction include the emerging literature on Lagrangian Neural Networks (LNNs) (Cranmer et al., 2020; Liu et al., 2024a; Lutter et al., 2019) and Hamiltonian Neural Networks (HNNs) (Greydanus et al., 2019; Liu et al., 2024a), both of which explicitly learn the system's kinetic and potential energies. Nevertheless, these approaches still present substantial challenges: deriving the Equation of Motion (EOM) necessitates real-time higher-order derivatives, which is computationally demanding—particularly during training, but also at inference—and escalates (GPU) memory usage (Lutter et al., 2019). Moreover, without careful implementation, they do not offer formal stability guarantees, and the computation of higher-order derivatives can cause numerical stiffness or instability (Greydanus et al., 2019) unless managed through precise auto-differentiation (Rumelhart et al., 1986) or meticulous hyperparameter tuning. Finally, there is a clear need for learned models with additional inductive bias tailored to the specific dynamics of soft robots.

In conclusion, this thesis addresses the central challenge of achieving closed-form control of soft robots while maintaining accurate and safe operation. The key research goal is to advance our understanding of the physical structure of soft robots and incorporate it into learned models, thereby enabling the use of effective closed-form controllers and allowing for stability verification via Lyapunov-based methods. In the following section, we will pose the principal research question directly connecting to this identified challenge.

1

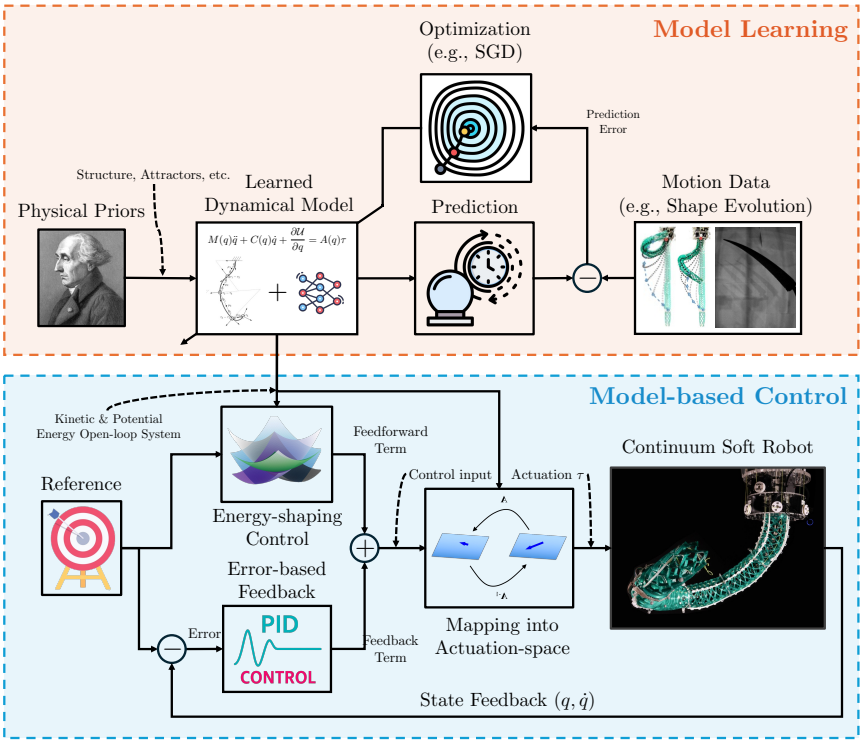


Figure 1.5: **Core Contribution of this Thesis.** The primary contribution of this thesis is the integration of learned models with closed-form controllers. To enable this, we learn dynamical models with a physical structure that allows us to derive the system's potential and kinetic energy. Beyond the specific dynamical model structure, the learning process follows a well-established approach in the literature: we predict the future evolution of the soft robotic system, compare these predictions with actual motion data, and use the resulting prediction error to optimize the free parameters of the learned model. On the control side, we modify the closed-loop system's potential energy by incorporating a feedforward term that steers the system toward a designated reference (Della Santina et al., 2023). This potential shaping feedforward is complemented by an integral-saturated PID feedback term, known as P-satI-D (Pustina et al., 2022a), which rejects disturbances and compensates for modeling errors. Finally, the control input is translated into an actuation signal, where this mapping may also be learned. This insight into the potential and kinetic energy permits an analysis of the stability and convergence properties of both open-loop and closed-loop systems using Lyapunov arguments (Khalil, 2002).

1.2 Research Questions

In this section, we lay out the research questions that guided the research conducted as part of this thesis. The principal Research Question (RQ) that we pose in this thesis is

Principal Research Question

How can we leverage learned models that integrate physical structure to efficiently control soft robots, ensuring both precise and safe behavior?

This principal RQ investigates how we can develop (dynamical) models for soft robots with a physical structure that enables the use of closed-form error-based feedback+model-based feedforward control strategies, thereby addressing the research gap identified in Sec. 1.1. To make answering this principal RQ more tractable, we break it down into several key RQs, which we list below.

Research Question 1. *How do we quantify safety in soft robotics?*

As highlighted in Sec.1.1, a key driver of soft robotics is the belief that their passive compliance enhances or guarantees safety, a critical factor for deploying soft robots in human-centric environments (Mengaldo et al., 2022; Rus and Tolley, 2015). While some studies link material softness to safety using simplified models, such as the Timoshenko beam (Abidi and Cianchetti, 2017), the safety of soft robots has yet to be: a) quantified using a safety metric, and b) compared quantitatively to rigid robots. To justify continued investment in soft robotics research, we argue that the field must begin quantifying the safety benefits of soft robots. Alternatively, framing the issue differently, quantifying safety would allow us to certify that soft robots meet the safety requirements for specific human-centric applications.

The principal RQ mandates that we ensure both precise and safe behavior. While precision can be measured relatively easily (e.g., by assessing the distance to a reference), quantifying safety is considerably more challenging—and this research question is designed to tackle that issue.

Research Question 2. *How can we enhance the robustness and effectiveness of shape sensing for soft robots by leveraging prior (kinematic) knowledge?*

Perceiving its own state (*proprioception*) and the state of the environment (*exterioception*) are two fundamental tasks for any autonomous robot (Thuruthel et al., 2019). Despite its importance, perception algorithms for soft robots have been relatively less investigated.

In this thesis, we specifically focus on *proprioception*, as accurate information about its own shape is vital for any feedback control strategy. In most soft robot control schemes, measuring the configuration space, which is commonly an approximation of the robot's backbone shape, is essential, regardless of the chosen kinematic parametrization. Unlike rigid manipulators, where joint encoders suffice to estimate the robot's state, shape sensing in soft robots introduces unique challenges. Two major factors are (a) the theoretically infinite number of degrees of freedom that would need to be measured and (b) the need to avoid compromising the robot's inherent softness by integrating stiff sensors or electronics.

1

Although the field has made considerable progress in developing new proprioception and shape-sensing methods for soft robots, interpreting sensor readings and mapping them to configuration variables remains a significant hurdle. Notably, existing approaches to proprioception only rarely (Stella et al., 2023b) incorporate prior knowledge, such as a kinematic model, which could make these algorithms more robust and performant. For these reasons, Chapters 4 & 9 investigate how to employ kinematic priors as an inductive bias in shape-sensing algorithms.

Research Question 3. *An aspect of the physical priors that needed a closer look: Which advanced actuation models are necessary for exploiting the dynamic behavior of soft robots, and how can they be utilized for control?*

Over the past decade, soft robotics has seen remarkable advances in both modeling (Armanini et al., 2023) and control (Della Santina et al., 2023). Much of this research has focused on creating expressive yet low-dimensional models (kinematic and dynamic) that accurately capture a soft robot's backbone shape, as such models are central to improving the performance of model-based controllers. In many cases, actuation is introduced into these models via actuation-affine terms in the EOM, typically through a configuration-dependent or constant actuation matrix (Della Santina et al., 2023). We argue, however, that this design decision is too restrictive and that more advanced actuation models are required to fully harness the high dynamic capabilities of soft robots. Indeed, the actuators in soft robots—be they tendon-driven, pneumatically powered, or otherwise—often span large parts of the robot's body or reside in its base, indirectly applying forces and torques that affect multiple Degrees of Freedoms (DOFs). This is fundamentally different from rigid robots, where actuators commonly drive joints directly. Moreover, explicitly modeling the actuator dynamics can reduce response time and improve transient performance. Consequently, in Chapters 5 & 8, we introduce more accurate physics-based actuation models for soft robots, illustrating the concept with both complex auxetic metamaterial-based and pneumatic piston-driven actuators. We also explore how these advanced models can be applied in model-based control schemes, from operational space control (Chapter 6) to backstepping control (Chapter 8). Through this work, we highlight the limitations of purely physics-based approaches in fully capturing a soft robot's dynamic behavior, as well as the extensive domain expertise required to design such models—thereby motivating the study of more expressive, neural network-based alternatives. Additionally, these advanced physics-based models can later serve as an inductive bias when learning data-driven models.

Research Question 4. *Which physical structure should learned models exhibit to allow for provably-stable closed-form model-based control?*

To address this question, we propose learning soft robot models in a data-driven fashion. This approach can increase both expressiveness and accuracy while reducing the amount of expert knowledge needed when deriving models from first principles. Various machine learning techniques already exist for this purpose (Armanini et al., 2023; Chen et al., 2024; Kim et al., 2021), including neural networks (Thuruthel et al., 2017) (e.g., MLPs, RNNs (Schäfke et al., 2024), NODEs (Chen et al., 2018; Kidger, 2021)) and Gaussian Process (GP)-based methods (Sabelhaus and Majidi, 2021). Yet most of these

models lack a (physical) structure, preventing us from applying efficient closed-form model-based control methods or from conducting a formal stability analysis of the closed-loop system. As a result, approaches such as MPC (Alora et al., 2023b; Gillespie et al., 2018; Schäfke et al., 2024), optimal control, or RL (Thuruthel et al., 2018b) are often necessary for exploiting the learned models for control, coming with the cost of computational overhead, limiting the maximum control frequencies. Recent research has taken steps toward incorporating physical structure into learned models. For example, Koopman operator theory learns linearized dynamics, permitting LQR-based control (Bruder et al., 2020), and LNNs/HNNs (Lutter et al., 2019) learn a system's energy function for use in control or stability analysis. However, more specialized priors are still needed to reflect the unique dynamics of soft robots. Accordingly, in Chapters 10 & 11, we examine how to learn models with embedded physical structures that enable closed-form model-based control strategies, such as energy-shaping feedforward control augmented by error-based feedback terms. This endeavor directly supports the principal RQ of this thesis.

Research Question 5. *How can we generate compliant and precise motion behaviors for soft robots?*

A compliant structure, by virtue of the robot's soft materials, together with a compliant low-level controller, e.g., impedance control, does not inherently guarantee safe or compliant motion in human-centric settings. High-level motion policies that provide references to the low-level controller may still introduce unstable, unnatural, or unsafe motions. Furthermore, developing motion policies specifically suited to soft robots is, on the whole, an underexplored field. To address this, Chapters 7 & 12 introduce methods aimed at producing references that promote compliant, natural, stable, and safe motion. Specifically, Chapter 7 investigates Brain-Machine Interfaces (BMIs), and Chapter 12 examines learning motion policies from demonstration to supply setpoints to the low-level controller. This work contributes to the principal RQ by fostering a safe, holistic approach—beyond mere low-level control—to operating soft robots.

1.3 Thesis Contributions

In the following, we will detail the key contributions made by this dissertation. We start with the core contribution, which addresses the *principal RQ*.

Core Contribution

Safe and Precise Soft Robots via Closed-form Control with Learned Models

In this thesis, we introduce two approaches for learning kinematic and dynamic models of soft robots that incorporate a physical structure (discussed in more detail under Contribution IV). Because these learned models encode well-defined kinetic and potential energy terms, we can leverage established error-based feedback+energy-shaping control strategies originally designed for rigid manipulators (Kelly, 1995, 1998; Kelly and Carelli, 1996; Sciavicco and Siciliano, 2012). Specifically, we combine a potential-shaping feedforward term (Della Santina et al., 2023) with an integral-saturated PID (Pustina et al., 2022a)

for feedback. This setup allows us to utilize the learned model within the feedforward component while still handling disturbances and compensating for modeling inaccuracies via the feedback term. Moreover, instead of relying on computationally expensive methods such as RL or MPC, which are often necessary for learned models, our controller provides a closed-form solution. This makes it possible to analyze the characteristics of the closed-loop system and formally prove stability using Lyapunov arguments (Khalil, 2002), as already demonstrated for the open-loop case (Stölzle and Della Santina, 2024). These stability guarantees—supported by the interpretability of the learned model—constitute an initial step toward *safe control*. To further promote and quantify safety, we i) propose a safety metric that evaluates potential injury severity from the closed-loop system dynamics and specifically establishes bounds on the proportional feedback gains to ensure safety requirements and ii) develop stable high-level strategies to generate compliant motion behaviors by providing references/setpoints to the low-level controller. Experimental verification of the contributions is an important aspect of this thesis, and we present some of the (soft) robot embodiments used in Fig. 1.6.

This core contribution is enabled by multiple key contributions, which we detail in the following.

Contribution I. *A First Quantitative Safety Metric for Soft Robots*

To address RQ 1, we introduce the first quantitative safety metric for soft robots in Chapter 3. We begin by examining possible uses of such a metric in design, control, and certification. From there, we define a set of requirements that the metric should ideally satisfy to be suitable for these applications. Building on the existing, standardized criterion for collaborative robots (Standard, 2016)—which uses maximum contact pressure during a collision with a human as a proxy for injury severity—we develop a metric that considers the distinctive features of soft robots, including their continuous deformability, elasticity, and the potential for contact along the entire robot body. The proposed safety metric has two variants: the Soft Robot Injury Severity Criterions (SRISCs) models injury severity for a given contact geometry, robot configuration, and velocity, while the Soft Robot Design Hazardousness Criterions (SRDHCs) assesses how safe a soft robotic design is under certain operating conditions (e.g., maximum velocity and actuation torques).

This contribution outlines a pathway for quantifying, optimizing, ensuring through control, and ultimately certifying the safety of soft robots—a central objective of this thesis. Moreover, it highlights the significant impact controllers have on the safety of the closed-loop system, an aspect often overlooked in soft robotics. Consequently, it guides us toward designing compliant controllers that maintain safety, for instance, by eliminating integral terms and minimizing feedback gains, while instead relying on effective feedforward strategies based on an accurate dynamic model of soft robots.

Contribution II. *Leveraging Kinematic Models for Soft Robot Shape Sensing*

This contribution, which addresses RQ 2, improves the robustness and accuracy of shape sensing in soft robots by harnessing existing kinematic model knowledge. We achieve this by solving nonlinear optimization problems that align sensor measurements with the backbone shapes predicted by the kinematic model. Enhanced state estimation is critical for deploying feedback controllers, thus directly supporting the thesis's core contribution. Below, we outline two distinct approaches.

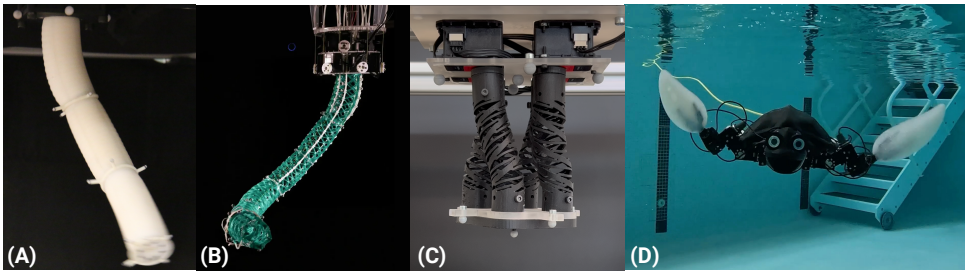


Figure 1.6: Soft robot prototypes were used for experimental validation of the methods proposed in this thesis. **Panel (A):** A three-segment pneumatic continuum soft robot made of silicon with three air cavities per segment. The picture is taken from Katzschmann et al. (2019). **Panel (B):** A tendon-driven continuum soft robot consisting of five segments based on a helicoid structure, coined Helix Robot (Guan et al., 2023). **Panel (C):** An Handed Shearing Auxetic (HSA) soft robot based on 3D-printed auxetic metamaterials driven by four servo motors (Lipton et al., 2018; Truby et al., 2021). **Panel (D):** A hybrid soft-rigid turtle robot combining a rigid body with limbs consisting of three servos and a rigid-soft flipper.

The first method presented in Chapter 4 combines monocular cameras embedded into the soft robot surveilling the environment with visual SLAM (vSLAM) and a projection onto the kinematic model. Specifically, the vSLAM algorithm estimates multiple poses along the soft robot's backbone based on the camera images. We then solve an optimization problem that projects these pose measurements onto the kinematic model, producing configuration estimates. This strategy mitigates estimation drift and preserves consistency between pose estimates and the kinematic model. We verify the algorithm both in simulation and on a real soft segment equipped with a low-cost Raspberry Pi camera. Its success stems from blending four well-established elements: (1) low-cost monocular cameras, (2) vSLAM, (3) the Piecewise Constant Curvature (PCC) (Webster III and Jones, 2010) kinematic model, and (4) Levenberg-Marquardt optimization (Levenberg, 1944; Marquardt, 1963). Future work could further boost performance simply by, for example, using more advanced vSLAM methods or more expressive kinematic parametrizations such as Geometric Variable Strain (GVS) (Renda et al., 2020).

The second method, included in Chapter 9, centers on shape proprioception with magnetic sensors. We embed multiple magnets and magnetic sensors in the robot, which yields complex 3D magnetic fields that are not sufficiently modeled by purely physics-based approaches. Consequently, we combine modern ML techniques with a physics-based kinematic model rather than learning the entire sensor-to-configuration mapping in an end-to-end fashion. Specifically, the kinematic model represents the backbone shape and parameterizes the spatial relationship between each sensor and magnet using just a few variables. We then train a neural network to predict sensor readings based on these parameters. During inference, we solve an optimization problem via gradient descent that matches the actual sensor outputs to the model's predictions, thereby recovering the robot's shape. This approach illustrates how incorporating a known kinematic model reduces the size of the *black-box* that needs to be learned in a data-driven approach.

Contribution III. *Soft Robot Control with Advanced Physics-Based Actuation Models*

We address RQ 3 by deriving advanced actuation models from first principles and then

embedding them in model-based controllers. Two case studies guide our work: HSA robots and pneumatic piston-driven soft robots. These advanced actuation models feed into the thesis's core contribution by informing us about the available physical priors for learning models.

HSA Robots. HSAs robot (Chin et al., 2018; Lipton et al., 2018) typically feature several HSA rods arranged in parallel and joined at the tip. Applying torsional torque at the base (e.g., via a servo motor) twists the rods, which in turn elongates and/or bends the robot through the rod's auxetic metamaterial. This complex actuation mechanism can also induce twisting along the robot's length, a rare deformation for soft robots. Despite ongoing research into new designs (Good et al., 2025, 2022), fabrication processes (Truby et al., 2021), and proprioception (Zhang et al., 2022a), accurate models (Garg et al., 2022) and control methods are lacking. In Chapters 5 and 6, we propose and experimentally validate various models and controllers for HSA robots. Specifically, Chapter 5 introduces a suite of kinematic, dynamic, and actuation models. First, we develop the Selective Piecewise Constant Strain (SPCS) model to parameterize the 3D shape of HSA rods minimally by fusing the Constant Strain (CS) and Piecewise Constant Strain (PCS) approaches (Renda et al., 2018). Next, we derive an actuation model that captures how the rod's stiffness and rest length vary with twist angle, allowing us to integrate this phenomenon into existing Discretized Cosserat Rod Model (DCM)-based simulators. We then implement our findings in a PyElastica (Naughton et al., 2021) extension, enabling 3D simulations of HSA robots. For planar HSA robots, we detail a simpler kinematic model based on CS, culminating in a closed-form inverse kinematics solution. Finally, we present a dynamic model in Euler-Lagrange form suitable for planar HSA manipulators. Chapter 6 leverages these models to develop two control schemes. The first is a configuration-space P-satI-D plus potential-shaping regulator, integrating an integral-saturated PID controller (Pustina et al., 2022a) with force compensation at the setpoint to guide the end-effector to a desired location in task space. In the future, a strategy that renders a complex, underactuated, and non-affine control term more amenable to control may also prove valuable when developing closed-form controllers based on learned models—for instance, if the latent space exceeds the dimension of the actuation space. The second is an operational space impedance controller that cancels the soft robot's inherent dynamics and uses a Cartesian PD feedback term to shape the desired stiffness field. Key challenges include mapping generalized torques into the underactuated control input, linearizing actuation terms, and solving the static inversion problem (Della Santina, 2025).

Pneumatic Actuation Dynamics & Backstepping Control. Pneumatic actuators are popular in soft robotics for their quick response and distributed force generation (Marchese et al., 2015; Zaidi et al., 2021), with some variants employing fluidic pistons (Malas et al., 2024; Marchese et al., 2014; Marchese and Rus, 2016; Parlikar et al., 2024). Such piston-driven actuation has at least two advantages: (1) the robot's deformation is directly related to the piston position, and (2) they are closed fluid systems allowing easy use of the ideal gas law (Marchese and Rus, 2016). However, relatively little work has been done on modeling these systems (Marchese et al., 2014; Xavier et al., 2020), and existing controllers often adopt a cascaded approach that does not consider actuator dynamics, as a PID controller

unaware of the soft robot dynamics is usually employed to control the piston. This is viable only if the fluidic system can quickly achieve pressure setpoints, necessitating high-performance (often costly) actuators or slow robot motions. Chapter 8 addresses this limitation by modeling the potential energy stored in the fluid as a function of the robot's configuration and the piston position, thereby coupling the piston and soft robot dynamics in Euler-Lagrange form. We then apply a backstepping approach (Khalil, 2002; Kokotovic, 1992; Lozano et al., 1992) to design a nonlinear, model-based feedback controller for the entire system, enabling tighter integration of actuator and robot dynamics. In the future, such an actuation model could serve as a structural prior when learning models, such as learning the potential energy stored in the fluid analog to LNNs (Lutter et al., 2019), therefore connecting to the core contribution. Furthermore, applying the backstepping procedure to a learned model appears to be an unexplored but interesting research avenue.

Contribution IV. *Integrating Physical Structure and Stability Guarantees into Learned Models*

In this thesis, addressing RQ 4, we propose two approaches for learning control-oriented dynamical models of soft robots that incorporate the necessary (physical) structure to facilitate closed-form control through energy shaping, as well as stability analysis of both the open- and closed-loop systems using Lyapunov arguments (Khalil, 2002). We achieve this by incorporating physics-based dynamical models into the learning algorithm, which then identifies the free parameters of the dynamics and optionally optimizes over a change of coordinates—such as encoding into latent space. We detail the two approaches below.

In Chapter 10, we introduce an algorithm designed to learn a soft robot strain model, specifically a PCS-based model, directly from data representing the shape evolution of the soft robot's backbone. Although PCS (Renda et al., 2018) and similar kinematic parametrizations (Alessi et al., 2024a) are well-established in the literature, their design often requires significant expert knowledge and/or trial-and-error. For instance, the modeling engineer must determine the number of segments, the length of each segment, and the active strains for each segment. Additionally, identifying dynamic system parameters through nonlinear optimization can be computationally intensive and potentially ill-conditioned. Our proposed approach, comprising two algorithms, circumvents these challenges by directly identifying all necessary parameters from pose samples approximating the soft robot's shape evolution. The first algorithm uses an iterative procedure to determine an appropriate kinematic parameterization, including the number and length of the segments, by analyzing the strain progression along the backbone. The second algorithm then derives a physics-based dynamical model from first principles using the previously identified kinematic parameterization, representing the Euler-Lagrangian dynamics as a sum of basis functions. This enables dynamic parameter identification in closed form through linear least-squares. Additionally, the algorithm employs heuristics, such as identified strain stiffness, to neglect certain strains, iteratively reducing the DOF of the model, thereby improving efficiency and control suitability. We validate the proposed approach through simulations and benchmark its performance in predicting the soft robot's future shape evolution over long horizons against State of the Art (SOTA) ML methods, including RNNs, NODEs, and others.

The second approach, presented in Chapter 11, considers the problem setting of learning the dynamics of physical systems, and specifically, of continuum soft robots from high-

dimensional observations, such as sequences of images. As learning the dynamics directly in image space is considered to be intractable, an established procedure is to employ an autoencoder, such as a Variational Autoencoder (VAE) (Kingma and Welling, 2014), and then learning the dynamics of a lower-dimensional space, referred to as the latent space. Contrary to existing literature, which mostly uses MLPs, RNNs, or NODEs, we propose to learn the latent dynamics with a network of coupled oscillators that consists of damped harmonic oscillators that are connected by a nonlinear potential. Crucially, this now allows us to assign a mechanical interpretation to each latent variable - the position of one of the harmonic oscillators. Furthermore, the physical structure of the Coupled Oscillator Network (CON) dynamics, in particular the network's energy terms, allows us to derive very strong stability guarantees, such as Global Asymptotic Stability (GAS) for the unactuated and Input-to-State Stability (ISS) for the actuated latent dynamical system, using Lyapunov arguments (Khalil, 2002). Finally, we propose an approximated closed-form solution to the rollout of the Coupled Oscillator Network (CON) dynamics that enables a speed-up of both training and inference. We benchmark the proposed approach against SOTA ML approaches extensively on rendered image sequences of various simulated systems, including different simulated soft robots.

Contribution V. *Exploiting Learned Models for Closed-Form Model-Based Control*

Additionally contributing to RQ4 and building on Contribution V, we here propose a closed-form regulation strategy based on learned models. In doing so, we draw on seminal work that introduced closed-form model-based controllers for rigid manipulators (Kelly, 1997, 1998; Kelly and Carelli, 1996; Sciavicco and Siciliano, 2012), subsequently adapted for the physics-based control of soft robots by considering both linear (Della Santina et al., 2023, 2020c) and, as shown in Chapter 6, nonlinear elastic forces (Borja et al., 2022). The controller combines an integral-saturated PID (P-satI-D) (Pustina et al., 2022a) with an energy-shaping feedforward term. We motivate the considerations behind this strategy in the following.

For the error-based feedback component, we use a P-satI-D controller (Pustina et al., 2022a), which offers stronger robustness against modeling errors via its integral action while saturation reduces the risk of instability (Pustina et al., 2022a). Compared to PD feedback laws (Della Santina et al., 2020c), this approach is more effective at mitigating unmodeled or wrongly modeled dynamics. A key assumption—akin to purely physics-based models—is that the state space must be continuous. Consequently, specialized methods (Maithripala and Berg, 2015) are necessary for other geometric spaces (e.g., Lie groups), where continuity in this sense does not hold directly.

For the feedforward portion, we compensate the potential forces at the target state, requiring the potential field to be (locally) convex (Borja et al., 2022; Della Santina et al., 2023). The aim is to rely on the integral action as little as possible, effectively reshaping the closed-loop system's potential energy so its (local) minimum is at the desired setpoint. Additionally, the control must be able to reach the goal, which requires the setpoint to be a reachable equilibrium of the closed-loop system. In the underactuated case, it is also crucial for the null dynamics to be asymptotically stable in order to guarantee a stable control response (Borja et al., 2022). Compensating for the dynamics—specifically the static forces—at the target state, rather than canceling them at the current state, enhances

robustness against mismatches between the learned model and the actual dynamics, as unmodeled dynamics are less likely to dominate in the closed-loop system.

This combination of integral-saturated PID feedback action and potential shaping feedforward action yields a straightforward yet powerful control approach. Having already applied this P-satI-D+potential shaping strategy to a fully physics-based HSA model as part of Contribution III, we further validate it in simulation with the learned strain-based model in Chapter 10 and for latent space control using CON dynamics in Chapter 11.

Contribution VI. *Beyond Low-Level Control: Generating Compliant Motion Behaviors for Soft Robots*

Turning to RQ 5, we propose two methods for generating compliant soft robot motion at a higher control level. Chapter 7 covers a BMI-based protocol that uses wearable Electroencephalography (EEG) to supply references to a low-level impedance controller, while Chapter 12 focuses on learning motion policies from demonstration via a dynamical system. These strategies ensure that high-level policies do not undermine the inherent safety offered by the soft robot body and compliant low-level controllers.

Guiding Soft Robots via Motor Imagery. BMIs allow users to operate machines through neural signals—particularly appealing for assisting individuals with limited mobility. Yet classification accuracies for motor imagery on wearable EEG devices are low for more than two classes (Arpaia et al., 2022; Lee et al., 2024), raising safety issues if used with fast, high-inertia rigid robots. Soft robots, by contrast, offer a passive compliance that can reduce risks from erroneous BMI commands. However, controlling soft robots with BMI has not yet been explored. In Chapter 7, we introduce a brain-computer interface protocol that, for the first time, lets users operate a soft robotic manipulator using motor imagery. We employ a wearable, three-channel EEG cap, aiming to in the future transition brain-controlled manipulations beyond traditional lab settings. Since robust motor imagery classification beyond two classes remains challenging, we devised an effective protocol that combines two parallel binary classifiers: one to select the active coordinate axis (based on yaw clinching) and another to determine movement direction along this axis (based on motor imagery). We validate our approach on a planar HSA robot, where the BMI-derived commands serve as setpoints for a compliant impedance controller (developed in Contribution III). In a quantitative evaluation of the brain signal-driven control, we visually project step-function-based operational space goals onto a screen behind the HSA robot. For benchmarking, we compare this BMI system against (1) a keyboard interface and (2) a privileged scenario where the low-level controller has direct access to goal positions. Additionally, we show how this scheme can assist with a real-world Activities of Daily Living (ADL) task—guiding the robot's end-effector with brain signals to spray hairspray—highlighting the synergy between the soft robot's compliance, a suitable low-level controller, and higher-level BMI-based commands.

Learning Stable Period Motions from Demonstration. Dynamic Motion Primitives (DMPs) represent a well-known framework for learning complex motions from demonstrations (Ijspeert et al., 2013; Saveriano et al., 2023). In soft robotics, adopting time-invariant DMPs is particularly appealing for ensuring compliance under perturbations (Ijspeert et al.,

2013), yet this approach remains unexplored. In Chapter 12, we present a method for learning periodic motions from demonstrations and reproducing them in a provably stable and compliant manner—without explicit time dependence. Our framework extends Stable Motion Primitive (SMP) (Ijspeert et al., 2013; Pérez-Dattari and Kober, 2023; Rana et al., 2020) by combining a bijective encoder with latent dynamics governed by a supercritical Hopf bifurcation. The Hopf system's simple latent dynamics facilitate stability proofs, while the diffeomorphism introduced by Euclideanizing flows (Dinh et al., 2016; Rana et al., 2020) enables us to learn complex motions and transfer stability guarantees back to the original demonstration space. Practically, we encode the robot's current position into the latent space, evaluate the Hopf system to obtain a latent velocity, and then map that velocity back to the demonstration space via the encoder's analytical inverse Jacobian—yielding a velocity reference for a low-level motor controller, such as the configuration or operational space controllers presented in Chapter 6. We conduct extensive experiments with diverse systems: a helicoid soft robot (Guan et al., 2023), a swimming turtle robot, a UR5 manipulator for whiteboard cleaning, and a KUKA cobot for tasks involving human contact. The results demonstrate that our approach can learn intricate motion patterns while preserving compliance, making it a promising path forward for soft robot motion planning.

1.4 Thesis Outline

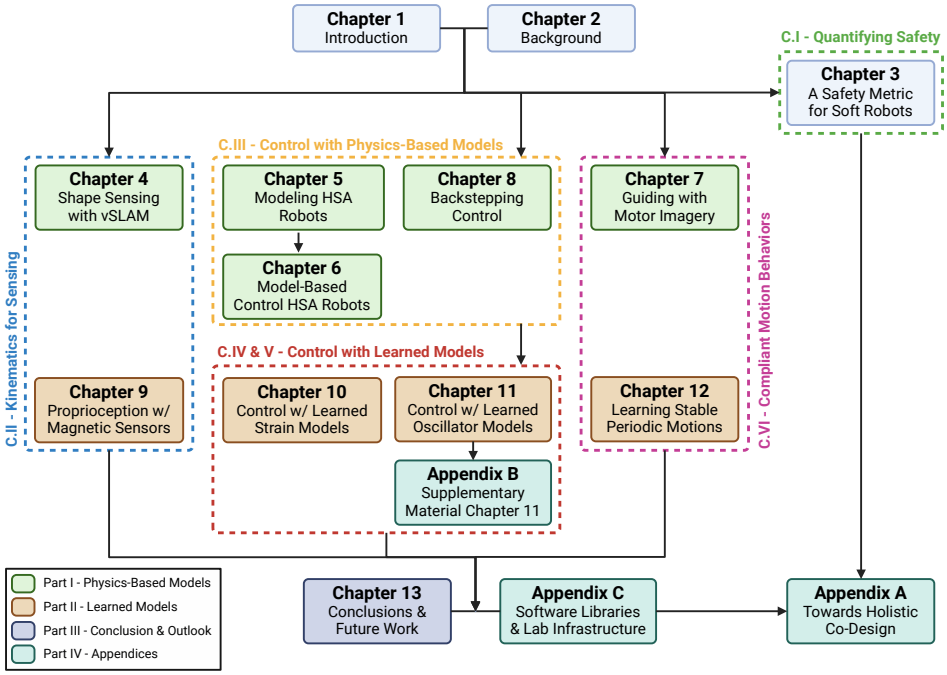


Figure 1.7: Outline of the thesis emphasizing its contributions. The block background colors denote the various parts of the thesis, while *C.X* refers to the *X*th contribution. Dashed frames highlight the chapters corresponding to each contribution.

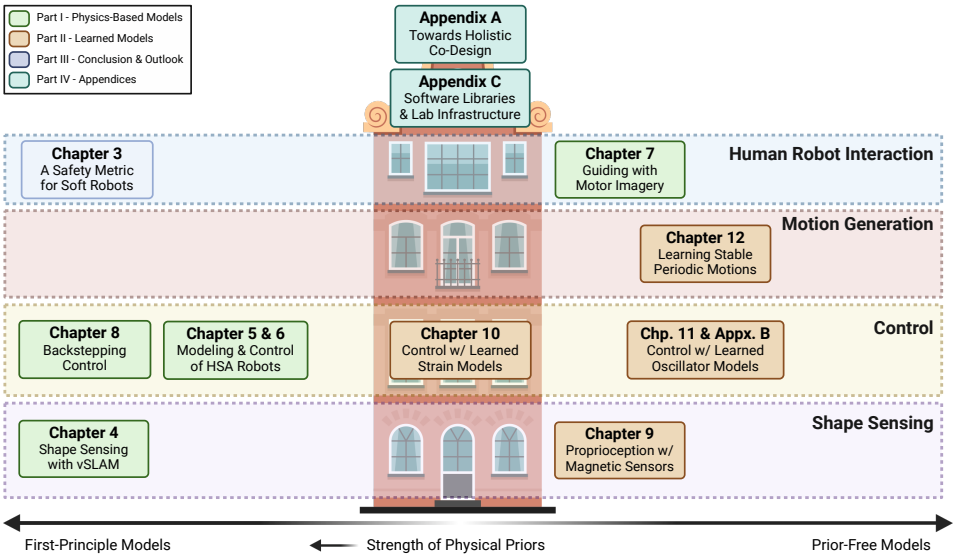


Figure 1.8: Outline of this thesis, highlighting the models' physical priors and their application areas, including proprioception/shape sensing, control, motion generation/policies, and Human Robot Interface (HRI).

In the following, we will outline the structure of this thesis, which is also visualized in Figs. 1.7-1.8.

Chapter 2 aims to provide background on the existing literature on modeling and control of soft robots and introduce specific kinematic and dynamic models, model-based controllers, and other concepts that are recurring throughout this thesis. For example, this includes the PCS kinematic parametrization, Euler-Lagrangian dynamic models of soft robots, and P-satI-D (Pustina et al., 2022a)+potential shaping (Della Santina et al., 2023) controllers.

Chapter 3, incorporating Contribution I, analyzes the inherent safety of soft robots and recognizes the need for a quantitative safety metric that encompasses both the embodied and computational intelligence of the robotic system. After identifying interesting applications of a safety metric, we devise the requirements such a metric would need to meet. We then propose the first quantitative safety metric for soft robotic manipulators. Finally, we give recommendations for the safe design of soft robotic systems.

At the outset of this thesis, our understanding of several fundamental aspects of soft robot behavior was still limited. For instance, we lacked detailed insights into actuation dynamics (Della Santina et al., 2023)—such as the coupled behavior between soft robots and pneumatic piston actuators or how power is transmitted from electric actuators through auxetic metamaterials to induce deformations in HSA robots—as well as in kinematic parameterizations that allow for an optimal tradeoff between dimensionality and expressiveness - as ones that maintain specific strains, such as bending, twist, axial, either constant or piecewise constant along the backbone. Consequently, we devoted a significant portion of this thesis to advancing our understanding by developing physics-based models, experimentally validating them, and leveraging the results to enhance shape sensing and model-based

control. This work is presented in Part I, where we introduce advanced physics-based models for these two applications.

This deeper insight into soft robot characteristics has equipped us to propose more suitable physical priors for integration into learned models. Specifically, Part II focuses on incorporating priors related to kinematics, stability, and physical structure into ML-based methods, thereby employing these hybrid models for improved shape sensing and control.

Parts I & II are discussed in further detail in the following subsections.

Part I - Shape Sensing and Control with Advanced Physics-based Models

In Part I, we develop advanced physics-based models for soft robots and apply them to shape sensing and control. Specifically, we: i) enhance shape sensing based on visual sensors by leveraging pose estimates from Simultaneous Localization and Mapping (SLAM) algorithms and projecting them onto a kinematic model, ii) propose kinematic and dynamic models for HSA robots and utilize them for model-based control, iii) introduce a motor-imagery-based Brain-Computer Interface (BCI) interface for guiding a low-level impedance controller with brain signals, and iv) model the actuation dynamics of pneumatic piston-driven soft robots, designing a provably stable backstepping controller for piston motion regulation. This part includes Contributions II-V and comprises the following chapters:

- **Chapter 4** describes a method for combining low-cost monocular cameras with vSLAM algorithms and projecting pose estimates onto the soft robot's kinematics to enable shape sensing. The approach is validated through simulations using a Constant Curvature (CC) model and experimental tests on a pneumatic soft robot segment moving in 3D space.

- **Chapter 5** presents kinematic, dynamic, and actuation models for HSA robots comprising parallel auxetic metamaterial rods. First, we introduce a novel kinematic parametrization (SPCS) based on the PCS (Renda et al., 2018) model, effectively capturing HSA rod shapes with minimal configuration variables. Next, we formalize the inclusion of auxetic trajectories (e.g., HSA rods changing length with twist strains) in Cosserat rod-based models, enabling a PyElastica (Naughton et al., 2021)-based simulator for HSA robots. Additionally, we devise kinematic and underactuated dynamic models for planar HSA robots, incorporating auxetic trajectories that influence rest length and stiffness. These models are experimentally validated.

- **Chapter 6** extends Chapter 5 by proposing two model-based control strategies for setpoint regulation with planar HSA robots: a) a configuration-space controller using an integral-saturated PID feedback term and potential shaping feedforward term, and b) an operational space impedance controller that enables end-effector stiffness shaping in Cartesian space through a PD feedback term. The configuration-space controller leverages a feedforward term evaluating potential forces at desired end-effector positions via static inversion and a feedback term enabled by linearization of the actuation term and mapping into collocated form. The operational space controller uses model knowledge to cancel the existing soft robot dynamics and applies a least-squares optimizer to solve the underactuated control problem (i.e.,

mapping configuration space torques to actuation inputs). Both approaches are extensively validated experimentally and benchmarked against a model-free PID controller.

- **Chapter 7** enhances the operational space impedance controller from Chapter 6 with a BMI protocol, enabling users to control a soft robot through motor imagery. The protocol processes wearable EEG device measurements, classifies motor imagery using Linear Discriminant Analysis (LDA), and translates the output into incremental end-effector attractor movements. The low-level impedance controller then tracks these setpoints. Experiments with a planar HSA robot compare this approach's performance on a reference trajectory against two baselines: a computational control with privileged access to goal positions and a keyboard-based HRI interface. Finally, we demonstrate a simple ADL—releasing hairspray by guiding the soft robot to press a button using motor imagery.

- **Chapter 8** develops dynamic models of the actuation dynamics for pneumatic piston-driven soft robots and designs a nonlinear backstepping controller (Khalil, 2002; Kokotovic, 1992; Lozano et al., 1992). The model calculates the potential energy stored in the fluid as a function of the robot's configuration and piston position, with partial derivatives representing the potential forces exerted by the fluid. The backstepping controller integrates a configuration-space setpoint regulator into a piston feedback controller. This approach is tested through simulations on a three-segment PCC soft robot.

Part II - Incorporating Physical Structure and Stability Guarantees into Learned Models and Controllers

Part II combines several research efforts integrating modern ML techniques with physics-based models. The goal is to simplify the learning process, provide insights into the learned components through physical interpretation, ensure stability guarantees, and/or leverage physical structures for model-based control using energy-shaping approaches. This part covers Contributions II,IV-VI and includes the following chapters:

- **Chapter 9** presents a methodology for achieving 3D shape sensing in soft robots by embedding multiple magnets and magnetic sensors within their body. The primary challenge with magnetic sensors lies in interpreting their measurements and mapping them to a configuration estimate. To address this, we use the kinematic model of the soft robot to parameterize the spatial relationship between a magnetic sensor and its surrounding magnets. We then learn the forward measurement model based on this low-dimensional parameterization to predict the measurements for each sensor. Notably, by incorporating knowledge of the kinematic model and reducing the complexity of the learning *black-box*, the resulting sensor measurement model generalizes effectively across all embedded magnetic sensors. Finally, to achieve proprioception, we formulate an optimization problem where the cost function minimizes the squared error between predicted and actual sensor measurements. With both the kinematic model and the neural sensor measurement predictor being analytically differentiable, this optimization problem is efficiently solved using gra-

dient descent. The proposed approach is validated through simulations with PCC and Piecewise Affine Curvature (PAC) soft robots, as well as experiments with a pneumatically actuated soft segment.

- **Chapter 10** introduces an algorithmic framework for identifying low-dimensional strain models from data describing the shape evolution of soft robots. The approach takes as input pose measurements along the soft robot's backbone and consists of two key steps: First, a suitable low-dimensional PCS-based parameterization is identified through an iterative procedure to approximate the backbone shape accurately. Next, a dynamical model is derived from this kinematic parameterization, with its parameters regressed in closed form. Additionally, the algorithm employs heuristics, such as the identified stiffness, to determine whether certain strains can be neglected, potentially further reducing the DOF of the dynamical model. The proposed method is validated by deriving the dynamical model of a simulated soft robot directly from video data. Its performance is quantitatively benchmarked against SOTA ML approaches, including RNNs and NODEs. Results demonstrate that this fully physics-based strain model significantly outperforms others in extrapolation settings, particularly for actuation sequences outside the training set. Finally, the learned model is employed for model-based control using a controller that incorporates an integral-saturated PID feedback term alongside a potential-shaping feedforward term, similar to the control approach outlined in Chapter 6.

- **Chapter 11** introduces a method for learning the latent dynamics of physical systems, such as soft robots, using Coupled Oscillator Networks (CONs). Specifically, high-dimensional observations, like images of the soft robot, are mapped into a low-dimensional latent space using a learned autoencoder. Each latent variable in this space corresponds to the position of a mechanical oscillator, providing a clear mechanical interpretation. A key advantage of this approach is that the CON formulation offers strong stability guarantees: the unactuated system exhibits Global Asymptotic Stability (GAS), while we prove for the actuated system Input-to-State Stability (ISS) (Khalil, 2002). The proposed learning method is validated on multiple datasets comprising image sequences of simulated soft robot motions, and its motion prediction accuracy is benchmarked against various SOTA ML approaches, such as Coupled Oscillatory Recurrent Neural Network (coRNN) and Gated Recurrent Unit (GRU). Furthermore, the physical structure of the CON dynamics is leveraged for closed-form model-based feedback control in the latent space. This involves using an integral-saturated PID feedback term and reshaping the CON potential field to exhibit its minimum at the desired latent goal. The proposed control strategy is demonstrated on a simulated two-segment PCC soft robot.

- **Chapter 12** proposes a method for learning orbitally stable motion policies from demonstration, which can serve as a reference for the low-level motor controllers introduced in Chapters 6 and 10–11. Building on pioneering research in Stable Motion Primitive (SMP), which parameterizes motion policies using (latent) dynamical systems (Ijspeert et al., 2013; Rana et al., 2020), this chapter integrates a learned bijective encoder implemented as a Euclideanizing flow (Dinh et al., 2016; Rana et al., 2020) with an orbitally stable supercritical Hopf bifurcation as the latent dynamics.

This approach enables the learning of periodic motions from demonstrations with stability guarantees. The inverse Jacobian of the bijective encoder facilitates the projection of predicted latent velocities back into the oracle space. The proposed method is experimentally validated on various robot embodiments, including a tendon-driven helicoid soft robot (Guan et al., 2023), a swimming turtle robot, a UR5 robotic manipulator, and a KUKA cobot.

Conclusion and Appendices

In the following, we will detail the content of Parts III & IV IV.

- **Chapter 13** summarizes our findings and suggests promising directions for future research.
- **Appendix A** introduces a framework for the holistic co-design of soft robots, enabling the simultaneous optimization of both the robot's body (e.g., its morphology) and its brain (e.g., the perception and control systems) (Navez, 2024; Spielberg et al., 2019; Wang et al., 2024). Importantly, the framework considers a broad array of criteria beyond task-centric performance, including safety, manufacturability, and regulatory considerations. It explicitly addresses the practical realization of designs by demonstrating how prototyping can reduce uncertainty in the computational evaluation metrics, while also proposing modifications to the co-design process that could greatly enhance computational efficiency. In summary, this holistic co-design framework provides guidance on how the soft robot's structural shape, actuator and sensor placement (see Chapters 4 & 9), reduced-order modeling (see Chapters 5, 10, & 11), and controller (see Chapters 6, 8, 10, & 11) might be jointly optimized in the future to boost performance while ensuring that the closed-loop system remains sufficiently compliant and safe (see Chapter 3).
- **Appendix B** contains supplementary material, including details on the experimental setup and additional results, for Chapter 11.
- **Appendix C** outlines some of the laboratory infrastructure developed during this PhD—such as a motion capture system and pneumatic pressure regulation—to support the research presented in this thesis. Additionally, we introduce several software libraries that underpin many of the thesis chapters, including a JAX implementation of control-oriented soft robot models for simulation, model-based control, and motion planning, as well as a ROS2 ecosystem for model-based control of HSA robots.

1.4.1 Important Notes

We acknowledge that, although we strived to maintain consistent notation across chapters, the diversity of problem settings and methods, coupled with the limited pool of available symbols, necessitated the use of tailored notations for each chapter. Also, please note that, unless explicitly stated otherwise, the term "soft robot" in this thesis refers to "continuum soft robots", as opposed to articulated soft robots (Della Santina et al., 2020b). The code and data associated with this thesis are made available as a dataset on the *4TU.ResearchData* platform[2].

[2] https://doi.org/10.4121/a9ee4280-4ef1-4c2b-bcef-526cd50292a9

2 Background on Modeling and Control of Soft Robots

In this chapter, we review existing approaches for both kinematic and dynamic modeling of soft robots and discuss how these models can subsequently be exploited in closed-form model-based controllers. These topics are highly relevant to the core of the thesis and many concepts presented in this are employed throughout all chapters of this thesis. Examples include the PCC (Della Santina et al., 2020a; Webster III and Jones, 2010) and PCS (Renda et al., 2018) kinematic models, as well as soft robot dynamics expressed in an Euler-Lagrangian formulation (Della Santina et al., 2023) and their projection into operational (Della Santina et al., 2020c; Khatib, 1987) and actuation (Pustina et al., 2024a) spaces. Moreover, we report various closed-form soft robot controllers, such as those based on error-based feedback combined with energy shaping (Della Santina et al., 2023), integral-saturated PIDs (Pustina et al., 2022a), or Cartesian impedance controllers (Della Santina et al., 2020c). We emphasize that kinematic and dynamic modeling (Armanini et al., 2023) of soft robots, as well as model-based control (Della Santina et al., 2023), remain highly active areas of research, and that much of the content presented in this chapter has only been proposed in recent years (Yasa et al., 2023).

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q} = \mathcal{Q}$$

2.1 Kinematics: From Planar Constant Strain to Geometric Variable Strain

In this section, we review relevant kinematic parameterizations and models for soft robots that the community has developed in recent years.

2.1.1 Preliminaries

The goal of a kinematic model for robots is to provide a description of the robot's current shape—specifically, to determine $SE(3)$ poses for every point on the robot body given the current value of its configuration variables $q \in \mathbb{R}^n$. For rigid manipulators, one can readily derive a kinematic model that describes the position and orientation of the links by considering the joint angles as the configuration variables (Siciliano et al., 2010). In contrast, the continuous three-dimensional deformations of continuum soft robots make this task considerably more challenging (Armanini et al., 2023). To simplify the problem, the community adopts the *slender structure* hypothesis for continuum soft robots, which assumes that the body's radius is much smaller than its length: $R \ll L$. This assumption permits the use of Cosserat rod theory (Cosserat, 1909) by reducing the analysis to the one-dimensional deformation of a geometric line that represents an infinitesimally thin backbone of the continuum robot (Gazzola et al., 2018)[1]. We then model the backbone's deformation in 3D space by examining the variation of the strain field $\xi(t,s) = \begin{bmatrix} \kappa_x & \kappa_y & \kappa_z & \sigma_x & \sigma_y & \sigma_z \end{bmatrix}^\top \in \mathbb{R}^6$, where t denotes time and $s \in (0, L]$ is the curvilinear abscissa along the backbone curve. Here, κ_* and σ_* represent the rotational and linear strains, measured in units of $[\mathrm{rad/m}]$ and $[-]$, respectively. In particular, when the backbone is aligned with the local z-axis, κ_x and κ_y capture bending strains along the two local directions, κ_z describes the twist strain, while σ_x and σ_y account for shear strains, and σ_z corresponds to the axial strain (Della Santina et al., 2023).

Nonetheless, since these Cosserat rod models reside in an infinite-dimensional state space, it becomes impossible to measure the state of a soft robot directly. Moreover, the associated dynamics are formulated as Partial Differential Equations (PDEs) (Gazzola et al., 2018), complicating their direct use in control applications. Therefore, the ultimate aim is to recast the dynamics as Ordinary Differential Equations (ODEs)—specifically, in Euler-Lagrangian form—by approximating the backbone deformation with finite-dimensional parameterizations.

In fact, the continuous strain field can be approximated as $\xi(t,s) = \Psi(t,q,s)$, where $\Psi(t,q,s) : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}^6$ serves as the functional parameterization, $q \in \mathbb{R}^n$ denotes the finite-dimensional configuration, and $\xi^0 \in \mathbb{R}^6$ represents the rest strain (Mathew et al., 2025). Then, the forward kinematics can be formalized as $\pi(q,s) : \mathbb{R}^n \times \mathbb{R}_+ \to SE(3)$.

Below, we introduce several commonly used kinematic parametrizations—many of which are employed in this thesis—that range from relatively simple single-segment planar cases to the most general and expressive GVS models.

[1] Models that consider the full three-dimensional deformations of soft robots, extending volumetric Finite Element Method (FEM) to control-oriented modeling, are beyond the scope of this thesis. We refer the interested reader to Armanini et al. (2023); Coevoet et al. (2017); Faure et al. (2012).

2.1.2 Planar Constant Strain

Analog to Chapter 5[2], we first consider a planar CS model that considers a segment that exhibits constant strain. In the planar case, we need to consider the following three deformations: the bending strain $\kappa_{\text{be}}(s)$, the shear strain $\sigma_{\text{sh}}(s)$, and the axial strain $\sigma_{\text{ax}}(s)$. As these strains are assumed to be constant, we can state for any point s along the segment that the strain field is given by $\xi(t) = \begin{bmatrix}\kappa_{\text{be}}(t) & \sigma_{\text{sh}}(t) & \sigma_{\text{ax}}(t)\end{bmatrix}^\top \in \mathbb{R}^3$. After defining the rest strain $\xi^0 = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^\top$, we can express the strain field as a function of the finite-dimensional configuration $q \in \mathbb{R}^3$

$$\xi(t) = \underbrace{\mathbb{I}_3\, q(t)}_{\Psi(q)} + \xi^0. \tag{2.1}$$

The goal is now to determine the planar $SE(2)$ pose $\chi = \begin{bmatrix}p_{\text{x}} & p_{\text{y}} & \theta\end{bmatrix}^\top \in \mathbb{R}^3$ for each $s \in (0, L]$, where p_{x} is the x-position, p_{y} the y-position, and $\theta \in \mathbb{S}^1$ the planar orientation measured as the angle with respect to the x-axis. Assuming that the robot backbone is aligned with the local y-axis, the infinitesimal change of pose along the backbone is given as

$$\frac{\partial \pi(q,s)}{\partial s} = \begin{bmatrix}\cos(\kappa_{\text{be}}\, s)\,\sigma_{\text{sh}} - \sin(\kappa_{\text{be}}\, s)\,\sigma_{\text{ax}} \\ \sin(\kappa_{\text{be}}\, s)\,\sigma_{\text{sh}} + \cos(\kappa_{\text{be}}\, s)\,\sigma_{\text{ax}} \\ \kappa_{\text{be}}\end{bmatrix}. \tag{2.2}$$

We can now integrate $\frac{\partial \chi}{\partial s}$ in order to obtain the forward kinematics

$$\chi = \pi(q,s) = \int_0^s \frac{\partial \pi}{\partial s}(q,s')\,\mathrm{d}s' = \begin{bmatrix} p_{\text{x}}^0 + \sigma_{\text{sh}}\frac{-\sin(\theta^0)+\sin(\theta^0+\kappa_{\text{be}}\, s)}{\kappa_{\text{be}}} + \sigma_{\text{ax}}\frac{-\cos(\theta^0)+\cos(\theta^0+\kappa_{\text{be}}\, s)}{\kappa_{\text{be}}} \\ p_{\text{y}}^0 + \sigma_{\text{sh}}\frac{\cos(\theta^0)-\cos(\theta^0+\kappa_{\text{be}}\, s)}{\kappa_{\text{be}}} + \sigma_{\text{ax}}\frac{-\sin(\theta^0)+\sin(\theta^0+\kappa_{\text{be}}\, s)}{\kappa_{\text{be}}} \\ \theta^0 + \kappa_{\text{be}}\, s \end{bmatrix}, \tag{2.3}$$

where $\chi^0 = \begin{bmatrix}p_{\text{x}}^0 & p_{\text{y}}^0 & \theta^0\end{bmatrix}^{\text{T}}$ is the $SE(2)$ pose of the base. The geometric Jacobian (Siciliano et al., 2010) associated with the forward kinematics is given as

$$J(q,s) = \frac{\partial \pi(q,s)}{\partial q} = \begin{bmatrix} \frac{\kappa_{\text{be}}\, s\,(\sigma_{\text{sh}} c_{\text{be}} - \sigma_{\text{ax}} s_{\text{be}}) + \sigma_{\text{sh}}\left(s_{\theta^0} - s_{\text{be}}\right) + \sigma_{\text{ax}}\left(c_{\theta^0} - c_{\text{be}}\right)}{\kappa_{\text{be}}^2} & \frac{-s_{\theta^0}+s_{\text{be}}}{\kappa_{\text{be}}} & \frac{-c_{\theta^0}+c_{\text{be}}}{\kappa_{\text{be}}} \\ \frac{\kappa_{\text{be}}\, s\,(\sigma_{\text{sh}} s_{\text{be}} + \sigma_{\text{ax}} c_{\text{be}}) + \sigma_{\text{sh}}\left(-c_{\theta^0} + c_{\text{be}}\right) + \sigma_{\text{ax}}\left(s_{\theta^0} - s_{\text{be}}\right)}{\kappa_{\text{be}}^2} & \frac{c_{\theta^0}-c_{\text{be}}}{\kappa_{\text{be}}} & \frac{-s_{\theta^0}+s_{\text{be}}}{\kappa_{\text{be}}} \\ s & 0 & 0 \end{bmatrix}, \tag{2.4}$$

where $s_{\theta^0} = \sin(\theta^0)$, $c_{\theta^0} = \cos(\theta^0)$ and $s_{\text{be}} = \sin(\theta^0 + \kappa_{\text{be}}\, s)$, $c_{\text{be}} = \cos(\theta^0 + \kappa_{\text{be}}\, s)$. As demonstrated in Chapter 5, the inverse kinematics $q = \varrho(\chi, s)$ of a planar constant strain segment are available in closed form

$$q = \varrho(\chi,s) = \frac{\theta - \theta^0}{2\, s} \begin{bmatrix} 2 \\ -\sin(\theta^0)\, p_{\text{x}} + \cos(\theta^0)\, p_{\text{y}} - \frac{\left(\cos(\theta^0)\, p_{\text{x}} + \sin(\theta^0)\, p_{\text{y}}\right)\sin(\theta-\theta^0)}{\cos(\theta-\theta^0)-1} \\ -\cos(\theta^0)\, p_{\text{x}} - \sin(\theta^0)\, p_{\text{y}} - \frac{\left(-\sin(\theta^0)\, p_{\text{x}} + \cos(\theta^0)\, p_{\text{y}}\right)\sin(\theta-\theta^0)}{\cos(\theta-\theta^0)-1} \end{bmatrix}. \tag{2.5}$$

[2] In Chapter 5, we leverage a planar constant strain model to describe the kinematics of planar HSA robots.

Please note that even though both the forward and inverse kinematics - i.e., Eqs. (2.3)- (2.5) - exhibit a well-defined limit for the case of no bending

$$\begin{aligned}\lim_{\kappa_{\text{be}}\to 0} \pi(q,s) &= \begin{bmatrix} p_{\text{x}}^0 + \sigma_{\text{sh}}\cos(\theta^0)\,s - \sigma_{\text{ax}}\sin(\theta^0)\,s \\ p_{\text{y}}^0 + \sigma_{\text{sh}}\sin(\theta^0)\,s + \sigma_{\text{ax}}\cos(\theta^0)\,s \\ \theta^0 \end{bmatrix}, \\ \lim_{\theta\to\theta^0} \varrho(\chi,s) &= \begin{bmatrix} 0 \\ \frac{\cos(\theta^0)\,p_{\text{x}} + \sin(\theta^0)\,p_{\text{y}}}{s} \\ \frac{-\sin(\theta^0)\,p_{\text{x}} + \cos(\theta^0)\,p_{\text{y}}}{s} \end{bmatrix}, \end{aligned} \tag{2.6}$$

in practice, we face numerical issues when evaluating the kinematics or the matrices of the dynamical model. Therefore, we found it most effective to manually modulate the configuration for the bending strain κ_{be} to exhibit a minimum magnitude $\varepsilon \in \mathbb{R}_+$ when evaluating the kinematics or dynamics of such strain-based models, as, for example, the case for the *JAX Soft Robot Modeling* package presented in Appendix C.4

$$\tilde{\kappa}_{\text{be}} = \begin{cases} \text{sign}(\kappa_{\text{be}})\,\varepsilon, & \text{if } |\kappa_{\text{be}}| < \varepsilon \\ \kappa_{\text{be}}, & \text{else} \end{cases}, \tag{2.7}$$

where $\varepsilon \approx 0.01\,\text{rad/m}$ determines the minimum magnitude of the bending strain when evaluating the dynamical matrices. Therefore, $\tilde{\kappa}_{\text{be}}$ exhibits a discontinuity at $\kappa_{\text{be}} = 0$ when it switches from $-\varepsilon$ to ε and vice-versa. Furthermore, the gradient exhibits a discontinuity at $|\kappa_{\text{be}}| = \varepsilon$ when it transitions between a linear and a constant function.

Planar Constant Curvature

In the special case of a planar CC segment (Della Santina et al., 2023; Webster III and Jones, 2010), we would set $\sigma_{\text{sh}} = 0$, $\sigma_{\text{ax}} = 1$ and reformulate the strain field as

$$\xi(t) = \underbrace{\begin{bmatrix} 1 \\ 0_{2\times 1} \end{bmatrix}}_{\Phi} q, \tag{2.8}$$

where now the configuration only contains the bending strain $q = \kappa_{\text{be}} \in \mathbb{R}$.

2.1.3 Piecewise Constant Strain

The PCS model (Renda et al., 2018, 2016) extends the idea of planar constant strain to (i) the 3D case with six-dimensional strains and (2) multiple segments with each segment exhibiting constant strain. The strain field of a N segment PCS soft robot is

$$\xi(t,s) = \Psi(t,q,s) = \underbrace{\begin{bmatrix} \Phi_1(s) & \cdots & \Phi_i(s) & \cdots & \Phi_N \end{bmatrix}}_{\Phi(s)} q(t) + \xi^0, \tag{2.9}$$

where $q \in \mathbb{R}^{6N}$ (i.e., $n = 6N$) is the configuration, $\xi^0 \in \mathbb{R}^6$ is the rest strain, that is usually chosen as $\xi^0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^\top$ for the case of the backbone being aligned with the

local z-axis, $\Phi(s) : \mathbb{R}_+ \to \mathbb{R}^{6\times 6N}$ is called the strain basis (Mathew et al., 2025; Renda et al., 2020) and consists of N discontinuous functions $\Phi_i \, \forall i \in \{1,\dots,N\}$ that are defined as

$$\Phi_i(s) = \begin{cases} \mathbb{I}_6, & \text{if } \sum_{i'=1}^{i-1} l_{i'} < s \leq \sum_{i'=1}^{i} l_{i'} \\ 0_{6\times 3}, & \text{else} \end{cases} \in \mathbb{R}^{6\times 6} \quad \forall i \in \{1,\dots,N\}, \tag{2.10}$$

where l_i is the length of each segment, therefore fulfilling the property $L = \sum_{i=1}^{N} l_i$.

For the derivation of the forward kinematics, we point the reader to Renda et al. (2018). In Section 5.3, we introduce a variation of the PCS parametrization, named SPCS, that allows for selective strains to remain constant over the entire backbone length instead of just for each segment - allowing for a reduction of the DOFs of the model. Additionally, we use a planar version of the PCS model in Chapters 10 & 11.

Piecewise Constant Curvature

In the special case of PCC (Webster III and Jones, 2010)—the most popular soft robot parametrization—the twist and shear strains are neglected (i.e., $\kappa_z, \sigma_x, \sigma_y = 0$), resulting in the following strain basis functions:

$$\Phi_i(s) = \begin{cases} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^\top, & \text{if } \sum_{i'=1}^{i-1} l_{i'} < s \leq \sum_{i'=1}^{i} l_{i'}, \\ 0_{6\times 3}, & \text{else,} \end{cases} \in \mathbb{R}^{6\times 6} \quad \forall i \in \{1,\dots,N\}. \tag{2.11}$$

Even though the name of the model implies otherwise, the configuration of the ith segment $q_i = \begin{bmatrix} \kappa_{x,i} & \kappa_{y,i} & \sigma_z - 1 \end{bmatrix}^\top \in \mathbb{R}^3$ usually contains the two bending strains κ_x, κ_y and the axial elongation strain σ_z. We emphasize that the PCC model extends the previously mentioned planar CC case to 3D and multi-segment soft robots by approximating the bending as piecewise constant. The forward kinematics, given by $\chi = \pi(q, s)$, is introduced in Della Santina et al. (2020a); Webster III and Jones (2010), and the inverse kinematics for up to three segments is available in closed form (Li et al., 2023).

Various alternative versions of the PCC parametrization appear in the literature. In particular, the seminal work by Webster III and Jones (2010) introduced the (κ, ϕ, ℓ) parametrization, while Della Santina et al. (2020a) proposed a Δ-parametrization. We discuss both parameterizations in greater detail below.

(κ, ϕ, ℓ)-Parametrization. Webster III and Jones (2010) defines the configuration of the ith segment as $q_i = \begin{bmatrix} \kappa_i & \phi_i & \ell_i \end{bmatrix}^\top$, where $\kappa_i \in \mathbb{R}$ and $\ell_i \in \mathbb{R}_+$ are the curvature and length of the arc, respectively, and $\phi_i \in [-\pi, \pi]$ the angle of the direction of bending. We can easily identify the mapping between the strain-based parametrization and the (κ, ϕ, ℓ) parametrization

$$\kappa_i = \sqrt{\kappa_{x,i}^2 + \kappa_{y,i}^2}, \qquad \phi_i = \operatorname{arctan2}(\kappa_{y,i}, -\kappa_{x,i}), \qquad \ell_i = \sigma_{z,i}\, s, \tag{2.12}$$

and the inverse map as

$$\kappa_{x,i} = -\sin(\phi_i)\, \kappa_i, \qquad \kappa_{y,i} = \cos(\phi_i)\, \kappa_i, \qquad \sigma_{z,i} = \frac{\ell_i}{s}. \tag{2.13}$$

Although the (κ, ϕ, ℓ) parametrization has significantly influenced the soft robotics field and been widely adopted, it presents several drawbacks affecting both kinematics and dynamics, as initially noted by Della Santina et al. (2020a): (a) when $\kappa_i = 0$, there is a redundancy because ϕ_i can take any value without changing the backbone shape; (b) a sign redundancy exists for κ_i, since the configurations $(\kappa_i, \phi_i, \ell_i)$ and $(-\kappa_i, \phi_i - \pi, \ell_i)$ produce the same shape; and (c) a discontinuity is observed when ϕ_i transitions from π to $-\pi$, or vice versa.

Δ-Parametrization. Della Santina et al. (2020a) introduced the Δ-parametrization to address some of the shortcomings of the (κ, ϕ, ℓ) parametrization. In this approach, the configuration of the ith PCC segment is represented as $q_i = \begin{bmatrix}\Delta_{\mathrm{x},i} & \Delta_{\mathrm{y},i} & \delta L_i\end{bmatrix}^\top$, which depends on the difference between the soft segment side lengths and the change in backbone length, denoted by δL_i. The underlying intuition is that each segment's shape is uniquely defined by the lengths of four arcs that are equiangularly distributed at a radial distance of $d_i \in \mathbb{R}_+$ from the backbone center line. Taking the example of the four arcs at polar angles $\theta_1 = 0\,\mathrm{rad}, \theta_2 = \frac{\pi}{2}\,\mathrm{rad}, \theta_3 = \pi\,\mathrm{rad}, \theta_4 = \frac{3\pi}{2}\,\mathrm{rad}$ with the arc lengths $L_{1,i}, L_{2,i}, L_{3,i}, L_{4,i}$, given as a function of the strains $\kappa_{\mathrm{x},i}, \kappa_{\mathrm{y},i}, \sigma_{\mathrm{z},i}$

$$\begin{aligned} L_{1,i} &= \left(\sigma_{\mathrm{z},i} - \kappa_{\mathrm{y},i}\, d_i\right) s, \qquad & L_{2,i} &= \left(\sigma_{\mathrm{z},i} + \kappa_{\mathrm{y},i}\, d_i\right) s, \\ L_{3,i} &= \left(\sigma_{\mathrm{z},i} + \kappa_{\mathrm{x},i}\, d_i\right) s, \qquad & L_{4,i} &= \left(\sigma_{\mathrm{z},i} - \kappa_{\mathrm{x},i}\, d_i\right) s, \end{aligned} \tag{2.14}$$

the configuration can be computed as

$$\Delta_{\mathrm{x},i} = \frac{L_{2,i} - L_{1,i}}{2}, \qquad \Delta_{\mathrm{y},i} = \frac{L_{4,i} - L_{3,i}}{2}, \qquad \delta L_i = \frac{L_{1,i} + L_{2,i}}{2} - s. \tag{2.15}$$

Then, $\Delta_i = \sqrt{\Delta_{\mathrm{x},i}^2 + \Delta_{\mathrm{y},i}^2}$ would then denote the magnitude of the bending. As the mapping between the (κ, ϕ, ℓ)- and the Δ-parametrization is already presented in Della Santina et al. (2020a), we here only report the mapping from the strain-based parametrization into the Δ-parametrization

$$\Delta_{\mathrm{x},i} = \kappa_{\mathrm{y},i}\, d_i\, s, \qquad \Delta_{\mathrm{y},i} = -\kappa_{\mathrm{x},i}\, d_i\, s, \qquad \delta L_i = (\sigma_{\mathrm{z},i} - 1)\, s, \tag{2.16}$$

with the inverse map defined as

$$\kappa_{\mathrm{x},i} = -\frac{\Delta_{\mathrm{y},i}}{d_i\, s}, \qquad \kappa_{\mathrm{y},i} = \frac{\Delta_{\mathrm{x},i}}{d_i\, s}, \qquad \sigma_{\mathrm{z},i} = 1 + \frac{\delta L_i}{s}. \tag{2.17}$$

We conclude that the Δ-parametrization exhibits many of the same properties and characteristics—such as the absence of redundancies and discontinuities—as the strain-based formulation, and that configurations can be easily mapped back and forth.

We use a planar version of the PCC model in Chapter 8 and leverage the Δ-parametrization of PCC in Chapters 4 & 9.

2.1.4 Geometric Variable Strain

To our knowledge, GVS (Boyer et al., 2020; Mathew et al., 2025; Renda et al., 2020) represents the most expressive kinematic parametrization available for Cosserat rods. In particular,

the strain field is defined as

$$\xi(t,s) = \Phi(t,q,s)\, q + \xi^0(s), \tag{2.18}$$

which is composed of a configuration- and time-dependent strain basis function $\Phi(t,q,s) : \mathbb{R}\times\mathbb{R}^n\times\mathbb{R}+ \to \mathbb{R}^{6\times n}$ (potentially exhibiting discontinuities and a configuration and time dependencies) and an abscissa-dependent rest strain function $\xi^0(s) : \mathbb{R}+ \to \mathbb{R}^6$ (Mathew et al., 2025). Conventionally, $\Phi(s)$ is constructed using abscissa-dependent basis functions, such as those derived from monomial or Legendre polynomial functions (Mathew et al., 2025). For instance, Della Santina and Rus (2019) examine the polynomial curvature case.

Affine Curvature

A special case of variable strain is Affine Curvature (AC) (Della Santina, 2020; Stella et al., 2022; Tiburzio et al., 2025), later extended to multi-segment cases via PAC by Stella et al. (2023c). A strain-based parametrization of AC could look like

$$\xi(t,s) = \Psi(t,q,s) = \underbrace{\begin{bmatrix} 1 & s & 0 & 0 \\ 0 & 0 & 1 & s \\ 0_{4\times1} & 0_{4\times1} & 0_{4\times1} & 0_{4\times1} \end{bmatrix}}_{\Phi(s)\in\mathbb{R}^{6\times4}} q(t) + \xi^0, \tag{2.19}$$

where $q \in \mathbb{R}^4$ is the configuration of the AC robot. Alternatively, we could enforce the same affine curvature function for both bending directions and additionally allow for an axial extension of the segment. Accordingly, Baaij et al. (2023); Stella et al. (2023c) parameterized the configuration of the ith AC segment as

$$q_i = \begin{bmatrix} \kappa_{0,i} & \kappa_{1,i} & \phi_i & \delta L_i \end{bmatrix}^\top \in \mathbb{R}^4, \tag{2.20}$$

where $\kappa_{0,i}, \kappa_{1,i} \in \mathbb{R}$ are the coefficients of the AC function, $\phi_i \in [-\pi, \pi]$ signifies the bending direction, and $\delta L_i = L_i(t) - L_i^0$ captures the elongation of the segment, where L_i^0 is the equilibrium length of the segment. The strains of the segment can be recovered using the map

$$\xi(t,s) = \begin{bmatrix} -\left(\kappa_{0,i} + \kappa_{1,i}\, s\right) \sin(\phi_i) \\ +\left(\kappa_{0,i} + \kappa_{1,i}\, s\right) \cos(\phi_i) \\ 0_{3\times1} \\ 1 + \frac{\delta L_i}{s} \end{bmatrix}, \tag{2.21}$$

where $\kappa_i = \kappa_{0,i} + \kappa_{1,i}\, s$ represents the curvature of the (ith) segment. We leverage this AC parametrization in Chapter 9 of this thesis.

2.2 Euler-Lagrangian Dynamics

In the following section, we introduce the structure and key characteristics of continuum soft robot dynamics, formulated in Euler-Lagrangian form. For a detailed derivation of these dynamics, readers are referred to several survey papers in the literature (Alessi et al., 2024a; Armanini et al., 2023; Della Santina et al., 2023; Gazzola et al., 2018).

Regardless of the chosen kinematic parameterization, the EOMs for a soft robot can generally be expressed in Euler-Lagrangian form (Della Santina et al., 2023):

$$M(q)\,\ddot{q} + C(q,\dot{q})\,\dot{q} + K(q) + G(q) + D\,\dot{q} = A(q)\,\tau, \tag{2.22}$$

where the mass matrix $M(q) \succ 0 \in \mathbb{R}^{n\times n}$ represents the robot's inertia, $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ accounts for Coriolis and centrifugal effects, $K(q)$ and $G(q)$ (in $\mathbb{R}^n$) correspond to elastic and gravitational forces, respectively, and $D \succ 0$ is the damping matrix. This model shares properties with traditional rigid robot dynamics, such as the skew-symmetry of $\dot{M}(q) - 2\,C(q,\dot{q})$, provided that $C(q,\dot{q})$ is defined using Christoffel symbols (Della Santina et al., 2020c).

We assume that the soft robot is actuated by m actuators via the mapping $\alpha : q \times \tau \mapsto A(q)\,\tau$, where $A(q) \in \mathbb{R}^{n\times m}$. [3] Depending on the relationship between n and m, if $n = m$, the robot is fully actuated; if $n < m$ (which is less common), it is considered overactuated (Pustina et al., 2024a); and if $n > m$, the soft robot is regarded as underactuated (Pustina, 2025).

2.2.1 Formulation as a 1st-Order Ordinary Differential Equation (ODE)

After defining the state of the soft robot as $y = \begin{bmatrix} q^\top & \dot{q}^\top \end{bmatrix}^\top \in \mathbb{R}^{2n}$, (2.22) can be expressed as a control-affine ODE in nonlinear state-space representation

$$\dot{y} = f(y,\tau) = \begin{bmatrix} \frac{\mathrm{d}q}{\mathrm{d}t} \\ \frac{\mathrm{d}\dot{q}}{\mathrm{d}t} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ -M^{-1}(q)\left(C(q,\dot{q})\,\dot{q} + K(q) + G(q) + D\,\dot{q}\right) \end{bmatrix} + \underbrace{\begin{bmatrix} 0_{n\times m} \\ M^{-1}(q)\,A(q) \end{bmatrix}}_{B(q)} \tau. \tag{2.23}$$

2.2.2 Kinetic and Potential Energy

Typically, we can derive analytical expressions for the system's kinetic and potential energy:

$$\mathcal{T}(q,\dot{q}) = \frac{1}{2}\dot{q}^\top M(q)\,\dot{q}, \qquad \mathcal{U}(q) = \mathcal{U}_{\mathrm{K}}(q) + \mathcal{U}_{\mathrm{G}}(q) = \int K(q) + G(q)\,\mathrm{d}q, \tag{2.24}$$

where $\mathcal{T}(q,\dot{q}) \geq 0$ is the kinetic and $\mathcal{U}(q)$ the potential energy consisting of elastic and gravitational terms $\mathcal{U}_{\mathrm{K}}(q)$, $\mathcal{U}_{\mathrm{G}}(q)$. which results in $K(q) + G(q) = \frac{\partial \mathcal{U}}{\partial q}$. In the case of linear elasticity $K(q) = S\,q$, where $S \succ 0 \in \mathbb{R}^{n\times n}$ is the stiffness of the generalized coordinates, the elastic potential energy becomes $\mathcal{U}_{\mathrm{K}}(q) = \frac{1}{2}q^\top S\,q$. Consequently, the Lagrangian is defined as $\mathcal{L}(q,\dot{q}) = \mathcal{T}(q,\dot{q}) - \mathcal{U}(q)$.

As demonstrated by Della Santina et al. (2023, 2020c) and in Chapters 8 and 11, these energetic expressions provide a solid foundation for analyzing the stability of both open- and closed-loop systems using Lyapunov arguments (Khalil, 2002).

[3] While this control-affine formulation with a configuration-dependent actuation matrix $A(q)$ captures the characteristics of most existing soft robot actuation methods—such as pneumatic or tendon-driven actuation—the actuation can also be represented as a more general nonlinear function $\alpha(q,\tau)$, as is the case for the HSA robot actuation model presented in Chapter 5. Furthermore, when accounting for actuator dynamics, as discussed in Chapter 8 for pneumatic pistons, the generalized actuation torque may also depend on the actuator's state $(\mu,\dot{\mu})$.

2.2.3 Operational Space Dynamics

As originally proposed in the seminal work by Khatib (1987) and later adapted for soft robots by Della Santina et al. (2020c), the system dynamics can be directly expressed in operational space[4]. Regardless of the precise definition of operational space (e.g., end-effector pose), given the forward kinematics $\pi : q \mapsto x$, where $x \in \mathbb{R}^o$ denotes the operational space coordinate with $o \leq n$, and the corresponding Jacobian $J(q) = \frac{\partial \pi}{\partial q} \in \mathbb{R}^{o \times n}$, (2.22) can equivalently be written in operational space as

$$\Lambda_{\mathrm{x}}(q)\,\ddot{x} + \eta_{\mathrm{x}}(q,\dot{q})\,\dot{q} + J_{\mathrm{M}}^{+\top}(q)\left(G(q) + K(q) + D\,\dot{q}\right) = J_{\mathrm{M}}^{+\top}(q)\,A(q)\,\tau, \tag{2.25}$$

where

$$J_{\mathrm{M}}^{+}(q) = M^{-1}(q)J^{\top}(q)\left(J(q)M^{-1}(q)J^{\top}(q)\right)^{-1} \in \mathbb{R}^{n \times o}, \tag{2.26}$$

denotes the dynamically consistent pseudo-inverse (Chang and Khatib, 1995), which maps operational space velocities to configuration space and transforms generalized torques into operational space. Moreover,

$$\Lambda_{\mathrm{x}}(q) = \left(J(q)M^{-1}(q)J^{\top}(q)\right)^{-1} \in \mathbb{R}^{o \times o}, \tag{2.27}$$

defines the operational space inertia matrix, and the operational space Coriolis and centrifugal matrix $\eta(q,\dot{q})$[5] is defined as (Della Santina et al., 2020c; Khatib, 1987)

$$\eta_{\mathrm{x}}(q,\dot{q}) = \Lambda(q)\left(J(q)M^{-1}(q)\,C(q,\dot{q}) - \dot{J}(q,\dot{q})\right) \in \mathbb{R}^{o \times o}. \tag{2.28}$$

Null Space Dynamics

When $o < n$, there are $(n-o)$ redundant DOFs—commonly known as the null space dynamics (Siciliano et al., 2010). It is crucial to consider these null space dynamics to ensure, for instance, the stability of the closed-loop robotic system rather than focusing solely on the end-effector's operational space movement. Various approaches in the literature address the derivation of null space dynamics, ranging from projection-based methods to operational space augmentation techniques (Ott, 2008). In the following, we focus on the *Joint Space Decomposition Method* (Ott, 2008).

First, we introduce the additional null space velocities defined as $\nu_{\mathrm{n}} = N(q)\,\dot{q} \in \mathbb{R}^{n-o}$, which represent the time derivatives of the null space coordinates $n(t) \in \mathbb{R}^{n-o}$ [6]. With an appropriate choice of $N(q) \in \mathbb{R}^{(n-o)\times n}$, this yields the formulation described in (Della Santina et al., 2020c; Ott, 2008).

$$\begin{bmatrix} \dot{x} \\ \nu_{\mathrm{n}} \end{bmatrix} = \begin{bmatrix} J(q) \\ N(q) \end{bmatrix} \dot{q}, \tag{2.29}$$

where for the nullspace base matrix $Z(q) \in \mathbb{R}^{(n-o)\times n}$ we define (Ott, 2008)

$$N(q) = \left(Z(q)M(q)Z(q)^{\top}\right)^{-1} Z(q)M(q). \tag{2.30}$$

[4]The operational space is also commonly referred to as task space. Since end-effectors (e.g., grippers) are typically attached to the distal end of the robot, the operational space is often defined by the pose $\chi \in SE(3)$ of the end-effector.

[5]Please note that in Chapter 6 this operational space Coriolis matrix is denoted as $\mu(q,\dot{q})$.

[6]Note that the null space coordinates can always be recovered via integration, i.e., $n(t) = \int_0^t \nu_{\mathrm{n}}(t')\,\mathrm{d}t'$, although they may not have a direct geometric interpretation (Ott, 2008).

In the general case, $Z(q)$ can be constructed as (Huang and Varma, 1991; Ott, 2008)

$$Z(q) = \left[-J_{\mathrm{n-o}}^{\top}(q) J_{\mathrm{o}}^{-\top} \quad \mathbb{I}_{n-o}\right], \tag{2.31}$$

for which we partitioned the Jacobian $J(q)$ as

$$J(q) = \begin{bmatrix} J_{\mathrm{o}} & J_{\mathrm{n-o}} \end{bmatrix} \in \mathbb{R}^{o \times n}, \tag{2.32}$$

such that $J_{\mathrm{o}}(q) \in \mathbb{R}^{o \times o}$ is at least locally invertible.

The EOM of the combined operational and null space dynamics are now given as (Ott, 2008)

$$\underbrace{\begin{bmatrix} \Lambda_{\mathrm{x}}(q) & 0_{o\times(n-o)} \\ 0_{(n-o)\times o} & \Lambda_{\mathrm{n}}(q) \end{bmatrix}}_{\Lambda(q)\in\mathbb{R}^{n\times n}} \begin{bmatrix} \ddot{x} \\ \dot{\nu}_{\mathrm{n}} \end{bmatrix} + \underbrace{\begin{bmatrix} \eta_{\mathrm{x}}(q,\dot{q}) \\ \eta_{\mathrm{n}}(q,\dot{q}) \end{bmatrix}}_{\eta(q,\dot{q})\in\mathbb{R}^{n\times n}} \dot{q} + \underbrace{\begin{bmatrix} J_{\mathrm{M}}^{+\top}(q) \\ Z(q) \end{bmatrix}}_{J_{\mathrm{MZ}}^{\top}(q)\in\mathbb{R}^{n\times n}} (G(q) + K(q) + D\,\dot{q}) = \begin{bmatrix} f_{\mathrm{x}} \\ f_{\mathrm{n}} \end{bmatrix} = J_{\mathrm{MZ}}^{\top}(q)\, A(q)\, \tau, \tag{2.33}$$

where $J_{\mathrm{MZ}}(q)$ can be used to map operational and null space velocities into configuration space (Ott, 2008)

$$\dot{q}(t) = \underbrace{\begin{bmatrix} J_{\mathrm{M}}^{+}(q) \\ Z(q) \end{bmatrix}}_{J_{\mathrm{MZ}}(q)} \begin{bmatrix} \dot{x}(t) \\ \nu_{\mathrm{n}}(t) \end{bmatrix}, \tag{2.34}$$

which allows us to decompose $\eta(q,\dot{q})\,\dot{q}$ into

$$\eta(q,\dot{q})\,\dot{q} = \begin{bmatrix} \eta_{\mathrm{xx}}(q,\dot{q}) & \eta_{\mathrm{xn}}(q,\dot{q}) \\ \eta_{\mathrm{nx}}(q,\dot{q}) & \eta_{\mathrm{nn}}(q,\dot{q}) \end{bmatrix} \begin{bmatrix} \dot{x} \\ \nu_{\mathrm{n}} \end{bmatrix}, \tag{2.35}$$

where

$$\begin{aligned} \eta_{\mathrm{xx}}(q,\dot{q}) &= \eta_{\mathrm{x}}(q,\dot{q}) J_{\mathrm{M}}^{+}(q), & \eta_{\mathrm{xn}}(q,\dot{q}) &= \eta_{\mathrm{x}}(q,\dot{q})\, Z(q), \\ \eta_{\mathrm{nx}}(q,\dot{q}) &= \eta_{\mathrm{n}}(q,\dot{q}) J_{\mathrm{M}}^{+}(q), & \eta_{\mathrm{nn}}(q,\dot{q}) &= \eta_{\mathrm{n}}(q,\dot{q})\, Z(q). \end{aligned} \tag{2.36}$$

Furthermore, the null space inertia $\Lambda_{\mathrm{n}}(q)$ is defined as (Ott, 2008)

$$\Lambda_{\mathrm{n}}(q) = Z(q)\, M(q)\, Z(q)^{\top}. \tag{2.37}$$

2.2.4 Actuation Space Dynamics

Pustina et al. (2024a) demonstrated that it is often feasible to directly formulate the dynamics in actuation coordinates $\varphi \in \mathbb{R}^n$, where the actuation force τ acts directly on the first m actuation coordinates via an identity matrix. This feature is especially appealing for control applications because it enables a straightforward mapping of a control input devised in the generalized (collocated) coordinates into an actuation τ. Moreover, this approach may also be useful in the future in other applications such as inverse kinematics (Della Santina, 2025) or workspace estimation. Since this collocated mapping is particularly beneficial in underactuated settings, as later demonstrated in Chapter 6, we focus on the case $m \leq n$, which includes both fully actuated and underactuated systems. In the following, we will assume for the actuation matrix $A(q)$ to be full-rank (i.e., $\mathrm{rank}(A) = m$).

According to Pustina et al. (2024a), there exists a transformation into collocated form when there is a function $g(q) : \mathbb{R}^n \mapsto \mathbb{R}^m$ such that $\frac{\partial g}{\partial q} = A^\top(q)$, which is referred to as *integrability assumption*. In practice, as shown in Sec. 6.2), this mapping $g(q)$ is found by solving the integral $g(q) = \int_0^t \dot{q}(t')^\top A(q')\,\mathrm{d}t'$. If $A(q)$ is constant, the mapping is given by $g(q) = A^\top q$ (Pustina et al., 2024a). If $m < n$, we stress that the coordinates given by $g(q)$ would not be sufficient to fully describe the dynamics of the system, as there would exist a null space of dimensionality $n-m$. Therefore, we can choose which $n-m$ coordinates from the original configuration space to keep. For example, the following transformation into actuation coordinates would be valid for a given $g(q)$ (Pustina et al., 2024a)

$$\varphi = \begin{bmatrix} \varphi_\mathrm{a} \\ \varphi_\mathrm{u} \end{bmatrix} = h(q) = \begin{bmatrix} g(q) \\ 0_{n-m} \end{bmatrix} + \begin{bmatrix} 0_{m\times m} & 0_{m\times(n-m)} \\ 0_{(n-m)\times m} & \mathbb{I}_{n-m} \end{bmatrix} q, \tag{2.38}$$

where $\varphi_\mathrm{a} \in \mathbb{R}^m$ and $\varphi_\mathrm{u} \in \mathbb{R}^{n-m}$ refer to the actuated and unactuated actuation coordinates, respectively, and the function $h(q) : \mathbb{R}^n \to \mathbb{R}^n$ maps from configuration space into collocated form. $J_\mathrm{h}(q)$ is its associated Jacobian, and it is defined together with its inverse[7]

$$J_\mathrm{h}(q) = \frac{\partial h(q)}{\partial q} = \begin{bmatrix} A_\mathrm{a}^\top(q) & A_\mathrm{u}^\top(q) \\ 0_{(n-m)\times m} & \mathbb{I}_{n-m} \end{bmatrix}, \quad J_\mathrm{h}^{-1}(q) = \left(\frac{\partial h(q)}{\partial q}\right)^{-1} = \begin{bmatrix} A_\mathrm{a}^{-\top}(q) & -A_\mathrm{a}^{-\top}(q)A_\mathrm{u}^\top(q) \\ 0_{(n-m)\times m} & \mathbb{I}_{n-m} \end{bmatrix}, \tag{2.39}$$

and $A_\mathrm{a}(q) \in \mathbb{R}^{m\times m}$, $A_\mathrm{u}(q) \in \mathbb{R}^{(n-m)\times m}$ stem from the following partitioning: $A(q) = \begin{bmatrix} A_\mathrm{a}(q) \\ A_\mathrm{u}(q) \end{bmatrix}$. The dynamics in actuation coordinates are then given by

$$M_\varphi(q)\,\ddot{\varphi} + \eta_\varphi(q,\dot{q})\,\dot{\varphi} + J_{\mathrm{h,M}}^{+\top}(q)\,(G(q) + K(q) + D\,\dot{q}) = \begin{bmatrix} \tau \\ 0_{n-m} \end{bmatrix}, \tag{2.40}$$

where analog to the operational space dynamics, the actuation space inertia is defined as (Khatib, 1987)

$$M_\varphi(q) = \left(J_\mathrm{h}(q)\,M^{-1}(q)\,J_\mathrm{h}^\top(q)\right)^{-1} = J_\mathrm{h}^{-\top}(q)\,M(q)\,J_\mathrm{h}^{-1}(q) \in \mathbb{R}^{n\times n}, \tag{2.41}$$

$J_{\mathrm{h,M}}^{+\top}(q) \in \mathbb{R}^{n\times n}$ is the dynamically-consistent pseudo-inverse (Chang and Khatib, 1995)

$$J_{\mathrm{h,M}}^{+}(q) = J_\mathrm{h}^{-1}(q) = \begin{bmatrix} A_\mathrm{a}^{-\top}(q) & -A_\mathrm{a}^{-\top}(q)A_\mathrm{u}^\top(q) \\ 0_{m\times(n-m)} & \mathbb{I}_{n-m} \end{bmatrix}, \tag{2.42}$$

$\eta_\varphi(q,\dot{q})$ captures the Coriolis and centrifugal effects

$$\eta_\varphi(q,\dot{q}) = M_\varphi(q)\left(J_\mathrm{h}(q)\,M^{-1}(q)\,C(q,\dot{q}) - \dot{J}_\mathrm{h}(q,\dot{q})\right)J_\mathrm{h}^{-1}(q) \in \mathbb{R}^{n\times n}. \tag{2.43}$$

The property $\dot{M}_\varphi(q) = \eta_\varphi(q,\dot{q}) + \eta_\varphi^\top(q,\dot{q})$ still holds. Crucially, we now notice in (2.41) how τ directly acts via an identity actuation matrix on the first m actuation coordinates. We

[7] We assume in this Chapter for $J_\mathrm{h}(q)$ to be either invertible for all $q \in \mathbb{R}^n$ or at least in the relevant workspace $q \in \mathcal{Q}$ of the soft robot.

can partition the dynamics of (2.41) into the first m (actuated) rows, and the last $n-m$ unactuated rows:

$$\begin{bmatrix} M_{\varphi,\mathrm{a}}(q) \\ M_{\varphi,\mathrm{u}}(q) \end{bmatrix} \ddot{\varphi} + \begin{bmatrix} \eta_{\varphi,\mathrm{a}}(q,\dot{q}) \\ \eta_{\varphi,\mathrm{u}}(q,\dot{q}) \end{bmatrix} \dot{\varphi} + \underbrace{\begin{bmatrix} A_{\mathrm{a}}^{-1}(q) & 0_{m\times(n-m)} \\ -A_{\mathrm{u}}(q)A_{\mathrm{a}}^{-1}(q) & \mathbb{I}_{n-m} \end{bmatrix}}_{J_{\mathrm{h}}^{-\top}(q)} \left(\begin{bmatrix} G_{\mathrm{a}}(q) \\ G_{\mathrm{u}}(q) \end{bmatrix} + \begin{bmatrix} K_{\mathrm{a}}(q) \\ K_{\mathrm{u}}(q) \end{bmatrix} + \begin{bmatrix} D_{\mathrm{aa}}(q) & D_{\mathrm{au}}(q) \\ D_{\mathrm{ua}}(q) & D_{\mathrm{uu}}(q) \end{bmatrix} \dot{q} \right) = \begin{bmatrix} \tau \\ 0_{n-m} \end{bmatrix}, \tag{2.44}$$

where $G_{\mathrm{a}}(q), K_{\mathrm{a}}(q) \in \mathbb{R}^{m}$, $G_{\mathrm{u}}(q), K_{\mathrm{u}}(q) \in \mathbb{R}^{n-m}$, and $D_{\mathrm{aa}} \in \mathbb{R}^{m\times m}$, $D_{\mathrm{au}} \in \mathbb{R}^{m\times(n-m)}$, $D_{\mathrm{ua}} \in \mathbb{R}^{(n-m)\times m}$, $D_{\mathrm{uu}} \in \mathbb{R}^{(n-m)\times(n-m)}$. We can simplify the expression while leveraging the structure of $J_{\mathrm{h}}^{-\top}$

$$\begin{bmatrix} M_{\varphi,\mathrm{aa}}(q) & M_{\varphi,\mathrm{au}}(q) \\ M_{\varphi,\mathrm{ua}}(q) & M_{\varphi,\mathrm{uu}}(q) \end{bmatrix} \ddot{\varphi} + \begin{bmatrix} \eta_{\varphi,\mathrm{a}}(q,\dot{q}) \\ \eta_{\varphi,\mathrm{u}}(q,\dot{q}) \end{bmatrix} \dot{\varphi} + \begin{bmatrix} A_{\mathrm{a}}^{-1}(q)(G_{\mathrm{a}}(q)+K_{\mathrm{a}}(q)) \\ -A_{\mathrm{u}}(q)A_{\mathrm{a}}^{-1}(q)(G_{\mathrm{a}}(q)+K_{\mathrm{a}}(q)) + G_{\mathrm{u}}(q) + K_{\mathrm{u}}(q) \end{bmatrix} + \underbrace{\begin{bmatrix} D_{\varphi,\mathrm{a}}(q) \\ D_{\varphi,\mathrm{u}}(q) \end{bmatrix}}_{D_{\varphi}(q)} \dot{\varphi} = \begin{bmatrix} \tau \\ 0_{n-m} \end{bmatrix}, \tag{2.45}$$

where $D_{\varphi}(q) \succ 0 \in \mathbb{R}^{n\times n}$ is the damping matrix in actuation coordinates[8]

$$\begin{aligned} D_{\varphi}(q) &= J_{\mathrm{h}}^{-\top}(q) D J_{\mathrm{h}}^{-1}(q) \succ 0, \\ &= \begin{bmatrix} A_{\mathrm{a}}^{-1}(q) D_{\mathrm{aa}} A_{\mathrm{a}}^{-\top}(q) & -A_{\mathrm{a}}^{-1}(q) D_{\mathrm{aa}} A_{\mathrm{a}}^{-\top}(q) A_{\mathrm{u}}^{\top}(q) + A_{\mathrm{a}}^{-1}(q) D_{\mathrm{ua}} \\ -A_{\mathrm{u}}(q) A_{\mathrm{a}}^{-1}(q) D_{\mathrm{aa}} A_{\mathrm{a}}^{-\top}(q) + D_{\mathrm{au}} A_{\mathrm{a}}^{-\top}(q) & -A_{\mathrm{u}}(q) A_{\mathrm{a}}^{-1}(q) D_{\mathrm{ua}} + D_{\mathrm{uu}} \end{bmatrix}, \end{aligned} \tag{2.46}$$

$\eta_{\varphi,\mathrm{a}}(q,\dot{q}) \in \mathbb{R}^{m\times n}$, $\eta_{\varphi,\mathrm{u}}(q,\dot{q}) \in \mathbb{R}^{(n-m)\times n}$, $M_{\varphi,\mathrm{aa}}(q) \in \mathbb{R}^{m\times m}$, $M_{\varphi,\mathrm{uu}}(q) \in \mathbb{R}^{(n-m)\times(n-m)}$, and $M_{\varphi,\mathrm{ua}}(q) \in \mathbb{R}^{(n-m)\times m}$, and $M_{\varphi,\mathrm{au}}(q) \in \mathbb{R}^{m\times(n-m)}$.

Assuming access to the inverse map $q = h^{-1}(\varphi)$, we can also directly map the potential and kinetic energy of the system into the new coordinates:

$$\mathcal{T}_{\varphi}(\varphi,\dot{\varphi}) = \mathcal{T}(q,\dot{q})\big|_{q=h^{-1}(\varphi),\,\dot{q}=J_{\mathrm{h}}^{-1}\varphi}, \qquad \mathcal{U}_{\varphi}(\varphi) = \mathcal{U}(q)\big|_{q=h^{-1}(\varphi)}. \tag{2.47}$$

Therefore, we can write the dynamics equivalently as

$$\begin{bmatrix} M_{\varphi,\mathrm{aa}}(q) & M_{\varphi,\mathrm{au}}(q) \\ M_{\varphi,\mathrm{ua}}(q) & M_{\varphi,\mathrm{uu}}(q) \end{bmatrix} \ddot{\varphi} + \begin{bmatrix} \eta_{\varphi,\mathrm{a}}(q,\dot{q}) \\ \eta_{\varphi,\mathrm{u}}(q,\dot{q}) \end{bmatrix} \dot{\varphi} + \begin{bmatrix} \partial_{\varphi_{\mathrm{a}}} \mathcal{U}_{\varphi}(q) \\ \partial_{\varphi_{\mathrm{u}}} \mathcal{U}_{\varphi}(q) \end{bmatrix} + \begin{bmatrix} D_{\varphi,\mathrm{a}}(q) \\ D_{\varphi,\mathrm{u}}(q) \end{bmatrix} \dot{\varphi} = \begin{bmatrix} \tau \\ 0_{n-m} \end{bmatrix}, \tag{2.48}$$

with $\partial_{\varphi}\mathcal{U}(\varphi) = \frac{\partial}{\partial\varphi}\mathcal{U}_{\varphi}(\varphi) \in \mathbb{R}^{n}$ the potential forces.

Fully Actuated Dynamics

Here, we note some expressions of the actuation space dynamics in the fully-actuated case (i.e., $n = m$). For example, we can observe that

$$J_{\mathrm{h}}(q) = A^{\top}(q) \in \mathbb{R}^{n\times n}, \qquad J_{\mathrm{h}}^{-1}(q) = A^{-\top}(q), \qquad J_{\mathrm{h,M}}^{+\top}(q) = J_{\mathrm{h}}^{-\top}(q) = A^{-1}(q), \tag{2.49}$$

which leads to the actuation space dynamics

$$\underbrace{A^{-1}(q)M(q)A^{-\top}(q)}_{M_{\varphi}(q)\succ 0}\ddot{\varphi} + n_{\varphi}(q,\dot{q})\dot{\varphi} + A^{-1}(q)(G(q)+K(q)) + \underbrace{A^{-1}(q)DA^{-\top}(q)}_{D_{\varphi}(q)\succ 0}\dot{\varphi} = \tau, \tag{2.50}$$

where

$$n_{\varphi}(q,\dot{q}) = A^{-1}(q)C(q,\dot{q})A^{-\top}(q) - A^{-1}(q)M(q)A^{-\top}(q)\dot{A}^{\top}(q)A^{-\top}(q). \tag{2.51}$$

[8] As $D \succ 0$ and $\mathrm{rank}(J_{\mathrm{h}}^{-1}(q)) = n\,\forall q \in \mathcal{Q}$, we can conclude that $D_{\varphi}(q) \succ 0\,\forall q \in \mathcal{Q}$, which will be crucial for the stability proof of many controllers.

Constant Actuation Matrix. In the case of a constant actuation matrix $A(q) = A$, the map $h(q)$ is now given by $\varphi = A^\top q$, which renders the damping matrix to not be configuration-dependent anymore: $D_\varphi \succ 0$. Furthermore, the actuation space Coriolis matrix simplifies to

$$\eta_\varphi(q, \dot{q}) = A^{-1}\, C(q, \dot{q})\, A^{-\top}. \tag{2.52}$$

Finally, in case of linear elasticity $K(q) = S\, q$, we can define

$$K_\varphi(\varphi) = \underbrace{A^{-1}\, S\, A^{-\top}}_{S_\varphi}\, \varphi = S_\varphi\, \varphi, \tag{2.53}$$

where $S_\varphi \succ 0$ if $S \succ 0$.

End-to-End Mapping from Actuation into Task Space

At least in the fully-actuated case with a square matrix $A(q)$, which we additionally also assume to be invertible, we can formulate an end-to-end mapping of velocities and forces between actuation and task space. Specifically, as $\dot{q} = J_\mathrm{h}^{-1}(q)\, \dot{\varphi} = A^{-\top}(q)\, \dot{\varphi}$ (Pustina et al., 2024a), we can state

$$\dot{x} = J(q)\, \dot{q} = \underbrace{J(q) J_\mathrm{h}^{-1}(q)}_{J_{\varphi \to x}(q)}\, \dot{\varphi}, \qquad \tau = J_{\varphi \to x}^\top(q)\, f, \tag{2.54}$$

where $f \in \mathbb{R}^o$ are forces in task space.

2.3 Model-Based Control

In this section, we review recently proposed closed-form controllers for soft robots that can leverage advanced nonlinear models[9]. Many of these approaches appear throughout the thesis, such as the P-satI-D+potential shaping controller in Section 6.2 and Chapters 10 and 11, the PD+ controller—augmented with backstepping to consider actuator dynamics—in Chapter 8, and the operational space impedance controller in Section 6.3. This section provides a high-level overview of the most common closed-form model-based control laws developed for soft robots. For a more detailed discussion of these controllers, including Lyapunov-based stability proofs, interested readers should consult the respective publications (Borja et al., 2022; Della Santina et al., 2023, 2020c; Pustina et al., 2022a,b). For an in-depth study of controlling underactuated soft robots, we recommend the thesis by Pustina (2025).

Unlike much of the existing literature, which formulates controllers in configuration space (Caasenbrood and Nijmeijer, 2021; Della Santina et al., 2023, 2020c), we develop most control laws directly in actuation space, which was only very recently made possible by the mapping into actuation coordinates proposed by Pustina et al. (2024a), which we already detailed in Sec. 2.2.4. This approach simplifies both the design and convergence analysis of controllers—even for fully actuated soft robots, where the number of configuration variables equals the number of actuators—especially when dealing with configuration-dependent actuation matrices $A(q)$ that are common in tendon-driven soft robots. The limitation of this

[9] Please note that this excludes approaches relying on linear/linearized models such as LQR and that model-based controllers that are not available in closed-form, such as MPC, are out of scope of this thesis.

strategy is that it requires establishing a mapping into actuation coordinates to formulate the dynamics in actuation space, as shown in Section 2.2.4, which might not always possible. Additionally, experimental validation of actuation-space controllers remains very limited (Pustina, 2025; Stölzle et al., 2024b).

2.3.1 Preliminaries

Problem Statement

The literature on closed-form model-based controllers generally considers two problem settings (Sciavicco and Siciliano, 2012): *setpoint regulation* and *trajectory tracking*. Typically, a reference trajectory is represented by the tuple $(q^{\mathrm{d}}(t), \dot{q}^{\mathrm{d}}(t), \ddot{q}^{\mathrm{d}}(t))$, which comprises the desired configuration, velocity, and acceleration at each time step. For *trajectory tracking*, our objective is to design a controller that enables the soft robotic system to follow the reference trajectory, and specifically its positional references, as precisely as possible while rejecting any disturbances

$$\lim_{t\to\infty} \left(q^{\mathrm{d}}(t) - q(t)\right) = 0_n. \tag{2.55}$$

The trajectory tracking controller proposed by Della Santina et al. (2020c) requires both $q^{\mathrm{d}}(t)$ and $\dot{q}^{\mathrm{d}}(t)$ to be bounded. Therefore, they mandate two positive constants $\gamma_q, \gamma_{\dot{q}} > 0$ to exist such that

$$\|q^{\mathrm{d}}(t)\| < \gamma_q, \qquad \|\dot{q}^{\mathrm{d}}(t)\| < \gamma_{\dot{q}}. \tag{2.56}$$

Alternatively, if only the final goal of the motion is of interest, the problem can be simplified to a *setpoint regulation* scenario, where the controller drives the system toward a desired configuration q^{d} (with $\dot{q}^{\mathrm{d}} = 0$). Then, we would like to design a regulator that achieves the following asymptotic convergence

$$\lim_{t\to\infty} \left(q^{\mathrm{d}} - q(t)\right) = 0_n. \tag{2.57}$$

Note that these motion references—whether setpoints or trajectories—can be specified not only in configuration space but also in various coordinate frames, a topic we discuss further in the subsection *Coordinate Frames*.

Control Design Objectives

When designing a model-based controller, we often balance several, sometimes conflicting, objectives. The following section outlines a selection of these criteria, enabling a comparison of the tradeoffs between various model-based controllers.

- **Control Frequency/Computational Complexity.** In this thesis, we aim to derive a control law in closed form, which greatly enhances computational efficiency compared to, for example, optimization-based methods and supports higher control frequencies. This improvement not only boosts performance but also reduces the risk of instability in feedback controllers. When a closed-form control law is available, the computational complexity is determined by (a) the complexity of the model and (b) the specific model terms that must be evaluated for the controller. For instance, a potential shaping controller, which requires only knowledge about the potential forces, is computationally more efficient than a computed torque controller that necessitates evaluating all the dynamical matrices within the EOMs.

- **Stability.** Our goal is generally to achieve exponential Global Asymptotic Stability (GAS) of the closed-loop system. This is best demonstrated in setpoint regulation scenarios, where the system should converge to a specified setpoint regardless of its initial state and do so at an exponential rate (Khalil, 2002). If exponential convergence is unattainable, we still aim for asymptotic stability, which ensures convergence but without a guaranteed rate. At a minimum, we require local asymptotic stability around the setpoint, meaning that if the system state starts within a certain region of attraction, it will stabilize at the setpoint. However, if initialized outside this region, the system may not converge due to the presence of other stable or unstable equilibria.

- **Control Effort.** To facilitate energy-efficient (mobile) robots, we strive to minimize the power consumed by the actuators, which includes both the applied force/torque τ and the corresponding actuator velocity $\dot{\mu}$. Since the necessary steady-state force is typically determined by the soft robot's design and the task at hand, our focus is on reducing the control effort—often defined as the integral of τ over time—which in turn reduces the required actuator velocity and overall power consumption.

- **Compliance.** To complement the inherent mechanical compliance of soft robots, the controller should also be compliant. This means designing feedback controllers with minimal integral gains to avoid compromising stability margins and potential safety issues, as well as using low proportional feedback gains to prevent increasing the closed-loop stiffness—ensuring the robot retains its desired softness. Model-based control approaches actually generally allow us to analyze the closed-loop compliance (Stella et al., 2023d) and to certify that they match the task requirements.

- **Robustness.** Model-based controllers leverage our understanding of system behavior to craft intelligent feedback and feedforward components that simplify control. However, as Box (1976) aptly noted, "All models are wrong." Controllers that rely on perfect model knowledge to cancel out most of the system dynamics, such as those based on feedback linearization, are highly sensitive to modeling errors. When the modeled dynamics are canceled, any unmodeled dynamics can quickly dominate, potentially inducing instability and poor performance. In contrast, model-based feedforward controllers, employing partial cancellations or energy shaping, tend to be more robust to modeling errors, as they reshape rather than entirely eliminate the open-loop system dynamics.

Coordinate Frames

In the following section, we introduce several coordinate frames for designing control laws and outline the advantages and disadvantages of each approach.

Control in Configuration Space. As introduced in Sec.2.1, the configuration space is typically defined by the kinematic variables that parameterize the soft robot's backbone shape, which is why control in this space is often referred to as *shape control*. One advantage of designing a controller in configuration space is that the robot dynamics are naturally defined here, as seen in (2.22), making it simpler to derive the control law and prove

its stability. However, there are disadvantages: (a) establishing a consistent reference in configuration space can be challenging since the key motion is often specified in operational space, and in underactuated settings, not all configurations can be statically achieved by the actuators (Della Santina, 2025); and (b) mapping control inputs defined as generalized torques into an actuation τ can be difficult—especially when $A(q)$ exhibits singularities or when the soft robot is underactuated, which requires specialized controllers (Pustina et al., 2022b).

Control in Actuation Space. To address the challenges of mapping generalized torques into actuator signals, we can exploit the collocated dynamics (Pustina et al., 2024a) presented in Sec. 2.2.4 to design controllers directly in actuation space. Even in underactuated scenarios, the first m actuation coordinates φ_a are directly influenced by τ through an identity actuation matrix, significantly simplifying the derivation of controllers. In practice, a configuration-space reference $(q^\mathrm{d}, \dot{q}^\mathrm{d})$ can be transformed into an actuation-space reference using

$$\varphi^\mathrm{d} = h(q^\mathrm{d}), \qquad \dot{\varphi}^\mathrm{d} = J_\mathrm{h}(q^\mathrm{d})\,\dot{q}, \qquad \ddot{\varphi}^\mathrm{d} = J_\mathrm{h}(q^\mathrm{d})\,\ddot{q}^\mathrm{d} + \dot{J}_\mathrm{h}(q^\mathrm{d})\,\dot{q}^\mathrm{d}. \tag{2.58}$$

This approach not only simplifies the mapping of control inputs into an actuation τ but also makes it easier to enforce actuator limits. Nevertheless, devising a consistent actuation-space reference, specifically one that is statically or dynamically feasible, remains a challenge.

Control in Operational Space. In this setting, we use the dynamics defined in (2.25) to directly design a controller in operational space (Khatib, 1987). Typically, the operational space reference's dimensionality o must satisfy $o \leq n$. Operational space impedance controllers are particularly popular because they allow for direct specification of operational space stiffness and damping characteristics, which is crucial for robot-environment interactions. Additionally, since the reference is provided directly in operational space, complex inverse kinematics routines are unnecessary. However, disadvantages include the lack of formal stability guarantees—especially regarding the null-space dynamics—and the greater difficulty in considering actuator limits.

In the following sections, we will generally define the control law as a function of the robot configuration q and its time derivative $\dot{q}$, since mapping into operational space via the forward kinematics $\chi = \pi(q, s)$ or into actuation coordinates through $\varphi = h(q)$ is usually straightforward if needed. The same applies to the control reference, as discussed in the next paragraph.

Taxonomy of Control Terms

Most controllers considered in this thesis consist of two components: either (a) a pure feedforward or mixed feedforward-feedback term that leverages model knowledge and (b) an error-based feedback term. Specifically, a control law $\tau(t, q, \dot{q})$ can be decomposed into

$$\tau(t, q, \dot{q}, q^\mathrm{d}, \dot{q}^\mathrm{d}, \ddot{q}^\mathrm{d}) = \underbrace{\tau_\mathrm{mb}(t, q, \dot{q}, q^\mathrm{d}, \dot{q}^\mathrm{d}, \ddot{q}^\mathrm{d})}_{\text{Model-Based}} + \underbrace{\tau_\mathrm{fb}(t, q, \dot{q}, q^\mathrm{d}, \dot{q}^\mathrm{d})}_{\text{Error-Based Feedback}}. \tag{2.59}$$

Here, $\tau_{\text{mb}}(t, q, \dot{q}) : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^m$ denotes a model-based term—not necessarily a pure feedforward[10]—while $\tau_{\text{fb}}(t, q, \dot{q})$ is an error-based feedback term that does not incorporate any model knowledge. To simplify the notation, we will thereafter not explicitly state the dependence on the reference $(q^{\text{d}}, \dot{q}^{\text{d}}, \ddot{q}^{\text{d}})$.

In the following sections, we introduce a range of approaches for error-based feedback and model-based terms that can be combined quite flexibly. Finally, we highlight several control strategies that closely integrate these model-based and feedback components, including a computed torque controller based on full feedback linearization and an operational space impedance controller.

2.3.2 Error-Based Feedback Terms

In this thesis, we consider the following formulation for an error-based feedback controller

$$\tau_{\text{fb}} = K_{\text{p}} \left(\varphi_{\text{a}}^{\text{d}}(t) - \varphi_{\text{a}}(t) \right) + K_{\text{i}} \int_0^t \sigma \left(\varphi_{\text{a}}^{\text{d}}(t') - \varphi_{\text{a}}(t') \right) \, \mathrm{d}t' + K_{\text{d}} \left(\dot{\varphi}_{\text{a}}^{\text{d}}(t) - \dot{\varphi}_{\text{a}}(t) \right), \tag{2.60}$$

where $K_{\text{p}}, K_{\text{p}}, K_{\text{d}} \in \mathbb{R}^{m \times m}$ are control gain matrices, which we usually choose to be diagonal. $\sigma(r) : \mathbb{R}^m \to \mathbb{R}^m$ is a function that transforms the control error $\varphi_{\text{a}}^{\text{d}}(t) - \varphi_{\text{a}}(t)$ before integration. From this general formulation, we can derive several specific versions, which we outline below.

PID Control. A standard PID controller is obtained by setting the saturation function as the identity, i.e., $\sigma(r) = r$. We have used this feedback controller as a baseline in Chapters 6 and 8.

PD Control. The PD+ variant (Kelly, 1997), commonly employed for the low-level control of rigid robots, is realized by combining a PD controller with static force compensation. In practice, this is achieved by setting $K_{\text{i}} = 0$.

P-satI-D Control. Pustina et al. (2022a) introduced an integral-saturated PID for the shape regulation of soft robots via error-based feedback control. A typical choice for the saturation function is $\sigma(r) = \tanh(\gamma\, r)$, where $\gamma \in \mathbb{R}_+$ is a control gain that compresses the control error prior to saturation. We implement this integral-saturated PID in Section 6.2 and Chapters 10 and 11.

2.3.3 Model-Based Terms: The Fully Actuated Case

First, we consider common model-based terms for the fully actuated setting (i.e., $n = m$). Please note that for all controllers introduced below, we assume that (1) the model is reasonably accurate—or that sufficiently high feedback gains are chosen, as it is the case in high-gain control (Marino, 1985),—and (2) the integrability condition (Pustina et al., 2024a) along with the mapping into actuation coordinates can be identified. Furthermore, (3) we assume that the square matrix $A(q) \in \mathbb{R}^{n \times n}$ is invertible for any $q \in \mathbb{R}^n$, at least locally within the relevant workspace.

[10] A pure feedforward term would have the signature $\tau(t)$, as it depends solely on the time-based reference $(q^{\text{d}}, \dot{q}^{\text{d}})$ rather than the current soft robot state $(q, \dot{q})$.

For setpoint regulators, we additionally assume that (4) the setpoint $q^{\mathrm{d}}/\varphi^{\mathrm{d}}$ is an attainable equilibrium of the underactuated system (Pustina et al., 2022b)—meaning it satisfies

$$G(q^{\mathrm{d}}) + K(q^{\mathrm{d}}) = A(q^{\mathrm{d}})\,\tau^{\mathrm{ss}}, \tag{2.61}$$

for a constant control action τ^{ss}—and that (5) the system is initialized within the reference's region of attraction. For controllers where GAS can be proven, this condition is inherently satisfied; however, many controllers only exhibit Local Asymptotic Stability (LAS). In the latter case, the region of attraction can often be expanded by increasing the proportional feedback gains to enhance the system's stiffness that comes with a loss of compliance (Della Santina et al., 2017, 2023) or by canceling some non-convex potential forces via nonlinear feedback. Additionally, (6) the actuator must be fast enough in tracking its reference $(\mu^{\mathrm{d}}, \mu^{\mathrm{d}})$ so that we can approximate $\tau^{\mathrm{d}} \approx \tau(\mu, \dot{\mu})$, where τ^{d} denotes the desired actuation/control input and $\tau(\mu, \dot{\mu})$ represents the actual forces, torques, currents, or pneumatic pressure generated.

Finally, many controllers formulated in actuation coordinates have not yet been tested in simulation, and most have not been experimentally verified, as the mapping into actuation coordinates was only very recently proposed (Pustina et al., 2024a).

Setpoint Regulation with Feedforward Compensation

The following pure feedforward term regulates the system towards the setpoint $\varphi^{\mathrm{d}} = h(q^{\mathrm{d}})$ in actuation coordinates (Borja et al., 2022; Della Santina et al., 2023; Kelly and Salgado, 1994; Pustina, 2025)

$$\tau_{\mathrm{mb}} = \partial_{\varphi} \mathcal{U}_{\varphi}(\varphi^{\mathrm{d}}) = A^{-1}(q^{\mathrm{d}}) \left(G(q^{\mathrm{d}}) + K(q^{\mathrm{d}}) \right). \tag{2.62}$$

This controller with constant control input τ_{mb} is often also referred to as an energy or potential shaping feedforward term (Borja et al., 2022). Oftentimes, this pure feedforward term is complemented by a feedback controller that contains a PD-like term (Della Santina et al., 2023), e.g., $\tau_{\mathrm{fb}}(q, \dot{q}) = K_{\mathrm{p}} \left(\varphi^{\mathrm{d}} - \varphi \right) - K_{\mathrm{d}}\, \dot{\varphi}$, which renders the closed-loop dynamics to take the form

$$M_{\varphi}(q)\, \ddot{\varphi} + \eta_{\varphi}(q, \dot{q})\, \dot{\varphi} + \partial_{\varphi} \mathcal{U}_{\varphi}(\varphi) - \partial_{\varphi} \mathcal{U}_{\varphi}(\varphi^{\mathrm{d}}) + K_{\mathrm{p}} \left(\varphi - \varphi^{\mathrm{d}} \right) + \left(D_{\varphi}(q) + K_{\mathrm{d}} \right) \dot{\varphi} = 0_n. \tag{2.63}$$

As mentioned already before, for this regulator to be effective, φ^{d} needs to be an attainable equilibrium of the system. The stability of the regulator can be demonstrated via the Lyapunov candidate (Della Santina et al., 2023)

$$\begin{aligned} V(\varphi, \dot{\varphi}) = & \underbrace{\frac{1}{2}\, \dot{\varphi}^{\top} M_{\varphi}(h^{-1}(\varphi))\, \dot{\varphi}}_{\text{Kinetic Energy}} + \underbrace{\mathcal{U}_{\varphi}(\varphi) - \mathcal{U}_{\varphi}(\varphi^{\mathrm{d}})}_{\text{Centered Potential Energy}} \\ & + \underbrace{\left(G(h^{-1}(\varphi^{\mathrm{d}})) + K(h^{-1}(\varphi^{\mathrm{d}})) \right)^{\top} A^{-\top}(h^{-1}(\varphi^{\mathrm{d}})) \left(\varphi^{\mathrm{d}} - \varphi \right)}_{\text{Correction Term}} + \underbrace{\frac{1}{2} (\varphi^{\mathrm{d}} - \varphi)^{\top} K_{\mathrm{p}} (\varphi^{\mathrm{d}} - \varphi)}_{\text{Artificial Potential Energy of Feedback}}, \end{aligned} \tag{2.64}$$

which is a valid Lyapunov function - fulfilling the requirements $V(\varphi^{\mathrm{d}}, 0_n) = 0$ and $V(\varphi, \dot{\varphi}) > 0 \; \forall \, (\varphi, \dot{\varphi}) \in (\mathbb{R}^n, \mathbb{R}^n) \setminus \{(\varphi^{\mathrm{d}}, 0_n)\}$ - if the closed-loop potential energy is locally or globally

convex

$$\frac{\partial^2}{\partial \varphi^2} \mathcal{U}_\varphi(\varphi)\Big|_{\varphi=\varphi^\mathrm{d}} + K_\mathrm{p} = \frac{\partial}{\partial \varphi}\left(A^{-1}(h^{-1}(\varphi))\left(G(h^{-1}(\varphi)) + K(h^{-1}(\varphi))\right)\right)\Big|_{\varphi=\varphi^\mathrm{d}} + K_\mathrm{p} \succ 0. \tag{2.65}$$

The Lyapunov function exhibits the time derivative (Della Santina et al., 2023)

$$\begin{aligned}\dot{V}(\varphi,\dot{\varphi}) &= \dot{\varphi}^\top M_\varphi(h^{-1}(\varphi))\,\ddot{\varphi} + \frac{1}{2}\,\dot{\varphi}^\top \dot{M}_\varphi(h^{-1}(\varphi))\,\dot{\varphi} - \dot{\varphi}^\top\left(\partial_\varphi \mathcal{U}_\varphi(\varphi) - \partial_\varphi \mathcal{U}_\varphi(\varphi^\mathrm{d}) + K_\mathrm{p}\left(\varphi - \varphi^\mathrm{d}\right)\right),\\ &= -\dot{\varphi}^\top \eta_\varphi(q,\dot{q})\,\dot{\varphi} + \frac{1}{2}\,\dot{\varphi}^\top \dot{M}_\varphi(q)\,\dot{\varphi} + \dot{\varphi}^\top\left(D_\varphi(h^{-1}(\varphi)) + K_\mathrm{d}\right)(-\dot{\varphi}),\\ &= -\dot{\varphi}^\top\left(D_\varphi(h^{-1}(\varphi)) + K_\mathrm{d}\right)\dot{\varphi} \leq 0 \quad \forall\, \varphi,\dot{\varphi} \in \mathbb{R}^n.\end{aligned} \tag{2.66}$$

Finally, as $\dot{V}(\dot{\varphi}) \leq 0\,\forall\,\varphi,\dot{\varphi} \in \mathbb{R}^n$, asymptotic stability can be proven via LaSalle's invariance theorem (Khalil, 2002) as φ^d is a static equilibrium of the closed-loop system.

In the following, we will discuss the stability properties of the controller in more detail. If the open-loop potential energy of the soft robot, consisting of elastic and gravitational terms, is globally convex

$$\frac{\partial^2}{\partial \varphi^2} \mathcal{U}_\varphi(\varphi)\Big|_{\varphi=\varphi^\mathrm{d}} = \frac{\partial}{\partial \varphi}\left(A^{-1}(h^{-1}(\varphi))\left(G(h^{-1}(\varphi)) + K(h^{-1}(\varphi))\right)\right)\Big|_{\varphi=\varphi^\mathrm{d}} \succ 0, \tag{2.67}$$

what is sometimes also referred to as the soft robot being *elastically dominated* (Borja et al., 2022; Della Santina et al., 2023; Pustina, 2025). This means that the robot is considered to be relatively stiff and able to withstand gravity. In such case, the proportional feedback term is not strictly necessary (i.e., we can set $K_\mathrm{p} = 0_{n\times n}$, and φ^d is still an asymptotically stable equilibrium of the closed-loop system. In turn, when the open-loop potential energy of the soft robot is non-convex, we can still stabilize the system locally or even globally by choosing sufficiently large proportional feedback gains $K_\mathrm{p} \succ 0$. This, however, comes at the cost of increased system stiffness, reduced compliance, and enlarged risk of instability because of measurement noise. A derivative feedback term is not necessary as for soft robots generally $D_\varphi(\varphi) \succ 0\,\forall\,\varphi \in \mathbb{R}^n$. Still, we can increase dissipation by choosing $K_\mathrm{d} \succ 0$.

We leverage such a potential shaping feedforward term in Chapters 6, 10-11 of this thesis. The controller exhibits great robustness to modeling errors and measurement noise. Also, it is computationally extremely efficient as the controller only needs to be evaluated once per setpoint. However, this comes at the cost of the necessity for increased feedback gains to ensure (global) convergence. Also, exponential convergence is not guaranteed.

Setpoint Regulation with Gravity Cancellation and Compensation of Elastic Forces

When the elastic potential is convex but the gravitational potential makes the overall potential energy non-convex, we can lower the required (proportional) feedback gains—compared to the *Setpoint Regulation with Feedforward Compensation* controller—by canceling rather than compensating for the gravitational forces. The control law is then given as (Della Santina et al., 2020c)

$$\tau_\mathrm{mb}(q) = \underbrace{A^{-1}(q)\,G(q)}_{\text{Cancel Gravity}} + \underbrace{A^{-1}(q^\mathrm{d})\,K(q^\mathrm{d})}_{\text{Comp. Elasticity}}, \tag{2.68}$$

leading to the closed-loop dynamics

$$M_\varphi(q)\,\ddot{\varphi} + \eta_\varphi(q,\dot{q})\,\dot{\varphi} + A^{-1}(q)\,K(q) - A^{-1}(q^{\mathrm{d}})\,K(q^{\mathrm{d}}) + D_\varphi(q)\,\dot{\varphi} = \tau_{\mathrm{fb}}(t,q,\dot{q}), \tag{2.69}$$

2

which we can rewrite, assuming that a PD term is part of the feedback as

$$M_\varphi(q)\,\ddot{\varphi} + \eta_\varphi(q,\dot{q})\,\dot{\varphi} + A^{-1}(h^{-1}(\varphi))\,K(h^{-1}(\varphi)) - A^{-1}(h^{-1}(\varphi^{\mathrm{d}}))\,K(h^{-1}(\varphi^{\mathrm{d}})) + K_{\mathrm{p}}\left(\varphi - \varphi^{\mathrm{d}}\right) + \left(D_\varphi(q) + K_{\mathrm{d}}\right)\,\dot{\varphi} = 0_n, \tag{2.70}$$

which is asymptotically stable if the elastic potential $\mathcal{U}_{\varphi,\mathrm{K}}(\varphi)$ is locally or globally convex

$$\frac{\partial^2}{\partial\varphi^2}\mathcal{U}_{\varphi,\mathrm{K}}(\varphi)\bigg|_{\varphi=\varphi^{\mathrm{d}}} + K_{\mathrm{p}} = \frac{\partial}{\partial\varphi}\left(A^{-1}(h^{-1}(\varphi))\,K(h^{-1}(\varphi))\right)\bigg|_{\varphi=\varphi^{\mathrm{d}}} + K_{\mathrm{p}} \succ 0. \tag{2.71}$$

This can be shown via the Lyapunov function (Della Santina et al., 2023, 2020c; Khalil, 2002)

$$\begin{aligned} V(\varphi,\dot{\varphi}) = \underbrace{\frac{1}{2}\,\dot{\varphi}^\top M_\varphi(h^{-1}(\varphi))\,\dot{\varphi}}_{\text{Kinetic Energy}} + \underbrace{\mathcal{U}_{\varphi,\mathrm{K}}(\varphi) - \mathcal{U}_{\varphi,\mathrm{K}}(\varphi^{\mathrm{d}})}_{\text{Centered Elastic Potential Energy}} \\ + \underbrace{K^\top(h^{-1}(\varphi^{\mathrm{d}}))\,A^{-\top}(h^{-1}(\varphi^{\mathrm{d}}))\left(\varphi^{\mathrm{d}} - \varphi\right)}_{\text{Correction Term}} + \underbrace{\frac{1}{2}(\varphi^{\mathrm{d}} - \varphi)^\top K_{\mathrm{p}}(\varphi^{\mathrm{d}} - \varphi)}_{\text{Artificial Potential Energy of Feedback}}, \end{aligned} \tag{2.72}$$

with the associated time derivative (Della Santina et al., 2023)

$$\dot{V}(\varphi,\dot{\varphi}) = -\dot{\varphi}^\top\left(D_\varphi(h^{-1}(\varphi)) + K_{\mathrm{d}}\right)\,\dot{\varphi} \leq 0 \quad \forall\,\varphi,\dot{\varphi} \in \mathbb{R}^n \tag{2.73}$$

If the condition from (2.71) is locally met for $\varphi = \varphi^{\mathrm{d}}$, then the setpoint φ^{d} is locally asymptotically stable. If the convexity condition is met $\forall\,\varphi \in \mathbb{R}^n$, then the setpoint is globally asymptotically stable.

Here, it is easy to see how the proportional feedback gain makes the robot stiffer and, with that, can potentially allow for the system to be stable even when the open-loop elastic potential is non-convex. Still, this comes at the cost of decreased compliance and decreased robustness against measurement noise. In the case of the actuation space elastic potential being convex, i.e., $\frac{\partial}{\partial\varphi}\left(A^{-1}(h^{-1}(\varphi))\,K(h^{-1}(\varphi))\right) \succ 0$, then the proportional feedback gains are not necessary and can also be chosen as zero. An interesting case is also when (i) a constant actuation matrix $A(q) = A$, which is, for example, the case for many pneumatically-actuated soft robots, together with (ii) and a convex configuration space elastic potential, as for example the case for linear elasticity $K(q) = S\,q$, where $S \succ 0$. Then, the closed-loop dynamics are given by

$$M_\varphi(q)\,\ddot{\varphi} + \eta_\varphi(q,\dot{q})\,\dot{\varphi} + S_\varphi(\varphi - \varphi^{\mathrm{d}}) + D_\varphi\,\dot{\varphi} = \tau_{\mathrm{fb}}(t,q,\dot{q}), \tag{2.74}$$

which renders φ^{d} to be globally asymptotically stable. We leverage such a controller in Chapter 6 of this thesis.

Setpoint Regulation with Cancellation of Static Forces

Akin to PD+ (Kelly and Carelli, 1996; Paden and Panja, 1988) controllers originally developed for rigid manipulators, we can fully cancel the static, including gravitational and elastic, forces of the soft robot dynamics via nonlinear feedback and achieve setpoint regulation solely based on the potential field established by the (proportional) feedback term. The associated control law is given by (Patterson et al., 2024; Pustina, 2025)

$$\tau_{\mathrm{mb}}(q) = \underbrace{A^{-1}(q)\,G(q)}_{\text{Cancel Gravity}} + \underbrace{A^{-1}(q)\,K(q)}_{\text{Cancel Elasticity}} \tag{2.75}$$

which results in the closed-loop dynamics

$$M_\varphi(q)\,\ddot{\varphi} + \eta_\varphi(q,\dot{q})\,\dot{\varphi} + D_\varphi(q)\,\dot{\varphi} = \tau_{\mathrm{fb}}(t,q,\dot{q}), \tag{2.76}$$

where the potential of the system is now fully dominated by the feedback controller. Such a closed-loop system with a proportional feedback term, for example,

$$M_\varphi(q)\,\ddot{\varphi} + \eta_\varphi(q,\dot{q})\,\dot{\varphi} + K_{\mathrm{p}}(\varphi - \varphi^{\mathrm{d}}) + D_\varphi(q)\,\dot{\varphi} = 0_n, \tag{2.77}$$

is for any positive proportional gain $K_{\mathrm{p}} \succ 0 \in \mathbb{R}^{n\times n}$ globally asymptotically stable - even for small gains. This can be proven using the Lyapunov function

$$V(\varphi,\dot{\varphi}) = \frac{1}{2}\,\dot{\varphi}^\top M_\varphi(h^{-1}(\varphi))\,\dot{\varphi} + \frac{1}{2}\left(\varphi - \varphi^{\mathrm{d}}\right)^\top K_{\mathrm{p}}\left(\varphi - \varphi^{\mathrm{d}}\right), \tag{2.78}$$

that exhibits an associated time derivative of

$$\dot{V}(\varphi,\dot{\varphi}) = -\dot{\varphi}^\top D_\varphi(h^{-1}(\varphi)))\,\dot{\varphi} \leq 0 \quad \forall \varphi,\dot{\varphi} \in \mathbb{R}^n, \tag{2.79}$$

as $D_\varphi(h^{-1}(\varphi))) \succ 0\,\forall\,\varphi \in \mathbb{R}^n$. By applying LaSalle's invariance principle, GAS with respect to the equilibrium φ^{d} can be established (Khalil, 2002). However, this guarantee of global convergence—even for configuration-dependent actuation matrices $A(q)$—may come at the expense of robustness. In particular, with low proportional feedback gains K_{p}, unmodeled system dynamics might dominate, leading to steady-state errors or even instability.

Trajectory Tracking with Feedforward Compensation

In the following, we will assume the actuation matrix to be constant with $A(q) = A$. Then, extending the previously presented setpoint regulator with feedforward compensation, we can design a trajectory tracking algorithm with feedforward compensation as (Della Santina et al., 2023; Kelly and Salgado, 1994)

$$\tau_{\mathrm{mb}}(t) = \underbrace{M_\varphi(q^{\mathrm{d}}(t))\,\ddot{\varphi}^{\mathrm{d}}(t) + \eta_\varphi(q^{\mathrm{d}}(t),\dot{q}^{\mathrm{d}}(t))\,\dot{\varphi}^{\mathrm{d}}(t) + D_\varphi\,\dot{q}^{\mathrm{d}}(t)}_{\text{Compensate Dynamic Forces}} + \underbrace{A^{-1}\left(G(q^{\mathrm{d}}(t)) + K(q^{\mathrm{d}}(t))\right)}_{\text{Compensate Static Forces}}. \tag{2.80}$$

This controller is locally asymptotically stable, assuming sufficiently large proportional and dissipative feedback gains $K_{\mathrm{p}}, K_{\mathrm{d}} \succ 0$. We refer the interested reader to Della Santina et al. (2023); Kelly and Salgado (1994) for the full proof. The advantages and disadvantages mirror the ones of the *Setpoint Regulation with Feedforward Compensation* controller: robust against modeling errors and measurement noise and requires relatively high error-based feedback gains that come with a loss of compliance.

Trajectory Tracking with Mixed State Feedback

In the following, we assume that the actuation matrix is constant, i.e., $A(q) = A$. Moreover, since this controller is particularly well-suited for soft robots with a convex elastic potential, we assume linear elasticity $K(q) = S\,q$ with $S \succ 0$. Note that the controller would still converge globally for any convex potential field—or locally for non-convex elastic potentials provided sufficiently high feedback gains are used. For a more detailed discussion, please refer to the section on *Setpoint Regulation with Gravity Cancellation and Compensation of Elastic Forces*.

A trajectory tracking controller that employs full state feedback, except for compensating instead of canceling elastic forces, can be formulated as (Della Santina et al., 2020c; Kelly and Carelli, 1996)

$$\tau_{\mathrm{mb}}(t,q,\dot{q}) = \underbrace{M_\varphi(q(t))\,\ddot{\varphi}^{\mathrm{d}}(t) + \eta_\varphi(q(t),\dot{q}(t))\,\dot{\varphi}^{\mathrm{d}}(t) + A^{-1}\,G(q(t))}_{\text{Rigid Manipulator Nonlinear PD/PD+}} + \underbrace{S_\varphi\,\varphi^{\mathrm{d}}(t) + D_\varphi\,\dot{\varphi}^{\mathrm{d}}(t)}_{\text{Shaping Soft Robot Impedance}}, \tag{2.81}$$

where $S_\varphi = A^{-1}\,S\,A^{-\top} \succ 0 \in \mathbb{R}^{n\times n}$. This leads to the closed-loop dynamics

$$M_\varphi(q(t))\left(\ddot{\varphi}(t) - \ddot{\varphi}^{\mathrm{d}}(t)\right) + \eta_\varphi(q(t),\dot{q}(t))\left(\dot{\varphi}(t) - \dot{\varphi}^{\mathrm{d}}(t)\right) + S_\varphi\left(\varphi(t) - \varphi^{\mathrm{d}}(t)\right) + D_\varphi\left(\dot{\varphi}(t) - \dot{\varphi}^{\mathrm{d}}(t)\right) = \tau_{\mathrm{fb}}(t,q,\dot{q}). \tag{2.82}$$

The goal is now to show that the closed-loop error dynamics, which are defined as

$$M_\varphi(q(t))\,\ddot{e}(t) + \left(\eta_\varphi(q(t),\dot{q}(t)) + D_\varphi\right)\,\dot{e}(t) + S_\varphi\,e(t) = \tau_{\mathrm{fb}}(t,q,\dot{q}), \tag{2.83}$$

with $e(t) = \varphi(t) - \varphi^{\mathrm{d}}(t) \in \mathbb{R}^n$, $\dot{e}(t) = \dot{\varphi} - \dot{\varphi}^{\mathrm{d}}(t)$, and $\ddot{e}(t) = \ddot{\varphi}(t) - \ddot{\varphi}^{\mathrm{d}}(t)$ asymptotically converge to zero: $\lim_{t\to\infty} e(t) = 0$. Even without an error-based feedback term (i.e., $\tau_{\mathrm{fb}}(t,q,\dot{q}) = 0$), the asymptotic convergence to the trajectory $(\varphi^{\mathrm{d}}, \dot{\varphi}^{\mathrm{d}})$ can be shown via the Lyapunov function

$$V(e,\dot{e}) = \frac{1}{2}\,\dot{e}^\top(t)\,M_\varphi(h^{-1}(\varphi^{\mathrm{d}}))\,\dot{e}(t) + \frac{1}{2}\,e^\top(t)\left(S_\varphi + K_{\mathrm{d}}\right)\,e(t), \tag{2.84}$$

that exhibits the time derivative

$$\dot{V}(\dot{e}) = -\dot{e}^\top\left(D_\varphi + K_{\mathrm{d}}\right)\,\dot{e} \leq 0 \quad \forall\,\dot{e} \in \mathbb{R}^n, \tag{2.85}$$

by applying Barbalat's Lemma (Della Santina et al., 2020c; Slotine et al., 1991). Here, $K_{\mathrm{p}}, K_{\mathrm{d}} \in \mathbb{R}^{n\times n} \succeq 0$ are any PD feedback gains that are optionally applied.

2.3.4 Model-Based Terms: The Underactuated Case

Preliminaries

In the following, we will consider the underactuated where $m < n$. As underactuation is not a primary focus of this thesis, we will only broach the subject. For a much more detailed discussion, we refer the interested reader to the seminal thesis by Pustina (2025).

First, we need to revisit the definition of attainable equilibria. In the case of underactuated soft robots, a setpoint $\varphi^{\mathrm{d}} = \begin{bmatrix}\varphi_{\mathrm{a}}^{\mathrm{d}\top} & \varphi_{\mathrm{u}}^{\mathrm{d}\top}\end{bmatrix}^\top = h(q^{\mathrm{d}})$, where $q^{\mathrm{d}} = \begin{bmatrix}q_{\mathrm{a}}^{\mathrm{d}\top} & q_{\mathrm{u}}^{\mathrm{d}\top}\end{bmatrix}^\top$, needs to be a solution to the equilibrium equation (Pustina, 2025)

$$\begin{bmatrix} A_{\mathrm{a}}^{-1}(q^{\mathrm{d}})\left(G_{\mathrm{a}}(q^{\mathrm{d}}) + K_{\mathrm{a}}(q^{\mathrm{d}})\right) \\ -A_{\mathrm{u}}(q^{\mathrm{d}})\,A_{\mathrm{a}}^{-1}(q^{\mathrm{d}})\left(G_{\mathrm{a}}(q^{\mathrm{d}}) + K_{\mathrm{a}}(q^{\mathrm{d}})\right) + G_{\mathrm{u}}(q^{\mathrm{d}}) + K_{\mathrm{u}}(q^{\mathrm{d}}) \end{bmatrix} = \begin{bmatrix} \tau^{\mathrm{ss}} \\ 0_{n-m} \end{bmatrix}. \tag{2.86}$$

Zero Dynamics in the Regulation Case

It is essential for the stability of the system to ensure that the zero dynamics of the system, i.e., the last $n-m$, unactuated DOFs of the closed-loop dynamics, remain bounded. If the zero dynamics are unstable, the system state will eventually diverge, even if initial stability is observed from an input-output perspective (Pustina, 2025). For a regulator, that drives the system towards $\varphi = \varphi^{\mathrm{d}}, \dot{\varphi} = 0$, the associated zero dynamics are given as (Pustina, 2025)

$$M_{\varphi,\mathrm{uu}}(q^{\mathrm{d_a}})\,\ddot{\varphi}_{\mathrm{u}} + \eta_{\varphi,\mathrm{u}}(q^{\mathrm{d_a}}, \dot{q}^{\mathrm{d_a}})\,\dot{\varphi}_{\mathrm{u}} + \partial_{\varphi_{\mathrm{u}}} \mathcal{U}_{\varphi}(\varphi^{\mathrm{d_a}}) + D_{\varphi,\mathrm{uu}}(q^{\mathrm{d_a}})\,\dot{\varphi}_{\mathrm{u}} = 0_{n-m} \tag{2.87}$$

where $q^{\mathrm{d_a}} = \begin{bmatrix} q_{\mathrm{a}}^{\mathrm{d}\top} & q_{\mathrm{u}}^{\top} \end{bmatrix}^{\top}$, $\dot{q}^{\mathrm{d_a}} = \begin{bmatrix} 0_m^{\top} & \dot{q}_{\mathrm{u}}^{\top} \end{bmatrix}^{\top}$, and $\varphi^{\mathrm{d_a}} = \begin{bmatrix} \varphi_{\mathrm{a}}^{\mathrm{d}\top} & \varphi_{\mathrm{u}}^{\top} \end{bmatrix}^{\top}$. The Lyapunov function

$$V(\varphi_{\mathrm{u}}, \dot{\varphi}_{\mathrm{d}}) = \frac{1}{2}\,\dot{\varphi}_{\mathrm{u}}^{\top} M_{\varphi,\mathrm{uu}}(h^{-1}(\varphi^{\mathrm{d_a}}))\,\dot{\varphi}_{\mathrm{u}} + \mathcal{U}_{\varphi}(\varphi^{\mathrm{d_a}}), \tag{2.88}$$

that is non-negative provided that both the elastic and gravitational forces are Lipschitz and upper-bounded in their magnitude (Pustina, 2025), exhibits the time derivative

$$\dot{V}(\varphi_{\mathrm{u}}, \dot{\varphi}_{\mathrm{u}}) = -\dot{\varphi}_{\mathrm{u}}^{\top} D_{\varphi,\mathrm{uu}}(q^{\mathrm{d_a}})\,\dot{\varphi}_{\mathrm{u}} \leq 0 \quad \forall\, q^{\mathrm{d_a}} \in \mathbb{R}^n, \dot{\varphi}_{\mathrm{u}} \in \mathbb{R}^{n-m}, \tag{2.89}$$

as $D_{\varphi,\mathrm{uu}}(q^{\mathrm{d_a}}) \succ 0$. Subsequently, it can be shown that these zero dynamics are bounded and converge to $(\varphi_{\mathrm{u}}, \dot{\varphi}_{\mathrm{u}}) = (\varphi_{\mathrm{u}}^{\mathrm{d}}, 0)$ (Pustina, 2025), where $\varphi_{\mathrm{u}}^{\mathrm{d}}$ is a solution to

$$\frac{\partial}{\partial \varphi_{\mathrm{u}}} \mathcal{U}_{\varphi}\left(\begin{bmatrix} \varphi_{\mathrm{a}}^{\mathrm{d}} \\ \varphi_{\mathrm{u}}^{\mathrm{d}} \end{bmatrix}\right) = -A_{\mathrm{u}}(h^{-1}(\varphi^{\mathrm{d}}))\,A_{\mathrm{a}}^{-1}(h^{-1}(\varphi^{\mathrm{d}}))\left(G_{\mathrm{a}}(h^{-1}(\varphi^{\mathrm{d}})) + K_{\mathrm{a}}(h^{-1}(\varphi^{\mathrm{d}})\right) + G_{\mathrm{u}}(h^{-1}(\varphi^{\mathrm{d}})) + K_{\mathrm{u}}(h^{-1}(\varphi^{\mathrm{d}}) = 0_{n-m}. \tag{2.90}$$

Setpoint Regulation with Feedforward Compensation

A setpoint regulator with feedforward compensation that shapes the potential of the actuated coordinates is given by (Pustina, 2025)

$$\tau_{\mathrm{mb}} = \left.\frac{\partial \mathcal{U}_{\varphi}(\varphi)}{\partial \varphi_{\mathrm{a}}}\right|_{\varphi=\varphi^{\mathrm{d}}} = A_{\mathrm{a}}^{-1}(q^{\mathrm{d}})\left(G_{\mathrm{a}}(q^{\mathrm{d}}) + K_{\mathrm{a}}(q^{\mathrm{d}})\right), \tag{2.91}$$

leading to the closed-loop dynamics

$$M_{\varphi}(q)\,\ddot{\varphi} + \begin{bmatrix} \eta_{\varphi,\mathrm{a}}(q,\dot{q}) \\ \eta_{\varphi,\mathrm{u}}(q,\dot{q}) \end{bmatrix}\dot{\varphi} + \begin{bmatrix} A_{\mathrm{a}}^{-1}(q)(G_{\mathrm{a}}(q)+K_{\mathrm{a}}(q)) - A_{\mathrm{a}}^{-1}(q^{\mathrm{d}})\left(G_{\mathrm{a}}(q^{\mathrm{d}})+K_{\mathrm{a}}(q^{\mathrm{d}})\right) + K_{\mathrm{p}}\left(\varphi_{\mathrm{a}} - \varphi_{\mathrm{a}}^{\mathrm{d}}\right) + K_{\mathrm{d}}\,\dot{\varphi}_{\mathrm{a}} \\ -A_{\mathrm{u}}(q)\,A_{\mathrm{a}}^{-1}(q)(G_{\mathrm{a}}(q)+K_{\mathrm{a}}(q)) + G_{\mathrm{u}}(q) + K_{\mathrm{u}}(q) \end{bmatrix} + \begin{bmatrix} D_{\varphi,\mathrm{a}}(q) \\ D_{\varphi,\mathrm{u}}(q) \end{bmatrix}\dot{\varphi} = \begin{bmatrix} 0_m \\ 0_{n-m} \end{bmatrix}, \tag{2.92}$$

where we assumed that the feedback controller $\tau_{\mathrm{fb}}(\varphi, \dot{\varphi})$ contains a PD with gains $K_{\mathrm{p}}, K_{\mathrm{d}} \in \mathbb{R}^{m\times m}$. These closed-loop dynamics are equivalent to

$$\begin{bmatrix} M_{\varphi,\mathrm{aa}}(q) & M_{\varphi,\mathrm{au}}(q) \\ M_{\varphi,\mathrm{ua}}(q) & M_{\varphi,\mathrm{uu}}(q) \end{bmatrix}\ddot{\varphi} + \begin{bmatrix} \eta_{\varphi,\mathrm{a}}(q,\dot{q}) \\ \eta_{\varphi,\mathrm{u}}(q,\dot{q}) \end{bmatrix}\dot{\varphi} + \begin{bmatrix} \partial_{\varphi_{\mathrm{a}}} \mathcal{U}_{\varphi}(\varphi) - \partial_{\varphi_{\mathrm{a}}} \mathcal{U}_{\varphi}(\varphi^{\mathrm{d}}) + K_{\mathrm{p}}\left(\varphi_{\mathrm{a}} - \varphi_{\mathrm{a}}^{\mathrm{d}}\right) + K_{\mathrm{d}}\,\dot{\varphi}_{\mathrm{a}} \\ \partial_{\varphi_{\mathrm{u}}} \mathcal{U}_{\varphi}(\varphi) \end{bmatrix} + \begin{bmatrix} D_{\varphi,\mathrm{a}}(q) \\ D_{\varphi,\mathrm{u}}(q) \end{bmatrix}\dot{\varphi} = \begin{bmatrix} 0_m \\ 0_{n-m} \end{bmatrix}. \tag{2.93}$$

We can devise the Lyapunov candidate

$$\begin{aligned} V(\varphi, \dot{\varphi}) = \underbrace{\frac{1}{2}\,\dot{\varphi}^{\top} M_{\varphi}(h^{-1}(\varphi))\,\dot{\varphi}}_{\text{Kinetic Energy}} + \underbrace{\mathcal{U}_{\varphi}(\varphi) - \mathcal{U}_{\varphi}(\varphi^{\mathrm{d}})}_{\text{Centered Potential Energy}} \\ + \underbrace{\left.\frac{\partial \mathcal{U}_{\varphi}(\varphi)}{\partial \varphi_{\mathrm{a}}}\right|_{\varphi=\varphi^{\mathrm{d}}}^{\top}\left(\varphi_{\mathrm{a}}^{\mathrm{d}} - \varphi_{\mathrm{a}}\right)}_{\text{Correction Term on Actuated Coord.}} + \underbrace{\frac{1}{2}(\varphi_{\mathrm{a}} - \varphi_{\mathrm{a}}^{\mathrm{d}})^{\top} K_{\mathrm{p}}\,(\varphi_{\mathrm{a}} - \varphi_{\mathrm{a}}^{\mathrm{d}})}_{\text{Artificial Potential Energy}}, \end{aligned} \tag{2.94}$$

which is valid if

$$\left.\frac{\partial^2 \mathcal{U}_\varphi(\varphi)}{\partial \varphi^2}\right|_{\varphi=\varphi^{\mathrm{d}}} + \begin{bmatrix} K_{\mathrm{p}} & 0_{m\times(n-m)} \\ 0_{(n-m)\times m} & 0_{(n-m)\times(n-m)} \end{bmatrix} \succ 0, \tag{2.95}$$

which can be locally ensured by choosing $K_{\mathrm{p}} \succ 0$ sufficiently large (Pustina, 2025). The time derivative of the Lyapunov function is given as

$$\dot{V}(\varphi,\dot{\varphi}) = -\dot{\varphi}^\top D_\varphi(h^{-1}(\varphi))\,\dot{\varphi} - \dot{\varphi}_{\mathrm{a}}^\top K_{\mathrm{d}}\,\dot{\varphi}_{\mathrm{a}}, \succ 0 \quad \forall\, \varphi,\dot{\varphi} \in \mathbb{R}^m, K_{\mathrm{d}} \succeq 0, \tag{2.96}$$

As previously mentioned, the statically attainable equilibrium $(\varphi^{\mathrm{d}}, 0_n)$ is locally asymptotically stable for sufficiently large proportional gains K_{p}. If the robot is elastically dominated with

$$\frac{\partial^2 \mathcal{U}_\varphi(\varphi)}{\partial \varphi_{\mathrm{u}}^2} = \frac{\partial}{\partial \varphi_{\mathrm{u}}}\left(-A_{\mathrm{u}}(h^{-1}(\varphi))A_{\mathrm{a}}^{-1}(h^{-1}(\varphi))\left(G_{\mathrm{a}}(h^{-1}(\varphi))+K_{\mathrm{a}}(h^{-1}(\varphi))\right)+G_{\mathrm{u}}(h^{-1}(\varphi))+K_{\mathrm{u}}(h^{-1}(\varphi))\right) \succ 0 \quad \forall\, \varphi \in \mathbb{R}^n, \tag{2.97}$$

the regulator additionally achieves GAS for a sufficiently large $K_{\mathrm{p}} \succ 0$ (Pustina, 2025).

Remark 1. *The controller in* (2.91) *is equivalent to the energy-shaping controller* $\tau_{\mathrm{mb}} = A^{\mathrm{L}}(q^{\mathrm{d}})\left(K(q^{\mathrm{d}})+G(q^{\mathrm{d}})\right)$ *as proposed by Della Santina et al. (2023) iff the left inverse is defined as* $A^{\mathrm{L}} = A_{\mathrm{a}}^{-1}\begin{bmatrix}\mathbb{I}_m & 0_{m\times(n-m)}\end{bmatrix}$.

Proof. We can substitute $A^{\mathrm{L}}(q^{\mathrm{d}})$ until we obtain (2.91)

$$\begin{aligned}\tau_{\mathrm{mb}} &= A^{\mathrm{L}}(q^{\mathrm{d}})\left(K(q^{\mathrm{d}})+G(q^{\mathrm{d}})\right),\\ &= A_{\mathrm{a}}^{-1}(q^{\mathrm{d}})\begin{bmatrix}\mathbb{I}_m & 0_{m\times(n-m)}\end{bmatrix}\begin{bmatrix}K_{\mathrm{a}}(q^{\mathrm{d}})+G_{\mathrm{a}}(q^{\mathrm{d}})\\ K_{\mathrm{u}}(q^{\mathrm{d}})+G_{\mathrm{u}}(q^{\mathrm{d}})\end{bmatrix},\\ &= A_{\mathrm{a}}^{-1}(q^{\mathrm{d}})\left(G_{\mathrm{a}}(q^{\mathrm{d}})+K_{\mathrm{a}}(q^{\mathrm{d}})\right).\end{aligned} \tag{2.98}$$

Please note, though, that this statement generally does not hold if the pseudo-inverse is chosen as the left inverse such that $A^{\mathrm{L}} = A^{+} = (A^\top A)^{-1}A^\top$. □

Setpoint Regulation with Gravity Cancellation and Compensation of Elastic Forces for Elastically Decoupled Soft Robots

Elastically decoupled soft robots are defined as the generalized elastic force taking the form (Pustina, 2025)

$$\frac{\partial \mathcal{U}_{\varphi,\mathrm{K}}}{\partial \varphi}(\varphi) = \begin{bmatrix}\frac{\partial \mathcal{U}_{\varphi,\mathrm{K}}}{\partial \varphi_{\mathrm{a}}}(\varphi_{\mathrm{a}})\\ \frac{\partial \mathcal{U}_{\varphi,\mathrm{K}}}{\partial \varphi_{\mathrm{u}}}(\varphi_{\mathrm{u}})\end{bmatrix} = \begin{bmatrix} A_{\mathrm{a}}^{-1}(h^{-1}(\varphi_{\mathrm{a}}))K_{\mathrm{a}}(h^{-1}(\varphi_{\mathrm{a}}))\\ -A_{\mathrm{u}}(h^{-1}(\varphi_{\mathrm{u}}))\,A_{\mathrm{a}}^{-1}(h^{-1}(\varphi_{\mathrm{u}}))K_{\mathrm{a}}(h^{-1}(\varphi_{\mathrm{u}}))+K_{\mathrm{u}}(h^{-1}(\varphi_{\mathrm{u}}))\end{bmatrix}, \tag{2.99}$$

which is for example the case for $A(q) = A$ and linear elasticity $K(q) = S\,q$, where $S \in \mathbb{R}^{n\times n}$ is defined in a decoupled form

$$S = \begin{bmatrix} S_{\mathrm{aa}} & 0_{m\times(n-m)}\\ 0_{(n-m)\times m} & S_{\mathrm{uu}}\end{bmatrix}. \tag{2.100}$$

In this case, a regulator that cancels the gravitational forces and compensates the elastic forces on the actuated coordinates

$$\tau_{\mathrm{mb}}(\varphi) = \underbrace{\frac{\partial \mathcal{U}_{\varphi,\mathrm{G}}(\varphi)}{\partial \varphi_{\mathrm{a}}}}_{\text{Cancel Gravity}} + \underbrace{\left.\frac{\partial \mathcal{U}_{\varphi,\mathrm{K}}(\varphi)}{\partial \varphi_{\mathrm{a}}}\right|_{\varphi_{\mathrm{a}}=\varphi_{\mathrm{a}}^{\mathrm{d}}}}_{\text{Compensate Elasticity}} = \underbrace{A_{\mathrm{a}}^{-1}(q)\, G_{\mathrm{a}}(h^{-1}(\varphi))}_{\text{Cancel Gravity}} + \underbrace{A_{\mathrm{a}}^{-1}(h^{-1}(\varphi_{\mathrm{a}}^{\mathrm{d}}))\, K_{\mathrm{a}}(h^{-1}(\varphi_{\mathrm{a}}^{\mathrm{d}}))}_{\text{Compensate Elasticity}}, \tag{2.101}$$

renders the $\varphi_{\mathrm{a}} = \varphi_{\mathrm{a}}^{\mathrm{d}}$ to be globally asymptotically stable provided sufficiently large proportional feedback gains $K_{\mathrm{p}} \succeq 0$ (Pustina, 2025). This system exhibits the closed-loop dynamics

$$\begin{bmatrix} M_{\varphi,\mathrm{aa}}(q) & M_{\varphi,\mathrm{au}}(q) \\ M_{\varphi,\mathrm{ua}}(q) & M_{\varphi,\mathrm{uu}}(q) \end{bmatrix} \ddot{\varphi} + \begin{bmatrix} \eta_{\varphi,\mathrm{a}}(q,\dot{q}) \\ \eta_{\varphi,\mathrm{u}}(q,\dot{q}) \end{bmatrix} \dot{\varphi} + \begin{bmatrix} \partial_{\varphi_{\mathrm{a}}} \mathcal{U}_{\varphi,\mathrm{K}}(\varphi) - \partial_{\varphi_{\mathrm{a}}} \mathcal{U}_{\varphi,\mathrm{K}}(\varphi^{\mathrm{d}}) + K_{\mathrm{p}}\left(\varphi_{\mathrm{a}} - \varphi_{\mathrm{a}}^{\mathrm{d}}\right) + K_{\mathrm{d}}\, \dot{\varphi}_{\mathrm{a}} \\ \partial_{\varphi_{\mathrm{u}}} \mathcal{U}_{\varphi,\mathrm{G}}(\varphi) + \partial_{\varphi_{\mathrm{u}}} \mathcal{U}_{\varphi,\mathrm{K}}(\varphi) \end{bmatrix} + \begin{bmatrix} D_{\varphi,\mathrm{a}}(q) \\ D_{\varphi,\mathrm{u}}(q) \end{bmatrix} \dot{\varphi} = \begin{bmatrix} 0_m \\ 0_{n-m} \end{bmatrix}. \tag{2.102}$$

We can use the Lyapunov candidate

$$\begin{aligned} V(\varphi,\dot{\varphi}) = {} & \underbrace{\frac{1}{2}\, \dot{\varphi}^{\top} M_{\varphi}(h^{-1}(\varphi))\, \dot{\varphi}}_{\text{Kinetic Energy}} + \underbrace{\mathcal{U}_{\varphi,\mathrm{G}}(\varphi) - \mathcal{U}_{\varphi,\mathrm{G_a}}(\varphi) - \mathcal{U}_{\varphi,\mathrm{G_u}}(\varphi^{\mathrm{d}})}_{\text{Centered Gravitational Potential Energy}} + \underbrace{\mathcal{U}_{\varphi,\mathrm{K}}(\varphi) - \mathcal{U}_{\varphi,\mathrm{K}}(\varphi^{\mathrm{d}})}_{\text{Centered Elastic Potential Energy}} \\ & + \underbrace{\left.\frac{\partial \mathcal{U}_{\varphi}(\varphi)}{\partial \varphi_{\mathrm{a}}}\right|^{\top}_{\varphi=\varphi^{\mathrm{d}}} \left(\varphi_{\mathrm{a}}^{\mathrm{d}} - \varphi_{\mathrm{a}}\right)}_{\text{Correction Term on Actuated Coord.}} + \underbrace{\frac{1}{2}(\varphi_{\mathrm{a}} - \varphi_{\mathrm{a}}^{\mathrm{d}})^{\top} K_{\mathrm{p}} (\varphi_{\mathrm{a}} - \varphi_{\mathrm{a}}^{\mathrm{d}})}_{\text{Artificial Potential Energy}}, \end{aligned} \tag{2.103}$$

where

$$\mathcal{U}_{\varphi,\mathrm{G}}(\varphi) = \mathcal{U}_{\varphi,\mathrm{G_a}}(\varphi) + \mathcal{U}_{\varphi,\mathrm{G_u}}(\varphi), \tag{2.104}$$

with $\mathcal{U}_{\varphi,\mathrm{G_a}}(\varphi)$ and $\mathcal{U}_{\varphi,\mathrm{G_u}}(\varphi)$ the gravitational potential energy of the actuated and unactuated coordinates, respectively. These quantities can be computed as

$$\begin{aligned} \mathcal{U}_{\varphi,\mathrm{G_a}}(\varphi) &= \int A_{\mathrm{a}}^{-1}(q)\, G_{\mathrm{a}}(h^{-1}(\varphi))\, \mathrm{d}\varphi_{\mathrm{a}}, \\ \mathcal{U}_{\varphi,\mathrm{G_u}}(\varphi) &= \int -A_{\mathrm{u}}(h^{-1}(\varphi))\, A_{\mathrm{a}}^{-1}(h^{-1}(\varphi))\, G_{\mathrm{a}}(h^{-1}(\varphi)) + G_{\mathrm{u}}(h^{-1}(\varphi))\, \mathrm{d}\varphi_{\mathrm{u}}, \end{aligned} \tag{2.105}$$

and need to fulfill the properties

$$\frac{\partial \mathcal{U}_{\varphi,\mathrm{G_a}}(\varphi)}{\partial \varphi_{\mathrm{a}}} = \frac{\partial \mathcal{U}_{\varphi,\mathrm{G}}(\varphi)}{\partial \varphi_{\mathrm{a}}}, \qquad \frac{\partial \mathcal{U}_{\varphi,\mathrm{G_u}}(\varphi)}{\partial \varphi_{\mathrm{u}}} = \frac{\partial \mathcal{U}_{\varphi,\mathrm{G}}(\varphi)}{\partial \varphi_{\mathrm{u}}}. \tag{2.106}$$

This Lyapunov function has the associated time derivative

$$\dot{V}(\varphi,\dot{\varphi}) = -\dot{\varphi}^{\top} D_{\varphi}(h^{-1}(\varphi))\, \dot{\varphi} - \dot{\varphi}_{\mathrm{a}}^{\top} K_{\mathrm{d}}\, \dot{\varphi}_{\mathrm{a}}, \succ 0 \quad \forall\, \varphi, \dot{\varphi} \in \mathbb{R}^m, K_{\mathrm{d}} \succeq 0, \tag{2.107}$$

to analyze the stability properties in more detail.

2.3.5 Unified Controllers

In the following, we will discuss some common closed-form model-based controllers that cannot (easily) be separated into a model-based term and an error-based feedback term. For simplicity, we assume in the following the fully actuated case - i.e., $n = m$, $\varphi = \varphi_a$, $M_\varphi(q) = M_{\varphi,a}(q)$, etc.

P-satI-D+Potential Shaping

The *P-satI-D+Potential Shaping* controller consists of *P-satI-D* feedback term and the *Set-point Regulation with Feedforward Compensation* model-based term. As we frequently use this controller throughout this thesis, such as in Chapter 6 (underactuated version), and Chapter 10 & 11, we will specify below the control law, assuming a constant actuation matrix $A(q) = A$

$$\tau(t,q,\dot{q}) = \underbrace{A^{-1}\left(G(q^{\mathrm{d}}) + K(q^{\mathrm{d}})\right)}_{\text{Model-Based Feedforward}} + \underbrace{K_{\mathrm{p}}\left(\varphi^{\mathrm{d}}(t) - \varphi(t)\right) + K_{\mathrm{i}}\int_0^t \tanh\left(\gamma\left(\varphi^{\mathrm{d}}(t') - \varphi(t')\right)\right)\,\mathrm{d}t' + K_{\mathrm{d}}\left(\dot{\varphi}^{\mathrm{d}}(t) - \dot{\varphi}(t)\right)}_{\text{Model-Free Feedback Term: P-satI-D}}, \tag{2.108}$$

which generates the closed-loop dynamics

$$\begin{aligned} &M_\varphi(q)\,\ddot{\varphi} + \eta_\varphi(q,\dot{q})\,\dot{\varphi} + A^{-1}\left(G(q) + K(q) - G(q^{\mathrm{d}}) - K(q^{\mathrm{d}})\right) + D_\varphi\,\dot{\varphi} \\ &+K_{\mathrm{p}}\left(\varphi(t) - \varphi^{\mathrm{d}}(t)\right) + K_{\mathrm{d}}\left(\dot{\varphi}(t) - \dot{\varphi}^{\mathrm{d}}(t)\right) + K_{\mathrm{i}}\int_0^t \tanh\left(\gamma\left(\varphi(t') - \varphi^{\mathrm{d}}(t')\right)\right)\,\mathrm{d}t' = 0_n, \end{aligned} \tag{2.109}$$

where $K_{\mathrm{p}}, K_{\mathrm{i}}, K_{\mathrm{d}} \succeq 0 \in \mathbb{R}^{n\times n}$ are the proportional, integral, and derivative feedback gains, respectively.

Feedback Linearization via Computed Torque

The idea behind (full) feedback linearization (Slotine and Li, 1987; Spong et al., 2020) is to cancel the existing dynamics of the robot and ensure that the closed-loop system exhibits linear dynamics - usually established via a PD term. A computed torque controller for soft robots assumes the form

$$\tau(t,q,\dot{q}) = M_\varphi(q)\left(K_{\mathrm{p}}\left(\varphi^{\mathrm{d}} - \varphi\right) + K_{\mathrm{d}}\left(\dot{\varphi}^{\mathrm{d}} - \dot{\varphi}\right) + \ddot{\varphi}^{\mathrm{d}}\right) + \left(\eta_\varphi(q,\dot{q}) + D_\varphi(q)\right)\,\dot{\varphi} + A^{-1}(q)(G(q) + K(q)), \tag{2.110}$$

where $K_{\mathrm{p}}, K_{\mathrm{d}} \in \mathbb{R}^{n\times n}$ are proportional and derivative control gains for tracking the desired actuation coordinate and actuation velocity, respectively. The closed-loop dynamics read as

$$M_\varphi(q)\left(K_{\mathrm{p}}\left(\varphi - \varphi^{\mathrm{d}}\right) + K_{\mathrm{d}}\left(\dot{\varphi} - \dot{\varphi}^{\mathrm{d}}\right) + \left(\ddot{\varphi} - \ddot{\varphi}^{\mathrm{d}}\right)\right) = 0. \tag{2.111}$$

After defining the control error as $e(t) = \varphi(t) - \varphi^{\mathrm{d}}(t)$ with $\dot{e}(t) = \dot{\varphi}(t) - \dot{\varphi}^{\mathrm{d}}(t)$ and $\ddot{e}(t) = \ddot{\varphi}(t) - \ddot{\varphi}^{\mathrm{d}}(t)$, the dynamics of (2.111) can be rewritten as

$$\ddot{e}(t) + K_{\mathrm{d}}\,\dot{e}(t) + K_{\mathrm{p}}\,e(t) = 0, \tag{2.112}$$

which lets us notice that now linear dynamics govern with stiffness K_p and damping factor K_d govern the closed-loop system behavior. If we assume scalar control gains $K_\mathrm{p}, K_\mathrm{d} \in \mathbb{R}_+$, these error dynamics converge exponentially fast to the global equilibrium $e(t) = 0$ with decay time $\frac{1}{K_\mathrm{d}}$. Specifically, we can always choose the control such that the closed-loop system exhibits a critically damped behavior $\frac{K_\mathrm{d}}{2} = \sqrt{K_\mathrm{p}}$. The evolution of the error is then given in closed form as

$$e(t) = \left(e_0 + \left(\dot{e}_0 + \frac{K_\mathrm{d}}{2} e_0 \right) (t - t_0) \right) e^{-\frac{K_\mathrm{d}}{2}(t-t_0)}, \tag{2.113}$$

where $e_0, \dot{e}_0$ are the initial error and its velocity at t_0.

As seen in this derivation, computed torque controllers exhibit very nice stability and convergence properties. However, this comes at the cost of requiring high control rates, precise actuator motion, high control effort, and, most importantly, very accurate modeling of the system dynamics. In case of modeling errors, the stability characteristics can be lost, and the unmodelled dynamics can quickly become dominant. So far, computed torque controllers have been verified for fully actuated soft robots in simulation (Boyer et al., 2006; Pustina et al., 2024b), but to the best of our knowledge, not yet experimentally.

Operational Space Impedance Control

An operational space impedance controller uses partial feedback linearization to cancel out the original task dynamics and replace them with a new set of linear dynamics in operational space $x \in \mathbb{R}^n$. These new dynamics are defined by a stiffness matrix $K_\mathrm{x} \in \mathbb{R}^{o \times o}$ and a damping matrix $D_\mathrm{x} \in \mathbb{R}^{o \times o}$, which together drive the closed-loop system toward the desired operational space reference $x^\mathrm{d} \in \mathbb{R}^o$ (Della Santina et al., 2020c; Khatib, 1987).

$$\begin{aligned} \tau = \underbrace{A^{-1}(q)}_{J_\mathrm{h}^{-\top}(q)} \Big(& \underbrace{J^\top(q) J_\mathrm{M}^{+\top}(q) (K(q) + D\,\dot{q})}_{\text{Cancel Elastic \& Diss. Forc. Acting on Task}} + \underbrace{G(q)}_{\text{Cancel Gravity}} \Big) \\ + A^{-1}(q) \Big(& \underbrace{J^\top(q)\, \eta_\mathrm{x}(q,\dot{q}) \left(\mathbb{I}_n - J_\mathrm{M}^{+}(q) J(q) \right) \dot{q}}_{\text{Cancel Coupling of Null-Space Coriolis Force on Task}} + \underbrace{J^\top(q) \left(K_\mathrm{x} (x^\mathrm{d} - x) - D_\mathrm{x}\, \dot{x} \right)}_{\text{PD for Shaping Operational Space Impedance}} \Big). \end{aligned} \tag{2.114}$$

As noted earlier, the impedance controller relies on several key assumptions. It presumes (a) full actuation (i.e., $n = m$), (b) that the matrix $A(q) \in \mathbb{R}^{n \times n}$ is invertible (Della Santina et al., 2020c), (c) that the operational space—characterized by coordinates $x \in \mathbb{R}^o$—has lower dimensionality than the configuration space (i.e., $m \leq n$), and (d) that the null space is asymptotically stable, a condition typically met when $K(q) = S\,q$ with $S \succ 0$ and $D \succ 0$.

In the following, we will guide the reader step-by-step through the control design. First, we cancel the gravitational forces of the system via $\tau_\mathrm{q} = G(q)$. Please note that this immediately ensures that gravity is not acting anymore on the closed-loop null-space dynamics, which could potentially cause instability. Now, the updated operational and null space dynamics are given by

$$\begin{bmatrix} \Lambda_\mathrm{x}(q) & 0_{o \times (n-o)} \\ 0_{(n-o) \times o} & \Lambda_\mathrm{n}(q) \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \dot{\nu}_\mathrm{n} \end{bmatrix} + \begin{bmatrix} \eta_\mathrm{xx}(q,\dot{q}) & \eta_\mathrm{xn}(q,\dot{q}) \\ \eta_\mathrm{nx}(q,\dot{q}) & \eta_\mathrm{nn}(q,\dot{q}) \end{bmatrix} \begin{bmatrix} \dot{x} \\ \nu_\mathrm{n} \end{bmatrix} + \begin{bmatrix} J_\mathrm{M}^{+\top}(q) \\ Z(q) \end{bmatrix} (K(q) + D\,\dot{q}) = \begin{bmatrix} f_\mathrm{x} \\ f_\mathrm{n} \end{bmatrix}. \tag{2.115}$$

Next, we devise an operational space force $f_{\mathrm{x}} \in \mathbb{R}^o$

$$f_{\mathrm{x}} = \underbrace{J_{\mathrm{M}}^{+\top}(q)\left(K(q) + D\,\dot{q}\right)}_{f_{\mathrm{KD}}} + \underbrace{\eta_{\mathrm{x}}(q,\dot{q})\left(\mathbb{I}_n - J_{\mathrm{M}}^{+}(q)J(q)\right)\,\dot{q}}_{f_{\eta_{\mathrm{xn}}}} + \underbrace{K_{\mathrm{x}}\left(x^{\mathrm{d}} - x\right) - D_{\mathrm{x}}\,\dot{x}}_{f_{\mathrm{PD}}}, \tag{2.116}$$

2

that cancels the existing operational space dynamics, decouples the remaining null space interactions, and establishes new linear dynamics with a tunable impedance and an asymptotically stable equilibrium at x^{d} via a PD term. In more detail, the force f_{KD} eliminates the inherent elastic and dissipative forces in the operational space. Next, the term $f_{\eta_{\mathrm{xn}}}$ is designed to remove any residual coupling from the null space dynamics on the operational space, specifically by canceling the null space Coriolis forces $\eta_{\mathrm{xn}}(q,\dot{q})\,\nu_{\mathrm{n}}$. To avoid computing the null space online, we bypass its derivation by performing the substitution outlined in (Della Santina et al., 2020c; Ott, 2008).

$$\begin{aligned} f_{\eta_{\mathrm{xn}}} &= \eta_{\mathrm{xn}}(q,\dot{q})\,\nu_{\mathrm{n}} = \eta_{\mathrm{x}}(q,\dot{q})\,\dot{q} - \eta_{\mathrm{xx}}(q,\dot{q})\,\dot{q}, \\ &= \eta_{\mathrm{x}}(q,\dot{q})\,\dot{q} - \eta_{\mathrm{x}}(q,\dot{q})J_{\mathrm{M}}^{+}(q)\,\dot{x} = \eta_{\mathrm{x}}(q,\dot{q})\,\dot{q} - \eta_{\mathrm{x}}(q,\dot{q})J_{\mathrm{M}}^{+}(q)J(q)\,\dot{q} \\ &= \eta_{\mathrm{x}}(q,\dot{q})\left(\mathbb{I}_n - J_{\mathrm{M}}^{+}(q)J(q)\right)\,\dot{q}, \end{aligned} \tag{2.117}$$

as $\dot{x} = J(q)\,\dot{q}$ and $\eta_{\mathrm{xx}}(q,\dot{q}) = \eta_{\mathrm{x}}(q,\dot{q})J_{\mathrm{M}}^{+}(q)$. The removal of the existing operational space dynamics allows us then to establish the desired impedance. Specifically, PD term f_{PD} creates an artificial potential field $\mathcal{U}_{\mathrm{x}}(x) = \frac{1}{2}x^{\top}K_{\mathrm{x}}\,x$ and establishes dissipation which ensures asymptotic stability of the operational space dynamics. This forcing f_{x} results in the closed-loop task and null space dynamics

$$\begin{bmatrix} \Lambda_{\mathrm{x}}(q) & 0_{o\times(n-o)} \\ 0_{(n-o)\times o} & \Lambda_{\mathrm{n}}(q) \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \dot{\nu}_{\mathrm{n}} \end{bmatrix} + \begin{bmatrix} \eta_{\mathrm{xx}}(q,\dot{q}) & 0_{o\times(n-o)} \\ \eta_{\mathrm{nx}}(q,\dot{q}) & \eta_{\mathrm{nn}}(q,\dot{q}) \end{bmatrix} \begin{bmatrix} \dot{x} \\ \nu_{\mathrm{n}} \end{bmatrix} + \begin{bmatrix} K_{\mathrm{x}}(x - x^{\mathrm{d}}) - D_{\mathrm{x}}\,\dot{x} \\ Z(q)\left(K(q) + D\,\dot{q}\right) \end{bmatrix} = 0_n. \tag{2.118}$$

We can extract several interesting observations by analyzing these closed-loop dynamics in more detail. First, the operational space dynamics

$$\Lambda_{\mathrm{x}}(q)\,\ddot{x} + \eta_{\mathrm{xx}}(q,\dot{q})\,\dot{x} + K_{\mathrm{x}}\left(x - x^{\mathrm{d}}\right) + D_{\mathrm{x}}\,\dot{x} = 0_o, \tag{2.119}$$

now mirror the ones of coupled damped harmonic oscillators. Furthermore, similar to the computed torque controller, we can readily demonstrate that the closed-loop operational-space dynamics converge exponentially fast to the global asymptotic equilibrium x^{d}[11]. In line with the previously introduced setpoint regulation controllers that cancel gravitational forces while compensating for elastic forces, we do not entirely eliminate the null space dynamics. Instead, we enhance robustness against disturbances and modeling errors by preserving the system's natural elasticity and dissipation while only removing the gravitational forces. This approach yields closed-loop null-space dynamics that settle in steady-state at the (local) minimum of their potential field.

After designing the desired forces in operational space f_{x}, the control input can be derived via a projection into configuration space. First, f_{x} is mapped into configuration space, resulting in a combination of the projected operational space forces and gravity cancellation: $\tau_{\mathrm{q}} = J^{\top}(q)\,f + G(q)$. Finally, by applying the inverse of the actuation matrix,

[11] Please note that this does not automatically prove the stability of the null-space dynamics.

the control input based on the desired configuration space torques is obtained as $\tau = A^{-1}(q)\,\tau_{\mathrm{q}}$.

We propose a variation of this operational space impedance controller tailored to the Cartesian control of underactuated planar HSA robots in Chapter 6.

3

Towards Quantifying the Safety of Soft Robots

Foreword. Traditionally, the robotics community has focused on achieving safety through computational intelligence—constraining motions within safe state sets, avoiding obstacles, and using compliant controllers like impedance control. In contrast, one of the most fascinating and promising features of soft robots is their intrinsic compliance due to material softness. However, the safety achieved by soft robots has yet to be quantified, preventing us from determining whether they are indeed safe and how much safer they are compared to (collaborative) robots—a crucial factor in motivating soft robotics research and development. Moreover, the community often emphasizes the correlation between material softness and safety, while overlooking other important factors such as mass density, operating conditions, and control policy. This lack of understanding about the factors that significantly influence soft robot safety often leads to overly soft designs that suffer from degraded performance in practical tasks (e.g., limited payload capacity, imprecise or oscillatory motion). Since control is a central focus of this thesis, it is vital to identify which controller characteristics affect the safety of the closed-loop system, an often overlooked aspect in the soft robotics domain, and how to design controllers that maximize performance while preserving safety. Therefore, this chapter focuses on deriving a quantitative safety metric for soft robots.

M.S. conceived and led the project, derived and implemented the safety metric, characterized its behavior, and wrote the manuscript. N.P. and F.S. contributed Figures 3.1 & 3.2. C.D.S. and G.Z. supervised the project and contributed funding.

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q} = Q$$

Abstract. *As we work toward integrating robots into human-centric environments, it is crucial to ensure safety without overly constraining robot design or behavior. Efforts to enhance robot safety have a long history, primarily focusing on improving control systems to make them safer and more compliant. Soft robots offer a fundamentally different approach to this challenge by achieving safety through the passive compliance of their entire structure. Despite significant advancements in soft robotics over recent years—spanning novel actuators to sophisticated model-based control methods—one of the field's core promises, safety, has rarely been systematically studied. Notably, there is a lack of a quantitative metric to evaluate how safe a soft robot truly is. Moreover, most existing studies oversimplify the concept of safety, equating it to material softness. In reality, the safety of a closed-loop system depends on multiple factors, such as topology and control strategies. This narrow focus often results in suboptimal soft robot designs, such as robots that are too soft to handle practical payloads effectively. A quantitative safety metric would enable designers to evaluate and balance the tradeoff between safety and performance, leading to more optimal designs that meet safety requirements without sacrificing functionality. In this chapter, we outline the essential criteria for a safety metric tailored to soft robots. We then propose a metric that assesses injury severity based on the maximum contact pressure during a collision, aligning with established standards for collaborative robots. Finally, we characterize the proposed safety metric on continuum soft robots modeled using the PCS parametrization.*

3.1 Introduction

As noted in Chapter 1, the soft robotics community often highlights the "intrinsic safety" and natural compliance of soft robots (Abidi and Cianchetti, 2017) as a key advantage over conventional rigid robots and even cobots (Laschi and Cianchetti, 2014; Rus and Tolley, 2015; Yasa et al., 2023). Yet, in contrast to established scientific practices, the safety improvements of continuum soft robots compared to rigid manipulators have not been rigorously quantified. We believe it is essential for the community to validate intrinsic compliance as a fundamental benefit of soft robots, thereby justifying continued investments in their research and development (Hawkes et al., 2021).

Moreover, in the absence of clear guidelines for determining the optimal level of safety, designers currently tend to favor overly compliant materials. Many treat the challenge as a trade-off between performance/precision and softness, assuming that higher material compliance automatically results in enhanced safety. However, we find that this oversimplifies the matter; in reality, the lightweight nature and numerous/infinite DOFs exhibited by soft robots significantly enhance their safety, and the control approach also has a considerable influence. Furthermore, choosing materials that are too soft can undermine the robot's effectiveness, leading to imprecise or oscillatory motions, reduced payload capacity, and an inability to apply sufficient force to the environment (Cianchetti et al., 2013; Hawkes et al., 2017; Iida and Laschi, 2011; Majidi, 2014; Mazzolai et al., 2022). We argue that framing the issue as merely a balance between performance and softness oversimplifies the problem, since the safety of soft robots depends on both their body (e.g., morphology) and their brain (e.g., control and perception systems). Formalizing and quantifying this safety–performance trade-off would enable designers to find an optimal balance that maximizes performance while ensuring the necessary safety, a consideration that is particularly critical in human-centered environments.

Beyond *safety-aware design*, we expect that a comprehensive safety metric will be invaluable for other applications, such as the *safety certification* of soft robotic designs and *safety-aware control*, where the controller adjusts its actions to continuously meet required safety levels. After discussing these potential applications in detail, we establish the requirements a safety metric must satisfy for these purposes. We then introduce a new model-based safety metric for soft robots by extending the injury criterion from the existing ISO norms for collaborative robots (Standard, 2016). Our approach modifies the underlying model to capture key characteristics of soft robots, such as their elastic structures—which deform under internal actuation and external forces—and the possibility of collisions occurring along their entire body. By modeling the contact interactions and collision dynamics between soft robots and humans, and analogously to ISO/TS 15066:2016 (Standard, 2016), we use the maximum contact pressure during a collision as a proxy for injury severity. Instead of relying on computationally expensive simulations, we develop a conservative approximation of the collision dynamics that allows us to compute the maximum contact pressure in closed form. Importantly, our safety metric considers the closed-loop system dynamics, incorporating the control policy when assessing injury severity. It is proposed in two variants: the Soft Robot Injury Severity Criterion (SRISC), which calculates injury severity for a known soft robot state, control input, and contact geometry (e.g., the specific point of contact and collision direction), making it especially suited for *safety-aware control* applications; and the Soft Robot Design Hazardousness

Criterion (SRDHC), which evaluates safety across all possible contact geometries and robot states within a given operating envelope. We envision the SRDHC as the metric of choice for both *safety-aware design* and *design safety certification.*

3.2 Towards Quantifying the Safety of Soft Robots

In the following, we contextualize the topic by reviewing the literature on how safety has been assessed and quantified in the realm of robotics before, where almost all prior work is on the safety of industrial and collaborative rigid robotic manipulators. Subsequently, we will motivate the need for a quantitative safety metric for soft robots by showcasing future applications that such a metric would enable. This will then allow us to define a list of requirements for characteristics that a safety metric needs to exhibit.

3.2.1 Background on Safety Criteria for Robotic Manipulators

A safety criterion is defined as a measure of the potential injury severity resulting from a collision between a machine and a human (Haddadin, 2013). Given the scarcity of literature on the safety quantification of soft robots (Abidi and Cianchetti, 2017), we will primarily concentrate on work related to establishing safety criteria for rigid, and specifically collaborative, robots (De Santis et al., 2008; Haddadin, 2013; Zinn et al., 2004a). In particular, we want to highlight the seminal work by Haddadin (2013) as well as the established international regulations that ensure safe robot (e.g., cobot) behavior in collaborative human-robot settings.

Motivation

Various factors have motivated the robotics community to assess robot safety and develop corresponding criteria (De Santis et al., 2008; Van Diepen and Shea, 2018). First, by understanding the key safety-related factors, we can enhance robotic designs and control algorithms (Albu-Schäffer et al., 2007; Bicchi and Tonietti, 2004; Bischoff et al., 2010; Mansfeld et al., 2018; Park et al., 2011; Zinn et al., 2004b). Second, quantifying the injury risks associated with robots enables the establishment of minimal safety standards and constraints on design and actuation—illustrated, for instance, by ISO 10218-1:2011 (ISO 10218-1: 2011, 2011) for industrial robots and ISO/TS 15066:2016 (Standard, 2016) for collaborative robots. This process allows designers and manufacturers to certify that their designs are *safe*, which is crucial for adoption by industrial customers and consumers. Third, modeling injury risk and explicitly setting operational constraints paves the way for safety-aware control and motion planning (Bertino et al., 2023; Ferraguti et al., 2020; Haddadin, 2013; Lacevic and Rocco, 2011; Lacevic et al., 2022; Pupa and Secchi, 2024; Standard, 2016).

There are several approaches to achieving such safety-aware control. One well-established method is the use of collision avoidance algorithms (Haddadin, 2013), which trigger evasive maneuvers when a force/torque observer detects contact or collision (De Luca et al., 2006; Haddadin et al., 2008a, 2011a). However, this reactive approach has its drawbacks—time delays (from observation and reaction times) and actuator limitations (due to system inertia) may sometimes prevent injuries from being avoided. Moreover, aiming to avoid contact altogether runs counter to the vision of close human-robot collaboration.

To address these shortcomings, another strategy is to incorporate safety criteria directly into control and motion planning (Haddadin, 2013). A common approach is to define kinematic constraints that inherently ensure safety. These constraints can be applied to the robot's joint-space or task-space positions—often referred to as danger zones in the literature (Lacevic and Rocco, 2011; Lacevic et al., 2022)—to maintain a safe separation distance from humans (Standard, 2016) and ensure that the robot can always be slowed or stopped in time. This can be achieved either by preventing the robot from entering unsafe areas, for example, using safety filters (Bertino et al., 2023; Haddadin et al., 2010b), or by requiring humans to steer clear of these zones during operation. Alternatively, kinematic constraints can be defined in terms of maximum joint- or task-space velocities (Ferraguti et al., 2020; Pupa and Secchi, 2024; Standard, 2016), enforced via direct saturation or optimization-based techniques—such as MPC (Pupa and Secchi, 2024) or CBF (Ferraguti et al., 2020)—which aim to preserve the desired task behavior (Haddadin et al., 2010b). Finally, dynamic constraints (e.g., maximum contact force or pressure) may also be formulated and enforced through MPC, CBFs (Ames et al., 2016), or additional safety filters (Hewing et al., 2020).

Classification of Collisions

Collisions between robots and humans are generally categorized as either blunt or sharp contacts (Haddadin, 2013; Haddadin et al., 2009, 2011b). In blunt contact, the robot engages with a human using a large, flat, or rounded surface, which spreads the force over a broader area. Although this distribution may lower the risk of penetration, it can still lead to injuries—such as bruises, contusions, lacerations (both crush and gash), internal injuries, or even fractures—if the impact energy is sufficiently high (Haddadin, 2013; Haddadin et al., 2009, 2011b). Conversely, sharp or acute contacts occur when the robot makes contact with a small, focused area, like a point, edge, or any protruding element with a fine geometry. Because the force is concentrated over a smaller area, sharp contact has a greater potential for causing lacerations (cuts, stabs, or punctures) or abrasion wounds, making even relatively low forces particularly dangerous (Haddadin et al., 2011a, 2010a).

Since sharp contact is most often caused by tool use or by other sharp elements along the body of a (soft) robot, it can typically be avoided through an appropriate robot surface design[1]. Consequently, our focus will primarily be on reviewing literature related to safety criteria for blunt contacts (Haddadin, 2013; Haddadin et al., 2011a). Typically, injury severity is analyzed separately for each body part (e.g., head, neck, chest, arm, eye, etc.) (Haddadin, 2013; Haddadin et al., 2008a), and safety criteria are often specialized to represent the injury severity on a specific area[2]. Finally, collisions can also be classified based on the contact scenario, distinguishing between quasi-static and dynamic interactions, and considering whether injuries result from direct impact, clamping within the robot structure, or secondary impacts with the environment following the collision (Haddadin et al., 2009, 2011b).

[1] According to Haddadin et al. (2011a), similar force-based techniques can be used to derive safety criteria for sharp contacts as for blunt contacts, with the primary difference being that the relevant safety thresholds need to be adjusted. For example, the pain/injury tolerance for uncovered skin is approximately 75 ± 45N and 12.4 ± 4.9mm for forces and deflection, respectively (Haddadin et al., 2011a, 2010a).

[2] One exception is soft-tissue injuries caused by sharp contact, which generally apply across various body parts (except for the eye), with the main variation being the lethal penetration depth (Haddadin et al., 2011a, 2010a).

Safety Criteria for Collaborative Robots

The safety criteria—often referred to as injury severity criteria—that have been proposed for rigid robots can be broadly categorized into acceleration-, force-, stress/pressure-, compression-, viscous-, and energy-based criteria (Haddadin, 2013; Haddadin et al., 2009).

Acceleration-based criteria are commonly employed to assess head injuries and, to a lesser extent, chest injuries (Haddadin, 2013). Most acceleration-based head injury severity measures were originally developed for automotive applications. For instance, the 3 ms Criterion requires that the maximum 3 ms-average head acceleration during a collision remains below 72 g (Got et al., 1978; Haddadin, 2013). Another notable example—and the most widely adopted head injury index—is the Head Injury Criterion (HIC) (Versace, 1971), which evaluates the maximum acceleration integrated over a 36 ms period during a collision (Haddadin, 2013). To our knowledge, the first adaptation of the HIC for the robotics community (Bicchi and Tonietti, 2004; Zinn et al., 2004a,b), later explored further by Bicchi et al. (2008); Haddadin et al. (2008b); Shin et al. (2008), spearheaded the research on safety criteria for rigid robotic manipulators (De Santis et al., 2008; Haddadin, 2013). However, subsequent studies have shown that many injury severity criteria originally designed for automotive use—and in particular the HIC—are not well-suited as safety measures for collaborative robots because (a) they are not properly calibrated for the accelerations typical in human–robot collisions but instead are focused assessing the fatality of impacts, and (b) even when the robots operate at maximum speed, which is approximately 2 m/s (Haddadin et al., 2009) for common cobots (Albu-Schäffer et al., 2007; Bischoff et al., 2010), the resulting injury level is assessed as very low on the EuroNCAP (EuroNCAP, 2004) scale. While this might be acceptable for the relatively rare car crashes, even minor injuries are unacceptable in human–robot collaboration settings (Haddadin et al., 2009, 2011b).

Force-based criteria (Kirschner et al., 2021a) evaluate the (maximum) contact force experienced during a collision and are often used to assess the severity of neck and chest injuries (Haddadin, 2013). In automotive literature, EuroNCAP (EuroNCAP, 2004) cites a critical neck force of 1100 N based on biomechanical and forensic studies (Haddadin et al., 2009; Madea and Brinkmann, 2019). ISO/TS 15066:2016 (Standard, 2016) establishes a threshold for minor injuries—derived from biomechanical studies with adults (Muttray et al., 2014b)—at quasi-static contact forces of $65-300$ N, with the lowest pain threshold applying to the face. It is important to note that in transient scenarios, humans can tolerate roughly twice these force levels. In these assessments, a contact area of approximately ~ 2 cm^2 is assumed (Standard, 2016).

Pressure/stress-based criteria (Haddadin, 2013; Wassink and Stramigioli, 2007) are closely related to force-based criteria, as they assess the maximum local ratio of contact force to contact area. According to the ISO norm (Standard, 2016), these criteria are pertinent for evaluating pain sensitivity and injury severity from blunt and even sharp impacts across most body parts. Biomechanical pain tolerance studies (Muttray et al., 2014b) have set the quasi-static contact threshold at $1.1-3$ MPa, with peak pressures measured at a resolution of 1 mm^2 (Standard, 2016).

Compression-based criteria are particularly prevalent for modeling chest injuries (Haddadin, 2013; Haddadin et al., 2007). For instance, the Compression Criterion (Kroell et al., 1974; Lobdell et al., 1973) stipulates that the maximum chest deformation should not exceed 22 mm. **Viscous-based criteria** take into account not only the (chest) deformation but also

the associated velocity (Haddadin, 2013; Haddadin et al., 2007). An example is the Viscous Criterion (Lau and Viano, 1986; Viano and Lau, 1985), also known as the soft-tissue criterion, which multiplies these two factors; however, it is prone to measurement noise (Haddadin et al., 2011b).

Energy-based criteria evaluate the (kinetic) energy transferred to a human during a collision with the robot (Haddadin, 2013; Haddadin et al., 2011b; Standard, 2016). Such energy criteria have been used particularly to assess head injury severity (Haddadin et al., 2009), where a kinetic energy transfer of just $50-100$ J can fracture the skull (Haddadin et al., 2009). Additionally, ISO/TS 15066:2016 (Standard, 2016) has set maximum transferred energy thresholds for other body parts—approximately 0.1 J for the face and 2.6 J for the pelvis. Although the strength of this safety criterion lies in its integration of transient forces over time, rather than just considering the (peak) contact pressure at a single moment, a drawback is that it does not factor in the contact area (e.g., blunt versus sharp contacts), which significantly influences injury severity in practice. This limitation is addressed by considering normalized energy, also known as energy density, where the threshold is set at $2.5\,\mathrm{J/cm^2}$ for soft tissue (Haddadin et al., 2011b). Energy density has also proven to be an effective indicator for eye injuries (Haddadin, 2013).

The collision between a robot and a human (or a specific body part) is typically modeled by representing them as two point masses, m_R and m_H, connected by a spring that captures the serial elasticity of both the robot's surface and the human soft tissue (Bicchi and Tonietti, 2004; Haddadin et al., 2009; Standard, 2016). This model facilitates the straightforward evaluation of the safety criteria introduced earlier, since, for example, the contact force is determined by the evolution of the spring force over time. The robot's effective mass, m_R (Kirschner et al., 2021b), is obtained by evaluating the system's reflected inertia at a given position using operational-space dynamics (Khatib, 1987) (see Section 2.2.3 for further details). The reflected inertia of the human, m_H, is provided for various body parts in ISO/TS 15066:2016 (Standard, 2016). However, this value is relevant only for contact scenarios in which the human is free to move (i.e., unconstrained). In contrast, collision scenarios involving a (partially) constrained human typically result in significantly higher injury severities. To represent the worst-case scenario—where the human body part is completely constrained (Haddadin et al., 2009, 2011b)—we would assume that $m_\mathrm{H} \gg m_\mathrm{R}$.

Finally, it is worth reiterating that safety criteria for rigid manipulators operating in human environments (e.g., cobots) have been standardized in ISO/TS 15066:2016 (Standard, 2016). More recently, the methodologies for measuring forces and pressures in physical human-robot interactions, particularly those involving pressure-force measurement devices, have been detailed in ISO/PAS 5672:2023 (Standard, 2023).

Towards Safety Criteria for Soft Robots

However, these existing methods for deriving safety metrics for collaborative robots cannot be directly applied to soft robots without modification. There are several reasons for this: (a) The aforementioned simplified point-mass models only account for the robot's inertia and implicitly assume that the collision point is free to move in space, meaning no forces act on the joints. In contrast, with soft robots—even when a collision is detected, and actuation is deactivated—substantial forces still arise from the structure's inherent elasticity. (b) Furthermore, while in rigid robots, the considerable mass ensures that contact

forces and transferred energy are primarily dominated by kinetic energy at the time of collision (Haddadin et al., 2009), in soft robots, steady-state forces such as elasticity and gravitational effects play a far more significant role. (c) Finally, existing literature on safety criteria for rigid robots mainly focuses on contact at the end-effector (Haddadin, 2013; Haddadin et al., 2009), where the highest velocities are typically observed. In contrast, soft robots require consideration of collisions along their entire body, particularly given their higher operational-space stiffness near the proximal end.

To the best of our knowledge, the only preliminary study on the intrinsic safety of soft robots is the initial work by Abidi and Cianchetti (2017), which approximates a soft robot as a Timoshenko beam (Timoshenko, 1921) and subsequently links the low stiffness of soft robots (Rus and Tolley, 2015) to reduced forces. However, this approach falls short as a safety criterion for several reasons: (a) the force is expressed as a PDE in time and spatial deformation, making practical evaluation challenging; (b) the Timoshenko-Ehrenfest beam theory (Timoshenko, 1921) accounts only for shear and bending deformations in a plane, whereas many soft robots also experience additional deformation modes such as axial and twist strains (Armanini et al., 2023); (c) it is tailored to small deflections and linear behavior, even though soft robots can exhibit large deformations and significant nonlinear behavior (Della Santina et al., 2023); and (d) it does not consider the actuation characteristics of the soft robot.

3.2.2 Potential Applications for a Safety Metric

We envision a safety metric for soft robots that will enable a variety of applications, as illustrated in Fig. 3.1. These applications can be categorized into two main domains: *safety-aware control* and *safety-aware design*.

In the realm of *safety-aware control*, we assume the soft robotic design is already established and focus on controlling its actions to maintain an acceptable safety level. This approach does not preclude contact or collisions with the environment; rather, it ensures that such interactions do not result in significant injuries. Examples include *safety filters* (Bertino et al., 2023), which allow the use of high-performance control policies, such as RL, that do not explicitly account for safety constraints while still guaranteeing safety by filtering or saturating the control input. Alternative methods like MPC (Hewing et al., 2020; Pupa and Secchi, 2024) or CBFs (Ames et al., 2016; Ferraguti et al., 2020)—referred to as *safety-constrained control*—integrate safety constraints directly into the control strategy.

Another promising direction unlocked by a safety metric is *safety-aware design*, which involves evaluating an integrated soft robot design for its safety. We see this as comprising two subcategories: *safety-aware design optimization*, which incorporates safety into the design process either through inequality constraints (ensuring a minimum safety level) or by maximizing safety via the cost function; and *safety certification*, where, after a design is finalized, manufacturers can certify that their product meets the necessary safety standards for specific applications (e.g., healthcare, agri-food, manufacturing, etc.). This approach aligns well with established safety standards for collaborative rigid robots as defined in ISO/TS 15066:2016 (Standard, 2016).

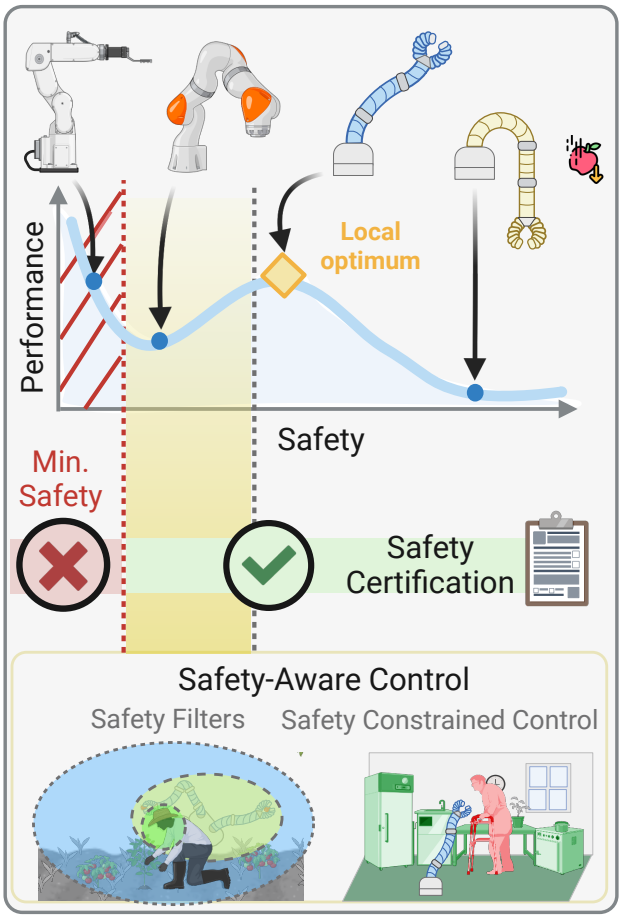


Figure 3.1: Future applications that a quantitative safety metric for soft robots would enable. **Top:** Here, we showcase how the safety metric could be used for *safety-aware design optimization* and, particularly, for analyzing and exploiting the performance vs. safety tradeoff. Subsequently, the safety metric could be used for certifying a given design as *safe* for the respective application. **Bottom:** The safety metric could also be used for *safety-aware control* - either by filtering the outputs of the controller without safety guarantees or by explicitly including the safety requirements as constraints during the optimization of the control input sequence (e.g., MPC, CBFs).

3.2.3 Requirements for a Soft Robotic Safety Metric

In the following, we outline several requirements that, in our opinion, a safety metric must satisfy to be well-suited for the applications described in Sec. 3.2.2. First (1), the metric should account for the dynamics inherent to continuum soft robots and their unique characteristics. For example, a primary difference between rigid and soft manipulators is that free-moving joints with integrated motors are replaced by a compliant structure that deforms under internal actuation and external forces. Consequently, any safety metric must consider the elastic and inertial properties generated by the distributed material along the robot's backbone. Secondly (2), the safety metric must evaluate collisions occurring anywhere along the soft robot's body—not just at the end-effector. This represents a significant departure from existing safety metrics for rigid or collaborative robots, which often focus solely on the end-effector due to its typically higher motion velocities (Haddadin et al., 2009, 2011b; Standard, 2016). In contrast, soft robots exhibit high Cartesian stiffness near their proximal end, making it necessary to assess safety along the entire structure. Next (3), any simplifying assumptions should yield a conservative safety estimate. For instance, while soft robot models used in the metric might employ a finite-dimensional approximation of the continuum shape, in reality, these flexible structures possess infinite degrees of freedom (Armanini et al., 2023; Della Santina et al., 2023). Therefore, a safety metric based on such approximations should tend to underestimate rather than overestimate the design's safety. Fourthly (4), the computation of the safety metric must be tractable, which is essential for safety-aware control and design applications that require evaluations on sub-second to second timescales, respectively. Finally, we highlight a few desirable characteristics: (5) in applications such as safety-aware control or design, it is advantageous if the safety metric is differentiable with respect to design parameters, robot states, and control inputs; and (6) ideally, the metric should also reflect how "safe" the soft robot's design (e.g., its friendly appearance) and behavior (e.g., smooth and predictable movements) are perceived by users. In addition to user studies, the evaluation of this metric might be enhanced by leveraging Vision-Language Models (VLMs) (Grattafiori et al., 2024; Touvron et al., 2023).

3.3 A Metric for Quantifying the Safety of Soft Robots

Thereafter, we propose a quantitative safety metric for *blunt* contacts (Haddadin et al., 2011b) between soft robots and humans that captures the particular characteristics that continuum soft robots exhibit (e.g., elasticity, actuation through their structure, Cosserat rod dynamics, etc.). Importantly, the presented safety metric fulfills all the mandatory requirements that we laid out previously. First, we state the necessary background on soft robotic dynamics and contact models. Subsequently, we derive the dynamics of a collision between the continuum soft robot and the human (body part). Next, we propose two flavors of the safety metric: (i) the Soft Robot Injury Severity Criterion (SRISC) captures the injury risk for a *given* contact geometry, soft robot state, and actuation sequence. We envision this criterion to be useful for control applications with safety guarantees. The second flavor, (ii) the Soft Robot Design Hazardousness Criterion (SRDHC), captures the inherent safety of an integrated soft robot design (e.g., also considering the control policy) and leverages the

SRISC for estimating the maximum injury risk over all possible contact geometries, feasible robot states, and actuation sequences. Apart from the procedure for formulating this safety metric, one of the key innovations here is that we identify a closed-form solution to the collision dynamics, which renders the computation of the SRISC to be computationally tractable.

3.3.1 Background on Soft Robot Dynamics and Contact Model

Following the Cosserat rod theory, we can capture the kinematic behavior of slender structures such as continuum soft robots by considering the deformations of the robot's backbone. As the 1D spatial deformations of this backbone are still an infinite-dimensional problem, the field has developed many methods, such as PCC (Webster III and Jones, 2010), PCS (Renda et al., 2018), and GVS (Renda et al., 2020), to describe such deformations with a finite number of finite vectors of configuration variables $q \in \mathbb{R}^n$. The associated forward kinematic model then allows us to define the geometric positional Jacobian $J_{\mathrm{p}}(q,s) \in \mathbb{R}^{3\times n}$, where $s \in (0, L]$ is the backbone abscissa/coordinate and L is the length of the entire continuum structure. Independent of the specific chosen kinematic model, the EOM of a continuum soft robot can often be stated as (Armanini et al., 2023; Della Santina et al., 2023)

$$M(q)\,\ddot{q} + C(q,\dot{q})\,\dot{q} + \partial_q \mathcal{U}(q) + D\,\dot{q} = A_\tau(q)\,\tau + \tau_{\mathrm{c}}, \tag{3.1}$$

$M(q) \in \mathbb{R}^{n\times n}$ and $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ consider the inertial and Coriolis effects of the soft robot system, respectively. $\partial_q \mathcal{U}(q) \in \mathbb{R}^n$ captures the forces stemming from the potential $\mathcal{U}(q) : \mathbb{R}^n \to \mathbb{R}$. Often times, the potential forces simplify to $\partial_q \mathcal{U}(q) = G(q) + Kq$, where $G(q) \in \mathbb{R}^n$ describes the gravitational forces, and $K \succ 0 \in \mathbb{R}^{n\times n}$ is the stiffness matrix. Dissipation is integrated through the damping matrix $D \succ 0 \in \mathbb{R}^{n\times n}$. $\tau(t,q,\dot{q}) \in \mathbb{R}^m$ contributes the actuation (determined by a control policy) that acts through the linear map $A_\tau(q) \in \mathbb{R}^{n\times m}$ on the generalized coordinates.

The term $\tau_{\mathrm{c}} \in \mathbb{R}^n$ collects all contributions by external contact forces on the generalized coordinates. In the following, we will assume that the soft robot is only in contact with the human at one discrete point and that only pure forces are reflected between the bodies during the contact (i.e., no Cartesian torques). Specifically, we assume that the contact occurs at the backbone abscissa $s_{\mathrm{c}} \in (0, L]$ and that the contact exhibits a constant surface normal of $n_{\mathrm{c}} \in \mathcal{S}^3$ which is a unit vector and, with that, $\mathcal{S}^3 = \{n_{\mathrm{c}} \in \mathbb{R}^3 : \|n_{\mathrm{c}}\|_2 = 1\}$. We now describe with $\delta_{\mathrm{c}} > 0$ a penetration between the soft robot and the soft tissue of the human. Then, the generalized torque acting on the soft robot as a consequence of the contact is given by $\tau_{\mathrm{c}} = -J^\top(q, s_{\mathrm{c}})\, n_{\mathrm{c}}\, f_{\mathrm{c}}(\delta_{\mathrm{c}}, \dot{\delta}_{\mathrm{c}}) = -J_{\mathrm{c}}^\top(q, s_{\mathrm{c}}, n_{\mathrm{c}})\, f_{\mathrm{c}}(\delta_{\mathrm{c}}, \dot{\delta}_{\mathrm{c}})$, where $f_{\mathrm{c}}(\delta_{\mathrm{c}}, \dot{\delta}_{\mathrm{c}}) \in \mathbb{R}_{\geq 0}$ is the scalar non-negative contact force. In the following, we will frequently omit the dependency of symbols, such as $J_{\mathrm{c}}(q)$, on the $(s_{\mathrm{c}}, n_{\mathrm{c}})$ to simplify the notation. While the formulation that we use in this chapter for formulating the safety metric is compatible with many of the contact models that have been studied in the literature, such as Hunt-Crossley (Aouaj et al., 2021; Hunt, 1975), Hertz (Johnson, 1987; Park et al., 2011; She et al., 2020b), etc., we will mainly focus in the following on a linear spring-damper contact model (Haddadin et al., 2009; Standard, 2016) given by

$$f_{\mathrm{c}}(\delta_{\mathrm{c}}, \dot{\delta}_{\mathrm{c}}) = \begin{cases} 0 & \delta_{\mathrm{c}} \leq 0, \\ k_{\mathrm{c}}\,\delta_{\mathrm{c}} + d_{\mathrm{c}}\,\dot{\delta}_{\mathrm{c}} & \delta_{\mathrm{c}} > 0, \end{cases} \tag{3.2}$$

where $k_c \in \mathbb{R}_{>0}$ is the contact stiffness and $d_c \in \mathbb{R}_{\geq 0}$ is the contact damping coefficient. If we assume the soft robot surface material and the human soft tissue to have spring constants and damping coefficients of $k_{R,surf}$, $k_{H,st}$, and d_R, d_H, respectively, then we can connect the spring-dampers in series

$$k_c = \left(\frac{1}{k_{R,surf}} + \frac{1}{k_{H,st}} \right)^{-1}, \qquad d_c = \left(\frac{1}{d_R} + \frac{1}{d_H} \right)^{-1}. \tag{3.3}$$

3

Please note that the effective spring constant of many human body parts is reported in ISO/TS 15066:2016 (Standard, 2016).

3.3.2 Collision Dynamics

We now progress towards a formulation of the collision dynamics as motions of the soft robot and the human body part along the contact surface normal n_c.

First, we describe the motion of the contact point of the soft robot with position and velocity $x_R, \dot{x}_R \in \mathbb{R}$. We can project the dynamics of (3.1) into this 1D motion through the expression $\dot{x}_R = J_c \dot{q}$ yielding the form (Della Santina et al., 2020c, 2019b; Khatib, 1987; Stölzle et al., 2024a)

$$\Lambda_c(q)\ddot{x}_R + \eta_c(q,\dot{q})\dot{x}_R + J_{c,M}^{+\top}(q)(\partial_q \mathcal{U}(q) + D\dot{q}) = J_{c,M}^{+\top}(q) A_\tau(q)\tau - f_c(\delta_c, \dot{\delta}_c), \tag{3.4}$$

where $J_{c,M}^{+\top}(q, s_c, n_c) = M^{-1}J_c^\top(J_c M^{-1} J_c^\top)^{-1} \in \mathbb{R}^{n\times 1}$ is the dynamically consistent pseudo-inverse, $\Lambda_c(q, s_c, n_c) = (J_c M^{-1} J_c^\top)^{-1} \in \mathbb{R}^{1\times 1}$ is the reflected inertia of the soft robot at the contact point (Haddadin et al., 2009; Standard, 2016), and $\eta_c(q, \dot{q}, s_c, n_c) = \Lambda_c(q)(J_c M^{-1} C - \dot{J}_c) \in \mathbb{R}^{1\times n}$ collects the Cartesian Coriolis and centrifugal terms (Khatib, 1987). As mentioned already previously, if not explicitly stated otherwise, we will in the following, to simplify the notation, drop the specific dependency on the contact geometry (s_c, n_c): $J_{c,M}^{+\top}(q) = J_{c,M}^{+\top}(q, s_c, n_c)$, $\Lambda_c(q) = \Lambda_c(q, s_c, n_c)$, etc.

Next, we move towards modeling the behavior of the human body part. In literature, the human body part is usually modeled as a point mass[3] m_H (Haddadin et al., 2011b; Standard, 2016) that moves in 1D along the surface normal of the contact with state $(x_H, \dot{x}_H)$. Instead, we take here a conservative approach and assume that the human body is constrained in its motion with velocity $v_H \in \mathbb{R}$ towards the soft robots (i.e., $m_H \gg \Lambda(q)\,\forall q$). This represents the *worst case*. After the coordinate change $\delta_c(t) = x_R(t) - x_H$, $\dot{\delta}_c = \dot{x}_R(t) + v_H$, where $x_H \in \mathbb{R}$ is the position of the soft tissue surface, and while only considering the case of contact (i.e., $\delta_c \geq 0$), the collision dynamics are given by

$$\Lambda_c(q)\ddot{\delta}_c + \eta_c(q,\dot{q})\dot{\delta}_c + J_{c,M}^{+\top}(q)(\partial_q \mathcal{U}(q) + D\dot{q}) = J_{c,M}^{+\top}(q) A_\tau(q)\tau - k_c\,\delta_c - d_c\,\dot{\delta}_c. \tag{3.5}$$

We are now interested in identifying the maximum force $f_c(t)$ that occurs during the entire time of the contact. Therefore, we can neglect any damping forces, such as $d_c\,\dot{\delta}_c$ and $D\,\dot{q}$, as they dissipate energy, and, therefore, reduce the maximum contact force. Furthermore, we assume that the Coriolis effects are sufficiently small and can be neglected as well. Finally,

[3] Please note that the effective mass of various human body parts is reported in ISO/TS 15066:2016 (Standard, 2016).

we assume that the change of configuration during the collision is sufficiently small such that the dynamic matrices can be approximated as constant:

$$m_{\mathrm{R}} = \Lambda_{\mathrm{c}}(q_{\mathrm{c}}^0) \approx \Lambda_{\mathrm{c}}(q), \qquad A_{\tau,\mathrm{c}} = J_{\mathrm{c,M}}^{+\top}(q_{\mathrm{c}}^0)\, A_\tau(q_{\mathrm{c}}^0) \approx J_{\mathrm{c,M}}^{+\top}(q)\, A_\tau(q), \quad \forall\, t \geq t_{\mathrm{c}}^0, \tag{3.6}$$

where the q_{c}^0 is the configuration of the robot at the beginning of the contact. The same assumption also allows us to linearize the potential forces of the soft robot with

$$f_{\mathcal{U}} = J_{\mathrm{c,M}}^{+\top}(q)\, \partial_q\, \mathcal{U}(q) \approx \underbrace{J_{\mathrm{c,M}}^{+\top}(q_{\mathrm{c}}^0)\, \partial_q\, \mathcal{U}(q_{\mathrm{c}}^0)}_{f_{\mathcal{U}}^{\mathrm{c}0}} + \underbrace{\frac{\partial}{\partial q} J_{\mathrm{c,M}}^{+\top}(q)\, \partial_q\, \mathcal{U}(q)\Big|_{q=q_{\mathrm{c}}^0} J_{\mathrm{c,M}}^{+}(q_{\mathrm{c}}^0)}_{k_{\mathrm{R}}}\, \delta_{\mathrm{c}}, \tag{3.7}$$

where $k_{\mathrm{R}} \in \mathbb{R}$ is the local stiffness of the system against small perturbations and $f_{\mathcal{U}}^{\mathrm{c}0}$ are the potential forces present at the start of the contact. Furthermore, we assume the actuation force to be constant, which can be easily accomplished by conservatively considering the maximum actuation force $f_\tau = \max_t |A_{\tau,\mathrm{c}}\, \tau(t)|$ that the robot experiences during the collision. Integrating the stated assumptions into the EOM results in the approximated collision dynamics (during contact)

$$m_{\mathrm{R}}\, \ddot{\delta}_{\mathrm{c}} + (k_{\mathrm{R}} + k_{\mathrm{c}})\, \delta_{\mathrm{c}} = f_\tau - f_{\mathcal{U}}^{\mathrm{c}0}. \tag{3.8}$$

To avoid computationally expensive simulations of the collision, we identify a closed-form solution to the collision dynamics

$$\begin{aligned}
\delta_{\mathrm{c}}(t) &= \left(\delta_{\mathrm{c}}^0 - \frac{f_\tau - f_{\mathcal{U}}^{\mathrm{c}0}}{k_{\mathrm{R}} + k_{\mathrm{c}}}\right) \cos\left(\sqrt{\frac{k_{\mathrm{R}} + k_{\mathrm{c}}}{m_{\mathrm{R}}}}\, t\right) + \dot{\delta}_{\mathrm{c}}^0 \sqrt{\frac{m_{\mathrm{R}}}{k_{\mathrm{R}} + k_{\mathrm{c}}}} \sin\left(\sqrt{\frac{k_{\mathrm{R}} + k_{\mathrm{c}}}{m_{\mathrm{R}}}}\, t\right) + \frac{f_\tau - f_{\mathcal{U}}^{\mathrm{c}0}}{k_{\mathrm{R}} + k_{\mathrm{c}}}, \\
\dot{\delta}_{\mathrm{c}}(t) &= -\sqrt{\frac{k_{\mathrm{R}} + k_{\mathrm{c}}}{m_{\mathrm{R}}}} \left(\delta_{\mathrm{c}}^0 - \frac{f_\tau - f_{\mathcal{U}}^{\mathrm{c}0}}{k_{\mathrm{R}} + k_{\mathrm{c}}}\right) \sin\left(\sqrt{\frac{k_{\mathrm{R}} + k_{\mathrm{c}}}{m_{\mathrm{R}}}}\, t\right) + \dot{\delta}_{\mathrm{c}}^0 \cos\left(\sqrt{\frac{k_{\mathrm{R}} + k_{\mathrm{c}}}{m_{\mathrm{R}}}}\, t\right),
\end{aligned} \tag{3.9}$$

where we assume without loss of generality that $t_{\mathrm{c}}^0 = 0$ at the start of the collision, and δ_{c}^0 is the initial penetration depth, although generally $\delta_{\mathrm{c}}^0 = 0$. The initial penetration velocity can be computed as a function of the configuration-space velocity as $\dot{\delta}_{\mathrm{c}}^0 = J_{\mathrm{c}}(q_{\mathrm{c}}^0)\, \dot{q}_{\mathrm{c}}^0 + v_{\mathrm{H}}$.

3.3.3 Soft Robot Injury Severity Criterion

Following the standards established in ISO/TS 15066:2016 (Standard, 2016), we consider the maximum contact pressure, also sometimes referred to as stress (Haddadin et al., 2009), experienced during the collision as a proxy for the injury risk. Therefore, we define the SRISC for a given tuple $(q_{\mathrm{c}}^0, s_{\mathrm{c}}, n_{\mathrm{c}})$ capturing the contact geometry as

$$\mathrm{SRISC}(q_{\mathrm{c}}^0, \dot{\delta}_{\mathrm{c}}^0, \tau, s_{\mathrm{c}}, n_{\mathrm{c}}) = \max_t p_{\mathrm{c}} = \max_t \frac{f_{\mathrm{c}}(t)}{A_{\mathrm{c}}(t)} \leq \frac{\max_t f_{\mathrm{c}}(t)}{\min_t A_{\mathrm{c}}(t)} = \frac{k_{\mathrm{c}} \max_t \delta_{\mathrm{c}}(t)}{\min_t A_{\mathrm{c}}(t)}, \tag{3.10}$$

where $p_{\mathrm{c}}(t)$ is the contact pressure/stress, and A_{c} is the contact area. We illustrate the derivation and definition of the SRISC in Fig. 3.2.

The closed-form solution to the collision dynamics of (3.9) allows us to upper-bound the maximum contact force $\max_t f_{\mathrm{c}}(t)$ that is encountered during the collision as

$$\max_t f_{\mathrm{c}}(t) = k_{\mathrm{c}} \left(\frac{f_\tau - f_{\mathcal{U}}^{\mathrm{c}0}}{k_{\mathrm{R}} + k_{\mathrm{c}}} + \sqrt{\left(\delta_{\mathrm{c}}^0 - \frac{f_\tau - f_{\mathcal{U}}^{\mathrm{c}0}}{k_{\mathrm{R}} + k_{\mathrm{c}}}\right)^2 + \left(\dot{\delta}_{\mathrm{c}}^0\right)^2 \frac{m_{\mathrm{R}}}{k_{\mathrm{R}} + k_{\mathrm{c}}}}\right). \tag{3.11}$$

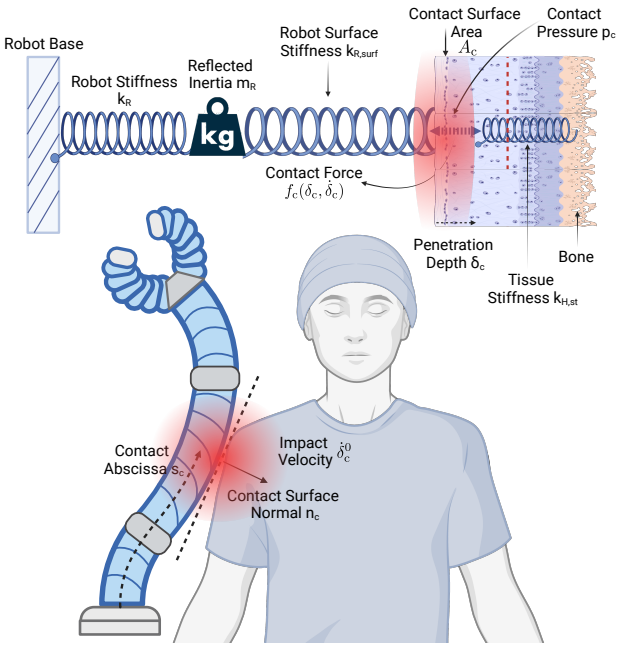


Figure 3.2: Proposed Soft Robot Injury Severity Criterion (SRISC) as a safety criterion for soft robots: The maximum contact pressure/stress $\max_t p_c(t) = \max_t \frac{f_c(t)}{A_c}$ experienced during the (potential) collision acts as a proxy for the expected injury risk (Standard, 2016), where $f_c(t)$ denotes the contact force and A_c the contact area. For computing $f_c(t)$, we derive the dynamics of the collision (i.e., the time evolution of the penetration depth $\delta_c(t)$) by projecting the dynamics of the soft robot onto a 1D Cartesian motion along the contact surface normal. In order to get a conservative estimate of the injury risk, we assume the human body part to be constrained in its motion (i.e., that the inertia of the human body part dominates the reflected inertia of the soft robot m_R).

We verify and visualize the behavior of the closed-form expression from (3.9) and its upper bound (3.11) in Fig. 3.3. It can be clearly seen how (3.11) represents a conservative upper bound on the actual contact forces the system experiences if we were to also account for the hybrid nature of the dynamics and the damping of the system.

Example: Mass-Spring Robot

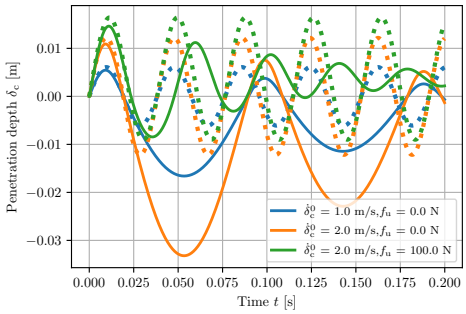
First, we consider the most *basic* soft robot - a damped mass-spring system with dynamics

$$m_R \ddot{q} + k_R (q - q^0) + d_R \dot{q} = u + f_c, \tag{3.12}$$

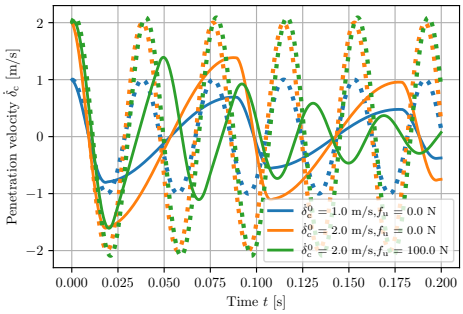
where $q \in \mathbb{R}$ and $q^0 \in \mathbb{R}$ is the equilibrium extension. Assuming $\delta_c(t) = q(t)$ (i.e., the robot is in contact with the human when $q \geq 0$) and $v_H = 0$, the SRISC is then given by

$$\text{SRISC} = \frac{k_c}{A_c} \left(\frac{u - k_R(q_c^0 - q^0)}{k_R + k_c} + \sqrt{\left(\frac{u - k_R(q_c^0 - q^0)}{k_R + k_c} \right)^2 + \left(\dot{q}_c^0 \right)^2 \frac{m_R}{k_R + k_c}} \right), \tag{3.13}$$

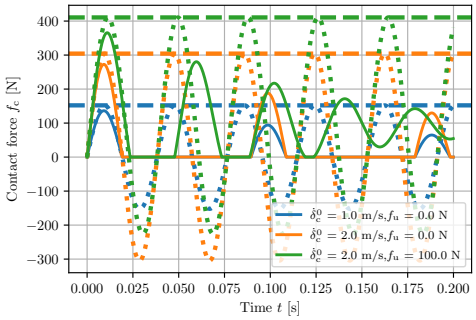
where q_c^0 is the mass-spring position at which the collision starts and $\dot{q}_c^0$ is the associated velocity.



(a) Penetration depth $\delta_c(t)$



(b) Penetration velocity $\dot{\delta}_c(t)$



(c) Contact force $f_c(t)$

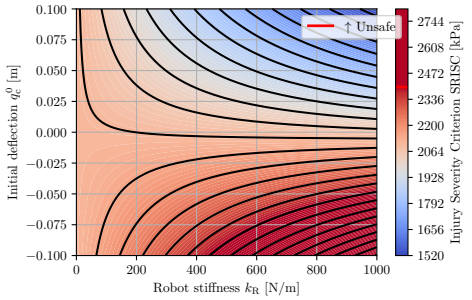
Figure 3.3: Verification of closed-form solution to (3.8): The solid lines represent numerical integrations of the hybrid dynamics $m_R\,\ddot{\delta}_c + k_R\,\delta_c + d_R\,\dot{\delta}_c + f_c(\delta_c) = f_\tau - f_U^{c0}$ with $f_c(\delta_c) = k_c\,\delta_c + d_c\,\dot{\delta}_c\,\forall\,\delta_c > 0$ and $f_c(\delta_c) = 0\,\forall\,\delta_c \leq 0$. The dotted lines represent the closed-form solution to the time evolution reported in (3.9) based on the dynamics in (3.8) that describe the behavior during the contact phase. The dashed lines represent the (conservative) maximum contact force that could be encountered during the collision as determined by the closed-form expression (3.11). As system parameters, we choose $m_R = 1\,\text{kg}$, $k_R = 2\,\text{kN/m}$, $d_R = 4\,\text{Ns/m}$, $k_c = 25\,\text{kN/m}$, which represents the spring constant of the human chest according to ISO/TS 15066:2016 (Standard, 2016), $d_c = 20\,\text{Ns/m}$, and $\delta_c^0 = 0\,\text{m}$. Please note that in all case we assume $f_U^{c0} = 0$ and $v_H = 0$.

Here, the SRISC exhibits the limits

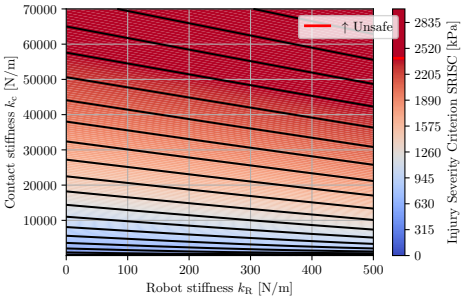
$$\lim_{k_R \to 0, u \to 0} \text{SRISC} = \frac{\sqrt{m_R\,k_c}}{A_c}\,\dot{q}_c^0, \qquad \lim_{k_R \to \infty} \text{SRISC} = \frac{k_c}{A_c}\left(-(q_c^0 - q^0) + \left|q_c^0 - q^0\right|\right). \tag{3.14}$$

If $q_c^0 \geq q^0$, which can be interpreted as the spring being in its equilibrium or stretched at the beginning of the collision, then the limit $k_R \to \infty$ simplifies to $\lim_{k_R \to \infty} \text{SRISC} = 0$.

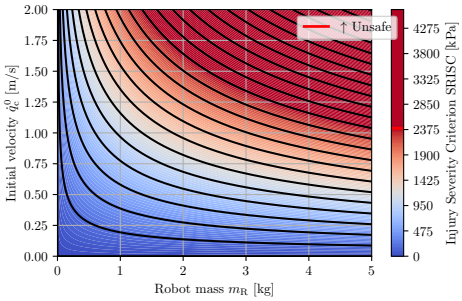
We characterize the SRISC for this mass-spring robot example in the case of a collision with a human chest and present the results in Fig. 3.4. Fig. 3.4(a) allows us to analyze what influence the robot stiffness has on the SRISC: In the limit of $k_R \to 0$ with $u = 0$, the contact pressure is solely influenced by the robot's initial velocity $\dot{q}_c^0$, its mass m_R, and the contact stiffness k_c. However, as $k_R > 0$, we can notice a bifurcation behavior: if the robot is stretched at the beginning of the collision (i.e., $q_c^0 > 0$), then the contact stress SRISC



(a) Initial deflection q_c^0 vs. Robot stiffness k_R



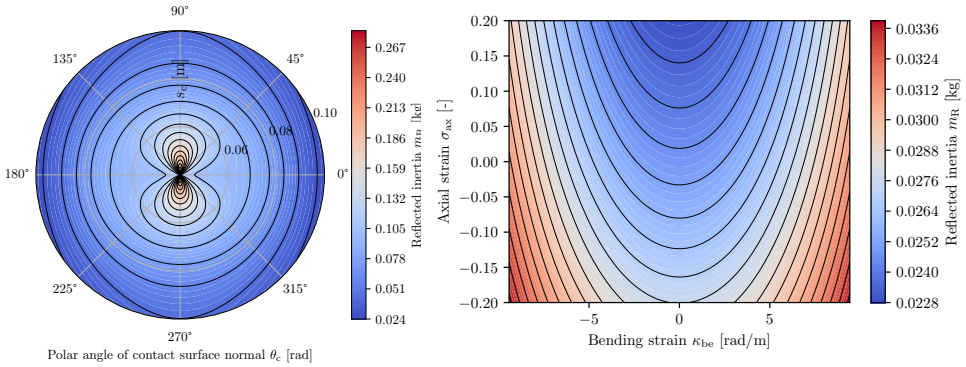
(b) Contact stiffness k_c vs. robot stiffness k_R



(c) Initial velocity $\dot{q}_c^0$ vs. robot mass m_R

Figure 3.4: Characterization of the SRISC on the mass-spring robot example. Here, we (closely) consider the collision between the mass-spring robot and the human chest. Therefore, if we conservatively assume the robot surface to be rigid, the spring constant of the contact with the chest is given as $k_c = k_H = 25\,\mathrm{kN/m}$. Furthermore, we assume a contact area of $1.5\,\mathrm{cm}^2$, an external force of $u = 0\,\mathrm{N}$, and a equilibrium position of $q^0 = 0\,\mathrm{m}$. Finally, according to ISO/TS 15066:2016 (Standard, 2016), the maximum acceptable contact pressure/stress in transient conditions is given as 2400 kPa, which serves as our threshold on the maximum acceptable SRISC. **Panel (a):** Evaluating the influence of robot stiffness k_R and the deflection of the soft robot at the beginning of the collision q_c^0 on the SRISC. We choose $m_R = 1\,\mathrm{kg}$ and $\dot{q}_c^0 = 2\,\mathrm{m/s}$. **Panel (b):** Evaluating the influence of contact stiffness k_c and the robot stiffness k_R on the SRISC. We choose $m_R = 1\,\mathrm{kg}$, $q_c^0 = -0.1\,\mathrm{m}$, and $\dot{q}_c^0 = -1.5\,\mathrm{m/s}$. **Panel (c):** Evaluating the influence of robot mass m_R and the robot velocity at the beginning of the collision $\dot{q}_c^0$ on the SRISC. We choose $k_R = 1\,\mathrm{kN/m}$ and $q_c^0 = 0\,\mathrm{m}$.

is decreased as k_R is increased. Oppositely, if the robot is compressed at the beginning of the collision (i.e., $q_c^0 < 0$), then the contact stress SRISC is increased as k_R is increased. Fig. 3.4(b) shows, as expected, that increasing the contact stiffness leads to a higher peak contact pressure. Analog to the known results in the realm of rigid robotics (Haddadin, 2013; Haddadin et al., 2009), it can be easily seen in Fig. 3.4(c) how higher velocities and higher robot masses/inertia lead to higher injury severities and safety risks. This analysis allows us to draw two important takeaways: (1) it is crucial to consider the maximum deformation of the soft robot we would expect to occur during operation to assess its safety, (2) as the robot's stiffness is increased, the *worst-case* injury severity is also increased (Abidi and Cianchetti, 2017) - confirming the intuition of how soft robots with their material softness and lower inertia exhibit a more compliant behavior than traditional rigid robots.



(a) Contact backbone coordinate s_c vs. polar angle θ_c

(b) Bending strain κ_{be} vs. axial strain σ_{ax}

Figure 3.5: Characterization of the reflected inertia, also called effective mass (Haddadin et al., 2009; Kirschner et al., 2021b), on the example of a planar CS soft robot. **Left:** Variation of the collision backbone coordinate s_c against the polar angle of contact θ_c, where $\theta_c = 0$ corresponds to a perpendicular collision with the backbone and $n_c = (1, 0)$ and $\theta_c = \frac{\pi}{2}$ relates to a parallel collision with the robot backbone and $n_c = (0, 1)$. We assume here a soft robot in its equilibrium configuration (i.e., $q = 0_3$). **Right:** Variation of the bending strain κ_{be} and the axial strain σ_{ax} for a perpendicular collision ($n_c = (0, 1)$) at the distal end of the robot ($s_c = 0.1\,\text{m}$).

Example: Planar Piecewise Constant Strain Robot

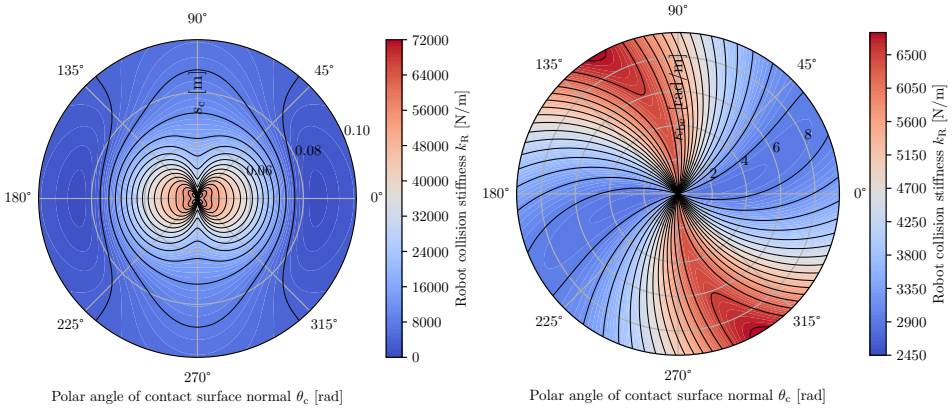
We now consider a planar N segment PCS robot, as introduced in Chapter 2. Here, the configuration of the ith segment is characterized by its spatially constant strain $\xi_i = \begin{bmatrix} \kappa_{be,i} & \sigma_{sh,i} & \sigma_{ax,i} \end{bmatrix}^\top$, where $\kappa_{be,i}$ is the bending, $\sigma_{sh,i}$ the shear, and $\sigma_{ax,i}$ the axial/elongation strain (Renda et al., 2018). The collision dynamics are then given as

$$\Lambda_c(q)\,\ddot{\delta}_c + \eta_c(q,\dot{q})\,\dot{\delta}_c + J_{c,M}^{+\top}(q)(G(q) + S\,q + D\dot{q}) = J_{c,M}^{+\top}(q)\,A_\tau(q)\,\tau - k_c\,\delta_c - d_c\,\dot{\delta}_c, \qquad (3.15)$$

where $S \in \mathbb{R}^{3N\times 3N}$ is the linear stiffness of the robot, $G(q) \in \mathbb{R}^{3N}$ captures the gravitational forces, and $\tau \in \mathbb{R}^m$ represents the actuator forces. We build the JAX (Bradbury et al., 2018) implementation of these dynamics on the *JSRM*[4] package (Stölzle et al., 2024b).

As we have seen in the previous example of the mass-spring robot, two of the variables that have the largest impact on the SRISC are the reflected inertia m_R, as defined in (3.6), and the local soft robot stiffness in the collision direction k_R, as defined in (3.7). Therefore, we present in Figures 3.5 & 3.6 a characterization of the reflected inertia m_R and the local robot stiffness k_R, respectively. In both cases, we consider a planar CS segment (i.e., $N = 1$) that has a length of 0.1 m, a radius of 0.02 m, a material density of $1070\,\text{kg/m}^3$, an elastic modulus of $E = 0.5\,\text{MPa}$ and a shear modulus of $G = 0.2\,\text{MPa}$. The results show that the reflected inertia is highest at the proximal end of the robot/segment (i.e., $s_c \to 0$) and for collisions that are parallel to the backbone, which, for our definition of our coordinate system with the soft robot in its straight configuration aligned with the y-axis, means that $\theta_c = \frac{\pi}{2} + \pi\,n$ and $n_c = \begin{bmatrix} 0 & \pm 1 \end{bmatrix}^\top$. Furthermore, when considering perpendicular collisions with the backbone, the inertia increases with the bending strain and a compressed backbone (i.e., an increase in mass density). Concerning the robot collision stiffness, we find that

[4] `https://github.com/tud-phi/jax-soft-robot-modelling`

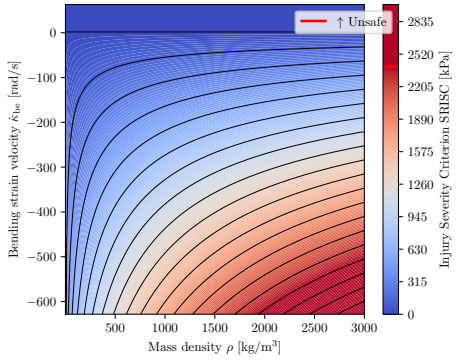


(a) Contact backbone coordinate s_c vs. polar angle θ_c (b) Bending strain κ_{be} vs. contact polar angle θ_c

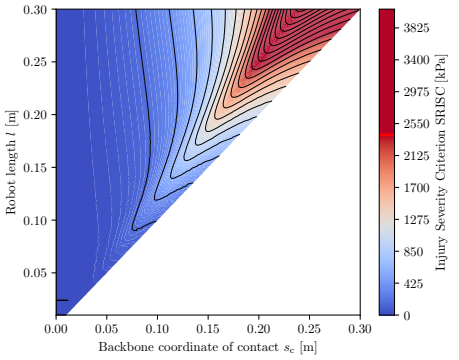
Figure 3.6: Characterization of the local robot collision stiffness k_R vs. polar angle θ_c on the example of a planar CS soft robot. **Left:** Variation of the collision backbone coordinate s_c against the polar angle of contact θ_c, where $\theta_c = 0$ corresponds to a perpendicular collision with the backbone and $n_c = (1,0)$ and $\theta_c = \frac{\pi}{2}$ relates to a parallel collision with the robot backbone and $n_c = (0,1)$. We assume here a soft robot in its equilibrium configuration (i.e., $q = 0_3$). **Right:** Variation of the bending strain κ_{be} against the polar angle of contact θ_c at the distal end of the robot ($s_c = 0.1\,\text{m}$).

it is highest at the proximal end of the soft robot. At the distal end, we identify larger stiffnesses for parallel collisions, although the exact characteristics will depend on the choice of backbone radius/second moment of area. With increased bending strain, the polar angle with the highest stiffness will also change, as seen in Fig. 3.6(b).

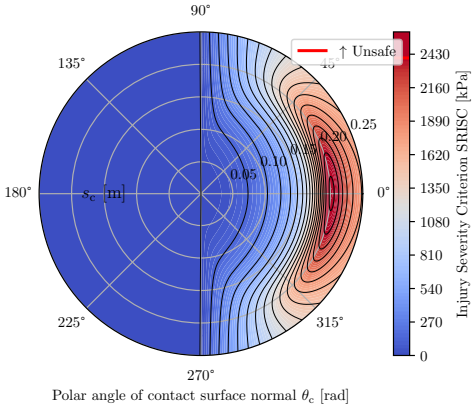
In Fig. 3.7, we characterize the behavior of the SRISC on the case of a planar CS soft robot with a nominal robot length $L = 0.25\,\text{m}$, a backbone radius of $R = 0.02\,\text{m}$, a nominal mass density of $\rho = 1070\,\text{kg/m}^3$, an elastic modulus of $E = 0.5\,\text{MPa}$, a shear modulus of $G = 0.2\,\text{MPa}$, a contact area of $A_c = 1.5\,\text{cm}^2$, the spring constant of the human chest $k_{H,st} = 25\,\text{kN/m}$, and a soft robot surface material stiffness of $k_{R,surf} = 7.5\,\text{kN/m}$. In all cases, we set the external forcing to $f_\tau = 0_3$, consider the human to be stationary and constrained with $v_H = 0\,\text{m/s}$, and assume a maximum threshold of 2400 kPa on the SRISC, which mirrors the maximum transient contact stress that is acceptable for collisions with the chest sternum according to ISO/TS 15066:2016 (Standard, 2016). In Fig.3.7(a), we see that, as expected, the maximum contact stress—and therefore the SRISC—rises with increased mass density and higher bending velocity in the collision direction. The results in Fig. 3.7(b) are particularly intriguing: while the SRISC increases with longer segment lengths—as longer soft robots exhibit higher Cartesian velocities for the same strain-space velocities—the maximum contact stress is typically not located at the very tip of the distal end but rather around 75 % of the robot's length. We hypothesize that this occurs because certain factors, such as stiffness and reflected inertia, peak at the proximal end, whereas others—especially Cartesian velocity—are maximized at the distal end, leading to an overall maximum SRISC somewhere between the two. Similarly, Fig. 3.7(c) displays analogous behavior, with the polar angle corresponding to the maximum contact stress varying based



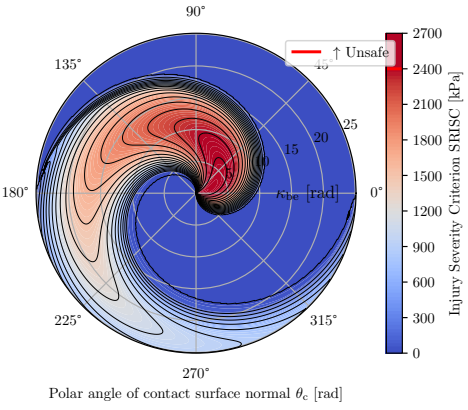
(a) Bending strain velocity $\dot{\kappa}_{be}$ vs. mass density ρ



(b) Robot length L vs. backbone coordinate s_c



(c) Contact backbone coordinate s_c vs. polar angle θ_c



(d) Bending strain κ_{be} vs. polar angle θ_c

Figure 3.7: Characterization of the SRISC on a planar CS robot. **Panel (a):** Variation of the robot's mass density ρ and the bending strain velocity $\dot{\kappa}_{be}$. We assume a perpendicular collision ($n_c = (1,0)$) at the tip of the segment ($s_c = 0.25\,\text{m}$). **Panel (b):** Variation of the total robot length L and the backbone coordinate s_c where the collision occurs. We assume a perpendicular collision ($n_c = (1,0)$) with a soft robot in its equilibrium configuration (i.e., $q = 0_3$). **Panel (c):** Variation of the backbone coordinate s_c against the polar angle θ_c of the contact surface where the collision occurs. Here, $\theta_c = 0$ corresponds to a surface normal of $n_c = (1,0)$ and $\theta_c = \frac{\pi}{2}$ to $n_c = (0,1)$. The soft robot is assumed to be in its equilibrium configuration (i.e., $q = 0_3$). **Panel (d):** Variation of the bending strain κ_{be} against the polar angle θ_c of the contact surface where the collision occurs. We assume a collision with the distal end of the soft robot (i.e., $s_c = 0.25\,\text{m}$).

on the velocities and actuation applied to the soft robot. Finally, Fig. 3.7(d) shows that the maximum SRISC occurs at a straight backbone configuration, with the maximum SRISC decreasing as bending strain increases.

Example: Integrating a Control Policy

An important question when analyzing the safety of a closed-loop soft robotic system is what influence the control policy has on the SRISC. First, we consider a case where the

behavior of the control policy $\tau(q, \dot{q})$ cannot be bounded or even inspected, such as it would be the case for controllers that contain integral terms or for many RL-based control policies. In this case, access to actuation bounds $[\tau_{\min}, \tau_{\max}]$ provides us with the injury risk in the *worst case scenario*.

Next, we consider the example of a *PD+Feedforward*-like control structure that is relevant for many control policies that involve feedforward and/or feedback terms. Specifically, we consider a fully-actuated setting (i.e., $n = m$) with an identity actuation matrix $A_\tau(q) = \mathbb{I}^n$. Then, a regulator $\tau(q, \dot{q}) = \partial_q \mathcal{U}(q^{\mathrm{d}}) + K_{\mathrm{p}}(q^{\mathrm{d}} - q) - K_{\mathrm{d}}\,\dot{q}$ drives the system towards the setpoint q^{d} (Della Santina et al., 2023) and establishes the closed-loop dynamics

$$M(q)\,\ddot{q} + C(q, \dot{q})\,\dot{q} + \partial_q \mathcal{U}(q) + K_{\mathrm{p}}\,q + (D + K_{\mathrm{d}})\,\dot{q} = \partial_q \mathcal{U}(q^{\mathrm{d}}) + K_{\mathrm{p}}\,q^{\mathrm{d}} + \tau_{\mathrm{c}}, \tag{3.16}$$

where $K_{\mathrm{p}}, K_{\mathrm{d}} \in \mathbb{R}^{n \times n}$ are the proportional and derivative feedback gains, respectively. When re-formulating the simplified collision dynamics of (3.8),

$$m_{\mathrm{R}}\,\ddot{\delta}_{\mathrm{c}} + (k_{\mathrm{R}} + k_{\mathrm{c}})\,\delta_{\mathrm{c}} = J^{+}_{\mathrm{c,M}}(q^0_{\mathrm{c}})\,(\partial_q \mathcal{U}(q^{\mathrm{d}}) + K_{\mathrm{p}}\,q^{\mathrm{d}} - \partial_q \mathcal{U}(q^0_{\mathrm{c}})). \tag{3.17}$$

We notice that the feedforward control term acts through a constant force on the oscillatory system. The proportional feedback term increases the local stiffness of the robot: $k_{\mathrm{R}} = \frac{\partial}{\partial q} J^{+\top}_{\mathrm{c,M}}(q)\,\left(\partial_q \mathcal{U}(q) + K_{\mathrm{p}}\,q\right)\Big|_{q=q^0_{\mathrm{c}}} J^{+}_{\mathrm{c,M}}(q^0_{\mathrm{c}})$. This analysis agrees with similar results known in literature (Della Santina et al., 2017).

3.3.4 Soft Robot Design Hazardousness Criterion

As presented in Fig. 3.1, one application of a soft robotic safety metric would be to answer the question "*How safe is this proposed soft robot design?*". The SRISC on its own is not sufficient to answer this question as it relies on a knowledge of the soft robot state at the beginning of the collision $(q^0_{\mathrm{c}}, \dot{q}^0_{\mathrm{c}})$, and the contact geometry $(s_{\mathrm{c}}, n_{\mathrm{c}})$. Therefore, we define the SRDHC as the maximum injury severity that can be imposed by the soft robot over all feasible soft robot states, all possible contact geometries, and actuation sequences (Wassink and Stramigioli, 2007)

$$\begin{aligned} \mathrm{SRDHC} &= \max_{q \in \mathcal{Q}} \max_{\dot{\delta}^0_{\mathrm{c}} \in [0, \dot{\delta}^{\max}_{\mathrm{c}}]} \max_{\tau \in [\tau_{\min}, \tau_{\max}]} \max_{s_{\mathrm{c}} \in (0, L]} \max_{n_{\mathrm{c}} \in S^3} \mathrm{SRISC}(q^0_{\mathrm{c}}, \dot{\delta}^0_{\mathrm{c}}, \tau, s_{\mathrm{c}}, n_{\mathrm{c}}), \\ &\leq \max_{q \in \mathcal{Q}} \max_{s_{\mathrm{c}} \in (0, L]} \max_{n_{\mathrm{c}} \in S^3} \mathrm{SRISC}(q^0_{\mathrm{c}}, \dot{\delta}^{\max}_{\mathrm{c}}, \|\tau_{\max}\|_2, s_{\mathrm{c}}, n_{\mathrm{c}}), \end{aligned} \tag{3.18}$$

where $\mathcal{Q}$ is the set of feasible soft robot configurations. To make this optimization more tractable, we can leverage the stated upper bound with $\dot{\delta}^{\max}_{\mathrm{c}} = \|J_{\mathrm{c}}\|_2\,\|\dot{q}_{\max}\|_2 + v_{\mathrm{H}}$ and $f^{\max}_{\tau} = \|A_{\tau,\mathrm{c}}\|_2\,\|\max(|\tau_{\min}|, |\tau_{\max}|)\|_2$. We note that the maximum value of $\dot{q}_{\max}$ that the robot can achieve autonomously is, in practice, often given by certain actuator characteristics (e.g., maximum servo velocity for tendon-driven actuation).

We conducted a study to assess how discretization affects the estimated safety, with a particular focus on the SRDHC. The initial results shown in Fig. 3.8 for varying the number of PCS segments while keeping the robot length and all other system parameters constant indicate that approximating the soft robot with one or only a few segments leads to a conservative safety estimate—that is, an overestimation of the injury severity. This occurs because actual soft robots can deform in an infinite-dimensional space, whereas the planar

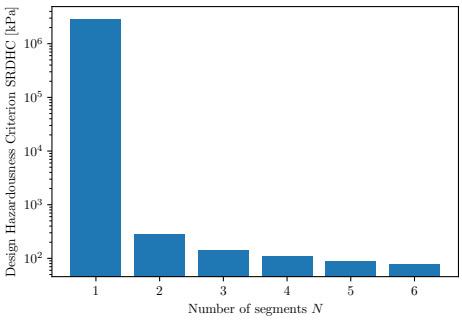


Figure 3.8: Analysis of the impact of varying the discretization of a planar PCS robot on SRDHC while keeping the total soft robot length constant at 1 m.

PCS model limits the available deformation modes and often increases the stiffness at the point of contact. By refining the discretization of the underlying kinematic model, we can capture more deformation modes and thereby reduce the overestimation of injury severity.

3.4 Conclusion

In this chapter, we first underscore the absence of a safety metric for soft robots—a gap that hinders the community from quantifying their safety benefits compared to rigid robots. This shortfall often leads to designs that perform poorly in real-world tasks because they tend to be overly soft and lack sufficient payload capacity. We then explored how a quantitative safety metric could be applied in both design and control contexts, ensuring safety without forcing the robot to be too soft or its motion overly cautious. Finally, we defined the criteria that such a safety metric must satisfy.

Next, we derived a quantitative safety metric for soft robots by considering the maximum contact pressure during a collision, analogous to the existing ISO norms for cobots (Standard, 2016), as our safety criterion. For the derivation, we relied on established soft robot dynamic models, such as PCS (Renda et al., 2018)[5], and assumed a constrained human model to represent the *worst case* scenario (Haddadin et al., 2009). The proposed safety metric is offered in two forms: the SRISC, which can be computed in closed-form and estimates the maximum contact pressure during a soft robot–human collision given a specific initial condition—comprising the robot's configuration, velocity, contact geometry, and a predetermined (e.g., worst-case) actuation—and is particularly well-suited for *safety-aware control* applications. In contrast, when designing soft robots, it is necessary to guarantee safety across an operating envelope that encompasses infinitely many initial conditions. For this purpose, we developed the SRDHC, which determines the maximum SRISC for a given set of operating conditions (e.g., maximum deformation, actuator bounds, configuration-space velocities), especially considering all potential contact geometries.

Notably, the proposed safety metric fulfills all mandatory requirements outlined in Sec. 3.2.3. For instance, it takes into account collisions along the entire soft robot body, and

[5]Please refer to Chapter 2 for a discussion of alternative soft robot models that could serve as the basis for the safety metric.

the assumptions and simplifications made during its derivation result in a conservative estimate of true safety. Additionally, the metric is computationally tractable because (a) it bypasses the need to simulate collision dynamics over time by providing the maximum contact pressure in closed form, and (b) portions of the computation can be vectorized or parallelized. Moreover, the metric is differentiable—even analytically—since our implementation in JAX (Bradbury et al., 2018) offers gradients through autodifferentiation.

Notably, the characterization of the proposed safety metric demonstrates that the safety of soft robots is determined not only by their material softness but also by factors such as mass density, actuation, control, and operating conditions (e.g., velocity, maximum deformation). In particular, increased feedback gains raise the stiffness of the closed-loop system (Della Santina et al., 2017, 2023), thereby reducing safety even for soft robots. Moreover, contrary to common intuition, the maximum injury severity does not occur at either the proximal or distal end, but rather somewhere in between. Finally, preliminary research suggests that the numerous degrees of freedom (DOFs) in soft robots may play a significant role in distributing contact energy and reducing the maximum local contact pressure.

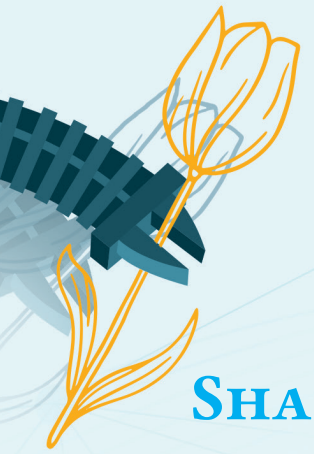
Looking ahead, two key aspects emerge for future work: (1) the experimental validation of the safety criterion—i.e., quantifying the discrepancy between predicted and actual safety—and (2) the establishment of appropriate thresholds (Behrens et al., 2022; Standard, 2016) for the safety criterion (in terms of acceptable injury risk) to ensure safety in human-centric environments. Drawing inspiration from existing literature on rigid robots (Behrens et al., 2022; Muttray et al., 2014a; Yamada et al., 1997), empirical biomechanical studies could play a crucial role in this process. It is also essential to recognize that safety requirements may vary considerably depending on the specific robotic application and task. For example, a soft robot designed as a children's toy must comply with much stricter safety standards than one intended for industrial refueling. Understanding these differences is key to achieving safe and effective performance.

Furthermore, the safety metric itself can be extended and refined. By developing specialized metrics tailored to different soft robot modeling techniques—such as FEM- or GVS-based approaches—more accurate risk assessments can be achieved for specific designs. Extending these metrics to closed-chain robots, locomotors, and wearable systems will further strengthen safety evaluation frameworks across a diverse range of robotic applications.

Finally, with regard to practical applications, it is vital to develop design and control strategies that take full advantage of the proposed safety metric. In the context of *safety-aware design*, a crucial next step is to identify which design parameters most significantly impact safety for various actuation types, such as tendon-driven versus pneumatically actuated soft robots. Mapping these dependencies will provide designers with actionable insights to enhance safety. Additionally, as outlined in Appendix A, we propose a holistic co-design strategy for optimizing the safety of soft robots (Navez, 2024; Spielberg et al., 2019; Wang et al., 2024; Zardini, 2023), which will enable the explicit formalization and quantification of the trade-off between safety and performance in soft robotics. Additionally, *safety-aware control* can be implemented, for example, using MPC or CBFs (Ames et al., 2016; Ferraguti et al., 2020), enabling us to maximize performance, such as the motion speed (Haggerty et al., 2023), while ensuring safety in all scenarios.

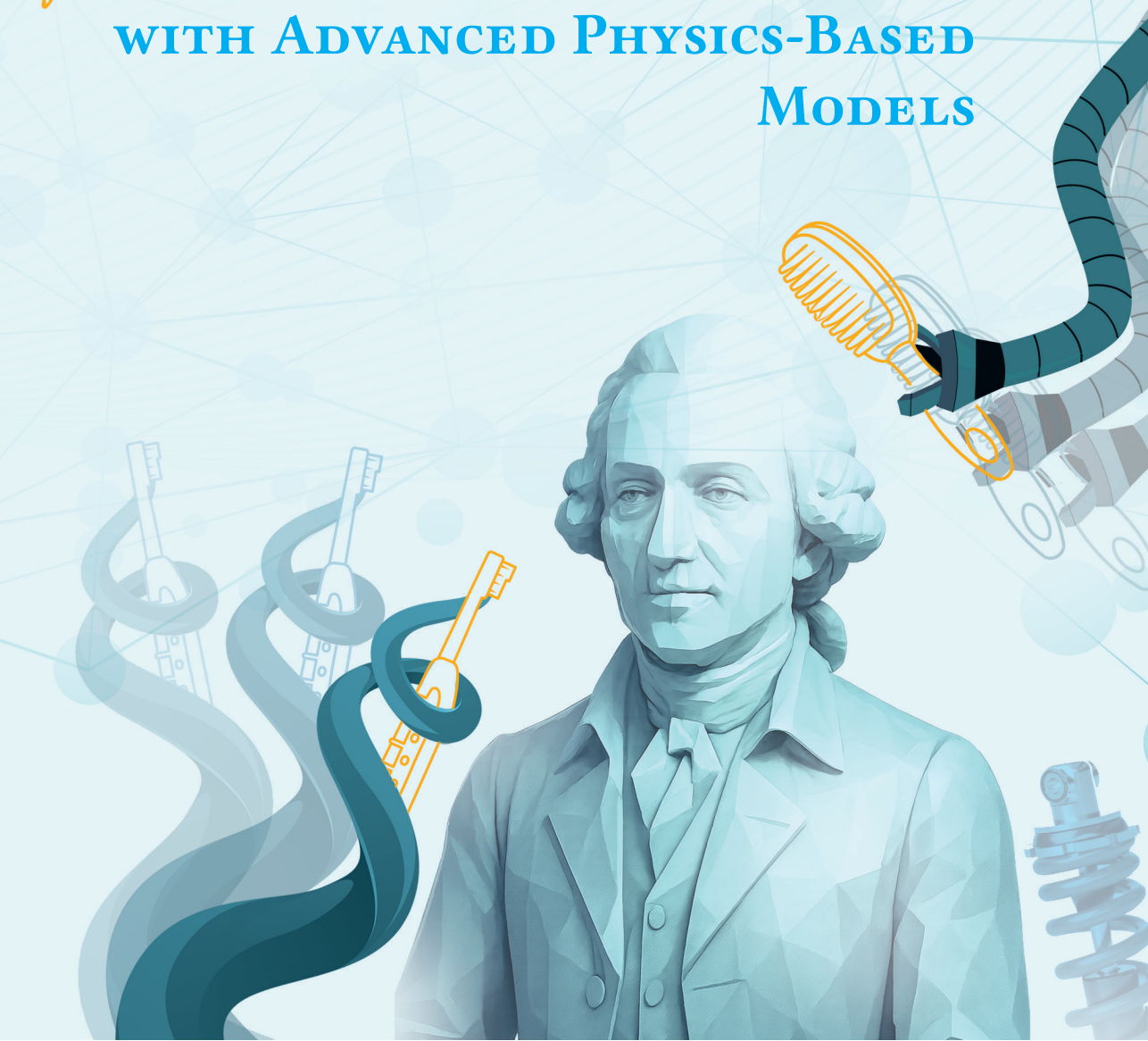
Afterword

In this chapter, we introduced a quantitative safety metric for continuum soft robots and discussed potential applications—such as safety-aware design and control—that could benefit from it. Specifically, it outlines a pathway for determining, ensuring, and certifying soft robot safety in the future. Additionally, characterizing the metric revealed that factors like material softness, mass density, and operating conditions significantly affect soft robot safety. Notably, the control policy—often overlooked in the soft robotics community—plays a critical role; our analysis confirms that, similar to findings in collaborative robotics, higher proportional feedback gains tend to reduce safety. Consequently, since ensuring safety is an important goal of this thesis, it is essential to design control strategies that maintain compliance (for instance, by eliminating integral terms and reducing feedback gains). This insight motivates both (a) the adoption of compliant controllers for soft robots, such as the impedance controller presented in Chapter 6, and (b) the development of effective feedforward terms based on accurate system dynamics modeling—a topic we explore from first principles in Chapters 5, 6, and 8, as well as through learning-based approaches in Chapters 10–11.



I 3

Shape Sensing and Control with Advanced Physics-Based Models



4

Shape Sensing with Cameras: An Investigation on Kinematics-Aware SLAM

Foreword. Before we can develop advanced soft robot controllers, we require access to state information. Specifically, we need to know the configuration of the soft robot, which, as introduced in Chapter 2, is usually defined for soft robots as the parametrized shape of the backbone. In this chapter, we demonstrate how we can augment in a plug-and-play fashion existing state-of-the-art SLAM algorithms with kinematic knowledge to achieve shape sensing for soft robots.

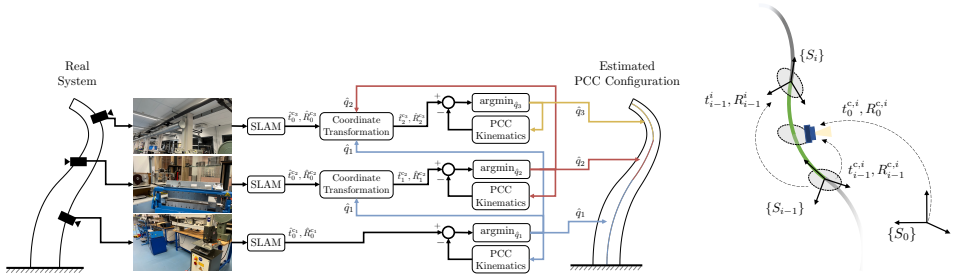
Abstract. *One way to achieve proprioception of the soft robot's shape while not substantially modifying their bodies' softness is to develop innovative and completely deformable sensors. However, these solutions tend to be less reliable than classic sensors for rigid robots. As an alternative, we consider here the use of monocular cameras. By admitting a small rigid component in our design, we can leverage well-established solutions from mobile robotics. We propose a shape-sensing strategy that combines an SLAM algorithm with nonlinear optimization based on the robot's kinematic model. We prove the method's effectiveness in simulation and with experiments of a single-segment continuous soft robot with a camera mounted to the tip. We achieve mean relative translational errors below 9% in simulations and experiments alike, and as low as 0.5% on average for some simulation conditions.*

This chapter is partly based on *E. Rosi*, **M. Stölzle***, F. Solari, and C. Della Santina (2022, April). Sensing Soft Robots' Shape with Cameras: an Investigation on Kinematics-Aware SLAM. In 2022 IEEE 5th International Conference on Soft Robotics (RoboSoft) (pp. 795-801). IEEE.* (Rosi et al., 2022).

C.D.S. conceived and led the project. E.R. implemented the methodology, collected the simulation results, and executed the lab experiments with a soft segment. M.S. contributed ideas to the methodology, helped debug the implementation of the approach in code, contributed to the experimental setup (e.g., motion capture system, fabrication of the soft segment), and wrote the paper. C.D.S., E.R., and M.S. revised the manuscript. C.D.S., F.S., and M.S. supervised the research project. C.D.S. provided funding.

4.1 Introduction

A major challenge with shape perception in soft robots is that sensing strategies must not compromise the intrinsic softness of these systems (Polygerinos et al., 2017; Wang et al., 2018a). To this end, researchers have proposed several entirely deformable sensors over the years, including capacitive (Scimeca et al., 2019) and optical sensors (Li et al., 2021), liquid metal (Wall et al., 2017). These solutions are quite attractive since they minimally corrupt the physical softness of the robot. However, they usually require complex learning strategies to be used since their behavior is hard to model (Thuruthel et al., 2019; Truby et al., 2020). An alternative is to relax the constraint of complete deformability and allow for small rigid components. This strategy enables rethinking the use of existing sensing technologies in this radically new context. Examples are hall sensors (Guo et al., 2019), Inertial Measurement Units (IMUs) (Hughes et al., 2020), and microphones (Zöller et al., 2018). The information gathered from inward-facing cameras looking at features in soft chambers' inner walls has proven sufficient to estimate the configuration (She et al., 2020a; Werner et al., 2020), and characterize contacts with the environment (Lin et al., 2020; Ward-Cherrier et al., 2018). However, all these strategies need machine learning to transform the image information into the desired physical quantity. Alternatively, cameras mounted outwards on the robot's tip have been used to execute visual servoing (Homberg et al., 2019; Wang et al., 2013). To the best of the Authors' knowledge, the only two works dealing with continuum (non-soft) robots are (Weber et al., 2012) and (Cheng et al., 2020). The first uses Bundle Adjustment (BA) to integrate the output of multiple cameras embedded in a single segment. The second uses hand-tuned features to estimate the robot configuration within a novel kinematic model. Thus, no general strategy exists in the literature to estimate the whole state of a soft segment from a single monocular camera. SLAM is one of the most effective and largely used strategies for vision-based localization for mobile robots and autonomous vehicles (Fuentes-Pacheco et al., 2015; Mur-Artal and Tardós, 2017). In this chapter, we investigate using monocular SLAM to estimate the location of selected points across the soft robot. We then propose a mechanism for simultaneously refining the estimation and reconstructing the complete shape of the robot. We do that by retracting the output of the SLAM to the manifold of camera configurations admitted by the kinematic model of the soft robot. We formulate this action as a nonlinear optimization problem. Fig. 4.1(a) summarizes the proposed architecture. We test the strategy with simulations and experiments, achieving mean relative translational errors between 0.4 % and 9 % in the former and from 5 % to 9 % in the latter.



(a) Overview (b) Kinematics

Figure 4.1: Panel (a) shows a pictorial representation of the proposed perception strategy. Cameras are attached to a soft continuum robot. We propose to use ORB-SLAM (Mur-Artal and Tardós, 2017) to gather a pose estimate for each camera. The results are iteratively combined to extract local transformation and refined by projecting the resulting postures onto the manifold of configurations attainable with the PCC kinematics. The result is an estimation of the full shape within the selected kinematic description $\hat{q}$. Panel (b) reports the main quantities of the kinematic model of one segment.

4.2 Proposed Methodology: Shape Estimation with Kinematics-Aware SLAM

In this chapter, we propose to use SLAM for the proprioception of continuum soft robots. Common kinematic parametrizations for soft robots, such as PCC (Webster III and Jones, 2010) or PCS (Renda et al., 2018), model the soft arm to consist of multiple segments with independent kinematic state variables. In the following, we assume that the robot's model consists of n_S segments and that a monocular camera is attached to each segment. We rely on a PCC kinematic formulation. However, the proposed results can be directly generalized to the PCS case. Our goal is to reconstruct the full shape (i.e., a configuration q_i for each segment) of the soft robot from the stream of images recorded by the cameras.

Fig. 4.1(a) shows an overview of the proposed architecture. We use SLAM algorithms such as monocular ORB-SLAM (Mur-Artal and Tardós, 2017) to estimate the pose of cameras attached to the soft robotic arm. The pose estimation consists of 3D translation and rotations describing the relative camera movement from initial calibration to the current state ($\hat{t}_0^{c_i}, \hat{R}_0^{c_i}$ in the figure). Then, the kinematic model is simultaneously used to refine the outputs of the SLAM and transform them to the desired estimation of the configurations ($\hat{q}_1, \hat{q}_3, \hat{q}_3$ in the figure). Starting from the base segment and progressively iterating up until reaching the tip of the robot, we express translation and rotations in local coordinates, and we optimize the estimated configurations segment-by-segment starting at the proximal end by projecting the pose estimate into the 3D PCC kinematics.

In the following subsections, we provide more details on the various components of this architecture.

4.2.1 Background: Monocular ORB-SLAM

We choose monocular ORB-SLAM (Mur-Artal and Tardós, 2017) for pose estimation of the camera locations. Although this technological solution has never been applied to soft

robots, the algorithm itself is well established, and its applications to mobile robotics are widespread. As such, we briefly describe here only the major steps of the algorithm.

1. **Map Initialization**: ORB-SLAM initializes a map of 3D points based on two video frames. The 3D points and relative camera pose are computed using triangulation of 2D ORB feature correspondences.

2. **Tracking**: Once the map is initialized, the camera pose is estimated for each new frame by matching features in the current frame to features in the last keyframe. The estimated camera pose is refined by tracking the local map.

3. **Local Mapping**: If the current frame is identified as a keyframe, it is used to create new 3D map points. At this stage, BA is used to minimize re-projection errors by adjusting the camera pose and 3D points.

4. **Loop Closure**: Loops are detected for each keyframe by comparing it against all previous ones. This information is used to optimize the poses.

An important aspect of SLAM algorithms are keyframes, which are a subset of video frames that contain cues for localization and tracking. Two consecutive keyframes usually involve sufficient visual change.

4.2.2 Projection into PCC-Kinematics

Once the SLAM algorithm provides us with the estimated camera poses, we want to interpret and correct them such as that they are coherent with the PCC kinematic model. In this chapter, we consider the *Delta* parametrization (Della Santina et al., 2020a) of the PCC kinematics, but the formulation could also be easily adapted to other kinematic parametrizations such as PCS (Renda et al., 2018).

We show a pictorial representation of this kinematic model in Fig. 4.1(b). Each segment of original length $L_{0,i}$ is described with three configuration variables $q_i \in \mathbb{R}^3 = \begin{pmatrix}\Delta_{x,i} & \Delta_{y,i} & \delta L_i\end{pmatrix}^\top$, where δL_i is the segment's extension, and $\Delta_{x,i}$ and $\Delta_{y,i}$ are the differences of the arc lengths of the segment at a radial distance of d_i from the center-line along both cardinal directions of the base (Della Santina et al., 2020a). The complete robot's configuration is $q \in \mathbb{R}^{3n_\mathrm{s}}$. The coordinate transformation for segment i from the base frame $\{S_{i-1}\}$ into the tip frame $\{S_i\}$ as a function of the configuration q_i is given by (Della Santina et al., 2020a)

$$
\begin{aligned}
R^i_{i-1} &= \begin{pmatrix}
1+\frac{\Delta^2_{x,i}}{\Delta^2_i}(\mathrm{c}_i-1) & \frac{\Delta_{x,i}\Delta_{y,i}}{\Delta^2_i}(\mathrm{c}_i-1) & \frac{\Delta_{x,i}}{\Delta_i}\mathrm{s}_i \\
\frac{\Delta_{x,i}\Delta_{y,i}}{\Delta^2_i}(\mathrm{c}_i-1) & 1+\frac{\Delta^2_{y,i}}{\Delta^2_i}(\mathrm{c}_i-1) & \frac{\Delta_{y,i}}{\Delta_i}\mathrm{s}_i \\
\frac{-\Delta_{x,i}}{\Delta_i}\mathrm{s}_i & \frac{-\Delta_{y,i}}{\Delta_i}\mathrm{s}_i & \mathrm{c}_i
\end{pmatrix}, \\
t^i_{i-1} &= \frac{d_i(L_{0,i}+\delta L_i)}{\Delta^2_i}\begin{pmatrix}\Delta_{x,i}(1-\mathrm{c}_i) & \Delta_{y,i}(1-\mathrm{c}_i) & \Delta_i\mathrm{s}_i\end{pmatrix}^\top,
\end{aligned}
\tag{4.1}
$$

where we substituted $\Delta_i = \sqrt{\Delta^2_{x,i}+\Delta^2_{y,i}}$, $\mathrm{s}_i = \sin\left(\frac{\Delta_i}{d_i}\right)$, and $\mathrm{c}_i = \cos\left(\frac{\Delta_i}{d_i}\right)$ for conciseness.

We describe the coordinate frame of camera i with $\{S_{\mathrm{c}_i}\}$. It is assumed that there exists a fixed transformation $T^{\mathrm{c}_i}_{\check{\mathrm{c}},i} \in SE(3)$ from frame $\{S_{\check{\mathrm{c}},i}\}$ to the camera frame $\{S_{\mathrm{c}_i}\}$. $\{S_{\check{\mathrm{c}},i}\}$ is localized

at a distance l_{c_i} along the center-line from the base of segment i. The transformation $T_{i-1}^{\check{c},i}(q_{\check{c},i})$ from the base to the frame $\{S_{\check{c},i}\}$ can be found with (4.1) by plugging in the adjusted configuration $q_{\check{c},i}$ defined as

$$q_{\check{c},i} = \frac{l_{c_i}}{L_{0,i}} q_i, \tag{4.2}$$

and the adjusted original length l_{c_i}.

The SLAM algorithm provides a pose estimate for the translation $\hat{t}_{c,t0,i}^{c_i} \in \mathbb{R}^3$ and rotation $\hat{R}_{c,t0,i}^{c_i} \in SO(3)$ relative to the known initial reference frame of the camera $\{S_{c,t0,i}\}$. Thus, we first transform the pose estimates to the inertial frame of the robot $\{S_0\}$

$$\begin{pmatrix} \hat{t}_0^{c_i} \\ 1 \end{pmatrix} = T_0^{c,t0,i} \begin{pmatrix} \hat{t}_{c,t0,i}^{c_i} \\ 1 \end{pmatrix}, \quad \hat{R}_0^{c_i} = R_0^{c,t0,i} \hat{R}_{c,t0,i}^{c_i}, \tag{4.3}$$

where $T_0^{c,t0,i} \in SE(3)$ and $R_0^{c,t0,i} \in SO(3)$ are the known transformation and rotation matrices from the world frame to the initial frame of the ith camera, respectively.

We introduce the following notations for the PCC kinematics

$$\begin{aligned} \hat{t}_{i-1}^{c_i} &= \Pi_t(\hat{q}_i) = \hat{T}_{i-1}^{\check{c},i} \left(\frac{l_{c_i}}{L_{0,i}} \hat{q}_i \right) t_{\check{c},i}^{c_i} \\ \hat{R}_{i-1}^{c_i} &= \Pi_R(\hat{q}_i) = \hat{R}_{i-1}^{\check{c},i} \left(\frac{l_{c_i}}{L_{0,i}} \hat{q}_i \right) R_{\check{c},i}^{c_i}, \end{aligned} \tag{4.4}$$

where $\hat{t}_{i-1}^{\check{c},i}$ and $\hat{R}_{i-1}^{\check{c},i}$ describe the translation and rotation from the base of the segment to the camera frame according to the PCC kinematic model for an estimated configuration of the segment $\hat{q}_i$. $\hat{T}_{i-1}^{\check{c},i}(q_{\check{c},i})$ and $\hat{R}_{i-1}^{\check{c},i}(q_{\check{c},i})$ are based on (4.1) and a function of the adjusted configuration $q_{\check{c},i}$ referenced in (4.2).

Additionally, the pose estimates by the SLAM algorithm need to be transformed to the base frame of segment i. Thus, we introduce the following

$$\begin{aligned} \hat{t}_{i-1}^{c_i} &= \Psi_t(q_1 \dots q_{i-1}, \hat{t}_0^{c_i}) = \prod_{\tilde{i}=1}^{i-1} \left(T_{\tilde{i}-1}^{\tilde{i}} \left(\hat{q}_{\tilde{i}} \right) \right)^\top \begin{pmatrix} \hat{t}_0^{c_i} \\ 1 \end{pmatrix}, \\ \hat{R}_{i-1}^{c_i} &= \Psi_R(q_1 \dots q_{i-1}, \hat{R}_0^{c_i}) = \prod_{\tilde{i}=1}^{i-1} \left(R_{\tilde{i}-1}^{\tilde{i}} \left(\hat{q}_{\tilde{i}} \right) \right)^\top \hat{R}_0^{c_i}. \end{aligned} \tag{4.5}$$

Next, we define a cost function to optimize the pose estimate by projecting it into the PCC-kinematics

$$\min_{\hat{q}} \sum_{i=1}^{n_S} f_{t,i}(\hat{q}) + \lambda_R f_{R,i}(\hat{q}), \tag{4.6}$$

with

$$\begin{aligned} f_{t,i}(\hat{q}) &= \left\| \Pi_t(\hat{q}_i) - \Psi_t(q_1 \dots q_{i-1}, \hat{t}_0^{c_i}) \right\|_2, \\ f_{R,i}(\hat{q}) &= \left\| \Pi_R(\hat{q}_i) - \Psi_R(q_1 \dots q_{i-1}, \hat{R}_0^{c_i}) \right\|_F, \end{aligned} \tag{4.7}$$

where the Euclidean norm is used to compute the translational error between the predicted translation by the PCC kinematic model and the estimated translation by SLAM. The rotational error is weighted with $\lambda_R \in \mathbb{R}$ and computed with the Frobenius norm between the predicted rotation matrix by the PCC kinematics and the estimated orientation by SLAM represented as a rotation matrix as well.

Please note that the optimization of the configuration estimate $\hat{q}$ can be decoupled for each segment. We start by optimizing the configuration of the first segment $\hat{q}_1$ based on $\Psi_t(\hat{t}_0^{c,1})$ and $\Psi_R(\hat{R}_0^{c,1})$. Next, we optimize the configuration of the second segment $\hat{q}_2$ taking into account the already optimized configuration of segment one in $\Psi_t(\hat{q}_1, \hat{t}_0^{c,2})$ and $\Psi_R(\hat{q}_1, \hat{R}_0^{c,2})$. Subsequently, we move on to optimize the remaining segments sequentially as described by Algorithm 1. This procedure is also graphically represented by the right side of Fig. 4.1(a).

Algorithm 1 Pose estimation for soft robots through SLAM

Input: $o \in \mathbb{R}^{n_S \times h_o \times w_o}$ ▷ Observations of all cameras
Input: $R_0^{c,t0,i} \in SO(3)$, $T_0^{c,t0,i} \in SE(3)$ ▷ Transformation world to initial ith camera frame.
Input: $\lambda_R \in \mathbb{R}$ ▷ Weight of rotational error
Input: SLAM : $\mathbb{R}^{h_o \times w_o} \to \mathbb{R}^3 \times SO(3)$ ▷ SLAM algorithm providing translations and orientations based on images.
Input: $\Pi_t : \mathbb{R}^3 \to \mathbb{R}^3$ ▷ Translation component of CC forward kinematics.
Input: $\Pi_R : \mathbb{R}^3 \to SO(3)$ ▷ Rotational component of CC forward kinematics.
Input: $\Psi_t : \mathbb{R}^{3(i-1)} \times \mathbb{R}^3 \to \mathbb{R}^3$ ▷ Translation of SLAM estimates.
Input: $\Psi_R : \mathbb{R}^{3(i-1)} \times SO(3) \to SO(3)$ ▷ Rotation of SLAM estimates.
Output: $\hat{q} \in \mathbb{R}^{3n_S}$ ▷ Estimated robot configuration

$i \leftarrow 1$
while $i \leq n_S$ **do**
  $\left(\hat{t}_{c,t0,i}^{c_i}, \hat{R}_{c,t0,i}^{c_i}\right) \leftarrow \text{SLAM}(o_i)$ ▷ SLAM algorithm
  $\left(\hat{t}_0^{c_i}, \hat{R}_0^{c_i}\right) \leftarrow \left(T_0^{c,t0,i}\, \hat{t}_{c,t0,i}^{c_i}, R_0^{c,t0,i}\, \hat{R}_{c,t0,i}^{c_i}\right)$ ▷ Transformation into world frame.
  $f_{t,i}(\hat{q}) \leftarrow \left\|\Pi_t(\hat{q}_i) - \Psi_t(q_1 \dots q_{i-1}, \hat{t}_0^{c_i})\right\|_2$
  $f_{R,i}(\hat{q}) \leftarrow \left\|\Pi_R(\hat{q}_i) - \Psi_R(q_1 \dots q_{i-1}, \hat{R}_0^{c_i})\right\|_F$
  $f_{c,i}(\hat{q}) \leftarrow f_{t,i}(\hat{q}) + \lambda_R\, f_{R,i}(\hat{q})$ ▷ Cost function for $\hat{q}_i$
  $\hat{q}_i \leftarrow \text{argmin}_{\hat{q}_i} f_c(\hat{q})$
  $i \leftarrow i + 1$
end while

4.3 Simulations

We quantitatively evaluate our approach in simulation for one soft robotic segment with a camera attached to the tip of the robot. We first compute trajectories that behave according to PCC kinematics. Next, we render photo-realistic images for the camera attached to the tip of the segment for every time step using a virtual environment implemented in Blender. Subsequently, we process the synthetic camera images with the ORB-SLAM (Mur-Artal

Table 4.1: Parameters of the implemented trajectories. We list the amplitudes and the frequencies of the trajectories parametrized by $\Delta_{x,1}$ and $\Delta_{y,1}$ as specified in (4.8).

Trajectory	A_x	A_y	F_x	F_y
Trajectory 1: planar side bending	0.1	0	0.5	0
Trajectory 2: half 8-shape	0.05	0.05	1	0.5
Trajectory 3: full 8-shape	0.05	0.05	2	1

and Tardós, 2017) algorithm and project the estimated poses of the tip of the segment into the PCC kinematic model as outlined in Section 4.2. Finally, we compare the estimated poses against the ground truth and statistically evaluate the Root Mean Squared Error (RMSE) both for translational and rotational estimates. More details follow.

4.3.1 System

We consider a soft robotic segment of diameter 20 mm diameter and with varying lengths $L_{0,1}$ between 15 cm and 100 cm. As the camera is attached to the tip of the segment, we set $l_{c_1} = L_{0,1}$ and define $T_{\check{c},1}^{c,1} = \mathbb{I}^4$.

4.3.2 Trajectories and Calibration Sequence

Three different trajectories are considered for the simulated movement of the soft robotic segment and its attached virtual camera. While the first one represents a planar side bending, the second one describes an "8" shape with the tip, and the last one covers a lobe of the "8". Those trajectories were commanded in $\Delta_{x,1}$ and $\Delta_{y,1}$, with the following mathematical formulas:

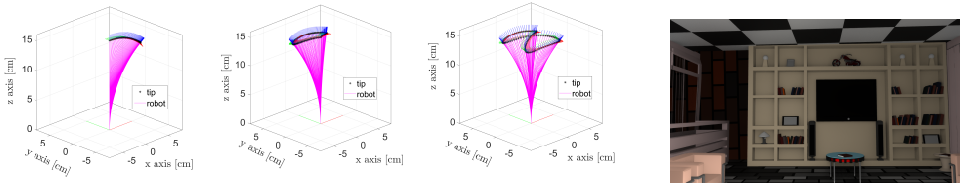
$$\Delta_{x,1} = L_{0,1} A_x \sin(2\pi f_x) \qquad \Delta_{y,1} = L_{0,1} A_y \sin(2\pi f_y), \tag{4.8}$$

with $L_{0,1}$ the unextended length of the robot, A_x and A_y amplitudes of the sinusoids and f_x and f_y frequencies of the sinusoids. The parameters f_x and f_y are defined as follows with k representing the current time index:

$$f_x = \frac{F_x(k-1)}{n_t}, \qquad f_y = \frac{F_y(k-1)}{n_t}. \tag{4.9}$$

Please note that these trajectories do not contain any segment elongation. We list the chosen amplitudes and frequencies of the trajectories in Table 4.1. Each trajectory is generated considering different robot lengths, namely 15 cm, 30 cm and 100 cm. The number of time steps n_t is chosen at 120. The commanding of $\Delta_{x,1}$ and $\Delta_{y,1}$ is such that the amplitude and frequency of the trajectory are independent of the number of frames (i.e., time steps). In Figure 4.2, we show a 3D visualization of the trajectories corresponding to a segment length of 15 cm.

In addition to the robot trajectory, a calibration sequence trajectory is designed to initialize the SLAM map. It is good practice to move the camera parallel to the scene captured. Accordingly, we decide to move the camera into the x cardinal direction of the segment base frame with the translation distance proportional to the robot length.



(a) Trajectory 1 (b) Trajectory 2 (c) Trajectory 3 (d) Virtual interior scene

Figure 4.2: 3D visualization of the three trajectories used in our simulations and experiments for a segment with 15 cm length. In magenta, we visualize the trajectory of the full robot, and in black, the tip positions. Additionally, the tip orientation (red = x-axis, green = y-axis, blue = z-axis) is displayed. The virtual interior scene used for the Blender renderings is presented in the last column.

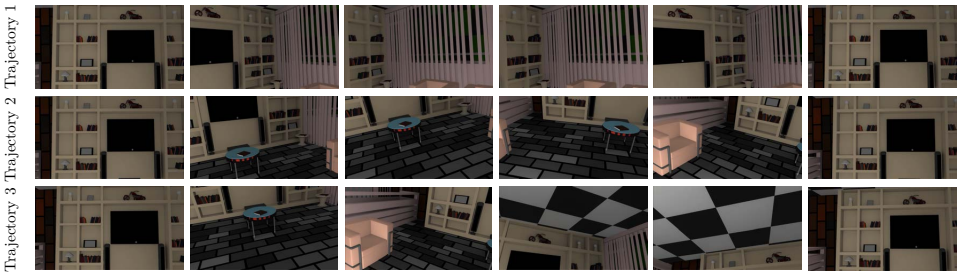


Figure 4.3: Sequence of stills showing the rendered images by the virtual camera in Blender for three different trajectories and a robot segment of length 15 cm. The trajectories are visualized in Figure 4.2.

4.3.3 Rendering of Synthetic Images

The rendering software Blender allows us, among other things, to load a 3D model of the environment, follow customized trajectories with a virtual camera, and render photo-realistic synthetic pictures of the environment from the chosen camera perspective. We use an interior scene published by Nextwave Multimedia. We report a view of the scene in Fig. 4.2(d). The virtual camera is set to be perspective with a focal length of 30 mm. For each run, we randomly initialize the trajectory at one of seven predefined launch points in the indoor environment to diversify the coverage of the environment. The x-, y-, and z-coordinates of the seven initial positions have a standard deviation of 0.5 m, 0.2 m, 0.4 m, respectively. The initial orientation represented in XYZ Euler angles varies with a standard deviation of 0.05 rad, 0.13 rad, and 1.17 rad. For each trajectory, we render 120 synthetic images along the trajectory and save them to a folder for later offline processing by the ORB-SLAM (Mur-Artal and Tardós, 2017) algorithm. Fig. 4.3 reports a few representative stills of what the robot sees during one execution of the three trajectories discussed above.

4.3.4 Implementation of ORB-SLAM

The synthetic images along the trajectory are processed offline by the ORB-SLAM (Mur-Artal and Tardós, 2017) algorithm. We rely on the official MATLAB implementation of ORB-SLAM. While we run our simulations offline to decouple any delays by the rendering

and/or SLAM pipeline, we would like to point out that other ORB-SLAM implementations, such as, for example, in C++, are able to be run in real-time at frame rates of between 10 Hz to 30 Hz (Mur-Artal and Tardós, 2017).

4.3.5 Projection into PCC kinematics

The trade-off parameter λ_{R} between the rotational and the translational error in the cost function (4.6) was manually tuned and set to $\lambda_{\mathrm{R}} = 0.4$. As the simulations do not contain any elongations of the segment, we set $\delta L_1 = 0$ in (4.1). We solve the optimization problem outlined in (4.6) using nonlinear least-squares with the Levenberg-Marquardt solver (Levenberg, 1944; Marquardt, 1963) implemented in MATLAB as *lsqnonlin*.

4.3.6 Evaluation metrics

To quantitatively evaluate the performance of our proposed approach, we introduce error metrics for both the translation and orientation estimates. We measure the translational pose prediction error with a relative RMSE e_{t}

$$e_{\mathrm{t}} = \frac{\sqrt{\sum_{t=1}^{n_{\mathrm{t}}} \left(\| \hat{t}_{0,t}^{\mathrm{c},1} - t_{0,t}^{\mathrm{c},1} \|_2 \right)^2}}{\sqrt{n_{\mathrm{t}}}\, l_{\mathrm{traj}}}, \tag{4.10}$$

where l_{traj} corresponds to the length of the trajectory and n_{t} the total number of data points along the trajectory. Similarly, we leverage the Frobenius norm for the rotational error e_{R}

$$e_{\mathrm{R}} = \sqrt{\sum_{t=1}^{n_{\mathrm{t}}} \frac{\left(\| \hat{R}_{0,t}^{\mathrm{c},1} - R_{0,t}^{\mathrm{c},1} \|_F \right)^2}{n_{\mathrm{t}}}}. \tag{4.11}$$

As torsion can often be neglected for soft robotic arms, we state the angle error for the orientation of the local z-axis of the tip of the segment for intuitive analysis of the orientation estimates. First, the unit vector of the local z-axis $\{o_1\}_0$ is computed in the base frame $\{S_0\}$

$$\{o_1\}_0 = R_{0,t}^1 \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^\top, \quad \{\hat{o}_1\}_0 = \hat{R}_{0,t}^1 \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^\top, \tag{4.12}$$

which allows us to subsequently compute the angle error between the ground-truth z-axis of the tip $\{o_1\}_0$ and the estimated z-axis $\{\hat{o}_1\}_0$

$$e_{\theta_z} = \sqrt{\sum_{t=1}^{n_{\mathrm{t}}} \frac{\left(\arccos\left(\{o_1\}_0 \cdot \{\hat{o}_1\}_0 \right) \right)^2}{n_{\mathrm{t}}}}. \tag{4.13}$$

4.3.7 Results

We evaluate our proposed method in simulation on three different robot segment lengths (15 cm, 30 cm, and 100 cm) and for the three trajectories previously described. We state statistical results such as mean, standard deviation, and lower and upper bounds over seven separate trials, each covering a different part of the indoor environment. The errors are reported both for the SLAM estimates *before* optimization and *after* projection into the PCC kinematics. While the results for the relative RMSE of translation estimates through the

Table 4.2: Relative RMSE [%] for translations as referenced in (4.10) of various trajectories and of robot segments with different lengths (15 cm, 30 cm, 100 cm). We state the error as mean ± stdev (min, max) and compute the statistics over seven trials from different initial poses.

Trajectory	**Optimization**	$L_{0,1} = 15\,\text{cm}$	$L_{0,1} = 30\,\text{cm}$	$L_{0,1} = 100\,\text{cm}$
Trajectory 1	No	$9 \pm 3\ (5, 12)$	$7 \pm 3\ (3, 13)$	$3 \pm 2\ (1, 7)$
Trajectory 1	Yes	$0.4 \pm 0.2\ (0.3, 0.8)$	$2 \pm 1\ (0, 4)$	$1.0 \pm 0.7\ (0.5, 2.3)$
Trajectory 2	No	$9 \pm 4\ (4, 17)$	$6 \pm 2\ (3, 8)$	$1.9 \pm 0.9\ (1.0, 3.0)$
Trajectory 2	Yes	$0.7 \pm 0.7\ (0.3, 1.8)$	$0.7 \pm 0.4\ (0.2, 1.2)$	$0.6 \pm 0.3\ (0.3, 0.9)$
Trajectory 3	No	$6 \pm 5\ (3, 16)$	$2.6 \pm 0.6\ (1.7, 3.3)$	$6 \pm 14\ (1, 37)$
Trajectory 3	Yes	$2 \pm 3\ (0, 9)$	$0.5 \pm 0.3\ (0.1, 0.8)$	$2 \pm 5\ (0, 15)$

Table 4.3: Rotational errors of various trajectories and for robot segments with different lengths (15 cm, 30 cm, 100 cm). We report both an absolute RMSE computed with the Frobenius norm between the rotation matrices as stated in (4.11) and an angle error [rad] for the orientation of the z-axis of the tip of the segment as defined in (4.13). We state the error as mean ± stdev and compute the statistics over seven trials from different initial poses.

Trajectory	**Optim.**	$L_{0,1} = 15\,\text{cm}$		$L_{0,1} = 30\,\text{cm}$		$L_{0,1} = 100\,\text{cm}$	
		e_R	e_{θ_z} [rad]	e_R	e_{θ_z} [rad]	e_R	e_{θ_z} [rad]
Trajectory 1	No	0.010 ± 0.004	0.005 ± 0.002	0.016 ± 0.007	0.009 ± 0.005	0.027 ± 0.017	0.02 ± 0.01
Trajectory 1	Yes	0.007 ± 0.002	0.005 ± 0.002	0.012 ± 0.007	0.008 ± 0.005	0.027 ± 0.019	0.02 ± 0.01
Trajectory 2	No	0.02 ± 0.02	0.01 ± 0.02	0.011 ± 0.006	0.006 ± 0.002	0.015 ± 0.008	0.009 ± 0.005
Trajectory 2	Yes	0.02 ± 0.02	0.01 ± 0.02	0.016 ± 0.009	0.005 ± 0.002	0.019 ± 0.008	0.013 ± 0.006
Trajectory 3	No	0.1 ± 0.2	0.1 ± 0.1	0.010 ± 0.002	0.006 ± 0.001	0.2 ± 0.4	0.1 ± 0.2
Trajectory 3	Yes	0.1 ± 0.2	0.1 ± 0.1	0.02 ± 0.01	0.005 ± 0.002	0.2 ± 0.3	0.1 ± 0.2

entire trajectory are shown in Table 4.2, the absolute RMSE of rotation matrices computed with the Frobenius norm of the rotation matrices or the z-axis orientation axis angle error are displayed in Table 4.3.

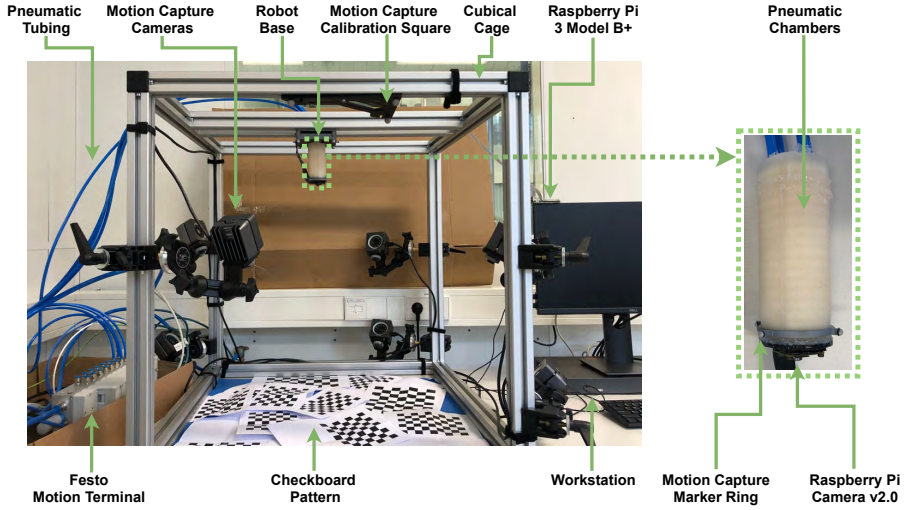
Our results show translation errors of on average 6 % to 9 % for short segments and 2 % to 6 % RMSE relative to the trajectory length for long segments before optimization. The projection into PCC kinematics significantly decreases the translational error by between 66 % and 96 % to 0.4 % to 2 % for short segments and 0.6 % to 2 % for long segments. We state an absolute RMSE for the orientation estimates of the z-axis of the tip e_{θ_z} as defined in Eq. 4.13 of between 0.005 rad and 0.1 rad after optimization. The rotational error of the orientation estimates varies by trial but, on average, stays constant across the optimization. Choosing a bigger weight λ_R on the rotational loss during the optimization resulted in larger improvements for estimation of the orientation at the cost of higher translational errors.

4.4 Experiments

We confirm the simulation results in a preliminary experimental study by mounting a Raspberry Pi camera to the tip of a soft segment (Katzschmann et al., 2019). The robotic segment is guided to follow three trajectories similar to the ones tested in simulation (see Section 4.3.2). A motion capture setup is employed to gather an accurate ground truth on the shape of the segment.

4.4.1 Experimental Setup

We show the experimental setup in Figure 4.4(a). We consider a soft robotic silicone segment consisting of four independently inflatable cavities. The segment has a cylindrical shape with a length $L_{0,1}$ of 11 cm and a radius d_1 of 21 mm. A 3D-printed ring with four motion capture markers is located near the tip of the segment. We mount a Raspberry Pi camera module v2.0 to the tip of the segment. This camera has a 8 MP sensor and records frames at a sampling rate of 30 Hz and a resolution of 1080p. The focal length is 3.04 mm and the field of view is $62.2^\circ \times 48.8^\circ$. The camera module is attached to a Raspberry Pi 3B+ single-board computer, which saves the frames for later processing by the ORB-SLAM (Mur-Artal and Tardós, 2017) algorithm. The camera is screwed onto a custom 3D-printed holder, which in turn is glued with the tip plane of the segment. The segment with its four air chambers is actuated with a proportional pressure regulator. Tubing attached to the base of the segment connects each chamber with the assigned pneumatic valve of the pressure regulator. The segment is attached in an up-side-down configuration to the top plane of a cubical cage of 750 mm side length. For a straight segment, the camera is facing downwards towards the floor of the cage, which is covered by multiple printed checkerboard patterns. We acquire ground-truth pose information of the tip of the segment using an Optitrack motion capture system. The ground-truth poses of the tip of the segment are recorded at 30 Hz. We also include the elongation of the segment δL_1 in the cost function (4.6) of our optimization. To resemble the calibration sequence from the simulation for the SLAM map, we manually move the robot laterally into the x-coordinate direction before fixing it to the cage for the start of the experiments.



(a) Experimental setup

(b) Trajectory 3

Figure 4.4: In Panel (a), a soft robotic segment is mounted to a cage with attached motion capture cameras. The segment is pneumatically actuated by a pressure regulator (Festo Motion Terminal). A Raspberry Pi camera v2.0 is attached to the tip of the segment and, in a straight segment configuration, looks down towards check-board patterns. Panel (b) depicts a sequence of stills showing the robot following trajectory 3 from two different vantage points. The second vantage point differs 90° from the first one. The third row displays a few representative frames, as recorded by the camera attached to the tip of the segment.

Table 4.4: Real-world results before and after optimization. The translational errors are stated through a relative RMSE as described in (4.10). For rotation, we report both an absolute RMSE computed with the Frobenius norm between the rotation matrices as stated in (4.11) and an angle error [rad] for the orientation of the z-axis of the tip of the segment as defined in (4.13). The results are averaged over two trials for each trajectory.

Error category	**Opt.**	**Traj. 1**	**Traj. 2**	**Traj. 3**
Translation e_t	No	20.3 %	14.2 %	9.1 %
Translation e_t	Yes	9.1 %	8.9 %	5.0 %
Rotation e_R	No	0.145	0.103	0.126
Rotation e_R	Yes	0.130	0.099	0.120
Rotation e_{θ_z}	No	0.079 rad	0.068 rad	0.084 rad
Rotation e_{θ_z}	Yes	0.080 rad	0.067 rad	0.084 rad

4.4.2 Results

Our experimental results reported in Table 4.4 and visualized for trajectory 3 in Figure 4.5 show translational relative RMSE of between 9 % and 20 % for the three trajectories before optimization. The orientation of the z-axis of the tip is estimated with a mean error of approximately 0.075 rad. The translational error is improved to between 5 % and 9 % after projection into the PCC-kinematics. The optimization also slightly improves the rotational RMSE by 4 % to 10 % relative to naive SLAM.

The experimental results of the SLAM algorithm are coherent with the simulations, as the small segment length (11 cm) used in the experiments increases the translational errors, as shown similarly in the simulations for a robot of length 15 cm. Even though the translational error is greatly reduced through optimization, it is still significantly higher than in simulation. Two reasons for this difference could be that a) the segment in the simulation was modeled as in-extensible, while the real robot segment is extended via pneumatic pressurization, which introduces additional errors by SLAM not accurately estimating the elongation movement and b) the real robot does not perfectly behave according to the CC approximation as the simulated robot does.

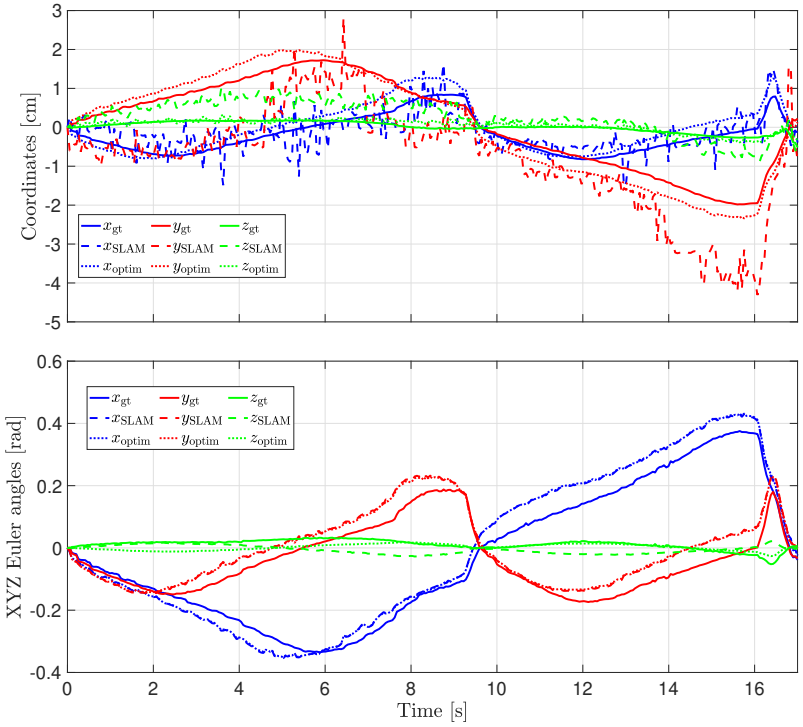


Figure 4.5: Experimental results for trajectory 3. Comparison between ground-truth (solid line), SLAM (dashed line), and optimized through projection into PCC kinematics (dotted line) for translation and orientation estimates.

4.5 Conclusion

This chapter investigated using a monocular camera in shape sensing of continuum soft robots, with the ultimate goal of implementing precise and reliable estimations at the cost of introducing small rigid parts into the hardware design. The contribution of this chapter has been twofold. First, we proposed to use monocular SLAM with a soft robot. Second, we propose a regularization of the estimation based on a nonlinear projection in the manifold of the admitted configuration. A nonlinear optimization implements the latter based on the kinematic model of the robot. We have performed extensive simulations with rendered images in Blender and lab experiments with a single-segment soft robot. The nonlinear optimization based on the robot's kinematic model led to a significant improvement in translations and a marginal improvement in rotations. Future work will focus on extending the experimental validation of the method to multiple segments and cameras, bettering the SLAM by feeding back the kinematic projection in its state, and using this estimation to implement closed-loop control. While we conducted our experiments in a lab environment under ideal conditions with the camera pointed at checkerboard patterns, thus resulting in plenty of image features for SLAM to track, future work should investigate whether deployment environments for soft robots would be sufficiently feature-rich for the use of

our proposed method.

Afterword

This chapter demonstrated how inexpensive commercialized monocular cameras can be effectively used together with established SLAM algorithms and kinematic models (e.g., PCC) to achieve shape sensing for soft robots. One of the main advantages of this solution is that all necessary components are readily available - either commercially or even via open source. Sharing the same vision, recent work by Caroleo et al. (2025) integrates commercial miniature Time of Flight (TOF) sensors mounted at the segment tip with a particle filter to estimate the end-effector pose by considering the deviation from a known map of the environment.

However, the approach presented in this chapter also has some drawbacks: i) Firstly, while cameras have been significantly miniaturized in recent years, the requirement for them to have a clear, unobstructed view of the environment necessitates the inclusion of a rigid component on the surface of the soft robot, which we generally prefer to avoid for safety reasons. Secondly, ii), the performance of SLAM algorithms is affected in environments with limited visually distinguishable features. Finally, iii) high-dimensional perceptive data, such as images, are generally computationally expensive to process, leading to higher computational requirements and/or relatively low sampling rates of the shape-sensing information. Therefore, we present in Chapter 9 an alternative shape-sensing approach based on magnetic sensors. The necessary magnets and magnetic sensors can be deeply embedded into the soft robot body, thus allowing us to keep the robot surface entirely soft and compliant. Furthermore, the sensory data is several orders of magnitude lower-dimensional, and thus, its processing is potentially much less computationally demanding.

5

Modeling of Handed Shearing Auxetics (HSA) Robots

Foreword. Novel soft robots based on HSAs show great promise by integrating multiple DOF in a compact form. Their parallel nature promises a larger payload capacity while preserving compliance. Notably, they represent a paradigm shift in the soft robotics domain by i) establishing compliance through their metamaterial structure rather than material softness and ii) transmitting actuation through their elasticity with the auxetic metamaterial instead of directly applying forces and torques into the desired directions of deformation. However, despite the extensive literature on their design, fabrication, and integration of sensing, the absence of kinematic and dynamic models hinders their widespread adoption. These models are crucial for developing accurate simulators and effective model-based controllers. This chapter addresses this gap by demonstrating how kinematic (Section 5.3) and dynamical (Section 5.2) models can be derived for general 3D motion scenarios. Furthermore, we present in Section 5.4 low-dimensional kinematical parametrizations and a control-oriented Euler-Lagrangian model tailored explicitly to planar HSA robot motions. We follow here a strategy of exploiting the vast and well-grounded literature on control-oriented modeling of rigid robotic manipulators and augment them with a specialized kinematic model based on the PCS parametrization and a nonlinear, actuation-dependent elastic model.

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q} = Q$$

Abstract. *Electrically-actuated continuum soft robots based on HSAs promise rapid actuation capabilities while preserving structural compliance. However, the foundational models of these novel actuators required for precise control strategies are missing. This chapter proposes two key components extending the Discretized Cosserat Rod Model (DCM) to allow for modeling HSAs. First, we propose a mechanism for incorporating the auxetic trajectory into DCM dynamical simulations. We also implement this extension as a plugin for the Elastica simulator. Second, we introduce a SPCS kinematic parameterization that can describe an HSA rod's shape with fewer configuration variables. Subsequently, based on the augmented DCM model, we devise a low-dimensional kinematic and a control-oriented dynamical model for planar HSA robots. We verify all theoretical contributions experimentally. The HSA robot simulator is used to replicate experimental data of the mechanical characterization of HSA rods. For the kinematic description of the HSA rods, we attach motion capture markers at various points to a parallel HSA robot and find that the shape of the HSAs can be kinematically represented with an average accuracy of* 0.3 mm *for positions and* 0.07 rad *for orientations. Finally, the identified Euler-Lagrangian model for planar HSA robots exhibits exceptional accuracy at predicting future system states over a horizon of more than* 4 s.

5

This chapter is partly based on

- ***M. Stölzle**, L. Chin, R. L. Truby, D. Rus, and C. Della Santina (2023, April). Modeling Handed Shearing Auxetics: Selective Piecewise Constant Strain Kinematics and Dynamic Simulation. In 2023 IEEE International Conference on Soft Robotics (RoboSoft) (pp. 1-8). IEEE* (Stölzle et al., 2023).
- ***M. Stölzle**, D. Rus, and C. Della Santina (2023, December). An Experimental Study of Model-based Control for Planar Handed Shearing Auxetics Robots. In Experimental Robotics: The 18th International Symposium. Springer* (Stölzle et al., 2024b).

M.S. and C.D.S. conceived the project. M.S. and C.D.S. devised the modeling approaches. L.C., R.L.T., and D.R. designed and fabricated the HSA robot. M.S. implemented the model, planned and executed the simulations and the experiments, performed the data analysis, and wrote the manuscript. C.D.S. supervised the project and revised the manuscript. C.D.S. and D.R. provided funding.

5.1 Introduction

Robots based on Handed Shearing Auxetic (HSA) are a recent development in the soft robotics field (Chin et al., 2018, 2019; Lipton et al., 2018; Truby et al., 2021; Zhang et al., 2022a), which directly transform applied motor torques into complex motion primitives. This novel type of actuator is based on an architected metamaterial. The most important characteristic of this cylindrical metamaterial is that twist strains along the handedness of the structure lead to an elongation of the rod, which is also called auxetic trajectory (Good et al., 2022). An HSA robot combines multiple HSAs of different handedness with a platform constraining the movement of the distal ends in the fashion of a soft parallel manipulator. Differential elongation of the rods enables complex motion primitives such as elongation, bending, and twisting (Chin et al., 2018), which can be seen in Fig. 5.2. HSA robots are particularly difficult to model and control as the forces and torques causing the evolution of the system are not directly produced by the actuator but instead are intrinsically generated as an effect of the modified cell state of the metamaterial and of the interaction forces coming from the parallel arrangement.

3D-FEM based approaches (Farrell et al., 2020) have proven to be effective in simulating soft parallel structures (Vanneste et al., 2021) and could be a good candidate for representing the complex behavior of HSA robots. However, in this chapter, we strive for a less computationally expensive solution - towards applications in model-based control (Della Santina et al., 2023). For this reason, we look at the framework of the DCM. The Cosserat rod theory assumes the slenderness of the object, e.g., that the length is much larger than the radius, and allows for the rod to exhibit all six principal strains. The 1D discretization of the rod along its length dramatically reduces the computational demand compared to FEM (Gazzola et al., 2018). Several works in recent literature have successfully applied this framework to soft robotics (Armanini et al., 2023; Grazioso et al., 2019; Sadati et al., 2021). Among them, in PCS (Renda et al., 2018), the continuum dynamics of the Cosserat model are discretized in space by keeping a selection of strains constant along a segment of the continuum. The most popular PCS is PCC (Webster III and Jones, 2010), which assumes a sequence of arcs. Functional extensions of PCS use a continuous function to approximate the strain (Della Santina and Rus, 2019; Renda et al., 2020).

However, none of these methods are currently applicable to HSA robots, as they do not embed a mechanism for incorporating the effect of the auxetic trajectory. We are aware of just one work looking into kinematic modeling of HSA robots (Garg et al., 2022), which,

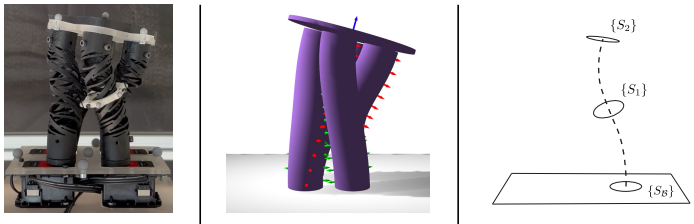


Figure 5.1: An HSA robot in a twisted state: simulation and schematic of the kinematic model of a single HSA rod.

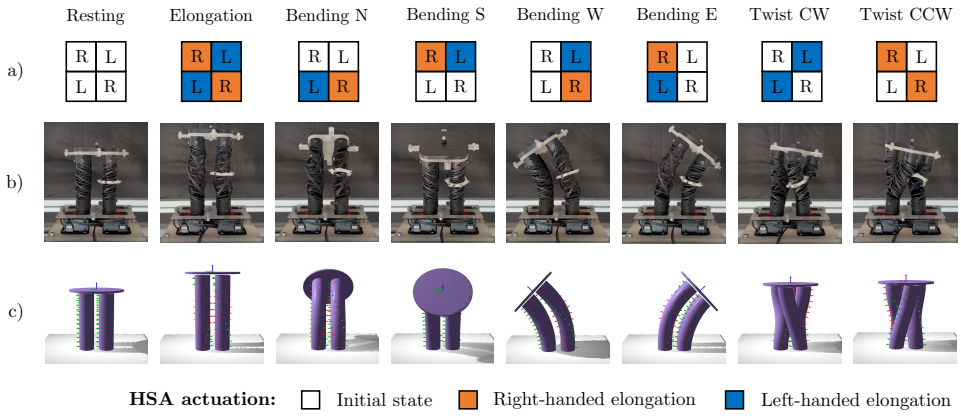


Figure 5.2: Motion primitives of HSA robots: elongation, bending in the four cardinal directions (e.g., north (N), south (S), west (W), east (E)), and clockwise (CW) and counter-clockwise (CCW) twisting. **First row (a):** depicts the necessary actuation inputs to generate these motion primitives. Pure elongation is achieved by applying the motor torques of the same magnitude but in opposite directions to the left-handed (L) and right-handed (R) HSAs. For bending, a delta exists in the elongation of the rods, while the sum of torques is still zero. Last but not least, counter-clockwise twisting is achieved by applying more torque to the right-handed than to the left-handed HSA rods. **Second row (b):** Snapshots of the experimental platform when actuated according to the above-specified sequence. **Third row (c):** Renderings of simulated steady-states of an HSA robot. It consists of four HSA rods and a platform at the distal end. The red arrows point along the local x-axis, and the green arrows along the local y-axis, respectively. The blue arrow signifies the z-axis of the local frame of the platform.

however, models the backbone of the robot with PCC instead of modeling the HSAs. As a consequence, the model cannot represent complex behaviors of the module, like the twist in Fig. 5.1.

The goal of this chapter is to provide such a mechanism for the general movement of HSA robots in 3D-space, to introduce a strategy for further reducing the dimensionality of the model, to derive a control-oriented model for planar HSA robots, and to provide extensive experimental validation for all. More specifically, our extension to the DCM framework couples the twisting strain of the HSA rod to its rest length. Additionally, we allow the rigidity of the rod to be modified as a function of the twist strain. We have implemented this mechanism as a plugin for Elastica (Naughton et al., 2021), which we provide open source[1].

We then use a combination of CS and PCSs (Renda et al., 2018) to describe the shape of HSA rods, which we call a SPCS model: while some strains, such as twist & stretch, are mostly constant over the length of the entire HSA, other strains, such as bend & shear, significantly vary and are thus captured in a piecewise parametrization. Our results show that a kinematic parametrization with 11 DOFs is sufficient to capture the shape of HSAs. Compared to the DCM strategy used for simulating HSAs, we have, therefore, significantly reduced the DOF of the kinematics. We provide an open-source implementation of this kinematic model in JAX[2].

[1] https://github.com/tud-phi/HSA-PyElastica
[2] https://github.com/tud-phi/jax-spcs-kinematics

This kinematic model could then be used to parametrize the deformation of each limb in the parallel HSA rod. However, this would require the enforcement of kinematic constraints (Armanini et al., 2021), which can be algorithmically and computationally challenging. We propose to avoid this complexity in the planar case by natively incorporating the kinematic constraints into the model. In particular, we defining the CS of a virtual backbone in the center of the robot to be our configuration variable. Finally, we derive the system dynamics of a planar HSA robot in Euler-Lagrangian form.

In summary, we contribute to the state of the art in the modeling of soft robots with:

1. A mechanism for integrating the auxetic trajectory of HSAs into the discrete Cosserat rod theory (Gazzola et al., 2018; Mathew et al., 2022).
2. A plugin for the Elastica simulator (Naughton et al., 2021), which also includes the necessary boundary conditions and joint formulations to simulate HSA robots.
3. A Selective Piecewise Constant Strain (SPCS) kinematic model to parameterize the shape of a single HSA rod with a dramatically reduced number of states.
4. A closed-form solution for the inverse kinematics of a planar CS formulation.
5. A Euler-Lagrangian dynamical model for planar HSA robots.

Contributions (1) and (2) are covered in Section 5.2. Subsequently, we introduce the kinematic model from contribution (3) and experimentally verify it in Section 5.3. The control-oriented model for planar HSA robots from contributions (4) and (5) and the experimental validation are presented in Section 5.4.

5.2 Dynamic Simulation of HSA Robots

We introduce a new concept to enable the simulation of HSA robots with the discretized Cosserat rod theory, which is used by many of the SOTA simulators of soft continuum robots (Mathew et al., 2022; Naughton et al., 2021). While we provide an implementation of the proposed concept as a plugin to Elastica (Naughton et al., 2021), the same strategy could be used to adapt other simulators, such as SoRoSim (Mathew et al., 2022), to HSA robots. We give some background on the DCM in Section 5.2.1. Then, in 5.2.2, we propose a mechanism to infuse the auxetic trajectory for a HSA into the DCM framework. Subsequently, we verify the steady-state behavior of an HSA against the mechanical characteristics in 5.2.3. Next, we lay out in 5.2.4 the necessary boundary conditions of the HSAs and describe the joint mechanism connecting the platform with the rods. Finally, we explain in Section 5.2.5 how we were able to reproduce in simulation the main motion primitives of HSA robots.

5.2.1 Background: Discretized Cosserat Rod Model

This subsection will introduce the governing equations of the DCM following the work by Gazzola et al. (2018). According to the Cosserat rod theory, a slender rod's shape can be purely described by the line along its backbone. The backbone curve is divided into a discrete set of nodes with position $r_i(t) \in \mathbb{R}^3 \, \forall \, i \in \{1, \dots, n_v + 1\}$ and n_v links of orientation $Q_i(t) \in \mathbb{R}^{3\times3}$. Differentiating the position and orientation with respect to time gives the translational and angular velocities $v_i = \frac{\partial r_i}{\partial t} \in \mathbb{R}^3$ and $\omega^i_{\mathcal{L}} \in \mathbb{R}^3$. Each node has a mass of m_i,

and the rigid links are modeled to have a second mass moment of inertia J_i. When a rod of unstretched length $\hat{L}$ is at rest, each link has a length of $\hat{l}_i$ and connects two consecutive vertices. The circumflex accent will denote quantities in the rest configuration of the rod. When the rod is in a deformed state, l describes the current edge length and the shear and axial strains are considered in the vector $\sigma = \begin{pmatrix}\sigma_x & \sigma_y & \sigma_z\end{pmatrix}^\top$. The curvature vector $\kappa_{\mathcal{L}} = \begin{pmatrix}\kappa_x & \kappa_y & \kappa_z\end{pmatrix}^\top$ captures the bending and twist strains. All strains are defined with respect to the rest length of the link $\hat{l}_i$, and the dilation factor $e_i = \frac{l_i}{\hat{l}_i}$ denotes the deviation from that rest length. The shear and stretch stiffness is specified through the diagonal matrix $S = \mathrm{diag}(EI_{xx}, EI_{yy}, GI_{zz}) \in \mathbb{R}^{3\times3}$, where E, G are the elastic and shear modulus respectively, and $I \in \mathbb{R}^{3\times3}$ is the second area moment of inertia. Analogue, the bending and twist rigidity is stored in $B = \mathrm{diag}(B_x, B_y, B_z) \in \mathbb{R}^{3\times3}$. For conciseness, we include only the equation for the translational accelerations below. We refer the interested reader to (Gazzola et al., 2018) for the equation on rotational accelerations and more complementary details about the DCM.

5

$$m_i \frac{\partial v_i}{\partial t} = \Delta^h \left(\frac{Q_i^\top \hat{S}_i \sigma_{\mathcal{L}}^i}{e_i} \right) + F_i, \quad i \in \{1, \dots, n_\mathrm{v} + 1\}, \tag{5.1}$$

where $F_i \in \mathbb{R}^3$ is the external force acting on the ith vertex. Several quantities are expressed in the Voronoi domain $\mathcal{D}$, in which the length of the region $\mathcal{D}_i$ can be computed as $\mathcal{D}_i = \frac{l_{i+1}+l_i}{2}, i \in [1, n_\mathrm{v} - 1]$. Examples are the the Voronoi curvature $\hat{\kappa}_{\mathcal{L}}^i$ over the interior vertices, and the bend twist stiffness matrix $\hat{B}_i$. $\Delta^h : \{\mathbb{R}^3\}_N \to \{\mathbb{R}^3\}_{N+1}$ is used as the discrete difference operator.

5.2.2 Auxetic Trajectory

We propose several adjustments to the standard definition of the DCM to allow for realistic simulation of HSAs. The main assumption behind the proposed concept is that twist strains agreeing with the handedness of the rod will modify the internal angle between the auxetic pattern cells and, with that, also change the system characteristics such as spring constant, blocked force, etc.

Most importantly, we introduce a distinction between the printed, initial length of the HSA $\bar{L}$ and the rest length of the rod $\hat{L}$. This allows us to mirror the auxetic trajectory, as the minimum energy length is increased with applied twist angles / strains (Good et al., 2022). Similar to the hat accent, which denotes rest quantities, the bar accent will point out quantities of the HSA in the initial / printed state. We propose to linearly scale the edge rest length $\hat{l}_i$ with the twist strain $\kappa_{\mathcal{L},z}^i$:

$$\hat{l}_i = (1 + \varepsilon_i)\, \bar{l} \quad i \in \{1, \dots, n_\mathrm{v} + 1\}, \tag{5.2}$$

$$\varepsilon_i = \max\left(\min\left(h\, C_\varepsilon\, \mathcal{A}^h(\kappa_{\mathcal{L},z}^i), \varepsilon_{\max}\right), \varepsilon_{\min}\right). \tag{5.3}$$

In this expression, the twist strain $\kappa_{\mathcal{L},z}^i$ is elevated from the Voronoi to the vertex domain with the averaging operator $\mathcal{A}^h : \{\mathbb{R}^3\}_N \to \{\mathbb{R}^3\}_{N+1}$. $h \in \{-1, 1\}$ is the handedness of the rod. Right is defined as the positive, and left as the negative handedness. C_ε is the extension factor, which needs to be tuned with respect to the chosen auxetic pattern. The minimum

and maximum extension $\varepsilon_{\min}$, $\varepsilon_{\max}$ are the limits of the auxetic trajectory and depend on the HSA type: for example, closed HSAs can only exhibit positive elongations (Good et al., 2022). After the rest length is adjusted, the axial stiffness of the rod will guide the current edge length l_i towards the (target) edge rest length. Furthermore, we recall the definition of bend/twist strains: $\kappa^i_{\mathcal{L}} = \frac{\log(Q_{i+1}\, Q_i^\top)}{\hat{D}_i}$. To keep the twist strain constant across the entire auxetic trajectory, we define the twist strain with respect to the initial Voronoi length $\bar{D}_i = \frac{\bar{l}_{i+1}-\bar{l}_i}{2}$:

$$\kappa^i_{\mathcal{L},z} = \frac{\log(Q_{i+1}\, Q_i^\top)}{\bar{D}_i}, \quad i \in \{1, \dots, n_\mathrm{v}+1\} \tag{5.4}$$

Table 5.1: Parameters of simulated HSA rods in Section 5.2.3 for various number of HSA row tilings n_{rows}. Row tilings represent the number of vertically stacked unit cells (Good et al., 2022). The rest length $\hat{L}$ and the elastic modulus E are linear functions of the twist strain κ_z. B_z represents the twist rigidity.

n_{rows}	$\hat{L}$ [mm]	E [kPa]	B_z [Nm^2/rad]
4	$75(1+3.04\,\kappa_z)$	$576.9+36.1\,\kappa_z$	0.00375
6	$89(1+3.50\,\kappa_z)$	$309.3+13.1\,\kappa_z$	0.00213
8	$100(1+3.77\,\kappa_z)$	$203.5+10.6\,\kappa_z$	0.00183
10	$112(1+3.64\,\kappa_z)$	$197.6+7.5\,\kappa_z$	0.00167
12	$124(1+3.53\,\kappa_z)$	$197.6+2.4\,\kappa_z$	0.00124

Finally, recent work by Good et al. (2022) has shown that HSAs exhibits special mechanical characteristics, such as the spring constant increasing with the twist angle. Therefore, we allow the shear/stretch and bend / twist rigidity matrices S and B to be modified dynamically during the simulation with the twist strain. For example, the axial stiffness of a closed HSA can be modeled as a linear function of the twist strain (Good et al., 2022)

$$\hat{S}^i_z = \bar{S}^i_z + C_{S_z}\mathcal{A}^h(\kappa^i_{\mathcal{L},z}), \tag{5.5}$$

where C_{S_z} is a tunable constant.

5.2.3 Verification of HSA Rod Steady-State Behaviour

We verify that our simulator can represent the steady-state behavior of a real HSA by re-producing the characterization results for closed HSAs by Good et al. (2022). More specifically, we let the simulator converge to a steady state and then identify several mechanical properties such as blocked force (F_b), minimum energy length, holding torque (τ_h), and the spring constant (k). Following the reporting in (Good et al., 2022), we tune the parameters of our simulation to match the behavior of closed Carbon FPU50 HSAs with 19 mm outside diameter, 2 mm wall thickness, as good as possible. We report the chosen simulation parameters in Table 5.1. The HSA rod is modelled to consist of $n_\mathrm{v} = 10$ nodes and 9 links with a material density of $\rho = 1050\,\mathrm{kg/m^3}$. For all simulations, the proximal end of the rod is constrained, and only rotations around the z-axis are allowed to mirror the actuation with electric motors. Furthermore, twisting is constrained at the distal end, which allows twist strains to build up in the rod. Otherwise, the distal end is unconstrained.

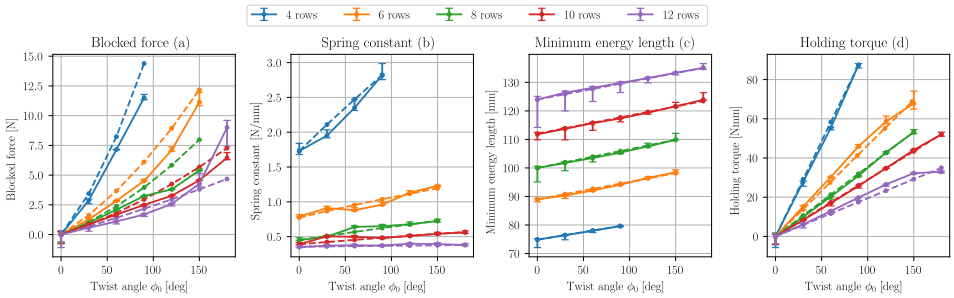


Figure 5.3: Results for verification of steady-state behavior of the proposed simulator: the solid lines represent the mechanical characteristics obtained for closed HSA rods by Good et al. (2022) with corresponding error bars. The dashed lines correspond to the same characteristics obtained with our simulator. The simulation parameters are separately tuned for HSAs with a variety of row tilings. When an HSA contains a higher number of row tiling, it will allow for larger elongations while simultaneously trading off the spring constant (Good et al., 2022).

5

Next, we will go into more detail about each mechanical characteristic. **Holding torque:** We apply a given torsional torque τ_h at the proximal end of the HSA and then record the twist angle of the base ϕ_0 at steady-state. **Minimum energy length:** The proximal end of the HSA is rotated to a given twist angle ϕ_0. The minimum energy length is then identified as the steady-state length of the HSA. **Spring constant:** For a given twist angle ϕ_0 with the HSA at rest, the spring constant is identified by applying a small pulling force to the distal end and then measuring the displacement of the tip at steady-state. **Blocked force:** Different from the other simulations, the distal end is constrained at its initial position to prevent the rod from extending. The blocked force F_b is identified by evaluating the internal axial force for a given twist angle.

The results show that the proposed simulator can accurately represent the steady-state behavior of the HSAs with the simulated characteristics mostly staying within the stated error range of the experimental measurements by Good et al. (2022). The only exception is Fig. 5.3(a), in which the simulation is overestimating the blocked force F_b. This points to the fact that this linear approximation of the auxetic trajectory is only accurate in a limited range of the motion range of the closed HSAs. Further research is necessary to come up with auxetic trajectory models for semi-closed and open HSAs.

5.2.4 Simulating HSA Robots: Boundary Conditions and Joints

We discuss here how to combine a platform and multiple HSAs to form a HSA robot. Assume to have n_{HSA} rods equally distributed along a circle of radius R_{cHSA} in the x-y plane with the rods pointing towards the positive z-direction in a straight configuration. We need boundary conditions for the proximal ends of the rods to generate the parallel structure. The positions of the proximal nodes are constrained to remain at their initial position $\bar{r}_0$. For the same purpose, the translational rates, e.g., v_0, are set to zero at each time step.

In our plug-in to Elastica, we provide the user with two options for actuating the HSAs. (a) The orientation of the proximal link Q_0 is moved to a desired orientation Q_0^d. In this case, the twist angle ϕ_0^d of the proximal end is controlled. Again, the rotational rates ω_0

are set to zero. (b) Twist torques $\tau_{0,z}$ are applied to the proximal link of the HSA. The two remaining rotational DOFs (rolling and pitching) of the proximal link are constrained by setting their rotational rates $\omega_{0,x}$ and $\omega_{0,y}$ to zero.

Additionally, rigid joints between the rods and the platform are necessary. These are achieved by simulating a spring-damper system between the distal end of each HSA and the platform. For the translations, we compute the contact force F_c as

$$F_\mathrm{c} = k_F\,(r_\mathrm{p}^j - r_{n_\mathrm{v}+1}) + \nu_F\,(v_\mathrm{p}^j - v_{n_\mathrm{v}+1}), \tag{5.6}$$

where k_F is the translational joint stiffness, and ν_F the translational damping coefficient. While $r_{n_\mathrm{v}+1}$, and $v_{n_\mathrm{v}+1}$ are the position and the velocity of the distal node of the rod, respectively, r_p^j and $v_{n_\mathrm{v}+1}^j$) are the position and velocity of the attachment point of the same rod (e.g. the jth rod) on the platform. We determine the position and velocity of this attachment point using rigid body kinematics with regard to the Center of Mass (COM) of the platform. The contact force F_c is applied with an opposite sign to the distal end of the HSA and to the platform, respectively. Please note that the contact force F also generates a torque τ_{F_c} on the platform, as the force is not applied at the COM of the rigid body.

Similarly to the contact force, a contact torque τ_c is computed to reduce any error in the orientation and angular velocity between the two systems

$$\tau_\mathrm{c} = k_\tau\,(Q_\mathrm{p}^\top \log(Q_\mathrm{p}\,Q_{n_\mathrm{v}}^\top)) + \nu_\tau\,(Q_\mathrm{p}^\top \omega_\mathrm{p}^j - Q_{n_\mathrm{v}}^\top \omega_{n_\mathrm{v}}) \tag{5.7}$$

where the $\log(\cdot) : \mathbb{R}^{3\times3} \to \mathbb{R}^3$ operator computes the rotation vector from the rotation matrix (Gazzola et al., 2018), and Q_p is the material frame of the platform.

5.2.5 Qualitative Evaluation of Motions in Simulation

We reproduce the typical motion primitives of a HSA robot consisting of four HSAs (e.g., $n_\mathrm{HSA} = 4$) in simulation and show the final steady-states in Fig. 5.2. Two of the HSAs are left-handed and positioned diagonally from each other. Each rod is discretized by $n_\mathrm{v} = 25$ links and 26 point mass vertices. Furthermore, it has a printed length of $\bar{L} = 100\,\mathrm{mm}$, an outside radius of 25.4 mm and a wall-thickness of 2.43 mm. The rods are placed at a radial distance of $R_\mathrm{cHSA} = 24\,\mathrm{mm}$ from the center of the robot, and a material density of $\rho_\mathrm{HSA} = 1050\,\mathrm{kg/m^3}$ is assumed. Therefore, the chosen simulation parameters mirror the geometric characteristics of our experimental platform. Based on an elastic modulus $E = 10\,\mathrm{MPa}$ and a shear modulus $G = 0.6\,\mathrm{MPa}$, the shear and stretch stiffnesses amount to $S_{x,y} = 101.5\,\mathrm{N/m}$, and $S_z = 1753.5\,\mathrm{N/m}$. We set the bend and twist rigidities B_x, B_y, and B_z to $0.02\,\mathrm{Nm^2/rad}$ and $0.014\,\mathrm{Nm^2/rad}$ respectively. When twist strains are present, we extend the rest length of the rod by $0.01\,\mathrm{m/rad}$ after taking into account the handedness of the HSA.

The cylindrical platform is of diameter 95 mm, has a thickness of 3 mm and is modeled to have a density of $\rho_\mathrm{p} = 700\,\mathrm{kg/m^3}$. The joint stiffness parameters $k_F = 5 \cdot 10^5\,\mathrm{N/m}$ and $k_\tau = 20\,\mathrm{Nm/rad}$ are chosen for the fixed joint between HSAs and platform. The joint damping coefficients ν_F, ν_τ are set to zero.

Our qualitative results in Fig. 5.2 demonstrate that we are able to generate all motion primitives in simulation. For the shown deformations, we apply maximum twist angles of $\phi_{0,\max} = \pi\,\mathrm{rad}$.

5.3 A Kinematic Model for HSA Rods

In this section, we aim to derive a forward kinematic model that can be used to describe the shape of a single HSA rod with a minimum amount of parameters. More specifically, we want to describe the transformation from the base frame $\{S_\mathcal{B}\}$ to a local frame $\{S_s\}$, a coordinate $s \in (0, \bar{L}]$, which is a function of the soft robot's configuration q. This coordinate lies on the backbone of a HSA rod of printed (i.e., initial) length $\bar{L}$.

5.3.1 Selective Piecewise Constant Strain (SPCS) Kinematics

For this purpose, we combine the existing kinematic models CS and PCS (Renda et al., 2018) to selectively keep specific strains constant along the entire length of the robot or vary them piece-wise among the segments. This parametrization can be combined with the results in Sec. 5.2 to generate a compact dynamic model for the movement of HSA robots in 3D space.

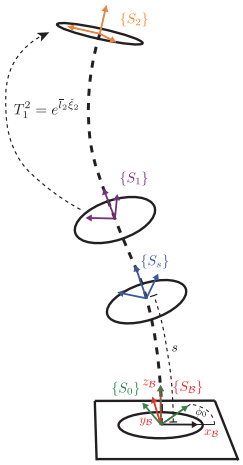


Figure 5.4: Visualization of the proposed SPCS kinematic model for the case of $n_\mathrm{S} = 2$ segments: The forward kinematics describe a transformation from the base frame $\{S_\mathcal{B}\}$ to the local frame $\{S_s\}$ at the coordinate $s \in [0, \bar{L}]$ and consist of a) a rotation around the $z_\mathcal{B}$-axis of the base frame by angle ϕ_0, b) an exponential map $e^{(s-\bar{L}_i)\hat{\xi}_i}$ for the transformation from the proximal end of the ith segment to the local frame of the coordinate s.

First, we define that

$$\xi(q, s) = \begin{pmatrix} \kappa_x & \kappa_y & \kappa_z & \sigma_x & \sigma_y & \sigma_z \end{pmatrix}^\top \in \mathbb{R}^6 \tag{5.8}$$

represents the three rotational and three linear strains present in a rod (Renda et al., 2018). Subsequently, propose the following configuration vector for an HSA

$$q = \begin{pmatrix} \phi_0 & q_\mathrm{CS}^\top & q_{\mathrm{PCS},1}^\top & \cdots & q_{\mathrm{PCS},i}^\top & \cdots & q_{\mathrm{PCS},n_\mathrm{S}}^\top \end{pmatrix}^\top \tag{5.9}$$

where $\phi_0 \in \mathbb{R}$ is the twist angle at the base and allows for the rotation of the motor actuating the HSA rod. $q_\mathrm{CS} \in \mathbb{R}^{n_{q,\mathrm{CS}}}$ is a strain component constant along the entire rod, and $q_{\mathrm{PCS},i} \in \mathbb{R}^{n_{q,\mathrm{PCS}}}, i \in \{1, \ldots, n_\mathrm{PCS}\}$ is the configuration of each PCS segment. The ith segment

has a initial length of $\bar{l}_i$ with its tip at the coordinate $\bar{L}_i$ The strain in the ith segment is then the sum of the rest strain $\xi^0 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}^\top$, ξ_{CS}, and $\xi_{\mathrm{PCS},i}$:

$$\xi_i = \xi^0 + \Phi_{\mathrm{CS}}\, q_{\mathrm{CS}} + \Phi_{\mathrm{PCS},i}\, q_{\mathrm{PCS},i}, \quad i \in \{1, \dots, n_{\mathrm{S}}\}. \tag{5.10}$$

Analogue to the concept introduced in (Renda et al., 2020), $\Phi_{\mathrm{CS}} \in \mathbb{R}^{6\times n_{q,\mathrm{CS}}}$, $\Phi_{\mathrm{PCS}} \in \mathbb{R}^{6\times n_{q,\mathrm{PCS}}}$ are the strain bases of q_{CS} and q_{PCS} respectively.

In this chapter, we specifically investigate a setting where the twist & stretch strains are constant across the entire rod and the bend & shear strains vary for each segment. Accordingly, we choose $q_{\mathrm{CS}} = \begin{pmatrix} \kappa_z & \sigma_z \end{pmatrix}^\top$ and $q_{\mathrm{PCS},i} = \begin{pmatrix} \kappa_{x,i} & \kappa_{y,i} & \sigma_{x,i} & \sigma_{y,i} \end{pmatrix}^\top$. Then, the corresponding strain bases are determined to be

$$\begin{aligned} \Phi_{\mathrm{CS}} &= \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^\top \in \mathbb{R}^{6\times 2}, \\ \Phi_{\mathrm{PCS},i} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^\top \in \mathbb{R}^{6\times 4}. \end{aligned} \tag{5.11}$$

Next, we find homogeneous forward kinematic mappings for the given configuration and strains. As the twist angle ϕ_0 demands a rotation around the local $z_{\mathcal{B}}$-axis of the base frame, the matrix $R^0_{\mathcal{B}}(\phi_0) \in SO(3)$ contains the rotation from the base frame $\{S_{\mathcal{B}}\}$ to the proximal end of the rod denoted as frame $\{S_0\}$. For a point s on the ith segment with constant strain ξ_i, the transformation matrix from the segment's proximal frame $\{S_{i-1}\}$ to the local frame at coordinate s is given by the exponential map $e : \mathfrak{se}(3) \mapsto SE(3)$ (Renda et al., 2018)

$$\begin{aligned} e^{(s-\bar{L}_{i-1})\check{\xi}_i} =& I_4 + (s-\bar{L}_{i-1})\,\check{\xi}_i + \left(1 - \cos((s-\bar{L}_{i-1})\,\theta_i)\right) \\ & \frac{\check{\xi}_i^2}{\theta_i^2} + \left((s-\bar{L}_{i-1})\,\theta_i - \sin((s-\bar{L}_{i-1})\theta_i)\right) \frac{\check{\xi}_i^3}{\theta_i^3}. \end{aligned} \tag{5.12}$$

where $\check{\xi}_i \in \mathfrak{se}(3)$ is the strain twist vector and $\theta_i = \sqrt{\kappa_{x,i}^2 + \kappa_{y,i}^2 + \kappa_{z,i}^2}$ is the magnitude of the rotational strain. Therefore, the fully assembled transformation $T^i_{\mathcal{B}}(q)$ from the base frame $\{S_{\mathcal{B}}\}$ to the tip frame of the ith segment can be expressed as

$$T^i_{\mathcal{B}}(q) = T^0_{\mathcal{B}}(\phi_0)\,\Pi_{j=1}^i\, e^{\bar{l}_j \check{\xi}_j}(q) \in SE(3). \tag{5.13}$$

5.3.2 Verification of SPCS Kinematics

The section is structured as follows. We introduce relevant actuation sequences for the HSA robot. Next, we present an inverse kinematic approach to identify the kinematic configuration. Translational and rotational error metrics are then defined to evaluate the quality of reconstructions. Finally, we verify the performance of the proposed SPCS kinematic model both for simulated data and on experimental datasets. The code and all datasets are made available on GitHub [3].

[3] https://github.com/tud-phi/hsa-kinematic-model

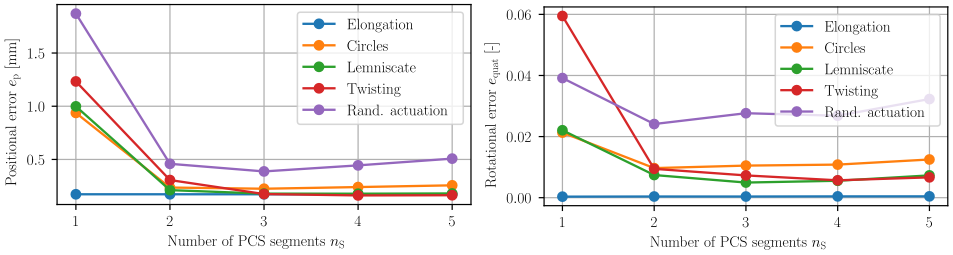


Figure 5.5: Verification of kinematic models in simulation. The plot in the first row shows the positional error e_p of the kinematic model against the simulated HSA. The plot in the second row visualizes the rotational error metric e_{quat}, which is based on the vector component of the unit quaternion. For more information on the evaluation metrics, we refer to Section 5.3.2. The kinematic model used here assumes the twist & stretch strains to be constant along the entire HSA and the bend & shear strains to be captured by n_S segments. Along this line, we report the performance of the kinematic model for a parametrization containing one to five segments.

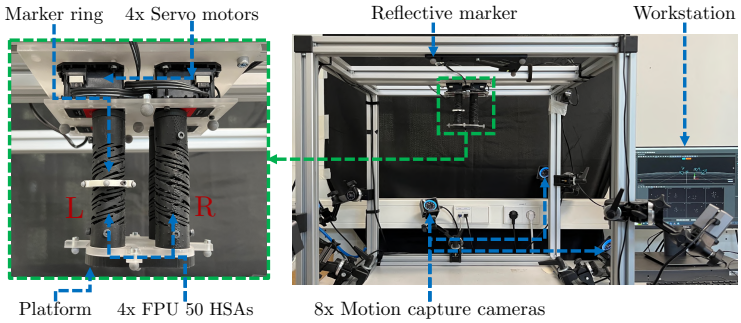


Figure 5.6: Experimental setup with the HSA robot attached in platform-down configuration to the motion capture cage. The robot contains two left-handed (L) and two right-handed (R) HSA rods, respectively. Rods of the same handedness are placed opposite of each other. The reflective markers allow us to determine the pose information of the base, an intermediate point along the left HSA rod, and the platform.

Actuation Sequences

We collect datasets with a variety of actuation sequences, which include both pure motion primitives and random actuation. For all sequences, we apply a twist angle of magnitude $|u_d| \in [0, \pi\,\text{rad}]$ at the base of each HSA. The sign of u_d is determined by the handedness h of the respective HSA. The elongation dataset consists of samples between the rest and fully-elongated HSA state. Please refer to Fig. 5.2(a) for more details on how each motion primitive can be invoked. For the bending motion primitive, we consider two separate trajectory types: a) a Lemniscate trajectory and b) a trajectory containing circles of varying bending angles. The different bending angles are achieved by varying the actuation angle from 20 % to 100 % of its maximum magnitude. For each fixed bending angle, we collect 15 samples along the circle, e.g. 15 different azimuth angles. To achieve the desired azimuth angle, we smoothly interpolate between the east, north, west, and south actuation specifications of Fig. 5.2(a). The twisting trajectory collects discrete samples between maximum clockwise (CW) and maximum counterclockwise (CCW) twisting. Finally, we collect a dataset of

randomly sampled actuation inputs, which combines the elongation, bending, and twisting motion primitives. In total, the elongation and Lemniscate trajectories contain 100 samples each, and the circles and twisting trajectory have 225 and 100 samples, respectively. 500 samples are included in the random actuation sequence.

Inverse Kinematics

Differential inverse kinematics can be used to reconstruct the rod's configuration q from N known poses $T_{\mathcal{B}}^{s_i} \in SE(3), i \in [1, N]$ along the rod. We implemented an inverse kinematics algorithm based on the analytical Jacobian $J_{\mathrm{A}}^{s_i} \in \mathbb{R}^{6\times 7}$ of the pose representation

$$\chi_{\mathcal{B}}^{s_i} = \begin{pmatrix} \varepsilon_x & \varepsilon_y & \varepsilon_z & \eta & \varepsilon_z & p_x & p_y & p_z \end{pmatrix}^\top \in \mathbb{R}^7,$$

which includes rotational orientation estimates in unit quaternion representation $\mathcal{Q} = \begin{pmatrix} \varepsilon_x & \varepsilon_y & \varepsilon_z & \eta \end{pmatrix}^\top$ and positions in Cartesian space $p = \begin{pmatrix} p_x & p_y & p_z \end{pmatrix}^\top$. Please note that usually ϕ_0 does not need to be found through (differential) inverse kinematics but can rather be directly read out from the encoders of the electric servos. All N poses and Jacobians can be vertically stacked as $\chi \in \mathbb{R}^{7N}$ and $J_{\mathrm{A}} \in \mathbb{R}^{7N\times 6}$ respectively. This then allows us to iteratively optimize the pose error $e_\chi = \chi_{\mathrm{d}} - \tilde{\chi}$ between the known pose χ_{d} and the pose $\tilde{\chi}$ computed using the forward kinematics

$$\tilde{q}_{\mathrm{it}+1} = \tilde{q}_{\mathrm{it}} + \lambda J_{\mathrm{A}}^\top(\tilde{q})\,(\chi_{\mathrm{d}} - \tilde{\chi}(\tilde{q})), \tag{5.14}$$

where $\tilde{q}$ is the current configuration estimate, and λ is the step size.

Evaluation Metrics

We briefly introduce the metrics to quantify shape reconstruction accuracy by the proposed kinematic parametrization. We first define a RMSE for comparing each ground-truth position $p_t^i \in \mathbb{R}^3, t \in \{1, \dots, n_t\}, i \in \{1, \dots, N\}$ to the position estimated by the kinematic model $\tilde{p}_t^i$ over a time period of n_t steps

$$e_{\mathrm{p}} = \sqrt{\sum_{t=1}^{n_{\mathrm{t}}} \sum_{i=1}^{N} \frac{\left(\left\|\tilde{p}_t^i - p_t^i\right\|_2\right)^2}{n_{\mathrm{t}} N}} \in \mathbb{R}. \tag{5.15}$$

The rotational RMSEs e_{quat} is computed analogue by substituting p in (5.15) with the quaternion vector component $\varepsilon = \begin{pmatrix} \varepsilon_x & \varepsilon_y & \varepsilon_z \end{pmatrix}^\top$. Finally, we compute the XYZ Euler angle error as

$$e_{\mathrm{eul}} = \sqrt{\sum_{t=1}^{n_{\mathrm{t}}} \sum_{i=1}^{N} \frac{\left(f_\vartheta(R_{t,i} \tilde{R}_{t,i}^\top)\right)^2}{n_{\mathrm{t}} N}} \in \mathbb{R}^3, \tag{5.16}$$

where $f_\vartheta(\cdot)$ is the operator to compute the XYZ Euler angles $\vartheta = \begin{pmatrix} \alpha & \beta & \gamma \end{pmatrix}^\top$ from a rotation matrix $R \in SO(3)$.

Simulation Results

We employ the higher-dimensional HSA robot simulator proposed in Section 5.2 to generate steady-state HSA states. We use the same simulation parameters as in Section 5.2.5.

Table 5.2: Experimental verification of kinematic models on an HSA robot. Motion capture markers attached to one of the HSAs provide two ground-truth poses along the rod, and we measure ϕ_0 from the servo readings (13 constraints in total). In the spirit of an ablation study, we investigate different variations of the proposed kinematic parametrization. For Constant Curvature (CC), Constant Twist (CT), Constant Shear (CSH), and Constant Axial (CA) strain, the strain is kept constant along the entire HSA. For Piecewise Constant Curvature (PCC), Piecewise Constant Twist (PCT), Piecewise Constant Shear (PCSH), and Piecewise Constant Axial (PCA) strain, the strain components are parameterized separately for each of the two segments. Additionally, we test the importance of including the shear strain component. For each kinematic parametrization, we state the DOFs. We report RMSEs for both translations and rotations (see Sec. 5.3.2).

Trajectory	**CC**	**CT**	**CSH**	**CA**	**PCC**	**PCT**	**PCSH**	**PCA**	**DOF**	e_p [mm]	e_quat [-]	$e_{\mathrm{eul},\alpha}$ [rad]	$e_{\mathrm{eul},\beta}$ [rad]	$e_{\mathrm{eul},\gamma}$ [rad]
Elongation	✗	✓	✗	✓	✓	✗	✗	✗	7	1.010	0.0092	0.0029	0.0079	0.0166
Elongation	✗	✓	✗	✓	✓	✗	✓	✗	11	0.126	0.0082	0.0020	0.0031	0.0166
Elongation	✗	✗	✗	✗	✓	✓	✓	✓	13	0.009	0.0042	0.0020	0.0031	0.0070
Circles	✗	✓	✗	✓	✓	✗	✗	✗	7	1.744	0.0136	0.0108	0.0123	0.0228
Circles	✗	✓	✗	✓	✓	✗	✓	✗	11	0.227	0.0082	0.0110	0.0122	0.0227
Circles	✗	✗	✗	✗	✓	✓	✓	✓	13	0.092	0.0093	0.0116	0.0125	0.0075
Lemniscate	✗	✓	✗	✓	✓	✗	✗	✗	7	1.227	0.0098	0.0042	0.0051	0.0195
Lemniscate	✗	✓	✗	✓	✓	✗	✓	✗	11	0.215	0.0082	0.0045	0.0053	0.0195
Lemniscate	✗	✗	✗	✗	✓	✓	✓	✓	13	0.023	0.0052	0.0043	0.0054	0.0076
Twisting	✗	✓	✗	✓	✓	✗	✗	✗	7	2.931	0.0136	0.0121	0.0186	0.0195
Twisting	✗	✓	✗	✓	✓	✗	✓	✗	11	0.263	0.0141	0.0130	0.0196	0.0194
Twisting	✗	✗	✗	✗	✓	✓	✓	✓	13	0.030	0.0127	0.0131	0.0217	0.0017
Rand. actuation	✗	✓	✗	✓	✓	✗	✗	✗	7	4.345	0.0381	0.0678	0.0544	0.0195
Rand. actuation	✗	✓	✗	✓	✓	✗	✓	✗	11	0.365	0.0383	0.0681	0.0544	0.0193
Rand. actuation	✗	✗	✗	✗	✓	✓	✓	✓	13	0.255	0.0380	0.0700	0.0527	0.0200

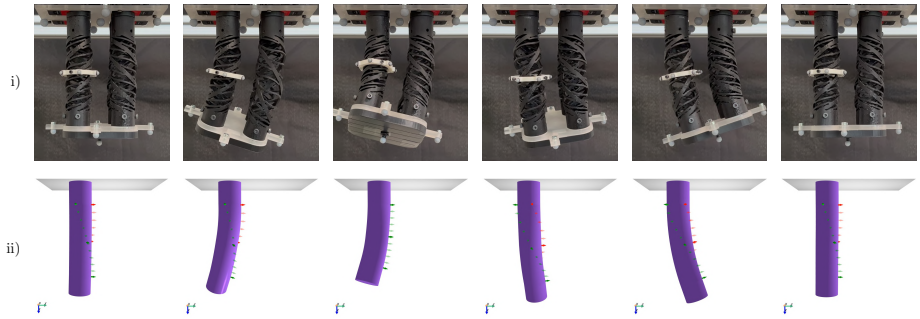
This provides us with 25 discrete poses along each of the four HSAs. Then, we perform differential inverse kinematics with a step size of $\lambda = 0.2$ to find an optimal configuration q describing the shape of the HSA. We choose a higher step size ($\lambda = 1$) for regressing the twist strains.

For the kinematic model, we assume that the twist & stretch strains are constant along the entire HSA rod. The bend & shear strains, on the other hand, are instead piece-wise constant across n_S segments. We evaluate the influence of the n_S parameter and test the performance of a kinematic model involving between 1 and 5 PCS segments.

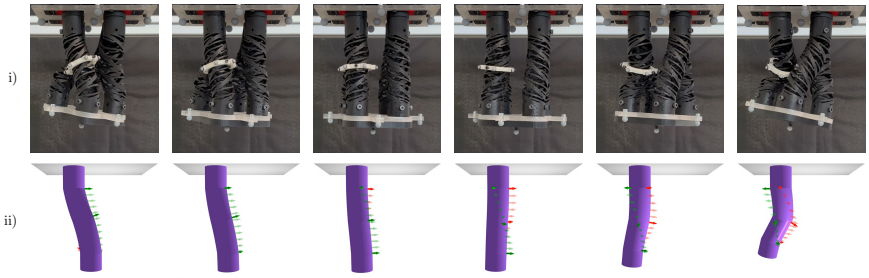
The results are in Fig. 5.5. While a kinematic model with a single CS segment still works sufficiently well for the elongation and bending motion primitives, its performance deteriorates for any trajectories involving twisting. Instead, two segments of our model are sufficient to accurately represent the shape of the HSA.

Experimental Results

In addition to the simulations, we also experimentally verify the kinematic model using an HSA robot consisting of four closed rods 3D-printed via digital projection lithography from the flexible photopolymer resin Carbon FPU 50 (Truby et al., 2021). Each HSA rod was printed to a length of $\bar{L} = 101.6\,\mathrm{mm}$ and is independently actuated by DYNAMIXEL MX-28T servo motors. As seen in Fig. 5.6, we attach motion capture markers to several points on the robot to track the ground-truth pose information. Namely, we measure the pose of the motor base, the platform, and the midpoint of one of the right-handed HSA rods (i.e., the front-left HSA on the picture). Please note that we extract the rotation angle ϕ_0 from the servo encoders directly. The robot is mounted at its base to a cubical cage of



(a) Lemniscate trajectory



(b) Twisting trajectory

Figure 5.7: Sequence of stills for a Lemniscate and a twisting trajectory. **Top row i):** frames of a video recording of the HSA robot during the experiments. The shape of the front-left HSA rod is fitted using inverse kinematics and rendered in ii). **Bottom row ii):** Rendered shape of the HSA rod produced by evaluating the forward kinematics along the backbone length. The arrows with full opacity denote the ground-truth pose of three points along the HSA rod as measured by the motion capture system. The red arrow points along the local x-axis, and the green arrow along the local y-axis, respectively. The arrows with slight transparency represent poses along the backbone computed with the forward kinematics. We assume the last 25 mm and 20 mm at the proximal and distal end of the rod, respectively, to be rigid and therefore do not include them in the kinematic parametrization.

side length 750 mm in platform-down configuration. Eight Optitrack Prime X 13 cameras are attached to the cage, tracking the reflective markers at 30 Hz.

We actuate the robot from a workstation next to it, with the control loop running at 10 Hz. The control loop communicates motor position setpoints $u_\mathrm{d} \in \mathbb{R}^4$ to the servos. The inner control loop of the servos then applies the appropriate torques to guide the motors toward the desired position. As soon as the motors have reached their goal position, we wait for 2 s to reach steady-state and then read out the pose measurements.

In Fig. 5.7, we show sequences of stills for the Lemniscate and twisting trajectory. The kinematic model used here assumes a constant twist strain along the entire rod and employs two PCS segments to capture the remaining five strains. We see that, except for extreme twisting states, e.g., the far right image in Fig. 5.7(b), the kinematic model is able to represent the complex HSA shape very well.

In Tab. 5.2, we quantitatively evaluate multiple kinematic models on the trajectories defined in 5.3.2. The first (7 DOFs) and second (11 DOFs) kinematic models are very similar,

as both assume constant twist and constant stretch along the entire HSA. The other strains are contained in two PCS segments in both cases. However, the first model exhibits much larger positional errors as it neglects shear strains, which are very important in HSA robots but were not accounted for in the literature (Garg et al., 2022). The third model provides the upper bound on the performance, as it has with 13 comparatively many DOFs and uses a piecewise formulation with two segments for all segments.

5.4 Kinematic and Dynamic Modeling of Planar HSA Robots

In this section, we present a control-oriented, compact kinematic and dynamic model for planar HSA robots. We first introduce the kinematic model of the robot and then derive the dynamic model in Euler-Lagrangian form. We then verify the predictive capabilities of the dynamic model on unseen trajectories.

5.4.1 Kinematic Model

Following the discrete Cosserat approach (Renda et al., 2018), we characterize the configuration space of the virtual backbone by assuming a CS model ${}_{\mathcal{V}}\xi(t) = \begin{bmatrix} {}_{\mathcal{V}}\kappa_{\text{be}} & {}_{\mathcal{V}}\kappa_{\text{sh}} & {}_{\mathcal{V}}\sigma_{\text{ax}} \end{bmatrix}^\top = \mathbb{I}_3\, q(t) \in \mathbb{R}^3$, where κ_{be}, σ_{sh}, and σ_{ax} denote the bending, shear, and axial strain respectively. Given q, the pose $\chi = \begin{bmatrix} p_x & p_y & \theta \end{bmatrix}^\top \in SE(2)$, and a point coordinate along the backbone $s \in [0, l^0]$, the forward and inverse kinematics are provided in closed form as

$$\chi = \pi(q,s) = \begin{bmatrix} \sigma_{\text{sh}} \frac{\text{s}_{\text{be}}}{\kappa_{\text{be}}} + \sigma_{\text{ax}} \frac{\text{c}_{\text{be}}-1}{\kappa_{\text{be}}} \\ \sigma_{\text{sh}} \frac{1-\text{c}_{\text{be}}}{\kappa_{\text{be}}} + \sigma_{\text{ax}} \frac{\text{s}_{\text{be}}}{\kappa_{\text{be}}} \\ \kappa_{\text{be}}\, s \end{bmatrix}, \qquad q = \varrho(\chi, s) = \frac{\theta}{2s} \begin{bmatrix} 2 \\ p_y - \frac{p_x\, \text{s}_\theta}{\text{c}_\theta - 1} \\ -p_x - \frac{p_y\, \text{s}_\theta}{\text{c}_\theta - 1} \end{bmatrix}, \tag{5.17}$$

where we use the shorthand notations $\text{s}_{\text{be}} = \sin(\kappa_{\text{be}} s)$, $\text{c}_{\text{be}} = \cos(\kappa_{\text{be}} s)$, $\text{s}_\theta = \sin(\theta)$, and $\text{c}_\theta = \cos(\theta)$. Furthermore, the forward kinematics of the physical rods $\mathcal{P}_i$, $i \in \{1,2\}$ can be derived by first following the transformations of the virtual backbone and then adding a local translation $[\pm r_{\text{off}}, 0]^\top$ with r_{off} being the offset distance from the virtual backbone to the centerline of the HSA rod. After closing the kinematic chain, we identify a mapping $\beta_i : {}_{\mathcal{V}}\xi \to {}_{\mathcal{P}_i}\xi$ from the strains of the virtual backbone to the strains in the physical rods: $\beta_i({}_{\mathcal{V}}\xi) = \begin{bmatrix} {}_{\mathcal{V}}\kappa_{\text{be}}, & {}_{\mathcal{V}}\sigma_{\text{sh}}, & {}_{\mathcal{V}}\sigma_{\text{ax}} \pm r_{\text{off}}\, {}_{\mathcal{V}}\kappa_{\text{be}} \end{bmatrix}^\top$. Analog to the dynamic simulator in Sec. 5.2, we model the auxetic trajectory of the HSAs by coupling the rest length $\tilde{l}_i$ to the twist strain $\kappa_{\text{tw},i}$ of the ith HSA rod: $\tilde{l}_i = (1+\epsilon_i) l^0 = (1 + h_i C_\epsilon \kappa_{\text{tw},i})$ where l^0 is the printed length of the rod and C_ϵ a positive constant. The handedness $h_i \in \{-1, 1\}$ describes if positive or negative twist angles are needed to elongate the closed HSA. For a given vector of rod twist angles $\phi \in \mathbb{R}^2$ and after defining $\phi_i^+ = h_i \phi_i$, the elongation of the ith rod is then $\epsilon_i = C_\epsilon \frac{\phi_i^+}{l^0}$.

5.4.2 Dynamic Model

We aim to devise a dynamic model in the Euler-Lagrange form $M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + K(q - q^0) + D\dot{q} = \alpha(q,\phi)$, where $M(q), C(q,\dot{q}), K, D \in \mathbb{R}^{3\times 3}$ are the inertia, Coriolis (derived with Christoffel symbols), elastic, and damping matrices, respectively. $q^0 \in \mathbb{R}^3$ captures the rest configuration. The terms $G(q)$ and $\alpha(q,\phi) \in \mathbb{R}^3$ describe the gravitational and

actuation forces acting on the generalized coordinates. The state of the robot at time t can be therefore described by $x(t) = \begin{bmatrix} q^\top(t) & \dot{q}^\top \end{bmatrix}^\top \in \mathbb{R}^6$. The inertia matrix is found by following the standard procedure of integrating mass and rotational inertia along the HSA rods (Della Santina et al., 2023). Additionally, we consider the inertial contribution of the platform mounted to the distal end of the robot. Under the small strain assumption, the elastic forces of the ith HSA rod can be modeled as

$$ {}_{\mathcal{P}}\tau_{\mathrm{K},i} = \begin{bmatrix} S_{\mathrm{be},i}(\phi_i) & S_{\mathrm{be,sh}} & 0 \\ S_{\mathrm{be,sh}} & S_{\mathrm{sh},i}(\phi_i) & 0 \\ 0 & 0 & S_{\mathrm{ax},i}(\phi_i) \end{bmatrix} \left(\begin{bmatrix} {}_{\mathcal{P}_i}\kappa_{\mathrm{be}} \\ {}_{\mathcal{P}_i}\sigma_{\mathrm{sh}} \\ {}_{\mathcal{P}_i}\sigma_{\mathrm{ax}} \end{bmatrix} - \begin{bmatrix} \kappa^0_{\mathrm{be}} \\ \sigma^0_{\mathrm{sh}} \\ \sigma^0_{\mathrm{ax}} + \epsilon_i(\phi_i) \end{bmatrix} \right), \tag{5.18} $$

where ${}_{\mathcal{P}_i}\xi^0 = \begin{bmatrix} \kappa^0_{\mathrm{be}} & \sigma^0_{\mathrm{sh}} & \sigma^0_{\mathrm{ax}} \end{bmatrix}^\top$ denotes the rest strain, $S_{\mathrm{be},i}(\phi_i)$, $S_{\mathrm{sh},i}(\phi_i)$, $S_{\mathrm{ax},i}(\phi_i)$ are the bending, shear, and axial stiffnesses which are defined as linear functions with respect to the twist angle of the rod ϕ_i (Good et al., 2022; Stölzle et al., 2023):

$$ S_{\mathrm{be},i}(\phi_i) = \hat{S}_{\mathrm{be}} + C_{S_{\mathrm{be}}}\,\phi_i^+, \quad S_{\mathrm{sh},i}(\phi_i) = \hat{S}_{\mathrm{sh}} + C_{S_{\mathrm{sh}}}\,\phi_i^+, \quad S_{\mathrm{ax},i}(\phi_i) = \hat{S}_{\mathrm{ax}} + C_{S_{\mathrm{ax}}}\,\phi_i^+. \tag{5.19} $$

The coefficient $S_{\mathrm{be,sh}}$ accounts for the elastic coupling between the bending and the shear strain. Subsequently, we project the forces into the virtual backbone by premultiplying with $J_\beta^\top = \frac{\partial \beta}{\partial q}^\top$ and then sum the contribution of all rods. Finally, we group all terms depending on the control input ϕ in $\alpha(q, \phi)$ and everything else in K. After modeling the dissipative forces in each HSA as $\mathrm{diag}(\zeta_{\mathrm{be}}, \zeta_{\mathrm{sh}}, \zeta_{\mathrm{ax}})\,{}_{\mathcal{P}_i}\dot{\xi}$, we derive the damping matrix in configuration space as $D = \sum_{i=1}^{2} J_{\beta,i}^\top \mathrm{diag}(\zeta_{\mathrm{be}}, \zeta_{\mathrm{sh}}, \zeta_{\mathrm{ax}}) J_{\beta,i} = 2\,\mathrm{diag}\left((\zeta_{\mathrm{be}} + r_{\mathrm{off}}^2\,\zeta_{\mathrm{ax}}), \zeta_{\mathrm{sh}}, \zeta_{\mathrm{ax}}\right)$. We open-source the derivation of the Euler-Lagrangian dynamics and a JAX implementation of a simulator based on them on GitHub[4].

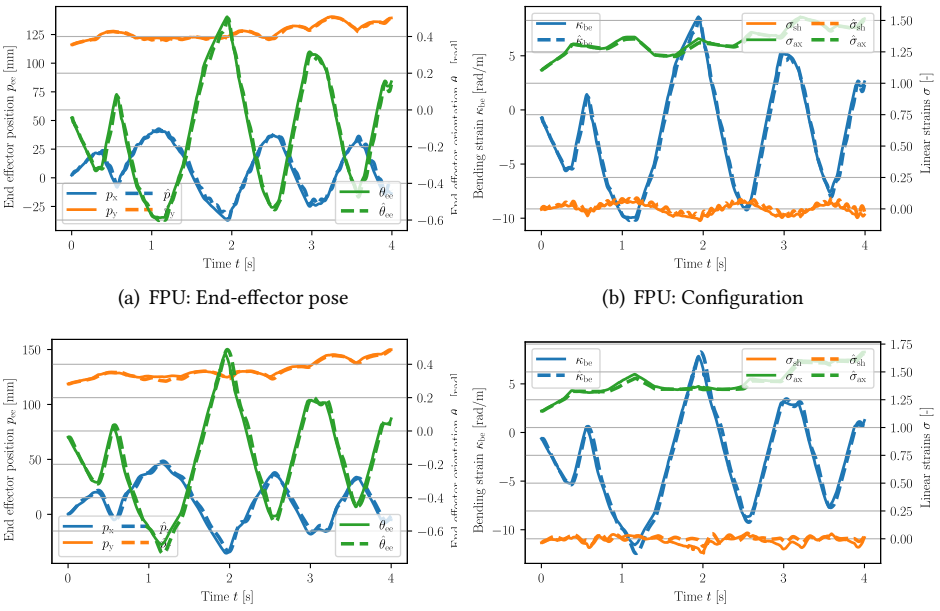
5.4.3 Model Verification

Experimental Setup

We evaluate the planar HSA model experimentally. The material choice of the HSA is crucial and has a significant influence on the resulting mechanical characteristics of the robot (e.g., blocked force, holding torque, bending stiffness, etc.) (Truby et al., 2021). Furthermore, specific material requirements are dictated by the nature of the design of the HSA rod. The structure of the metamaterial is made of struts connected by living hinges. These living hinges must be thin, flexible, and accommodate high strains (Truby et al., 2021). In order to ensure that the system model is effective and suitable for a range of different materials, we decided to conduct the experimental verification on HSAs 3D-printed via digital projection lithography either using the photopolymer resin Carbon FPU 50 (Car, 2020b) (stiffer) or the elastomeric polyurethane EPU 40 resin (Car, 2020a) (softer).

The Dynamixel MX-28 servo motors are set to use position control mode. The robot is mounted platform-down on a cage with an Optitrack motion capture system, which measures the SE(3) pose of the platform at 200 Hz. Our algorithms run within a ROS2 framework[5]. The pose measurements are first projected into the plane of actuation and serve as an input to the closed-form inverse kinematics introduced in (5.17).

[4] https://github.com/tud-phi/jax-soft-robot-modeling
[5] https://github.com/tud-phi/ros2-hsa



(a) FPU: End-effector pose

(b) FPU: Configuration

(c) EPU: End-effector pose

(d) EPU: Configuration

Figure 5.8: Verification of the system model and the identified system parameters on an unseen trajectory with the HSA being randomly actuated through a GBN sequence: the solid line denotes the actual trajectory. In contrast, the dashed line visualizes the trajectory simulated with the system model. We report results for both FPU and EPU-based HSAs.

System Identification

Next, we strive to identify the parameters used in our dynamic model. We assume the robot's geometric and mass density properties to be known or easily measurable. As knowledge about the damping coefficients is not required by the control law, only the experimental identification of elongation and stiffness characteristics remains. For this, we measure the response of the system to step and staircase actuation sequences. Afterward, the parameters are regressed using least squares. For the FPU-based robot, we identify $C_\varepsilon^{\text{FPU}} = 0.0098\,\text{m/rad}$, $S_{\text{be}}^{\text{FPU}} = 0.00057 \cdot 10^{-5} - 9.7 \cdot 10^{-6}\,\frac{\phi_i^+}{l^0}\text{Nm}^2$, $S_{\text{sh}}^{\text{FPU}} = 0.591 - 0.00048\,\frac{\phi_i^+}{l^0}\text{N}$, $S_{\text{ax}}^{\text{FPU}} = 5.665 + 0.0151\,\frac{\phi_i^+}{l^0}\text{N}$, and $S_{\text{b,sh}}^{\text{FPU}} = 4.48 \cdot 10^{-3}\text{Nm/rad}$ where $l^0 = 0.059\,\text{m}$. Furthermore, we regress $C_\varepsilon^{\text{EPU}} = 0.0079\,\text{m/rad}$, $S_{\text{be}}^{\text{EPU}} = -2.5 \cdot 10^{-5} + 3.9 \cdot 10^{-7}\,\frac{\phi_i^+}{l^0}\text{Nm}^2$, $S_{\text{sh}}^{\text{EPU}} = 0.0428 - 0.0029\,\frac{\phi_i^+}{l^0}\text{N}$, $S_{\text{ax}}^{\text{EPU}} = 0.0 + 0.0098\,\frac{\phi_i^+}{l^0}\text{N}$, and $S_{\text{b,sh}}^{\text{EPU}} = -0.0005\,\text{Nm/rad}$ for the EPU HSAs which have the same length as the FPU HSAs. Finally, we identify the axial rest strain σ_{ax}^0 before the start of each experiment. We notice that the EPU-based HSA robot is approximately one order of magnitude more flexible than the FPU-based robot.

Results

We verify the accuracy of the proposed system model and the identified parameters on trajectories unseen during system identification. We generate the trajectories by actuating

the robot with a Generalized Binary Noise (GBN) (Tulleken, 1990) sequence with a settling time of 0.5 s and at each time step k randomly sample $\phi(k) \sim \mathcal{U}(0, \phi_{\max})$. We simulate the model evolution with a Dormand-Prince 5(4) integrator and a time step of 0.1 ms. Fig. 5.8(a) shows the model exhibiting excellent accuracy for representing the behavior of FPU-based HSA robots. We observe more significant errors in the shear estimate for EPU-based HSA robots in Fig.5.8(d). Specifically, the CS model no longer seems sufficient for capturing the robot's shape, particularly for larger bending angles. Therefore, we suggest for future work to employ kinematic models with more DOF such as PCS as proposed, for example, in Sec. 5.3 or (Renda et al., 2018).

5.5 Conclusion

This chapter provided solutions for the first time for modeling the kinematics and the dynamics of electrically-actuated continuum soft robots based on Handed Shearing Auxetics. We have shown that coupling the twist strains to rest lengths can allow simulators based on the discrete Cosserat rod theory. While the proposed linear approximation of the auxetic trajectory works well for closed HSAs within a bounded motion range, future work shall derive a more general model also applicable for semi-closed and open HSAs (Good et al., 2022). Furthermore, we have proposed the SPCS kinematic model that can express the shape of HSAs with 11 DOFs. Fitting this kinematic model to the experimental results showed a very good match for representing the shape of the HSAs. In particular, for large actuation magnitudes within the twisting motion primitive, the HSAs leave the auxetic trajectory and seem to experience buckling behavior. For this case, the SPCS model is not accurate anymore. Finally, we presented a control-oriented dynamic model for planar HSA robots and verified it experimentally. The conducted experiments gave us deep insights into the particular characteristics of HSAs and how well our model is able to capture them. We see excellent agreement for predicting the dynamical behavior of HSA robots made of FPU material. We observed the time lag of the model to be larger for EPU-based robots, as seen in Fig. 5.8(c). This probably is the result of the hysteresis characteristics of HSAs (Good et al., 2022). For EPU-based HSAs robots, we observe that the model does not fully capture the shear dynamics.

Afterword

This chapter presented approaches for representing the kinematics and dynamics of HSA rods in the general case. Additionally, we introduced the first low-dimensional kinematic parameterizations and control-oriented dynamical models for planar HSA robots. Next, Chapter 6 will focus on leveraging the models proposed in this chapter for model-based control of planar HSA robots. Namely, this includes the development of stable and effective configuration-space integral-saturated PID+potential-shaping and operational space impedance controllers. Subsequently, Chapter 7 will develop a BMI strategy for guiding soft robots, and specifically HSA robots, using motor imagery in a safe and compliant fashion.

6

Model-Based Control of Handed Shearing Auxetics (HSA) Robots

Foreword. In Chapter 6, we developed kinematic and dynamical models for planar HSA robots. However, the control of HSA remains an unexplored area, both in our previous chapters and in the existing literature. We present in this chapter various model-based control approaches for planar HSA robots, ranging from PID (with integral saturation) + potential-shaping configuration-space control (Section 6.2) to Cartesian-space impedance control (Section 6.3). Notably, we rigorously validate and benchmark the proposed control strategies experimentally on two different HSA robot prototypes.

Abstract. *The control of robots based on Handed Shearing Auxetics (HSAs) is especially challenging due to varying and coupled stiffness, shearing, non-affine terms in the actuation model, and underactuation. In this chapter, we present two model-based control strategies for planar HSA robots enabling regulation in operational space. Firstly, we propose a control strategy composed of steady-state planning for identifying a statically admittable robot shape with matching actuation and a P-satI-D feedback controller compensating elastic and gravitational forces in configuration space. Secondly, we derive an operational space impedance controller that allows us to unite the soft robot's embodied intelligence with computational intelligence to guarantee compliance and interaction safety. We experimentally verify both proposed control strategies in closed loop.*

This chapter is partly based on

- ***M. Stölzle**, D. Rus, and C. Della Santina (2023, December). An Experimental Study of Model-based Control for Planar Handed Shearing Auxetics Robots. In Experimental Robotics: The 18th International Symposium. Springer* (Stölzle et al., 2024b).
- ***M. Stölzle***, S. S. Baberwal*, D. Rus, S. Coyle, and C. Della Santina (2024). Guiding Soft Robots with Motor-Imagery Brain Signals and Impedance Control. In Proceedings of the 2024 IEEE 7th International Conference on Soft Robotics (RoboSoft) (pp. 1-8). IEEE. Received the **Best Paper Award*** (Stölzle et al., 2024a).

M.S. and C.D.S. conceived the project. M.S. and C.D.S. devised both control approaches. L.C., R.L.T., and D.R. designed and fabricated the HSA robot. M.S. implemented the model, planned and executed the simulations and the experiments, performed the data analysis, and wrote the manuscript. S.S.B. and S.C. were not directly involved with the control aspects presented in this chapter, but instead contributed to the BMI presented in Chapter 7. C.D.S. supervised the project and revised the manuscript. C.D.S. and D.R. provided funding.

6.1 Introduction

Recent work has investigated proprioception (Zhang et al., 2022a), the mechanical characterization (Good et al., 2022), simulation (Stölzle et al., 2023), and kinematic modeling (Garg et al., 2022; Stölzle et al., 2023) of HSA robots but control has yet to be tackled. In this work, we make a first step towards achieving task space control by designing model-based regulators for planar motions. Our approach considers essential characteristics of HSA robots, such as underactuation, shear strains, and varying stiffness.

In Chapter 5, we derived a dynamic model for planar HSA in Euler-Lagrange form and experimentally verified it. We notice that the resulting planar dynamics are underactuated and that the actuation forces are non-affine with respect to the control inputs, which are the motor angles. The latter is a peculiarity of these systems, rarely observed in other robots. Based on the model knowledge, we propose in this chapter two control strategies for planar HSA robots capable of regulating the end-effector towards a desired position in task space. The first strategy, as shown in Fig. 6.1(a), performs steady-state planning to identify an admittable configuration and steady-state control input matching the desired end-effector position and then subsequently applies a P-satI-D feedback controller (Pustina et al., 2022a) on the collocated form (Pustina et al., 2024a) of the system dynamics. The second strategy, as shown in Fig. 6.13, directly regulates the end-effector position using a Cartesian impedance controller that fully preserves the softness of the robot.

In summary, we state our contributions as (i) a provably stable model-based control strategy for guiding the end-effector of the robot towards a desired position in Cartesian space with a configuration-space controller that combines an integral-saturated PID with a potential shaping feedforward term, (ii) a Cartesian impedance controller that allows combining the passive compliance of the HSA robot with active compliance in the control strategy and (iii) extensive experimental verification of both control strategies. A video accompanies this chapter explaining the methodology and displaying video recordings of the control experiments[1].

6.2 Configuration Space Regulation

In this section, we derive a model-based control strategy in configuration space for achieving setpoint regulation that combines an integral-saturated PID controller with a potential-shaping feedforward term. We first introduce the dynamical model of the planar HSA robot and then present the control strategy. Subsequently, we present a steady-state planning procedure to identify admittable configurations and matching steady-state actuation. Finally, we experimentally verify the proposed control strategy in closed loop.

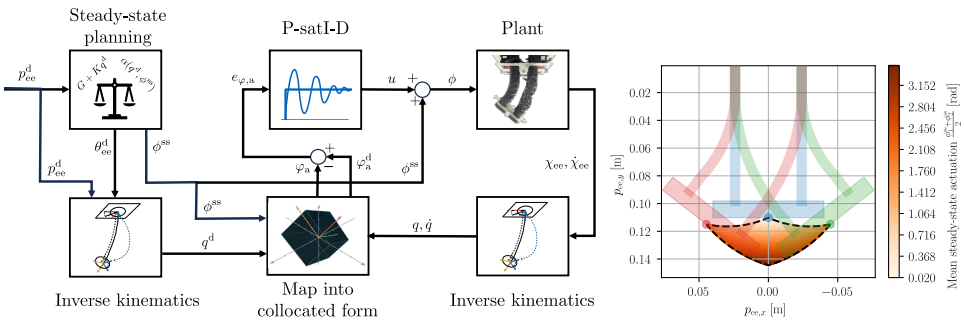
6.2.1 Background: Dynamical Model

As introduced in Sec. 5.4, the state of a planar HSA robot at time t can be therefore described by $x(t) = \begin{bmatrix} q^\top(t) & \dot{q}^\top \end{bmatrix}^\top \in \mathbb{R}^6$, where κ_{be}, σ_{sh}, and σ_{ax} denote the bending, shear, and axial strain respectively. The dynamical model is then given in Euler-Lagrange form as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + K(q - q^0) + D\dot{q} = \alpha(q,\phi), \tag{6.1}$$

[1] https://youtu.be/7PgKnE_MOsY

where $M(q), C(q,\dot{q}), K, D \in \mathbb{R}^{3\times3}$ are the inertia, Coriolis, elastic and damping matrices, respectively. $q^0 \in \mathbb{R}^3$ captures the rest configuration. The terms $G(q)$ and $\alpha(q,\phi) \in \mathbb{R}^3$ describe the gravitational and actuation forces acting on the generalized coordinates. We provide examples in Fig. 6.1(b) of the operational workspace that can be achieved with this kinematic model. We stress that (a) the derived dynamical model is not affine in the control input and (b) the system is underactuated.



(a) Blockscheme (b) Operational workspace

Figure 6.1: **Panel (a):** Block scheme for configuration-space regulator: we plan the steady-state behavior such that the end-effector matches the given desired position p_{ee}^{d}. The outputs of this planning are the steady-state actuation ϕ^{ss} and a suitable end-effector orientation θ_{ee}^{d}. After leveraging inverse kinematics to identify the desired and current configuration, q is mapped into a collocated form where the inputs are decoupled. Finally, we use a P-satI-D feedback controller on the actuation coordinates φ. **Panel (b):** Visualization of the operational workspace of a planar HSA robot consisting of FPU rods. The colored area within the black dashed borders represents the positions the end-effector (visualized as a dot) can reach. The coloring denotes the mean magnitude of actuation (i.e., twisting of the rods). Furthermore, we plot three sample configurations: the unactuated straight configuration $q = [0,0,0]^\top$ (blue), maximum clockwise bending $q = [-11.2\,\text{rad/m}, 0.08, 0.30]^\top$ (red), and maximum counter-clockwise bending $q = [11.2\,\text{rad/m}, -0.08, 0.30]^\top$ (green).

6

6.2.2 Control Strategy

Our goal is to control the end-effector, which is defined as the distal surface of the platform, to a desired position in Cartesian space $p_{ee}^{d} \in \mathbb{R}^2$. However, the mapping into configuration space is not trivial as we do not know which end-effector orientation θ_{ee} is feasible at steady-state. To tackle this challenge, we perform steady-state planning identifying admittable configurations q^{d} and matching steady-state actuations ϕ^{ss}, which allow the robot's end-effector to statically remain at p_{ee}^{d}. More details on the used planning procedure can be found in Section 6.2.3.

In principle, we can command $\phi = \phi^{ss}$ to achieve regulation towards the desired end-effector position. Nevertheless, we add a feedback controller to compensate for any errors in ϕ^{ss} caused by unmodelled effects such as hysteresis. Unfortunately, as illustrated in Fig. 6.2, the non-affine actuation $\alpha(q,\phi)$ would complicate the design of such a feedback controller. Therefore, we perform a first-order Taylor expansion of the actuation forces with respect to ϕ resulting in a configuration-dependent actuation matrix $A_{\phi^{ss}}(q) = \frac{\partial \alpha}{\partial \phi}|_{\phi=\phi_{ss}} \in \mathbb{R}^{3\times2}$. This allows us to re-write the right side of the EOM as $\tau_q = \alpha(q^{ss}, \phi^{ss}) + A_{\phi^{ss}}(q)\,u$ where $u = \phi - \phi^{ss}$ is the new control input. To improve the robustness of the control loop, we compute u with a P-satI-D control law (Pustina et al., 2022a). However, our

system is underactuated and in a non-collocated form. Therefore, we apply a coordinate transformation $h : q \to \varphi \in \mathbb{R}^3$ recently introduced by Pustina et al. (2024a) which maps the EOM into a form where ϕ applies direct forces on the actuated configuration variables. The map is given by $h(q) = \left[\int_0^t \dot{q}^\top A_{\phi^{\mathrm{ss}}}(q)\mathrm{d}\tau, \quad \sigma_{\mathrm{sh}}\right]^\top = \left[h_1(q), \quad h_2(q), \quad \sigma_{\mathrm{sh}}\right]^\top$ with

$$h_i(q) = C_{\mathrm{S,ax}} \frac{h_i}{l^0}\left[2\,\varepsilon_i(\phi_i^{\mathrm{ss}})(\pm r_{\mathrm{off}}\kappa_{\mathrm{be}} + \sigma_{\mathrm{ax}}) \mp r_{\mathrm{off}}^2 \frac{\kappa_{\mathrm{be}}^2}{2} \pm r_{\mathrm{off}}\,\sigma_{\mathrm{ax}}^0\,\kappa_{\mathrm{be}} \mp r_{\mathrm{off}}\,\kappa_{\mathrm{be}}\,\sigma_{\mathrm{ax}} + \sigma_{\mathrm{ax}}^0\,\sigma_{\mathrm{ax}} - \frac{\sigma_{\mathrm{ax}}^2}{2}\right] + C_{\mathrm{S,b}}\frac{h_i}{l^0}\left[\kappa_{\mathrm{be}}^0\,\kappa_{\mathrm{be}} - \frac{\kappa_{\mathrm{be}}^2}{2}\right] + C_{\mathrm{S,sh}}\frac{h_i}{l^0}\left[\sigma_{\mathrm{sh}}^0\,\sigma_{\mathrm{sh}} - \frac{\sigma_{\mathrm{sh}}^2}{2}\right] + \hat{S}_{\mathrm{ax}}\frac{h_i}{l^0}\,C_\varepsilon\left[\pm r_{\mathrm{off}}\,\kappa_{\mathrm{be}} + \sigma_{\mathrm{ax}}\right]. \quad (6.2)$$

The Jacobian $J_{\mathrm{h}}(q) = \frac{\partial h}{\partial q}$ is used to formulate the dynamics $M_\varphi\ddot{\varphi} + \eta_\varphi(\varphi,\dot{\varphi})\,\dot{q} + G_\varphi + K_\varphi + D_\varphi\,\dot{\varphi} = J_{\mathrm{h}}^{-\mathrm{T}}(q)\,\alpha(q^{\mathrm{ss}},\phi^{\mathrm{ss}}) + A_\varphi\,u$ in the collocated variables (Khatib, 1987), where $A_\varphi^\top = \left[\mathbb{I}^2 \quad 0^{2\times 1}\right]^\top$. In the following, we will denote with the subscript a the first two actuated coordinates φ_{a}. Finally, the full control law of the *P-satI-D* is given in collocated form as

$$\phi = \phi^{\mathrm{ss}} + K_{\mathrm{p}}(\varphi_{\mathrm{a}}^{\mathrm{d}} - \varphi) - K_{\mathrm{d}}\dot{\varphi}_{\mathrm{a}} + K_{\mathrm{i}}\int_0^t \tanh(\gamma\,(\varphi_{\mathrm{a},t'}^{\mathrm{d}} - \varphi_{\mathrm{a},t'}))\,\mathrm{d}t', \quad (6.3)$$

where $K_{\mathrm{p}}, K_{\mathrm{d}}, K_{\mathrm{i}} \in \mathbb{R}^{2\times 2}$ are the proportional, derivative, and integral gains respectively, and $\gamma \in \mathbb{R}^{2\times 2}$ horizontally compresses the hyperbolic tangent. While the proposed P-satI-D control law compensates gravity through ϕ^{ss}, we can extend the approach to include gravity cancellation (*P-satI-D + GC*) by evaluating $G_{\varphi,\mathrm{a}}$ at the current configuration:

$$\phi = \phi^{\mathrm{ss}} - G_{\varphi,\mathrm{a}}(q^{\mathrm{d}}) + G_{\varphi,\mathrm{a}}(q) + K_{\mathrm{p}}(\varphi_{\mathrm{a}}^{\mathrm{d}} - \varphi) - K_{\mathrm{d}}\dot{\varphi}_{\mathrm{a}} + K_{\mathrm{i}}\int_0^t \tanh(\gamma\,(\varphi_{\mathrm{a},t'}^{\mathrm{d}} - \varphi_{\mathrm{a},t'}))\,\mathrm{d}t'. \quad (6.4)$$

The implementation of all control laws is available on GitHub[2].

6.2.3 Steady-State Planning

Our approach, as detailed in Section 6.2.2, requires us for a given desired end-effector position $p_{\mathrm{ee}}^{\mathrm{d}}$ to identify a statically-feasible configuration q^{d} with the matching steady-state actuation ϕ^{ss}.

We perform online static inversion to identify admittable desired configurations q^{d} and matching steady-state control inputs ϕ^{ss} during our experiments involving the FPU HSA robots. First, we substitute the inverse kinematics $\varrho_{\mathrm{ee}}(\chi_{\mathrm{ee}})$ into the static EOM. Then, as illustrated in Fig 6.3, we find the roots of the equation $G\circ\varrho_{\mathrm{ee}}(\chi_{\mathrm{ee}}^{\mathrm{d}}) + K\circ\varrho_{\mathrm{ee}}(\chi_{\mathrm{ee}}^{\mathrm{d}}) - \alpha(\varrho_{\mathrm{ee}}(\chi_{\mathrm{ee}}^{\mathrm{d}}),\phi_{\mathrm{ss}})$ with respect to $(\theta_{\mathrm{ee}},\phi_{\mathrm{ss}}) \in \mathbb{R}^3$ using nonlinear least-squares while enforcing constraints on the sign of ϕ. We solve this optimization problem with projected gradient descent.

In contrast, the static inversion optimization problem is not well-behaved for the identified EPU system parameters. Instead, we rely on rolling out the dynamics over a duration t_{ss} to steady-state and then optimize the steady-state input ϕ^{ss} such that the final end-effector error $\|p_{\mathrm{ee}}^{\mathrm{d}} - p_{\mathrm{ee}}^{\mathrm{ss}}\|$ is as small as possible. We formalize this optimization problem

[2] https://github.com/tud-phi/hsa-planar-control

in a least-squares fashion

$$\begin{aligned} \phi^{\text{ss}} = \underset{\phi}{\text{argmin}} \quad & \frac{1}{2} \| p_{\text{ee}}^{\text{d}} - p_{\text{ee}}^{\text{ss}}(\phi) \|_2^2, \\ \text{s.t.} \quad & x^{\text{ss}} = x(t_0) + \int_{t_0}^{t_{\text{ss}}} f(x(t), \phi) \, \mathrm{d}t, \quad \chi_{\text{ee}}^{\text{ss}} = \begin{bmatrix} p_{\text{ee}}^{\text{ss}} \\ \theta_{\text{ee}}^{\text{ss}} \end{bmatrix} = \pi_{\text{ee}}(q^{\text{ss}}), \end{aligned} \tag{6.5}$$

where $\dot{x}(t) = f(x(t), \phi)$ are the nonlinear state-space dynamics based on the EOM derived in Section 5.4.2 and $\phi \in \mathbb{R}^2$ is constant in time. We solve (6.5) online using the Levenberg-Marquardt algorithm. Finally, we choose $q^d = q^{\text{ss}}$ and $\chi_{\text{ee}}^{\text{d}} = \pi_{\text{ee}}(q^d)$.

6.2.4 Experimental Setup

We evaluate the system model and our proposed control approach on a robot consisting of four HSA rods. The material choice of the HSA is crucial and has a significant influence on the resulting mechanical characteristics of the robot (e.g., blocked force, holding torque, bending stiffness, etc.) (Truby et al., 2021). Furthermore, specific material requirements are dictated by the nature of the design of the HSA rod. The structure of the metamaterial is made of struts connected by living hinges. These living hinges must be thin, flexible, and accommodate high strains (Truby et al., 2021). Therefore, we decided to 3D-print the HSAs via digital projection lithography either from the photopolymer resin Carbon FPU 50 (stiffer) or the elastomeric polyurethane EPU 40 resin (softer).

Each HSA rod is actuated by a Dynamixel MX-28 servo motor. The Dynamixel motors are set to use position control mode. The robot is mounted platform-down on a cage with an Optitrack motion capture system, which measures the SE(3) pose of the platform at 200 Hz. Our algorithms run within a ROS2 framework[3]. The pose measurements are first projected into the plane of actuation and serve as an input to the closed-form inverse kinematics introduced in (5.17). We use a Savitzky-Golay filter with a window duration of 0.1 s to numerically differentiate $\chi_{\text{ee}}(t)$, $q(t)$ and gather with that $\dot{\chi}_{\text{ee}}(t)$ and $\dot{q}(t)$. We fit a cubic spline to the last 16 configuration measurements and differentiate (Kaptanoglu et al., 2022) to gather $\dot{q}(t)$.

6.2.5 System Identification

We follow the same system identification procedure as in Sec. 5.4.3. For the FPU-based robot, we identify $C_\varepsilon^{\text{FPU}} = 0.0079\,\text{m/rad}$, $S_{\text{be}}^{\text{FPU}} = -2.5 \cdot 10^{-5} + 3.9 \cdot 10^{-7} \, \frac{\phi_i^+}{l^0} \text{Nm}^2$, $S_{\text{sh}}^{\text{FPU}} = 0.043 + 0.0029 \, \frac{\phi_i^+}{l^0} \text{N}$, $S_{\text{ax}}^{\text{FPU}} = 0.74 + 0.0098 \, \frac{\phi_i^+}{l^0} \text{N}$, and $S_{\text{b,sh}}^{\text{FPU}} = -5.0 \cdot 10^{-4} \text{Nm/rad}$ where $l^0 = 0.059\,\text{m}$. Furthermore, we regress $C_\varepsilon^{\text{EPU}} = 0.0098\,\text{m/rad}$, $S_{\text{be}}^{\text{EPU}} = 5.7 \cdot 10^{-4} - 9.7 \cdot 10^{-6} \, \frac{\phi_i^+}{l^0} \text{Nm}^2$, $S_{\text{sh}}^{\text{EPU}} = 0.59 - 0.00047 \, \frac{\phi_i^+}{l^0} \text{N}$, $S_{\text{ax}}^{\text{EPU}} = 5.7 + 0.015 \, \frac{\phi_i^+}{l^0} \text{N}$, and $S_{\text{b,sh}}^{\text{EPU}} = -0.000\,48 \text{Nm/rad}$ for the EPU HSAs which have the same length as the FPU HSAs. Finally, we identify the axial rest strain σ_{ax}^0 before the start of each experiment.

6.2.6 Implementation of Closed-Loop Control

Next, we implement the closed-loop control strategy laid out in Section 6.2.2. After evaluating the control law at a rate of 40 Hz and saturating the control inputs to the ranges

[3] https://github.com/tud-phi/ros2-hsa

$[0, 3.40]$ rad for FPU and $[0, 4.71]$ rad for EPU, respectively, we map $\phi \in \mathbb{R}^2$ to desired positions of the four motors. For this, we consider the handedness of the HSAs and apply the same actuation magnitude to both rods on the same side of the virtual backbone. After tuning the gains for the feedback part of the model-based control laws in (6.3) and (6.4), we select $K_\mathrm{p} = \mathrm{diag}(0.3, 0.3)$, $K_\mathrm{i} = \mathrm{diag}(0.05, 0.05)\,1/\mathrm{s}$, $K_\mathrm{d} = \mathrm{diag}(0.01, 0.01)\,\mathrm{s}$, and $\gamma = \mathrm{diag}(100, 100)$. Furthermore, we report the performance of a model-free PID controller as a baseline. Here, the control input in task-space is given by $u_\mathrm{ts} = \begin{bmatrix} u_\mathrm{ts,x} & u_\mathrm{ts,y} \end{bmatrix}^\top = K_\mathrm{p}^\mathrm{PID}(p_\mathrm{ee}^\mathrm{d} - p_\mathrm{ee}) - K_\mathrm{d}^\mathrm{PID}\,\dot{p}_\mathrm{ee} + K_\mathrm{i}^\mathrm{PID}\int_0^t p_{\mathrm{ee},t'}^\mathrm{d} - p_{\mathrm{ee},t'}\,\mathrm{d}t'$, which is then mapped to the actuation via $\phi = \begin{bmatrix} u_\mathrm{ts,x} + u_\mathrm{ts,y}, & -u_\mathrm{ts,x} + u_\mathrm{ts,y} \end{bmatrix}^\top$. Here, we select $K_\mathrm{p}^\mathrm{PID} = \mathrm{diag}(10, 10)\,\mathrm{rad/m}$, $K_\mathrm{i}^\mathrm{PID} = \mathrm{diag}(110, 110)\,\mathrm{rad/m/s}$, and $K_\mathrm{d}^\mathrm{PID} = \mathrm{diag}(0.25, 0.25)\,\mathrm{rad\,s/m}$.

6.2.7 Evaluation

We define a reference trajectory $p_\mathrm{ee}^\mathrm{d}(k), k \in \{1, \ldots, n_k\}$ with a duration of 110 s and consisting of eleven step functions as the reference trajectory. We report the RMSE metric $\sqrt{\sum_{k=1}^{n_k} \frac{\|p_\mathrm{ee}^\mathrm{d}(k) - p_\mathrm{ee}(k)\|_2^2}{n_k}}$ for assessing the control performance, where $p_\mathrm{ee}(k)$ is the actual trajectory of the end-effector.

6.2.8 Results for Controlling an FPU-based HSA Robot

The *baseline PID* achieves an RMSE of 5.86 mm with respect to the reference trajectory. The *P-satI-D* based on (6.3) (with gravity compensation) exhibits an RMSE of 4.17 mm. Similarly, the *P-satI-D + GC* based on (6.4) (with gravity cancellation) displays an RMSE of 4.13 mm. We present a comparison of the three different controllers for a step response in Fig. 6.5 and plot the entire trajectories of the *baseline PID* and the *P-satI-D* in Figures 6.6 and 6.7, respectively. Additionally, we discretize various continuous reference trajectories into setpoints: star trajectory (873 setpoints and duration of 109 s), the flame of the TU Delft logo (680 setpoints and duration of 85 s), the contour of the MIT-CSAIL logo (1046 setpoints and duration of 131 s), and the outline of a bat at three different sizes (1510 setpoints and 189 s duration). The resulting Cartesian evolutions of the *P-satI-D* controller tracking these continuous references are displayed in Fig. 6.9.

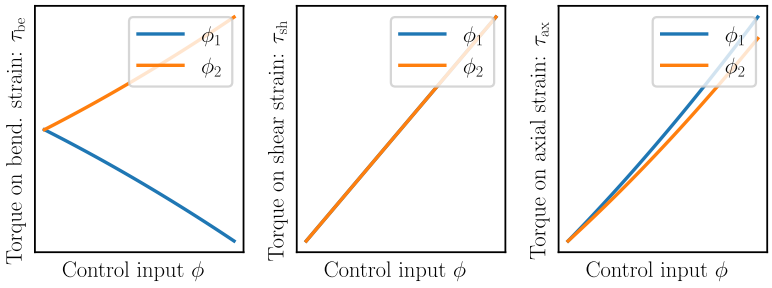
The step response in Fig. 6.5 shows how the two model-based controllers *P-satI-D* and *P-satI-D + GC* can leverage the planned ϕ^ss and q^d to achieve a fast response time of roughly 1.2 s. In contrast, the baseline PID needs to wait for the integral error to build up and thus has a much slower response time of approximately 4.2 s. Furthermore, overshooting caused by the baseline PID is usually more extensive than that caused by the model-based controllers. We conclude that *P-satI-D* (gravity compensation) and *P-satI-D + GC* (gravity cancellation) exhibit quite similar behavior. Sometimes, *P-satI-D* exhibits undershooting at the beginning of the transient and *P-satI-D + GC* overshooting towards the end of the transient (see Fig. 6.5(a)).

Results for Controlling an EPU-based HSA robot

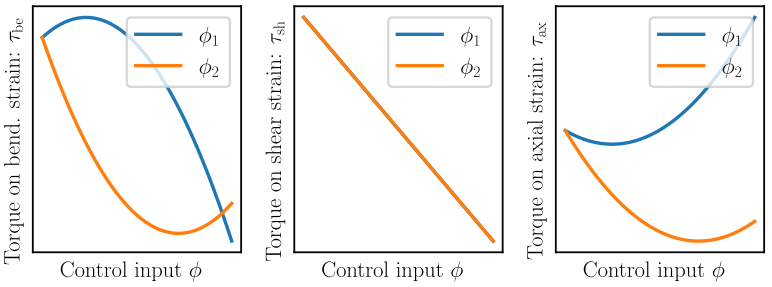
Tracking the reference trajectory of eleven-step functions with an EPU-based robot, the *baseline PID* controller has an RMSE of 4.40 mm. The *P-satI-D* (with gravity compensation)

can able to achieve an RMSE of 3.63 mm. The *P-satI-D + GC* controller exhibits similar performance(RMSE of 3.71 mm). We visualize the step response of all three controllers in Fig. 6.11 and the entire trajectory of the *P-satI-D* controller in Fig. 6.12.

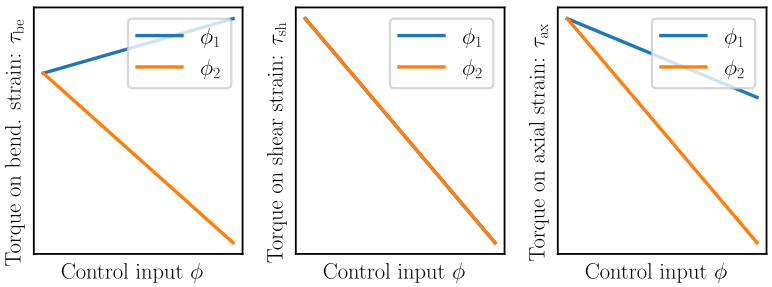
Again, we notice that the response time of the model-based controllers (0.54 s) is much shorter than the response time of the baseline PID (3.84 s). Furthermore, the importance of a model-based control law is motivated by the oscillations in the transient of the baseline PID (see x-coordinate in Fig. 6.11(a)). The steady-state error for the model-based controllers on the EPU material is slightly higher compared to the FPU material, as seen in Figures 6.11(a) & 6.12(a). In Section 5.4.3, we noticed that the shear model doesn't fully capture the actual system behavior. This then results in an error in the planned desired configuration q^{d}, which the controller is not able to resolve because of the underactuation of the robot (see Fig. 6.12(b)).



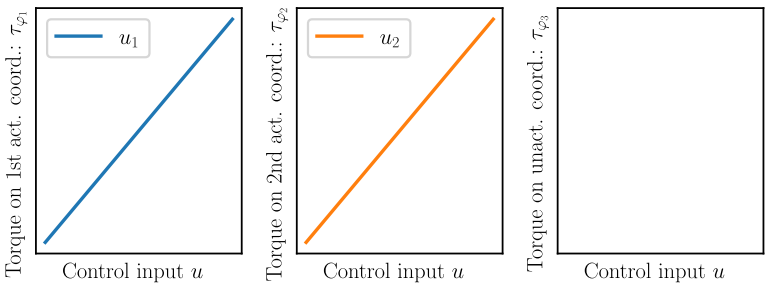
(a) $\alpha(q, \phi)$ for FPU



(b) $\alpha(q, \phi)$ for EPU



(c) $\alpha(q^{ss}, \phi^{ss}) + A_{\phi^{ss}}(q)(\phi - \phi^{ss})$ for EPU



(d) $A_\varphi(\phi - \phi^{ss})$

Figure 6.2: Mapping of actuation $\phi \in \mathbb{R}^2$ to configuration-space torques $\tau \in \mathbb{R}^3$ for planar HSA robots. The first two rows visualize the modeled nonlinear, coupled mapping for the FPU and EPU materials, respectively. The third row illustrates the mapping for a linearized actuation term. Finally, in actuation coordinates, the mapping is fully decoupled, as shown in the fourth row.

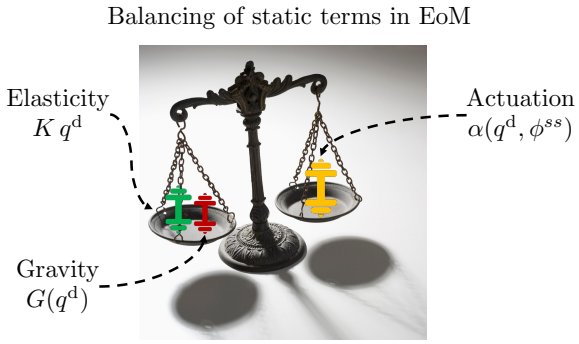


Figure 6.3: Illustration of the idea behind steady-state planning: we balance the static forces that are acting at steady-state on the robot with torques generated by a steady-state actuation ϕ^{ss}.

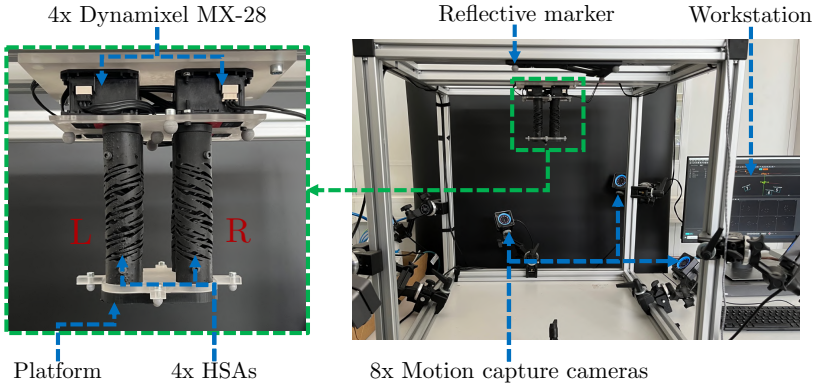
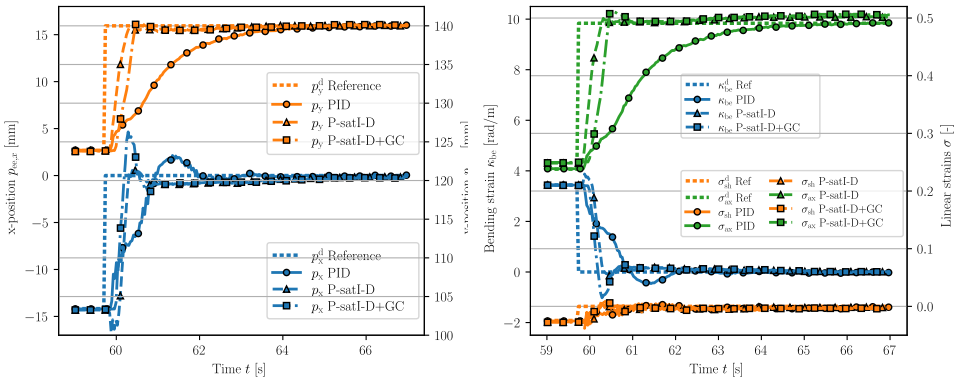
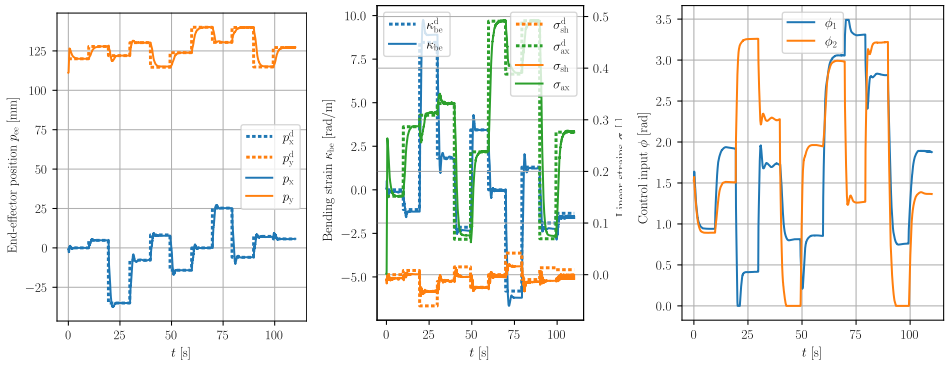


Figure 6.4: Experimental setup: the parallel robot consists of four HSA rods connected by a platform at their distal end. Four servo motors actuate the HSAs. We track the pose of the end-effector with a motion capture system by attaching reflective markers to the platform.



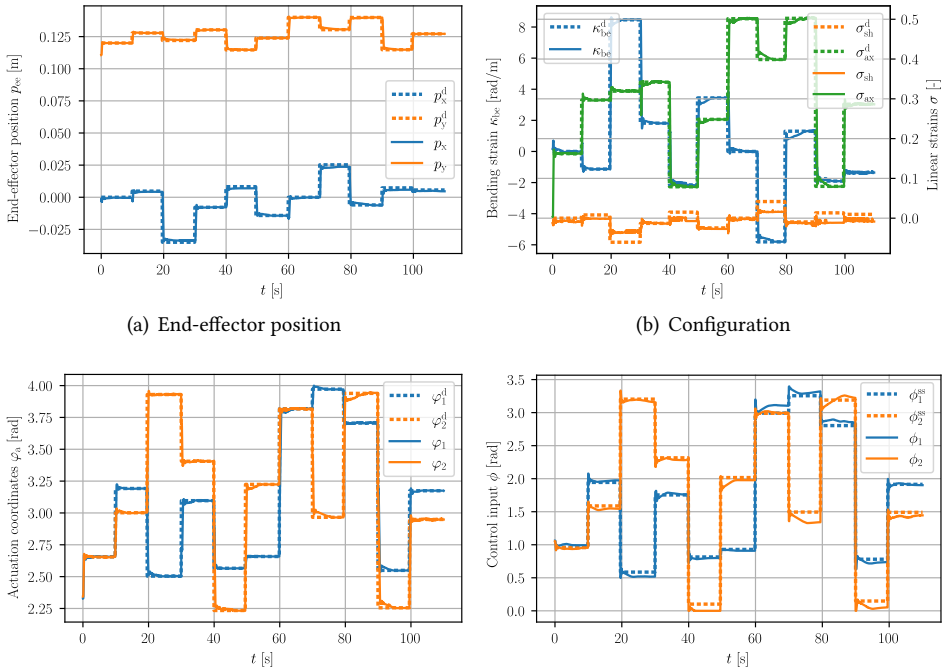
(a) End-effector position (b) Configuration

Figure 6.5: Step response of the *baseline PID*, *P-satI-D* (with gravity compensation), and *P-satI-D + GC* (with gravity cancellation) controllers on an FPU-based HSA robot.



(a) End-effector position (b) Configuration (c) Control input

Figure 6.6: Experimental results for tracking a reference trajectory of eleven step functions with the baseline PID controller on an FPU-based HSA robot. **Panel (a):** End-effector position with the dotted and solid lines denoting the task space reference and actual position, respectively. **Panel (b):** The planned (dotted) and the actual (solid) configuration. **Panel (c):** The planned (dotted) and the actual (solid) actuation coordinates of the collocated system. **Panel(d):** The saturated planar control inputs are visualized with solid lines and the computed steady-state actuation with dotted lines.



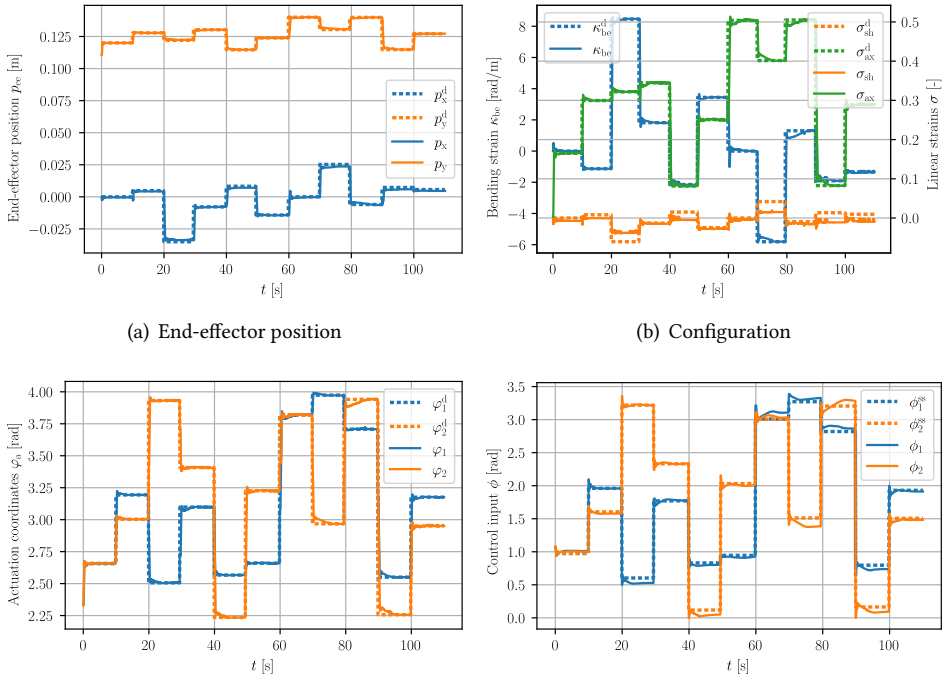
(a) End-effector position

(b) Configuration

(c) Actuation coordinates

(d) Control input

Figure 6.7: Experimental results for tracking a reference trajectory of eleven step functions with the P-satI-D controller on an FPU-based HSA robot. **Panel (a):** End-effector position with the dotted and solid lines denoting the task-space reference and actual position, respectively. **Panel (b):** The planned (dotted) and the actual (solid) configuration. **Panel (c):** The planned (dotted) and the actual (solid) actuation coordinates of the collocated system. **Panel(d):** The saturated planar control inputs are visualized with solid lines and the computed steady-state actuation with dotted lines.



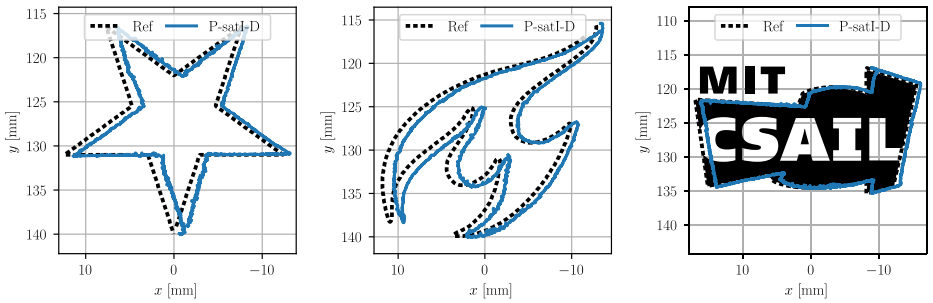
(a) End-effector position

(b) Configuration

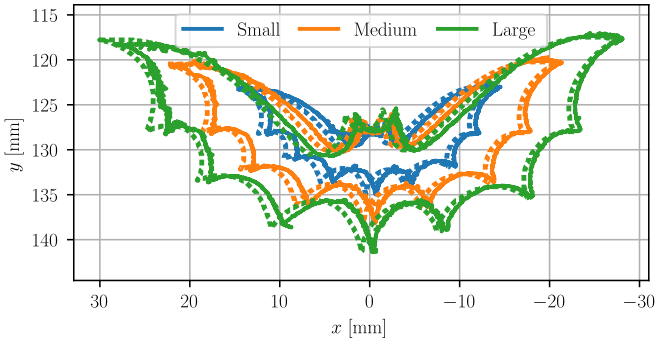
(c) Actuation coordinates

(d) Control input

Figure 6.8: Experimental results for tracking a reference trajectory of eleven step functions with the P-satI-D + gravity cancellation controller on an FPU-based HSA robot. **Panel (a):** End-effector position with the dotted and solid lines denoting the task-space reference and actual position, respectively. **Panel (b):** The planned (dotted) and the actual (solid) configuration. **Panel (c):** The planned (dotted) and the actual (solid) actuation coordinates of the collocated system. **Panel(d):** The saturated planar control inputs are visualized with solid lines and the computed steady-state actuation with dotted lines.

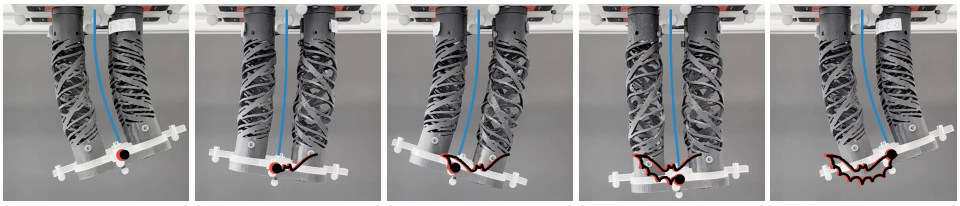


(a) Star (b) TUD flame (c) MIT-CSAIL



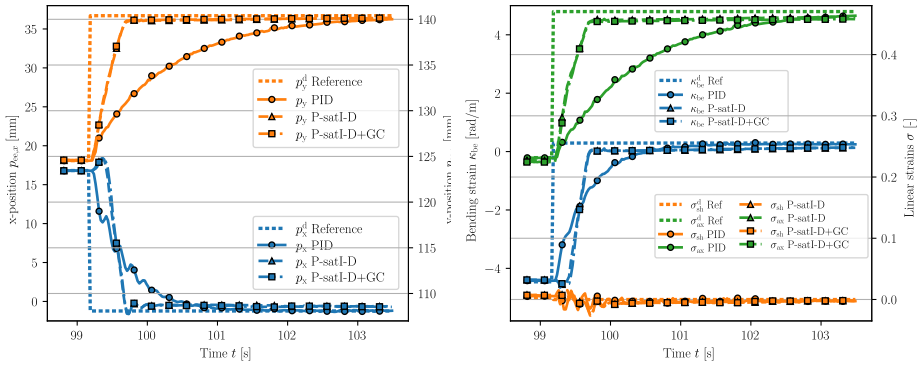
(d) Bat trajectories

Figure 6.9: Cartesian evolution of the proposed P-sat-D controller (solid lines) tracking various continuous reference trajectories (dotted lines) on the FPU robot.



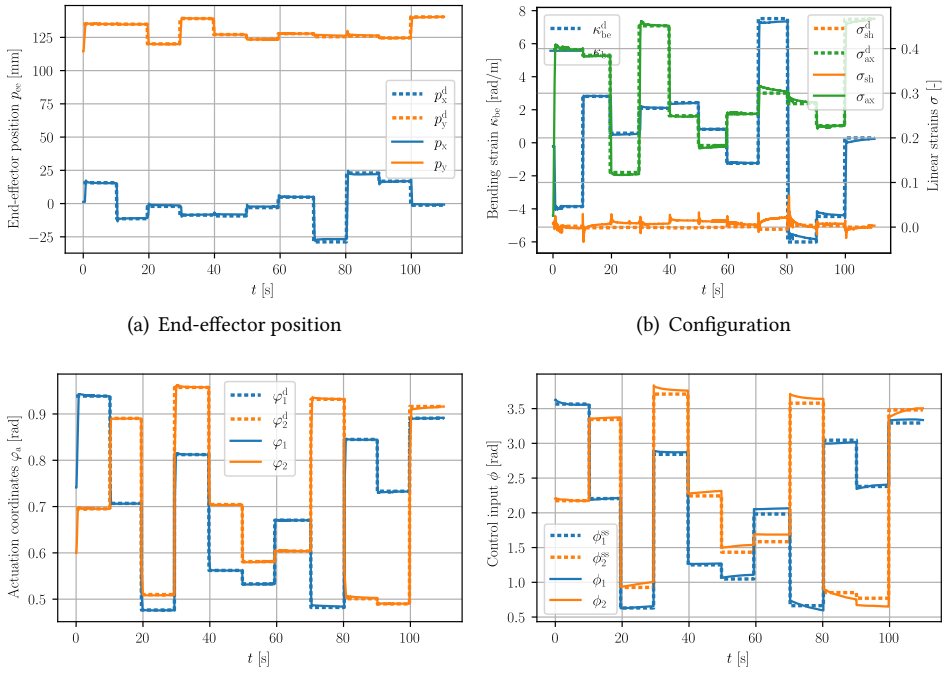
(a) t=0 s (b) t=47 s (c) t=94 s (d) t=141 s (e) t=188 s

Figure 6.10: Sequence of stills for the large bat trajectory performed with the P-satD controller on the FPU robot. The red and black dots visualize the desired and current end-effector positions, respectively. The past trajectory is plotted in red (reference) and black (actual). The blue line renders the shape of the virtual backbone.



(a) End-effector position (b) Configuration

Figure 6.11: Step responses of the *baseline PID*, *P-satI-D* (with gravity compensation), and *P-satI-D + GC* (with gravity cancellation) controllers on an EPU-based HSA robot.



(a) End-effector position (b) Configuration

(c) Actuation coordinates (d) Control input

Figure 6.12: Experimental results for tracking a reference trajectory of eleven step functions with the P-satI-D controller on an EPU-based HSA robot. **Panel (a):** End-effector position with the dotted and solid lines denoting the task-space reference and actual position, respectively. **Panel (b):** The planned (dotted) and the actual (solid) configuration. **Panel (c):** The planned (dotted) and the actual (solid) actuation coordinates of the collocated system. **Panel(d):** The saturated planar control inputs are visualized with solid lines and the computed steady-state actuation with dotted lines.

6.3 Cartesian Space Impedance Control

We introduce below a novel control strategy that addresses some limitations of the configuration-space regulation controller presented in Sec. 6.2.2 that are critical for achieving both structural and computational compliance. Namely, we (i) avoid computationally demanding planning procedures, (ii) remove integral terms that are unsafe for environment interaction, and (iii) enable impedance shaping in operational space. This Cartesian-space impedance controller is inspired by (Ott, 2008) and (Della Santina et al., 2020c), but specifically designed for and tailored to HSA robots. Crucially, we need to overcome the challenges of underactuation and the nonlinearity in the actuation, which were not present in that original work.

6.3.1 Background: Operational Space Dynamics

Although this has never been done in the context of HSA robots, it is immediate to see that their dynamics (6.1) can be projected into operational space, yielding the form (Della Santina et al., 2020c, 2019b)

$$\Lambda(q)\ddot{x}+\mu(q,\dot{q})\dot{q}+J_{\mathrm{M}}^{+\top}(G(q)+K(q-q^0)+D\dot{q})=J_{\mathrm{M}}^{+\top}\alpha(q,\phi), \tag{6.6}$$

where $J_{\mathrm{M}}^{+}(q)=M^{-1}J^{\top}(JM^{-1}J^{\top})^{-1}\in\mathbb{R}^{3\times 2}$ is the dynamically consistent pseudo-inverse, $\Lambda(q)=(JM^{-1}J^{\top})^{-1}\in\mathbb{R}^{2\times 2}$ is the inertia matrix in operational space, and $\mu(q,\dot{q})=\Lambda(q)(JM^{-1}C-\dot{J})\in\mathbb{R}^{2\times 3}$ collects the Cartesian Coriolis and centrifugal terms.

6.3.2 Control Strategy

We will present the control strategy in two steps: first, we will introduce the control law and derive the associated desired configuration-space torques, and second, we will discuss how to map the control inputs to the actuation ϕ.

Proposed Controller

We propose the following dynamic feedback law that renders x^{d} an attractor of the closed-loop system

$$\begin{aligned}\tau=&J^{\top}(q)\left(K_x(x^{\mathrm{d}}-x)-D_x\dot{x}\right)+G(q)+K(q-q^0)\\&+J^{\top}(q)J_{\mathrm{M}}^{+\top}(q)D\dot{q}+J^{\top}(q)\mu(q,\dot{q})\left(\mathbb{I}_3-J_{\mathrm{M}}^{+}(q)J(q)\right)\dot{q}\end{aligned} \tag{6.7}$$

where $\tau\in\mathbb{R}^3$ is the desired torque in configuration space, $G(q)+K(q-q^0)$ cancels the acting gravitational and elastic forces, and $J^{\top}J_{\mathrm{M}}^{+\top}D\dot{q}$ removes the natural dissipation in operational space. We emphasize that because the system is underactuated, we need to cancel the stiffness directly in the configuration instead of operational space as done in previous work (Della Santina et al., 2020c). We can shape our desired impedance characteristics in Cartesian space with the PD term $f_{\mathrm{PD}}=K_x(x^{\mathrm{d}}-x)-D_x\dot{x}$ which is then projected into configuration space by premultiplying with $J^{\top}(q)$.

The term $\mu(q,\dot{q})\left(\mathbb{I}_3-J_{\mathrm{M}}^{+}(q)J(q)\right)\dot{q}$ decouples the operational space dynamics from the residual of the null-space dynamics (Della Santina et al., 2020c; Ott, 2008). The identity $\dot{q}=J_{\mathrm{M}}^{+}\dot{x}+Z^{\top}\nu_{\mathrm{N}}$, where $Z^{\top}\in\mathbb{R}^{3\times 1}$ is the dynamically-consistent pseudo-inverse of the null space, allows us to formulate $\dot{q}$ as a sum of the operational space velocity $\dot{x}$ and the null-space velocity ν_{N}. Leveraging this identity, the Coriolis and centrifugal matrix $\mu(q,\dot{q})$ can be

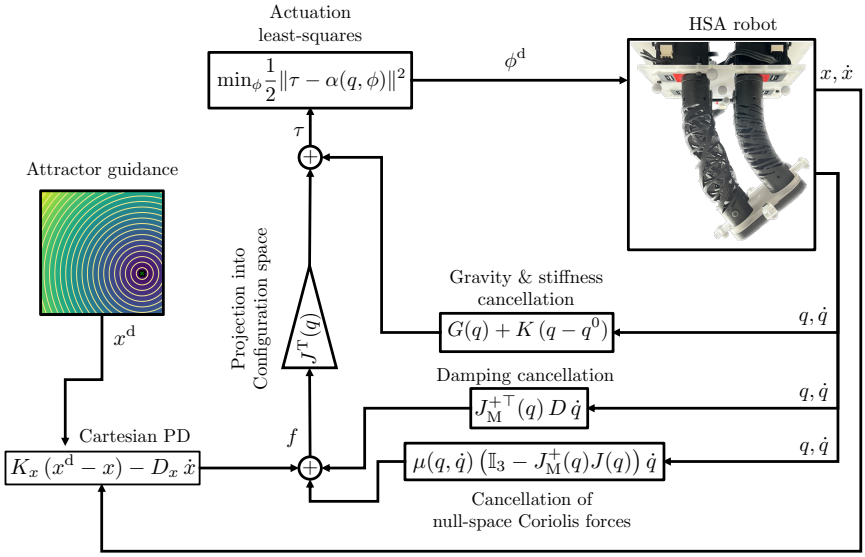


Figure 6.13: Block scheme of the operational space space impedance controller: after the cancellation of the existing soft robot dynamics (e.g., null-space Coriolis forces, gravity, stiffness, and damping), we reshape the potential field according to the desired Cartesian impedance K_x, D_x with a globally asymptotically stable equilibrium at $x = x^{\mathrm{d}}$. Subsequently, we project the desired forcing f from operational space to configuration space. Finally, we solve a nonlinear least-squares optimization problem to match the desired configuration-space torques τ as closely as possible with the torques generated by the actuation ϕ^{d}.

split into a term $\mu_x(q,\dot{q}) = \mu J_{\mathrm{M}}^{+} \in \mathbb{R}^{2\times 2}$ excited by x and the expression $\mu_{\mathrm{N}}(q,\dot{q}) = \mu Z^{\top} \in \mathbb{R}^{2\times 1}$ that is excited by the null-space coordinates resulting in $\mu(q,\dot{q})\,\dot{q} = \mu_x(q,\dot{q})\,\dot{x} + \mu_{\mathrm{N}}(q,\dot{q})\,\nu_{\mathrm{N}}$. This allows us to cancel the term $\mu_{\mathrm{N}}(q,\dot{q})\,\nu_{\mathrm{N}}$ through $\mu(q,\dot{q})\left(\mathbb{I}_3 - J_{\mathrm{M}}^{+}(q)J(q)\right)\dot{q}$ without having to compute the null space explicitly.

In summary, the closed-loop dynamics in operational space can be stated as

$$\Lambda(q)\,\ddot{x} + \mu(q,\dot{q})\,J_{\mathrm{M}}^{+}\,\dot{x} + K_x\,(x - x^{\mathrm{d}}) + D_x\,\dot{x} = 0, \tag{6.8}$$

which results in x^{d} being the globally asymptotically stable equilibrium of the closed-loop operational space dynamics. [4]

Mapping to Actuation

Now that we have formulated our control law τ in configuration space, we need to identify a strategy to specify the motor angles $\phi \in \mathbb{R}^2$ such that $\alpha(q,\phi) \approx \tau$. Note that, in contrast to other continuum soft robots studied in literature (Della Santina et al., 2023), the actuation term $\alpha(q,\phi)$ is not affine in control. In previous work (Stölzle et al., 2024b), we side-stepped this challenge by linearizing with respect to the steady-state actuation ϕ^{ss}: $A(q) = \|\frac{\partial \alpha}{\partial \phi}\|_{\phi=\phi^{\mathrm{ss}}}$ therefore recovering the usual scenario of an affine actuation function. Unfortunately, this is not possible in the setting of this work as i) we do not have access to such ϕ^{ss}, and ii) linearizing around ϕ causes the closed-loop system to become unstable. We, therefore, propose to formulate instead a nonlinear least-squares problem $\phi^{\mathrm{d}} = \mathrm{argmin}_{\phi}\,\frac{1}{2}\|\tau - \alpha(q,\phi)\|^2$ and solve it in real-time with a Levenberg Marquardt solver implemented in JAX (Blondel et al., 2021).

We note that this approach is not guaranteed to be valid for the general case of an underactuated soft robot but for this particular structure of $\alpha(q,\phi) \in \mathbb{R}^3$ with $\phi \in \mathbb{R}^2$ it is possible to identify solutions ϕ with the Euclidean norm of the residual being smaller than 0.001. The source code of the controller is available on GitHub[5].

6.3.3 Experimental Setup

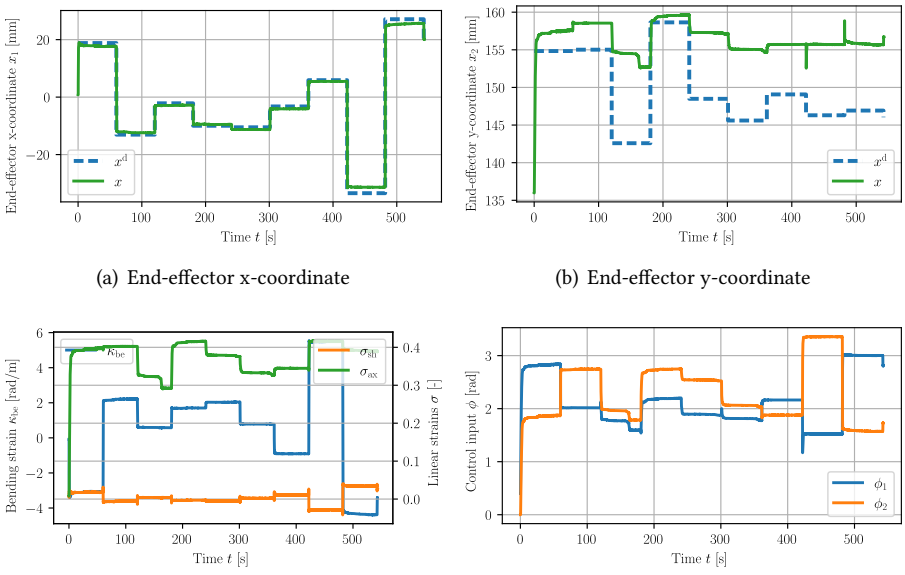
We use the same experimental setup as introduced in Sec. 6.2.4. Over a duration of 540 s, we randomly generate nine setpoints $x^{\mathrm{d}}(t) \in \mathbb{R}^2$ within the operational workspace of the robot (see Fig. 6.1(b)). We evaluate the Cartesian impedance controller using the gains $K_{\mathrm{p}} = 300\,\mathrm{N/m}$, $K_{\mathrm{d}} = 1.5\,\mathrm{Ns/m}$ at a frequency of 50 Hz and finally send the desired motor positions to the servos.

6.3.4 Results

In Fig. 6.14, we present the results of the experimental evaluation of the Cartesian impedance controller. The fast response time, a well-known characteristic of model-based control approaches, is evident. However, the errors in the model (for example, caused by hysteresis or unmodeled nonlinearities) (Stölzle et al., 2024b), together with the lack of integral action, lead to steady-state errors, which are especially pronounced for the y-coordinate.

[4] Please note that this only holds under the assumption that the desired operational-space control input can be perfectly tracked by the actuators, which is generally not the case in underactuated settings.

[5] `https://github.com/tud-phi/hsa-planar-control`



(a) End-effector x-coordinate

(b) End-effector y-coordinate

(c) Configuration q

(d) Control input ϕ

Figure 6.14: Experimental results for tracking a reference trajectory of nine-step functions with the Cartesian impedance controller. **Panel (a) & (b):** The x/y-coordinate of the end-effector position with the solid line denoting the actual position, the dotted line the attractor position, and the dashed line the reference (i.e., the setpoint). **Panel (c):** The evolution of the configuration. **Panel (d):** The saturated planar control inputs.

6.4 Conclusion

This chapter presented two strategies for effective, model-based regulation of planar HSA robots: the first proposes a configuration-space controller in conjunction with steady-state planning for mapping task-space setpoints to configuration-space setpoints, and the second introduces an operational space impedance controller.

6.4.1 Configuration Space Regulation

First, we introduced a configuration-space controller that combines an integral-saturated PID controller with a potential-shaping feedforward term. The excellent agreement of the model with the actual system behavior (as shown in Section 5.4) enables our model-based controllers to perform very well at the task of setpoint regulation. For the model-based controllers, any mismatch between the dynamic model and the actual system (as analyzed in Section 5.4.3) has two impacts: (i) the steady-state planning provides us with a desired configuration q^d which the underactuated robot cannot achieve. This then, in turn, causes a small steady-state error in the end-effector position as seen for the manual setpoints in Fig. 6.7(a) & 6.8(a) and for the continuous references in Fig. 6.9. This steady-state error is absent in the baseline PID as its integral term acts directly in task space. We suggest that future work include an integral term directly on the end-effector position to remove the remaining steady-state error of the model-based controller. Secondly, as (ii), model

errors will lead to an offset in the planned steady-state actuation ϕ^{ss}. Therefore, applying a constant ϕ^{ss} will not move the robot exactly to p_{ee}^{d}. As shown in Fig. 6.7(d) & 6.8(d), the P-satI-D feedback term can compensate for this effect through its proportional and integral terms applied in the collocated variables. As clearly seen in Fig. 6.7(d), the control input drifts away from the planned ϕ^{ss} to counteract the modeling errors. In particular, for small to medium bending and axial strains, we observe a very good agreement between the model and the actual system behavior and, therefore, excellent control performance. The story slightly changes for more significant bend and twist angles. While the system identification showed that the expressiveness of the model was sufficient, we saw significant errors in our feedforward terms in some of our control experiments, which needed to be compensated for by the P-satI-D feedback controller. Precisely, Fig. 6.7(b) displays errors in the planned shear strain, which subsequently also leads to minor steady-state deviations in the x-coordinate of the end-effector, as can be seen in Fig. 6.7(a). We hypothesize that these errors are mainly caused by the hysteresis characteristics of the HSAs (Good et al., 2022). This hypothesis is corroborated by the fact that we had to recalibrate the axial rest strain σ_{ax}^{0} before the start of each experiment.

6.4.2 Operational Space Impedance Control

The Cartesian impedance controller, on the other hand, is designed to regulate the end-effector position directly and allows us to shape the impedance in operational space and, with that, fully preserve the softness of the robot. It is, therefore, particularly suitable for applications where the robot interacts with the environment. For example, we will in Chapter 7 demonstrate how the impedance controller can be combined with motor-imagery brain signals to guide the robot towards a desired position and how the impedance controller guarantees safety even when the setpoints are wrongly planned. This impedance needs to be traded off with steady-state errors. We observed that the impedance controller is particularly sensitive to the model errors, as it does not have an integral term to compensate for them.

Afterword

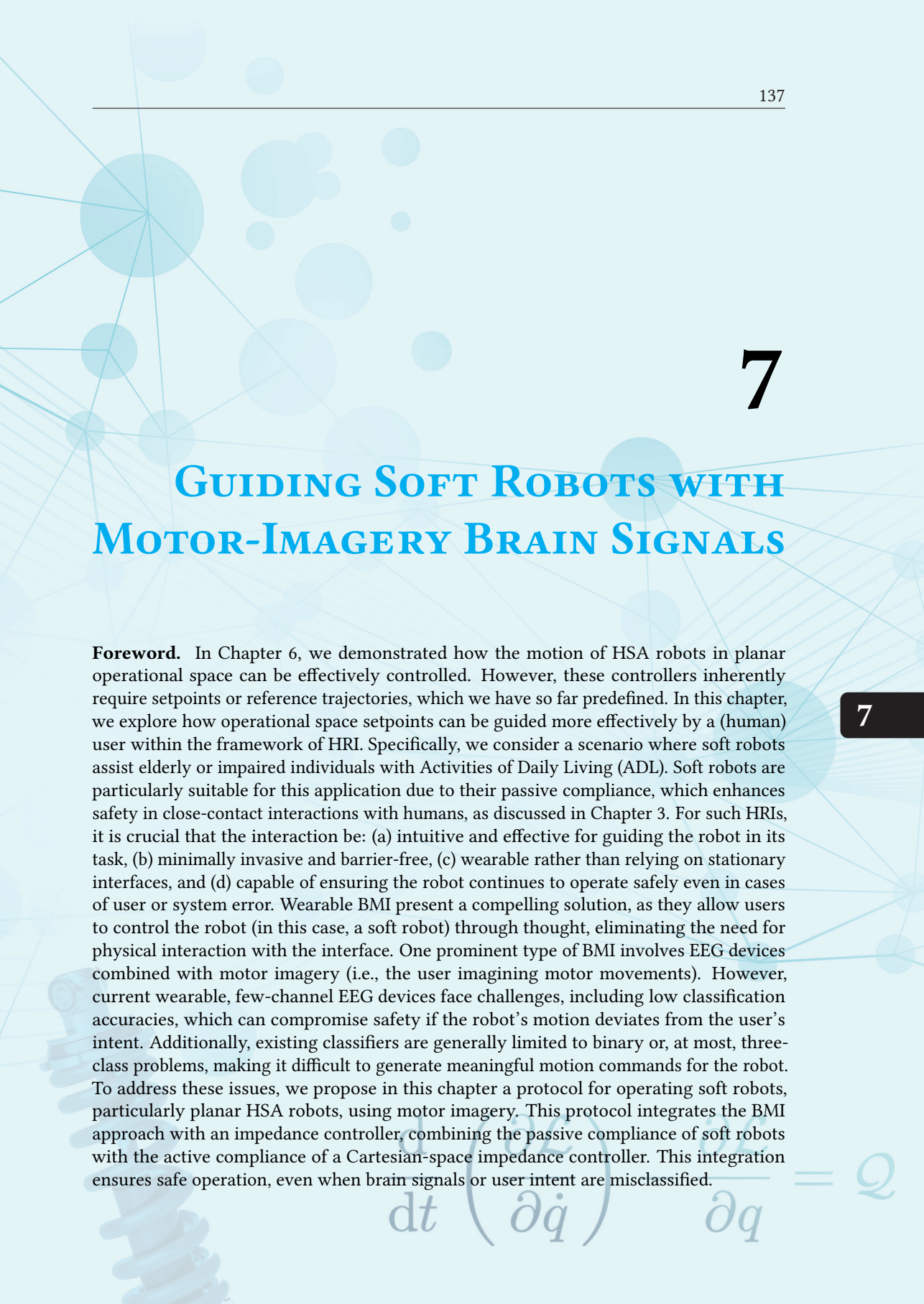
This chapter presented the first control strategies in literature for HSA robots and verified them experimentally on two HSA robot prototypes in the planar case. Each of the two presented controllers offers unique advantages: On the one hand, the configuration-space P-satI-D + potential shaping controller exhibits exceptional regulation accuracy, with minimal or no steady-state errors and exceptionally short response times. Excitingly, follow-up work by Soleti et al. (2025) has extended and adapted this control approach to underactuated dielectric elastomer soft robots. On the other hand, the operational space impedance controller complements the passive compliance of the soft robot with active compliance in the control strategy. It accomplishes this by eliminating integral terms that could pose potential safety hazards. Additionally, it enables us to precisely shape the stiffness of the closed-loop system in Cartesian space. In envisioning a future where soft robots assist elderly individuals with activities of daily living, an effective HRI is paramount. While this chapter has focused on programmatically specifying setpoints for regulation controllers, we recognize a gap in how humans can intuitively command

these robots. A BMI emerges as a compelling solution, offering a direct and barrier-free interaction method that could significantly enhance the usability of assistive robotics. Consequently, in Chapter 7, we explore the feasibility of guiding an operational space impedance controller via EEG-based motor imagery, aiming to create a more natural and accessible means of control of soft, and specifically HSA, robots. Furthermore, we illustrate how the Cartesian-space stiffness enables us to apply force in the desired directions (in this case, releasing hairspray by pressing a button with the end-effector) while maintaining the compliance and, consequently, the embodied intelligence of the soft robot in all other directions.

7

Guiding Soft Robots with Motor-Imagery Brain Signals

Foreword. In Chapter 6, we demonstrated how the motion of HSA robots in planar operational space can be effectively controlled. However, these controllers inherently require setpoints or reference trajectories, which we have so far predefined. In this chapter, we explore how operational space setpoints can be guided more effectively by a (human) user within the framework of HRI. Specifically, we consider a scenario where soft robots assist elderly or impaired individuals with Activities of Daily Living (ADL). Soft robots are particularly suitable for this application due to their passive compliance, which enhances safety in close-contact interactions with humans, as discussed in Chapter 3. For such HRIs, it is crucial that the interaction be: (a) intuitive and effective for guiding the robot in its task, (b) minimally invasive and barrier-free, (c) wearable rather than relying on stationary interfaces, and (d) capable of ensuring the robot continues to operate safely even in cases of user or system error. Wearable BMI present a compelling solution, as they allow users to control the robot (in this case, a soft robot) through thought, eliminating the need for physical interaction with the interface. One prominent type of BMI involves EEG devices combined with motor imagery (i.e., the user imagining motor movements). However, current wearable, few-channel EEG devices face challenges, including low classification accuracies, which can compromise safety if the robot's motion deviates from the user's intent. Additionally, existing classifiers are generally limited to binary or, at most, three-class problems, making it difficult to generate meaningful motion commands for the robot. To address these issues, we propose in this chapter a protocol for operating soft robots, particularly planar HSA robots, using motor imagery. This protocol integrates the BMI approach with an impedance controller, combining the passive compliance of soft robots with the active compliance of a Cartesian-space impedance controller. This integration ensures safe operation, even when brain signals or user intent are misclassified.



Abstract. *Integrating Brain-Machine Interfaces into non-clinical applications like robot motion control remains difficult - despite remarkable advancements in clinical settings. Specifically, EEG-based motor imagery systems are still error-prone, posing safety risks when rigid robots operate near humans. This chapter presents an alternative pathway towards safe and effective operation by combining wearable EEG with physically embodied safety in soft robots. We introduce and test a pipeline that allows a user to move a soft robot's end effector in real time via brain waves that are measured by as few as three EEG channels. A robust motor imagery algorithm interprets the user's intentions to move the position of a virtual attractor to which the end effector is attracted, thanks to a Cartesian impedance controller. We specifically focus here on planar soft robot-based architected metamaterials. Due to their compact and portable nature, they could be mounted to a mobile platform in the future and allow assistance with Activities of Daily Living (ADL). We preliminarily but quantitatively evaluate the approach on the task of setpoint regulation. We observe that the user reaches the proximity of the setpoint in 66% of steps and that for successful steps, the average response time is 21.5s. We also demonstrate the execution of simple real-world tasks involving interaction with the environment, which would be extremely hard to perform if it were not for the robot's softness.*

This chapter is partly based on 🏆 ***M. Stölzle****, S. S. Baberwal*, D. Rus, S. Coyle†, and C. Della Santina† (2024). Guiding Soft Robots with Motor-Imagery Brain Signals and Impedance Control. In Proceedings of the 2024 IEEE 7th International Conference on Soft Robotics (RoboSoft) (pp. 1-8). IEEE. Received the* ***Best Paper Award*** (Stölzle et al., 2024a).

*M.S. and S.S.B. contributed equally to this work. †C.D.S and S.C. shared equal supervising duties. M.S., S.S.B., and C.D.S. conceived the research project. M.S. and S.S.B. devised the protocol for translating motor imagery classification into end-effector motion, conducted all experiments, and wrote the paper together. M.S. formulated and developed the Cartesian impedance controller (presented in Chapter 6) and constructed the experimental setup (e.g., motion capture system, setpoint projection, mounting the hairspray container). S.S.B. implemented and trained the EEG data processing and classification pipeline. C.D.S. and S.C. supervised the project and provided funding.

7.1 Introduction

Brain Machine Interfaces (BMI) (Liu et al., 2024b) facilitate the translation of neural activity into actionable commands, enabling individuals to control external devices and systems through their thoughts and attention (Coyle et al., 2007; Lee et al., 2017). Compared to traditional bulky EEG setups (Van Erp et al., 2012), one of the emerging avenues towards practical and wearable EEGs devices are systems based on motor imagery signals due to their intuitiveness and no external dependency on (e.g., visual) stimuli. These have been use in stroke rehabilitation (Khan et al., 2020). Several works in literature have considered using this technology to control robot manipulators (Aldini et al., 2019; Bhattacharyya et al., 2017; Lee et al., 2024; Liu et al., 2018b; Schiatti et al., 2017).

However, the state-of-the-art classifiers on few-channel, online EEG signals are still limited in achieving an accuracy of 65-75 % (Arpaia et al., 2022; Lee et al., 2024) and are prone to producing outliers, which make it very challenging to operate robots safely and robustly using these techniques (Liu et al., 2024b). In (rigid) robotics literature, this has been addressed by relying on force-based (i.e., impedance) control (Schiatti et al., 2017) and by making the robot's behavior more predictable (Aldini et al., 2019). In this chapter, we follow a different path and investigate *embodying* safety by pairing soft robots (Della Santina et al., 2020b; Rus and Tolley, 2015) with BMI. This way, risks can be mitigated, and more natural interactions with an unstructured environment can be achieved by relying on structural compliance.

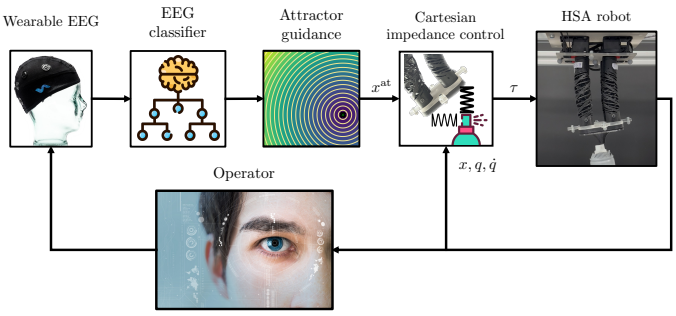


Figure 7.1: High-level control scheme of an operator steering an HSA robot with brain signals. We leverage the EEG signal classifications for guiding an attractor in the plane. Subsequently, an impedance controller is designed to shape the impedance in Cartesian space and render the attractor to be a globally asymptotically stable equilibrium of the potential.

While BMI-based assistance has been investigated with a focus on soft exosuits assisting hand-rehabilitation (Zhang et al., 2019) or with strenuous arm acitivites (Tacca et al., 2022), it is still an open challenge how BMI can be used for controlling soft manipulators. In this chapter, we make a first step towards solving this challenge by proposing a pipeline (see Fig. 7.1) that lets the user steer the soft robot's end-effector in Cartesian space. The two key ingredients are a novel mapping strategy transforming the brain signals into meaningful references and Cartesian impedance control. The latter is essential because it allows for preserving the robot's compliance in closed-loop (Della Santina et al., 2017). We build the proposed BMI pipeline around a HSA soft robot (Stölzle et al., 2023, 2024b), which relies on

architected metamaterials and electrical actuation to elongate, bend, and twist. HSA robots are an excellent match for our test case of assisting with ADL, as they can be portable because of their electrical actuation, they are compact and combine many DOFs in one segment, and easy to manufacture via 3D printing (Truby et al., 2021).

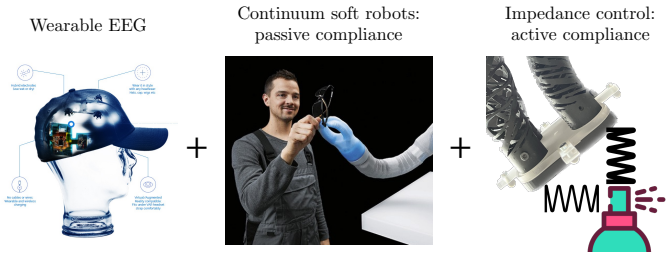


Figure 7.2: In this work we propose combining wearable EEG devices with continuum soft robots and impedance control. The classification accuracy of motor imagery based on signals from few-channel, wearable EEG devices is currently limited to less than 75 %. The inevitable operation errors can be accepted by establishing both passive and active compliance: The embodied intelligence of soft robots guarantees safety during interactions with the environment, which is complemented by an impedance controller that allows us to design the Cartesian stiffness and damping characteristics of the robot.

We quantitatively verify the entire approach on mind-controlled setpoint regulation involving tracking a reference consisting of nine-step functions and demonstrate the qualitative behavior when assisting with a simple daily living activity. Furthermore, we compare the performance of our motor imagery-based control to an approach that can be considered an upper bound on performance: The control using keyboard inputs is a very effective and efficient Human Machine Interface (HMI) strategy but requires the user to always interact physically with a keyboard, which is not always desirable or possible (e.g., for users with disabilities).

Our contribution is establishing a BMI strategy for continuum soft robots. This strategy is supported by experiments in which we perform setpoint regulation with a planar HSA robot and motor imagery, which we experimentally validated on a simple ADL task involving environment interaction: the user needs to steer the end-effector towards the tip of a hairspray container, apply force for releasing the fluid, and finally let the robot retreat from the contact.

A video attachment to this chapter, including recordings of experimental results, can be found on YouTube[1]. Furthermore, we have open-sourced our code, including the OpenVibe pipeline, on GitHub[2].

7.2 Methodology

In this chapter, we let the user steer with motor imaginary brain signals the Cartesian position $x \in \mathbb{R}^2$ of the end-effector (i.e., the platform) of a planar HSA robot. We realize this strategy by first classifying the motor imaginary signals into Cartesian-space movement directions (e.g., the active axis and sign of the movement). We use this information to adjust

[1] https://youtu.be/wZTOxBPZmPc
[2] https://github.com/tud-phi/sr-brain-control

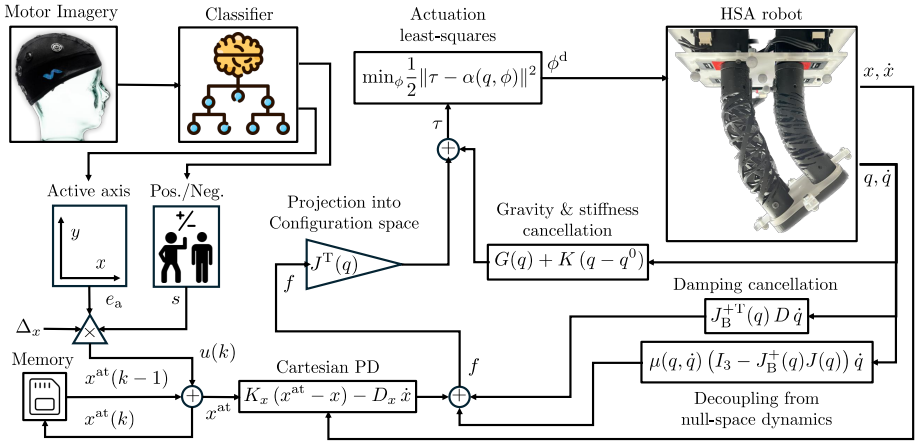


Figure 7.3: Scheme of the proposed approach to control HSA robots with motor imagery. Brain signals steer an attractor in operational space: first, we switch the active coordinate axis when we detect jaw clenching. If no jaw clenching is detected, we classify the EEG signals based on left/right motor imaginations into positive and negative movements along the active axis. Next, we regulate the robot towards the chosen attractor position x^{at} with a Cartesian impedance controller. This controller first cancels all static forces and the residual coupling of the null space on the operational space dynamics. This allows us to now shape our own potential with a PD term in operational space. As the robot is underactuated, we optimize least-squares to identify the actuation ϕ^{d} so that the residual between the desired and actual torques in the configuration space is minimized. Icons created by Flaticon©.

the position of a operational space attractor iteratively (see Section 7.2.2). We illustrate the methodology in Fig. 7.3.

7.2.1 Background: Motor Imagery-based BMI systems

Imagining the movement of body parts or limbs (e.g., hands, legs, tongue) without moving it or the mental rehearsal of a motor act without overt movement execution is termed Motor Imagery (Lotze and Halsband, 2006). The neuronal activities observable inside a frequency range of 8 Hz to 12 Hz (Mu) and 12 Hz to 30 Hz (Beta) are associated with cortical areas directly connected to the brain's motor output (activating primary sensorimotor areas that can be modulated with imaginary mental movement in healthy as well people with neuromuscular disabilities).

The motor imagery BMI framework typically consists of four integral components:

1. **Signal acquisition:** The initial stage involves the recording of neural signals while the person imagines the movements of the limbs, generally acquired using noninvasive methodologies (e.g., EEG).

2. **Feature extraction:** Following signal acquisition, signal processing techniques are applied to extract salient features from the neural patterns associated with specific cognitive processes or intentions.

3. **Feature translation:** This translation phase interprets the user's cognitive intent, converting it into actionable instructions for external devices.

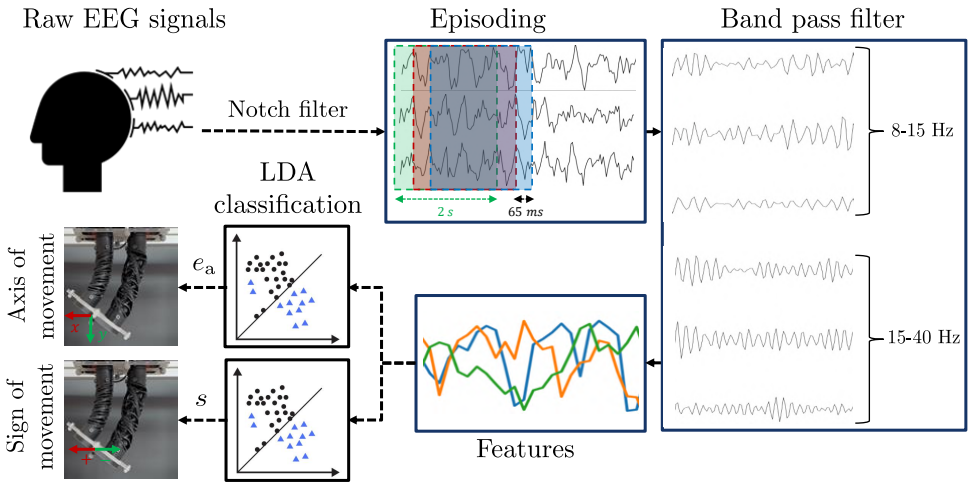


Figure 7.4: EEG data processing pipeline: The EEG data is acquired in real-time, pre-processed, and divided into episodes and subbands. Next, we extract power features and pass them to two LDA classifiers: the first outputs the axis of movement (for example, moving along the x- or y-axis), and the second provides the sign of movement (for example, positive or negative movement along the active axis). These commands are then used to move the attractor in Cartesian space.

4. **Device output:** The culmination of the BMI process is the application of the interpreted commands to external devices.

As detailed further in Sec. 7.3.2, we leverage the difference in signals when imagining motor actions vs. rest state to control the sign of movement. The active axis of movement can be switched by clenching the jaw.

7.2.2 Planning Attractors with Brain Signals

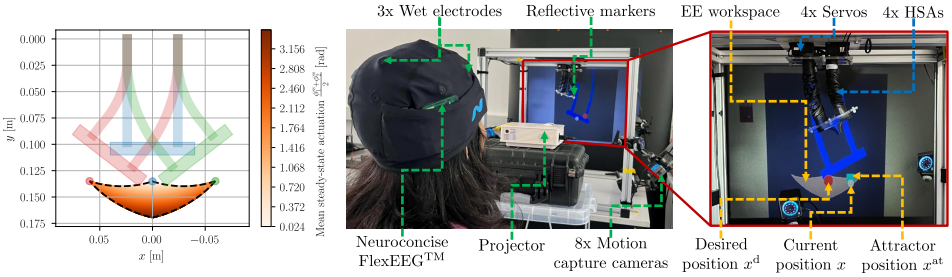
Our brain signal processing pipeline provides us with two pieces of information at each time step k: i) the unit vector $e_a(k) \in \{[1,0]^\top, [0,1]^\top\}$ corresponding to the current active axis of movement and ii) the sign of movement $s(k) \in \{-1, 1\}$. We use $e_a(k)$ and $s(k)$ to incrementally steer a virtual attractor defined in operational space $x^{at} \in \mathbb{R}^2$ as follows

$$u(k) = s(k)\, e_a(k) \in \mathbb{R}^2, \quad x^{at}(k) = x^{at}(k-1) + \Delta_x\, u(k), \tag{7.1}$$

where $\Delta_x \in \mathbb{R}^+$ is a tunable constant influencing the velocity of the attractor movement. Later, we will shape the potential field with a computational controller such that the attractor becomes a globally asymptotically stable equilibrium (see Section 7.2.3).

7.2.3 Computational Controller

We adopt the Cartesian impedance controller from Sec. 6.3.2 to shape the potential field with a computational controller such that the attractor x^{at} becomes a globally asymptotically



(a) Operational workspace (b) Experimental setup

Figure 7.5: **Panel (a:** Operational workspace of a HSA robot with attached end-effector: the color displays the mean steady-state actuation $\frac{\phi_1^{ss}+\phi_2^{ss}}{2}$ necessary for the end-effector to remain at the position. Additionally, we visualize three example shapes: the straight configuration with $\phi^{ss} = (0,0)$ (blue), maximum clockwise bending with $\phi^{ss} = (3.49,0)$ rad (red), and maximum counter-clockwise bending with $\phi^{ss} = (0,3.49)$ rad (green). **Panel (b):** The HSA robot is mounted platform-down to a motion capture cage with 8x Optitrack PrimeX 13 cameras, which track the 3D pose of the platform (i.e., the end-effector). A Dynamixel MX-28 servo actuates each of the four HSAs. We project a rendering of the current (white dot) and desired (red dot) end-effector position, the attractor (green square), and the operational workspace (grey area) onto the black screen in the background. The study subject wears a cap with the Neuroconcise FlexEEG sensor, and we acquire the data from three electrodes connected to the motor cortex.

stable equilibrium

$$\begin{aligned}\tau = J^{\top}(q)\left(K_x\left(x^{at}-x\right)-D_x\,\dot{x}\right)+G(q)+K\left(q-q^{0}\right)\\ +J^{\top}(q)\,J_M^{+\top}(q)\,D\,\dot{q}+J^{\top}(q)\,\mu(q,\dot{q})\left(I_3-J_M^{+}(q)J(q)\right)\dot{q}.\end{aligned} \tag{7.2}$$

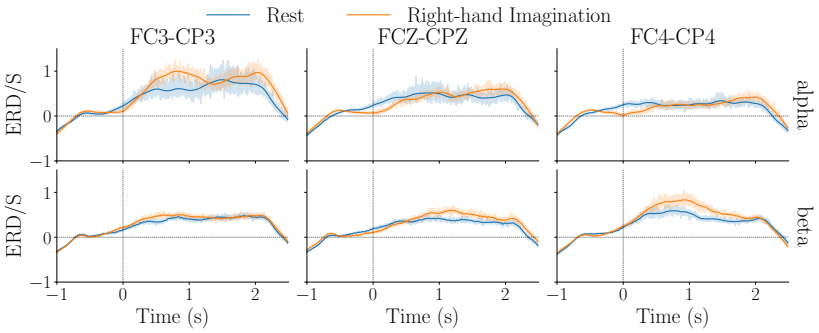
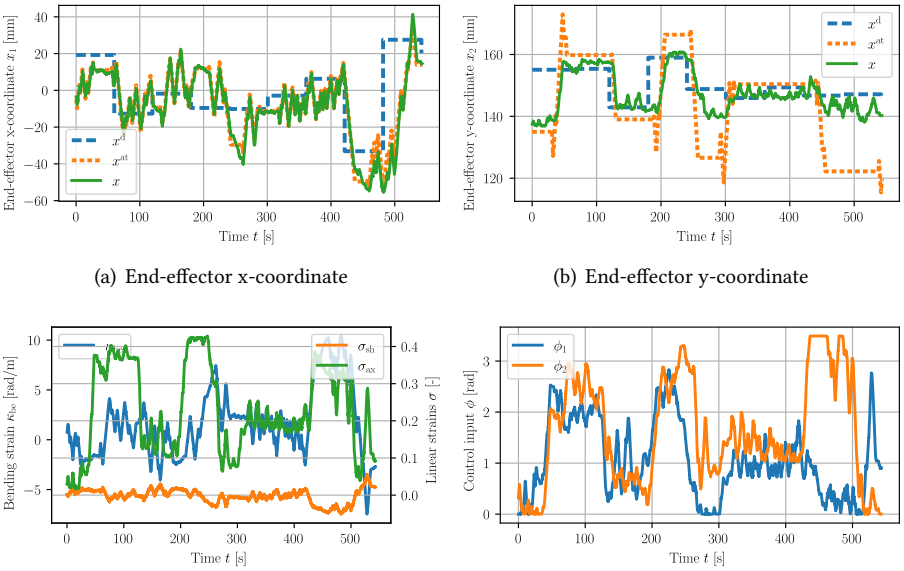


Figure 7.6: ERD/S (overall average) over a time period of 2.5 s of training data for right-hand Imagination v/s rest state including the Alpha and Beta bands of the EEG signals where the cue is presented at 0 s. We plot the data of three sensors (i.e., channels): FC3-CP3 (left), FCZ-CPZ (middle), and FC4-CP4 (right).



(a) End-effector x-coordinate

(b) End-effector y-coordinate

(c) Configuration q

(d) Control input ϕ

Figure 7.7: Experimental results for tracking a reference trajectory of nine-step functions with motor imagery. **Panel (a) & (b):** The x/y-coordinate of the end-effector position with the solid line denoting the actual position, the dotted line the attractor position, and the dashed line the reference (i.e., the setpoint). **Panel (c):** The evolution of the configuration. **Panel(d):** The saturated planar control inputs.

7.3 Experiments

7.3.1 Experimental Setup

In the following, we detail the EEG data processing procedure (see also Fig. 7.4) and present our experimental setup, which is annotated in Fig. 7.5(b).

EEG Data Processing

We integrate the 3-channel flexEEG Neuroconcise device with the OpenVibe software to acquire the EEG data and process it in real-time. This configuration facilitates data recording and cue presentation. We process the EEG signals at a sampling frequency of 125 Hz with a pipeline that involves three bi-polar channels around the motor cortex: FC3-CP3, FCZ-CPZ, and FC4-CP4. After a notch filter of 50 Hz, we apply Independent Component Analysis (ICA) to extract three independent components from the recorded EEG data, which is represented by the equation $S(t) = W\,X(t)$, where $S(t)$ are the extracted independent components and W represents the unmixing matrix, allowing us to separate eye blink artifacts in EEG signals, which is critical for enhancing the accuracy. Subsequently, we apply a Butterworth filter bank to isolate specific frequency bands of interest, including 8-15 Hz and 15-42 Hz. This enhances the ability to analyze EEG data by isolating and examining different frequency band components within the EEG signals. Once the signals are filtered in sub-bands and epoched with a duration of 2 s and time interval of 0.065 s, the features are extracted by the log of the power: $L_i(t) = \log(P_i(t))$, where the power $P_i(t) = |E_i(t)|^2$ is represented by square of magnitude of the EEG signal $E_i(t)$ at time instance t. These features are then provided to a classifier, which we select as LDA due to its simplicity (Lotte, 2014).

We implement a second classifier with the same pipeline, where jaw clenching is provided as a muscle artifact that is classified v/s raw EEG data.

Robotic System

We consider a robot consisting of four HSA rods, which were 3D printed via digital projection lithography from the photopolymer resin Carbon FPU 50. Each HSA rod is electrically actuated by a Dynamixel M-28 servo up to a maximum twist angle of $\phi_{\max} = 3.49$ rad. The robot is attached platform-down to a motion capture cage with eight Optitrack Prime X13 cameras tracking at 200 Hz the pose of reflective markers attached to the end-effector of the HSA robot. We estimate the current Cartesian-space velocity of the end-effector with a Savitzky-Golay filter. Subsequently, we leverage a closed-form expression of the inverse kinematics of a CS model (Stölzle et al., 2024b) to compute the current configuration q of the robot. We render an image of the current shape of the robot together with the present end-effector position (white dot), the attractor planned by the user (green square), the operational workspace (grey, see also Fig. 7.5(a)) and if applicable, the goal position (red dot). We specify the currently active axis of movement e_a with a double arrow and project the resulting image onto a black screen in the background of the motion capture cage. The robot is operated with a ROS2 software framework[3]. We receive the predicted and classified brain signals via TCP at a frequency of 18 Hz and move the attractor subsequently with $\Delta_x = 0.2$ mm. We evaluate the Cartesian impedance controller using the

[3] https://github.com/tud-phi/ros2-hsa

gains $K_p = 300\,\mathrm{N/m}$, $K_d = 1.5\,\mathrm{Ns/m}$ at a frequency of 50 Hz and finally send the desired motor positions to the servos.

The entire control pipeline from measuring EEG signals to sending the actuation commands to the motors exhibits a maximum latency of (i.e., is upper-bounded by) 130 ms.

7.3.2 BMI Protocol

In the following, we will describe the protocol for collecting the motor imagery dataset, training the EEG classifiers, and mapping classifier outputs into actionable robot commands.

Training Protocol

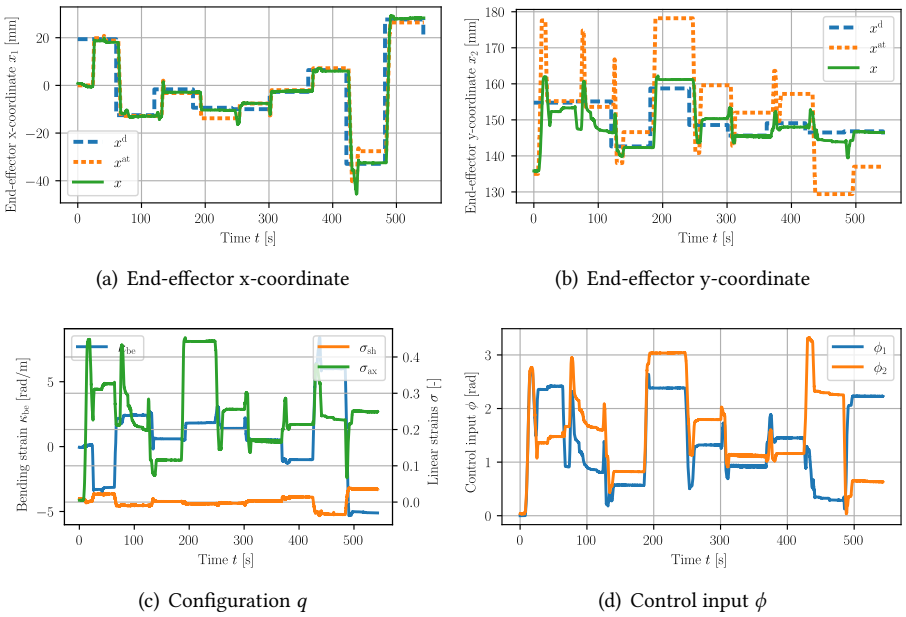
The participant is given brief instructions about motor imagery signals. We follow the Graz-BCI (Roc et al., 2021) paradigm, which assists with training, where the display of the cue instructs the participant to perform imagination of movements: when a left arrow appears, the participant is asked to rest and not focus on motor movement. When the right arrow appears on the screen, the participant is asked to imagine the motor activity from the dominant hand (here, the right hand), such as grasping or squeezing an object. The training protocol for right-hand motor imagery v/s rest demands fifty cues per class. We similarly collect data for the second classifier by asking participants to clench their jaw, which can be detected as muscular artifacts (vs. no muscular movement) in the EEG signals. At the end of the trial, we train both classifiers (see Sections 7.2.1 and 7.3.1 for more information) and repeat the procedure until an accuracy of 75 % is obtained for right-hand motor imagery v/s rest and accuracy of 85 % for jaw clench artifact v/s raw EEG. We evaluate the trained motor imagery classifier on the test set and report the results in Tab. 7.1.

		Prediction	
		Rest	Right-handed motor imagery
Label	Rest	70.7 %	29.3 %
	Right-handed motor imagery	33.3 %	68.7 %

Table 7.1: Confusion matrix for the LDA classifier at predicting right-hand motor imagery vs. rest based on the preprocessed EEG signals evaluated on a test set.

Online Robot Control

Now, the participant operates the HSA robot in real-time, with both classifiers being executed online. Moreover, we map the outputs of the classifier into actionable commands: we initialize the x-axis as the active axis of movement (i.e., $e_a = [1, 0]$). When the first classifier detects jaw clenching for at least 80 % of samples over a duration of 2.8 s, we switch to the y-axis: $e_a = [0, 1]$ and vice-versa. If we do not detect any jaw-clenching artifacts, we evaluate the output of the second classifier in parallel: if there is motor imagery activity identified in the EEG classifier, the attractor will be moved in the positive direction (i.e., $s = 1$) along e_a. In contrast, if the EEG signals are classified as the participant being at rest, $s = -1$ (i.e., movement in the negative direction).



(a) End-effector x-coordinate

(b) End-effector y-coordinate

(c) Configuration q

(d) Control input ϕ

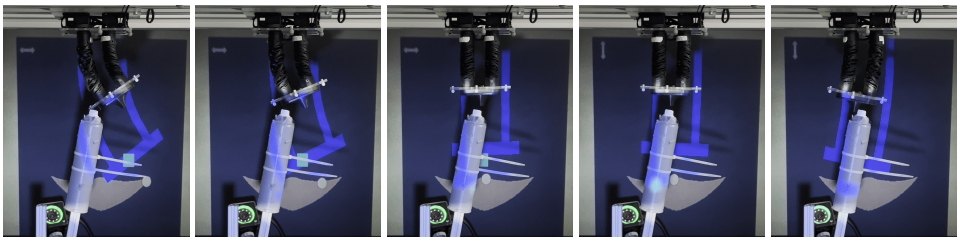
Figure 7.8: Experimental results for tracking a reference trajectory of nine step functions with keyboard inputs. **Panel (a) & (b):** The x/y-coordinate of the end-effector position with the solid line denoting the actual position, the dotted line the attractor position, and the dashed line the reference (i.e., the setpoint). **Panel (c):** The evolution of the configuration. **Panel(d):** The saturated planar control inputs.

7.3.3 Setpoint Regulation

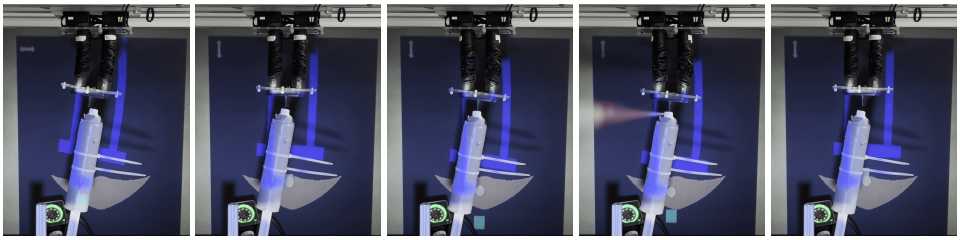
We randomly generate nine setpoints $x^{\mathrm{d}}(t) \in \mathbb{R}^2$ within the operational workspace of the robot (see Fig. 7.5(a) and display them as a red circle to the user over a duration of 540 s. The user can freely move the attractor to reach and keep the end-effector at the setpoint. In addition to the motor imagery-based control, we conduct experiments where the subject controls the robot with a keyboard. As a keyboard is a very efficient and precise HMI (Mahmud et al., 2020; Vasiljevic et al., 2016), this represents an upper bound on the performance we could expect from the Brain-Computer Interface (BCI). Here, the space button is used to switch that axis of movement e_{a} and positive/negative movement can be controlled via the left/right arrows.

7.3.4 Interacting with the Environment on a Real-World Task

We consider the ADL task of releasing hairspray by actuating the button of its container with the HSA robot's end-effector (see Fig. 7.10). For successful execution, the end-effector must be very stiff in the normal direction of the contact. On the other hand, we might want to benefit from the physical intelligence of the system by being relatively flexible in the tangential direction. Therefore, we first define the perpendicular stiffness $k_{\perp} = 500\,\mathrm{N/m}$ and the tangential stiffness as $k_{\parallel} = 50\,\mathrm{N/m}$. We assume that the normal direction of the contact can be described by the polar angle $\theta_{\perp}$ (with respect to the x-axis). We envision



(a) $t = 0\,\mathrm{s}$ (b) $t = 12\,\mathrm{s}$ (c) $t = 24\,\mathrm{s}$ (d) $t = 36\,\mathrm{s}$ (e) $t = 48\,\mathrm{s}$



(f) $t = 60\,\mathrm{s}$ (g) $t = 72\,\mathrm{s}$ (h) $t = 84\,\mathrm{s}$ (i) $t = 96\,\mathrm{s}$ (j) $t = 108\,\mathrm{s}$

Figure 7.9: Sequence of stills for completing a basic Activity of Daily Living (ADL) by controlling the robot with EEG-based motor imagery. *Note:* Fig. 7.9(i) is edited for improved contrast.

that in the future, the user can adjust such stiffness characteristics online via a BMI system similar to (Schiatti et al., 2017). In this chapter, however, we estimate by visual inspection that $\theta_\perp = 1.31\,\mathrm{rad}$. The Cartesian stiffness matrix in global coordinates is then given by $K_x = R(\theta_\perp)\,\mathrm{diag}(k_\perp, k_\parallel)\,R(\theta_\perp)^\top$ where $R(\theta_\perp) \in SO(2)$ is the rotation matrix between global and contact frames.

7.3.5 Evaluation Metrics

In the following, we will introduce and define a few metrics that help us assess the performance of the approach.

Event-Related Synchronization/De-Synchronization:

We apply Event-Related Desynchronization (ERD) / Event-Related Synchronization (ERS) (Pfurtscheller and Da Silva, 1999) to demonstrate the difference between EEG signals when the participant imagines right-hand movement vs. rest, i.e., no activity. ERD/ERS corresponds to a shift in power during imagination with respect to a baseline. It is defined by

$$\mathrm{ERD/ERS}(t, f) = \frac{P(t, f) - P_{\mathrm{base}}(f)}{P_{\mathrm{base}}(f)}, \tag{7.3}$$

where $\mathrm{ERD/ERS}(t, f)$ represents the ERD or ERS at a specific time t and frequency f, $P(t, f)$ stands for the power of brain activity during imagination, and $P_{\mathrm{base}}(f)$ denotes the baseline power.

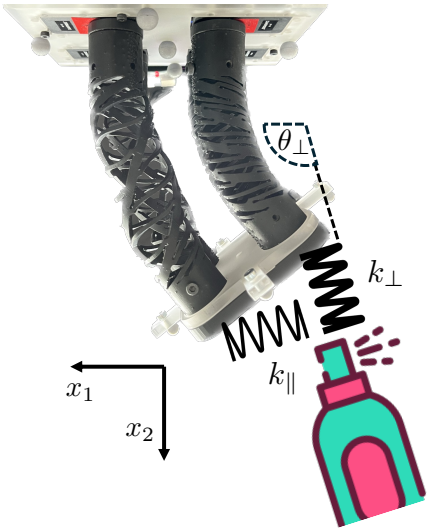
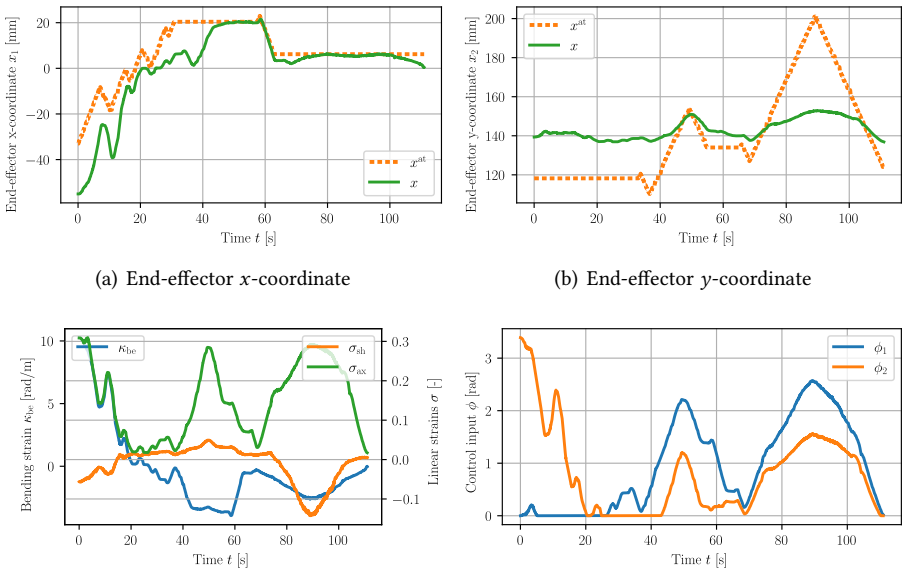


Figure 7.10: Cartesian impedance shaping when interacting with the environment, for example, while performing ADL.

7.3.6 Step Response Metrics

For the task of setpoint regulation, we analyze primarily two aspects: (a) is the participant able to reach the proximity of setpoint within the (generously) allotted time of 60 s? We define the proximity of the setpoints as $\|x^{\mathrm{d}} - x(t)\|_2 \leq 2\,\mathrm{mm}$. And (b) what is the response time for reaching the proximity of the setpoint for the first time?



(a) End-effector x-coordinate

(b) End-effector y-coordinate

(c) Configuration q

(d) Control input ϕ

Figure 7.11: Experimental results for completing a basic Activity of Daily Living (ADL) by controlling the robot with EEG-based motor imagery. **Panel (a) & (b):** The x/y-coordinate of the end-effector with the solid line and dotted lines denoting the actual and attractor position, respectively. **Panel (c):** The evolution of the configuration. **Panel(d):** The saturated planar control inputs.

7.4 Results and Discussion

First, we analyze the ERD/ERS behavior with respect to rest vs. motor imaginations in Fig. 7.6. It is evident that the baseline of rest remained the same in both scenarios when the participant did not perform motor imagery, but as soon as the cue is presented at 0.0 s, a shift in power for the right-hand motor imagery with comparison to rest state is noticeable.

We present the results for setpoint regulation employing motor imagery in Fig. 7.7. We observe that the participant can reach the proximity of the setpoint within the allotted time of 60 s six out of nine times (i.e., 66.6 %). For the successful steps, the average response time is 21.5 s. However, as our protocol does not contain a command to let the attractor rest, it is challenging to keep the end-effector at the setpoint and we observe oscillations, particularly with respect to the x-coordinate. From the results in Fig. 7.8, it is evident that a keyboard is a superior HMI. It only takes the participant two setpoints to get familiar with the interface, and afterwards, the performance displayed is excellent. However, for instance, for people with limb impairment, or people with Spinal Cord Injury (SCI), such an interface is inaccessible and, therefore, only represents an upper bound of what we strive to achieve with BCI systems.

Finally, we consider the ADL task of releasing hair spray using the end-effector of the HSA robot. We present a sequence of stills in Fig. 7.9 and plots of the entire sequence in Fig. 7.11. Already during the first attempt, the participant can steer the end-effector toward

the button, apply force, and release the fluid within 86 s. The impedance of the controller is clearly visible in Fig. 7.11(b) when the manipulator is in contact with the object at time 74 s to 104 s. Also, we noticed that the end-effector does not need to be perfectly aligned with the center of the button and can still complete the task successfully due to the compliance of the closed-loop system in the tangential direction.

We noticed that the variability of setting up the EEG device on each study participant and the EEG sensor noise caused by external factors (e.g., floor vibrations) still pose a considerable challenge for deploying motor imagery-based tools in practice. Furthermore, subject-specific factors such as the ability to focus on imagining motor actions, mental tiredness, etc., significantly affected the performance (e.g., classification accuracy, setpoint tracking error).

7.5 Conclusion

In this chapter, we proposed to combine motor imagery-based BMI systems with continuum soft robots. This symbiosis promises the safe and compliant operation of robots that can assist people with limb impairments in their daily lives. While the binary motor imagery classifier achieves an accuracy of only $\approx 70\,\%$, we demonstrated experimentally its effectiveness through assistance with an activity of daily living and safe operation. As demonstrated in the ADL experiment, the physical intelligence of the soft robot can compensate for errors and deviations in the output of the BMI classifier.

For future work, we recommend conducting a user study with a diverse group of non-expert participants and incorporating a broader range of soft robots. Connected, it would be interesting to apply this methodology to soft robots with more DOFs, such as the Helix soft robot (Guan et al., 2023). Finally, utilizing SOTA EEG classifiers like CTNet (Zhao et al., 2024a) could improve classification accuracy and ultimately enable full 3D spatial brain control of the soft robot's task space motion—including the addition of a *rest* mode to keep the end-effector at a fixed pose.

Afterword

In this chapter, we introduced a BMI protocol for soft robots using motor imagery with wearable, few-channel EEG devices. However, controlling the soft robot via motor imagery demands the user's full attention and can become highly exhausting after a few minutes. Consequently, our goal is to develop fully autonomous (soft) robots. Rather than conveying low-level end-effector setpoints, we aim to communicate high-level task assignments. In Chapter 12, we take a first step toward this objective by presenting a strategy for learning compliant and stable motion policies for periodic movements from demonstration. This approach enables the user to demonstrate the required (periodic) movement once, allowing the (soft) robot to perform the task autonomously in a compliant, stable, and safe manner.

8

Backstepping Control of Pneumatic Piston-Driven Soft Robots

Foreword. So far, our models and controllers, as presented in Chapters 5 and 6, have neglected the dynamics of the actuator (e.g., the dynamics of the servo motor driving the HSA rods). The underlying assumption is that the controller within the actuator is sufficiently fast and accurate to regulate the actuator towards the desired actuation (i.e., the control input). This approach is akin to a cascade controller, where the output of the outer control loop (the soft robot controller) serves as a reference to the inner control loop (the actuator controller), which operates at a faster rate. However, literature has shown that cascade controllers fail when the inner closed-loop system cannot track its reference accurately. While the mentioned assumption may be acceptable in certain scenarios, such as when the actuator is over-dimensioned (e.g., a fast-moving servo paired with a relatively slow-moving soft robot), it inherently limits our pursuit of more practical, fast-moving soft robots. It prevents us from fully exploiting the dynamics of the soft robot system. In this chapter, we model in Section 8.2 the integrated dynamics of a pneumatically-actuated soft robot where pressure is generated using pneumatic pistons. Subsequently, in Section 8.3, we exploit this dynamical model within a backstepping controller that considers the dynamics of the pneumatic pistons.

This chapter is based on 🗎 ***M. Stölzle**, C. Della Santina (2021). Piston-driven pneumatically-actuated soft robots: Modeling and backstepping control. IEEE Control Systems Letters, 6, 1837-1842.* (Stölzle and Della Santina, 2021).

C.D.S. conceived the project, provided funding, derived the backstepping controller, including proof of stability, and supervised the project. M.S. derived the model for the fluid volume contained in a planar PCC soft robot as a function of the configuration, implemented the controller, planned and carried out the simulations, and wrote the manuscript. C.D.S. revised the manuscript.

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q} = Q$$

Abstract. *Actuators' dynamics have been mostly neglected when devising feedback controllers for continuum soft robots since the problem under the direct actuation hypothesis is already quite hard to solve. Directly considering actuation would have made the challenge too complex. However, these effects are, in practice, far from being negligible. The present chapter focuses on model-based control of piston-driven pneumatically-actuated soft robots. We propose a model of the relationship between the robot's state, the acting fluidic pressure, and the piston dynamics, which is agnostic to the chosen model for the soft system dynamics. We show that backstepping is applicable even if the feedback coupling of the outer on the inner subsystem is not linear. Thus, we introduce a general model-based control strategy based on backstepping for soft robots actuated by fluidic drive. As an example, we derive a specialized version for a robot with piecewise constant curvature.*

8.1 Introduction

Accurate low-dimensional models of the continuum dynamics have been thoroughly investigated in recent years (Faure et al., 2012; Grazioso et al., 2019; Sadati et al., 2021), serving as the base for model-based controllers (Boyer et al., 2020; Della Santina et al., 2023). In comparison, researchers have devoted little or no attention to modeling the actuator dynamics, despite this being far from a negligible effect in practice, in particular for pneumatic actuation. The lack of models pairs with the scarcity of model-based dynamic controllers. Existing strategies only rarely reason on the actuators' dynamics, if not through simple heuristics. For example, (Della Santina et al., 2020c; Lindenroth et al., 2016; Marchese and Rus, 2016) use a combination of PID control and inversion of quasi-static linear approximations to compensate for the actuators' dynamics. This strategy may present clear limitations in terms of performance and stability assessment.

As model-based control of soft robots becomes a mature discipline, the need for general ways of dealing with actuators' dynamics becomes more pressing. In this chapter, we deal with this challenge by following a backstepping approach, which is an established strategy to deal with dynamical systems with a triangular structure. A pneumatic model based on the ideal gas law is derived, and the pneumatic actuation system is compensated in a quasi-static fashion in Falkenhahn et al. (2016). Recent work by Wang et al. (2019) uses backstepping for control of a continuum soft bending arm. Although interesting, the work is limited because it targets a linear model of a single DOF. Similarly, Franco et al. (2021) derive an energy-based control scheme for pneumatic manipulators while using a backstepping-based controller for comparison purposes. Both pieces of work focus on pneumatic actuation with valves and thus cannot be immediately applied to systems actuated with fluidic drive cylinders.

To conclude, this chapter targets the dynamic control of piston-driven pneumatic-actuated soft robots (see, for example, Fig. 8.2, 8.3). We provide general strategies for (i) augmenting existing dynamic models of soft robots through a description of pneumatic actuation, (ii) controlling these systems via model-based feedback. As an example, we specialize the model to planar soft robots satisfying the PCC assumption (Della Santina et al., 2020c), including the proposal of a kinematic model for the air volume in the chambers, and the controller for the set-point regulation of configuration. In this context, we also propose a simplified, potential coupling-aware PID-like controller. We provide simulations showing the effectiveness of both strategies.

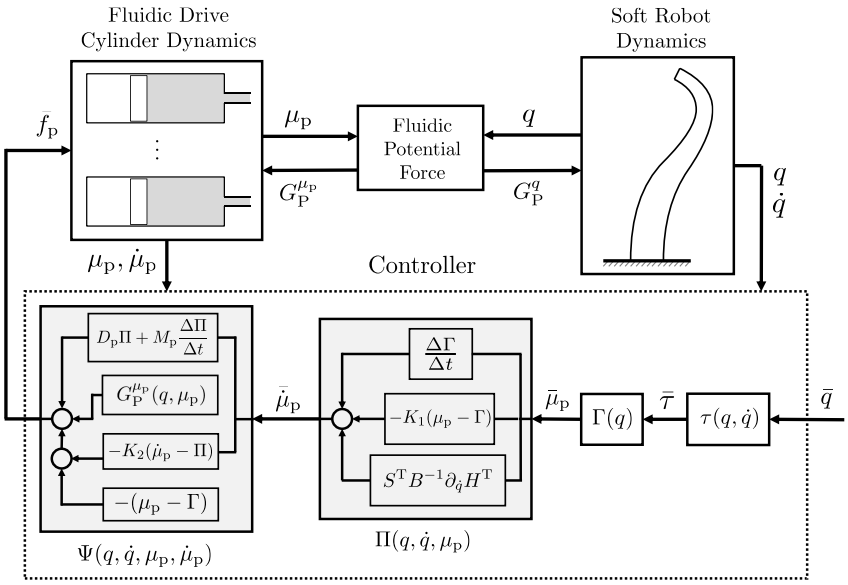


Figure 8.1: Schematic block diagram of the proposed nonlinear backstepping controller for a pneumatically-actuated soft robot. The approach considers both the fluidic drive cylinder and the soft system dynamics. It is agnostic to the chosen soft system controller in configuration-space $\tau(q, \dot{q})$.

8.2 Dynamic Model

We consider the robot made by a sequence of $n_\mathrm{S} \in \mathbb{N}_+$ segments. Each segment is described with $n_\mathrm{D} \in \mathbb{N}_+$ configuration variables by using one of the many modeling techniques being developed in the state-of-the-art (Boyer et al., 2020; Faure et al., 2012; Grazioso et al., 2019; Sadati et al., 2021). We denote with $n_\mathrm{q} = n_\mathrm{S} n_\mathrm{D} \in \mathbb{N}_+$ the total number of configuration variables, which also represents the approximated DOFs of the soft arm. Although we show planar kinematic relations for the PCC-case in Figure 8.3 as an example, the dynamic model derived in this section is agnostic to the chosen kinematic approximation.

All kinematic modeling techniques produce multi-body dynamics of the unactuated soft robot as follows (Della Santina et al., 2023)

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + K(q) + D(q, \dot{q}) = 0, \tag{8.1}$$

where $q \in \mathbb{R}^{n_\mathrm{q}}$ describes the configuration of the robot in generalized coordinates, $M(q) \in \mathbb{R}^{n_\mathrm{q} \times n_\mathrm{q}}$ the inertial matrix, $C(q, \dot{q}) \in \mathbb{R}^{n_\mathrm{q} \times n_\mathrm{q}}$ contains the Coriolis and Centrifugal forces and $G(q) \in \mathbb{R}^{n_\mathrm{q}}$ compensates for the gravitational effects. The elastic (restoring) forces are captured in the matrix $K \in \mathbb{R}^{n_\mathrm{q}}$ and the natural damping is represented by $D(q, \dot{q}) \in \mathbb{R}^{n_\mathrm{q}}$.

Each segment is actuated through a set of n_C dedicated chambers. Adapting the pressure in a chamber will lead to a different chamber volume, ultimately resulting in a changed configuration of the segment. Each chamber is connected to a dedicated piston, as shown in Figure 8.3. If more than one chamber is connected to the same piston, it can be considered to be the same chamber for the sake of this chapter. These hypotheses are not paramount,

but they are instrumental in making the notation easier. Accordingly, the total number of pistons is described with $n_{\mu_\mathrm{p}} = n_\mathrm{S} n_\mathrm{C}$. Please note that if $n_\mathrm{C} = n_\mathrm{D}$, the model of the soft robot is fully-actuated, if $n_\mathrm{C} > n_\mathrm{D}$ it is over-actuated, and with $n_\mathrm{C} < n_\mathrm{D}$ under-actuated respectively. The dynamics of the piston, when not interacting with the fluid, can be easily written as being

$$M_\mathrm{p}\ddot{\mu}_\mathrm{p} + D_\mathrm{p}\dot{\mu}_\mathrm{p} + G_\mathrm{P}^{\mu_p} = f_\mathrm{p}, \tag{8.2}$$

where $\mu_\mathrm{p} \in \mathbb{R}^{n_{\mu_\mathrm{p}}}$ denoting the displacement of every piston from the zero-volume configuration, $M_\mathrm{p} \in \mathbb{R}^{n_{\mu_\mathrm{p}} \times n_{\mu_\mathrm{p}}}$ the mass matrix of the piston system, $G_\mathrm{P}^{\mu_p} \in \mathbb{R}^{n_{\mu_\mathrm{p}}}$ describing the conservative force caused by the compressed fluid acting on the pistons, and $D_\mathrm{p} \in \mathbb{R}^{n_{\mu_\mathrm{p}} \times n_{\mu_\mathrm{p}}}$ the damping matrix of the piston system. As the piston system is fixed to the ground and connected via tubing to the robot chambers, note that the gravity force here is constant, so w.l.o.g., we consider it to be zero (or alternatively as being compensated by a constant offset in f_p).

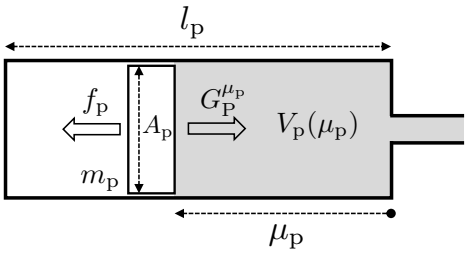


Figure 8.2: Fluidic drive cylinder parameters for a piston of mass m_p, length l_p and cross-sectional area A_p: f_p describes the actuation force while $G_\mathrm{P}^{\mu_\mathrm{p}}$ is the conservative force applied by the compressed fluid on the cylinder. μ_p represents the actuators' state variable. These pneumatic pistons could be, for example, actuated by current-controlled DC motors or linear electric actuators (Marchese et al., 2014).

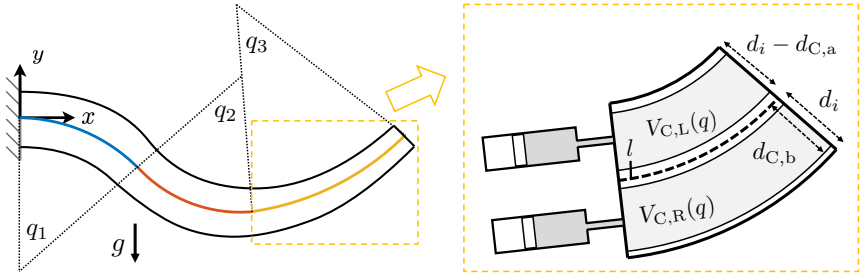


Figure 8.3: Shape regulation under PCC approximation - **Left:** A planar soft robot consisting of three segments, each modeled to have constant curvature **Right:** Model parameters for fluidic volume in soft segment chambers. Each chamber is actuated independently by a fluidic drive cylinder connected through tubing.

In the first approximation, we model the compressible fluid (typically air) as an ideal gas. Furthermore, we consider the process to be isothermal, and no fluid exchange with the external world is happening. We neglect the volume of fluid in any tubes connecting

the pistons with the chambers. The overall volume of the fluid can be evaluated as

$$V(q,\mu_\mathrm{p}) = V_\mathrm{C}(q) + V_\mathrm{p}(\mu_\mathrm{p}) = V_\mathrm{C}(q) + A_\mathrm{p}\mu_\mathrm{p}, \tag{8.3}$$

where $V(q,\mu_\mathrm{p}) \in \mathbb{R}^{n_{\mu_\mathrm{p}}}$ describes the total volume of fluid stored in the system, $V_\mathrm{C}(q) \in \mathbb{R}^{n_{\mu_\mathrm{p}}}$ the volume of fluid in each chamber and $V_\mathrm{p}(q) \in \mathbb{R}^{n_\mathrm{S} n_\mathrm{p}}$ the volume in the piston with $A_\mathrm{p} \in \mathbb{R}^{n_{\mu_\mathrm{p}}}$ the cross-sectional area of every piston. We will present an example of analytical derivation of $V_\mathrm{C}(q)$ in Section 8.4.1. For now, we consider it known.

The total energy stored in the system due to fluid compression is

$$\begin{aligned}\mathcal{U}_\mathrm{fluid}(q,\mu_\mathrm{p}) &= \sum_{j=1}^{n_{\mu_\mathrm{p}}} \int_{V_{j,0}}^{V_j(q_i,\mu_{\mathrm{p},j})} -\left(p_j(v) - p_\mathrm{atm}\right) \mathrm{d}v \\ &= \sum_{j=1}^{n_{\mu_\mathrm{p}}} -\alpha_{\mathrm{air},j} \left(\ln \frac{V_j(q_i,\mu_{\mathrm{p},j})}{V_{j,0}} - \frac{V_j(q_i,\mu_{\mathrm{p},j})}{V_{j,0}} + 1 \right),\end{aligned} \tag{8.4}$$

where $V_j(q_i,\mu_{\mathrm{p},j}) = \frac{\alpha_{\mathrm{air},j}}{p_j(q_i,\mu_{\mathrm{p},j})}$ represents the total fluidic volume in the system of chamber and piston j in segment i. We assume that this fluid system is filled with air at atmospheric pressure p_atm with an initial volume of $V_{j,0} = V_j(0, l_\mathrm{p})$ (e.g., straight robot configuration and with fully extended pistons). This lets us find an expression for α_air:

$$\alpha_{\mathrm{air},j} = n_j R T = p_\mathrm{atm} V_{j,0} = p_j(q,\mu_\mathrm{p}) V_j(q,\mu_\mathrm{p}). \tag{8.5}$$

The force exerted on the ith segment of the robot by the fluid is

$$G^\mathrm{q}_{\mathrm{P},i}(q_i,\mu_{\mathrm{p},j}) = \partial_{q_i} \mathcal{U}_\mathrm{fluid}(q,\mu_\mathrm{p}) = -\partial_{q_i} V_{C,j} \left(p_j(q_i,\mu_{\mathrm{p},j}) - p_\mathrm{atm}\right), \tag{8.6}$$

Similarly, the force applied on the jth piston by the fluid is

$$G^{\mu_\mathrm{p}}_{\mathrm{P},j}(q_i,\mu_{\mathrm{p},j}) = \partial_{\mu_{\mathrm{p},j}} \mathcal{U}_\mathrm{fluid}(q,\mu_\mathrm{p}) = -A_{\mathrm{p},j} \left(p_j(q_i,\mu_{\mathrm{p},j}) - p_\mathrm{atm}\right), \tag{8.7}$$

The overall dynamic model is

$$\begin{aligned} M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + K(q) + D(q,\dot{q}) + G^\mathrm{q}_\mathrm{P}(q,\mu_\mathrm{p}) &= 0, \\ M_\mathrm{p}\ddot{\mu}_\mathrm{p} + D_\mathrm{p}\dot{\mu}_\mathrm{p} + G^{\mu_\mathrm{p}}_\mathrm{P}(q,\mu_\mathrm{p}) &= f_\mathrm{p}, \end{aligned} \tag{8.8}$$

which is always underactuated. Note that this structure is similar to classic flexible joint robots under Spong's approximation (Della Santina, 2021) due to the fact that the fluidic drive cylinders are fixed to the ground. Two of the major differences making the control problem harder are that that $n_{\mu_\mathrm{p}} \neq n_\mathrm{q}$, and that G^q_P and $G^{\mu_\mathrm{p}}_\mathrm{P}$ are not linear. The latter renders the feedback coupling of the outer on the inner subsystems non-affine. In the rest of the chapter, we will use the following definitions to simplify the notation

$$\begin{aligned} f(q,\dot{q}) &= -B^{-1}(q)\left(C(q,\dot{q})\dot{q} + G(q) + K(q) + D(q,\dot{q})\right), \\ g(q,\mu_\mathrm{p}) &= -B^{-1}(q)\left(G^\mathrm{q}_\mathrm{P}(q,\mu_\mathrm{p})\right). \end{aligned} \tag{8.9}$$

8.3 Backstepping Control

This section discusses the main contribution of this chapter, a backstepping-based approach to generalize controllers $\Gamma(q, \dot{q})$ designed in the directly actuated case, to systems that can be modeled through (8.8). We suppose that we have access to $\Gamma(q, \dot{q})$ controlling the piston position μ_{p}. Next, we perform backstepping twice to the controllers of the piston velocity $\dot{\mu}_{\mathrm{p}}$ and the piston actuation force f_{p} and prove the stability of each controller with Lyapunov arguments. The derived model-based control approach assumes that all model parameters are known, and all states are measurable (namely the configuration q, its time derivative $\dot{q}$, the piston position μ_{p} and the piston velocity $\dot{\mu}_{\mathrm{p}}$), and that there are no disturbances or model uncertainties. We first introduce a Lemma, which will be instrumental to the proof of the main theorem. It allows us to relate an offset in the actuation space to a change in acceleration in the configuration space that is proportional to the offset.

Lemma 1. *The input field defined in* (8.9) *verifies*

$$g(q, \mu_{\mathrm{p,a}}) - g(q, \mu_{\mathrm{p,b}}) = -B^{-1}(q) S(q, \mu_{\mathrm{p,a}}, \mu_{\mathrm{p,b}})(\mu_{\mathrm{p,a}} - \mu_{\mathrm{p,b}}),$$

$\forall \mu_{\mathrm{p,a}}, \mu_{\mathrm{p,b}} \in \mathbb{R}^{n_{\mu_{\mathrm{p}}}}$ *and* $q \in \mathbb{R}^{n_{\mathrm{q}}}$ *with* $S \in \mathbb{R}^{n_{\mathrm{q}} \times n_{\mu_{\mathrm{p}}}}$ *defined via*

$$S_{ij} = \frac{A_{\mathrm{p},j} \alpha_{\mathrm{air},j} \partial_{q_i} V_{\mathrm{C},j}}{(V_{\mathrm{C},j}(q_i) + A_{\mathrm{p},j} \mu_{\mathrm{p,a},j})(V_{\mathrm{C},j}(q_i) + A_{\mathrm{p},j} \mu_{\mathrm{p,b},j})}.$$

Proof. We express the left term of the equality using (8.9)

$$g(q, \mu_{\mathrm{p,a}}) - g(q, \mu_{\mathrm{p,b}}) = -B^{-1}(q)(G^{\mathrm{q}}_{\mathrm{P}}(q, \mu_{\mathrm{p,a}}) - G^{\mathrm{q}}_{\mathrm{P}}(q, \mu_{\mathrm{p,b}})),$$

where we recognize the term $B^{-1}(q)$ appearing in the Lemma. The term between brackets can be adjusted by using (8.6)

$$\begin{aligned} G^{\mathrm{q}}_{\mathrm{P},i}(q, \mu_{\mathrm{p,a}}) - G^{\mathrm{q}}_{\mathrm{P},i}(q, \mu_{\mathrm{p,b}}) &= -\left(\sum_{j=1}^{n_{\mu_{\mathrm{p}}}} \frac{\alpha_{\mathrm{air},j} \partial_{q_i} V_{\mathrm{C},j}}{V_{\mathrm{C},j}(q_i) + A_{\mathrm{p},j} \mu_{\mathrm{p,a},j}} - \sum_{j=1}^{n_{\mu_{\mathrm{p}}}} \frac{\alpha_{\mathrm{air},j} \partial_{q_i} V_{\mathrm{C},j}}{V_{\mathrm{C},j}(q_i) + A_{\mathrm{p},j} \mu_{\mathrm{p,b},j}} \right), \\ &= \sum_{j=1}^{n_{\mu_{\mathrm{p}}}} \frac{A_{\mathrm{p},j} \alpha_{\mathrm{air},j} \partial_{q_i} V_{\mathrm{C},j} (\mu_{\mathrm{p,a},j} - \mu_{\mathrm{p,b},j})}{(V_{\mathrm{C},j}(q_i) + A_{\mathrm{p},j} \mu_{\mathrm{p,a},j})(V_{\mathrm{C},j}(q_i) + A_{\mathrm{p},j} \mu_{\mathrm{p,b},j})}, \\ &= \sum_{j=1}^{n_{\mu_{\mathrm{p}}}} S_{ij} (\mu_{\mathrm{p,a},j} - \mu_{\mathrm{p,b},j}) = S(\mu_{\mathrm{p,a}} - \mu_{\mathrm{p,b}}). \end{aligned} \tag{8.10}$$

The Lemma follows by simple factorization of the latter term. □

Thus, even if the robot side of the dynamics (8.8) is not affine in control when taking μ_{p} as input, still Lemma 1 provides some structure that we leverage in the next theorem. Assuming a soft robot controller $\tau(q, \dot{q})$ that regulates the soft robot to the setpoint $\bar{q}$ while neglecting actuator dynamics, Theorem 1 introduces—via the backstepping procedure—a fluidic piston force controller $f_{\mathrm{p}} = \Psi(q, \dot{q}, \mu_{\mathrm{p}}, \dot{\mu}_{\mathrm{p}})$ that ensures the closed-loop system converges to the setpoint. In particular, the system reaches the desired soft robot configuration $\bar{q}$ along with the corresponding steady-state piston position $\bar{\mu}_{\mathrm{p}} = \Gamma(q)$. As depicted in

Fig. 8.1, the function $\Gamma(q)$ maps positions from the configuration space to the actuation space, while the controller $\bar{\dot{\mu}}_\mathrm{p} = \Pi(q, \dot{q}, \mu_\mathrm{p})$ supplies piston velocity references that are then converted into piston force commands by $\Psi(q, \dot{q}, \mu_\mathrm{p}, \dot{\mu}_\mathrm{p})$. Here, $K_1 \succ 0 \in \mathbb{R}^{n_{\mu_\mathrm{p}} \times n_{\mu_\mathrm{p}}}$ is a proportional feedback gain applied to the error $(\mu_\mathrm{p} - \bar{\mu}_\mathrm{p})$ between actual and desired piston positions, and $K_2 \succ 0 \in \mathbb{R}^{n_{\mu_\mathrm{p}} \times n_{\mu_\mathrm{p}}}$ is a proportional feedback gain on the error $(\dot{\mu}_\mathrm{p} - \bar{\dot{\mu}}_\mathrm{p})$ between actual and desired piston velocities.

Theorem 1. *Suppose that a $\Gamma(q, \dot{q})$ exists s.t. the reduced system*

$$\ddot{q} = f(q, \dot{q}) + g(q, \Gamma(q, \dot{q})) \tag{8.11}$$

converges to a desired trajectory $\bar{q}(t)$, $\forall (q(0), \dot{q}(0)) \in \mathbb{R}^{2n_\mathrm{q}}$. Suppose that the convergence is proven by Lyapunov arguments through the function $H(q, \dot{q})$. Then the closed loop of the full system (8.8) *and the controller*

$$\begin{aligned} \bar{f}_\mathrm{p} &= \Psi(q, \dot{q}, \mu_\mathrm{p}, \dot{\mu}_\mathrm{p}) = G_\mathrm{P}^{\mu_\mathrm{p}} + D_\mathrm{p}\Pi + M_\mathrm{p}\dot{\Pi} - K_2(\dot{\mu}_\mathrm{p} - \Pi) - (\mu_\mathrm{p} - \Gamma), \\ \bar{\dot{\mu}}_\mathrm{p} &= \Pi(q, \dot{q}, \mu_\mathrm{p}) = \dot{\Gamma} - K_1(\mu_\mathrm{p} - \Gamma) + S^\top(q, \mu_\mathrm{p}, \Gamma) B^{-1}(q) \partial_{\dot{q}} H^\top, \end{aligned} \tag{8.12}$$

with $K_1, K_2 \succ 0$, is such that $q \to \bar{q}$ and $\mu_\mathrm{p} \to \Gamma(\bar{q}, \dot{\bar{q}})$, $\forall (\mu_\mathrm{p}(0), \dot{\mu}_\mathrm{p}(0)) \in \mathbb{R}^{2n_{\mu_\mathrm{p}}}$, and $\forall (q(0), \dot{q}(0)) \in \mathbb{R}^{2n_\mathrm{q}}$.

Proof. We first consider the problem of deriving a controller under the assumption that the velocity of the piston v_p is set by a controller. This serves as a first step toward the general solution of the problem. System (8.8) is thus reduced into

$$\begin{aligned} M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + K(q) + D(q, \dot{q}) + G_\mathrm{P}^\mathrm{q}(q, \mu_\mathrm{p}) &= 0, \\ \dot{\mu}_\mathrm{p} &= v_\mathrm{p}. \end{aligned} \tag{8.13}$$

We introduce the following control Lyapunov candidate

$$W(q, \dot{q}, \mu_\mathrm{p}) = H(q, \dot{q}) + \frac{1}{2}(\mu_\mathrm{p} - \Gamma)^\top (\mu_\mathrm{p} - \Gamma)\,, \tag{8.14}$$

which can thus be differentiated obtaining

$$\begin{aligned} \dot{W}(q, \dot{q}, \mu_\mathrm{p}) &= \partial_q W \dot{q} + \partial_{\dot{q}} W \ddot{q} + \partial_{\mu_\mathrm{p}} W \dot{\mu}_\mathrm{p}, \\ &= \dot{H} + (\mu_\mathrm{p} - \Gamma)^\top (v_\mathrm{p} - \dot{\Gamma}), \\ &= \partial_q H \dot{q} + \partial_{\dot{q}} H (f(q, \dot{q}) + g(q, \mu_\mathrm{p})) + (\mu_\mathrm{p} - \Gamma)^\top (v_\mathrm{p} - \dot{\Gamma}), \\ &= \partial_q H \dot{q} + \partial_{\dot{q}} H (f(q, \dot{q}) + g(q, \Gamma(q, \dot{q}))) \\ &\quad + \partial_{\dot{q}} H (g(q, \mu_\mathrm{p}) - g(q, \Gamma(q, \dot{q}))) + (\mu_\mathrm{p} - \Gamma)^\top (v_\mathrm{p} - \dot{\Gamma})\,, \end{aligned} \tag{8.15}$$

where we first used the chain rule on $\dot{H}$ and then we added and subtracted $\partial_{\dot{q}} H g(q, \Gamma(q, \dot{q}))$. We now propose the controller $v_\mathrm{p} = \Pi(q, \dot{q}, \mu_\mathrm{p})$ for stabilizing this system, with

$$\Pi(q, \dot{q}, \mu_\mathrm{p}) = \dot{\Gamma} - K_1(\mu_\mathrm{p} - \Gamma) + S^\top(q, \mu_\mathrm{p}, \Gamma) B^{-\top}(q) \partial_{\dot{q}} H^\top.$$

The derivative of the Lyapunov candidate for the closed-loop system is thus

$$\begin{aligned}\dot{W}(q,\dot{q},\mu_{\mathrm{p}}) = {} & \partial_q H\dot{q} + \partial_{\dot{q}} H(f(q,\dot{q}) + g(q,\Gamma(q,\dot{q}))) \\ & + \partial_{\dot{q}} H(g(q,\mu_{\mathrm{p}}) - g(q,\Gamma(q,\dot{q}))) - (\mu_{\mathrm{p}} - \Gamma)^{\top} K_1(\mu_{\mathrm{p}} - \Gamma) \\ & + \partial_{\dot{q}} H B^{-1}(q) S(q,\mu_{\mathrm{p}},\Gamma)(\mu_{\mathrm{p}} - \Gamma)\,,\end{aligned}$$

where we exploited that all terms are scalar to extract the transpose of the last one. This equation can be simplified by invoking Lemma 1 into

$$\dot{W}(q,\dot{q},\mu_{\mathrm{p}}) = \underbrace{\partial_q H\dot{q} + \partial_{\dot{q}} H(f(q,\dot{q}) + g(q,\Gamma(q,\dot{q})))}_{\dot{H}} - (\mu_{\mathrm{p}} - \Gamma)^{\top} K_1(\mu_{\mathrm{p}} - \Gamma). \tag{8.16}$$

Consider now that H is a Lyapunov function for (8.11) under the control action Γ. This assures that

$$0 > \dot{H} = \partial_q H\dot{q} + \partial_{\dot{q}} H(f(q,\dot{q}) + g(q,\Gamma(q,\dot{q}))). \tag{8.17}$$

Note that we are considering here the case of strict sign definiteness of $\dot{H}$. However, the same results can be achieved in the case of semi-definiteness. We can now conclude that $\dot{W} < 0$, thus proving that the controller Π stabilizes (8.13). This concludes the first step of the proof.

We now reiterate this sequence of operations to generalize the controller Π to work on the actual system (8.8). The complete Lyapunov candidate that we propose is

$$Q = W + \frac{1}{2}(\dot{\mu}_{\mathrm{p}} - \Pi)^{\top} M_{\mathrm{p}}(\dot{\mu}_{\mathrm{p}} - \Pi), \tag{8.18}$$

with time derivative

$$\begin{aligned}\dot{Q} &= \dot{W} + (\dot{\mu}_{\mathrm{p}} - \Pi)^{\top}(\Psi - G_{\mathrm{P}}^{\mu_{\mathrm{p}}} - M_{\mathrm{p}}\dot{\Pi}) \\ &= \partial_q W\dot{q} + \partial_{\dot{q}} W\ddot{q} + \partial_{\mu_{\mathrm{p}}} W\dot{\mu}_p + (\dot{\mu}_{\mathrm{p}} - \Pi)^{\top}(\Psi - G_{\mathrm{P}}^{\mu_{\mathrm{p}}} - D_{\mathrm{p}}\dot{\mu}_{\mathrm{p}} - M_{\mathrm{p}}\dot{\Pi}).\end{aligned}$$

We, therefore, propose the controller

$$\Psi(q,\dot{q},\mu_{\mathrm{p}},\dot{\mu}_{\mathrm{p}}) = G_{\mathrm{P}}^{\mu_{\mathrm{p}}} + D_{\mathrm{p}}\Pi + M_{\mathrm{p}}\dot{\Pi} - K_2(\dot{\mu}_{\mathrm{p}} - \Pi) - \partial_{\mu_{\mathrm{p}}} W^{\top}, \tag{8.19}$$

which generates the following closed-loop Lyapunov candidate

$$\dot{Q} = \underbrace{\partial_q W\dot{q} + \partial_{\dot{q}} W\ddot{q} + \partial_{\mu_{\mathrm{p}}} W\Pi}_{\dot{W}} - (\dot{\mu}_{\mathrm{p}} - \Pi)^{\top}(K_2 + D_{\mathrm{p}})(\dot{\mu}_{\mathrm{p}} - \Pi) < 0, \tag{8.20}$$

where we exploit that W is a Lyapunov function for the previous system when $\dot{\mu}_{\mathrm{p}} \equiv \Pi$. This assures the asymptotic stability of the closed loop system when (8.19) is used. The Theorem follows considering that $\partial_{\mu_{\mathrm{p}}} W^{\top} = (\mu_{\mathrm{p}} - \Gamma)$.

□

8.4 Shape Regulation under PCC Approximation

This section provides an example of the application of the proposed model augmentation and model-based control strategy for the setpoint regulation of a pneumatically actuated planar soft robot, modeled through PCC approximation with acting gravity forces.

8.4.1 Model

Background: PCC Dynamic Model

We consider a planar soft robotic arm consisting of three segments, analogous to (Della Santina et al., 2020c), modeled using the PCC (Jones and Walker, 2006) assumption, but the formulation can be easily extended to the 3D case while neglecting the torsional deformations. Alternatively, a strain-based parameterization could be employed (Boyer et al., 2020). We assume a weight distribution of $m_i = \int_0^{l_i} \rho_i(s')ds'$ along the center line of the segment i. Gravity is acting along the vector $g \in \mathbb{R}^2$. We consider the following EOM with diagonal matrices K and D:

$$\begin{aligned} M(q)\ddot{q} + C(q,\dot{q})\,\dot{q} + G(q) + K\,q + D\,\dot{q} + G_{\mathrm{P}}^{\mathrm{q}}(q,\mu_{\mathrm{p}}) = 0, \\ M_{\mathrm{p}}\,\ddot{\mu}_{\mathrm{p}} + D_{\mathrm{p}}\,\dot{\mu}_{\mathrm{p}} + G_{\mathrm{P}}^{\mu_{\mathrm{p}}}(q,\mu_{\mathrm{p}}) = f_{\mathrm{p}}. \end{aligned} \tag{8.21}$$

Model for Fluid Volume in Chamber

A model of the fluid volume in the chambers as a function of the configuration of the segment is required to evaluate the conservative forces by the fluid as specified in (8.6) and (8.7). In this section, we derive a simple analytical model based on CC kinematics. It is assumed that the volume of the chamber is only dependent on the curvature of the segment, as we model the segment length $l_{0,i}$ to stay constant and the chambers to be inextensible in the radial direction of the curvature. We visualize the model and its parameters in Figure 8.3. Thus, the volume of chamber j part of segment i is defined as:

$$V_{\mathrm{C},j}(q_i) = \int_{d_{\mathrm{C,a},j}}^{d_{\mathrm{C,b},j}} b_{\mathrm{C}}\, l_i(d'_{\mathrm{C}})\,\mathrm{d}d'_{\mathrm{C}}, \tag{8.22}$$

where b_{C} describes the constant planar thickness of the chamber and $l_i(d'_{\mathrm{C}})$ the length of segment i at offset d'_{C} from the center-line. The function $l_i(d'_{\mathrm{C}})$ is derived from the properties of CC of the segment

$$l_i(d'_{\mathrm{C}}) = l_{0,i} - q_i\, d'_{\mathrm{C}}. \tag{8.23}$$

The integration inherits an opposite sign for the change of volume with q_i for the left and right chamber, respectively for an inner and outer chamber wall radius of $0 < d_{\mathrm{C,a}} < d_{\mathrm{C,b}} < d_i$ and a continuum segment of radius d_i:

$$V_{\mathrm{C},j}(q_i) = b_{\mathrm{C}} \left(l_{0,i}(d_{\mathrm{C,b}} - d_{\mathrm{C,a}}) \mp \frac{q_i}{2}(d_{\mathrm{C,b}}^2 - d_{\mathrm{C,a}}^2) \right). \tag{8.24}$$

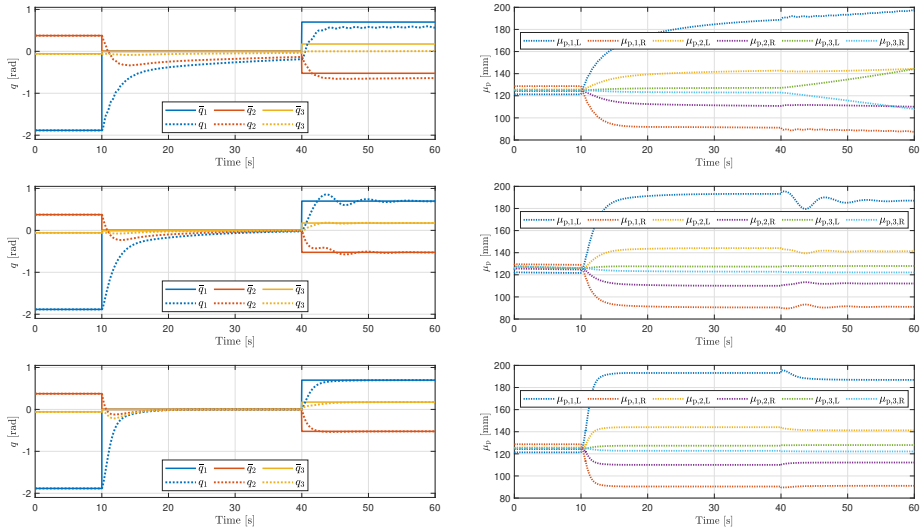
The partial derivative $\partial_q V_{\mathrm{C}}$ is determined as

$$\partial_{q_i} V_{\mathrm{C},j} = \mp 0.5\, b_{\mathrm{C}}\, (d_{\mathrm{C,b}}^2 - d_{\mathrm{C,a}}^2). \tag{8.25}$$

8.4.2 Setpoint Control

Configuration-space control: Consider the following regulator of desired configuration $\bar{q} \in \mathbb{R}^{n_{\mathrm{q}}}$,

$$\bar{\tau}(q,\bar{q}) = K\,\bar{q} + G(q), \tag{8.26}$$



(a) Configuration q (b) Piston position μ_{p}

Figure 8.4: Simulation of posture regulation under PCC approximation comparing the performance of an end-to-end PID baseline controller (1st row), with a coupling-aware PID controller (2nd row) and the nonlinear backstepping controller (3rd row). The set-point reference configuration is shown with solid lines.

where $\bar{\tau} \in \mathbb{R}^{n_{\mathrm{q}}}$ is the torque in configuration space. Asymptotic stability of the equilibrium $\bar{q}$ is proven through the Lyapunov function $H(q, \dot{q}) = \dot{q}^\top M(q)\dot{q}/2 + q^\top K q/2$, which yields the time derivative $\dot{H} = -\dot{q}^\top D\dot{q}/2 \leq 0$.

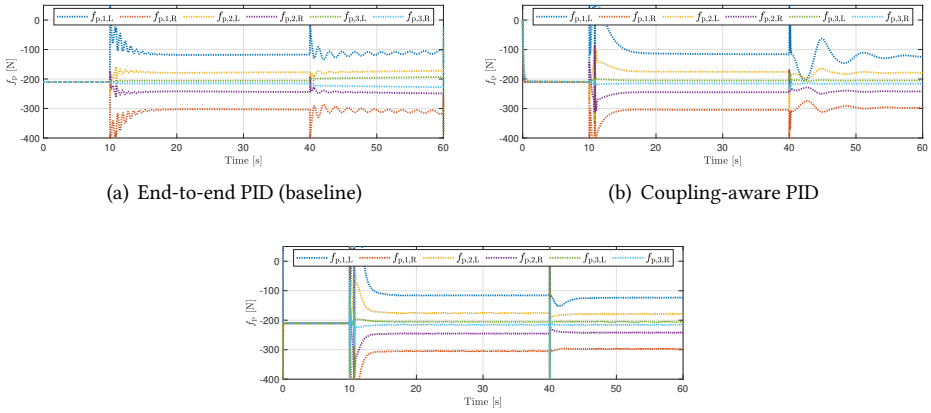
Mapping from configuration to actuation space with force balance: Our backstepping controller requires access to $\Gamma(q, \dot{q})$ which returns a desired piston $\bar{\mu}_{\mathrm{p}}$ given the desired torque $\bar{\tau}$ and the current state of the soft system $(q, \dot{q})$. In the planar case with inextensible segments, there exists a redundancy in actuating the pistons controlling the pressure in the left and right chambers of a segment to trigger $\bar{\tau}$ on the segment. Thus, we decide to solve this redundancy by equally attributing the desired torque to both pistons.

We assume that the system is calibrated at a straight configuration $q_{\mathrm{t0}} = 0$ with pistons preloaded at position $\mu_{p,t0} \in \mathbb{R}^{n_{\mu\mathrm{p}}}$ leading to a fluidic volume of

$$V_{\mathrm{t0},j} = V_{\mathrm{C},j}(q_{\mathrm{t0},i}) + \mu_{\mathrm{p,t0},j}\, A_{\mathrm{p},j} \tag{8.27}$$

in the system. After preloading, the fluids in the left and right chambers each apply a preloaded torque of magnitude $G^q_{\mathrm{P,t0}} \in \mathbb{R}^{n_{\mathrm{q}}}$ on the soft system. It is implicitly assumed that the piston length l_{p}, piston area A_{p}, the preloaded piston position $\mu_{\mathrm{p,t0}}$ and the preloaded volume V_{t0} are equal for the left and right chambers of segment i. We can write the conservative forces acting on the left chamber $G^q_{\mathrm{P,L}}(q, \mu_{\mathrm{p}}) \in \mathbb{R}^{n_{\mathrm{q}}}$ and right chamber $G^q_{\mathrm{P,R}}(q, \mu_{\mathrm{p}}) \in \mathbb{R}^{n_{\mathrm{q}}}$ as differences from the neutral conservative force $G^q_{\mathrm{P,t0}}$

$$G^q_{\mathrm{P,L}} = G^q_{\mathrm{P,t0}} + \Delta G^q_{\mathrm{P,L}}, \quad G^q_{\mathrm{P,R}} = -G^q_{\mathrm{P,t0}} + \Delta G^q_{\mathrm{P,R}}. \tag{8.28}$$



(a) End-to-end PID (baseline)

(b) Coupling-aware PID

(c) Backstepping controller

Figure 8.5: Simulation of posture regulation under PCC approximation comparing the control input (i.e., the piston actuation force f_p) of an end-to-end PID baseline controller, a coupling-aware PID controller, and the nonlinear backstepping controller.

The force applied by the fluid in the left and right chambers on the system is

$$G^q_{\mathrm{P}}(q,\mu_p) = G^q_{\mathrm{P,L}} + G^q_{\mathrm{P,R}} = \Delta G^q_{\mathrm{P,L}} + \Delta G^q_{\mathrm{P,R}}. \tag{8.29}$$

We re-arrange to find an expression for the desired conservative force offsets, which equally distribute a commanded torque $\bar{\tau}$ to the fluid in both chambers. Setting $\Delta\bar{G}^{\bar{q}}_{\mathrm{P,L}} = \Delta\bar{G}^{\bar{q}}_{\mathrm{P,R}} = 0.5\,\bar{\tau}(q,\bar{q})$ results for the chosen setpoint controller $\bar{\tau}$ and a diagonal elastic matrix K with elements k_i in:

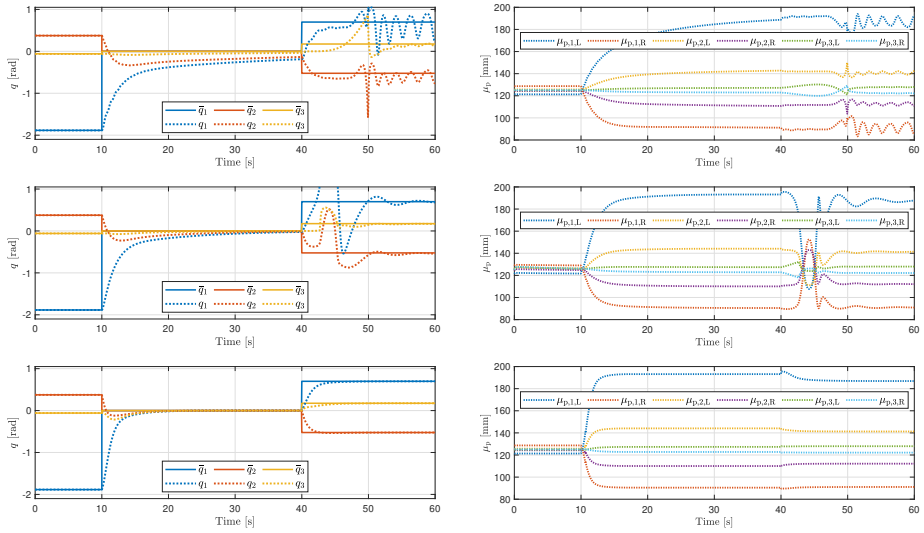
$$\bar{G}^{\bar{q}}_{\mathrm{P},j}(q,\bar{q}_i) = \pm G^q_{\mathrm{P,t0},i} - 0.5\,(k_i\bar{q}_i + G_i(q)). \tag{8.30}$$

Eq. (8.7) is inversed to compute the desired piston position $\bar{\mu}_{\mathrm{p}} = \Gamma(q,\bar{q})$:

$$\Gamma_j(q,\bar{q}) = \frac{1}{A_{\mathrm{p},j}}\left(\frac{\alpha_{\mathrm{air},j}\,\partial_{q_i} V_{\mathrm{C},j}}{p_{\mathrm{atm}}\,\partial_{q_i} V_{\mathrm{C},j} - \bar{G}^{\bar{q}}_{\mathrm{P},j}(q,\bar{q})} - V_{\mathrm{C},j}(q)\right). \tag{8.31}$$

Backstepping: The system is now in the form of Eq. (8.11), so Theorem 1 can be invoked and a specialized version of Eq. (8.12) can be derived for the PCC-case and our chosen set-point controller of Eq. (8.26). The partial derivative of the Lyapunov function of the soft system controller evaluates to $\partial_{\dot{q}} H(q,\dot{q}) = \dot{q}^\top M(q)$, which allows to re-formulate (8.12) into:

$$\begin{aligned}
\bar{\dot{\mu}}_{\mathrm{p}} &= \Pi(q,\dot{q},\mu_{\mathrm{p}}) = \dot{\Gamma}(q,\dot{q}) - K_1(\mu_p - \Gamma(q)) + S^\top(q,\mu_{\mathrm{p}},\Gamma(q))\,\dot{q},\\
\bar{f}_{\mathrm{p}} &= \Psi(q,\dot{q},\mu_{\mathrm{p}},\dot{\mu}_{\mathrm{p}}) = G_{\mathrm{P}}^{\mu_{\mathrm{p}}} + D_{\mathrm{p}}\,\Pi(q,\dot{q},\mu_{\mathrm{p}}) + M_{\mathrm{p}}\,\dot{\Pi} - K_2\,(\dot{\mu}_{\mathrm{p}} - \Pi(q,\dot{q},\mu_{\mathrm{p}})) - (\mu_{\mathrm{p}} - \Gamma(q)).
\end{aligned} \tag{8.32}$$



(a) Configuration q (b) Piston position μ_p

Figure 8.6: Simulation of posture regulation under PCC approximation for an actuation system with increased inertia ($m_p = 0.5\,\text{kg}$) comparing the performance of an end-to-end PID baseline controller (1st row), with a coupling-aware PID controller (2nd row) and the nonlinear backstepping controller (3rd row). The set-point reference configuration is shown with solid lines.

8.5 Simulations

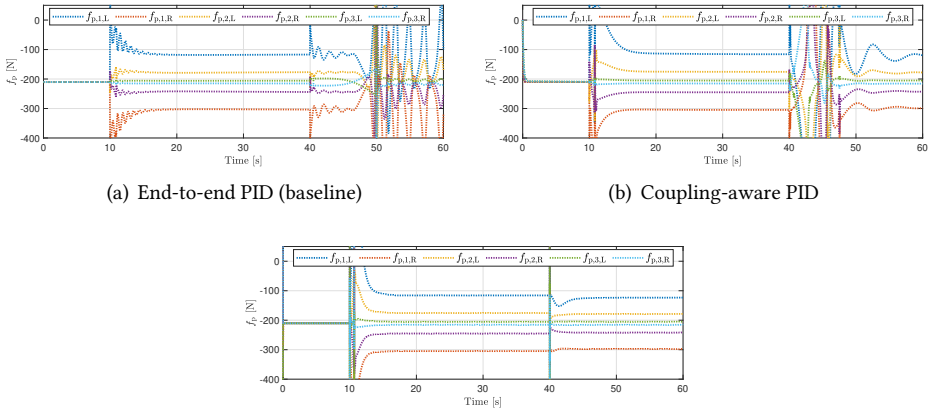
8.5.1 System

We consider a planar soft robot arm consisting of three independently actuated CC segments, modeled upon the second half of the robot in (Della Santina et al., 2020c). Segments have equal length $l_0 = 11\,\text{cm}$, uniform mass density $\rho = 0.99\,\text{kg/m}$ concentrated on the central axis. The stiffness K and damping D matrices are diagonal with constants $0.01\,\text{N/rad}$ and $0.01\,\text{Ns/rad}$. The segment has a diameter of $44.5\,\text{mm}$. Based on CAD analyses of a real system, we take $d_{C,a} = 7.14\,\text{mm}$, $d_{C,b} = 20.19\,\text{mm}$, and $b_C = 8.07\,\text{mm}$. A positive curvature and positive configuration q_i correspond to bending counter-clockwise. The straight configuration of the robot along the x-axis is perpendicular to gravity acting in the negative y-direction, as shown in Figure 8.3, so that gravity tends to induce clockwise bending. Moving to the pistons, $A_p = 7.9\,\text{cm}^2$, $m_p = 0.19\,\text{kg}$, $l_p = 0.5\,\text{m}$ are chosen. We consider a damping matrix D_p with damping constants $d_p = 10\,\text{kN s m}^{-1}$ along the diagonal. The pistons are filled with air at $\mu_{p,0} = l_p$ and $p_{atm} = 1\,\text{bar}$ and subsequently pre-loaded to $\mu_{p,t0} = 0.25\,l_p$. We set the backstepping gains to $K_1 = 6000\,\text{s}^{-1}$ and $K_2 = 4.5\,\text{kN m}^{-1}$.

8.5.2 End-to-End PID

We first introduce an end-to-end PID controller, which will serve as a baseline

$$\Delta f_p = K_p(\bar{q} - q) + K_i \int_0^t (\bar{q} - q)\,\mathrm{d}t' - K_d\,\dot{q} \tag{8.33}$$



(a) End-to-end PID (baseline)

(b) Coupling-aware PID

(c) Backstepping controller

Figure 8.7: Simulation of posture regulation under PCC approximation for an actuation system with increased inertia ($m_p = 0.5\,\mathrm{kg}$) comparing the control input (i.e., the piston actuation force f_p]) of an end-to-end PID baseline controller, of a coupling-aware PID controller and the nonlinear backstepping controller.

where $K_p, K_i, K_d \geq 0$ are scalar gains. $\Delta f_p \in \mathbb{R}^{n_q}$ is the scalar offset from the actuation force $f_{p,t0}$ corresponding to the pre-loaded pressure p_{t0}. Analog to (8.30), Δf_p can be equally distributed on both chambers within a segment. The PID gains have been selected so to achieve a similar transient behavior as for the backstepping controller and are equal to $K_p = 200\,\mathrm{N\,rad^{-1}}$, $K_i = 7\,\mathrm{N\,rad^{-1}\,s^{-1}}$, and $K_d = 200\,\mathrm{N\,s\,rad^{-1}}$.

8.5.3 Coupling-Aware PID

Next, we implement a control strategy that takes advantage of the understanding of the potential coupling and uses a PID for low-level control of the pistons

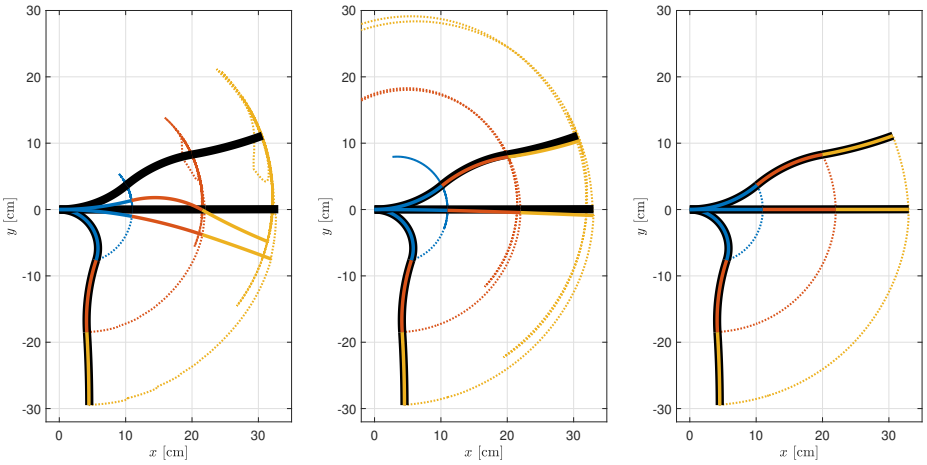
$$f_p = K_p\left(\Gamma(q,\bar{q}) - \mu_p\right) + K_i \int_0^t \left(\Gamma(q,\bar{q}) - \mu_p\right) \mathrm{d}t' - K_d\,\dot{\mu}_p. \tag{8.34}$$

Here, $K_p, K_i, K_d \geq 0$ are scalar gains, and $\Gamma(q,\bar{q})$ is the correction on (8.26) which takes the coupling defined in (8.31) in account. The PID gains are tuned similarly to the coupling-aware PID and are equal to $K_p = 150\,000\,\mathrm{N\,m^{-1}}$, $K_i = 15\,000\,\mathrm{N\,m^{-1}\,s^{-1}}$, and $K_d = 100\,\mathrm{N\,s\,m^{-1}}$.

8.5.4 Results

We simulate the response of the closed-loop generated by all three controllers to a sequence of step references. The segments are initialized at the equilibrium configuration. At 10 s, the reference is moved to the straight configuration $\bar{q} = 0$. After another 30 s, we change it again to $\bar{q} = \begin{bmatrix} 0.6981\,\mathrm{rad} & -0.5236\,\mathrm{rad} & 0.1745\,\mathrm{rad} \end{bmatrix}^\top$.

Figures 8.4-8.5 show that the backstepping controller is approaching the set-point reference with no oscillations or overshooting. These are instead visible for coupling-are PID controller after the second change in reference configuration. The end-to-end PID



(a) End-to-end PID (b) Coupling-aware PID (c) Backstepping

Figure 8.8: Cartesian evolution of the soft robot for an actuation system with increased inertia ($m_\mathrm{p} = 0.5\,\mathrm{kg}$). All gains remain unchanged and are tuned for the original system with $m_\mathrm{p} = 0.19\,\mathrm{kg}$. The dotted lines mark the evolution of the tip of the segments. The soft robot consists of three segments (blue, orange, and yellow). The reference configuration at the three set points is marked with a thick black line.

controller does not converge to the desired configuration within 60 s as it does not take into account gravity.

Next, we increase the inertia of the actuation system by setting the piston mass m_p to 0.5 kg. We leave both the backstepping and the PID gains unchanged. Figures 8.6-8.8 demonstrate that the backstepping-based approach is able to adapt to the new system, while the end-to-end PID shows large oscillations at 50 s and the coupling-aware PID displays significant overshoot in curvatures and piston positions. Note that the latter are especially dangerous in real experiments since they may signify that pistons reach their limits.

8.6 Conclusion

This chapter proposed a model for soft robots actuated using pneumatic fluidic drive cylinders and introduced a model-based controller to take the actuators' dynamics into account. The stability of this backstepping-based control strategy was proven using a Lyapunov argument. As an example of application, model and control strategy have been specialized for the planar PCC-case. We also proposed a coupling-aware extension of the standard hierarchical PID strategy as a middle-ground solution.

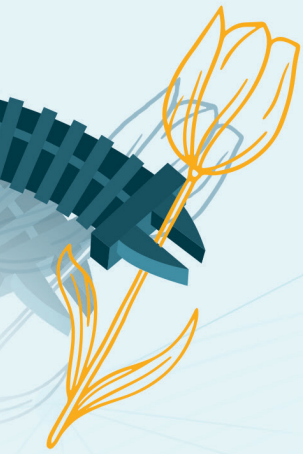
Future work will focus on extending this strategy to more sophisticated models and controllers while validating it experimentally in a lab environment. In particular, the model—especially the fluidic volume model—and the controller should be enhanced and augmented to enable 3D control rather than being limited to planar settings. Additionally,

it would be interesting to explore whether a similar backstepping-based control approach could be applied to alternative pneumatic actuation methods (Franco et al., 2024), such as pressure regulation via valve systems.

Afterword

In this chapter, we have presented a dynamical model that, in addition to the standard soft robot dynamics, also considers the dynamics of a piston-driven pneumatic actuation system. Subsequently, we exploited the actuation dynamics within a backstepping controller. We believe that similar approaches could also be applied to other soft robotic actuation methods, such as tendons pulled by electrical motors or Shape-Memory Alloy (SMA) actuators (Zaidi et al., 2021). For example, recently, a similar backstepping-based control approach has been successfully applied to soft robots actuated by valve-driven pressure regulators (Franco et al., 2024), and a singular-perturbation approach for taking into account the actuation dynamics of tendon-driven soft robots has been proposed by Ribeiro et al. (2025).

Sometimes, modeling the dynamics of soft robots from first principles can be challenging, making it difficult to control their behavior effectively. In other cases, such as pneumatic systems, hysteresis effects are prevalent (Vo-Minh et al., 2010), and direct measurement of the hysteretic displacement is not possible using available sensor data. In such situations, employing learning approaches (e.g., machine learning with neural networks) can be beneficial. Still, we would like to preserve the interpretability, structure, and guarantees provided by fully physics-based models. For that reason, we explore integrating physical structures and stability guarantees into learning-based models and controllers in Part II of this thesis.



II

Incorporating Physical Structure and Stability Guarantees Into Learned Models and Controllers

8

9

Exploiting Learned Magnetic Fields and Kinematic Models for Shape Sensing

Foreword. Proprioception is essential for (soft) robots, enabling applications like control and motion planning. In Chapter 4, we introduced a kinematics-aware SLAM approach based on images from a monocular camera attached to the robot. However, this method has drawbacks: i) Cameras, despite their miniaturization, require a clear, unobstructed view, necessitating rigid components on the robot's surface, which potentially compromises safety. ii) SLAM algorithms perform poorly in environments with few visually distinguishable features. iii) Processing high-dimensional image data is computationally intensive, leading to higher processing demands and/or lower shape-sensing sampling rates. To address these issues, this chapter proposes an alternative proprioception method using magnetic sensors. These sensors, along with embedded magnets, can be fully integrated into the robot's body, maintaining a soft, compliant surface. Additionally, the sensory data is much lower-dimensional, making it significantly less computationally demanding to process. However, the complex nature of magnetic fields makes the mapping from magnetic sensor readings to shape information very challenging, and it would be very hard to develop a fully physics-based model to help us interpret the sensor readings. Instead, we leverage modern ML methods for this task. Akin to projecting the SLAM pose measurements onto the kinematic model (Chapter 4), we again leverage here kinematic knowledge to simplify the problem setting. Specifically, instead of learning proprioception end-to-end from magnetic sensor measurements to configuration values, we simplify the learning problem by asking a neural network to predict the measurements of a sensor given a parametrization of the spatial relation between the magnetic sensor and all magnets embedded in the robot. We present the proposed methodology in detail in Sec. 9.2. We verify the proposed approach extensively in simulations based on PCC- & PAC-parametrized soft robots (Section 9.3 & 9.4), and experimentally in Section 9.5 on a one-segment soft robot moving in 3D space.

Abstract. *Sensing the shape of continuum soft robots without obstructing their movements and modifying their natural softness requires innovative solutions. This chapter proposes to use magnetic sensors that are fully integrated into the robot to achieve proprioception. Magnetic sensors are compact, sensitive, and easy to integrate into a soft robot. We also propose a neural architecture to make sense of the highly nonlinear relationship between the perceived intensity of the magnetic field and the shape of the robot. By injecting a priori knowledge from the kinematic model, we obtain an effective yet data-efficient learning strategy. We first demonstrate in simulation the value of this kinematic prior by investigating the proprioception behavior when varying the sensor configuration, which does not require us to re-train the neural network. We validate our approach in experiments involving one soft segment containing a cylindrical magnet and three magnetoresistive sensors. During the experiments, we achieve mean relative errors of 4.5%.*

This chapter is partly based on *T. Baaij*, M. K. Holkenborg*, **M. Stölzle***, D. van der Tuin*, J. Naaktgeboren, R. Babuška, and C. Della Santina (2023). Learning 3D shape proprioception for continuum soft robots with multiple magnetic sensors. Soft Matter, 19(1), 44-56* (Baaij et al., 2023).

T.B., M.K.H., M.S., and D.v.d.T contributed equally to this chapter. C.D.S. and R.B. conceived and led the project. M.S. invented the proprioception algorithm and implemented all simulations. T.B., M.K.H., J.N., and D.v.d.T designed the strategy to integrate sensors and magnets into the robot. T.B., M.K.H., J.N., and D.v.d.T developed the PCB and sensing electronics. T.B., M.K.H., and D.v.d.T fabricated the soft robot. T.B., M.K.H., J.N., and D.v.d.T implemented the experimental data processing pipeline. T.B., M.K.H., M.S., and D.v.d.T conducted experiments. T.B., M.K.H., M.S., and D.v.d.T tuned the learning strategy and developed suitable trajectories. T.B., M.K.H., M.S., and D.v.d.T wrote the manuscript. C.D.S. and R.B. revised the manuscript.

9.1 Introduction

Shape sensing for continuum soft robots is especially complex because it is both a technological and algorithmic challenge. Rigid sensors must not obstruct the natural behavior or reduce the compliance of soft robots. At the same time, non-collocated and nonlinear sensors require algorithms for the measurements to be interpreted and connected to a description of the robot's shape.

Several sensing modalities have been considered to implement shape sensing, such as resistive (Kramer et al., 2011; Shih et al., 2019), capacitive (Scimeca et al., 2019; Shintake et al., 2018), optical (Li et al., 2021), and visual (Rosi et al., 2022). Magnetic sensors (Guo et al., 2019; Luo et al., 2017; Mitchell et al., 2021; Ozel et al., 2015; Song et al., 2015) are a promising solution. Magnetic sensors are compact, highly sensitive, and can be easily integrated into existing soft robot designs. Thus, they can provide reliable and fully proprioceptive measurements at the cost of a minimal decrease in the robot's softness. Among the above-cited papers, the only work leveraging external magnetic fields is by Song et al. (2015), where coils placed at a distance generate the magnetic field. Along these lines, an obvious choice would be to measure the Earth's magnetic field for proprioception purposes. However, it would only allow for the estimation of two rotational components, and any translational effects, such as an elongation of the continuum robot, could not be captured. Alternatively, some papers (Guo et al., 2019; Luo et al., 2017; Ozel et al., 2015) use coaxial pairs of magnets and sensors embedded in the robot to estimate planar deformations. Recently, Mitchell et al. (2021) has shown that a similar strategy can sense 3D deformations. Such simple arrangements greatly simplify the analysis, allowing for connecting readings to shape through direct interpolation. Nevertheless, relying on isolated pairs of sensors and magnets also strongly limits the density and the amount of information gathered through this method.

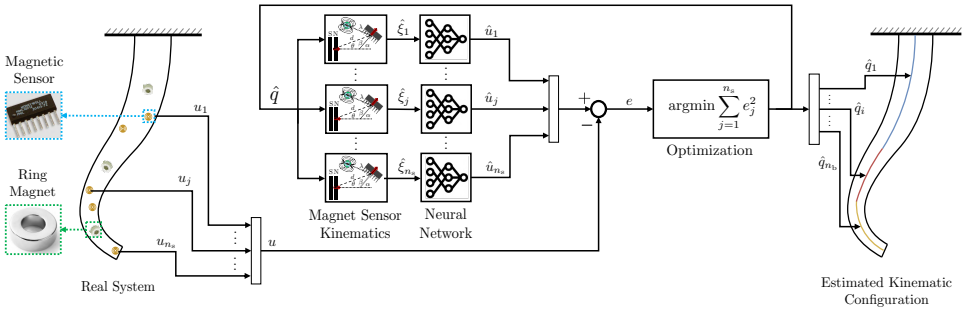


Figure 9.1: Proprioception for continuum soft robots with magnetic sensors: an initial configuration estimate $\hat{q}$ is employed to derive the kinematic relationship $\hat{\xi}$ between a sensor and each magnet. Subsequently, this kinematic description is used to predict the sensor measurements $\hat{u} \in \mathbb{R}^{n_s}$ with a neural network trained in advance. A Mean Squared Error (MSE) metric evaluates the accuracy of the predictions $\hat{u}$ compared to the actual measurements u. Finally, we optimize the configuration estimate $\hat{q}$ to achieve proprioception by minimizing the sensor measurement prediction loss.

This chapter proposes to use permanent ring magnets and multiple magnetic sensors for shape sensing of continuum soft robots. Importantly, we remove the requirement of magnets and sensors being placed in coaxial pairs. We have been inspired by recent

work leveraging deep learning to interpret various types of non-magnetic sensor data for proprioception purposes (Ding et al., 2021; Soter et al., 2018; Thuruthel et al., 2019; Truby et al., 2020). However, learning end-to-end mappings from sensors to configurations has three significant drawbacks: (i) it is data-intensive, (ii) it calls for recurrent architectures to encourage temporal consistency for the robot's configuration estimates, and (iii) it requires re-training when changing the kinematic model of the robot. We propose a neural architecture that circumvents all three issues (see Fig. 9.1). First, we train shallow neural networks to predict the measurements of the magnetic sensors from a parameterization describing their relative pose with respect to the magnets. We then optimize the configuration estimate - and thus the sensor positions - to minimize the error between the predicted and actual sensor measurements. This way, we introduce a priori information on the modes of deformation of a continuum soft robot, effectively removing the kinematics from the black box. The presented strategy enables us to re-arrange and remove redundant sensors during inference without requiring us to re-train the neural network on the adjusted sensor configuration.

To summarize, this chapter contributes to the state of the art in soft robot sensing with:

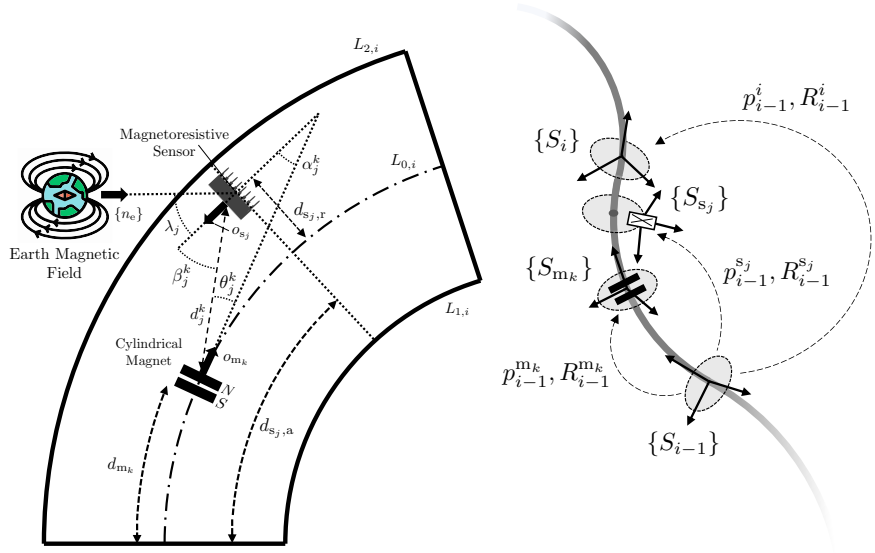
- A proprioceptive sensing modality relying on multiple magnetic sensors in conjunction with a neural network-based architecture that learns to estimate the full 3D shape of the robot from the sensor readings.
- Injection of kinematic priors through a description to spatially relate the poses of a sensor to the magnets. This proposed description serves as input to a neural network that predicts sensor measurements.
- Experimental verification of the approach for a one-segment robot with three integrated Magnetoresistive Sensors (MRSs) and proprioception of 3D curvature.

We open-source a Python/PyTorch implementation of the proposed algorithm and the corresponding datasets on GitHub [1].

9.2 Proposed Methodology: Proprioception with Magnetic Sensors

This section introduces a methodology to achieve proprioception for soft robots with magnetic sensors. We consider a continuum robot with the shape of its backbone described by the configuration variables $q \in \mathbb{R}^{n_q}$. In the commonly used PCC kinematic state parametrization (Webster III and Jones, 2010), the continuum robot is assumed to consist of n_b segments with each segment exhibiting constant curvature and elongation along its length. Therefore, the configuration of the soft robot can be described with $q \in \mathbb{R}^{3n_b}$. Please refer to Appendix 9.2.3 for more details. We indeed use PCC for most of our simulations and experiments. Note, however, that the proposed proprioception algorithm applies to any finite-dimensional kinematic description of a soft robot (Armanini et al., 2023). Indeed, we also specifically consider a robot with AC (Della Santina, 2020; Stella et al., 2023c) with its shape described by the configuration $q \in \mathbb{R}^4$. We document this alternative kinematic

[1] https://www.github.com/tud-phi/promasens



(a) Magnet Sensor Kinematics (b) Coordinate transformations

Figure 9.2: **Panel (a):** Parameters used to describe the kinematics between the j^{th} sensor and the k^{th} magnet. To simplify the illustration, we visualize the unextended planar case with both magnet and sensor part of the i^{th} segment. However, this is not a strict assumption, as magnets and sensors can also be part of different segments. **Panel (b):** Coordinate frames for a soft segment containing the j^{th} magnetoresistive sensor and the k^{th} cylindrical magnet placed along the center line. $\{S_{i-1}\}$ and $\{S_i\}$ describe the frames of the base and tip of the i^{th} segment, respectively.

model in Appendix 9.2.4. The proprioception methodology described in the remaining section is agnostic to the chosen kinematic model.

9.2.1 Proposed Method at a Glance

We integrate n_{m} axially symmetric magnets and n_{s} magnetic sensors into the robot. The magnets must be installed along the center line of the segment, while the sensors can be arbitrarily placed. Fig. 9.1 concisely describes the proposed shape-sensing strategy with a pictorial example of a soft robot described with three constant curvature segments equipped with five magnetic sensors and three cylindrical magnets.

The goal of the algorithmic part (center of the figure) is to regress the robot shape (left) by estimating the configuration $\hat{q}$ of all segments (right), starting from the measurements of the magnetic sensors u (e.g., usually the magnetic flux density). We achieve this by training a sensor measurement predictor and subsequently optimizing the configuration estimate $\hat{q}$ for the prediction $\hat{u}$ to match the actual sensor measurement u. Instead of predicting the sensor measurements end-to-end, we decouple the kinematics by explicitly describing the kinematic relationship $\hat{\xi}_j = f_{\xi,j}(\hat{q})$ between sensor j and each magnet. Then, we train a neural network f_{π_j} to predict the measurement $\hat{u}_j$ based on the input $\hat{\xi}_j$. To achieve proprioception, we jointly optimize $\hat{q}$ for all sensor measurement predictions.

9.2.2 Background: Continuum Soft Robot Kinematics

A kinematic description provides us with the forward kinematic transformation $T^{s}_{i-1}(q_i, s)$ from base frame $\{S_{i-1}\}$ at the proximal end of the i^{th} segment to the local frame $\{S_v\}$ at a coordinate $v \in [0,1]$ for a given configuration q_i. Furthermore, the tip frame of the i^{th} segment located at the coordinate $v = 1$ is denoted as $\{S_i\}$, as shown in Fig. 9.2(b).

9.2.3 Piecewise Constant Curvature Kinematics

Under the PCC hypothesis, the shape of each segment i of length L_i and radius d_i can be fully parameterized through three variables

$$q_i = \begin{bmatrix} \Delta_{x,i} & \Delta_{y,i} & \delta L_i \end{bmatrix}^\top \in \mathbb{R}^3 \tag{9.1}$$

where $\Delta_{x,i}$ and $\Delta_{y,i}$ represent bending into the local x and y directions respectively and δL_i defines the elongation of the segment. The base frame of segment i is referred to as $\{S_{i-1}\}$ as stated in Fig. 9.2(b). Given q_i, a homogeneous transformation $T^{v}_{i-1}(q_i, v)$ to the point frame $\{S_v\}$ is available

$$\begin{aligned} R^{v}_{i-1}(q_i, v) &= \begin{bmatrix} 1 + \frac{\Delta^2_{x,i}}{\Delta^2_i}(\mathrm{C}_v - 1) & \frac{\Delta_{x,i}\Delta_{y,i}}{\Delta^2_i}(\mathrm{C}_v - 1) & \frac{\Delta_{x,i}}{\Delta_i}\mathrm{S}_v \\ \frac{\Delta_{x,i}\Delta_{y,i}}{\Delta^2_i}(\mathrm{C}_v - 1) & 1 + \frac{\Delta^2_{y,i}}{\Delta^2_i}(\mathrm{C}_v - 1) & \frac{\Delta_{y,i}}{\Delta_i}\mathrm{S}_v \\ \frac{-\Delta_{x,i}}{\Delta_i}\mathrm{S}_v & \frac{-\Delta_{y,i}}{\Delta_i}\mathrm{S}_v & \mathrm{C}_v \end{bmatrix}, \\ p^{i}_{i-1}(q_i, v) &= \frac{d_i(L_{0,i} + \delta L_i)}{v\,\Delta^2_i} \begin{bmatrix} \Delta_{x,i}(1 - \mathrm{C}_v) & \Delta_{y,i}(1 - \mathrm{C}_v) & \Delta_i \mathrm{S}_v, \end{bmatrix}^\top \end{aligned} \tag{9.2}$$

where R^{v}_{i-1}, $p^{i}_{i-1}(q_i, v)$ denote the rotation matrix and translation vector respectively. We substituted $\Delta_i = \sqrt{\Delta^2_{x,i} + \Delta^2_{y,i}}$, $\mathrm{S}_v = \sin\left(\frac{v\Delta_i}{d_i}\right)$, and $\mathrm{C}_v = \cos\left(\frac{v\Delta_i}{d_i}\right)$ for conciseness.

9.2.4 Affine Curvature Kinematics

The AC hypothesis (Della Santina, 2020; Stella et al., 2023c) models the bending of the soft segment to be conforming to the affine function $\kappa(t, v) = \kappa_0(t) + \kappa_1(t)v$, where κ describes the local curvature of the backbone at the coordinate $v \in [0,1]$ along the segment and $\kappa_0(t)$, $\kappa_1(t)$ are the zero-order and first-order term of the curvature polynomial respectively (Della Santina and Rus, 2019). Specifically, we implement the recently proposed extension to 3D environments (Stella et al., 2023c), which specifies an azimuth angle of the bending direction $\phi(t)$ and additionally allows for an elongation $\delta L(t)$ of the segment. Accordingly, the configuration of the i^{th} segment is described at any point in time by

$$q_i = \begin{bmatrix} \kappa_{0,i} & \kappa_{1,i} & \phi_i & \delta L_i \end{bmatrix}^\top \in \mathbb{R}^4. \tag{9.3}$$

Now that the configuration space is defined, we aim to find a description of the forward kinematics. Firstly, the bending angle $\theta_i(q, v)$ is found by integrating the curvature

$$\theta(q, v) = \int_{v'=0}^{v} \kappa_i(q, v')\,\mathrm{d}v' = \kappa_{0,i}\, v + \kappa_{1,i}\, \frac{v^2}{2}. \tag{9.4}$$

The rotation to the frame $\{S_v\}$ can then be easily determined with $R^i_{i-1}(q,v) = R_{\phi_i}(q,v)\,R_\theta(q,v)R^\top_{\phi_i}(q,v)$. After substituting $S. = \sin(\cdot)$, $C. = \cos(\cdot)$ for conciseness, we state the homogeneous transformation as

$$
\begin{aligned}
R^v_{i-1}(q,v) &= \begin{bmatrix} S^2_{\phi_i} C_{\theta_v} + C^2_{\phi_i} & -S_{\phi_i} C_{\phi_i} C_{\theta_i} + S_{\phi_i} C_{\phi_i} & S_{\phi_i} S_{\theta_v} \\ -S_{\phi_i} C_{\phi_i} C_{\theta_i} + S_{\phi_i} C_{\phi_i} & S^2_{\phi_i} + C^2_{\phi_i} C_{\theta_v} & -S_{\theta_v} C_{\phi_i} \\ -S_{\phi_i} S_{\theta_v} & S_{\theta_v} C_{\phi_i} & C_{\theta_i} \end{bmatrix}, \\
p^v_{i-1}(q,v) &= \left(L_{0,i} + \delta L_i\right) \begin{pmatrix} \int_{v'=0}^{v} \sin(\theta(q,v'))\,\mathrm{d}v' \sin(\phi_i) \\ -\int_{v'=0}^{v} \sin(\theta(q,v'))\,\mathrm{d}v' \cos(\phi_i) \\ \int_{v'=0}^{v} \cos(\theta(q,v'))\,\mathrm{d}v' \end{pmatrix}.
\end{aligned}
\tag{9.5}
$$

We choose to integrate the translational terms in (9.5) numerically with 101 sample points using the Torchquad (Gómez et al., 2021) implementation of the Simpson's rule, which makes the forward kinematics fully and automatically differentiable.

9.2.5 Magnet Sensor Kinematics

This subsection derives the kinematic relationship $\xi_j = f_{\xi_j}(q) \in \mathbb{R}^{1+4n_\mathrm{m}}$ between the j^th sensor and all n_m magnets as we hypothesize that we can estimate the sensor measurement u_j solely based on a) the angle $\lambda_j \in \mathbb{R}^1$ to the earth's magnetic field, b) the distance d^k_j between the j^th sensor and k^th magnet, c) the angle α^k_j between the sensor measurement direction and the cylindrical axis of the magnet, d) the angle β^k_j between the sensor measurement direction and the vector from the magnet to the sensor, and e) the angle θ^k_j between the cylindrical axis of the magnet and the vector from the magnet to the sensor. Accordingly, ξ_j is defined as

$$
\xi_j = f_{\xi,j}(q) = \begin{pmatrix} \lambda_j & \xi^{1\top}_j & \cdots & \xi^{k\top}_j & \cdots & \xi^{n_\mathrm{m}\top}_j \end{pmatrix}^\top \in \mathbb{R}^{1+4n_\mathrm{m}},
\tag{9.6}
$$

with $\xi^k_j \in \mathbb{R}^4$ the kinematic relationship between the j^th sensor and the k^th magnet: $\xi^k_j = \begin{pmatrix} d^k_j & \alpha^k_j & \beta^k_j & \theta^k_j \end{pmatrix}^\top \in \mathbb{R}^4$. We visualize the parameters incorporated in ξ^k_j in Fig. 9.2(a).

In the following, we present the derivation of all components of ξ^k_j. Please note that all kinematic frames used in the following paragraphs are visualized in Fig. 9.2(b).

We define that the k^th magnet is integrated into the i^th segment. Now, we first derive a transformation matrix $T^{\mathrm{m}_k}_{i-1}$ from the base frame $\{S_{i-1}\}$ to the magnet frame $\{S_{\mathrm{m}_k}\}$. This can be achieved by evaluating the chosen kinematic model, two of which we report in the Appendix 9.2.2, at the segment coordinate $v = \frac{d_{\mathrm{m}_k}}{L_{0,i}}$. This means that the cylindrical magnet is integrated at a distance, which is measured along the backbone, of d_{m_k} from the base of the segment.

Subsequently, we describe the pose of the j^th sensor with respect to the base of the i^th segment. Denote with $d_{\mathrm{s}_j,\mathrm{r}}$ the radial distance of the sensor from the center line, with φ_j the azimuth angle of the sensor in the cylindrical plane, and with $d_{\mathrm{s}_j,\mathrm{a}}$ the axial distance along the center line from the base of the i^th segment. We derive the transformation $T^{\mathrm{s}_j,\mathrm{r}_0}_{i-1}$ to the center of the cylindrical plane of the sensor analogously as for the magnets by evaluating the forward kinematics at $v = \frac{d_{\mathrm{s}_j,\mathrm{a}}}{L_{0,i}}$. This is followed by applying the radial offset $d_{\mathrm{s}_j,\mathrm{r}}$ in

the cylindrical plane of the sensor

$$T_{i-1}^{\mathrm{s}_j} = T_{i-1}^{\mathrm{s}_j,\mathrm{r}_0} \begin{bmatrix} \cos\varphi_j & -\sin\varphi_j & 0 & d_{\mathrm{s}_j}\cos(\varphi_j) \\ \sin\varphi_j & \cos\varphi_j & 0 & d_{\mathrm{s}_j}\sin(\varphi_j) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{9.7}$$

Optionally, a static rotation offset can be applied to $R_{i-1}^{\mathrm{s}_j}$ such that local z-axis $\{o_{\mathrm{s}_j}\}$ corresponds to the sensor measurement direction.

Knowing the transformation matrices from the base frame of the respective segment to the sensor and magnet frames, we express them in the inertial frame $\{S_0\}$ by multiplying with the kinematic chain $T_0^{i-1} = \Pi_{\tilde{i}=1}^{i-1} T_{\tilde{i}-1}^{\tilde{i}}$.

Next, we need to express the sensor measurement direction $\{o_{\mathrm{s}_j}\}_0$ and the cylindrical axis of the magnet $\{o_{\mathrm{m}_k}\}_0$ in the inertial frame

$$\{o_{\mathrm{s}_j}\}_0 = R_0^{\mathrm{s}_j} \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^\top, \quad \{o_{\mathrm{m}_k}\}_0 = R_0^{\mathrm{m}_k} \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^\top. \tag{9.8}$$

As the sensor measures contributions of the earth's magnetic field, we need to state the angle λ_j between $\{o_{\mathrm{s}_j}\}_0$ and the earth's magnetic field unit vector $\{n_{\mathrm{e}}\}_0$. Similarly, we investigate the angle α_j^k between $\{o_{\mathrm{s}_j}\}_0$ and the cylindrical axis of the magnet $\{o_{\mathrm{m}_k}\}_0$

$$\cos(\lambda_j) = \{n_{\mathrm{e}}\}_0 \cdot \{o_{\mathrm{s}_j}\}_0, \quad \cos(\alpha_j^k) = \{o_{\mathrm{m}_k}\}_0 \cdot \{o_{\mathrm{s}_j}\}_0. \tag{9.9}$$

We define the translation and distance between the magnet and the sensor in the frame $\{S_0\}$ as:

$$p_j^k = p_0^{\mathrm{s}_j} - p_0^{\mathrm{m}_k}, \qquad d_j^k = \|p_j^k\|_2. \tag{9.10}$$

Building on the derivation in (9.10), we compute the angles β_j^k and θ_j^k using the dot product rule

$$\cos(\beta_j^k) = \frac{p_j^k \cdot \{o_{\mathrm{s}_j}\}_0}{\|p_j^k\|_2}, \qquad \cos(\theta_j^k) = \frac{p_j^k \cdot \{o_{\mathrm{m}_k}\}_0}{\|p_j^k\|_2}. \tag{9.11}$$

Lastly, the kinematic descriptions for all sensors are vertically stacked as

$$\xi = \left(\xi_1^\top \dots \xi_j^\top \dots \xi_{n_{\mathrm{s}}}^\top\right)^\top \in \mathbb{R}^{n_{\mathrm{s}}+4n_{\mathrm{m}}n_{\mathrm{s}}}. \tag{9.12}$$

We will in the following refer to the mapping $f_\xi(q) : q \in \mathbb{R}^{n_{\mathrm{q}}} \to \xi \in \mathbb{R}^{n_{\mathrm{s}}+4n_{\mathrm{m}}n_{\mathrm{s}}}$ as the magnet sensor kinematics.

9.2.6 Data-driven Sensor Measurement Model

We use a data-driven approach to learn the forward sensor model $\hat{u} = f_\pi(\xi_j)$ for each sensor using a neural network parameterized with π. We note that the same neural network weights can be shared for all sensors, but oftentimes, performance can be improved by training a specialized model with weights π_j for each sensor. During training on a dataset of length n_{t}, we minimize the MSE between the predicted sensor measurement $\hat{u}_j$ and the actual sensor measurement u_j:

$$\min_\pi \frac{1}{n_{\mathrm{t}}} \sum_{t=0}^{n_{\mathrm{t}}} \left(f_\pi(\xi_j(t)) - u_j(t)\right)^2, \tag{9.13}$$

Algorithm 2 Proprioception with magnetic sensors

Input: $u(t) \in \mathbb{R}^{n_\mathrm{s}}$ ▷ Sensor measurements
Input: $\hat{q}(t-1) \in \mathbb{R}^{3n_\mathrm{b}}$ ▷ Estimated configuration at the last time step
Input: $f_{\hat{\xi}} : \mathbb{R}^{2n_\mathrm{b}} \to \mathbb{R}^{n_\mathrm{s}+4n_\mathrm{m}n_\mathrm{s}}$ ▷ Magnet sensor kinematics
Input: $f_\pi : \mathbb{R}^{n_\mathrm{s}+4n_\mathrm{m}n_\mathrm{s}} \to \mathbb{R}^{n_\mathrm{s}}$ ▷ Trained neural network
Input: $\gamma \in \mathbb{R}$ ▷ Step size
Output: $\hat{q}(t) \in \mathbb{R}^{2n_\mathrm{b}}$ ▷ Estimated robot configuration

$\hat{q}_0 \leftarrow \hat{q}(t-1)$
$l \leftarrow 0$
$b_l \leftarrow 0 \in \mathbb{R}^{3n_\mathrm{b}}$
while $l < n_\mathrm{it}$ **do**
  $\hat{\xi}_l \leftarrow f_\xi(\hat{q}_l)$
  $\hat{u}_l \leftarrow f_\pi(\hat{\xi}_l)$
  $\hat{u}_l^{\mathrm{error}} \leftarrow \|\hat{u}_l - u(t)\|_2$
  $b_{l+1} \leftarrow \mu\, b_l + \frac{2}{n_\mathrm{s}} \frac{\partial}{\partial \hat{q}_l} f_{\hat{\xi}}^\top(\hat{q}_l) \frac{\partial}{\partial \hat{\xi}_l} f_\pi^\top(\hat{\xi}_l)(\hat{u}_l - u(t))$
  $\hat{q}_{l+1} \leftarrow \hat{q}_l - \gamma\, b_{l+1}$
  $l \leftarrow l+1$
end while
$l^* \leftarrow \mathrm{argmin}_l\, u_l^{error}$
$\hat{q}(t) \leftarrow \hat{q}_l^*$

where t denotes the current time index. Note that whenever we omit the time index in our notation, we always refer to the current time t. Finally, to simplify the notation, we combine each sensor measurement prediction $u_j \in \mathbb{R}$ into an array $u \in \mathbb{R}^{n_\mathrm{s}}$ and stack the neural networks as $f_\Pi(\xi) : \xi \to u$. We discuss the choice of the specific network architecture later in Section 9.3.

9.2.7 Proprioception via Optimization

Now, that we are able to predict the sensor measurement $\hat{u}$ using the composition of the kinematics $f_{\hat{\xi}}(\hat{q})$ and the neural networks $f_\Pi(\hat{\xi})$, we need to optimize the configuration estimate $\hat{q}$ for the predictions $\hat{u}$ to match the actual sensor measurements u as closely as possible. We detail the procedure in Alg. 2. We capture the error between the predicted sensor measurements $\hat{u}$ and the actual sensor measurements u by the MSE loss function we strive to minimize

$$\mathcal{L}_u(\hat{q}) = \sum_{j=1}^{n_\mathrm{s}} \frac{\left(f_\pi(\xi_j) - u_j\right)^2}{n_\mathrm{s}} \tag{9.14}$$

Accordingly, the optimal configuration estimate $\hat{q}$ can be found with

$$\hat{q} = \mathrm{argmin}\, \mathcal{L}_u(\hat{q}_l). \tag{9.15}$$

We optimize the cost function (9.14) through iterative gradient descent, as detailed in Fig. 9.3. The gradient descent is initialized with the best estimate of the previous time-step

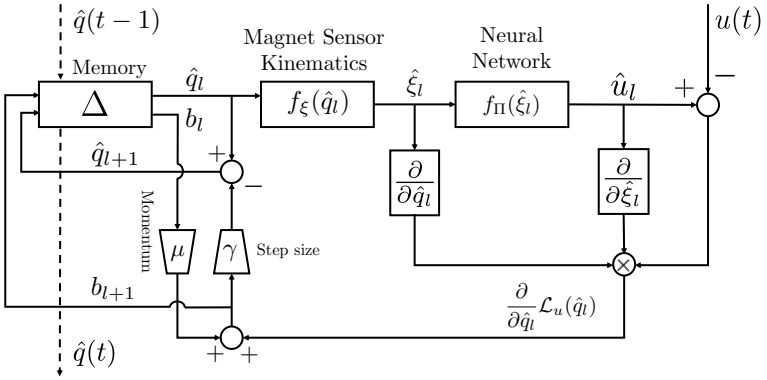


Figure 9.3: A block diagram of the gradient descent as an iterative update loop for the configuration belief $\hat{q}(t)$. The gradient descent is initialized with the optimized solution $\hat{q}_0 = \hat{q}(t-1)$ from the previous time-step. Note that during the iterative loop, $\hat{q}_{l+1}$ updates $\hat{q}_l$ in the memory block.

$\hat{q}_0(t) = \hat{q}(t-1)$. As common in literature, we optimize the state belief $\hat{q}_l$ with a step size γ and the momentum μ using the Jacobian of the loss $\frac{\partial}{\partial \hat{q}_t}\mathcal{L}_u(\hat{q})$

$$b_{l+1} = \mu\, b_l + \frac{\partial}{\partial \hat{q}_t}\mathcal{L}_u(\hat{q}), \quad \hat{q}_{l+1} = \hat{q}_l - \gamma\, b_{l+1}. \tag{9.16}$$

We can use the chain rule to derive an analytical expression for the gradient of the loss incorporating the gradient of the magnet sensor kinematics $\partial_{\hat{q}} f_\xi(\hat{q})$ and the gradient of the neural network $\partial_{\hat{\xi}} f_\Pi(\hat{\xi})$:

$$\frac{\partial}{\partial \hat{q}}\mathcal{L}_u(\hat{q}) = \frac{2}{n_\mathrm{s}}\left(\frac{\partial}{\partial \hat{q}} f_\xi(\hat{q})\right)^\top \left(\frac{\partial}{\partial \hat{\xi}} f_\Pi(\hat{\xi})\right)^\top (\hat{u} - u). \tag{9.17}$$

After executing the gradient descent for n_it iterations, we evaluate which iteration l^* had the lowest loss $\mathcal{L}_u$ and accordingly select $\hat{q}(t) = \hat{q}_{l^*}$ as the best configuration estimate of time-step t.

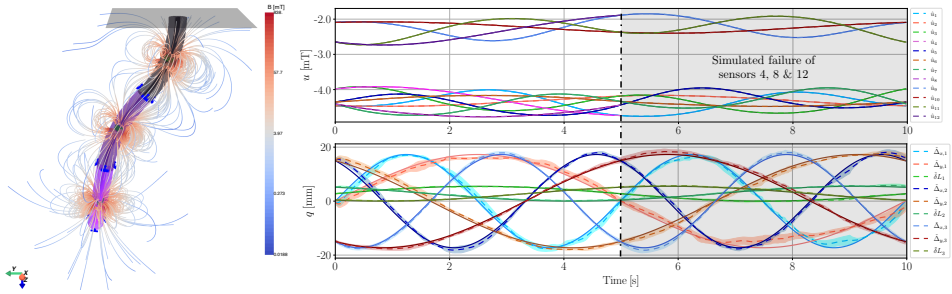
9

9.3 Piecewise Constant Curvature Simulations

We evaluate the proposed methodology for estimating the PCC kinematic configuration $q \in \mathbb{R}^{3n_\mathrm{b}}$ of soft continuum robots thoroughly in simulations. The PCC model allows for the bending and elongation of each segment in 3D space. Please refer to Appendix 9.2.3 for more details. We vary the number of robot segments n_b, remove and add sensors (i.e., change n_s), modify the arrangement of sensors, and the direction of the earth's magnetic field n_e. To motivate some of the unique advantages of our method, we use the same learned neural network weights for all these trials.

9.3.1 Simulation Setup

In our simulations, we consider a robot consisting of one, two, or three segments. All cylindrical segments have an unextended length of $L_{0,i} = 110\,\mathrm{mm}$ and a radius of $d_i = 22\,\mathrm{mm}$.



(a) Magnetic field for a three-segment robot

(b) Simulated sensor failure

Figure 9.4: **Panel (a):** Simulated magnetic field of a robot with three segments. The blue arrows mark the measurement directions of the sensors, the magnets are rendered in green, and the three segments are visualized in a color sequence from black to violet. The magnetic flux density B is shown via streamlines and logarithmic coloring. **Panel (b)**: Sensor measurement predictions (top) and configuration estimates (bottom) for a three-segment robot with twelve sensors nominally. We simulate a sensor failure of the 4th, 8th, and 12th sensors after 5 s of trajectory time by removing the measurements of these sensors from the gradient descent. This can be compensated for automatically by the redundancy of the sensor configuration. We plot the ground-truth values u and q in solid, the estimate $\hat{u}$ and $\hat{q}$ as a mean over three random seeds with dashed lines, and the standard deviation as an error band.

Each segment has a neodymium N50 ring magnet of outer diameter 12 mm, inner diameter 6 mm and thickness 6 mm integrated along the backbone at a distance of $d_{\mathrm{m}_k} = 55\,\mathrm{mm}$ from the base of the segment. In the nominal case, three sensors are placed symmetrically in the tip plane of each segment at a radius of 13 mm from the center with the sensor measurement direction pointing along the local, negative z-axis of the tip plane $\{S_i\}$. We also consider alternative placements of the sensors, which we further detail in Section 9.3.5.

We build on Magpylib (Ortner and Coliado Bandeira, 2020) to simulate the magnetic field behavior. We model the magnets as cylindrical neodymium grade N50 rings with a magnetization of 1450 mT in the local z-direction. After simulating the magnetic field, we rotate the B-field into the local reference frame of each sensor and take the local z-component of the magnetic flux density as the sensor measurement.

9.3.2 Prediction Network

The training set consists of 120 000 random configurations sampled from uniform distribution $\Delta_{x,i} \sim \mathcal{U}(-20.7\,\mathrm{mm}, 20.7\,\mathrm{mm})$, $\Delta_{y,i} \sim \mathcal{U}(-20.7\,\mathrm{mm}, 20.7\,\mathrm{mm})$, and $\delta L_i \sim \mathcal{U}(0, 5.5\,\mathrm{mm})$, where the upper bound represents a bending of the tip of 54° with respect to the base of a segment and an elongation of 5 %. We also randomize the placement of the sensors in the training set. While in the nominal case, the first of the symmetrically placed sensors is placed on the local x-axis, we randomly sample an offset angle $\varphi_{\mathrm{off}} \sim \mathcal{U}(0, \frac{2\pi}{n_{\mathrm{s}_i}})$ for each training sample, where $n_{\mathrm{s}_i} = 3$ is the number of sensors per segment. Finally, we also randomly sample the radial displacement of the sensors from the center with $d_{\mathrm{s,r}} \sim \mathcal{U}(8.7, 17.3)$mm and consider a tilting of the sensors (e.g. a rotation around the tangential axis) with $\psi_{\mathrm{s}} \sim \mathcal{U}(-20°, 20°)$. Before training, we randomly split off 30 % of the training set for validation purposes.

We conducted a selection study involving hyperparameters and feed-forward neural network architectures (number of layers, nodes, and nonlinear activation layer types) on the validation set. In particular, we aimed to generate a smooth loss landscape to improve the gradient descent convergence, leading us to employ a Stochastic Gradient Descent (SGD) (Ruder, 2016) optimizer in conjunction with the Stochastic Weight Averaging (SWA) (Izmailov et al., 2018) strategy. The neural network itself has 18 layers in total and contains, after an initial 1D batch norm layer, four blocks, and is concluded with a fully connected layer at the end. Each block consists of a dropout with a probability of 1 %, a linear layer, a ReLU, and a 1D batch norm layer. The hidden state is first increased to 96 nodes, then to 256 nodes, and finally reduced again to 64 and 24 nodes. We minimize a MSE loss of the neural network prediction $\hat{u}_j(t)$ for 250 epochs with batch size 650 while setting an initial learning rate of 0.18 for the cosine annealing learning rate scheduler (Loshchilov and Hutter, 2016). The SWA (Izmailov et al., 2018) strategy is started after 125 epochs. We train the neural network such that all sensors in the i^{th} segment share the same weights π_i. When the training is finished, we select the model from the epoch with the lowest validation loss and save it for later testing.

(a) $t = 0\,\text{s}$ (b) $t = 2\,\text{s}$ (c) $t = 4\,\text{s}$ (d) $t = 6\,\text{s}$ (e) $t = 8\,\text{s}$ (f) $t = 10\,\text{s}$

Figure 9.5: Sequence of stills for simulated sensor failure as shown in Figure 9.4(b): the number of sensors per segment, which are rendered in blue, is reduced from four to three at $t = 5\,\text{s}$. We visualize the ground-truth shape of the soft robot with full opacity and the estimated configuration with slight transparency. The magnets are rendered in green, and the three segments are visualized in a color sequence that ranges from black to violet.

9.3.3 Optimization

We optimize the configuration variables $\hat{q}_i = (\hat{\Delta}_{x,i}, \hat{\Delta}_{y,i}, \hat{\delta L}_i)^\top$ for each segment to minimize the sensor measurement prediction loss as defined in (9.14). The optimization strategy solely relies on gradient descent and uses the best solution from the last time step $\hat{q}^*(t-1)$ as an initialization $\hat{q}_0(t)$. For the first time step of the trajectory, we initialize with the ground truth. We use a step size $\gamma = 3.5 \cdot 10^{-4}$, $3 \cdot 10^{-3}$, and $2 \cdot 10^{-3}$ during gradient descent for a one-segment, two-segment, and three-segment robot, respectively. For all robots with more than one segment, the step size is reduced by a factor of ten when optimizing the elongation. The momentum μ is 0.3 for all trials and we perform 20 gradient descent iterations for each time step.

9.3.4 Evaluation

We evaluate the performance of our method at estimating the configuration of the segment by computing a relative RMSE metric with respect to the ground-truth configuration $q(t)$

for each configuration variable separately

$$e_{q_\star} = \frac{\sqrt{\sum_{t=0}^{n_\mathrm{t}} \left(\hat{q}_\star(t) - q_\star(t)\right)^2}}{\sqrt{n_\mathrm{t}}\, D_{q_\star}}, \tag{9.18}$$

where n_t is the number of discrete time-steps and $D_{q_\star}$ is the dynamic range of each configuration variable between the maximum and minimum value in the test set. All simulations are evaluated on a full lemniscate trajectory, which is very similar to what is plotted based on experimental data in Fig. 9.9(d). The maximum bending angle of this trajectory is 45° and the elongation of the segment follows a cosine wave with a minimum and maximum at 1.25 % and 3.75 % respectively.

9.3.5 Results

We present all simulation results in Tab. 9.1. In the first section, robots with one to three segments, which all exhibit a nominal sensor placement, are considered separately. Neural networks are trained separately for each of these robots, as the input dimension needs to be adjusted to the number of magnets. The results show that the method works well for robots with 1-3 segments. It can be observed that the estimation error is usually lower for the distal segment(s).

Next, the number of sensors n_s is varied for a three-segment robot. We always apply a symmetrical placement of the sensors in the tip plane of each segment. In the first trial, two sensors are mounted in the tip plane of each segment opposite to each other (6 sensors in total). While the bending along $\Delta_{x,i}$ and the elongation of the segments δL_i can still be estimated, the setup does not contain sufficient information to accurately determine the bending into $\Delta_{y,i}$. While the nominal case of nine sensors in total already achieves relative RMSEs in the range of 1.6 % to 6 %, the proprioception performance can be slightly improved by adding more sensors. We emphasize that the neural networks are not retrained when adding or removing sensors. In Figures 9.4(b) and 9.5, we plot the proprioceptive performance of a three-segment robot with four sensors per segment. Then, we simulated a failure of the 4th, 8th, and 12th sensor at 5 s by removing these sensor measurements from (9.16) (i.e., the gradient descent). Our method is able to adapt without re-training and leverage the nominal redundancy of sensor measurements, as three sensors per segment are sufficient for shape estimation.

Then, two adjusted sensor placements are investigated while keeping the neural network weights constant. First, the sensors are tilted from the nominal case of pointing along the local z-axis by $\psi_\mathrm{s} = 10°$ towards the inside. In a separate simulation, the sensors are moved radially from nominally 13 mm to 16 mm. As the results show, the configuration of all three segments can still be estimated accurately with a mean error of 3.3 %.

Finally, the Earth's magnetic field of magnitude 0.065 mT is added. A separate neural network is trained on a training set with randomly sampled magnetic field vector directions n_e. As the last section of Tab. 9.1 demonstrates, the methodology is able to adapt to any Earth's magnetic field direction by leveraging the λ_j input parameter.

Table 9.1: Simulation results: First, we report the absolute RMSE e_u of sensor measurement predictions on the test set averaged across all sensors on the robot. Next, we state the relative RMSE [%] of each robot configuration estimate. All results are trained on a trajectory with randomly sampled configurations and sensor kinematic parameters for each segment separately and evaluated on a lemniscate trajectory. The first section applies our methodology to robots consisting of a different number of segments n_b with three sensors attached to the tip of each segment. The number of sensors is varied in the second section for a three-segment robot with all sensors placed symmetrically. The third set of trials then investigates how robust the method is to changes in the kinematic parameters of the sensors, such as the tilting angle of the sensors ψ_s and the radial distance of the sensors $d_{\mathrm{s,r}}$. Finally, we apply the Earth's magnetic field along different cardinal directions in the inertial frame. The RMSE of the configuration estimates is normalized with the range of the dataset for each configuration variable as stated in (9.18). We report the error as mean ± stdev and compute the statistics over three different random seeds. The random seed determines the initialization of the neural network weights at the start of the training.

Simulation	**Specifications**	e_u [mT]	$e_{\Delta_{x,1}}$ [%]	$e_{\Delta_{y,1}}$ [%]	$e_{\delta L_1}$ [%]	$e_{\Delta_{x,2}}$ [%]	$e_{\Delta_{y,2}}$ [%]	$e_{\delta L_2}$ [%]	$e_{\Delta_{x,3}}$ [%]	$e_{\Delta_{y,3}}$ [%]	$e_{\delta L_3}$ [%]
Variation of # of segments	n_b=1, n_s=3	0.015 ± 0.002	1.7 ± 1.0	1.8 ± 1.1	2.8 ± 0.6	-	-	-	-	-	-
	n_b=2, n_s=6	0.015 ± 0.001	4.2 ± 1.3	3.9 ± 0.6	6.3 ± 1.3	2.3 ± 0.6	2.2 ± 0.3	4.1 ± 1.3	-	-	-
	n_b=3, n_s=9	0.015 ± 0.002	3.7 ± 2.1	4.7 ± 1.6	6.0 ± 2.0	2.6 ± 1.6	2.5 ± 1.5	5.5 ± 1.6	1.6 ± 1.2	1.6 ± 1.1	2.7 ± 1.4
Variation of # of sensors	n_b=3, n_s=6	0.015 ± 0.002	3.9 ± 1.2	24.4 ± 2.7	8.4 ± 2.6	2.5 ± 1.3	53.2 ± 5.0	6.0 ± 1.7	1.5 ± 1.1	52.3 ± 7.3	3.1 ± 1.0
	n_b=3, n_s=9	0.015 ± 0.002	3.7 ± 2.1	4.7 ± 1.6	6.0 ± 2.0	2.6 ± 1.6	2.5 ± 1.5	5.5 ± 1.6	1.6 ± 1.2	1.6 ± 1.1	2.7 ± 1.4
	n_b=3, n_s=12	0.015 ± 0.001	3.7 ± 1.6	3.9 ± 1.3	4.6 ± 2.4	2.6 ± 1.3	2.6 ± 1.2	4.4 ± 1.9	1.6 ± 1.2	1.5 ± 1.3	2.5 ± 1.6
	n_b=3, n_s=18	0.016 ± 0.002	3.2 ± 1.5	3.5 ± 1.3	4.2 ± 1.8	2.4 ± 1.3	2.5 ± 1.2	4.4 ± 1.8	1.5 ± 1.3	1.3 ± 1.1	2.5 ± 1.5
Nominal	n_b=3, n_s= 9	0.015 ± 0.002	3.7 ± 2.1	4.7 ± 1.6	6.0 ± 2.0	2.6 ± 1.6	2.5 ± 1.5	5.5 ± 1.6	1.6 ± 1.2	1.6 ± 1.1	2.7 ± 1.4
Sensors tilted	ψ_s=10°	0.015 ± 0.002	8.1 ± 4.0	6.4 ± 2.0	4.7 ± 0.7	5.1 ± 2.3	3.8 ± 0.8	5.6 ± 2.4	2.8 ± 1.2	1.9 ± 0.8	4.1 ± 0.6
Sensors shifted	$d_{\mathrm{s,r}}$=16 mm	0.017 ± 0.003	3.4 ± 1.5	4.0 ± 1.3	6.2 ± 2.5	2.6 ± 1.0	2.8 ± 1.8	6.7 ± 0.9	1.6 ± 1.2	1.6 ± 1.0	4.0 ± 2.0
Earth magnetic field	n_e=(1, 0, 0)	0.012 ± 0.002	2.0 ± 0.4	2.4 ± 1.1	4.3 ± 1.3	1.8 ± 0.7	1.9 ± 0.3	3.6 ± 2.0	1.5 ± 0.3	1.6 ± 0.4	3.2 ± 0.9
	n_e=(0, 1, 0)	0.012 ± 0.002	1.9 ± 0.5	2.5 ± 0.8	3.8 ± 1.1	1.9 ± 0.5	2.0 ± 0.2	3.4 ± 1.6	1.5 ± 0.3	1.6 ± 0.4	3.3 ± 0.8
	n_e=(0, 0, 1)	0.012 ± 0.001	2.0 ± 0.7	2.6 ± 0.9	4.1 ± 0.9	1.6 ± 0.3	1.7 ± 0.3	4.9 ± 1.5	1.5 ± 0.5	1.4 ± 0.1	3.9 ± 0.3

9.4 Affine Curvature Simulations

To demonstrate the efficacy of the proposed method for higher-order kinematic models than PCC, we also conduct simulations of an AC soft robot. The AC kinematic parametrization (Della Santina, 2020) has been shown capable of representing the shape of soft tentacles (Stella et al., 2023c, 2022) and provides a continuous function $\kappa = \kappa_0 + \kappa_1 v$ to describe the curvature of the soft robot, where κ_0, κ_1 are two configuration variables and $v \in [0, 1]$ is the backbone coordinate. We allow for movement in 3D space by also specifying an azimuth bending angle ϕ and the elongation δL. Please refer to Section 9.2.4 for more implementation details about the AC model.

9.4.1 Simulation Setup

We use the same simulation setup as described in Section 9.3.1. Therefore, we report in the following only the implemented modifications to simulate an AC soft robot in Magpylib (Ortner and Coliado Bandeira, 2020). Namely, we consider one AC segment of length $L_{0,i} = 200\,\text{mm}$ with in total $n_\text{s} = 9$ magnetic sensors. The sensors are placed on three separate cylindrical planes at distances d_{s_a} of 0 mm, 100 mm, and 200 mm from the base of the robot. In each plane, the three sensors are spaced at an angle of 120° and at a radial distance $d_{\text{s}_\text{r}} = 13\,\text{mm}$ from the backbone, as it can be seen in Fig. 9.7. Two ring magnets are positioned at a distance of 50 mm and 150 mm from the robot's base, respectively.

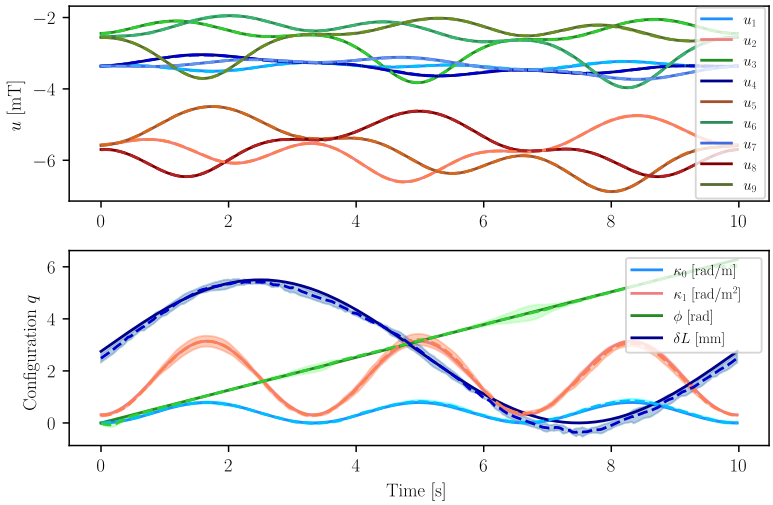


Figure 9.6: Sensor measurement predictions (top) and configuration estimates (bottom) for an AC segment with nine magnetic sensors. We plot the ground-truth values u and q in solid, the estimate $\hat{u}$ and $\hat{q}$ as a mean over three random seeds with dashed lines, and the standard deviation as an error band. The random seed determines the initialization of the neural network weights at the start of the training.

9.4.2 Prediction Network and Optimization

We simulate 120 000 random configurations of the AC robot to generate the training set. For this purpose, we sample the configuration variables from uniform distributions:

(a) $t = 0\,\mathrm{s}$ (b) $t = 2\,\mathrm{s}$ (c) $t = 4\,\mathrm{s}$ (d) $t = 6\,\mathrm{s}$ (e) $t = 8\,\mathrm{s}$ (f) $t = 10\,\mathrm{s}$

Figure 9.7: Sequence of stills for a simulated AC robot. We visualize the ground-truth shape of the soft robot with full opacity and the estimated configuration with slight transparency. The two magnets are rendered in green, and the nine sensors in blue.

the AC parameters $\kappa_0 \in \mathcal{U}(0, 0.942\,\mathrm{rad/m})$, $\kappa_1 \in \mathcal{U}(0, 3.770\,\mathrm{rad/m^2})$, the azimuth angle $\phi \in \mathcal{U}(0, 2\pi\,\mathrm{rad})$, and the elongation $\delta L \in \mathcal{U}(0, 6.6\,\mathrm{mm})$. Before training, we randomly split off 30 % of the training set for validation purposes. We train a specialized neural network $f_{\pi_j(\xi_j)}$ for each sensor and use the same neural network architecture as in Section 9.3.2 with the exception of the addition of a final layer $y(x) = \mathrm{sign}(x)\, e^{|x|}$. The training runs for a total of 250 epochs and uses the SWA (Izmailov et al., 2018) strategy with a learning rate of 0.01. All other training hyperparameters are the same as in Section 9.3.2.

The four configuration variables are optimized to minimize the loss between the predictions and simulated measurements of the nine sensors as defined in 9.14. For this optimization procedure, we employ gradient descent running at 40 Hz with step sizes of $\gamma_{\kappa_0} = 1$, $\gamma_{\kappa_1} = 5$, $\gamma_\phi = 1$, and $\gamma_{\delta L} = 2 \cdot 10^{-4}$. The momentum is set to $\mu = 0.3$ and 20 iterations are performed at each time step.

9.4.3 Evaluation

We evaluate the trained model on a flower trajectory of duration 10 s and sample rate 40 Hz. The evaluation trajectory has the following characteristics: κ_0 is actuated by a sinusoidal wave of frequency 0.3 Hz in the range $[0, \frac{\pi}{4}\,\mathrm{rad/m}]$. Similarly, κ_1 is also varied through a sinusoidal function of the same frequency and has a dynamic range of $[0.1\pi, \pi]\,\mathrm{rad/m^2}$. The azimuth angle ϕ is linearly scaled from 0 rad to 2π rad over the duration of the trajectory. Finally, δL follows a sinusoidal sequence of frequency 0.1 Hz in the range of $[0, 5.5]\,\mathrm{mm}$. We use the same evaluation metrics as first introduced in Section 9.3.4. We report the error as mean ± stdev and compute the statistics over three different random seeds. The random seed determines the initialization of the neural network weights at the start of the training.

9.4.4 Results

The trained neural networks achieve an RMSE error for predicting the magnetic sensor measurements of $0.025 \pm 0.002\,\mathrm{mT}$ on the test set. When we run inference (see Fig. 9.6) on the flower trajectory, the configuration variables can be estimated with an absolute RMSE of $e_{\kappa_0} = 0.042 \pm 0.005\,\mathrm{rad/m}$, $e_{\kappa_1} = 0.11 \pm 0.04\,\mathrm{rad/m^2}$, $e_\phi = 0.08 \pm 0.02\,\mathrm{rad}$, and $e_{\delta L} = 0.001 \pm 0.001\,\mathrm{mm}$. We state the relative RMSE errors for the configuration estimates as $e_{\kappa_0} = 5.4 \pm 0.7\,\%$, $e_{\kappa_1} = 4.0 \pm 1.4\,\%$, $e_\phi = 1.3 \pm 0.4\,\%$, and $e_{\delta L} = 3.8 \pm 1.6\,\%$. In Fig. 9.7, we render at six different points along the trajectory the robot's shape according to the ground truth and estimated configurations, respectively. The sequence qualitatively shows that our proposed method is able to estimate the affine curvature robot's shape very accurately.

9.5 Experiments

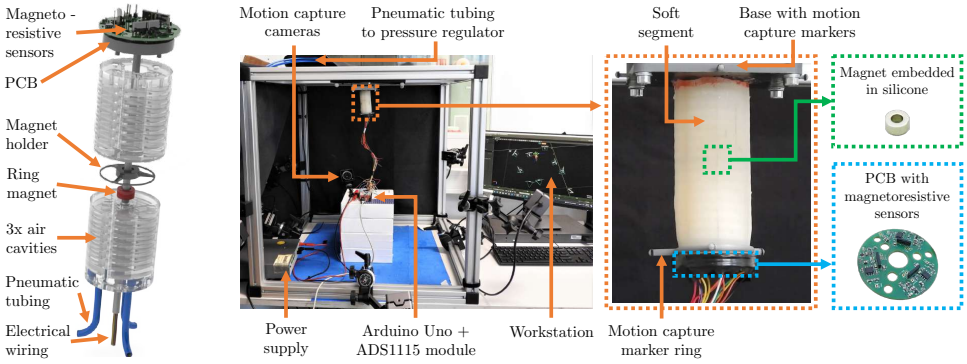


Figure 9.8: **Left:** Exploded rendering visualizing the robot design with the three MRSs integrated into a PCB at the tip of the segment. The electrical wires from the Printed Circuit Board (PCB) are passed through the backbone to the base. A ring magnet attached to a 3D-printed holder is integrated into the backbone at a half-segment-length distance from the proximal end. The air chambers of the segment are connected to a pressure regulator via tubing glued at the base of the segment. **Right:** Experimental setup with the soft robot segment attached in tip-down configuration to the motion capture cage.

We verify the performance of our proposed proprioception method in experiments involving one soft robot segment with three magnetoresistive sensors attached to the tip. We aim to estimate the CC configuration $\hat{q} = (\Delta_{x,1}, \Delta_{y,1})^\top \in \mathbb{R}^2$ from the measured sensor values $u(t) \in \mathbb{R}^3$. We let the robot follow a diverse set of trajectories and evaluate the proprioception performance. After measuring the ground-truth pose of the tip of the segment with a motion capture system, we perform inverse kinematics with the closed-form solution reported by Della Santina et al. (2020a) and quantitatively compare the proprioceptive configuration estimate of the segment with the ground-truth configuration.

9.5.1 Robot Design

We use a cylindrical, pneumatically-actuated soft robotic silicon segment of length $L_0 = 110\,\text{mm}$ and radius $d_1 = 22\,\text{mm}$ consisting of three independently inflatable cavities evenly spaced in the radial direction from the center line (Marchese et al., 2015). The proprioceptive sensing system is achieved by embedding one ring magnet in the backbone at a distance $d_{\text{m}_0} = 55\,\text{mm}$ from the base of the segment and three symmetrically-placed MRSs at the tip of the segment as visualized in Fig. 9.8. Although we use MRSs in our experimental setup for their high sensitivity (Popovic et al., 2002), this is not a strict condition, and other sensor types measuring the magnetic field, such as Hall-effect sensors, can be combined with the methodology proposed in this chapter, too. For casting the silicone segment, we use a 3D-printed mold with a magnet holder that keeps it in place inside the segment. The magnet used is a neodymium ring of grade N50 with a thickness and inner diameter of 6 mm each and an outer diameter of 12 mm. The MRSs of type Honeywell HMC1021Z are integrated into a Printed Circuit Board (PCB) and output a voltage difference of 50 mV/mT. The three sensors are equally spaced at 120° from each other and are placed at a radial distance of $d_{\text{s,r}} = 13\,\text{mm}$, and at a longitudinal distance of $d_{\text{s,a}} = 116\,\text{mm}$ from the base in a

Table 9.2: Experimental results: absolute [mm] and relative RMSE [%] of sensor measurement predictions and robot configuration estimates for various trajectories. The RMSE is normalized with the range of the dataset for u and each configuration variable, respectively, as stated in (9.18). We report the error as mean ± stdev and compute the statistics over three different random seeds. The random seed determines the initialization of the neural network weights at the start of the training.

Trajectory	e_u [mV]	e_u [%]	e_{Δ_x} [mm]	e_{Δ_x} [%]	e_{Δ_y} [mm]	e_{Δ_y} [%]
T5.train → T0.test	9.90 ± 0.90	3.90 ± 0.40	0.37 ± 0.06	3.70 ± 0.60	0.44 ± 0.01	3.50 ± 0.10
T5.train → T1.test	8.80 ± 0.30	4.40 ± 0.10	0.36 ± 0.08	6.50 ± 1.50	0.43 ± 0.05	-
T5.train → T2.test	11.10 ± 0.30	4.30 ± 0.10	0.78 ± 0.03	13.60 ± 0.60	0.74 ± 0.23	5.90 ± 1.80
T5.train → T3.test	12.30 ± 0.30	4.60 ± 0.10	0.52 ± 0.06	4.50 ± 0.50	0.47 ± 0.03	3.10 ± 0.20
T5.train → T4	3.01 ± 0.05	1.08 ± 0.02	0.33 ± 0.02	2.40 ± 0.10	0.52 ± 0.06	3.10 ± 0.40
T5.train → T5.test	1.60 ± 0.10	0.58 ± 0.05	0.24 ± 0.05	1.90 ± 0.30	0.24 ± 0.01	1.40 ± 0.05

straight configuration. For each sensor, we implemented a Set / Reset and an amplification circuit on the PCB. The Set / Reset circuit is used to calibrate the sensor by re-aligning the magnetic domains. After amplification of the sensor output by a factor of 100, the output of the sensors is processed with a Texas Instruments ADS1115 module resulting in a digital signal of 16 bit resolution. All sensor measurements u are in the range $[0\,\text{mV}, 2048\,\text{mV}]$, which corresponds to magnetic flux densities of $[0\,\text{mT}, 41\,\text{mT}]$.

9.5.2 Experimental Setup

We conducted our experiments in a lab environment with the base of the soft robot segment mounted in a tip-down configuration to a cubical cage, as shown in Figure 9.8. A 3D-printed ring with four reflective markers is mounted on the tip of the segment. Eight motion capture cameras are attached to the cage, tracking at 40 Hz the 3D pose of the ring. We transform the pose measurements of the tip to the base frame of the robot and compute the closed-form inverse kinematics (Della Santina et al., 2020a) to receive a ground-truth configuration estimate $q(t) \in \mathbb{R}^2$. Each of the three pneumatic chambers of the segment is connected via tubing to a separate valve of a proportional pressure regulator operated at 100 Hz. We read out the analog signals of the MRSs with an Arduino Uno at 40 Hz and save them for later offline processing. We temporally align the motion capture and the magnetic sensor data by detecting the initial extension of the robot with a suitable threshold. The sensor noise is determined for both an unelongated straight configuration and during fully inflated bending. Here, the standard deviations of the white noise are 0.24 mV and 3.55 mV, which normalizes to 0.03 % and 2 % of the dynamic range, respectively. Furthermore, we identify the earth's magnetic field direction in the base from as $\{\hat{n}_e\}_0 = (-0.311, -0.234, 0.921)^\top$ using a compass and the World Magnetic Model (WMM) (Chulliat et al., 2020).

9.5.3 Pneumatic Actuation and Trajectories

We consider, as visualized in Fig. 9.9, six continuous actuation sequences in this chapter: random configuration way-points which are connected through linear interpolation (T0), planar side bending (T1), the tip following a half lemniscate (T2) and full lemniscate (T3), a spiral with constant linear velocity (Carrasco-Zevallos et al., 2018) (T4) and finally a flower-shape (T5). We define our trajectories as wrenches $\tau_{xyz} = \begin{bmatrix} \tau_x & \tau_y \end{bmatrix}^\top$ on the tip of the segment in Cartesian space, where τ_x and τ_y cause bending around the local x- and

y-axis of the tip respectively. The pressures we command from the pressure regulator are given by inversely evaluating the force produced at the center of pressure at the tip of the segment for each chamber for a given chamber pressure (Della Santina et al., 2019a). All actuation sequences are preceded by first applying an offset pressure of 225 mBar in all chambers, which causes a near-constant elongation of the segment. The peak pressure, which causes maximum bending, is set for all trajectories to 450 mBar.

The 1D bending (T1), half lemniscate (T2) and full lemniscate (T3) are all executed periodically with a period of 5 s, 10 s, and 10 s respectively. Trajectories T0 and T4 are characterized by a constant velocity in torque-space of 0.025 Nm/s and 0.0125 kNm/s respectively. The flower trajectory T5 can be described as periodic 1D bending with a linearly changing azimuth angle. It exhibits an angular velocity of 0.0126 rad/s and a period of 10 s for the bending, which results in 50 bending cycles per circumnavigation. While the random configuration setpoints of T0 are recorded for 200 s, T1, T2 and T3 have total a duration of 90 s each, and the spiral T4 and flower T5 last for 120 s and 1500 s respectively. We split off the final 20 % of all datasets as a test set.

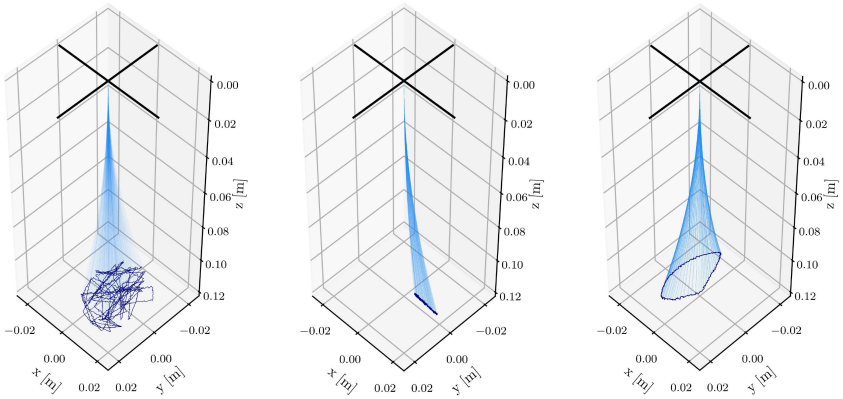
9.5.4 Prediction Network and Optimization

We use the same neural network architecture and training procedure as in Section 9.3.2, but with an adjusted initial learning rate of $5 \cdot 10^{-5}$ and train the model separately for each sensor on the 1200 s long T5 / flower trajectory, which results in 48 000 training samples for each neural network.

We optimize the configuration variables Δ_x and Δ_y for the one segment to minimize the sensor measurement prediction loss as defined in (9.14). The optimization strategy solely relies on gradient descent running at 40 Hz with a step size $\gamma = 1.5 \cdot 10^{-8}$ and momentum $\mu = 0.2$. We visualize a sample loss landscape in Fig. 9.10(a).

9.5.5 Results

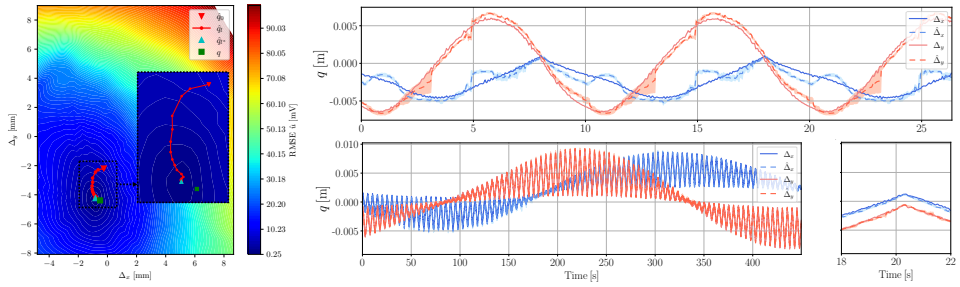
First, we quantify the performance of the neural network predicting the sensor measurements $\hat{u}(t)$ for a known, ground-truth configuration $q(t)$, which we report in Table 9.2. We observe that the relative RMSE lies between 0.6 % and 4.6 % of the range of the respective datasets with a mean of 3.1 %. As expected, the predictions are generally the most accurate when evaluated on a trajectory of the same type as the neural network was trained on (T5). Next, we analyze the proprioception performance on the same trajectories. We report relative RMSEs between 1.9 % and 13.6 % for estimating the bending of the robot. Additionally, we visualize the configuration estimates for two trajectories: in the top-right of Fig. 9.10(b), we run inference for our trained model on T2. While the proprioception estimate tracks the general shape of the trajectory well, the optimization, particularly for Δ_y, gets trapped in local minima, sometimes leading to periods of higher error. Next, we consider a model trained and evaluated on separated sets of the T5 trajectory. The configuration estimate tracks the ground truth very well, as can be seen at the bottom right. Finally, we present a sequence of stills based on camera views and renderings of the soft segment for inference on T3 in Fig. 9.11.



(a) **T0:** random setpoints (b) **T1:** 1D bending (c) **T2:** half lemniscate

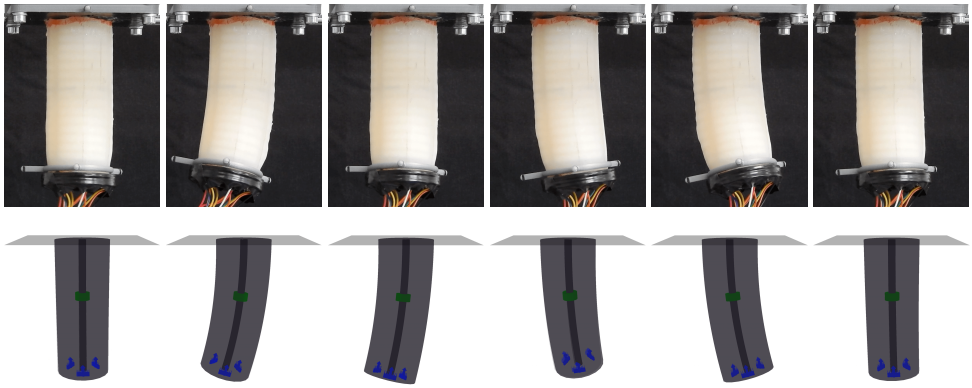
(d) **T3:** full lemniscate (e) **T4:** spiral (f) **T5:** flower

Figure 9.9: Trajectories used during the experiments. We plot the shape of the segment under PCC approximation in light blue and the position of the tip of the segment in dark blue.



(a) Loss landscape (b) Experimental results for T2 (top) and T5 (bottom)

Figure 9.10: **Panel (a):** Sample loss landscape for optimization of Δ_x and Δ_y on T5. With the hue, we visualize the RMSE of the sensor measurement prediction $\hat{u}$ for a given configuration $\hat{q} = (\Delta_x, \Delta_y)^\top$. Additionally, we denote the initial configuration estimate with $\hat{q}_0$, the trajectory of the gradient descent with $\hat{q}_l$, the optimal configuration with $\hat{q}$, and the ground truth with q. **Panel (b) top:** Proprioception on the test set of T2 using a model trained on T5. **Panel (b) bottom:** Configuration estimates for a model trained and evaluated on separated parts of trajectory 5. We plot the ground-truth configuration q in solid, the estimate $\hat{q}$ as a mean over three random seeds with dashed lines, and the standard deviation as an error band. The bottom right plot zooms onto a selected part of T5 (e.g., 18 s to 22 s) to more clearly visually distinguish the dashed lines from the solid lines.



(a) $t = 0\,\mathrm{s}$ (b) $t = 2\,\mathrm{s}$ (c) $t = 4\,\mathrm{s}$ (d) $t = 6\,\mathrm{s}$ (e) $t = 8\,\mathrm{s}$ (f) $t = 10\,\mathrm{s}$

Figure 9.11: Sequence of stills for inference on T3 (full lemniscate) of a model trained on T5. The top row shows camera recordings of an external view, and the sequence in the bottom row consists of renderings of the ground-truth state (in full opacity) and the estimated segment shape (slightly transparent). The sensors are colored in blue, and the ring magnet integrated into the backbone is shown in green.

9.6 Conclusion

This chapter proposed a sensing strategy for soft-bodied robots that relies on multiple magnetic sensors embedded directly in the robot. Thanks to a novel kinematics-aware neural architecture, we can simultaneously use information coming from all the sensors to reconstruct the full robot shape reliably. The decoupling of the kinematics from the learned sensor measurement predictor allows modifications to the placement of the sensors without requiring a re-training of the neural network. The proposed method is agnostic to the used kinematic state parametrization, which we verified in simulations using either PCC or AC models. Extensive experiments with a soft segment showed that a model can be trained on one trajectory type and then be used for inference on a variety of other trajectories in the same workspace. In future work, we will validate the proposed proprioception methodology to execute closed-loop control. We also invite future research studying the optimal placement of sensors in continuum soft robots, where the optimization procedure might take advantage of the gradients provided by our algorithm. Additionally, we would like to validate the proprioception methodology, which we have developed for the general case of n_b segments, experimentally on a multi-segment robot.

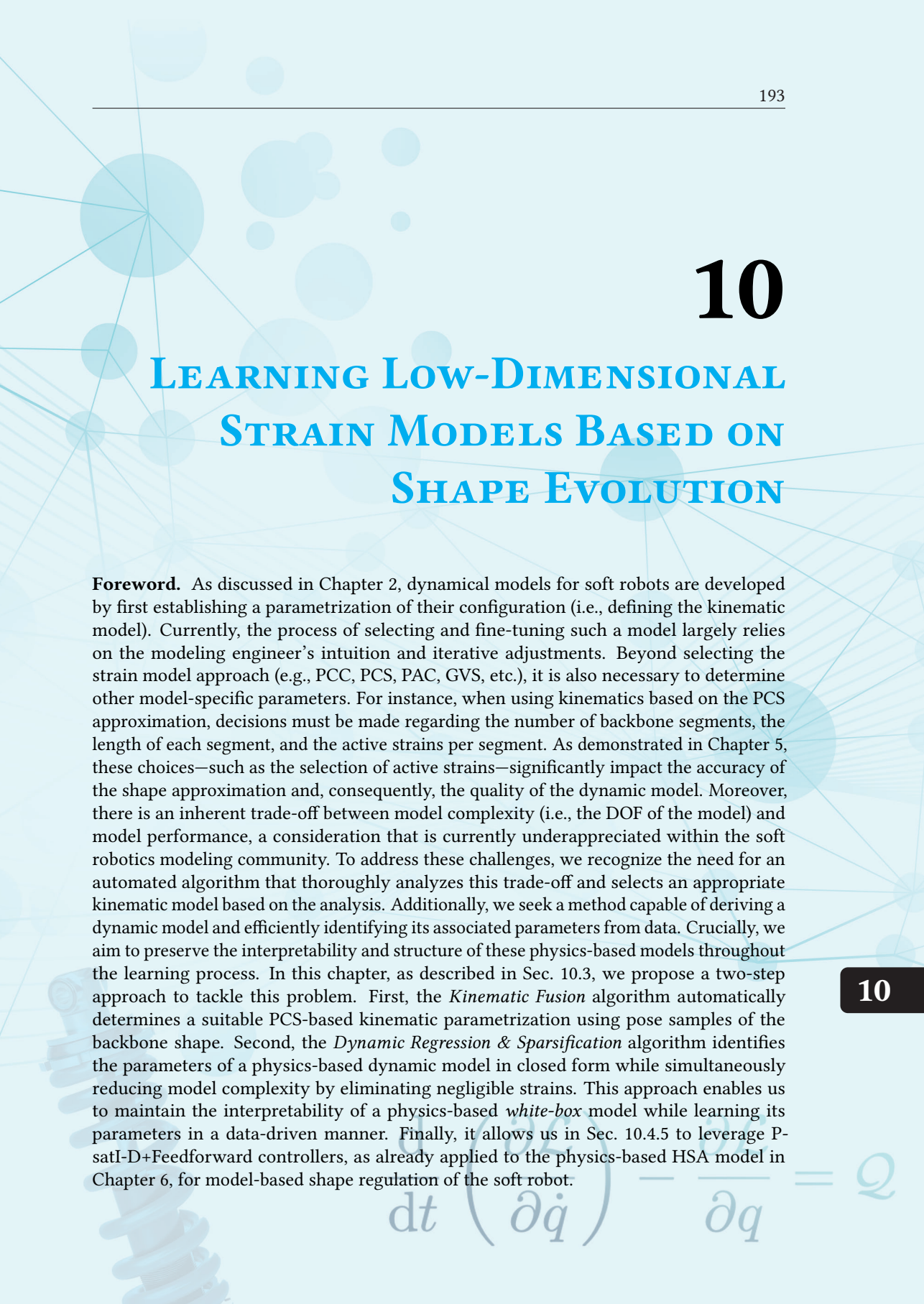
Afterword

This chapter presented an example of how prior knowledge, in this case, a soft robot kinematic model, can be leveraged to simplify a learning problem, which in turn improves the data efficiency of the learning, as we drastically reduced the number of inputs and outputs compared to a naive solution of learning proprioception end-to-end. Essentially, the prior knowledge here allowed us to reduce the size of the learned *black-box* by pulling out the kinematics. In the following two chapters, Chapters 10 & 11, we again strive to leverage prior knowledge, this time for learning the dynamics of soft robots. Contrary to this chapter, instead of reducing the size of the *black-box*, we endeavor towards a *grey-box* model that incorporates physical structure and exhibits certain stability guarantees.

10

Learning Low-Dimensional Strain Models Based on Shape Evolution

Foreword. As discussed in Chapter 2, dynamical models for soft robots are developed by first establishing a parametrization of their configuration (i.e., defining the kinematic model). Currently, the process of selecting and fine-tuning such a model largely relies on the modeling engineer's intuition and iterative adjustments. Beyond selecting the strain model approach (e.g., PCC, PCS, PAC, GVS, etc.), it is also necessary to determine other model-specific parameters. For instance, when using kinematics based on the PCS approximation, decisions must be made regarding the number of backbone segments, the length of each segment, and the active strains per segment. As demonstrated in Chapter 5, these choices—such as the selection of active strains—significantly impact the accuracy of the shape approximation and, consequently, the quality of the dynamic model. Moreover, there is an inherent trade-off between model complexity (i.e., the DOF of the model) and model performance, a consideration that is currently underappreciated within the soft robotics modeling community. To address these challenges, we recognize the need for an automated algorithm that thoroughly analyzes this trade-off and selects an appropriate kinematic model based on the analysis. Additionally, we seek a method capable of deriving a dynamic model and efficiently identifying its associated parameters from data. Crucially, we aim to preserve the interpretability and structure of these physics-based models throughout the learning process. In this chapter, as described in Sec. 10.3, we propose a two-step approach to tackle this problem. First, the *Kinematic Fusion* algorithm automatically determines a suitable PCS-based kinematic parametrization using pose samples of the backbone shape. Second, the *Dynamic Regression & Sparsification* algorithm identifies the parameters of a physics-based dynamic model in closed form while simultaneously reducing model complexity by eliminating negligible strains. This approach enables us to maintain the interpretability of a physics-based *white-box* model while learning its parameters in a data-driven manner. Finally, it allows us in Sec. 10.4.5 to leverage P-satI-D+Feedforward controllers, as already applied to the physics-based HSA model in Chapter 6, for model-based shape regulation of the soft robot.



Abstract. *Obtaining dynamic models of continuum soft robots is central to the analysis and control of soft robots, and researchers have devoted much attention to the challenge of proposing both data-driven and first-principle solutions. Both avenues have, however, shown their limitations; the former lacks structure and performs poorly outside training data, while the latter requires significant simplifications and extensive expert knowledge to be used in practice. This paper introduces a streamlined method for learning low-dimensional, physics-based models that are both accurate and easy to interpret. We start with an algorithm that uses image data (i.e., shape evolutions) to determine the minimal necessary segments for describing a soft robot's movement. Following this, we apply a dynamic regression and strain sparsification algorithm to identify relevant strains and define the model's dynamics. We validate our approach through simulations with various planar soft manipulators, comparing its performance against other learning strategies, showing that our models are both computationally efficient and 25x more accurate on out-of-training distribution inputs. Finally, we demonstrate that thanks to the capability of the method of generating physically compatible models, the learned models can be straightforwardly combined with model-based control policies.*

10

This chapter is partly based on *R. Valadas*, **M. Stölzle***, J. Liu, and C. Della Santina (2025). Learning Low-Dimensional Strain Models of Soft Robots by Looking at the Evolution of Their Shape with Application to Model-Based Control. In Proceedings of the 2025 IEEE 8th International Conference on Soft Robotics (RoboSoft) (pp. 1-8). IEEE.* (Valadas et al., 2025).

*R.V. and M.S. contributed equally to this work. M.S. conceived the research project, devised the overall approach, and wrote the paper. R.V. implemented the methodology, came up with the kinematic fusion algorithm, was in charge of most simulations, and created most graphics. M.S. implemented and compared against the baseline approaches and conducted the closed-loop control simulations. M.S., J.L., and C.D.S. contributed ideas, supervised the research project, and revised the manuscript. C.D.S. provided the funding.

10.1 Introduction

Modeling the dynamical behavior (Armanini et al., 2023) of soft robots with computationally tractable models is important for many applications, such as efficient simulation (Alkayas et al., 2025), model-based control (Della Santina et al., 2023), state estimation (Shao et al., 2023), and co-design (Wang et al., 2024). Developing such (low-dimensional) dynamic models is challenging and is an active area of research (Alora et al., 2023a; Armanini et al., 2023). The use of data-driven approaches has been extensively investigated in this context (Alora et al., 2023a; Bruder et al., 2020; Chen et al., 2024; Thuruthel et al., 2017). These learned models exhibit poor extrapolation performance (Kim et al., 2021), a lack of interpretability, and (physical) structure preventing us from directly leveraging closed-form control solutions such as the PD+feedforward (Della Santina et al., 2023). Instead, researchers had to fall back to computationally expensive planning methods such as MPC (Alora et al., 2023a; Bruder et al., 2020).

The traditional avenue established by the robotics and continuum dynamics communities has been to derive the dynamical model directly from first principles (Armanini et al., 2023; Boyer et al., 2020; Della Santina et al., 2023; Gazzola et al., 2018; Grazioso et al., 2019; Renda et al., 2018) which provides physical interpretability and structure at the cost of needing substantial expert knowledge, for example, in the selection of the proper kinematic approximations (e.g., PCC (Webster III and Jones, 2010), PCS (Renda et al., 2018), GVS (Boyer et al., 2020)). Suboptimal choices or even errors in applying this modeling procedure can lead to significant issues like inaccurate predictions and overly complex models. This hinders the democratization of soft robots, as only specialized research labs possess the required expertise (Aracri et al., 2024).

Recently, there has been a community push towards integrating physical structures and stability guarantees into learned models (e.g., Lagrangian Neural Networks (Liu et al., 2024a), residual dynamical formulations (Bruder et al., 2024; Gao et al., 2024), or oscillatory networks (Stölzle and Della Santina, 2024)) which combine benefits from both worlds: they are learned directly from data which reduces the expert knowledge that is needed but at the same time exhibit a physical structure that can be exploited for model-based control and stability analysis. This work positions itself in this new trend of research, specifically focusing on deriving kinematic and dynamic models for continuum soft robots in a data-driven way.

Indeed, deriving reduced-order kinematic representations remains the core challenge in physics-based modeling. Previous works have relied heavily on the modeling engineer's intuition and experience to make decisions on the number of PCS segments, the length of each segment, and which strains to consider (Toshimitsu et al., 2021). However, these decisions are not straightforward and could easily result in models that are higher-dimensional than necessary, or that important strains are ignored based on a wrong intuition (Garg et al., 2022). Very recently, Alkayas et al. (2025) proposed a data-driven algorithm to identify the optimal discrete GVS (Boyer et al., 2020) strain basis of continuum soft robots via Proper Orthogonal Decomposition (POD). However, the discrete strain basis requires careful numerical integration at runtime, and identifying the dynamical parameters (e.g., stiffness, damping coefficients) of the soft robot relies upon solving a nonlinear least-squares problem that is not always well behaved (Stölzle et al., 2024b).

This chapter proposes to solve these challenges by introducing an end-to-end approach

that can automatically learn both a PCS kinematic parametrization and the corresponding dynamical model, including its dynamic parameters, directly from image/Cartesian pose data. First, a kinematic fusion algorithm aims to minimize the DOFs of the PCS kinematic model while preserving a desired shape reconstruction accuracy for the given discrete shape measurements in Cartesian space. In contrast to previous work (Alkayas et al., 2025), we do not necessitate prior knowledge about strain discontinuities. Secondly, an integrated strategy is proposed to simultaneously sparsify the strains of the PCS model and estimate the parameters of the dynamical model with closed-form linear least-squares. Contrary to common symbolic regression approaches such as Sparse Identification of Nonlinear Dynamics (SINDy) (Kaiser et al., 2018), we crucially preserve the (physical) structure of the Euler-Lagrangian dynamics as derived according to the PCS model. This feature allows the derived dynamical model to be subsequently rapidly deployed within established model-based controllers (Della Santina et al., 2023).

We verify the approach in simulation in a diverse set of scenarios, including different robot topologies and the performance when measurement noise is present. Impressively, the method is able to accurately perform long-horizon (7 s) shape predictions when being trained on 4 s of trajectory data. We benchmark the proposed approach against several state-of-the-art dynamical model learning approaches (e.g., NODEs, CON, LNNs). On the training set, our proposed method exhibits a 70 % lower shape prediction error than the best-performing baseline method (NODE). However, we find that the difference is even greater in extrapolation scenarios (i.e., actuation sequences and magnitudes unseen during training): Here, our proposed method reduces the shape prediction error on the test set by 96 % compared to the best performing ML baseline (NODE). Finally, we demonstrate how the Lagrangian structure of the identified dynamical model allows us to easily design a model-based controller that is effective at regulating the shape of the soft robot.

A video attachment presents the research idea & methodology and contains supplementary plots and animations of the results presented in the chapter.[1]

10.2 Preliminaries

In the following, we will provide some background on the PCS kinematic model and the derivation of soft robot dynamics following a Lagrangian approach, which are two fundamental topics for this chapter.

10.2.1 Piecewise Constant Strain (PCS) Kinematics

The PCS model (Renda et al., 2018) describes the kinematics of continuum soft robots by assuming that the six elemental local backbone strains (shear, axial, bending, twist) are piecewise constant across n_s segments but variable in time. We remark that other popular kinematic models for soft robots, such as PCC (Webster III and Jones, 2010), are often a special case of the PCS model.

For the planar case, the configuration of the ith segment is referred to as

$$q_i = \begin{bmatrix} \kappa_{be,i} & \sigma_{sh,i} & \sigma_{ax,i} \end{bmatrix}^\top \in \mathbb{R}^3, \tag{10.1}$$

[1] https://youtu.be/dfO-PhDIiHI

where $\kappa_{\text{be},i}, \sigma_{\text{sh},i}, \sigma_{\text{ax},i}$ are the bending, shear, and axial strains, respectively, and $i \in \{1, \dots, n_\text{s}\}$. Therefore, the configuration of the entire soft robot is defined as $q \in \mathbb{R}^{n_\text{q}}$, where $n_\text{q} = 3n_\text{s}$. We also have access to closed-form expressions for the forward and inverse kinematics of a single constant strain segment (Stölzle et al., 2024b). As a consequence, forward and inverse kinematics for the entire planar PCS soft robot can be implemented using an iterative procedure starting at the proximal end without having to resort to differential (inverse) kinematic techniques. Namely, the forward kinematics $\pi : \mathbb{R}^{n_\text{q}} \to SE(2)$ allow us to compute the pose $\chi_j = \begin{bmatrix} p_{\text{x},j} & p_{\text{y},j} & \theta_j \end{bmatrix}^\top = \vartheta(q, s_j)$, where $s_j \in [0, L]$ is the backbone abscissa/coordinate, L is the length of the entire continuum structure in an undeformed configuration, $p_{\text{x},j}, p_{\text{y},j} \in \mathbb{R}$ and θ_j are the positions and orientations at point s, respectively. Given N poses along the backbone, we can also define the inverse kinematic mapping $\varrho : 3N \times N \to n_\text{q}$ that provides us with the configuration $q = \varrho(\chi, s)$ in closed-form. Here, q_i will describe the configuration of the ith constant strain segment connecting the $i-1$th and the ith markers with associated poses χ_{i-1} and χ_i. This means that $n_\text{s} = N - 1$ in order that ρ can be bijective.

10.2.2 Lagrangian Dynamics

The Lagrangian of a mechanical system as a function of the configuration $q \in \mathbb{R}^{n_\text{q}}$ and the corresponding time derivative $\dot{q} \in \mathbb{R}^{n_\text{q}}$ can be expressed as

$$\mathcal{L}(q, \dot{q}) = \mathcal{T}(q, \dot{q}) - \mathcal{U}(q) = \underbrace{\frac{1}{2} \dot{q}^\top M(q)\, \dot{q}}_{\mathcal{T}(q,\dot{q})} - \underbrace{\frac{1}{2} q^\top K q - \int G(q)\, \mathrm{d}q}_{\mathcal{U}(q)}, \tag{10.2}$$

where $\mathcal{T}(q, \dot{q})$, $\mathcal{U}(q)$ are the kinetic and potential energy of the system, respectively, and $M(q) \succ 0 \in \mathbb{R}^{n_\text{q} \times n_\text{q}}$ is referred to as the mass matrix. $G(q) \in \mathbb{R}^{n_\text{q}}$ contribute the gravitational forces and $K \in \mathbb{R}^{n_\text{q} \times n_\text{q}}$ represents the linear elastic stiffness of the system. Subsequently, the Euler-Lagrangian equation can be leveraged to derive the EOM of continuum soft robots as (Della Santina et al., 2023; Liu et al., 2024a)

$$\frac{\partial^2 \mathcal{L}}{\partial \dot{q}^2} \ddot{q} + \frac{\partial^2 \mathcal{L}}{\partial q\, \partial \dot{q}} \dot{q} - \frac{\partial \mathcal{L}}{\partial q} + D\dot{q} = \tau \tag{10.3}$$

where we are also considering the generalized dissipative forces $D\dot{q}$ and the actuation torques $\tau \in \mathbb{R}^{n_\text{q}}$. In this work, we assume, without loss of generality, that both the stiffness matrix $K = \operatorname{diag}(k_1, \dots, k_{n_\text{q}})$ and the damping matrix $D = \operatorname{diag}(d_1, \dots, d_{n_\text{q}}) \succeq 0$ are diagonal.

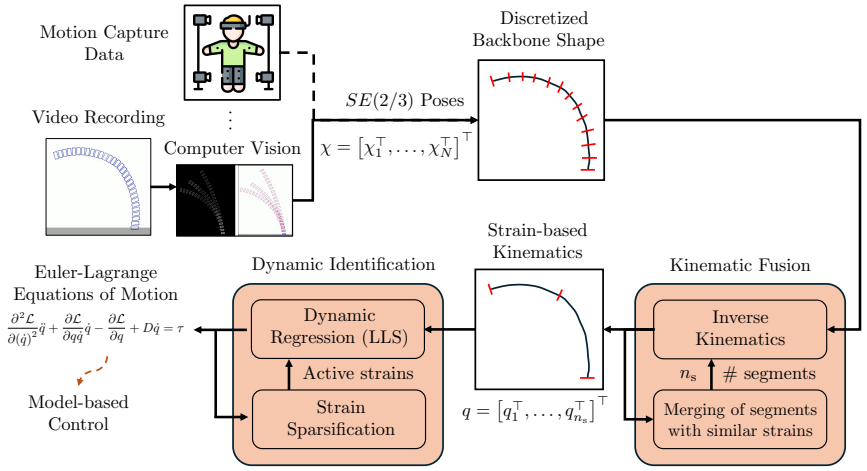


Figure 10.1: Overview of the proposed methodology with the key contributions (*Kinematic Fusion* and *Dynamic Regression and Strain Sparsification*) highlighted in orange. **Inputs:** We consider N Cartesian pose measurements χ distributed along the soft robot backbone, for example, obtained using Computer Vision (CV) techniques from video recordings, as inputs. **Kinematic Fusion:** We apply an iterative procedure that involves (i) computing the robot configuration q using PCS inverse kinematics, and (ii), to avoid overly complex and high-dimensional models, we merge adjacent segments with similar strains across the dataset into one segment of constant strain. **Dynamic Regression:** We identify the PCS dynamic model by iteratively regressing coefficients using linear least squares and further reduce the model complexity by neglecting insignificant strains. **Output:** The identified dynamic model has a Lagrangian structure suitable, for example, for model-based control applications.

10.3 Methodology

In this work, we propose a strategy for automatically identifying low-dimensional strain models for continuum robots directly from shape trajectories, as outlined in Figure 10.1. We assume that we have access to the poses of N markers along the backbone that represent a discretized shape description of the soft robot. In this work, we primarily focus on the planar case, where we extract SE(2) poses using computer vision techniques. However, the proposed *Kinematic Fusion* approach, along with the *Dynamic Regression and Strain Sparsification* strategy, can be extended to 3D scenarios with SE(3) inputs. While obtaining consistent and unoccluded SE(3) poses solely from vision-based information in 3D can be challenging, it is feasible (Zheng et al., 2024). Additionally, SE(3) pose measurements can always be acquired through other proprioceptive (Rosi et al., 2022) or exteroceptive methods, such as motion capture systems.

The goal is now to identify kinematic and dynamical models that allow (a) to represent the shape with n_s PCS segments (Renda et al., 2018), where necessarily the final $n_s \ll N$, and (b) to predict the future shape evolution of the soft robot. We tackle this task by (i) identifying a low-dimensional parametrization (e.g., number of constant strain segments, the length of each segment, etc.) of the kinematics over a series of static snapshots and (ii) identifying the parameters of the Lagrangian model and simultaneously eliminating strains from the model that do not have a significant effect on the shape evolution. We refer to component (i) as the *Kinematic Fusion* algorithm, as it is an iterative approach to

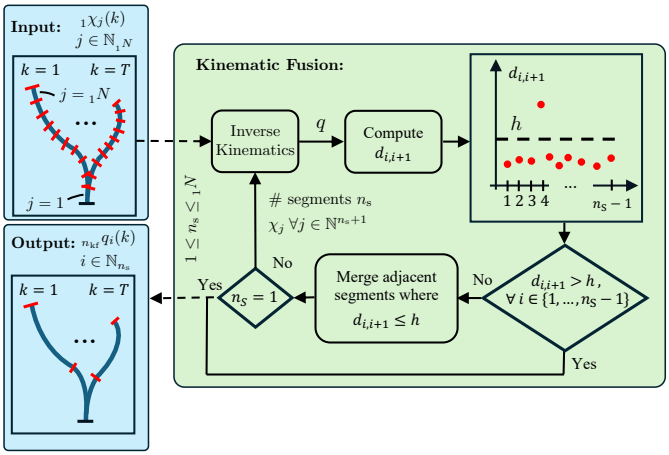


Figure 10.2: Schematic of the kinematic fusion algorithm. As inputs serve a sequence of N discrete poses along the backbone of the soft robot. Next, we execute inverse kinematics in closed form with a $n_s = N - 1$ segment PCS model to identify the (unmerged) configuration of the robot. Initially, each constant strain segment connects two neighboring backbone poses. Subsequently, we compute a strain similarity measure $\bar{d}_{i,i+1}$ between each pair of adjacent segments. If segments exhibit a similar strain (i.e., the metric falls below a threshold h), we merge them into one constant strain segment. This process is repeated until no more merging is possible, resulting in a kinematic model with (hopefully) fewer segments: $1 \leq n_s \leq N - 1$.

merge parts of the backbone that exhibit a similar strain into constant strain segments. The component (ii), named *Dynamic Regression & Strain Sparsification* algorithm, is an iterative procedure that, at each iteration, first regresses in closed-form the coefficients of the dynamic using linear least-squares and then eliminates strains from the dynamic model if the stiffness associated with a strain exceeds a given threshold. The intuition here is that strain would oscillate at very high frequencies, which are usually not relevant for practical control, and that it would take very high forces to introduce a significant deflection in the strain. The output of our approach is low-dimensional kinematic and dynamical models that preserve the physical & PCS strain model structures.

10.3.1 Kinematic Fusion

As previously introduced, the algorithm is provided for each training dataset sample $k \in \{1, \ldots, T\}$ with N pose measurements χ and associated backbone abscissas $s \in \mathbb{R}^N$ distributed along the backbone of the soft robot. Therefore, we initialize at $l = 1$: ${}_1N = N$ and ${}_1\chi = \chi$, where $l \in \mathbb{N}_{\geq 1}$ denotes the iteration index.

At the beginning of each iteration, we leverage the closed-form inverse kinematics to compute the configuration ${}_lq = \rho({}_l\chi, {}_ls)$ of a ${}_ln_s = {}_lN - 1$ segment PCS model. We now compute between each of the total ${}_ln_s - 1$ segment pairs, the following normalized strain similarity measure

$$ {}_l\bar{d}_{i,i+1} = \frac{1}{T} \sum_{k=1}^{T} \left\| \frac{{}_lq_{i+1}(k) - {}_lq_i(k)}{q^{\max} - q^{\min}} \right\|_2, \quad \forall i \in \{1, {}_ln_s - 1\}, \tag{10.4} $$

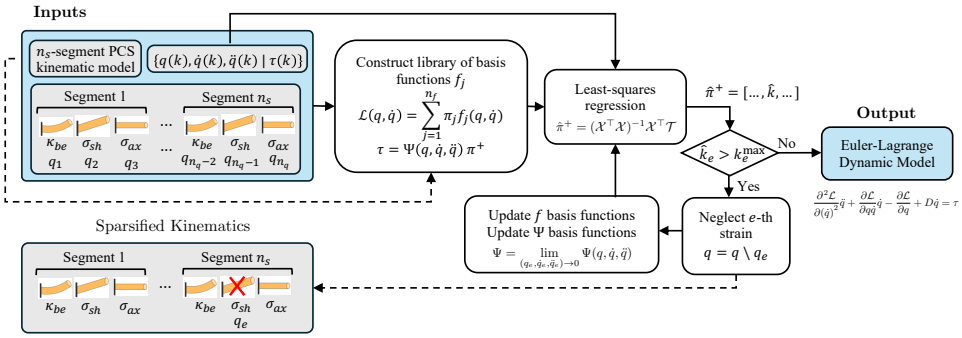


Figure 10.3: Schematic of the dynamic model identification process that simultaneously regresses the dynamic parameters and neglects unimportant strains. Based on a n_s-segment PCS model, a library of basis functions is constructed to parameterize the system's Lagrangian and EOM. A regression framework is established on a dataset of configuration-space positions $q(k)$, velocities $\dot{q}(k)$, accelerations $\ddot{q}(k)$, and actuation torques $\tau(k)$ that estimates the dynamic parameters $\hat{\pi}^+$ with closed-form, linear least squares. Strains that exhibit a stiffness higher than a predefined threshold are neglected, prompting adjustments to the basis functions. Subsequently, this procedure is repeated until all strain stiffnesses lie below the threshold.

where

$$q^{\min} = \min_{i\in\mathbb{N}_n, k\in\mathbb{N}_T} q_i(k), \qquad q^{\max} = \max_{i\in\mathbb{N}_n, k\in\mathbb{N}_T} q_i(k), \tag{10.5}$$

are the minimum and maximum configuration values across the dataset, respectively. This normalization is necessary as strains usually exhibit vastly different magnitudes. For example, the bending strains are usually more than one order of magnitude larger than the axial strains. We keep all segment pairs with ${}_l\bar{d}_{i,i+1} > h$, where h is a tunable threshold, separate. Oppositely, we merge all neighboring/adjacent segments with ${}_l\bar{d}_{i,i+1} \leq h$ into one single segment of constant strain. If indeed $\exists\, i \in \{\mathbb{N}_{{}_l n_s} \mid {}_l\bar{d}_{i,i+1} < h\}$, then then kinematic model is reduced to ${}_{l+1}n_s$ segments, where ${}_{l+1}n_s < {}_l n_s$. As a final step of the iteration, we now subsample the Cartesian poses ${}_{l+1}\chi$ and the associate backbone coordinates ${}_{l+1}s$ such that they only contain the tip of each of the fused segments.

The kinematic fusion step is repeated with $l = l+1$ for $n_{\text{kf}} - 1$ iterations until no more merging is possible, which can occur either if ${}_{n_{\text{kf}}}\bar{d}_{i,i+1} > h\ \forall i \in \{1, {}_{n_{\text{kf}}}n_s - 1\}$ (i.e., the strain similarity measure is larger than h for every pair of segments) or the model gets reduced to a single segment (i.e., ${}_{n_{\text{kf}}}n_s = 1$). We illustrate the kinematic fusion algorithm in Fig. 10.2, and an example of the thresholding is visualized in Fig. 10.4(a).

10

10.3.2 Dynamic Regression & Strain Sparsification

After obtaining a kinematic model with the *Kinematic Fusion* algorithm, we can employ the identified parametrization as a foundation for deriving a dynamic model. First, we symbolically derive the basis functions of the PCS dynamical model. Subsequently, we implement an iterative procedure to (i) regress dynamic coefficients with linear least squares and (ii) identify strains that can be neglected and remove them from the dynamical model.

Parametrization of the PCS Dynamic Model with Basis Functions

In order to easily regress the dynamic parameters with linear least squares, we first derive the PCS dynamics for a n_s segment soft robot from first principle (Armanini et al., 2023; Della Santina et al., 2023), with all rotational and linear strains taken into account, and subsequently parametrize both the Lagrangian and Euler-Lagrangian equations as a linear combination of monomial basis function.

$$\mathcal{L}(q,\dot{q}) = \sum_{j=1}^{n_\mathrm{f}} \pi_j f_j(q,\dot{q}), \quad \tau = \sum_{j=1}^{n_\Psi} \pi_j^+ \Psi_j(q,\dot{q},\ddot{q}) \in \mathbb{R}^{n_q}, \tag{10.6}$$

where $f_j : \mathbb{R}^{n_\mathrm{q}} \times \mathbb{R}^{n_\mathrm{q}} \to \mathbb{R}$ denotes each of the basis functions, and $\pi \in \mathbb{R}^{n_\mathrm{f}}$ the corresponding coefficients. Analog, we derive symbolically the EOM using an Euler-Lagrangian approach (see Section 10.2.2) and now state the corresponding basis functions as $\Psi(q,\dot{q},\ddot{q}) \in \mathbb{R}^{n_\Psi \times n_\mathrm{q}}$ with $\Psi_j : \mathbb{R}^{n_\mathrm{q}} \times \mathbb{R}^{n_\mathrm{q}} \times \mathbb{R}^{n_\mathrm{q}} \to \mathbb{R}^{n_\mathrm{q}}$ such that

$$\tau = \sum_{j=1}^{n_f} \left[\pi_j \left(\frac{\partial^2 f_j}{\partial \dot{q}^2} \ddot{q} + \frac{\partial^2 f_j}{\partial q \partial \dot{q}} \dot{q} - \frac{\partial f_j}{\partial q} \right) \right] + D\dot{q}, \tag{10.7}$$

where $\pi^+ = \begin{bmatrix} \pi^\top & d^\top \end{bmatrix}^\top \in \mathbb{R}^{n_\Psi}$ contains the associated coefficients and consists of π and the damping coefficients $d = \mathrm{diag}(D) \in \mathbb{R}^{n_\mathrm{q}}$.

Regression of Dynamic Parameters

In order to estimate the dynamic coefficients, we formulate the linear regression problem as $\mathcal{T} = X\pi^+$, which accommodates the dataset of positions, velocities, and accelerations $\mathcal{X} = [\Psi(q(1),\dot{q}(1),\ddot{q}(1))^\top, ..., \Psi(q(T),\dot{q}(T),\ddot{q}(T))^\top]^\top \in \mathbb{R}^{Tn_\mathrm{q} \times n_\Psi}$ and the corresponding actuation inputs $\mathcal{T} = [\tau^\top(1), \dots, \tau^\top(T)] \in \mathbb{R}^{Tn_\mathrm{q}}$. We solve this optimization problem with linear least squares, which minimizes the residual error as $\min \|\mathcal{T} - \mathcal{X}\pi^+\|_2^2$ and allows us to identify the dynamic model coefficients in closed form as

$$\hat{\pi}^+ = (\mathcal{X}^\top \mathcal{X})^{-1} \mathcal{X}^\top \mathcal{T}. \tag{10.8}$$

Sparsification of Strains

This dynamic identification method offers the advantage of having interpretable results, as each estimated coefficient has some physical meaning within the PCS dynamic derivation. Specifically, among those we can extract the estimated stiffness matrix $\hat{K} = \mathrm{diag}(\hat{k}_1, \dots, \hat{k}_{n_\mathrm{q}})$, allowing us to assess the importance of each strain through its stiffness magnitude $\hat{k}_e$. A strain with high stiffness usually exhibits low displacement, approximating rigid behavior. Therefore, such strain can be considered non-essential and neglected in the dynamics. We define a maximum allowable stiffness $k_e^{\max} \in \mathbb{R}_{\geq 0}$ for each strain/configuration variable as a function of the maximum Elastic and Shear moduli $E^{\max}, G^{\max} \in \mathbb{R}_{\geq 0}$. For example, in the planar case and for constant cross sections of area A_c and second moment of inertia I_c, this can be conveniently done as

$$k_{\mathrm{be}}^{\max} = I_c E^{\max}, \quad k_{\mathrm{sh}}^{\max} = A_c G^{\max}, \quad k_{\mathrm{ax}}^{\max} = A_c E^{\max}. \tag{10.9}$$

Given these maximum stiffnesses, the e-th strain is neglected if its estimated stiffness lies above the threshold $\hat{k}_e > k_e^{\max}$. Therefore, the e-th strain is removed as a configuration variable through

$$q' = \left[q_1, q_2, \dots, q_{e-1}, q_{e+1}, \dots, q_{n_q})\right]^\top, \tag{10.10}$$

and its influence on the dynamics needs to be eliminated as well. We update the Euler-Lagrange basis functions as $\Psi = \lim_{(q_e, \dot{q}_e, \ddot{q}_e) \to 0} \Psi(q, \dot{q}, \ddot{q})$. Any columns that turn into all-*zeros* are also removed, and the coefficient vector π^+ is updated accordingly by removing the corresponding rows. A similar procedure applies to the Lagrangian basis functions f and their coefficients π.

This procedure of regressing dynamic coefficients and sparsifying strains, as presented in Sections 10.3.2 and 10.3.2, respectively, is the number of strains/configuration variables converges (i.e., remains constant). We stress that the (likely) computationally expensive operation of symbolically deriving the library of basis functions only needs to be done once at the beginning, as we subsequently update the library by taking the limit at the end of each iteration.

10.4 Validation

To validate the proposed approach, we verify the kinematic and dynamic regression algorithms separately. We test the kinematic fusion algorithm on simulated continuum soft robots that behave according to the PCS and PAC model. Subsequently, we compare the dynamic prediction performance of the proposed method against multiple ML baseline methods on various PCS soft robots. Finally, we demonstrate how the regressed methods can be used in a plug-and-play fashion within a closed-form, model-based setpoint regulation framework.

10.4.1 Experimental Setup

Evaluation Cases

Cases 1-3 (1-3S PCS) These cases consider one-segment, two-segment, and three-segment planar PCS soft robots ($n_s \in \{1, 2, 3\}$), respectively, with configurations $q \in \mathbb{R}^{n_q}$ where $n_q \in \{3, 6, 9\}$ and actuation $\tau \in \mathbb{R}^3, \mathbb{R}^6, \mathbb{R}^9$, assuming full actuation.

Case 4 (1S PCS H-SH) This case is a one-segment PCS robot with high shear stiffness, three orders of magnitude larger than in *Case 1.*

Case 5 (2S PCS H-AX/SH) This case is a two-segment PCS robot, where the 1st segment has a significantly increased axial stiffness and 2nd segment an increased shear stiffness w.r.t *Case 2.*

Case 6 (1S PAC) This case considers a one-segment PAC robot whose curvature can be described by an affine function (Stella et al., 2023c).

Dataset Generation

In order to illustrate the end-to-end nature of our proposed method, we generate the datasets as short video sequences of the soft robot's movement. Therefore, we mimic a

camera placed parallel to the robot's plane of motion to capture the soft robot's ground-truth dynamics. At each time step, we render an image of the soft robot that contains $N = 21$ equally distant, visually salient features. In the real world, this could be achieved by attaching markers to the soft robot that allow tracking of points along the backbone across time (Stella et al., 2022). We simulate the robot's ground-truth dynamics using the planar PCS simulator presented in (Stölzle et al., 2024b).

For Cases 1 to 4, we include eight trajectories with randomly sampled initial conditions in the training set. We consider stepwise actuation sequences for which we randomly sample a torque every 10 ms. Each trajectory produces a 0.5 s video captured at 1000 Hz. For *Case 6*, since the PAC simulator only accounts for kinematics, we generate an image sequence featuring the robot in 500 randomly selected configurations. As the test set, an additional trajectory with 7 s duration is generated by applying a sinusoidal actuation sequence with $\tau = a_1 \sin(\omega_1 t) + a_2 \cos(\omega_2 t) \in \mathbb{R}^{n_q}$, where a_1 and a_2 are random amplitudes, and ω_1 and ω_2 are random frequencies.

Backbone Shape Detection from Images

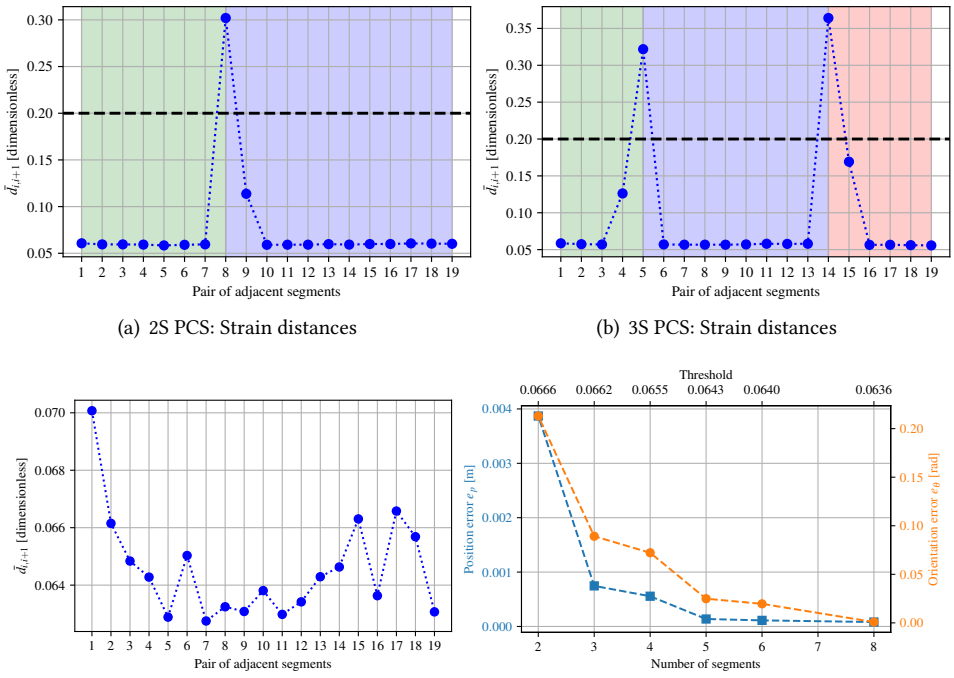
To apply our proposed model identification method, we first extract the motion of several Cartesian-space samples along the robot's backbone. As we consider a planar problem setting and rendered images of the soft robot's shape, the goal is to extract the SE(2) poses of N cross-sections along the robot. To satisfy the assumption behind the *Kinematic Fusion* algorithm, the number of extracted poses N should be significantly larger than the expected number of PCS segments required to model the robot's behavior accurately: $N \gg n_s$. We leverage the *OpenCV* library for detecting the soft robot contour (`findContours`) and extracting pose measurements along its backbone (`minAreaRect`). Each frame is binarized, and the contours of cross-sections are identified. This allows the extraction of the center position $(p_{x,j}, p_{y,j})$ and orientation θ_j of each cross-section (also referred to as *marker*). As in our case, the markers are equally distant, we compute, without loss of generality, the backbone abscissa as $s_j = \frac{j-1}{N} L$. For T video frames, this results in a time sequence of SE(2) poses $\{\chi(1), ..., \chi(T)\}$, $\chi_j = \left[\chi_1^\top \dots \chi_N^\top\right]^\top \in \mathbb{R}^{3N}$, and $j \in \{1, \dots, N\}$. We leverage a Savitzky-Golay filter (3^{rd}-order, window length 25) to estimate the associated pose velocities $\dot{\chi}$ and velocities $\ddot{\chi}$.

Evaluation metrics

To evaluate the models quantitatively, we introduce position and orientation task space metrics. We use a Cartesian-space Mean Absolute Error (MAE) measuring the deviation of the estimated from the actual robot body shape, given by

$$e_{\text{p}}^{\text{body}} = \sum_{k=1}^{T} \sum_{j=1}^{N} \frac{\|\hat{p}_j(k) - p_j(k)\|_2}{NT}, \quad e_{\theta}^{\text{body}} = \sum_{k=1}^{T} \sum_{j=1}^{N} \frac{|\hat{\theta}_j(k) - \theta_j(k)|}{NT}, \tag{10.11}$$

where $\hat{p}_i(k)$ and $\hat{\theta}_i(k) \in \mathbb{R}$ are the estimated position and orientation of point i along the structure, respectively, while $p_i(k)$ and $\theta_i(k)$ are the ground-truth counterparts. These metrics give the average pose error across all T frames of a trajectory and all N cross-sections tracked along the robot, enabling a good evaluation of the kinematic model by capturing how well it represents the overall shape of the soft robot structure.



(a) 2S PCS: Strain distances

(b) 3S PCS: Strain distances

(c) 1S PAC: Strain distances

(d) 1S PAC: Kinematic Pareto Front

Figure 10.4: Kinematic Fusion Results: **Panels (a) & (b):** Average strain distances between pairs of adjacent segments for *Cases 2 & 3*. The poses of 20 markers along the manipulators are tracked, resulting in 19 pairs of segments to be evaluated for strain similarity. The threshold is represented by a dashed line, and the background shading marks the resulting segments (separate segments are shaded in different colors). **Panel (c):** Average strain distances between segments after the first iteration of the kinematic fusion algorithm for *Case 6*. **Panel (d):** Pareto front that describes the trade-off between the DOF of the kinematic model (i.e., the number of segments) and the shape reconstruction error for *Case 6*. The blue and orange lines represent the average kinematic body position error $e_{\mathrm{p}}^{\mathrm{body}}$ and the body orientation error $e_{\theta}^{\mathrm{body}}$, respectively. The strain distance threshold h that is used for separating segments is plotted on the upper x-axis.

In addition, we consider an end-effector Cartesian-space MAE given by

$$e_p^{\mathrm{ee}} = \frac{1}{T}\sum_{k=1}^{T}\|\hat{p}_{\mathrm{ee}}(k) - p_{\mathrm{ee}}(k)\|_2, \quad e_\theta^{\mathrm{ee}} = \frac{1}{T}\sum_{k=1}^{T}|\hat{\theta}_{\mathrm{ee}}(k) - \theta_{\mathrm{ee}}(k)|, \tag{10.12}$$

where $\hat{p}_{\mathrm{ee}}(k) = \hat{p}_N(k) \in \mathbb{R}^2$ and $\hat{\theta}_{\mathrm{ee}}(k) \in \mathbb{R}$ are the estimated end-effector position and orientation, respectively, with $p_{\mathrm{ee}}(k)$ and $\theta_{\mathrm{ee}}(k)$ being the ground-truth counterparts. This metric is particularly useful for assessing the obtained dynamic models with a control perspective, as the accuracy of the end-effector predictions is crucial for effective control in task space.

10.4.2 Kinematic Fusion Results

Cases 1, 2, and 3

Figs. 10.4(a) & 10.4(b) presents the average strain distances between the adjacent segment pairs for *Cases 2-3*, as the result of the first and final iteration of the kinematic fusion algorithm. The plots for Cases 2 and 3 reveal one and two peaks, respectively, revealing where we need to separate segments in the kinematic model. The strain distance threshold h is a hyperparameter that trades off model complexity with model accuracy. For common soft robots that exhibit a PCS-like behavior, we recommend choosing h such as that the segmented model exhibits isolated peaks in the strain distance metric, as visible in Fig. 10.4. The number of segments and respective lengths for the resulting models obtained with a threshold of $h = 0.2$ are presented in the third column of Table 10.1, and by comparing to the second column, it is easily visible that our algorithm almost perfectly identified the segment lengths. As a consequence of the correct identification of the number of segments and segment lengths, the identified kinematic model also accurately captures the shape of the robot with the position errors below 0.2 % of the robots' length, as it can be seen from the body shape reconstruction errors stated in the first three rows of Table 10.1. We remark that the error can be further reduced by using a finer discretization of backbone pose markers (i.e., increasing N).

Table 10.1: Kinematic fusion results: The second and third columns contain the actual and estimated segment lengths, respectively. The Cartesian pose error between the actual and estimated backbone shape is stated in the third and fourth columns, respectively. *Case 6* represents a one-segment PAC soft robot of total length 150 mm and can be, therefore, not be represented with a (one-segment) PCS model.

Case	Actual segment lengths $\mathbf{L}$ [mm]	Estimated segment lengths $\hat{\mathbf{L}}$ [mm]	$\mathbf{e}_{\mathrm{p}}^{\mathrm{body}}$ [mm]	$\mathbf{e}_{\theta}^{\mathrm{body}}$ [rad]
1: 1S PCS	$[100]$	$[100]$	0.082	6.38×10^{-3}
2: 2S PCS	$[70, 100]$	$[68, 102]$	0.240	1.15×10^{-2}
3: 3S PCS	$[50, 100, 60]$	$[52.5, 94.5, 63.0]$	0.210	9.67×10^{-3}
6: 1S PAC	$[150]$	$[7.5, 105.0, 37.5]$	0.746	8.92×10^{-2}

Case 6

Even though the kinematics of many continuum soft robotic manipulators can be described by PCS/PCC kinematics, other continuum soft robots can only be described by piecewise constant models in the limit $N \to \infty$ as they exhibit polynomial curvature (Della Santina and Rus, 2019; Stella et al., 2022) or even more generally GVS (Boyer et al., 2020). This is particularly the case when external or gravitational forces dominate the elastic and actuation forces (Della Santina et al., 2023). In order to verify that our approach is also able to identify effective models in such situations, we test the kinematic fusion algorithm on the case of an affine curvature robot (Stella et al., 2023c) and plot the resulting average strain distances in Fig. 10.4(c). Indeed, the strain distance plot no longer exhibits clear, isolated peaks (i.e., a single solution). Therefore, we formulate a Pareto front in Fig. 10.4(d) (by varying the strain distance threshold h) that describes the tradeoff between the number of PCS segments (i.e., the DOF of the kinematic model) and the shape reconstruction accuracy. Analyzing and exploiting this tradeoff allows the user to choose their *sweetspot* between model complexity and performance. In this case, we find that three segments represent a

suitable compromise between model complexity and shape reconstruction accuracy, as it exhibits a position error of only 0.5 % of the robot's length.

10.4.3 Dynamic Model Identification Results

Verification of Dynamical Regression

We verify the dynamical regression algorithm on one and two-segment PCS robots (i.e., Cases 1 & 2), both with and without measurement noise. After regressing the dynamic parameters on the training set, we perform a rollout on the test set and compare the resulting predicted trajectory with the ground truth. To confirm that the dynamic regression is also effective when applied to real-world data, we apply in some experiments Gaussian noise that mirrors measurement noise as we would encounter it for motion capture data, computer vision detection errors, etc. to the poses included in the training set (i.e., $\tilde{\chi} = \chi + \mathcal{N}(0, \sigma_{\mathrm{n}})$). For *Case 1*, we sample the noise from a normal distribution with standard deviations 0.5 mm and 1° for the position and orientation measurements, respectively. Analog, we define the standard deviation of the noise for *Case 2* as 0.1 mm and 0.5°.

The results are reported in Tab. 10.2 (top four rows), and the rollouts of Cases 1 & 2 are included in Figs. 10.5-10.6. We also present a sequence of stills of the rollout of *Case 2* in Fig. 10.7. We conclude that even though the dynamical parameters are regressed on only 4 s of robot motion data, the dynamical predictions are extremely accurate on the long horizon of 7 s (most control algorithms such as MPC operate on a much smaller horizon). The position error for the experiments not involving noise stays below 5 % in both cases. When noise is present in the training data, the position error is roughly tripled. Still, we observe that the error is mostly related to the transient terms and the model converges during the slower sequences of the trajectory to the ground truth. Most importantly, the learned model, even when trained on noisy data, remains stable, as shown in Fig. 10.6.

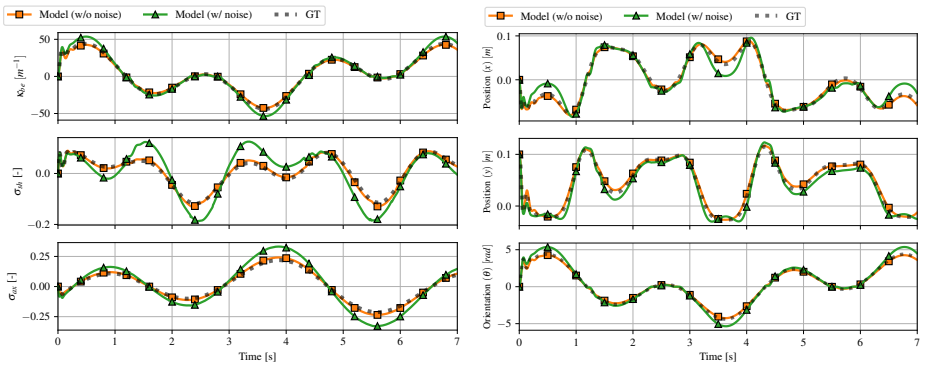
Verification of Strain Sparsification

Next, we verify that the strain sparsification algorithm can detect and eliminate strains that do not have a significant effect on the dynamics and can be, therefore, neglected to reduce the model complexity. For this purpose, we apply the integrated *Dynamic Regression and Strain Sparsification* algorithm to *Cases 4 & 5*, which exhibit no shear strain and no axial strain (1st-segment) & no shear strain (2nd-segment), respectively. We define the maximum elastic and shear modulus as $E^{\max} = 100\,\mathrm{MPa}$ which leads to the stiffness thresholds for each segment $K_j^{\max} = \mathrm{diag}(12.6\,\mathrm{Nm}^2, 168\,\mathrm{N}, 126\,\mathrm{kN})$. For example, in *Case 4*, after determining the dynamic parameters during the first iteration, the algorithm detects that the estimated shear stiffness $\hat{K}_{\mathrm{sh}} = 1200\,\mathrm{N} > K_{\mathrm{sh}}^{\max} = 168\,\mathrm{N}$. Therefore, the shear strain is eliminated from the dynamic model, and the dynamic parameters are newly regressed during the next iteration. Similarly, the algorithm correctly neglects the axial strain for the 1st segment and the shear strain for the 2nd segment of *Case 5*. We visualize the test set rollouts for *Cases 4 & 5* in Fig. 10.8 and report the error metrics in the last three rows of Tab. 10.2. The results show that in *Case 4*, the model without shear even exhibits a slightly smaller position error than the model that includes all strains. A possible explanation could be that with the Lagrangian being parametrized by fewer basis functions, the coefficients for the remaining strains can be more accurately regressed. If we compare *Case 2* and *Case 5* (both two-segment PCS robots), *Case 5* exhibits significantly increased position and

orientation errors. Still, we notice that the model predictions are usable, in particular for shorter horizons.

Table 10.2: Dynamic regression results: End-effector position and orientation errors $\mathbf{e}_{\mathrm{p}}^{\mathrm{ee}}, \mathbf{e}_{\theta}^{\mathrm{ee}}$ for the obtained dynamic models evaluated on the 7 s sinusoidal test set trajectory. In some cases, we add artificial measurement noise to the training data. For *Case 4*, we present two variants of the learned model: in the first instance, we report the performance of a model that neglects strains as suggested by the dynamic sparsification algorithm. For completeness, we furthermore also state the performance of a model that considers all strains. In *Case 5*, the 1st segment only exhibits bending and shear strains (i.e, no axial strain), and the 2nd segment only exhibits bending and axial strains (i.e., no shear strain)

Case	Meas. Noise	Model Strains	$\mathbf{e}_{\mathrm{p}}^{\mathrm{ee}}$ [mm]	$\mathbf{e}_{\theta}^{\mathrm{ee}}$ [rad]
1: 1S PCS	✗	All	4.89	0.113
1: 1S PCS	✓	All	13.7	0.307
2: 2S PCS	✗	All	5.22	0.138
2: 2S PCS	✓	All	16.8	0.135
4: 1S PCS H-SH	✗	No shear	4.57	0.099
4: 1S PCS H-SH	✗	All	5.14	0.116
5: 2S PCS H-AX/SH	✗	(✓, ✓, ✗, ✓, ✗, ✓)	17.9	0.305



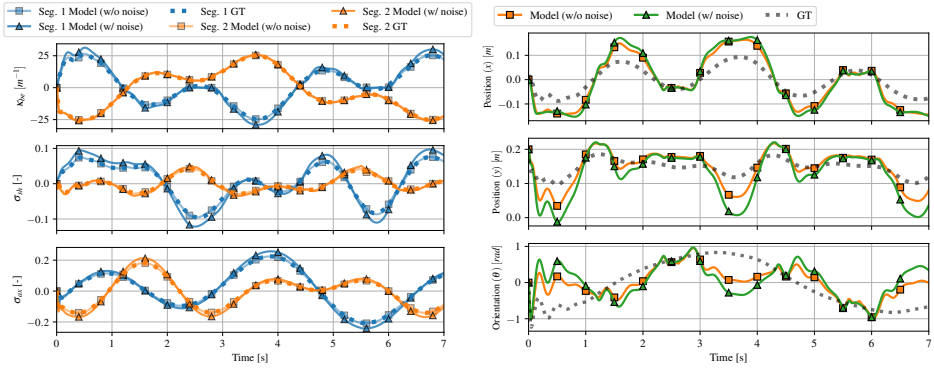
(a) Configurations (b) End-effector pose

Figure 10.5: Verification of the model obtained for Case 1 on a sinusoidal trajectory. The dotted line denotes the ground-truth (GT) trajectory, while the green and orange lines correspond to the models trained with and without noise, respectively.

10

10.4.4 Benchmarking of Identified Dynamics against ML Baselines

We benchmark the derived dynamical model of *Case 2* (i.e., a two-segment planar PCS soft robot) against several models trained using machine learning approaches. Specifically, we consider various learning-based approaches that range from completely data-driven (e.g., NODE) over approaches that take into account the structure of Lagrangian systems

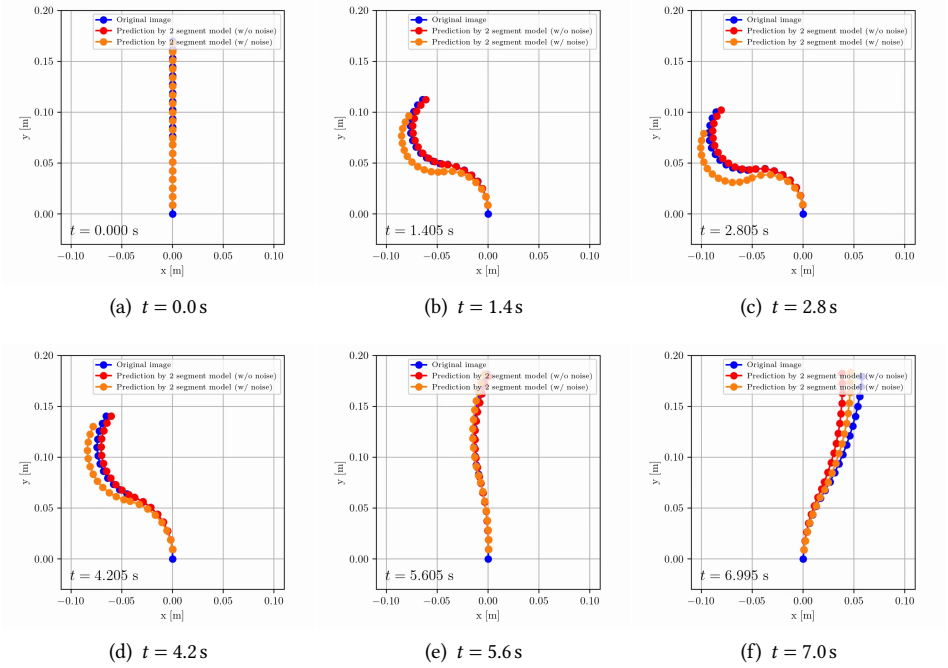


(a) Configurations (b) End-effector pose

Figure 10.6: Verification of the dynamical model with noise for a two-segment PCS soft robot (*Case 2*). The dotted lines denote the ground-truth (GT) trajectory. For the configuration plot, blue lines refer to the first segment, while orange lines are associated with the second segment. For the end-effector pose plot, the orange and green lines mark the models trained with and without noise, respectively.

Table 10.3: Results for learning dynamics of a two-segment PCS soft robot (Case 2). We report the position ($e_{\mathrm{p}}^{\mathrm{body}}$) and orientation ($e_{\theta}^{\mathrm{body}}$) metrics that capture the mean shape error averaged over all time steps. The shape error is computed by considering the dynamical evolution of three marker poses with $s_j \in \{68, 119, 170\}$ mm. We report the metrics for a rollout of the dynamics on both a 0.5 s sequence on the training set and the entire (i.e., 7 s) test set.

Method	**Train** $\mathbf{e}_{\mathrm{p}}^{\mathrm{body}}$	**Train** $\mathbf{e}_{\theta}^{\mathrm{body}}$	**Test** $\mathbf{e}_{\mathrm{p}}^{\mathrm{body}}$	**Test** $\mathbf{e}_{\theta}^{\mathrm{body}}$
NODE	10.4 mm	0.27 rad	245.6 mm	12.22 rad
LNN (Liu et al., 2024a)	550.8 mm	5.16 rad	∞	∞
CON (Stölzle and Della Santina, 2024)	12.9 mm	0.24 rad	789.5 mm	29.49 rad
PCS Regression (Ours)	**3.1** mm	**0.04** rad	**9.8** mm	**0.13** rad



(a) $t = 0.0\,\mathrm{s}$ (b) $t = 1.4\,\mathrm{s}$ (c) $t = 2.8\,\mathrm{s}$

(d) $t = 4.2\,\mathrm{s}$ (e) $t = 5.6\,\mathrm{s}$ (f) $t = 7.0\,\mathrm{s}$

Figure 10.7: Sequence of stills for the test rollout of the regressed dynamic model trained on the two-segment PCS dataset (*Case 2*). The blue and red dots represent the ground truth and estimated shape of the soft robot, respectively. The orange line represents the shape estimated by a model trained on a training set with added measurement noise.

(e.g., LNN, CON). To keep the comparison fair, we define the inputs for all methods as the Cartesian poses $\chi_j \in \mathbb{R}^3$ and the corresponding time derivative $\dot{\chi}_j$ of N discrete *markers* along the backbone. Furthermore, we also provide the actuation torques $\tau \in \mathbb{R}^6$ as a dynamic model input. Therefore, the total model input exhibits a dimensionality of $\mathbb{R}^{6N+6}$. The task of the dynamic model is to predict the acceleration $\ddot{\chi}(k) \in \mathbb{R}^{3N}$. We tried supplying all 21 markers, which our method also has access to, to the baseline approaches. However, this proved to be infeasible as the problem would become too high-dimensional in terms of the number of inputs and outputs, and the baseline approaches would overfit the training set. Therefore, we settled to give the baseline methods access to the pose measurements of $N = 3$ markers distributed along the backbone of the robot at $s_j \in \{68, 119, 170\}$ mm, where $s = 170\,\mathrm{mm}$ corresponds to the end-effector.

Implementation of Baseline Methods The NODE is parametrized by a six-layer MLP with hidden dimension 256 and hyperbolic tangent activation function that predicts based on the input $(\chi(t), \dot{\chi}(t), \tau(t))$ the acceleration $\ddot{\chi}$. We remark that with this strategy, we already infuse the prior knowledge that the time derivative of the pose is the velocity, which would not be the case in a naive implementation of a NODE. CONs (Stölzle and Della Santina, 2024) allow for learning of (latent) dynamics of Lagrangian systems with strong stability

guarantees (global asymptotic stability / input-to-state stability) by leveraging a network of damped harmonic oscillators that are coupled by a hyperbolic potential. In order to allow for arbitrary placement of the global asymptotically stable equilibrium point, we learn a linear coordinate transformation into the latent coordinates $z = W\chi + b \in \mathbb{R}^9$, $\dot{z} = W\dot{\chi}$. LNNs learn the components of the Lagrangian $\mathcal{L}(\chi, \dot{\chi}) = \frac{1}{2}\dot{\chi}^\top M(\chi)\dot{\chi} - \mathcal{U}(\chi)$ such as the mass matrix $M(\chi) \succ 0 \in \mathbb{R}^{9\times9}$, the potential energy $\mathcal{U}(\chi) \in \mathbb{R}$, the damping matrix $D \succeq 0 \in \mathbb{R}^{9\times9}$ and the actuation matrix $A \in \mathbb{R}^{6\times9}$ and subsequently derive the EOM as $M(\chi)\ddot{\chi} + \frac{\partial \mathcal{L}}{\partial \chi \partial \dot{\chi}}\dot{\chi} + \frac{\partial \mathcal{U}}{\partial \chi} + D\dot{\chi} = A\tau$ using autodifferentiation. We regard A and D as trainable weights and parametrize $M(\chi)$ and $\mathcal{U}(\chi)$ with six-layer MLPs with hidden dimension 256 (softplus activation). We leverage the Cholesky decomposition to make sure that $M(\chi), D \succ 0$.

Training The first loss term is a MSE between the predicted $\hat{\ddot{\chi}}(k)$ and actual acceleration $\ddot{\chi}(k)$. Additionally, we roll out the trajectories over a horizon of 0.3 s and add loss terms that compute the MSE error between the predicted states $(\hat{\chi}(k+r), \hat{\dot{\chi}}(k+r))$ and the labels $(\chi(k+r), \dot{\chi}(k+r))$, where r is the index of the rollout step. As training LNNs is computationally very demanding due to the need to differentiate w.r.t. both inputs and neural network parameters, we had to reduce the training rollout horizon.

Results We present the benchmarking results in Fig. 10.9 (evaluation on the training set) and Fig. 10.10 (evaluation on the test set). Quantitative error metrics are provided in Table 10.3. We report the performance for rollouts on both the training and the test set. All methods, except for LNN, are able to predict the Cartesian-space evolution of the markers attached to the soft robot backbone decently accurately over the training set trajectory. Still, the our proposed method exhibits an 70 % to 80 % lower error than NODE and CON (Stölzle and Della Santina, 2024). As LNN does not exhibit any stability guarantees and it is trained on a relatively short horizon, it diverges from the ground-truth trajectory after roughly 0.15 s. As shown in panels (d) of Figs. 10.9 and 10.10, the axial actuation forces on the validation set are one order of magnitude higher than in the training set. Therefore, these axial forces can be considered to be out-of-distribution for the trained models. Our proposed method is amazingly able to still exhibit very good performance, while all baseline methods are no longer able to predict the dynamic evolution of the system. LNN even becomes fully unstable after a few milliseconds, and we are, therefore, not able to report test errors for this method.

10.4.5 Demonstration of Model-based Control

10

To demonstrate how the derived models can be used in a plug-and-play fashion for model-based control, we simulate the closed-loop dynamics of a simulated two-segment PCS soft robot (*Case 2*) with configuration $q \in \mathbb{R}^7$ under a P-satI-D+FF (Della Santina et al., 2023; Stölzle and Della Santina, 2024; Stölzle et al., 2024b) setpoint control policy

$$\tau(t,q) = \underbrace{\hat{G}(q^{\mathrm{d}}) + \hat{K}q^{\mathrm{d}}}_{\text{Learned feedforward term}} + \underbrace{K_{\mathrm{p}}(q^{\mathrm{d}} - q) - K_{\mathrm{d}}\dot{q} + K_{\mathrm{i}}\, e_{\mathrm{int}}(t)}_{\text{P-satI-D feedback term (Pustina et al., 2022a)}}, \tag{10.13}$$

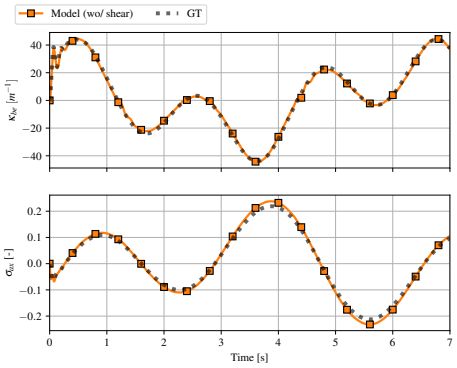
where $K_{\mathrm{p}}, K_{\mathrm{i}}, K_{\mathrm{d}} \in \mathbb{R}^{n\times n}$ are the proportional, integral, and derivative control gains, respectively. The feedback control term is a PID-like controller with integral saturation (Pustina

et al., 2022a) and the dimensionless gain $\mathrm{Y} \in \mathbb{R}$ which bounds the integral error at each time step to the interval $(-1, 1)$ and reduces the risk of instability for nonlinear systems

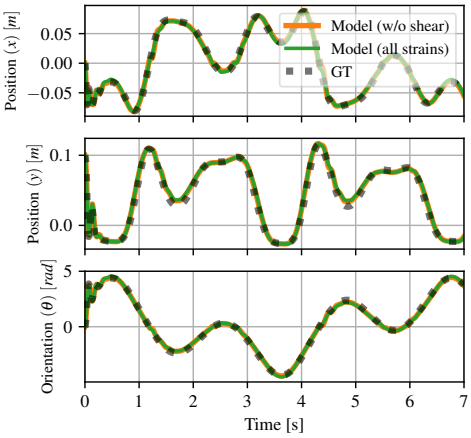
$$e_{\mathrm{int}}(t, q) = \int_0^t \tanh(\mathrm{Y}(q^{\mathrm{d}}(t') - q(t'))) \, \mathrm{d}t', \tag{10.14}$$

$\hat{G}(q) \in \mathbb{R}^6$ and $\hat{K} \in \mathbb{R}^{6\times6}$ are the estimated gravitational forces and stiffness matrix, respectively. We simulate the closed-loop dynamics with a Tsitouras 5(4) integrator at a timestep of 0.05 ms. Please note that we use the ground-truth dynamics as a state transition function.

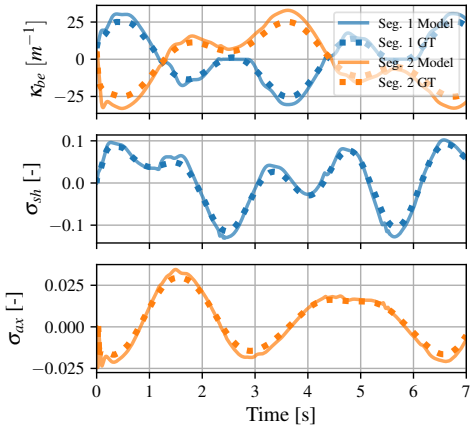
To verify that the learned model performs well within the model-based control policy, we create a sequence of 7 randomly sampled setpoints $q^{\mathrm{d}}(k) \in \mathbb{R}^6$. The results in Fig. 10.11 show that the controller is able to effectively regulate a two-segment planar soft robot. The tracking of the bending strain reference is perfect. For the linear strains, we notice small errors in the feedforward term, but the integral control is able to compensate for them and drive the system toward the reference. We stress that the structure and characteristics of the learned model enabled us to formulate the control policy in closed form easily, and we did not have to resort to techniques such as MPC or RL as it would be necessary for other model learning techniques (e.g., LSTMs, NODEs).



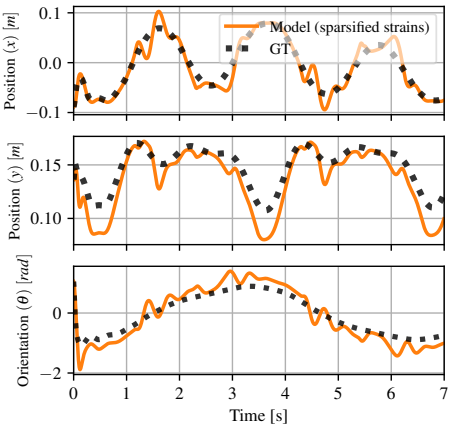
(a) Case 4: Configurations



(b) Case 4: End-effector pose



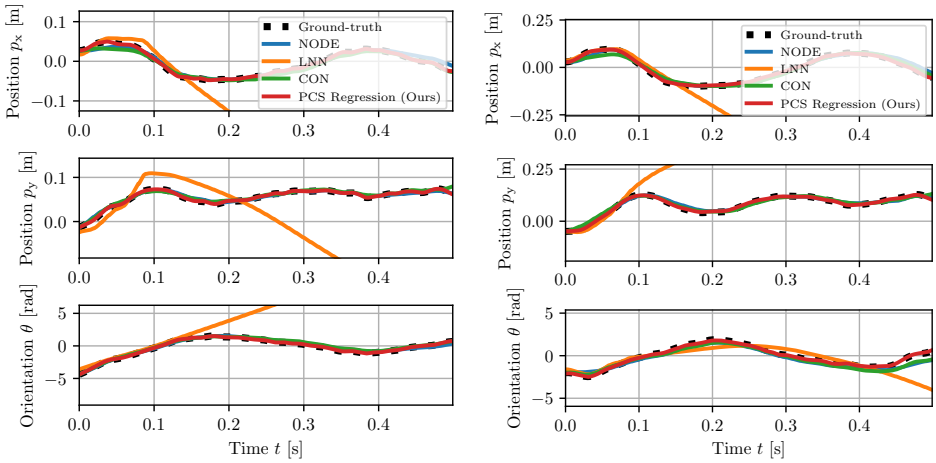
(c) Case 5: Configurations



(d) Case 5: End-effector pose

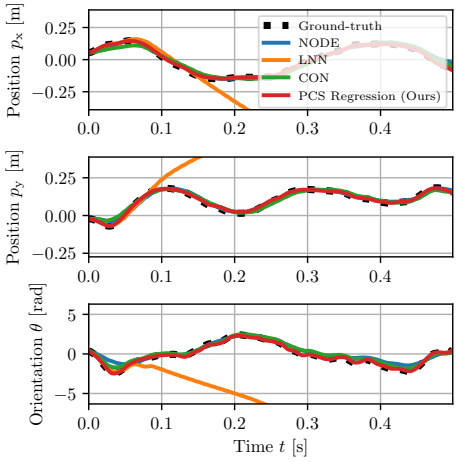
10

Figure 10.8: Verification of the strain sparsification algorithm on a one-segment PCS robot with shear modulus (*Case 4*) and on a two-segment PCS robot where the 1st segment exhibits high axial stiffness and the second segment high shear stiffness. We roll out both the ground truth (dotted line) and the learned (solid line) model dynamics from the same initial condition and for a given sinusoidal actuation sequence. Verification of the model obtained for Case 4 on a sinusoidal trajectory. The dotted line denotes the ground-truth trajectory, while the green and orange lines correspond to the models trained with and without noise, respectively.

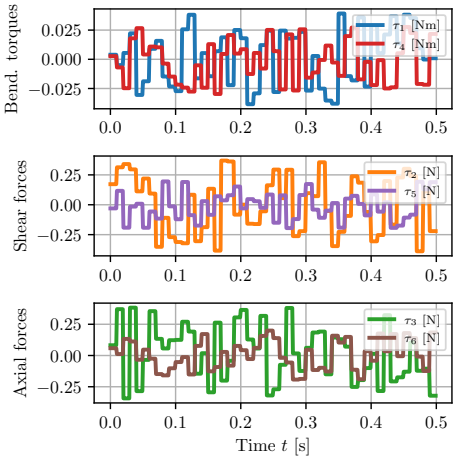


(a) Train: Marker $s = 68\,\text{mm}$

(b) Train: Marker $s = 119\,\text{mm}$

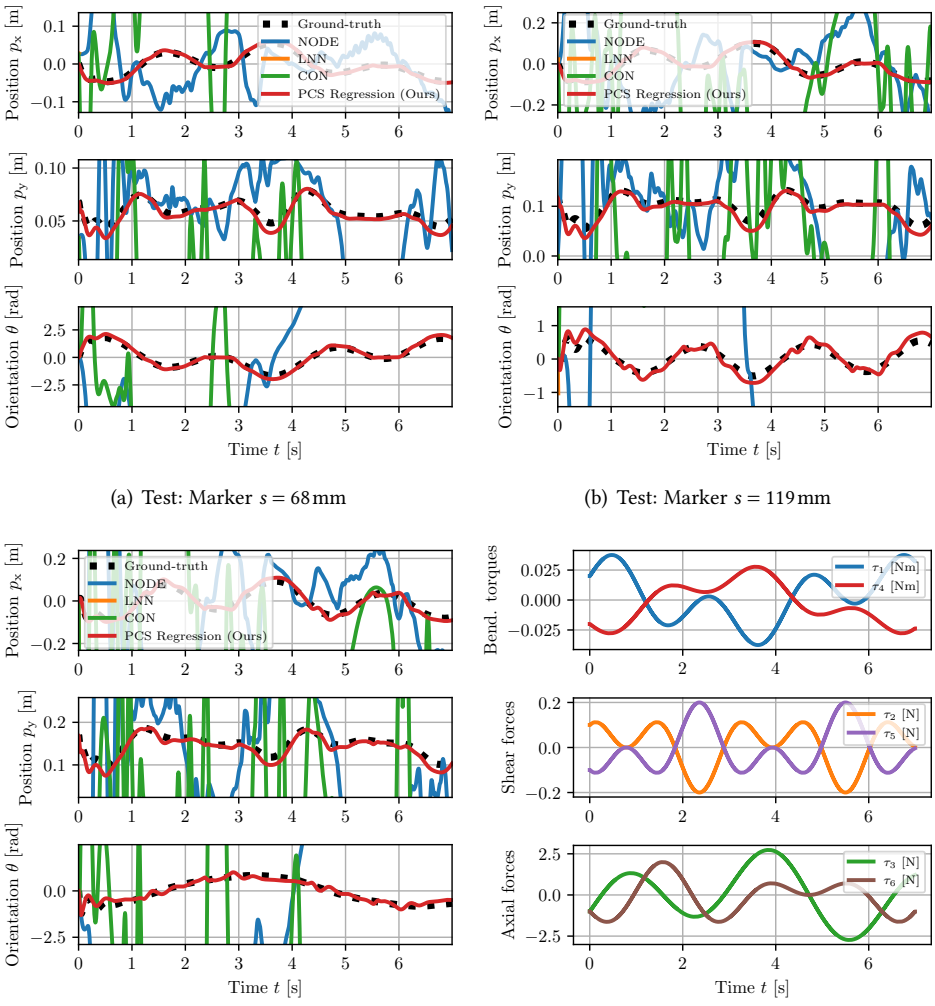


(c) Train: Marker $s = 170\,\text{mm}$ (EE)



(d) Train: Actuation torques

Figure 10.9: Benchmarking of the proposed method against various machine learning baselines on a PCS soft robot consisting of two segments (*Case 2*) on the **train dataset**: We train the baseline methods on the dynamical evolution of the Cartesian SE(2) poses of three *markers* distributed over the backbone of the soft robot with a total length of 170 mm. The upper and lower rows visualize the rollout of all methods on the training and the test set, respectively. The first three columns show the evolution of each of the three markers, where the last marker represents the end-effector. The last column displays the actuation torques that were used to generate the datasets.



(a) Test: Marker $s = 68\,\text{mm}$

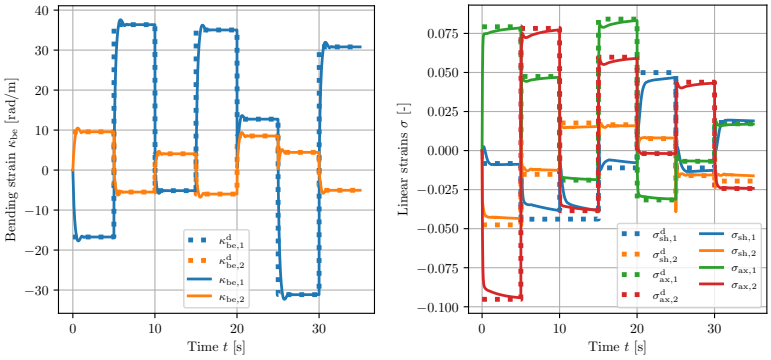
(b) Test: Marker $s = 119\,\text{mm}$

(c) Test: Marker $s = 170\,\text{mm}$ (EE)

(d) Test: Actuation torques

10

Figure 10.10: Benchmarking of the proposed method against various machine learning baselines on a PCS soft robot consisting of two segments (*Case 2*) on the **test dataset**: We train the baseline methods on the dynamical evolution of the Cartesian SE(2) poses of three *markers* distributed over the backbone of the soft robot with a total length of 170 mm. The upper and lower rows visualize the rollout of all methods on the training and the test set, respectively. The first three columns show the evolution of each of the three markers, where the last marker represents the end-effector. The last column displays the actuation torques that were used to generate the datasets.



(a) Bending strains (b) Linear strains

Figure 10.11: Demo of model-based control of a two-segment soft robot based on the learned dynamical model. We ask the controller to track a sequence of 7 setpoints $q^{d} \in \mathbb{R}^{6}$ which is denoted with dotted lines. The controller contains a feedforward and feedback term where the feedforward term compensates for the elastic and gravitational forces at the setpoint.

10.5 Conclusion

In this work, we propose a data-driven approach leveraging the PCS strain model to identify low-dimensional kinematic and dynamic models for continuum soft robots directly from discrete pose measurements of the backbone's shape. Compared to ML-based models (e.g., neural networks, symbolic regression), we preserve the physical structure of the continuum soft robot model, leading to a significantly improved training data efficiency and out-of-training-distribution performance. Furthermore, our method preserves the physical structure, enabling fast and efficient model-based control design. Compared to deriving and formulation continuum soft robot models by hand using PCC, PCS, etc. approximations, our approach requires (i) less expert knowledge as the number of segments, segment lengths, active strains, etc. are automatically determined and (ii) we can regress all dynamic parameters with closed-form linear least-squares which is more efficient and robust than traditionally used system identification procedures (e.g., constrained nonlinear least squares, determination of parameters using mechanical testing equipment).

We verified both the *Kinematic Fusion* and the *Dynamic Regression and Sparsification* algorithms in simulation. The *Kinematic Fusion* method can automatically and accurately determine the number of planar PCS segments and their respective length purely based on extracted SE(2) poses of the backbone shape. For continuum soft robots whose shape by definition cannot be represented by the PCS model (e.g., soft robots exhibiting affine curvature), we formulated a Pareto front between the DOF of the model and the shape reconstruction accuracy. This enables the user to easily choose the best kinematic model for a given computational budget. We showed that our proposed method is able to derive accurate dynamic models from just 4 s of video recordings and can generalize to out-of-distribution actuation sequences, which is not the case for the ML-based baseline methods that we considered (e.g., NODE, LNN (Liu et al., 2024a), and CON (Stölzle and Della Santina, 2024)). Furthermore, even for samples inside the training distribution, our method exhibits a more than 70 % lower shape prediction error than the baselines. Finally, we demonstrated that the physical structure of the dynamical model that remains intact during the regression allows us to leverage the learned dynamics within closed-form model-based control in a plug-and-play fashion. For future work, it would be interesting to validate the proposed approach on both 3D (i.e., $SE(3)$ input poses) and real-world data. Furthermore, it would be valuable to identify ways to regress a possibly underactuated actuation matrix $A(q) \in \mathbb{R}^{n_q \times m}$, where m is the number of actuators.

Afterword

In this chapter, we presented a data-driven approach for identifying strain-based models for continuum soft robots. Crucially, we preserve all insight and physical interpretability into the kinematic and dynamical models, respectively. These features enable us to *debug* the model and apply the fundamentally the same model-based control concepts (e.g., P-satID+potential shaping) that we previously used in Chapter 6 for the HSA robot. Furthermore, the strain distance threshold in the *Kinematic Fusion* algorithm and the maximum stiffness threshold in the *Strain Sparsification* algorithm allow us to make the tradeoff between model complexity and model performance (i.e., shape reconstruction for the kinematic model and prediction accuracy for the dynamic model) explicit. As we basically derive an

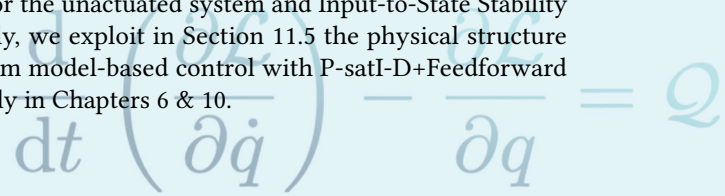
entirely *whitebox* model, the dynamic model exhibits excellent extrapolation performance compared to *blackbox* or even *greybox* (e.g., LNNs) SOTA ML techniques. Although the approach presented in this chapter demonstrates excellent performance and produces low-dimensional, control-oriented models, it cannot capture very complex soft robot behaviors. These include deformations of the backbone cross-sections, which the Cosserat rod model neglects, as well as time-dependent effects such as hysteresis. While features like nonlinear elasticity, observed in HSA robots (see Chapter 5), could theoretically be integrated into the model, they currently cannot be automatically identified by the algorithm introduced in this chapter. Additionally, since the approach relies on a PCS-based parametrization derived from Cosserat rod theory, it performs poorly for robots that do not satisfy the *slender rod* assumption. Moreover, the problem setting assumes access to pose samples of the backbone's shape. However, in scenarios where soft robots are intended to assist humans with daily activities, exteroceptive sensing methods like Motion Capture Systems (MCSs) become impractical. Extracting the backbone shape using Computer Vision (CV) is also prone to errors, especially when tracking key points along the soft robot's body over time is challenging.

Therefore, in Chapter 11, we propose an alternative approach that employs a network of basic mechanical systems (e.g., a harmonic oscillator) to learn latent dynamics directly from image pixels. This method can be viewed as a *greybox* model, where both the reduced-order coordinates (approximated by an autoencoder) and the dynamical system (approximated by a CON) are learned using modern ML techniques. Crucially, this approach retains a mechanical interpretation of both latent variables and their dynamics, enabling analysis of the learned latent system's properties (e.g., kinetic and potential energy). This insight can then be leveraged for model-based control within the latent space.

11

Learning Latent Dynamics with ISS Coupled Oscillator Networks

Foreword. In Chapter 10, we presented a data-driven approach for learning strain models for continuum soft robots that exhibits full *white-box* interpretability of the identified kinematic parametrization and the physics-based dynamics. However, the kinematic and dynamical behavior of certain more complex robots, such as non-slender soft robots or soft robots exhibiting nonlinear stiffness, varying cross-sections, and hysteresis, cannot be captured by the proposed strain-based approach that builds on the Cosserat rod theory. Furthermore, we sometimes cannot easily extract low-dimensional state information and instead only have access to high-dimensional observation in the form of images or depth measurements. As a solution, we leverage in this chapter Coupled Oscillator Network (CON) for learning latent dynamics of physical systems directly from image observations. The CON network, as introduced in Section 11.2, consists of harmonic oscillators that are coupled by a potential energy. As described in Section 11.4, we utilize a VAE-based autoencoder to encode the pixels into latent space. Interestingly, this setup now allows us to assign a mechanical interpretation to the latent encoding of the image, as each latent variable corresponds to the position of a harmonic oscillator. Furthermore, we can learn the potential energy of the system in such a way that strong stability guarantees for the latent dynamics are automatically incorporated. In particular, we can, for example, guarantee Global Asymptotic Stability (GAS) for the unactuated system and Input-to-State Stability (ISS) for the actuated system. Finally, we exploit in Section 11.5 the physical structure of the CON dynamics for closed-form model-based control with P-satI-D+Feedforward controllers, as already seen previously in Chapters 6 & 10.



Abstract. *Even though a variety of methods have been proposed in the literature, efficient and effective latent-space control (i.e., control in a learned low-dimensional space) of physical systems remains an open challenge. We argue that a promising avenue is to leverage powerful and well-understood closed-form strategies from control theory literature in combination with learned dynamics, such as potential-energy shaping. We identify three fundamental shortcomings in existing latent-space models that have so far prevented this powerful combination: (i) they lack the mathematical structure of a physical system, (ii) they do not inherently conserve the stability properties of the real systems, (iii) these methods do not have an invertible mapping between input and latent-space forcing. This work proposes a novel Coupled Oscillator Network (CON) model that simultaneously tackles all these issues. More specifically, (i) we show analytically that CON is a Lagrangian system - i.e., it possesses well-defined potential and kinetic energy terms. Then, (ii) we provide formal proof of global Input-to-State stability using Lyapunov arguments. Moving to the experimental side, we demonstrate that CON reaches SOTA performance when learning complex nonlinear dynamics of mechanical systems directly from images. An additional methodological innovation contributing to achieving this third goal is an approximated closed-form solution for efficient integration of network dynamics, which eases efficient training. We tackle (iii) by approximating the forcing-to-input mapping with a decoder that is trained to reconstruct the input based on the encoded latent space force. Finally, we leverage these three properties and show that they enable latent-space control. We use an integral-saturated PID with potential force compensation and demonstrate high-quality performance on a soft robot using raw pixels as the only feedback information.*

This chapter is partly based on 🏆 ***M. Stölzle**, and C. Della Santina (2024). Input-to-State Stable Coupled Oscillator Networks for Closed-form Model-based Control in Latent Space. In Proceedings of Advances in Neural Information Processing Systems (NeurIPS) 37, Accepted as a **Spotlight** paper (Stölzle and Della Santina, 2024).*

M.S. and C.D.S. conceived the project, developed the formulation of the coupled oscillator network, and derived the stability proof. M.S. devised the closed-form approximation to the CON dynamics, came up with the model-based control approach, and implemented the training and control pipeline. M.S. planned and conducted all experiments and wrote the manuscript. C.D.S. revised the manuscript, provided funding, and supervised the project.

11.1 Introduction

Learning how the environment evolves around us from high-dimensional observations (i.e., world models (Ha and Schmidhuber, 2018)) is essential for achieving both artificial and physical intelligence (Hafner et al., 2023). For example, world models are required for effectively planning an artificial/robotic agent's actions in complex and unstructured environments (Matsuo et al., 2022). However, learning such dynamics directly in high-dimensional observation space is usually intractable. Seminal works have shown that we can leverage autoencoders to compress the state information into a low-dimensional latent space (Kingma and Welling, 2014; Liou et al., 2014) in which it is much more feasible to learn the dynamics (Champion et al., 2019; Lenz et al., 2015; Wahlström et al., 2015; Watter et al., 2015; Zhong and Leonard, 2020). However, strong limitations still persist when it comes to using these learned models to generate low-level intelligence.

One outstanding challenge is how to perform closed-loop control in the learned latent space - i.e., how to generate control inputs based on a high dimensional sensory input such that a desired movement is generated. Prior works have explored, among other approaches, RL (Gelada et al., 2019; Hafner et al., 2019a; Schwarting et al., 2021; Van Hoof et al., 2016), MPC (Alora et al., 2023b; Hafner et al., 2019b; Hewing et al., 2020; Lenz et al., 2015), LQRs (Brunton et al., 2016; Haggerty et al., 2023; Mamakoukas et al., 2019) and gradient-based optimization (Yamada et al., 2023) for planning and control towards a target evolution that is given in observation space. However, all existing latent-space control strategies have shortcomings, such as a limited planning horizon and slow control rates (MPC and gradient-based approaches), sample inefficiency (RL), or they pose a requirement for learning linear dynamics (Watter et al., 2015) (LQR), which is not even possible for systems that are inherently non-linearizable (Cenedese et al., 2022). One interesting avenue is to leverage model-based control approaches, such as potential shaping (Bloch et al., 2001; Ortega et al., 2021; Zhong and Leonard, 2020), for effective and computationally efficient control in latent space (Lepri et al., 2023). For these techniques to be feasible, the dynamical model needs to fulfill four characteristics: (i) the dynamics need to have the mathematical structure of physical systems, (ii) conserve the stability properties of real systems, (iii) the latent state needs to be relatively low-dimensional, and (iv) there needs to exist a well-defined, invertible mapping between the input and the forcing in latent space. However, existing model structures that are used for learning latent dynamics (Botev et al., 2021) do not meet all of these criteria. Relevant examples are MLPs, NODEs (Chen et al., 2018; Sholokhov et al., 2023), many variants of RNNs (e.g., LSTMs (Hochreiter and Schmidhuber, 1997), GRUs (Cho et al., 2014; Roth et al., 2025), etc.), and physics-informed neural networks (e.g., LNNs (Cranmer et al., 2020; Lutter et al., 2019; Zhong and Leonard, 2020), HNNs (Greydanus et al., 2019)). For example, MLPs do not have a physical interpretation and do not provide an invertible mapping of the forcing generated by the input, NODEs are usually not easily stabilizable (White et al., 2023), most RNNs require a relatively high-dimensional latent space (i.e., many hidden states), and energy-shaping control approaches based on LNNs (Zhong and Leonard, 2020) do not come with any formal stability guarantees.

In recent years, oscillatory networks (Ceni et al., 2024; Lanthaler et al., 2024; Rusch and Mishra, 2020, 2021; Rusch and Rus, 2024) have been shown to exhibit state-of-the-art performance on time sequence modeling tasks while being parameter-efficient, thus

fulfilling our requirement (iii). Consequently, we believe that they are a promising option for control-oriented dynamics learning in latent space. Still, these models do not fulfill the remaining requirements that we have listed above. Despite being an interpretable combination of harmonic oscillators, they do not have the structure of a physical system - i.e., they do not possess a well-defined energy function. Moreover, only local stability (Ceni et al., 2024; Rusch and Mishra, 2020) has been shown, with sufficient conditions that appear to be very stringent. Finally, in addition to training an encoder that maps inputs to latent-space forcing, we propose also training a decoder that learns to reconstruct inputs based on latent-space forcing. This enables us to easily switch between inputs and forcing, which is essential when implementing control strategies.

We resolve all the above-mentioned challenges by proposing CONs, a new formulation of a coupled oscillator network that is inherently Input-to-State Stability (ISS) stable, for learning the dynamics of physical systems and subsequently exploiting its structure for model-based control in latent space. The network consists of damped, harmonic oscillators connected through elastic springs, damping elements, and a neuron-like coupling force and can be excited by a nonlinear actuation term. We identify a transformation into a set of coordinates from which we can derive the networks' kinetic and potential energy. This allows us to leverage Lyapunov arguments (Khalil, 2002) for proving the global asymptotic stability of the unforced system and ISS stability for the forced system under relatively mild assumptions on the network parameters. Even though we constrain the dynamics to a very specific structure, we demonstrate (a) the CON network achieves similar performance as NODEs when learning the dynamics of unactuated, mechanical systems with two orders of magnitude fewer parameters and (b) that the proposed model achieves, for the complex task of learning the actuated, highly nonlinear dynamics of continuum soft robots (Alora et al., 2023a; Stölzle and Della Santina, 2021) directly from pixels, a 60 % lower prediction error than coRNN (Rusch and Mishra, 2020) and reaches the SoA performance across all techniques that we tested. Finally, we show some initial results that the proposed CON model is also able to learn the latent dynamics of PDEs, in this case containing reaction-diffusion (Champion et al., 2019; Epstein and Xu, 2016) dynamics.

Subsequently, we derive an approximate closed-form solution, that is, in parameter regimes in which the linear, decoupled dynamics dominate transient, more accurate than numerical integrators with comparable computational requirements and which increases training speed by 2x with a small decrease in prediction accuracy. Finally, as we can derive the system's potential energy, we can leverage potential shaping (Bloch et al., 2001; Ortega et al., 2021) to derive a controller that combines an integral-saturated PID controller with a feedforward term compensating potential forces. As the feedback acts on a well-shaped potential field, tuning the feedback gains becomes very simple and out-of-the-box, and the controller exhibits a faster response time and a 26 % lower trajectory tracking RMSE than a pure feedback controller based on a latent NODE (Chen et al., 2018) model.

The proposed methodology is particularly well-suited for learning the latent dynamics of mechanical systems with continuous dynamics, dissipation, and a single, attractive equilibrium point. Examples of such systems include many soft robots, deformable objects with dominant elastic behavior, Lagrangian systems immersed in a dominant potential field, or locally other mechanical systems such as robotic manipulators, legged robots, etc. For these systems, we can fully leverage the structural prior of the proposed latent

dynamics, including the integrated stability guarantees. If the system is actuated, the learned dynamics can be subsequently exploited for model-based control, as demonstrated in Sec. 11.5.

The code associated with this chapter is available on GitHub[1]. Furthermore, we provide more details about the experimental setup and present additional results in Appendix B.

11.2 Input-to-State Stable (ISS) Coupled Oscillator Networks (CONs)

11.2.1 Formulation

The integral component to (coupled) oscillatory RNNs (Ceni et al., 2024; Lanthaler et al., 2024; Rusch and Mishra, 2020, 2021) are one-dimensional, potentially damped, harmonic oscillators, which are described by their state $y_i = \begin{bmatrix} x_i & \dot{x}_i \end{bmatrix}^\top \in \mathbb{R}^2$, where x_i and $\dot{x}_i$ are the position and velocity of the oscillator, respectively. Then, the oscillator's dynamics are defined by the following EOM

$$m_i \ddot{x}_i(t) + d_i \dot{x}_i(t) + \kappa_i x_i(t) = F_i(t), \qquad \text{with } m_i, \kappa_i, d_i \in \mathbb{R}^+. \tag{11.1}$$

Here, m_i is the mass, κ_i is the stiffness, and d_i is the damping coefficient of the damped harmonic oscillator. $F_i(t) \in \mathbb{R}$ is a (possibly time-dependent) external forcing term acting on the mass.

Even though the state is extremely low dimensional and the number of parameters is small, this single, damped harmonic oscillator can already exhibit a variety of (designable) behaviors: The expressions $\omega_{\mathrm{n},i} = \sqrt{\frac{\kappa_i}{m_i}}$ and $\zeta_i = \frac{d_i}{2\sqrt{\kappa_i m_i}}$ let us determine the natural frequency and the damping factor, respectively and allow us to design the transient behavior. For example, $\omega_{\mathrm{n},i}$ lets us isolate a spectrum of the input signal $F_i(t)$ (Ceni et al., 2024) and ζ_i determines the damping regime: underdamped ($\omega_{\mathrm{n},i} < 1$), critically damped ($\omega_{\mathrm{n},i} = 1$), overdamped ($\omega_{\mathrm{n},i} > 1$). Furthermore, as (damped) harmonic oscillators are omnipresent in nature (and especially in physical systems), they have been intensively studied and are well understood (e.g., characteristics, closed-form solutions, etc.). In this work, we will exploit some of these properties and knowledge to learn stable (latent) dynamics efficiently.

By intercoupling damped harmonic oscillators, we can drastically increase the expressiveness of the dynamical system (Ceni et al., 2024; Lanthaler et al., 2024; Rusch and Mishra, 2020) while preserving some of the intuition and understanding we have for these systems. In this work, we propose a ISS-stable CON consisting of n damped harmonic oscillators that are coupled through both linear and nonlinear terms. The networks' state is defined as $y = \begin{bmatrix} x^\top & \dot{x}^\top \end{bmatrix}^\top \in \mathbb{R}^{2n}$ and its dynamics can be formulated as a 2^{nd}-order ODE

$$\dot{y}(t) = \begin{bmatrix} \frac{\mathrm{d}x}{\mathrm{d}t} \\ \frac{\mathrm{d}\dot{x}}{\mathrm{d}t} \end{bmatrix} = f(y(t), u(t)) = \begin{bmatrix} \dot{x}(t) \\ g(u(t)) - Kx(t) - D\dot{x}(t) - \tanh(Wx(t) + b) \end{bmatrix}, \tag{11.2}$$

where $K, D \in \mathbb{R}^{n \times n}$ are the linear stiffness and damping matrices, respectively. The neuron-inspired term $\tanh(Wx(t)+b)$ with $W \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$ provides nonlinear coupling between the harmonic oscillators. The network is excited by the time-dependent input $u(t) \in \mathbb{R}^m$

[1] https://edu.nl/yu9vw

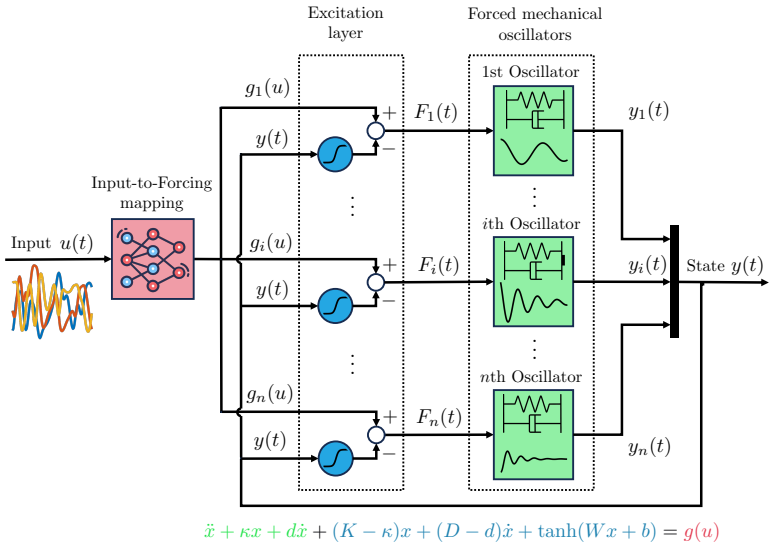


Figure 11.1: The proposed Coupled Oscillator Network (CON) network consists of n damped harmonic oscillators that are coupled through the neuron-like connection $\tanh(Wx+b)$ and the non-diagonal stiffness $K-k$ and damping coefficients $D-d$, respectively. The state of the network is captured by the positions $x(t)$ and velocities $\dot{x}(t)$ of the oscillators. The time-dependent input is mapped through the (possibly nonlinear) function $g(u)$ to a forcing τ acting on the oscillators.

through the possibly nonlinear mapping $g : \mathbb{R}^m \to \mathbb{R}^n$. Specifically, we consider in this work a formulation where an input-dependent matrix $B(u) \in \mathbb{R}^{n\times m}$ projects the input $u(t)$ to a time-dependent forcing on the oscillators: $\tau = g(u) = B(u)\,u$. Here, $B(u)$ could, for example, be parametrized by a MLP.

We specifically designed the network architecture such that (i) the system exhibits a unique and isolated equilibrium and (ii) we can derive expressions for the kinetic and potential energies. These two features allow us to (a) prove GAS and ISS stability using an established procedure based on strict Lyapunov arguments (Calzolari et al., 2020; Wu et al., 2022), and (b) implement model-based controller based on potential shaping.

11.2.2 Potential Energy and Derivation of Equilibria

One key insight of this work is that in the coordinates $x(t), \dot{x}(t)$, we cannot derive a potential as the hyperbolic force $\tanh(Wx(t)+b)$ is not symmetric. Therefore, we propose a coordinate transformation into $\mathcal{W}$-coordinates: $y_\mathrm{w}(t) = \begin{bmatrix} x_\mathrm{w}(t) \\ \dot{x}_\mathrm{w}(t) \end{bmatrix} = \begin{bmatrix} W\,x(t) \\ W\,\dot{x}(t) \end{bmatrix} \in \mathbb{R}^{2n}$. The coordinate transformation is valid if its Jacobian is full-rank, which is the case if $\mathrm{rank}(W) = n$. In $\mathcal{W}$-coordinates, the dynamics can be rewritten as

$$\dot{y}_\mathrm{w}(t) = \begin{bmatrix} \frac{\mathrm{d}x_\mathrm{w}}{\mathrm{d}t} \\ \frac{\mathrm{d}\dot{x}_\mathrm{w}}{\mathrm{d}t} \end{bmatrix} = f_\mathrm{w}(y(t), u(t)) = \begin{bmatrix} \dot{x}_\mathrm{w}(t) \\ M_\mathrm{w}^{-1}\left(g(u(t)) - K_\mathrm{w} x_\mathrm{w}(t) - D_\mathrm{w}\,\dot{x}_\mathrm{w}(t) - \tanh(x_\mathrm{w}(t)+b)\right) \end{bmatrix} \tag{11.3}$$

with $K_\mathrm{w} = K\,W$, $D_\mathrm{w} = D\,W$ and $M_\mathrm{w} = W^{-1}$.

A difference of this formulation compared to prior work (Ceni et al., 2024; Lanthaler et al., 2024; Rusch and Mishra, 2020, 2021) is that (i) the forcing produced by the input term $\tau = g(u)$ is fully separated from the forcing produced by the elastic coupling terms K_w, and (ii) the generalized force is symmetric as stated in Lemma 2, allowing us to define a potential energy expression, which we can later on leverage for stability analysis and control.

Lemma 2. *Let $\tilde{x}_\mathrm{w} \in \mathbb{R}^n$ be generalized coordinates and $\bar{x}_\mathrm{w}, b \in \mathbb{R}^n$ constants. Then, the potential force of system* (11.10)

$$\tilde{f}_{\mathcal{U}_\mathrm{w}}(\tilde{x}_\mathrm{w}) = K_\mathrm{w}(\bar{x}_\mathrm{w} + \tilde{x}_\mathrm{w}) + \tanh(\bar{x}_\mathrm{w} + \tilde{x}_\mathrm{w} + b), \tag{11.4}$$

stems from the potential

$$\mathcal{U}_\mathrm{w}(\tilde{x}_\mathrm{w}) = \sum_{i=1}^{n} \int_0^{\tilde{x}_{\mathrm{w},i}} \tanh(\bar{x}_{\mathrm{w},i} + \sigma + b_i)\,\mathrm{d}\sigma - \sum_{i=1}^{n} \int_0^{\tilde{x}_{\mathrm{w},i}} \tanh(\bar{x}_{\mathrm{w},i} + b_i)\,\mathrm{d}\sigma \in \mathbb{R}. \tag{11.5}$$

Proof. First, we take the derivative of $\mathcal{U}(\tilde{x}_\mathrm{w})$:

$$\frac{\partial \mathcal{U}_\mathrm{w}}{\partial \tilde{x}_\mathrm{w}} = K_\mathrm{w}(\bar{x}_\mathrm{w} + \tilde{x}_\mathrm{w}) + \tanh(\bar{x}_\mathrm{w} + \tilde{x}_\mathrm{w} + b) = \tilde{f}_{\mathcal{U}_\mathrm{w}}. \tag{11.6}$$

The Hessian of the potential is given by

$$H_{\mathcal{U}_\mathrm{w}}(\tilde{x}_\mathrm{w}) = \frac{\partial^2 \mathcal{U}_\mathrm{w}}{\partial \tilde{x}_\mathrm{w}^2} = \frac{\partial \tilde{f}_{\mathcal{U}_\mathrm{w}}}{\partial \tilde{x}_\mathrm{w}} = K_\mathrm{w} + S^2_{\mathrm{sech}}(\tilde{x}_\mathrm{w}) \in \mathbb{R}^{n\times n}. \tag{11.7}$$

As $K_\mathrm{w} \succ 0 \Rightarrow K_\mathrm{w} = K_\mathrm{w}^\top$, we can easily show that the potential force is symmetric:

$$H_{\mathcal{U}_\mathrm{w}}^\top = K_\mathrm{w}^\top + S^2_{\mathrm{sech}}(\tilde{x}_\mathrm{w})^\top = K_\mathrm{w} + S^2_{\mathrm{sech}}(\tilde{x}_\mathrm{w}) = H_{\mathcal{U}_\mathrm{w}}. \tag{11.8}$$

□

The equilibria $\bar{y}_\mathrm{w} = \begin{bmatrix}\bar{x}_\mathrm{w}^\top & 0^\top\end{bmatrix}^\top \in \mathbb{R}^{2n}$ of the unforced network are given by the roots of the characteristic equation $\tanh(\bar{x}_\mathrm{w} + b) + K_\mathrm{w}\,\bar{x}_\mathrm{w} = 0$.

Lemma 3. *Let $K_\mathrm{w} \succ 0$. Then, the dynamics defined in* (11.3) *have a single, isolated equilibrium $\bar{y}_\mathrm{w} = \begin{bmatrix}\bar{x}_\mathrm{w}^\top & 0^\top\end{bmatrix}^\top$.*

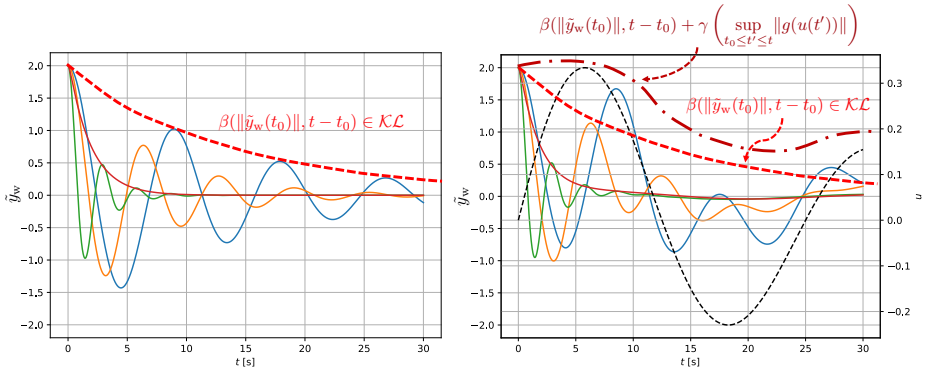
Proof. We regard the characteristic equation as a function: $h_\mathrm{eq}(x_\mathrm{w}) = \tanh(x_\mathrm{w} + b) + K_\mathrm{w}\,x_\mathrm{w}$. For there to exist multiple equilibria, $h_\mathrm{eq}(\bar{x}_\mathrm{w}) = 0$ would need to be true for multiple $\bar{x}$. However, we take the partial derivative of $h_\mathrm{eq}(x_\mathrm{w})$ w.r.t. x_w and see that

$$\frac{\partial h_\mathrm{eq}}{\partial x_\mathrm{w}} = K_\mathrm{w} + S^2_{\mathrm{sech}}(x_\mathrm{w}) \succ 0, \quad \forall x_\mathrm{w} \in \mathbb{R}^n \qquad \text{with } S_{\mathrm{sech}}(x_\mathrm{w}) = \mathrm{diag}(\mathrm{sech}(x_\mathrm{w} + b)) \in \mathbb{R}^{n\times n} \tag{11.9}$$

as $S_{\mathrm{sech}}(x_\mathrm{w}) \succ 0 \quad \forall x_\mathrm{w} \in \mathbb{R}^n$ and $K_\mathrm{w} \succ 0$. Therefore, $h_\mathrm{eq}(x_\mathrm{w})$ is continuously increasing and can only cross the zero line once. □

Next, we introduce a mapping into the tilde coordinates $\tilde{y}_\mathrm{w} = y_\mathrm{w} - \bar{y}_\mathrm{w}$. The residual dynamics (w.r.t. the equilibrium $\bar{y}_\mathrm{w}$) can now be stated as

$$\dot{\tilde{y}}_\mathrm{w}(t) = \tilde{f}_\mathrm{w}(y, u) = \begin{bmatrix} \dot{\tilde{x}}_\mathrm{w}(t) \\ M_\mathrm{w}^{-1}\left(g(u(t)) - K_\mathrm{w}(\bar{x}_\mathrm{w} + \tilde{x}_\mathrm{w}(t)) - D_\mathrm{w}\,\dot{\tilde{x}}_\mathrm{w}(t) - \tanh(\bar{x}_\mathrm{w} + \tilde{x}_\mathrm{w}(t) + b)\right) \end{bmatrix} \tag{11.10}$$



(a) GAS: $g(u) = 0$ (b) ISS

Figure 11.2: Illustration of global asymptotic stability for the unforced system with $g(u) = 0$ and input-to-state stability for the forced system, where the black dashed line denotes the input $u(t)$.

11.2.3 Global Asymptotic Stability (GAS) for the unforced system

We first consider the unforced system with $\tau = g(u) = 0, \forall t \in [t_0, t_\infty)$ and strive to prove global asymptotic stability (Khalil, 2002) for the attractor $\bar{x}$. For this, we need to first identify a valid (strict) Lyapunov function and subsequently demonstrate that, except at the global equilibrium, the time derivative of the Lyapunov function is negative definite.

Strict Lyapunov candidate

We propose a strict Lyapunov candidate with skewed level sets (Wu et al., 2022)

$$\begin{aligned} V_\mu(\tilde{y}_w) &= \frac{1}{2}\tilde{y}_w^\top P_V \tilde{y}_w + \sum_{i=1}^{n} \int_0^{\tilde{x}_{w,i}} \tanh(\bar{x}_{w,i} + \sigma + b_i)\,\mathrm{d}\sigma - \sum_{i=1}^{n} \int_0^{\tilde{x}_{w,i}} \tanh(\bar{x}_{w,i} + b_i)\,\mathrm{d}\sigma, \\ &= \frac{1}{2}\tilde{y}_w^\top P_V \tilde{y}_w + \sum_{i=1}^{n} \left(\mathrm{lcosh}(\bar{x}_{w,i} + \tilde{x}_{w,i} + b_i) - \mathrm{lcosh}(\bar{x}_{w,i} + b_i) - \tanh(\bar{x}_{w,i} + b_i)\,\tilde{x}_{w,i}\right), \\ \text{with } P_V &= \begin{bmatrix} K_w & \mu M_w \\ \mu M_w^\top & M_w \end{bmatrix} \in \mathbb{R}^{2n \times 2n}, \qquad \mathrm{lcosh}(\cdot) = \log(\cosh(\cdot)), \qquad \text{and } \mu > 0. \end{aligned} \tag{11.11}$$

In the following, we will write $\|A\|$ to denote the induced norm of matrix A and $\lambda_m(A)$, $\lambda_M(A)$ to refer to its minimum and maximum eigenvalue respectively. The gradient of $V_\mu(\tilde{y}_w)$ w.r.t. the residual coordinate $\tilde{y}_w$ is given by

$$\frac{\partial V_\mu}{\partial \tilde{y}_w}(\tilde{y}_w) = P_V \tilde{y}_w + \begin{bmatrix} \tanh(\bar{x}_w + \tilde{x}_w + b) \\ 0^n \end{bmatrix} - \begin{bmatrix} \tanh(\bar{x}_w + b) \\ 0^n \end{bmatrix}. \tag{11.12}$$

Next, the Hessian of the Lyapunov candidate can be derived as

$$H_V(\tilde{x}_w) = \frac{\partial^2 V_\mu}{\partial \tilde{y}_w^2} = \begin{bmatrix} K_w + S^2_{\mathrm{sech}}(\tilde{x}_w) & \mu M_w \\ \mu M_w^\top & M_w \end{bmatrix} \in \mathbb{R}^{2n \times 2n}, \tag{11.13}$$

where

$$S_{\text{sech}}(\tilde{x}_\text{w}) = \text{diag}(\text{sech}(\bar{x}_\text{w} + \tilde{x}_\text{w} + b)) \in \mathbb{R}^{n\times n} \succ 0 \quad \forall \tilde{x}_\text{w} \in \mathbb{R}^n. \tag{11.14}$$

Furthermore, the Schur complement of P_V is given by

$$S_{P_\text{V}} = M_\text{w} - \mu^2 M_\text{w}^\top K_\text{w} M_\text{w} \tag{11.15}$$

Below, we will first introduce three auxiliary Lemmas and subsequently proof Lemma 7 that states that (11.11) is a valid Lyapunov candidate.

Lemma 4. *Suppose* $M_\text{w} \succ 0, K_\text{w} \succ 0$ *and* $0 < \mu < \frac{\sqrt{\lambda_\text{m}(M_\text{w})\lambda_\text{m}(K_\text{w})}}{\|M_\text{w}\|} := \mu_\text{V}$. *Then* S_{P_V}, *as defined in* (11.15), *is positive definite.*

Proof. The minimum eigenvalue of S_{P_V} is bounded by

$$\begin{aligned} \lambda_\text{m}(S_{P_\text{V}}) &\geq \lambda_\text{m}(M_\text{w}) - \mu^2 \|M_\text{w}^\top K_\text{w} M_\text{w}\|, \\ &\geq \lambda_\text{m}(M_\text{w}) - \mu^2 \frac{\|M_\text{w}\|^2}{\lambda_\text{w}(K_\text{w})} \end{aligned} \tag{11.16}$$

Based on the assumption $M_\text{w} \succ 0, K_\text{w} \succ 0$, we can state $\frac{\|M_\text{w}\|}{\lambda_\text{w}(K_\text{w})} > 0$. Therefore, the critical case for the lower bound on $\lambda_\text{m}(S_{P_\text{V}})$ is $\mu = \frac{\sqrt{\lambda_\text{m}(M_\text{w})\lambda_\text{m}(K_\text{w})}}{\|M_\text{w}\|} := \mu_\text{V}$. Hence,

$$\lambda_\text{m}(S_{P_\text{V}}) > \lambda_\text{m}(M_\text{w}) - \frac{\lambda_\text{m}(M_\text{w})\lambda_\text{m}(K_\text{w})}{\|M_\text{w}\|^2} \frac{\|M_\text{w}\|^2}{\lambda_\text{w}(K_\text{w})} = 0 \tag{11.17}$$

Consequently, the eigenvalue sensitivity theorem (Golub and Van Loan, 2013) demands that $S_{P_\text{V}} \succ 0$. □

Lemma 5. *Let* $M_\text{w} \succ 0, K_\text{w} \succ 0$, *and* $0 < \mu < \frac{\sqrt{\lambda_\text{m}(M_\text{w})\lambda_\text{m}(K_\text{w})}}{\|M_\text{w}\|} := \mu_\text{V}$. *Then, it follows that* $P_\text{V} \succ 0$ *and* $H_\text{V}(\tilde{x}_\text{w}) \succ 0 \ \forall \tilde{x}_\text{w} \in \mathbb{R}^n$.

Proof. By inspecting the expressions for $P_\text{V} \succ 0$ and $H_\text{V}(\tilde{x}_\text{w}) \succ 0 \forall \tilde{x}_\text{w} \in \mathbb{R}^n$ in Equations (11.11) and (11.13), respectively, it can be easily seen that $H_\text{V}(\tilde{x}_\text{w}) \succeq P_\text{V} \ \forall \tilde{x}_\text{w} \in \mathbb{R}$. As Lemma 4 states that the Schur complement of P_V is positive definite, it follows that $H_\text{V}(\tilde{x}_\text{w}) \succeq P_\text{V} \succ 0$. □

Lemma 6. *Suppose* $\bar{x}_\text{w}, \tilde{x}_\text{w}, b \in \mathbb{R}^n$ *and* $n \in \mathbb{N}^+$. *Then,*

$$h_{\text{V,th}}(\tilde{x}_\text{w}) = \sum_{i=1}^{n} \int_0^{\tilde{x}_{\text{w},i}} \tanh(\bar{x}_{\text{w},i} + \sigma + b_i)\,\text{d}\sigma - \sum_{i=1}^{n} \int_0^{\tilde{x}_{\text{w},i}} \tanh(\bar{x}_{\text{w},i} + b_i)\,\text{d}\sigma \tag{11.18}$$

is a positive semi-definite function.

Proof. Proving $h_{\text{V,th}}(\tilde{x}_\text{w}) \geq 0$ is equivalent to showing that the scalar function $\breve{h}_{\text{V,th}}(r) = \int_0^r \tanh(\sigma + a)\,\text{d}\sigma - \int_0^r \tanh(a)\,\text{d}\sigma \geq 0 \ \forall\, r, a \in \mathbb{R}$, where we set $r = \tilde{x}_{\text{w},i}$ and $a = \bar{x}_{\text{w},i} + b_i$.

We strive to find the critical points (i.e., minimas and maximas) $\bar{r}$ of $\breve{h}_{\text{V,th}}(r)$ and, for this, analyze where the first derivative of $\breve{h}_{\text{V,th}}(r)$ is zero

$$\frac{\partial \breve{h}_{\text{V,th}}}{\partial r}(\bar{r}) = \tanh(\bar{r} + a) - \tanh(a) = 0, \tag{11.19}$$

which is the case only for $\bar{r} = 0$. Next, we compute the second derivative at $\bar{r}$ as

$$\frac{\partial \check{h}_{\mathrm{V,th}}}{\partial r}(\bar{r}) = \operatorname{sech}^2(\bar{r}) = 1. \tag{11.20}$$

Thus, $\check{h}_{\mathrm{V,th}}(r)$ is convex and its global minimum at $\bar{r} = 0$ takes the value $h_{\mathrm{V,th}}(0) = 0$. As a result, $h_{\mathrm{V,th}}(\tilde{x}_{\mathrm{w}})$ is also positive semi-definite. □

Lemma 7. *The scalar function $V_\mu(\tilde{y}_{\mathrm{w}})$ defined in* (11.11) *is continuously differentiable and verifies the condition $V_\mu(0) = 0$. Furthermore, let $M_{\mathrm{w}}, K_{\mathrm{w}} \succ 0$. Now, if we choose $0 < \mu < \frac{\sqrt{\lambda_{\mathrm{m}}(M_{\mathrm{w}})\lambda_{\mathrm{m}}(K_{\mathrm{w}})}}{\|M_{\mathrm{w}}\|} := \mu_{\mathrm{V}}$, then $V_\mu(\tilde{y}_{\mathrm{w}}) > 0 \,\forall\, \tilde{y}_{\mathrm{w}} \in \mathbb{R}^{2n} \setminus \{0\}$. Additionally, then $V_\mu(\tilde{y}_{\mathrm{w}})$ is radially unbounded as $\|\tilde{y}_{\mathrm{w}}\| \to \infty \Rightarrow V_\mu(\tilde{y}_{\mathrm{w}}) \to \infty$.*

Proof. **Step 1:** It can be easily seen that $V_\mu(\tilde{y}_{\mathrm{w}})$ in (11.11) is smooth and continuously differentiable.

Step 2: Proof that $V_\mu(0) = 0$.

$$V_\mu(0) = 0 + \sum_{i=1}^{n} \int_0^0 \tanh(\bar{y}_{\mathrm{w},i} + \sigma + b_i)\,\mathrm{d}\sigma - \sum_{i=1}^{n} \int_0^0 \tanh(\bar{y}_{\mathrm{w},i} + b_i)\,\mathrm{d}\sigma = 0. \tag{11.21}$$

Step 3: Proof that the Lyapunov candidate is positive definite; i.e., $V_\mu(\tilde{y}_{\mathrm{w}}) > 0 \;\forall \tilde{y}_{\mathrm{w}} \in \mathbb{R}^n \setminus \{0\}$.

As the gradient of the Lyapunov candidate, as defined in (11.12), is zero for $\tilde{y}_{\mathrm{w}} = 0$:

$$\frac{\partial V_\mu}{\partial \tilde{y}_{\mathrm{w}}}(0) = \begin{bmatrix} \tanh(\bar{x}_{\mathrm{w}} + b) \\ 0^n \end{bmatrix} - \begin{bmatrix} \tanh(\bar{x}_{\mathrm{w}} + b) \\ 0^n \end{bmatrix} = 0, \tag{11.22}$$

$\tilde{y}_{\mathrm{w}} = 0$ is a critical point of $V_\mu(\tilde{y}_{\mathrm{w}})$. According to Lemma 5, the Hessian in (11.13) is positive-definite (Boyd and Vandenberghe, 2004): $H_{\mathrm{V}}(\tilde{y}_{\mathrm{w}}) \succ 0 \;\forall \tilde{y}_{\mathrm{w}} \in \mathbb{R}^{2n}$. With that, (11.11) is convex and its global minimum is at $\tilde{y}_{\mathrm{w}} = 0$, where $V_\mu(0) = 0$. In summary, we state $V_\mu(\tilde{y}_{\mathrm{w}}) > 0 \;\forall \tilde{y}_{\mathrm{w}} \in \mathbb{R}^n \setminus \{0\}$.

Step 4: Proof that the Lyapunov candidate is radially unbounded: i.e., $\|\tilde{y}_{\mathrm{w}}\| \to \infty \Rightarrow V_\mu(\tilde{y}_{\mathrm{w}}) \to \infty$. Lemma 6 is exploited for identifying a lower bound on $V_\mu(\tilde{y}_{\mathrm{w}})$:

$$\begin{aligned} V_\mu(\tilde{y}_{\mathrm{w}}) &= \frac{1}{2}\,\tilde{y}_{\mathrm{w}}^\top P_{\mathrm{V}}\,\tilde{y}_{\mathrm{w}} + \sum_{i=1}^{n} \int_0^{\tilde{x}_{\mathrm{w},i}} \tanh(\bar{x}_{\mathrm{w},i} + \sigma + b_i)\,\mathrm{d}\sigma - \sum_{i=1}^{n} \int_0^{\tilde{x}_{\mathrm{w},i}} \tanh(\bar{x}_{\mathrm{w},i} + b_i)\,\mathrm{d}\sigma, \\ &\geq \frac{1}{2}\,\tilde{y}_{\mathrm{w}}^\top P_{\mathrm{V}}\,\tilde{y}_{\mathrm{w}} \geq \frac{1}{2}\,\lambda_{\mathrm{m}}(P_{\mathrm{V}})\|\tilde{y}_{\mathrm{w}}\|^2. \end{aligned} \tag{11.23}$$

Lemma 5 tells us that $P_{\mathrm{V}} \succ 0$ and with that $\lambda_{\mathrm{m}}(P_{\mathrm{V}}) > 0$. Therefore, if $\|\tilde{y}_{\mathrm{w}}\| \to \infty$, it also follows that $V_\mu(\tilde{y}_{\mathrm{w}}) \to \infty$. □

Proof of Global Asymptotic Stability

In the following, we will demonstrate how the strict Lyapunov candidate of (11.11) allows us to prove Global Asymptotic Stability (GAS) of the unforced system. We will first introduce two auxiliary Lemmas before stating Theorem 2.

Lemma 8. *Suppose $\bar{x}_\mathrm{w}, \tilde{x}_\mathrm{w}, b \in \mathbb{R}^n$ and $n \in \mathbb{N}^+$. Then, the function $h_{\dot{\mathrm{V}},\mathrm{th}}(\tilde{x}_\mathrm{w})$ defined as*

$$h_{\dot{\mathrm{V}},\mathrm{th}}(\tilde{x}_\mathrm{w}) = \left(\tanh(\bar{x}_\mathrm{w} + \tilde{x}_\mathrm{w} + b) - \tanh(\bar{x}_\mathrm{w} + b)\right)^\top \tilde{x}_\mathrm{w}, \tag{11.24}$$

is positive semi-definite.

Proof. Proving $h_{\dot{\mathrm{V}},\mathrm{th}}(\tilde{x}_\mathrm{w}) \geq 0$ is equivalent to proving that the scalar function $\check{h}_{\dot{\mathrm{V}},\mathrm{th}}(r) = (\tanh(r+a) - \tanh(a))\, r \geq 0 \;\forall\, r, a \in \mathbb{R}$, where we set $r = \tilde{x}_{\mathrm{w},i}$ and $a = \bar{x}_{\mathrm{w},i} + b_i$. Now, we expand the hyperbolic tangent:

$$\begin{aligned}\check{h}_{\dot{\mathrm{V}},\mathrm{th}}(r) &= (\tanh(r+a) - \tanh(a))\, r = \left(\frac{e^{2(r+a)}-1}{e^{2(r+a)}+1} - \frac{e^{2a}-1}{e^{2a}+1}\right) r, \\ &= \frac{2\, e^{2a}\left(e^{2r}-1\right)}{e^{2r+4a} + e^{2r+2a} + e^{2a} + 1} r \geq 0,\end{aligned} \tag{11.25}$$

as the denominator $e^{2r+4a} + e^{2r+2a} + e^{2a} + 1 > 0 \;\forall\, r \in \mathbb{R}$ and as $\mathrm{sign}\left(2\, e^{2a}\left(e^{2r}-1\right)\right) = \mathrm{sign}(r)$. For example, $e^{2r} - 1 \geq 0 \;\forall\, r \geq 0$. Analog, $e^{2r} - 1 < 0 \;\forall\, r < 0$. □

Lemma 9. *Let $M_\mathrm{w} \succ 0$, $K_\mathrm{w} \succ 0$, and $D_\mathrm{w} \succ 0$. Also, let $\mu \in \mathbb{R}^+$ be chosen such that $0 < \mu < \frac{\lambda_\mathrm{m}(D_\mathrm{w})}{\lambda_\mathrm{m}(M_\mathrm{w}) + \frac{\|D_\mathrm{w}\|^2}{4\lambda_\mathrm{m}(K_\mathrm{w})}} := \mu_{\dot{\mathrm{V}}}$. Then, the matrix $P_{\dot{V}} = \begin{bmatrix} \mu K_\mathrm{w} & \frac{1}{2}\mu D_\mathrm{w} \\ \frac{1}{2}\mu D_\mathrm{w}^\top & D_\mathrm{w} - \mu M_\mathrm{w} \end{bmatrix} \in \mathbb{R}^n$ is positive definite.*

Proof. The Schur complement of $P_{\dot{V}}$ is given by

$$S_{P_{\dot{V}}} = D_\mathrm{w} - \mu M_\mathrm{w} - \frac{1}{4}\,\mu D_\mathrm{w}^\top K_\mathrm{w}^{-1}\, D_\mathrm{w}. \tag{11.26}$$

The lower bound on the smallest eigenvalue of $S_{P_{\dot{V}}}$ can be identified as

$$\lambda_\mathrm{m}\left(S_{P_{\dot{V}}}\right) \geq \lambda_\mathrm{m}(D_\mathrm{w}) - \mu\,\lambda_\mathrm{m}(M_\mathrm{w}) - \mu\frac{\|D_\mathrm{w}\|^2}{4\,\lambda_\mathrm{m}(K_\mathrm{w})}. \tag{11.27}$$

As $K_\mathrm{w}, D_\mathrm{w} \succ 0$, we know that $\frac{\|D_\mathrm{w}\|^2}{\lambda_\mathrm{m}(K_\mathrm{w})} > 0$. Therefore, the case $\mu = \frac{\lambda_\mathrm{m}(D_\mathrm{w})}{\lambda_\mathrm{m}(M_\mathrm{w}) + \frac{\|D_\mathrm{w}\|^2}{4\lambda_\mathrm{m}(K_\mathrm{w})}} := \mu_{\dot{\mathrm{V}}}$ determines the lower bound on $\lambda_\mathrm{m}(S_{P_{\dot{V}}})$:

$$\lambda_\mathrm{m}\left(S_{P_{\dot{V}}}\right) > \lambda_\mathrm{m}(D_\mathrm{w}) - \mu_{\dot{\mathrm{V}}}\left(\lambda_\mathrm{m}(M_\mathrm{w}) - \frac{\|D_\mathrm{w}\|^2}{4\,\lambda_\mathrm{m}(K_\mathrm{w})}\right) = 0. \tag{11.28}$$

We conclude, based on the eigenvalue sensitivity theorem of symmetric matrices (Golub and Van Loan, 2013), that $S_{P_{\dot{V}}} \succ 0$ and with that $P_{\dot{V}} \succ 0$ (Boyd and Vandenberghe, 2004). □

Theorem 2. *Let $M_\mathrm{w}, K_\mathrm{w}$ and D_w be positive definite and suppose the system be unforced: $g(u(t)) = 0$. Then, $\tilde{y}_\mathrm{w} = 0$ is globally asymptotically stable for the system dynamics defined (11.10) such that $\dot{V}_\mu(\tilde{y}_\mathrm{w}) < 0, \quad \forall\, \tilde{y}_\mathrm{w} \in \mathbb{R}^{2n} \setminus \{0\}$.*

Proof. First, we show that $\tilde{y}_\mathrm{w} = 0$ is an equilibrium of (11.10):

$$\tilde{f}_\mathrm{w}(0,0) = \begin{bmatrix} 0 \\ M_\mathrm{w}^{-1}\left(-K_\mathrm{w}\,\bar{x}_\mathrm{w} - \tanh(\bar{x}_\mathrm{w} + b)\right) \end{bmatrix} = \begin{bmatrix} 0 \\ M_\mathrm{w}^{-1}\left(-K_\mathrm{w}\,\bar{x}_\mathrm{w} + K_\mathrm{w}\,\bar{x}_\mathrm{w}\right) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{11.29}$$

Lemma 7 states that we can always choose μ such that (11.11) is strict Lyapunov function (e.g., Lipschitz continuous, zero-valued at $\tilde{y}_\mathrm{w} = 0$, positive definite, and radially unbounded) (Khalil, 2002; Wu et al., 2022). We now evaluate the time-derivative of $\dot{V}_\mu(\tilde{y}_\mathrm{w})$ in the case of an unforced system (i.e., $g(u(t)) = 0$):

$$\begin{aligned}
\dot{V}_\mu(\tilde{y}_\mathrm{w}) &= \frac{\partial V_\mu}{\partial \tilde{y}_\mathrm{w}}\,\dot{\tilde{y}}_\mathrm{w} = \frac{\partial V_\mu}{\partial \tilde{y}_\mathrm{w}}\,\tilde{f}_\mathrm{w}(\tilde{y}_\mathrm{w}) = \tilde{y}_\mathrm{w}^\top P_\mathrm{V}\,\dot{\tilde{y}}_\mathrm{w} + \left(\tanh(\bar{x}_\mathrm{w}+\tilde{x}_\mathrm{w}+b)-\tanh(\bar{x}_\mathrm{w}+b)\right)^\top \dot{\tilde{x}}_\mathrm{w} \\
&= -\tilde{y}_\mathrm{w}^\top \underbrace{\begin{bmatrix} \mu K_\mathrm{w} & \frac{1}{2}\mu D_\mathrm{w} \\ \frac{1}{2}\mu D_\mathrm{w}^\top & D_\mathrm{w} - \mu M_\mathrm{w} \end{bmatrix}}_{P_{\dot{\mathrm{V}}}} \tilde{y}_\mathrm{w} - \mu\left(\tanh(\bar{x}_\mathrm{w}+\tilde{x}_\mathrm{w}+b)^\top + \tilde{x}_\mathrm{w}^\top K_\mathrm{w}\right)\tilde{x}_\mathrm{w}, \\
&= -\tilde{y}_\mathrm{w}^\top P_{\dot{\mathrm{V}}}\,\tilde{y}_\mathrm{w} - \mu\left(\tanh(\bar{x}_\mathrm{w}+\tilde{x}_\mathrm{w}+b)-\tanh(\bar{x}_\mathrm{w}+b)\right)^\top \tilde{x}_\mathrm{w}, \\
&\leq -\tilde{y}_\mathrm{w}^\top P_{\dot{\mathrm{V}}}\,\tilde{y}_\mathrm{w} \leq -\lambda_\mathrm{m}\left(P_{\dot{\mathrm{V}}}\right)\|\tilde{y}_\mathrm{w}\|_2^2,
\end{aligned} \tag{11.30}$$

where we exploited the force balance at equilibrium $K_\mathrm{w}\,\bar{x}_\mathrm{w} = -\tanh(\bar{x}_\mathrm{w}+b)$ and Lemma 8 for defining the upper bound on $\dot{V}_\mu(\tilde{y}_\mathrm{w})$. Lemma 9 states that $P_{\dot{\mathrm{V}}} \succ 0$ for $0 < \mu < \mu_{\dot{\mathrm{V}}}$. Similarly, Lemma 7 requests that $0 < \mu < \mu_\mathrm{V}$. Indeed, both conditions can always be fulfilled by choosing $\mu \in \left(0, \min\{\mu_\mathrm{V}, \mu_{\dot{\mathrm{V}}}\}\right)$. With $P_{\dot{\mathrm{V}}} \succ 0 \Leftrightarrow \lambda_\mathrm{m}\left(P_{\dot{\mathrm{V}}}\right) > 0$ (Golub and Van Loan, 2013), we can state that $\dot{V}_\mu(\tilde{y}_\mathrm{w}) < 0\ \forall\ \tilde{y}_\mathrm{w} \in \mathbb{R}^{2n} \setminus \{0\}$ and conclude that the unforced system is globally asymptotically stable around $\tilde{y}_\mathrm{w} = 0$. □

11.2.4 Global Input-to-State Stability (ISS) for the forced system

We now take the forcing $g(u)$ into account again and demonstrate that the system states remain proportionally bounded to the initial conditions and as a function of the supremum of the input forcing. In the following, we will first state two auxiliary Lemmas and subsequently prove Theorem 3 to provides the conditions for ISS.

Lemma 10. *Let $\bar{x}_\mathrm{w}, \tilde{x}_\mathrm{w}, b \in \mathbb{R}^n$. Then,*

$$h_{\mathrm{V},th}(\tilde{x}_\mathrm{w}) = \sum_{i=1}^{n} \int_0^{\tilde{x}_{\mathrm{w},i}} \tanh(\bar{x}_{\mathrm{w},i} + \sigma + b_i)\,\mathrm{d}\sigma - \sum_{i=1}^{n} \int_0^{\tilde{x}_{\mathrm{w},i}} \tanh(\bar{x}_{\mathrm{w},i} + b_i)\,\mathrm{d}\sigma \leq 2\,|\tilde{x}_\mathrm{w}|. \tag{11.31}$$

Proof. Proving $h_{\mathrm{V},th}(\tilde{x}_\mathrm{w}) \leq 2\,|\tilde{x}_\mathrm{w}|$ is equivalent to proving that the scalar function $\breve{h}_{\mathrm{V,th}}(r) = \int_0^r \tanh(\sigma + a)\,\mathrm{d}\sigma - \int_0^r \tanh(a)\,\mathrm{d}\sigma \leq 2\,|r|\ \forall\ r, a \in \mathbb{R}$, where we set $r = \tilde{x}_{\mathrm{w},i}$ and $a = \bar{x}_{\mathrm{w},i} + b_i$. We perform the integration contained in $\breve{h}_{\mathrm{V,th}}(r)$:

$$\begin{aligned}
\breve{h}_{\mathrm{V,th}}(r) &= \int_0^r \tanh(\sigma + a)\,\mathrm{d}\sigma - \int_0^r \tanh(a)\,\mathrm{d}\sigma, \\
\breve{h}_{\mathrm{V,th}}(r) &= \log(\cosh(r+a)) - \log(\cosh(a)) - \tanh(a)\,r.
\end{aligned} \tag{11.32}$$

Next, we demonstrate that the slope of $2\,|r|$ is always larger than the magnitude of the slope of $\breve{h}_{\mathrm{V,th}}(r)$:

$$\left|\frac{\partial \breve{h}_{\mathrm{V,th}}}{\partial r}\right| = |\tanh(r+a) - \tanh(a)| < 2 = \frac{\partial}{\partial r}\,(2\,|r|). \tag{11.33}$$

Additionally, $\breve{h}_{\mathrm{V,th}}(0) = 2\,|0| = 0$. We conclude that $\breve{h}_{\mathrm{V,th}}(r) \leq 2|r|\;\forall\, r \in \mathbb{R}$ and with that, $h_{\mathrm{V},th}(\tilde{x}_{\mathrm{w}}) \leq 2\,|\tilde{x}_{\mathrm{w}}|\;\forall\, \tilde{x}_{\mathrm{w}} \in \mathbb{R}^n$. □

Lemma 11. *Let $M_{\mathrm{w}} \succ 0$ and $K_{\mathrm{w}} \succ 0$. Then,* (11.11) *is bounded by the two scalar, class $\mathcal{K}_\infty$ functions $\alpha_1(r) = \frac{1}{2}\,\lambda_{\mathrm{m}}(P_{\mathrm{V}})\,r^2$ and $\alpha_2(r) = \frac{1}{2}\,\lambda_{\mathrm{M}}(P_{\mathrm{V}})\,r^2 + 2\,\sqrt{n}\,r$: $\alpha_1(\|\tilde{y}_{\mathrm{w}}\|_2^2) \leq V_\mu(\tilde{y}_{\mathrm{w}}) \leq \alpha_2(\|\tilde{y}_{\mathrm{w}}\|)_2^2$.*

Proof. With Lemma 7, we already showed that $V_\mu(\tilde{y}_{\mathrm{w}})$ is a Lyapunov candidate. Now, we additionally also verify the conditions for ISS-Lyapunov candidates (Khalil, 2002).

Step 1: Establishing bounds on $V_\mu(\tilde{y}_{\mathrm{w}})$.
We first identify the lower bound of $V_\mu(\tilde{y}_{\mathrm{w}})$ by leveraging Lemma 6:

$$\begin{aligned} V_\mu(\tilde{y}_{\mathrm{w}}) &= \frac{1}{2}\,\tilde{y}_{\mathrm{w}}^\top P_{\mathrm{V}}\,\tilde{y}_{\mathrm{w}} + \sum_{i=1}^{n}\int_0^{\tilde{x}_{\mathrm{w},i}} \tanh(\bar{x}_{\mathrm{w},i} + \sigma + b_i)\,\mathrm{d}\sigma - \sum_{i=1}^{n}\int_0^{\tilde{x}_{\mathrm{w},i}} \tanh(\bar{x}_{\mathrm{w},i} + b_i)\,\mathrm{d}\sigma, \\ &= \frac{1}{2}\,\tilde{y}_{\mathrm{w}}^\top P_{\mathrm{V}}\,\tilde{y}_{\mathrm{w}} + h_{\mathrm{V,th}}(\tilde{x}_{\mathrm{w}}), \\ &\geq \frac{1}{2}\,\tilde{y}_{\mathrm{w}}^\top P_{\mathrm{V}}\,\tilde{y}_{\mathrm{w}} \;\geq\; \frac{1}{2}\,\lambda_{\mathrm{m}}(P_{\mathrm{V}})\,\|\tilde{y}_{\mathrm{w}}\|_2^2 \;=\; \alpha_1(\|\tilde{y}_{\mathrm{w}}\|_2). \end{aligned} \tag{11.34}$$

Similarly, we derive an upper bound for $V_\mu(\tilde{y}_{\mathrm{w}})$ exploiting Lemma 10.

$$\begin{aligned} V_\mu(\tilde{y}_{\mathrm{w}}) &= \frac{1}{2}\,\tilde{y}_{\mathrm{w}}^\top P_{\mathrm{V}}\,\tilde{y}_{\mathrm{w}} + \sum_{i=1}^{n}\int_0^{\tilde{x}_{\mathrm{w},i}} \tanh(\bar{x}_{\mathrm{w},i} + \sigma + b_i)\,\mathrm{d}\sigma - \sum_{i=1}^{n}\int_0^{\tilde{x}_{\mathrm{w},i}} \tanh(\bar{x}_{\mathrm{w},i} + b_i)\,\mathrm{d}\sigma \\ &\leq \frac{1}{2}\,\lambda_{\mathrm{M}}(P_{\mathrm{V}})\,\|\tilde{y}_{\mathrm{w}}\|_2^2 + 2\,\|\tilde{x}_{\mathrm{w}}\|_1 \;\leq\; \frac{1}{2}\,\lambda_{\mathrm{M}}(P_{\mathrm{V}})\,\|\tilde{y}_{\mathrm{w}}\|_2^2 + 2\,\sqrt{n}\,\|\tilde{x}_{\mathrm{w}}\|_2 \\ &\leq \frac{1}{2}\,\lambda_{\mathrm{M}}(P_{\mathrm{V}})\,\|\tilde{y}_{\mathrm{w}}\|_2^2 + 2\,\sqrt{n}\,\|\tilde{y}_{\mathrm{w}}\|_2 = \alpha_2(\|\tilde{y}_{\mathrm{w}}\|_2). \end{aligned} \tag{11.35}$$

Step 2: Proof that $\alpha_1(r), \alpha_2(r)$ belong to class $\mathcal{K}_\infty$.
According to Lemma 5, $P_{\mathrm{V}} \succ 0$ and with that $\lambda_{\mathrm{m}}(P_{\mathrm{V}}) > 0$. First, we analyze the behavior of $\alpha_1(r)$: as it is strictly increasing and $\alpha_1(0) = 0$, it belongs to class $\mathcal{K}$. Furthermore, we can evaluate $\lim_{r\to\infty} \alpha_1(r) = \infty$. Therefore, $\alpha_1(r) \in \mathcal{K}_\infty$ (Khalil, 2002). $\alpha_2(r)$ is also strictly increasing for $r \in [0, \infty)$, $\alpha_2(0) = 0$, and it is radially unbounded as $\lim_{r\to\infty} \alpha_2(r) = \infty$. For that reason, $\alpha_2(r) \in \mathcal{K}_\infty$ as well. □

Theorem 3. *Suppose $M_{\mathrm{w}}, K_{\mathrm{w}}, D_{\mathrm{w}} \succ 0$, $0 < \theta < 1$, and that we choose $0 < \mu < \min\{\mu_{\mathrm{V}}, \mu_{\dot{\mathrm{V}}}\}$. Then,* (11.10) *is globally Input-to-State Stable (ISS) such that the solution $\tilde{y}_{\mathrm{w}}(t)$ verifies*

$$\|\tilde{y}_{\mathrm{w}}\|_2 \leq \beta(\|\tilde{y}_{\mathrm{w}}(t_0)\|_2, t - t_0) + \gamma\left(\sup_{t_0 \leq t' \leq t} \|g(u(t'))\|_2\right), \qquad \forall\, t \geq t_0 \tag{11.36}$$

where $\beta(r,t) \in \mathcal{KL}$, $\gamma(r) = \sqrt{\frac{(1+\mu^2)\,\lambda_{\mathrm{M}}(P_{\mathrm{V}})\,r^2 + 4\,\theta\,\sqrt{n}\,\sqrt{1+\mu^2}\,\lambda_{\mathrm{m}}(P_{\dot{\mathrm{V}}})\,r}{\theta^2\,\lambda_{\mathrm{m}}(P_{\mathrm{V}})\,\lambda_{\mathrm{m}}^2(P_{\dot{\mathrm{V}}})}} \in \mathcal{K}$.

Proof. **Step 1:** Bounds on ISS-Lyapunov candidate.
Lemma 11 provides the $\mathcal{K}_\infty$ functions $\alpha_1(r) = \frac{1}{2}\,\lambda_{\mathrm{m}}(P_{\mathrm{V}})\,r^2$ and $\alpha_2(r) = \frac{1}{2}\,\lambda_{\mathrm{M}}(P_{\mathrm{V}})\,r^2 + 2\,\sqrt{n}\,r$ such that $\alpha_1(\|\tilde{y}_{\mathrm{w}}\|_2^2) \leq V_\mu(\tilde{y}_{\mathrm{w}}) \leq \alpha_2(\|\tilde{y}_{\mathrm{w}}\|)_2^2$.

Step 2: Minimum energy dissipation.
Let $0 < \mu < \min\{\mu_{\mathrm{V}}, \mu_{\dot{\mathrm{V}}}\}$ as in the proof of Theorem 2. We compute the input-dependent time-derivative of the ISS Lyapunov candidate. We do not repeat the derivations already made as part of (11.30) (e.g., exploiting Lemmas 8 and 9).

$$\begin{aligned}
\dot{V}_\mu(\tilde{y}_{\mathrm{w}}, u(t)) &= -\,\tilde{y}_{\mathrm{w}}^\top P_{\dot{\mathrm{V}}}\,\tilde{y}_{\mathrm{w}} - \mu\,(\tanh(\bar{x}_{\mathrm{w}} + \tilde{x}_{\mathrm{w}} + b) - \tanh(\bar{x}_{\mathrm{w}} + b))^\top \tilde{x}_{\mathrm{w}} + \tilde{y}_{\mathrm{w}}^\top \begin{bmatrix} \mu\, g(u(t)) \\ g(u(t)) \end{bmatrix}, \\
&\leq -\,\lambda_{\mathrm{m}}\left(P_{\dot{\mathrm{V}}}\right)\,\|\tilde{y}_{\mathrm{w}}\|_2^2 + \left\| \tilde{y}_{\mathrm{w}}^\top \begin{bmatrix} \mu\, g(u(t)) \\ g(u(t)) \end{bmatrix} \right\|_1, \\
&\leq -\,\lambda_{\mathrm{m}}\left(P_{\dot{\mathrm{V}}}\right)\,\|\tilde{y}_{\mathrm{w}}\|_2^2 + \|\tilde{y}_{\mathrm{w}}\|_2 \left\| \begin{bmatrix} \mu\, g(u(t)) \\ g(u(t)) \end{bmatrix} \right\|_2, \\
&\leq -\,\lambda_{\mathrm{m}}\left(P_{\dot{\mathrm{V}}}\right)\,\|\tilde{y}_{\mathrm{w}}\|_2^2 + \sqrt{1+\mu^2}\,\|\tilde{y}_{\mathrm{w}}\|_2 \|g(u(t))\|_2,
\end{aligned} \tag{11.37}$$

where we leveraged Hölder's inequality. We choose θ such that $0 < \theta < 1$. As a consequence,

$$\dot{V}_\mu(\tilde{y}_{\mathrm{w}}, u(t)) \leq -(1-\theta)\,\lambda_{\mathrm{m}}\left(P_{\dot{\mathrm{V}}}\right)\,\|\tilde{y}_{\mathrm{w}}\|_2^2, \qquad \forall\, \|\tilde{y}_{\mathrm{w}}\|_2 \geq \frac{\sqrt{1+\mu^2}}{\theta\,\lambda_{\mathrm{m}}\left(P_{\dot{\mathrm{V}}}\right)}\,\|g(u(t))\|_2 > 0. \tag{11.38}$$

We define

$$\alpha_3(r) = (1-\theta)\,\lambda_{\mathrm{m}}\left(P_{\dot{\mathrm{V}}}\right)\,r^2, \qquad \text{and } \rho(r) = \frac{\sqrt{1+\mu^2}}{\theta\,\lambda_{\mathrm{m}}\left(P_{\dot{\mathrm{V}}}\right)}\,r. \tag{11.39}$$

Lemma 9 shows that $\lambda_{\mathrm{m}}\left(P_{\dot{\mathrm{V}}}\right) > 0$. Therefore, $\alpha_3(r)$ is a continuous positive function. Furthermore, as $\mu > 0$, $\rho(r)$ is a strictly increasing for $r \in [0, \infty)$. Additionally with $\rho(0) = 0$ verified, it can be stated that $\rho(r)$ belongs to class $\mathcal{K}$ (Khalil, 2002). We conclude that

$$\dot{V}_\mu(\tilde{y}_{\mathrm{w}}, u(t)) \leq -\alpha_3\left(\|\tilde{y}_{\mathrm{w}}\|_2^2\right), \qquad \forall\, \|\tilde{y}_{\mathrm{w}}\|_2 \geq \rho(\|g(u(t))\|_2) > 0. \tag{11.40}$$

Step 3: Conclusions.

As a result of Steps 1 and 2, the system is input-to-state stable, and with that, the solution $\tilde{y}_{\mathrm{t}}$ satisfies (Khalil, 2002)

$$\|\tilde{y}_{\mathrm{w}}\|_2 \leq \beta(\|\tilde{y}_{\mathrm{w}}(t_0)\|_2, t - t_0) + \gamma\left(\sup_{t_0 \leq t' \leq t} \|g(u(t'))\|_2\right), \tag{11.41}$$

with

$$\gamma(r) = \alpha_1^{-1} \circ \alpha_2 \circ \rho(r) = \sqrt{\frac{(1+\mu^2)\,\lambda_{\mathrm{M}}(P_{\mathrm{V}})\,r^2 + 4\,\theta\,\sqrt{n}\,\sqrt{1+\mu^2}\,\lambda_{\mathrm{m}}(P_{\dot{\mathrm{V}}})\,r}{\theta^2\,\lambda_{\mathrm{m}}(P_{\mathrm{V}})\,\lambda_{\mathrm{m}}^2(P_{\dot{\mathrm{V}}})}}. \tag{11.42}$$

Indeed, based on Theorem 2 and the associated proof, we can easily verify that $\gamma(r)$ is strictly increasing for $r \in [0, \infty)$ and that $\gamma(0) = 0$. As a consequence, $\gamma(r) \in \mathcal{K}$ (Khalil, 2002). □

11.3 An Approximate Closed-Form Solution to the CON Dynamics

11.3.1 Approach

To predict future system states, we need to integrate the ODE in Eq. (11.2), with the solution given by $y(t_{k+1}) = y_{t_k} + \int_{t_k}^{t_{k+1}} f(y(t'), u(t'))\,\mathrm{d}t'$. Unfortunately, a closed-form solution for the nonlinear dynamics $f(y,u)$ does not (yet) exist. Therefore, we traditionally need to revert to (high-order) numerical ODE solvers that are computationally very expensive and introduce additional memory overhead (Kidger, 2021). This considerably increases the training time of models involving such continuous-time dynamics. While the computational time can be reduced by increasing the (minimum) time step of the integrator, this comes at the expense of an integration error, and we lose (part of) the theoretical guarantees and practical characteristics that the nominal ODE provides. In this work, we take an alternative approach by splitting the problem into (i) decoupled linear dynamics that can be cheaply and precisely integrated using a closed-form solution and (ii) the residual, coupled nonlinear dynamics, which we integrate numerically at a slower time scale:

$$\ddot{x}(t) = \underbrace{F - \kappa x(t) - d\,\dot{x}(t)}_{f_{\ddot{x},\mathrm{ld}}(y):\ \text{decoupled, linear dynamics}} + \underbrace{g(u(t) - (K-\kappa)x(t) - (D-d)\,\dot{x}(t) - \tanh(W x(t) + b)}_{f_{\ddot{x},\mathrm{nld}}(y,u):\ \text{coupled, nonlinear dynamics}} \tag{11.43}$$

where $\kappa = \mathrm{diag}(K_{11} \dots K_{nn})$, $d = (D_{11} \dots D_{nn})$ are the diagonal components of the stiffness and damping matrices, respectively, and $F \in \mathbb{R}^n$ is a constant, external forcing term on the oscillators.

For a short-time-interval δt, we now approximate (11.43) as

$$\ddot{x}(t_k + \delta t) \approx f_{\ddot{x},\mathrm{ld}}(y(t_k + \delta t), F(t_k)) \qquad \text{with } F(t_k) = -f_{\ddot{x},\mathrm{nld}}(y(t_k), u(t_k)). \tag{11.44}$$

For a scalar 2nd-order, linear ODE of the form $\dot{y}_i = f_{\mathrm{ld},i}(x(t'), F(t_k))$, a well-known, closed-form solution (Pas, 2023) exists. We exploit this characteristic by formulating the approximate solution as

$$y(t_k + \delta t) \approx f_{\mathrm{CFA-CON}}(y(t_k), u(t_k)) = y(t_k) + \int_{t_k}^{t_k + \delta t} f_{\mathrm{ld}}(y(t'), F(t_k))\,\mathrm{d}t' \tag{11.45}$$

and denote $f_{\mathrm{CFA-CON}} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ as the Closed-Form Approximation of CON (CFA-CON) model. The implicit assumption behind (11.44) is that $f_{\ddot{x},\mathrm{ld}}(y) \succ f_{\ddot{x},\mathrm{nld}}(y,u)$ (i.e., the linear, decoupled dynamics dominate the nonlinear, coupled, time-varying dynamics).

11.3.2 Closed-Form Solution to a Forced Harmonic Oscillator

As introduced in (11.1), we consider the linear dynamics of a 1D forced harmonic oscillator with state $y_i = \begin{bmatrix} x_i & \dot{x}_i \end{bmatrix} \in \mathbb{R}^2$

$$\dot{y}_i = \begin{bmatrix} \frac{\mathrm{d}x_i}{\mathrm{d}t} \\ \frac{\mathrm{d}\dot{x}_i}{\mathrm{d}t} \end{bmatrix} = f_{\mathrm{ld},i}(y_i, F_i) = \begin{bmatrix} \dot{x}_i \\ F_i(t) - \kappa_i\, x_i(t) - d_i\, \dot{x}_i(t) \end{bmatrix}, \tag{11.46}$$

where $F_i(t) \in \mathbb{R}$ is the externally applied force acting on the oscillator.

The characteristic equation for the unforced dynamics (i.e., $F_i(t) = 0$) can be stated as (Pas, 2023)

$$\lambda^2 + 2\,\zeta_i\,\omega_{\mathrm{n},i}\,\lambda + \omega_{\mathrm{n},i}^2 = 0, \qquad \text{with the solutions } \lambda_{1,2} = -\zeta_i\,\omega_{\mathrm{n},i} \pm \omega_{\mathrm{n},i}\sqrt{\zeta_i^2 - 1}, \tag{11.47}$$

where $\omega_{\mathrm{n},i} = \sqrt{\kappa_i}$ and $\zeta_i = \frac{d_i}{2\sqrt{\kappa_i}}$ are the natural frequency and the damping factor of the ith homogeneous oscillator, respectively. This harmonic oscillator exhibits three regimes: underdamped ($\zeta_i < 1$), critically damped ($\zeta_i = 1$), and overdamped regime ($\zeta_i > 1$).

We approximate the forcing using the Heavyside function $H(t)$: $F_i(t) = F_i(t_k)\,H(t)$, where $F_i(t_k)$ is the constant external forcing as computed by (11.44). The solution for $\zeta_i \neq 1$ is given by (Pas, 2023)

$$y_i(t_{k+1}) = \begin{bmatrix} x_i(t_{k+1}) \\ \dot{x}_i(t_{k+1}) \end{bmatrix} = \begin{bmatrix} \left(c_{1,i}\cos(\beta_i\,\delta t) + c_{2,i}\sin(\beta_i\,\delta t)\right)\,e^{-\alpha_i\,\delta t} + \frac{F_i}{\kappa_i} \\ -\left((c_{1,i}\alpha_i - c_{2,i}\beta_i)\cos(\beta_i\,\delta t) + (c_{1,i}\beta_i + c_{2,i}\alpha_i)\sin(\beta_i\,\delta t)\right)\,e^{-\alpha_i\,\delta t} \end{bmatrix}, \tag{11.48}$$

where $\delta t = t_{k+1} - t_k$, $\alpha_i = \zeta_i\,\omega_{\mathrm{n},i}$, and $\beta_i = \omega_{\mathrm{n},i}\sqrt{1-\zeta_i^2}$. After enforcing the initial conditions $x_i(t_k), x_i(t_k)$, the integration constants

$$c_{1,i} = x_i(t_k) - \frac{F_i(t_k)}{\kappa_i}, \qquad c_{2,i} = -2\,j\,\frac{\dot{x}_i(t_k) + \alpha_i\,\left(x_i(t_k) - \frac{F_i}{k_i}\right)}{\Delta\lambda_i}, \tag{11.49}$$

can be identified with $\Delta\lambda_i = \lambda_{i,2} - \lambda_{i,1} = -2\,\beta_i\,j$, where j is the imaginary value. While we could derive the solution for the critically damped case $d_i < 2\sqrt{\kappa_i}$ separately, we instead approximate it in our network dynamics with (11.48) by setting $\Delta\lambda_i \cong \begin{cases} \mathrm{sign}(-2\,\beta_i\,j)\,\epsilon & |2\,\beta_i\,j| < \epsilon \\ -2\,\beta_i\,j & |2\,\beta_i\,j| \geq \epsilon \end{cases}$, where $\epsilon \in \mathbb{R}^+ \ll 1$ is a small, positive value.

11.3.3 Algorithmic Implementation

We can now leverage the closed-form solution to the evolution of a single, decoupled damped harmonic oscillator of (11.48) to solve the integral in (11.45)

$$\begin{aligned} y(t_{k+1}) &\approx f_{\text{CFA-CON}}(y(t_k), u(t_k)), \\ \begin{bmatrix} x(t_{k+1}) \\ \dot{x}(t_{k+1}) \end{bmatrix} &\approx \begin{bmatrix} (c_1 \odot \cos(\beta\,\delta t) + c_2 \odot \sin(\beta\,\delta t)) \odot e^{-\alpha\,\delta t} + \frac{F(t_k)}{\kappa} \\ -((c_1 \odot \alpha - c_2 \odot \beta)\cos(\beta\,\delta t) + (c_1 \odot \beta + c_2 \odot \alpha)\sin(\beta\,\delta t)) \odot e^{-\alpha\,\delta t} \end{bmatrix}, \end{aligned} \tag{11.50}$$

with

$$\begin{aligned} \kappa = \mathrm{diag}(K_{11} \ldots K_{nn}), \quad d = (D_{11} \ldots D_{nn}), \quad \omega_{\mathrm{n}} = \sqrt{\kappa}, \quad \zeta = \frac{d}{2\sqrt{\kappa}}, \\ F(t_k) = g(u(t_k)) - (K - \kappa)\,x(t_k) - (D - d)\,\dot{x}(t_k) - \tanh(W x(t_k) + b), \\ \delta t = t_{k+1} - t_k, \qquad \alpha = \zeta \odot \omega_{\mathrm{n}}, \qquad \beta = \omega_{\mathrm{n}}\sqrt{1-\zeta^2}, \\ c_1 = x(t_k) - \frac{F(t_k)}{\kappa}, \qquad c_2 = \frac{1}{\beta}\left(\dot{x}(t_k) + \alpha \odot \left(x(t_k) - \frac{F(t_k)}{\kappa}\right)\right). \end{aligned} \tag{11.51}$$

We summarize the approach of integrating/rolling out the CFA-CON dynamics in Algorithm 3.

Algorithm 3 Rollout of CFA-CON.

Inputs: initial state $y(t_0)$, input sequence $\{u(t_0), \dots u(t_k), \dots u(t_N)\}$
Outputs: state sequence $\{y(t_0), \dots y(t_k), \dots y(t_N)\}$

$k \leftarrow 0$
while $k \leq N$ **do**
  $(x(t_k), \dot{x}(t_k)) \leftarrow y(t_k)$
  $F(t_k) \leftarrow g(u(t_k)) - (K - \kappa)\, x(t_k) - (D - d)\, \dot{x}(t_k) - \tanh(W x(t_k) + b)$
  $\omega_{\mathrm{n}}, \zeta \leftarrow \sqrt{\kappa}, \frac{d}{2\sqrt{\kappa}}$ ▷ Compute the characteristics of the decoup. harm. oscillators.
  $\alpha, \beta \leftarrow \zeta \odot \omega_{\mathrm{n}}, \omega_{\mathrm{n}} \sqrt{1 - \zeta^2}$
  $c_1 \leftarrow x(t_k) - \frac{F(t_k)}{\kappa}$ ▷ Compute integration constants using initial conditions.
  $c_2 \leftarrow \frac{1}{\beta} \left(\dot{x}(t_k) + \alpha \odot \left(x(t_k) - \frac{F(t_k)}{\kappa} \right) \right)$
  $\delta t = t_{k+1} - t_k$ ▷ Set time step.
  ▷ Update state with approximated closed-form solution.
  $x(t_{k+1}) \leftarrow (c_1 \odot \cos(\beta\, \delta t) + c_2 \odot \sin(\beta\, \delta t)) \odot e^{-\alpha\, \delta t} + \frac{F}{\kappa}$
  $\dot{x}(t_{k+1}) \leftarrow -((c_1 \odot \alpha - c_2 \odot \beta) \cos(\beta\, \delta t) + (c_1 \odot \beta + c_2 \odot \alpha) \sin(\beta\, \delta t)) \odot e^{-\alpha\, \delta t}$
  $k \leftarrow k + 1$ ▷ Update time index.
end while

11.3.4 Approximation Bounds for CFA-CON

Lemma 12 demonstrates how, for the particular case of no external input and linearly decoupled oscillators (which we are always free to choose), we can establish bounds on the approximation error when using the closed-form solution instead of the ground-truth coupled oscillator dynamics.

Lemma 12. *Suppose that the network is unforced with $g(u(t)) = 0$ and that $K = \mathrm{diag}(\kappa_1, \dots, \kappa_n)$, $D = \mathrm{diag}(d_1, \dots, d_n)$ such that the oscillators are not linearly coupled. Then, given any $t \geq 0$, and the initial state $y(0) \in \mathbb{R}^{2n}$, the error between the continuous dynamics $\ddot{x}(t)$ of (11.2) and the approximated dynamics $\hat{\ddot{x}}(t)$ in (11.44), is bounded by $\|\ddot{x}(t) - \hat{\ddot{x}}(t)\| \leq 2$.*

Proof. (11.2), (11.44) and $F = -f_{\ddot{x},\mathrm{nld}}(y(0), 0)$ give us

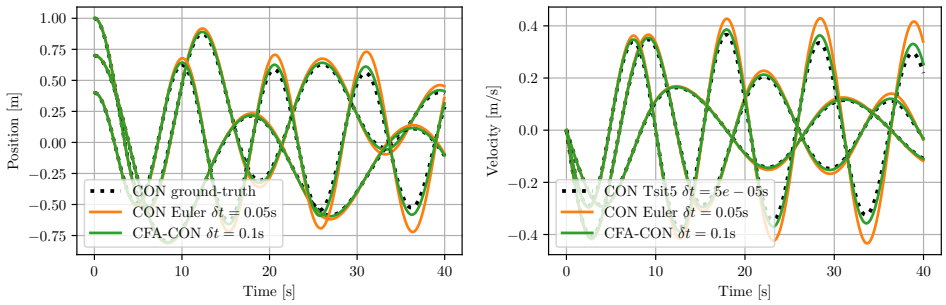
$$\begin{aligned} \|\ddot{x}(t) - \hat{\ddot{x}}(t)\| &= \left(f_{\ddot{x},\mathrm{ld}}(y(t), 0) + f_{\ddot{x},\mathrm{nld}}(y(t), 0) \right) - f_{\ddot{x},\mathrm{ld}}(y(t), F), \\ &= -Kx(t) - D\dot{x} - \tanh(Wx(t) + b) + Kx(t) + D\dot{x} + \tanh(Wx(0) + b), \\ &= \|-\tanh(Wx(t) + b) + \tanh(Wx(0) + b)\| \leq 2. \end{aligned} \tag{11.52}$$

□

11.3.5 Empirical Evaluation of Approximation Error

In Table 11.1, we present a comparison of CFA-CON with several other strategies for integrating nonlinear dynamics, such as CON. Following the implicit assumption made in the concept for the closed-form approximation (i.e., Sec. 11.3.1), we consider the case of $g(u) = 0$, $K = \mathrm{diag}(\kappa_1, \dots, \kappa_n)$ and $D = \mathrm{diag}(d_1, \dots, d_n)$ but with the hyperbolic coupling between the oscillators active (i.e., a full W matrix). Integrating the dynamics at a very

small time step (i.e., $\delta t = 5 \times 10^{-5}$ s) with a high-order ODE solver would give us a very accurate solution, but this is computationally infeasible in practice. We, therefore, regard this as the upper bound on the accuracy of the solution. A feasible solution would be to implement either a high-order solver such as Tsit5 at a larger integration time-step, e.g., $\delta t = 1 \times 10^{-1}$ s) or a low-order solver with a slightly smaller integration time step, e.g., $\delta t = 5 \times 10^{-2}$ s). Therefore, we also benchmark these options. We also benchmark an implementation specialized on the underdamped case (i.e., $\zeta_i < 1$): Closed-Form Approximation of Underdamped CON (CFA-UDCON). This specialized implementation allows us to avoid using complex numbers in the algorithm and reduces the number of computations necessary for calculating the approximated solution. As a result, we see a considerable increase in the sim-time to real-time factor.



(a) Positions (b) Velocities

Figure 11.3: Analysis of approximation error of CFA-CON: we compare the ground-truth solution of a 40 s rollout of the CON network consisting of three oscillators ($n = 3$) with the *CFA-CON* executed at a time step of $\delta t = 0.1$ s and a solution generated by integrating the ODE at a time step of $\delta t = 0.05$ s with the Euler method.

Integration error

We perform the integration error benchmark over 100 different network configurations, all consisting of 50 oscillators ($n = 50$): First, we sample the natural frequency of the ith oscillator from a uniform distribution as $\omega_{\mathrm{n},i} \sim \mathcal{U}(0.05\,\mathrm{Hz}, 0.5\,\mathrm{Hz})$, then we sample $\kappa_i \sim \mathcal{U}(0.2\,\mathrm{N/m}, 2\,\mathrm{N/m})$ such that $K = \mathrm{diag}(\kappa_1, \ldots, \kappa_n) \succ 0$, which lets us determine each mass $m_i = \frac{\kappa_i}{\omega_{\mathrm{n},i}^2}$. Next, the damping ratio is determined as $\zeta_i \sim \mathcal{U}(0.1, 0.9)$ and $\zeta_i \sim \mathcal{U}(0.1, 2.0)$ for the underdamped and general case, respectively. As a result, $D = \mathrm{diag}(d_1, \ldots, d_n) \succ 0$ with $d_i = 2\,\zeta_i\,\sqrt{m_i\,\kappa_i}$ given. Finally, by leveraging the Cholesky decomposition, we sample a $W \succ 0$ and $b_i \sim \mathcal{U}(-1, 1)$. We compute the estimation error of all integrated trajectories with respect to the high-precision solution (i.e., Tsitouras' 5/4 method (Tsit5) at $\delta t = 5 \times 10^{-4}$ s). For this, we compute the RMSE for each 60 s trajectory and then take the mean and standard deviation across the 100 different network configurations.

Simulation-Time to Real-Time Factor

The simulation vs. real-time factor is computed as the simulated rollout duration per second of computational time. For this, we let each method simulate a 60 s trajectory for

100 times and record the minimum run time on an Intel Core i7-10870H CPU (single core) over 10 trials. Because of computational constraints, we simulated the trajectory with the high-precision Tsit5 solver only 5 times.

Results

We also provide qualitative results for the integration accuracy in Fig. 11.3 The results in Table 11.1 show that CFA-CON is 30 % more accurate than the Euler integrator at half of the speed. Comparing against the Tsit5 integrator, CFA-CON exhibits a 1.56x speed increase while being significantly less accurate. For the underdamped case with $\zeta < 1$, the specialized implementation CFA-UDCON is 14.8 % faster and at the same time 32 % more accurate than the Euler integrator. Furthermore, CFA-UDCON is 3.7x faster and significantly less accurate than the Tsit5 integrator. We can conclude that in the pure rollout setting (i.e., no backpropagation involved) for a generic CON, the CFA-CON does not show clear advantages to an appropriately tuned Euler or Tsit5 solver. However, the specialized version CFA-UDCON demonstrates a 2.4x speed-up at no reduction of accuracy vs. CFA-CON for underdamped oscillator networks.

Method	**RMSE** [m] ↓	**RMSE** $\zeta < 1$ [m] ↓	**Complexity** ↓	$\frac{\textbf{Sim. time}}{\textbf{Real time}}$ ↑
CON with Tsit5 at $\delta t = 5\times10^{-5}$ s	n/a	n/a	$\mathcal{O}\left(\frac{n^{\log_2 7} ph}{\delta t}\right) = \mathcal{O}(3.5\times10^{11})$	5.68x
CON with Tsit5 at $\delta t = 1\times10^{-1}$ s	$5\times10^{-5} \pm 1\times10^{-5}$	$8\times10^{-6} \pm 1\times10^{-5}$	$\mathcal{O}\left(n\frac{^{\log_2 7} ph}{\delta t}\right) = \mathcal{O}(1.8\times10^{8})$	11310x
CON with Euler at $\delta t = 5\times10^{-2}$ s	0.010 ± 0.003	0.022 ± 0.005	$\mathcal{O}\left(\frac{n^{\log_2 7} h}{\delta t}\right) = \mathcal{O}(7.1\times10^{7})$	36500x
CFA-CON (our) with $\delta t = 1\times10^{-1}$ s	0.007 ± 0.002	0.015 ± 0.003	$\mathcal{O}\left(\frac{n^{\log_2 7} h}{\delta t}\right) = \mathcal{O}(3.5\times10^{7})$	17680x
CFA-UDCON (our) with $\delta t = 1\times10^{-1}$ s	n/a	0.015 ± 0.003	$\mathcal{O}\left(\frac{n^{\log_2 7} h}{\delta t}\right) = \mathcal{O}(3.5\times10^{7})$	**41900x**

Table 11.1: Benchmarking of various methods for integrating the CON dynamics. The RMSE is computed with respect to the Tsitouras' 5/4 method (Tsit5) (i.e., extremely high accuracy but also extremely high computational complexity). We denote with n the number of oscillators in the network (in this case $n = 50$), with p the order of the numerical ODE solver, and with δt the time-step. When stating the complexity, we refer to $h = t_N - t_0$ as the rollout horizon in seconds. In this case, we report the results for a horizon of $h = 60$ s. The RMSE column states the RMSE of the various integration strategies with respect to the *CON with Tsit5 at* $\delta t = 5\times10^{-5} s$ solution, which we consider to be the ground-truth. The *RMSE* $\zeta < 1$ computes the same metrics, but this time for a dataset that contains only underdamped oscillators. The $\frac{\textit{Sim. time}}{\textit{Real time}}$ column states the ratio between the duration of the simulation achieved (in seconds) per second of real-time (i.e., computational time). We report the mean and standard deviation of the RMSE over 100 different network configurations.

11.4 Learning Control-Oriented Latent Dynamics from Pixels

We now move towards learning latent dynamical models based on CON and CFA-CON. CONs are an ideal fit for learning latent dynamics as they guarantee that the latent states stay bounded.

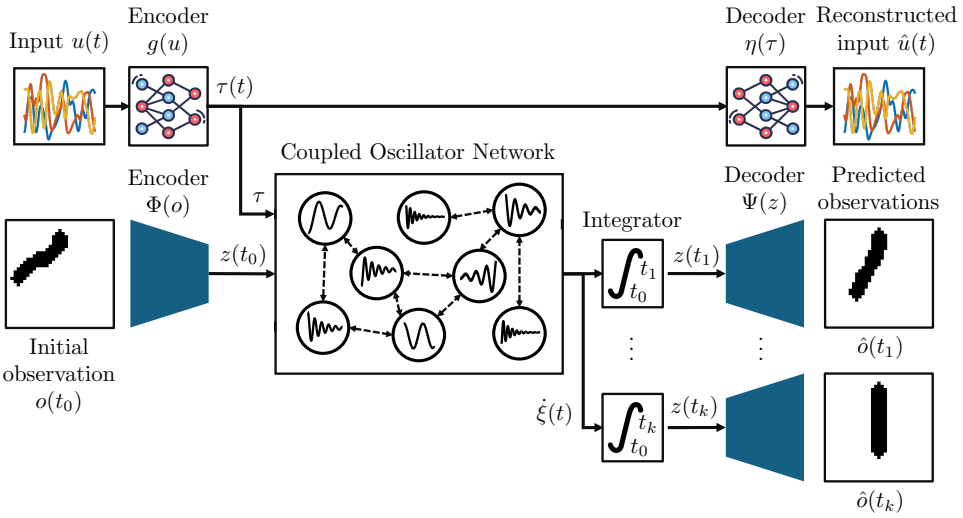


Figure 11.4: Exploiting CONs for learning latent dynamics from pixels: We encode the initial observation $o(t_0)$ and the input $u(t)$ into latent space where we leverage the CON to predict future latent states. Finally, we decode both the latent-space torques $\tau(t)$ and the predicted latent states $z(t)$.

We assume to have access to observations in the form of images $o \in \mathbb{R}^{h_o \times w_o \times c_o}$, where c_o denotes the number of channels. Please note that this could also be other high-dimensional observations such as LiDAR scans, point clouds, etc. We now leverage an encoder-decoder architecture to map these high-dimensional observations into a compressed latent space: The encoder $\Phi : \mathbb{R}^{h_o \times w_o \times c_o} \to \mathbb{R}^{n_z}$ with $n_z \ll h_o\, w_o$ identifies a low-dimensional latent representation $z \in \mathbb{R}^{n_z}$ of the images. The decoder $\Psi : \mathbb{R}^{n_z} \to \mathbb{R}^{h_o \times w_o \times c_o}$ approximates the inverse operation by reconstructing an image $\hat{o} \in \mathbb{R}^{h_o \times w_o \times c_o}$ based on the latent representation. To promote the learning of a smooth and monotonic mapping into latent space, we specifically choose to implement the autoencoder here as a β-VAE (Higgins et al., 2017; Kingma and Welling, 2014). Instead of just statically reconstructing the image $\hat{o}(t_k)$, we are interested in predicting future observations $\hat{o}(t_{k+l})$, where $l \in 1 \ldots N$. For this, we train a 2^{nd}-order dynamical model that is, when integrated, able to predict future latent representations $z(t_{k+l})$. This requires us to define a latent state $\xi(t) = \begin{bmatrix} z^\top(t) & \dot{z}^\top(t) \end{bmatrix}^\top \in \mathbb{R}^{2n_z}$ consisting of the latent representation and latent velocity $\dot{z}(t) \in \mathbb{R}^{n_z}$.

We now rely on CON with $n = n_z$ oscillators to provide us with the latent state derivative $\dot{\xi} = f_w(z(t), u(t))$, where we defined $\xi = y_w$, and $z = x_w$. To ensure stability, we make use of the Cholvesky decomposition to ensure that M_w, K_w and D_w always remain positive definite (see Theorem 3). It is important to note that we train the encoder, decoder, and dynamical model all jointly. Please refer to Appendix B.1 for more implementation details.

11

11.4.1 Training

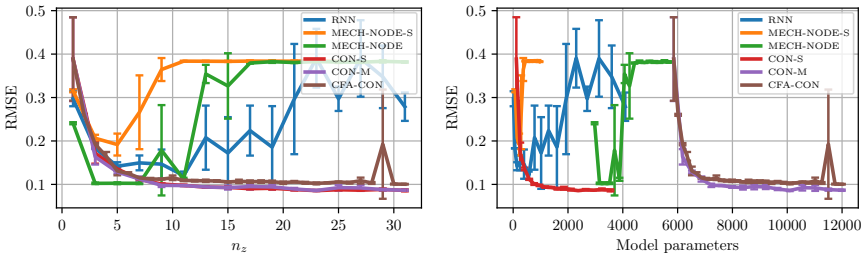
It is important to remember that because we are using a β-VAE (Higgins et al., 2017; Kingma and Welling, 2014), the image encoding becomes stochastic, and the encoder neural network actually outputs $\mu_z(o), 2\log(\sigma_z)(o) \in \mathbb{R}^{n_z}$. After executing the reparametrization trick as

Model	RMSE M-SP+F ↓	RMSE S-P+F ↓	RMSE D-P+F ↓	RMSE CS ↓	RMSE PCC-NS-2 ↓	RMSE PCC-NS-3 ↓	RMSE R-D ↓
RNN	0.2739 ± 0.0057	0.2378 ± 0.0352	0.1694 ± 0.0004	$\mathbf{0.1011 \pm 0.0009}$	0.1373 ± 0.0185	0.2232 ± 0.0075	0.3763 ± 0.0374
GRU (Cho et al., 2014)	0.0267 ± 0.0033	0.1457 ± 0.0078	0.1329 ± 0.0005	0.1125 ± 0.0100	$\mathbf{0.0951 \pm 0.0021}$	0.2148 ± 0.0196	0.3232 ± 0.0368
coRNN (Rusch and Mishra, 2020)	$\mathbf{0.0265 \pm 0.0002}$	0.1333 ± 0.0044	0.1324 ± 0.0016	0.2537 ± 0.0018	0.2504 ± 0.0899	0.2474 ± 0.0018	$\mathbf{0.0741 \pm 0.0001}$
NODE (Chen et al., 2018)	$\mathbf{0.0264 \pm 0.0010}$	$\mathbf{0.1260 \pm 0.0013}$	0.1324 ± 0.0024	0.2415 ± 0.0021	0.1867 ± 0.0561	0.3373 ± 0.0565	$\mathbf{0.0738 \pm 0.0007}$
MECH-NODE	0.0328 ± 0.0034	0.1650 ± 0.0205	0.1710 ± 0.0111	0.2494 ± 0.0028	0.1035 ± 0.0012	0.1900 ± 0.0024	N/A
CON-S (our)	0.0303 ± 0.0053	0.1303 ± 0.0064	0.1323 ± 0.0018	0.1993 ± 0.0646	0.0996 ± 0.0012	0.1792 ± 0.0038	0.1110 ± 0.0160
CON-M (our)	0.0303 ± 0.0053	0.1303 ± 0.0064	0.1323 ± 0.0018	0.1063 ± 0.0027	0.1008 ± 0.0006	$\mathbf{0.1785 \pm 0.0023}$	0.1110 ± 0.0160
CFA-CON (our)	0.0313 ± 0.0026	0.1352 ± 0.0073	$\mathbf{0.1307 \pm 0.0012}$	0.1462 ± 0.0211	0.1124 ± 0.0025	0.1803 ± 0.0003	0.1068 ± 0.0059

Table 11.2: Benchmarking of CON and CFA-CON at learning latent dynamics against baseline methods. The first three datasets, based on (Botev et al., 2021), contain samples of a mass-spring with friction (*M-SP + F*), a single pendulum with friction (*S-P + F*), and a double pendulum with friction (*D-P + F*) (all without system inputs). The *CS* dataset considers a continuum soft robot consisting of one segment with three constant planar strains. The *PCC-NS-2* and *PCC-NS-3* datasets contain trajectories of a continuum soft robot made of two and three piecewise constant curvature segments, respectively. We choose the latent dimensions of the models as $n_z = 4$, $n_z = 4$, and $n_z = 12$ for the *M-SP + F*, *S-P + F*, and *D-P + F* datasets, and $n_z = 8$, $n_z = 12$, and $n_z = 12$ for the *PCC-NS-2*, *PCC-NS-3*, and *CS* soft robotic datasets. We report the mean and standard deviation over three different random seeds.

Dataset	n_z	**RNN**	**GRU** (Cho et al., 2014)	**coRNN** (Rusch and Mishra, 2020)	**NODE** (Chen et al., 2018)	**MECH-NODE**	**CON-S (our)**	**CON-M (our)**	**CFA-CON (our)**
M-SP+F (Botev et al., 2021)	4	88	248	40	3368	3244	**34**	**34**	**34**
S-P+F (Botev et al., 2021)	4	88	248	40	3368	3244	**34**	**34**	**34**
D-P+F (Botev et al., 2021)	12	672	1968	348	4404	4032	**246**	**246**	**246**
CS	12	696	2040	**336**	4374	4002	1386	8568	8568
PCC-NS-2	8	320	928	**152**	3856	3062	676	7048	7048
PCC-NS-3	12	696	2040	**336**	4374	4002	1386	8568	8568
R-D (Champion et al., 2019)	4	**20**	52	**20**	3064	-	24	24	24

Table 11.3: Number of trainable parameters for the each latent dynamic models depending on the respective dataset: mass-spring with friction (*M-SP+F*), single pendulum with friction (*S-P+F*), double pendulum with friction (*D-P+F*), a constant strain soft robot (*CS*), a piecewise constant curvature with two segments (*PCC-NS-2*, a piecewise constant curvature soft robot with three segments (*PCC-NS-3*, and reaction-diffusion (*R-D*).



(a) RMSE vs. latent dimension n_z (b) RMSE vs. model parameters

Figure 11.5: Evaluation of prediction performance of the various models vs. the dimension of their latent representation n_z and the number of trainable parameters of the dynamics model, respectively, on the *PCC-NS-2* dataset. All hyperparameters are tuned for each model separately for $n_z = 8$. The error bar denotes the standard deviation across three random seeds.

$z(t_k) \sim \mathcal{N}(\mu_z(t_k), \sigma_z^2(t_k))$, we formulate the loss function, evaluated on each trajectory consisting of N time-steps, as

$$\begin{aligned}\mathcal{L} = \sum_{k=0}^{N} \Bigg(& \underbrace{\frac{\mathrm{MSE}(o(t_k), \Psi(z(t_k)))}{N+1}}_{\text{Static image reconstruction loss}} + \beta \underbrace{\frac{\mathcal{D}_{\mathrm{KL}}\left((\mu_z(t_k), \sigma_z(t_k)\right)}{N+1}}_{\text{Kullback–Leibler divergence}} \Bigg) \\ + \sum_{k=1}^{N} \Bigg(& \lambda_{\vec{o}} \underbrace{\frac{\mathrm{MSE}(o(t_k), \Psi(\hat{z}(t_k)))}{N}}_{\text{Dynamic image reconstruction loss}} + \lambda_z \underbrace{\frac{\mathrm{MSE}(z(t_k), \hat{z}(t_k))}{N}}_{\text{Latent dynamics consistency loss}} \Bigg),\end{aligned} \tag{11.53}$$

where $\hat{z}(t_k)$ is predicted by $\hat{\xi}(t_k) = \int_{t=t_0}^{t_k} f_\xi(\xi(t'), u(t'))\,\mathrm{d}t'$, and $\xi(t_0) = \begin{bmatrix} z^\top(t_0) & \dot{z}^\top(t_0) \end{bmatrix}^\top$. Here, $z(t_0)$ is given by the encoder, and $\dot{z}(t_0)$ is approximated using finite differences in image-space. $\beta, \lambda_{\vec{o}}, \lambda_z \in \mathbb{R}$ are loss weights.

Estimation of Initial Latent Velocity

For 2$^{\text{nd}}$-order systems and when integrating the evolution of the latent state $\xi(t) = \begin{bmatrix} z^\top(t) & \dot{z}^\top(t) \end{bmatrix}^\top$ in time, we need to have access to an initial latent velocity $\dot{z}(t_0)$ such that we can roll out the latent state $\xi(t)$ in time. A naive approach to estimating such an initial latent velocity would be to encode multiple (at least two) images of the system at the start of the trajectory into latent space and then perform numerical differentiation (e.g., finite differences) in latent space. However, we found the resulting $\dot{z}(t_k)$ to be relatively noisy and susceptible to small encoding errors. Instead, we propose to perform numerical differentiation in image space and then map this velocity into latent space using the encoder's Jacobian. First, we estimate the image-space velocity at t_k using finite differences: $\dot{o}(t_k) \approx \frac{o(t_{k+1}) - o(t_{k-1})}{t_{k+1} - t_{k-1}}$. The latent velocity is then estimated as $\dot{z}(t_k) = \frac{\partial \Phi}{\partial o}(o(t_k))\,\dot{o}(t_k)$, where $\frac{\partial \Phi}{\partial o}$ is obtained with forward-mode automatic differentiation.

11.4.2 Models

We train the CON with the input-to-forcing mapping $g(u) = B(u)\,u$, where $B(u)$ is parametrized by a MLP with a hyperbolic tangent activation function applied in between layers. We report results for two variants of the CON model: for the medium-sized *CON-M* and small-sized *CON-S*, the MLP consists of five and two layers with a hidden dimension of 30 and 12, respectively. The model CFA-CON uses the same architecture as *CON-M*. We compare against several popular latent space model architectures: The NODE model uses a MLP with an hyperbolic activation functions and predicts $\dot{\xi}(t) = f_{\text{NODE}}(\xi(t), u(t))$. To make the comparison fair, we parametrize the NODE's MLP in the same fashion as for *CON-M*. The *MECH-NODE* integrates prior knowledge towards learning 2^{nd}-order mechanical ODEs and, therefore, predicts $\ddot{z}(t) = f_{\text{MECH-NODE}}(\xi(t), u(t))$. Furthermore, we consider multiple autoregressive models: RNN, GRU, and coRNN and let them parameterize the following transition function: $\xi(t_{k+1}) = f_{\text{ar}}(\xi(t_k), u(t_k))$ As common in the relevant literature (Botev et al., 2021), we allow the autoregressive models to perform multiple time step transitions before predicting the next sample. For the autoencoder, we use a vanilla Convolutional Neural Network (CNN). More details can be found in Appendix B.1.

11.4.3 Datasets

We consider in total seven datasets that are based on simulations of unactuated mechanical systems, actuated continuum soft robots, and the reaction-diffusion dynamics. The first three, mechanical datasets are based on the work of Botev et al. (2021) and contain video sequences of a mass-spring system with friction (*M-SP+F*), a single pendulum with friction (*S-P+F*), and a double pendulum with friction (*D-P+F*). Continuum soft robots have theoretically infinite DOF, evolve with highly nonlinear and often time-dependent dynamical behaviors, and are notoriously difficult to model from first principles (Armanini et al., 2023). For that reason, it is a very interesting proposition if we could learn latent-space dynamical models directly from video (Thuruthel and Iida, 2023) and later leverage them for control (Almanzor et al., 2023). Therefore, we generate three datasets based on the PCS soft robot model. *CS* considers one segment with constant strain and is modeled using three configuration variables. *PCC-NS-2* and *PCC-NS-3* only consider bending deformations and contain soft robots with two and three segments, respectively. As all previous examples exampled ODEs, we strive to test the proposed approach also on a system that is governed by PDEs. Specifically, we consider the Reaction-diffusion (*R-D*) dataset as included in the SINDy Autoencoder paper (Champion et al., 2019). For all datasets, we generate images of size 32×32px and subsequently normalize the pixels to the interval $[-1, 1]$.

Unactuated Mechanical Datasets

We consider multiple mechanical datasets based on a standard implementation included in the *Toy Physics* category of the *NeurIPS 2021 Track on Datasets and Benchmarks* publication by Botev et al. (2021): mass-spring with friction (M-SP+F), a single pendulum with friction (S-P+F), and a double pendulum with friction (D-P+F). All datasets contain 5000 system trajectories in the training set and 1000 trajectories each in the validation and test set. Each trajectory is generated by first randomly initializing the system, then rolling it out for 3 s using an Euler integrator with a time step size of 5 ms. Samples are recorded at a rate of 20 Hz (i.e., a time step of 0.05 s). As a result, each trajectory contains 60 images of

the system's state. As all of these datasets are unactuated, we can deactivate the input-to-forcing mapping component from all models (e.g., set $g(u) = 0$ for the CON model).

The *M-SP+F* dataset contains motion samples of a damped harmonic oscillator with a mass of 0.5 kg, a spring stiffness of 2 N/m, and a damping coefficient of 0.05 Ns/m. For each trajectory, the initial condition of the mass-spring is randomly sampled by combining a random sign(q) with a uniformly sampled $|q| \sim \mathcal{U}(0.1\,\text{m}, 1\,\text{m})$. The position of the mass is rendered with a filled circle in a grayscale image.

The *SP+F* and *DP+F* datasets include the evolutions of a single-link pendulum and double-link pendulum, respectively, with a mass of 0.5 kg attached to the end of each link, which has a length of 1 m. The dataset considers a gravitational acceleration of $3\,\text{m/s}^2$. A rotational damper with coefficient 0.05 Nms/rad provides the friction. Similarly to the *M-SP+F* dataset, both the sign and the absolute value of the initial configuration are randomly sampled, where $|q(0)| \sim \mathcal{U}(1.3\,\text{rad}, 2.3\,\text{rad})$. The position of each mass is rendered with a filled circle. For the single-link pendulum, this is done in grayscale, and for the double pendulum, each mass is rendered with a different color (i.e., blue and red).

Actuated Continuum Soft Robot Datasets

The shape of slender and deformable rods can be approximated by considering the deformations along the 1D curve of the backbone (Gazzola et al., 2018). While this curve is still infinite-dimensional, it is possible to discretize the backbone into (many) segments with piecewise constant strain (Gazzola et al., 2018; Renda et al., 2016). Accordingly, we describe the kinematics of a planar continuum soft robot consisting of n_b segments with the PCS model (Renda et al., 2016). We assume each segment has a length of 100 mm and a diameter of 20 mm. The PCS model assumes each segment to have constant strain. In the planar case, this means that the shape of the ith segment can be parametrized by $\xi_i = \begin{bmatrix} \kappa_{\text{be},i} & \sigma_{\text{sh},i} & \sigma_{\text{ax},i} \end{bmatrix}^\top \in \mathbb{R}^3$ where $\kappa_{\text{be},i}$ is the bending strain (i.e., the curvature) in the unit rad/m, $\sigma_{\text{sh},i}$ is the shear strain (dimensionless), and $\sigma_{\text{ax},i}$ is the axial elongation strain (dimensionless). The robot's configuration is then defined as $q = \begin{bmatrix} \xi_1^\top & \cdots & \xi_i^\top & \cdots & \xi_{n_\text{b}}^\top \end{bmatrix}^\top$. In the case of PCC, only the bending strain is active as shear strains and axial strains are neglected, and the configuration is now $q \in \mathbb{R}^{n_\text{b}}$. The PCS model generates EOM in the form of (Della Santina et al., 2023)

$$M(q)\,\ddot{q} + C(q,\dot{q})\,\dot{q} + G(q) + K_\text{q}\,q + D_\text{q}\,\dot{q} = u(t), \tag{11.54}$$

where $M(q) \succ 0$ and $C(q,\dot{q})$ are the inertia and Corioli matrices, respectively. $G(q)$ collects the gravitational forces, $K_\text{q} \succ 0$ is the stiffness matrix, and $D_\text{q} \succ 0$ contains the damping coefficients. $u(t) \in \mathbb{R}^{n_\text{b}}$ is an external force acting on the generalized coordinates, and now $m = n_\text{b}$.

We derive the corresponding dynamics for a continuum soft robot of material density $600\,\text{kg/m}^3$, elastic modulus of 20 000 Pa, shear modulus of 10 000 Pa, and damping coefficients of $0.000\,01\,\text{Nm}^2\text{s}$ for bending strains, 0.01 Ns for shear strains, and 0.01 Ns for axial strains, respectively. Gravity is pointing downwards. The implementation of the dynamics in JAX (Bradbury et al., 2018) is based on the *JSRM* library (Stölzle et al., 2024a,b), and we simulate the robot using a constant integration time step of 0.1 ms. We render grayscale images of the robot with a size of 32×32px at a rate of 50 Hz using OpenCV (Bradski, 2000). We generate 10000 trajectories, each of duration 2.0 s and a sampling time-step of 0.02 s.

We use 60 % training, 20 % validation, and 20 % test split. For each trajectory, we randomly sample a constant actuation/input $u \sim \mathcal{U}(-u_{\max}, u_{\max})$. We choose the maximum actuation magnitude to be equal to the sum of the contribution of the potential forces (i.e., elastic and gravitational forces): $u_{\max} = G(q_{\max}) + K\, q_{\max}$ with $q_{\max,i} = \left[5\,\pi \text{ rad/m}, 0.2, 0.2\right]^\top$.

We generate three datasets based on this continuum soft robot model: in the *CS* dataset, we consider one segment with all three planar strains active (i.e., bending, shear, and elongation). This results in three DOF and six-state variables in the dynamical model. In the case of the *PCC-NS-2* and *PCC-NS-3* datasets, we base the dataset on a simulated system consisting of two planar bending segments, respectively. Each segment is parametrized using CC (Rosi et al., 2022; Webster III and Jones, 2010), which results in two configuration variables and a state dimension of four.

Unactuated PDE Reaction-Diffusion Dataset

We consider the 1st-order Reaction-diffusion (*R-D*) PDE on which Champion et al. (2019) evaluated their SINDy Autoencoder on. The PDE of the high-dimensional lambda-omega reaction-diffusion system is defined as

$$\begin{aligned}
\frac{\partial u}{\partial t} &= \left(1-(u^2+v^2)\right) u + \beta\,(u^2+v^2)\,v + d_1 \left(\frac{\partial^2 u}{\partial q_1^2} + \frac{\partial^2 u}{\partial q_2^2}\right), \\
\frac{\partial v}{\partial t} &= -\beta(u^2+v^2)\,u + (1-(u^2+v^2))\,v + d_2 \left(\frac{\partial^2 v}{\partial q_1^2} + \frac{\partial^2 v}{\partial q_2^2}\right),
\end{aligned} \tag{11.55}$$

where $u(t,q) : \mathbb{R}\times\mathbb{R}^2 \to \mathbb{R}$ and $v(t,q) : \mathbb{R}\times\mathbb{R}^2 \to \mathbb{R}$ are time-dependent two vector fields defined over the spatial domain $q \in \mathbb{R}^2$. We choose the same system parameters and initial condition as Champion et al. (2019): $d_1, d_2 = 0.1$, and $\beta = 1$ and

$$\begin{aligned}
u(0,q) &= \tanh\left(\sqrt{q_1^2+q_2^2}\,\cos\left(\angle(q_1+iq_2) - \sqrt{q_1^2+q_2^2}\right)\right), \\
v(0,q) &= \tanh\left(\sqrt{q_1^2+q_2^2}\,\sin\left(\angle(q_1+iq_2) - \sqrt{q_1^2+q_2^2}\right)\right).
\end{aligned} \tag{11.56}$$

After discretizing the spatial domain into 32 points along each dimension, we solve the PDE with a MATLAB ODE45 solver the solution of $u(t,q)$ and $v(t,q)$ at each time step and grid point. Subsequently, the solution is multiplied with a Gaussian centered at the origin (Champion et al., 2019)

$$\begin{aligned}
\bar{u}(t,q) &= \exp(-0.01\,(q_1^2+q_2^2))\,\bar{u}(t,q), \\
\bar{v}(t,q) &= \exp(-0.01\,(q_1^2+q_2^2))\,\bar{v}(t,q).
\end{aligned} \tag{11.57}$$

We integrate the system from the specified initial condition for 500 s and store samples at a time step of 0.05 s. We divide the entire sequence into 99 subsequences each containing 101 samples. We train the models to predict these subsequences that have a horizon of 5.0 s each.

We stack the solution of $\bar{u}(t,q)$ and $\bar{v}(t,q)$ contained in the two grids $o_{\mathrm{u}}(t), o_{\mathrm{v}}(t) \in \mathbb{R}^{32\times32}$, respectively, to gather the images $o(t) \in \mathbb{R}^{32\times32\times2}$ containing two channels. A sample sequence of the generated images is presented in Fig. 11.9. We use 60 % of the subsequences (i.e., 59) as our training set, and employ 20 % (i.e., 19) for the validation and test sets, respectively.

11.4.4 Results

Results for Unactuated Mechanical Datasets

The results in Tab. 11.2 show that the *NODE* model slightly outperforms the *CON* network on the *M-SP+F* and *S-P+F* datasets. However, as the datasets do not consider system inputs, we can remove the input mapping from all models (e.g., *RNN*, *GRU*, *coRNN*, *CON*, and *CFA-CON*). With that adjustment, the *CON* network has the fewest parameters among all models, particularly two orders of magnitude less than the NODE model. Therefore, we find it very impressive that the CON network is roughly on par with the NODE model. For the *D-P+F* dataset, we can conclude that the *CFA-CON* model offers the best performance across all methods. Finally, most of the time, the *CON* & *CFA-CON* networks outperform the other baseline methods that have more trainable parameters. Sequences of stills of the rollout of the CON model trained on the *M-SP+F*, and *D-P+F* datasets are presented in Figs. 11.6 and 11.7.

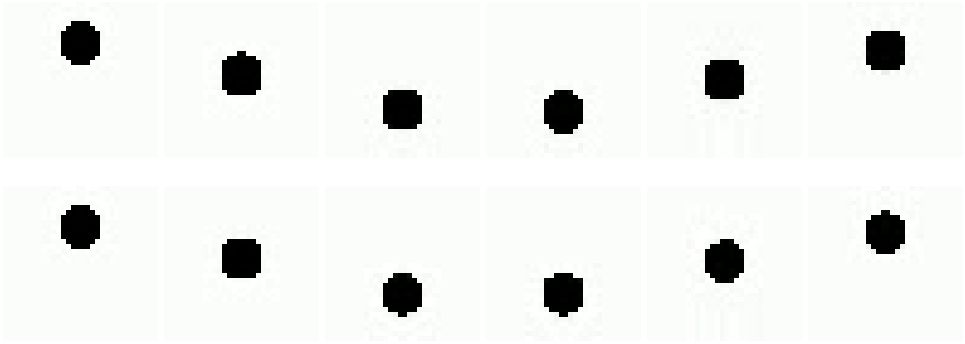
Results for Actuated Continuum Soft Robot Datasets

The results in Tab. 11.2 show that *CON-M* matches the performance of the state-of-the-art methods across all experiments. In the case of *PCC-NS-3*, *CON-M* even decreases the RMSE error by 6 % w.r.t. the closest baseline method (MECH-NODE). Impressively, the performance is not reduced (but instead often even improved) compared to other models that offer a much larger design space for learning the dynamics (e.g. NODE). Furthermore, *CON-S* and *CFA-CON* often only exhibit slightly lower performance than *CON-M*, even though they have significantly fewer parameters and consider an approximated solution, respectively. Supplementary results (e.g., more evaluation metrics) can be found in Appendix B.2. Sequences of stills for the CON model trained on the *PCC-NS-3* dataset are provided in Fig. 11.8.

We also conduct on the *PCC-NS-2* dataset an analysis concerning the effect of the latent dimension on the performance (see Fig. 11.5). For this experiment, all hyperparameters were tuned for $n_z = 8$, and we observe that the CON models have a much-improved consistency and smaller variance w.r.t. the baseline methods when the latent dimensionality is increased.

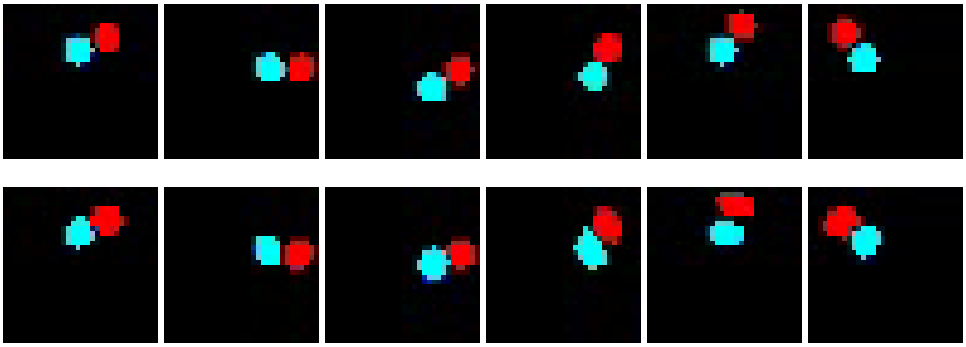
Results for Reaction-Diffusion Dataset

To address the unactuated nature of the *R-D* dataset, we remove, analog to the *M-SP+F*, *S-P+F*, and *D-P+F* datasets, the input-to-state mapping parameters of the dynamical models (e.g., the $B(u)$ and $E(\tau)$ MLPs for the CON models). Furthermore, the PDE describing the system dynamics is of 1^{st}-order. Therefore, we leverage the 1^{st}-order versions of the latent dynamics, in particular for the *coRNN*, *CON*, *CFA-CON* models, as specified in Apx. B.1.3. We report the RMSE of the test set evaluations in Tab. 11.2. Furthermore, we also present a sequence of stills of the rollout of a trained latent dynamics CON model in Fig. 11.9. We find it impressive that CON with its strong stability guarantees can accurately model the dynamics of a high-dimensional PDE system. Still, these initial results show that, for a comparable number of model parameters, the coRNN model exhibits a 32 % better performance than the CON model. Therefore, in particular as coRNN and CON are both oscillatory networks, it would be interesting to analze in future work which characteristic gives coRNN the most performance benefits compared to CON on this PDE datasets.



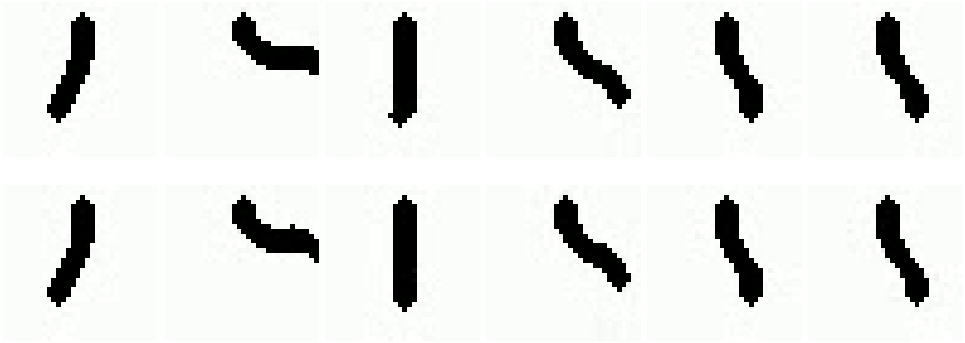
(a) t=0.0 s (b) t=0.55 s (c) t=1.10 s (d) t=1.65 s (e) t=2.20 s (f) t=2.75 s

Figure 11.6: Prediction sequence of a CON model with latent dimension $n_z = 4$ trained on the damped harmonic oscillator (*M-SP+F*) dataset (Botev et al., 2021). **Top row:** Ground-truth evolution of the system. **Bottom row:** Predictions of the *CON* model.
The prediction model is given three images centered around $t = 0$ for encoding the initial latent $z(0)$ and estimation of the initial latent velocity $\dot{z}(0)$. Subsequently, we roll out the autonomous network dynamics (i.e., unforced) and compare the decoded predictions with the ground-truth evolution of the system.



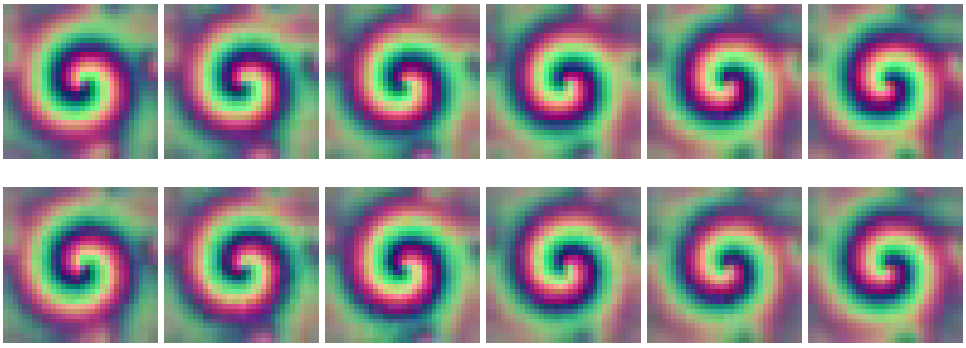
(a) t=0.0 s (b) t=0.55 s (c) t=1.10 s (d) t=1.65 s (e) t=2.20 s (f) t=2.75 s

Figure 11.7: Prediction sequence of a CON model with latent dimension $n_z = 12$ trained on the double pendulum with friction (*D-P+F*) dataset (Botev et al., 2021). **Top row:** ground-truth evolution of the system. **Bottom row:** predictions of the *CON* model.
The prediction model is given three images centered around $t = 0$ for encoding the initial latent $z(0)$ and estimation of the initial latent velocity $\dot{z}(0)$. Subsequently, we roll out the autonomous network dynamics (i.e., unforced) and compare the decoded predictions with the ground-truth evolution of the system.



(a) t=0.0 s (b) t=0.36 s (c) t=0.72 s (d) t=1.08 s (e) t=1.44 s (f) t=1.80 s

Figure 11.8: Prediction sequence of a forced CON model with latent dimension $n_z = 12$ trained on the soft robotic *PCC-NS-3* dataset containing trajectories of a simulated piecewise constant curvature robot with three segments. **Top row:** Ground-truth evolution of the system. **Bottom row:** Predictions of the *CON-M* model.
The prediction model is given three images centered around $t = 0$ for encoding the initial latent $z(0)$ and estimation of the initial latent velocity $\dot{z}(0)$. Subsequently, we roll out the autonomous network dynamics (i.e., unforced) and compare the decoded predictions with the ground-truth evolution of the system.



(a) t=0.0 s (b) t=1.0 s (c) t=2.0 s (d) t=3.0 s (e) t=4.0 s (f) t=5.0 s

Figure 11.9: Prediction sequence of an unforced, 1st-order CON model with latent dimension $n_z = 4$ trained on the reaction-diffusion (*R-D*) dataset. **Top row:** Ground-truth evolution of the system. **Bottom row:** Predictions of the *CON-M* model. We roll out the autonomous, 1st-order network dynamics (i.e., unforced) and compare the decoded predictions with the ground-truth evolution of the system.

11.5 Exploiting the Dynamic Structure for Latent Space Control

We consider the problem setting of guiding the system towards a desired observation o^{d} by providing a sequence of inputs $u(t_k)$ such that at time t_N, the actual observation $o(t_N)$ matches o^{d}. A relatively simple way would be to encode the desired observation into latent space $z^{\mathrm{d}} = \Phi(o^{\mathrm{d}})$ and then to design a feedback controller (e.g., PID) in latent space: $g(u) = \mathrm{PID}(z^{\mathrm{d}} - z(t), -\dot{z})$. Unfortunately, several challenges appear: first, it is not clear how the latent-space force $\tau = g(u)$ can be mapped back to an actual input $u(t)$ as the inverse input-to-forcing mapping g^{-1} is generally not known. Furthermore, relying entirely on a PID controller has several well-known drawbacks, such as poor and slow transient behavior, steady-state errors (in case the integral gain is chosen to be zero), and instability for high proportional and integral gains. We take inspiration from potential shaping strategies (Bloch et al., 2001; Ortega et al., 2021), which are widely used for effectively controlling (elastic) robots, and, therefore, combine a feedforward term compensating the latent-space potential forces with an integral-saturated, PID-like feedback term. For mapping the desired forcing τ back to an input $u(t)$, we train an forcing decoder $\eta : \mathbb{R}^n \to \mathbb{R}^m$ that approximates g^{-1}. Specifically, we consider here the structure $u = \eta(\tau) = E(\tau)\,\tau$, where $E \in \mathbb{R}^{m \times n}$ is parameterized by an MLP.

11.5.1 Control Strategy

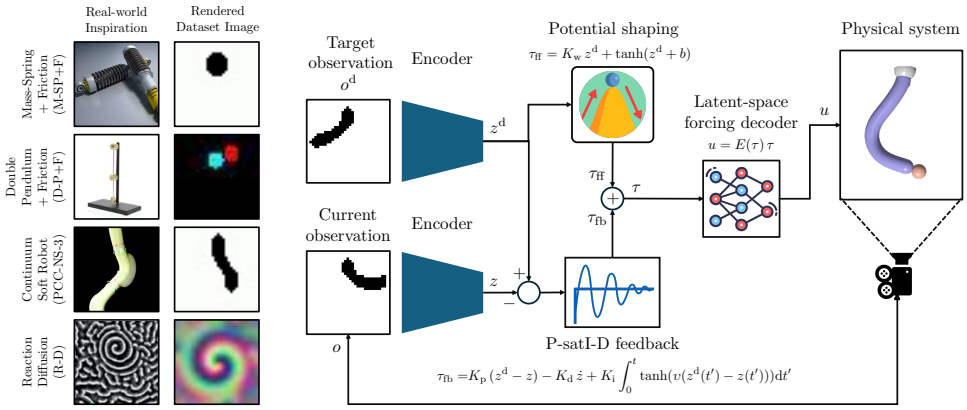
The latent-space control law is given by

$$\tau(t) = g(u) = \underbrace{K_{\mathrm{w}}\, z^{\mathrm{d}} + \tanh(z^{\mathrm{d}} + b)}_{\substack{\text{Feedforward term:}\\ \text{compensation of potential forces}}} \underbrace{+ K_{\mathrm{p}}\,(z^{\mathrm{d}} - z) - K_{\mathrm{d}}\,\dot{z} + K_{\mathrm{i}} \int_0^t \tanh(\upsilon(z^{\mathrm{d}}(t') - z(t')))\mathrm{d}t'}_{\text{Feedback term: P-satI-D}} \tag{11.58}$$

where $K_{\mathrm{p}}, K_{\mathrm{i}}, K_{\mathrm{d}} \in \mathbb{R}^{n \times n}$ are the proportional, integral, and derivative control gains, respectively. As integral terms can often lead to instability when applied to nonlinear systems (Stölzle et al., 2024b), we adopt an integral term saturation (Pustina et al., 2022a) with the associated dimensionless gain $\upsilon \in \mathbb{R}$, which ensures that the integral error added at each time step is bounded to the interval $(-1, 1)$. Subsequently, τ is decoded to the input as $u(t) = \eta(\tau) = E(\tau)\,\tau$. For training this decoder, we add a reconstruction loss to (11.53): $\mathcal{L}_{\mathrm{u}}(t_k) = \lambda_{\mathrm{u}}\,\mathrm{MSE}(u(t_k), \hat{u}(t_k)) = \lambda_{\mathrm{u}}\,\mathrm{MSE}\,(u(t_k), \eta(g(u(t_k))))$.

11.5.2 Implementation of Closed-Loop Control

We generate a trajectory of 7 setpoints, where $q^{\mathrm{d}}(t_j) \in \mathbb{R}^{n_{\mathrm{q}}}$ is a sampled configuration of the actual system. Then, we render an image $o^{\mathrm{d}}(t_j)$ that represents the target observation for the controller and encode it into latent space to retrieve $z^{\mathrm{d}} \in \mathbb{R}^{n_{\mathrm{z}}}$. At time step k, we render an image $o(t_k)$ of the robot's current configuration $q(t_k)$ and encode the image. Subsequently, we evaluate the control law and apply the decoder $u(t_k) = \eta(\tau(t_k)))$, which is finally passed to the simulator that integrates the ground-truth dynamics to the next time-step t_{k+1} considering the actuation $u(t_k)$. The controller runs at 100 Hz, and we simulate the ground-truth dynamics with a *Dopri5* ODE integrator at a time-step of 5×10^{-4} s.



(a) Samples of used datasets

(b) Blockscheme of model-based control in latent space

Figure 11.10: **Panel (a):** Samples of some of the datasets used as part of the experimental verification, specifically for the results reported in Tab. 11.2. The real-world Reaction-Diffusion image is adopted from (Epstein and Xu, 2016). **Panel (b):** Model-based control in latent space by exploiting the physical structure of the CON model.

11.5.3 Latent Space Control of a Damped Harmonic Oscillator

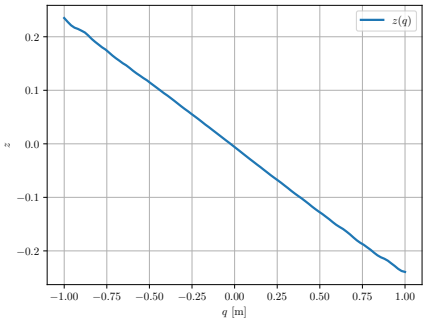
We consider an actuated version of the *M-SP+F* dataset (i.e., a damped harmonic oscillator) and denote it with *M-SP+F+A*. All system, trajectory sampling and rendering parameters remain the same, except that for each trajectory in the dataset we randomly sample a forcing $u \sim \mathcal{U}(-1\,\mathrm{N}, 1\,\mathrm{N})$.

We train a CON model with latent dimension $n_z = 1$ over three random seeds on the *M-SP+F+A* dataset. This means that the network consists of a single oscillator. From the three different random seeds, we choose the model that achieves the best validation loss, which results in an RMSE of 0.0327, a PSNR of 5.99, and SSIM of 0.9796 on the test set.

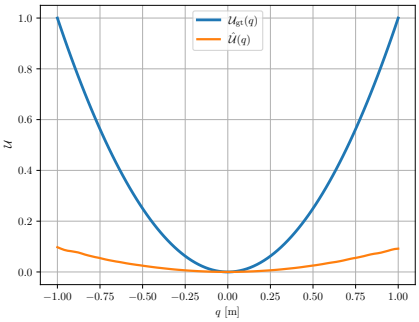
Fig. 11.11 shows how the encoder learns an almost linear relationship between the actual configuration of the system and the predicted latent space representation. Furthermore, we notice that both the ground-truth and the learned potential energy are convex and exhibit a global minimum at $q = 0\,\mathrm{m}$.

We compare the performance of *P-satI-D*, *D+FF*, and *P-satI-D+FF* controllers based on the CON model in Fig. 11.12. For the *P-satI-D* controller, we choose the control gains $K_\mathrm{p} = 10, K_\mathrm{i} = 10, K_\mathrm{d} = 5, \upsilon = 1$. The *D+FF* controller uses $K_\mathrm{d} = 3.5$. Finally, the *P-satI-D+FF* is configured with $K_\mathrm{p} = 2, K_\mathrm{i} = 0.3, K_\mathrm{d} = 3.5, \upsilon = 1$. The results show that the *P-satI-D+FF* controller exhibits thanks to its feedforward term no overshooting and a faster response time than the pure feedback controller *P-satI-D*. The high accuracy of the feedfoward term can be seen from the performance of the *D+FF* controller, that only exhibits relatively small steady-state error. Adding small proportional and integral feedback actions in the *P-satI-D+FF* controller keeps the compliance high while removing the steady-state error and reducing the response time.

Finally, we visualize the behavior of the *P-satI-D+FF* controller as a sequence of stills in Fig. 11.13.



(a) Configuration q to latent z mapping



(b) Ground-truth and learned potential energy $\mathcal{U}$

Figure 11.11: **Panel (a):** Learned mapping from configuration to latent space for the CON model with n_z (i.e., consisting of a single oscillator) trained on the actuated damped harmonic oscillator (*M-SP+F+A*) dataset. **Panel (b):** The blue line represents the ground-truth potential energy of the damped harmonic oscillator. The orange line represents the learned potential energy of the CON model evaluated vs. the system configuration by rendering and subsequently encoding into latent space each configuration value.

11.5.4 Latent Space Control of a Two Segment PCC Soft Robot

Experimental Setup.

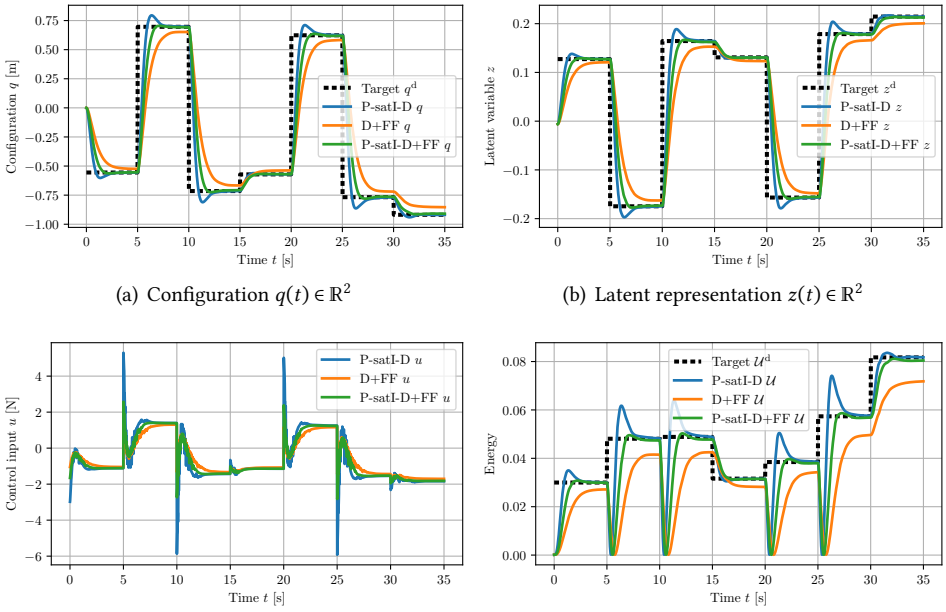
We train a CON model with two latent variables ($n_z = 2$) on the *PCC-NS-2* dataset. Analog to the input encoder mapping $B(u)$, the forcing decoder mapping $E(\tau)$ is parametrized by an MLP consisting of five layers with hidden dimension 30 and a hyperbolic tangent activation function. The *CON* model achieves an RMSE of 0.1628 on the test set.

Model Selection

For the control experiments, we train instances of the *MECH-NODE* and *CON-M* models with latent dimension $n_z = 2$ and with neural network weights initialized with three different random seeds. We found that model-based control does not perform as well when the latent stiffness Γ_{w} (as visualized in Fig. 11.14(a)) is significantly larger along one of the Eigenvectors than along the other one. Therefore, we evaluate the Eigenvalues of the learned stiffness matrix in $\mathcal{W}$-coordinates after training: $\lambda_{1,2}(\Gamma_{\text{w}})$. Particularly, we choose the seed that minimizes the normalized standard deviation of the Eigenvalues: $\text{seed} = \arg\min \frac{\sigma_\lambda}{\mu_\lambda}$, where

$$\mu_\lambda = \frac{\lambda_1(\Gamma_{\text{w}}) + \lambda_2(\Gamma_{\text{w}})}{2}, \quad \sigma_\lambda = \sqrt{\frac{(\lambda_1(\Gamma_{\text{w}}) - \mu_\lambda)^2 + (\lambda_2(\Gamma_{\text{w}}) - \mu_\lambda)^2}{2}}. \tag{11.59}$$

We benchmark two controllers on the simulated continuum soft robot consisting of two segments: (i) a pure *P-satI-D* controller (i.e., the feedback term in (11.58)) that leverages the smooth mapping into the latent representation enabled by the CON dynamic model and the β-VAE, and (ii) a *P-satI-D+FF* (i.e., (11.58)) that exploits the structure of the CON dynamics by compensating for the potential forces. The stable closed-loop system dynamics made the control gain tuning very easy, and we selected $K_{\text{p}} = 1, K_{\text{i}} = 2, K_{\text{d}} = 0.02, \upsilon = 1$ for the *P-satI-D* controller and $K_{\text{p}} = 0, K_{\text{i}} = 2, K_{\text{d}} = 0.05, \upsilon = 1$ for the *P-satI-D+FF* controller, respectively.



(a) Configuration $q(t) \in \mathbb{R}^2$

(b) Latent representation $z(t) \in \mathbb{R}^2$

(c) Control input $u(t) \in \mathbb{R}^2$

(d) Potential energy $U(t) \in \mathbb{R}$

Figure 11.12: Latent-space control of an actuated damped harmonic oscillator (*M-SP+F+A*) following a sequence of setpoints. We compare multiple controllers based on a trained CON network with $n_z = 1$. The CON model weights are initialized using a random seed of 0. The blue line represents a pure feedback controller (*P-satI-D*). The orange line visualizes the behavior of a feedforward controller with only a damping term applied in feedback (*D+FF*). The green line shows the performance of our proposed combination of feedback and feedforward terms (*P-satI-D+FF*). The dotted and solid lines show the reference and actual values, respectively. For each setpoint, we randomly sample a desired shape q^{d} and render the corresponding image o^{d}. This image is then encoded to a target latent z^{d}. The controller then computes a latent-space torque F^{d}, which is decoded to an input u. Finally, we provide this input to the simulator, which performs a roll-out of the closed-loop dynamics. Important: The robot's configuration (i.e., the first-principle, minimal-order state) is solely used for generating a target image and simulating the closed-loop system.

Furthermore, we compare the control performance of our model-based controllers with a baseline control strategy based on the *MECH-NODE* ($n_z = 2$) that achieves an error of 0.1104 on the test set. For *MECH-NODE*, we chose the model with the lowest validation loss (seed 0) First, we utilize the same P-satI-D feedback controller as for the CON model to generate the control action $\tau(t)$ in latent space. As the MECH-NODE uses an MLP to parameterize the function $\dot{\xi} = f_\xi(\xi, u)$, we cannot easily map $\tau(t)$ into an input $u(t)$. Therefore, we linearize the latent space dynamics w.r.t to the input as $f_{\xi,\mathrm{ac}}(\xi, u) = f_\xi(\xi, 0) + A(\xi)\, u$, where $A(\xi) = \frac{\partial f_\xi}{\partial u}(\xi, 0)$ is computed using autodiff. Then, $u(t) = A^\top(\xi)\, \tau(t)$. After tuning the control gains, we choose $K_{\mathrm{p}} = 0.001, K_{\mathrm{i}} = 0.02, K_{\mathrm{d}} = 1 \times 10^{-5}, v = 1$.

11

Potential energy landscape

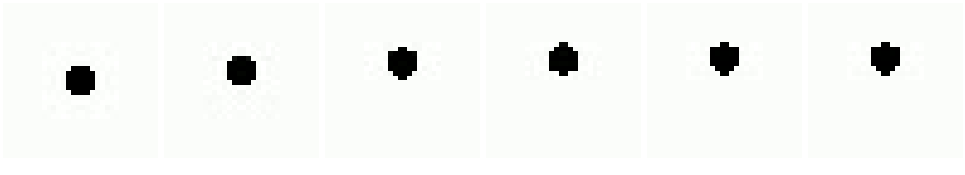
When leveraging (learned) dynamical models for setpoint regulation, it is essential to accurately estimate the potential energy as this dictates the efficacy of the feedforward

terms. Therefore, we qualitatively evaluate the potential energy landscape of the CON latent dynamic model.

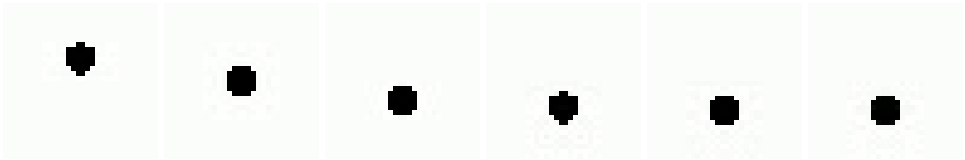
In Fig. 11.14(a), we can see how CON contains a single, isolated, and globally asymptotically stable equilibrium as proven in Section 11.2, respectively. Furthermore, we want to verify that the learned potential corresponds to the actual potential energy of the simulated system. An autonomous continuum soft robot with the tip pointing downwards in a straight configuration exhibits an isolated, globally asymptotically stable equilibrium at $q = 0$ (i.e., zero strains) (Stölzle and Della Santina, 2021). For this purpose, we can compare the learned potential energy field in Fig. 11.14(b) with the ground-truth potential energy field in Fig. 11.14(c). We confirm, based on Fig. 11.14(b), that, indeed, the learned potential also has its minimum close to/at $q = 0$. Although the field is shaped slightly differently, the potential forces are clearly pointing inwards towards the global attractor.

Results

As an evaluation metric, we consider the RMSE between the actual and the reference trajectory. The *P-satI-D* applied to the *MECH-NODE* model (baseline) achieves an RMSE of 2.88 rad/m w.r.t. to the desired configuration q^{d} (but unknown to the algorithm). The *P-satI-D* CON controller, which does not exploit the learned latent dynamics for control, exhibits an RMSE of 4.08 rad/m w.r.t. to the desired configuration q^{d}. The *P-satI-D+FF* controller exhibits an RMSE of 2.12 rad/m w.r.t. to the desired configuration q^{d}. We also visualize the closed-loop trajectories in Fig. 11.15 and as sequences of stills in Fig. 11.16. We conclude that the nicely structured latent space generated by the β-VAE allows the *P-satI-D* controller to effectively regulate the system towards the setpoint, although the response time is rather slow. The *P-satI-D+FF* controller is able to exploit the structure of the CON model through its potential shaping feedforward term. With that, *CON P-satI-D+FF* exhibits a faster response time and a 26 % lower RMSE than the *MECH-NODE P-satI-D* baseline.

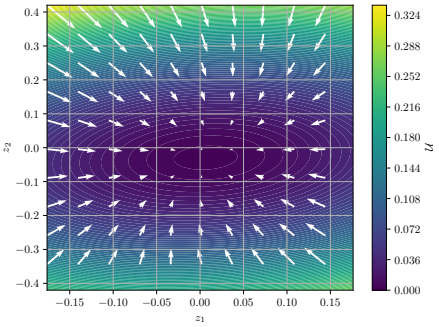


(a) t=0.0 s (b) t=0.5 s (c) t=1.0 s (d) t=1.5 s (e) t=2.0 s (f) Target $0-4.9$s

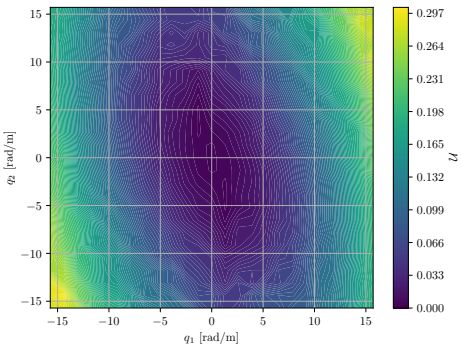


(g) t=5.00 s (h) t=5.5 s (i) t=6.0 s (j) t=6.5 s (k) t=7.5 s (l) Target $5-9.9$s

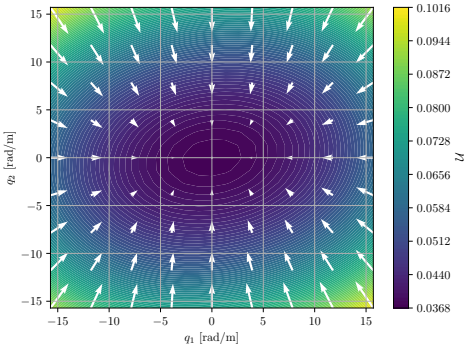
Figure 11.13: Sequence of closed-loop control of an actuated damped harmonic oscillator (*M-SP+F+A*) with a *P-satI-D+FF*controller based on a trained CON with $n_z = 1$. **Columns 1-4:** show the actual behavior of the closed-loop system. **Column 5:** demonstrates the target image that the control sees for all time instances in the row.



(a) Learned potential energy as a function of z

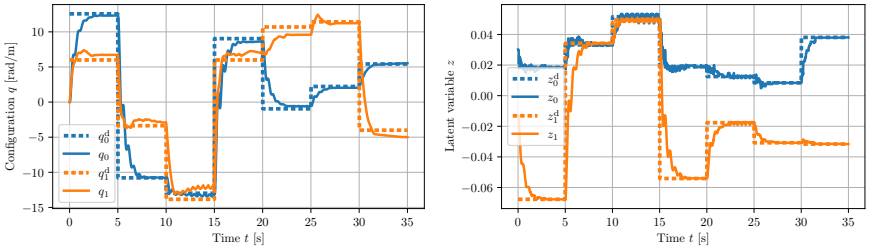


(b) Learned potential energy as a function of q



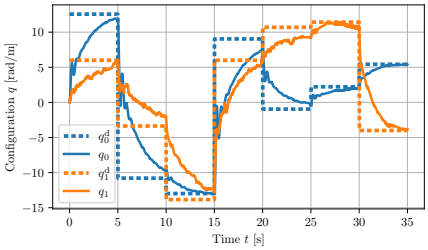
(c) Ground-truth potential energy as a function of q

Figure 11.14: Potential energy landscapes of a CON with $n_z = 2$ trained to learn the latent space dynamics of a continuum soft robots (simulated with two PCC segments). **Panel (a):** Here, we visualize the learned potential energy of CON using the color scale as a function of the latent representation $z = x_w \in \mathbb{R}^2$. The arrows denote the gradient of the potential field $\frac{\partial \mathcal{U}}{\partial z}$ (i.e., the potential force), with the magnitude of the gradient expressed as the length of the arrow. **Panel (b):** Again, we display the learned potential energy of CON using the color scale, but in this case, as a function of the configuration $q \in \mathbb{R}$ of the robot (that is hidden from the model). First, we render an image o of the shape of the robot for each configuration $q = \begin{bmatrix} q_1 & q_2 \end{bmatrix}^\top \in \mathbb{R}^2$. Then, we encode the image into latent space as $z = \Phi(o)$. This allows us then to compute the potential energy $\mathcal{U}(z)$ of the CON latent dynamics model. **Panel (c):** Here, we display the potential energy and its associated potential forces of the actual (i.e., simulated) system.

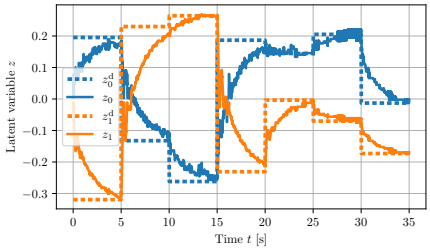


(a) Config. for P-satI-D with MECH-NODE.

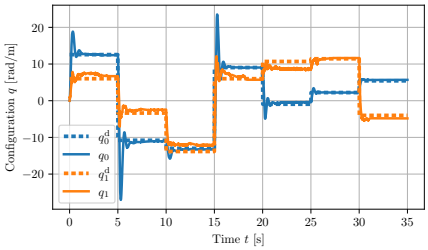
(b) Latent for P-satI-D with MECH-NODE.



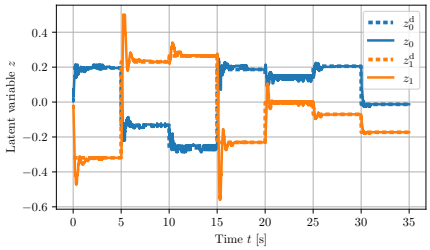
(c) Configuration for P-satI-D with CON.



(d) Latent for P-satI-D with CON.

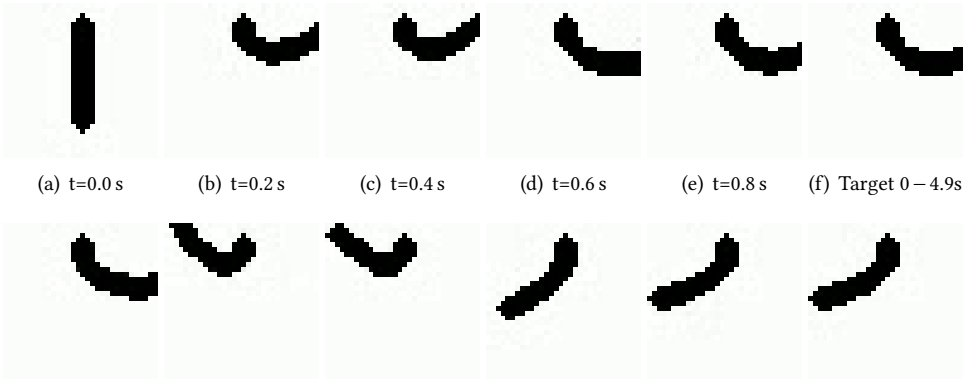


(e) Configuration for P-satI-D+FF with CON.



(f) Latent for P-satI-D+FF with CON.

Figure 11.15: Latent-space control of a continuum soft robot (simulated using two piecewise constant curvature segments) at following a sequence of setpoints: The upper two rows show the performance of a pure P-satI-D feedback controller operating in latent space z learned with the MECH-NODE and CON models, respectively. The lower row displays the results for a latent space controller based on the CON model that additionally also compensates for the learned potential forces.



(a) t=0.0 s (b) t=0.2 s (c) t=0.4 s (d) t=0.6 s (e) t=0.8 s (f) Target $0-4.9$s

(g) t=5.00 s (h) t=5.2 s (i) t=5.4 s (j) t=5.6 s (k) t=5.8 s (l) Target $5-9.9$s

Figure 11.16: Sequence of closed-loop control of a continuum soft robot consisting of two constant curvature segments with the *P-satI-D+FF* based on a trained CON with $n_z = 2$. **Columns 1-4:** show the actual behavior of the closed-loop system. **Column 5:** demonstrates the target image that the control sees for all time instances in the row.

11.6 Conclusion and Limitations

11.6.1 Conclusion

In this work, we propose a new formulation for a coupled oscillator that is inherently input-to-state stable. Additionally, we identify a closed-form approximation, that is able to simulate the network dynamics more accurately compared to numerical ODE integrators with similar computational costs. When learning latent dynamics with CON, we observe that the performance is on par or slightly better compared to SoA methods such as RNNs, NODEs, etc., even though we constrained the solution space to a ISS-stable coupled oscillator structure. Furthermore, we point out that the performance of the CON models is more consistent across latent dimensions compared to the baselines and improved when not specifically tuned for a given dimension. Furthermore, as seen in Tab. B.5, the closed-form approximation achieves, with the same number of model parameters, similar accuracies and double the training speed w.r.t. to the continuous-time model. Finally, we demonstrate that even a simple PID-like latent-space controller can effectively regulate the system to a setpoint. By exploiting the network structure and compensating for potential forces, regulation performance can be greatly improved, and response time decreased by more than 55 %.

11.6.2 Limitations

While we think our proposed method shows great potential and opens interesting avenues for future research, there exist certain limitations. For example, the proposed method of learning (latent) dynamics implicitly assumes that the underlying system adheres to the Markov property (e.g., the full state of the system is observable), that a system with mechanical structure can approximate it, and that it has an isolated, globally asymptotically stable equilibrium. This is, for example, the case for many mechanical systems (e.g., some continuum soft robots, deformable objects, and elastic structures) with continuous dynamics, convex elastic behavior, dissipation, and whose time-dependent effects (e.g., viscoelasticity, hysteresis) are negligible. Even if these conditions are not met globally, the method can be applied to model the local behavior around an asymptotic equilibrium point of the system (e.g., robotic manipulators, legged robots) with added stability benefits for out-of-distribution samples. Alternatively, the method could be extended to relax some of these assumptions, e.g., by allowing for multiple equilibria, zero damping, or by incorporating additional terms to capture discontinuous dynamics (e.g., stick-slip models) or period motions (e.g., limit cycles such as the Van der Pol oscillator). The proposed method might not be suitable for some physical systems, such as nonholonomic systems, partially observable systems, or systems with non-Markovian properties. Examples of such systems include mobile robots and systems with hidden states or delayed observations.

Furthermore, the approximated closed-form solution shows the best integration for situations where linear, decoupled dynamics dominate the transient. For dominant nonlinear, coupled forces, the performance of CFA-CON degrades, and it might be better to revert to numerical integration of the CON ODE. Finally, the control works exceptionally well in the setting where the latent dimension equals the input dimension. We hypothesize that this enables the method to identify a diffeomorphism between the input and the latent-space forcing. Still not investigated is how the performance could degrade if $n_z > m$ (or $n_z < m$

for that matter).

Afterword

In this chapter, we introduced Coupled Oscillator Network (CON) and leveraged it for learning mechanically interpretable latent dynamics of physical systems. The ability to derive kinetic and potential energy terms of the CON i) allows us to provide strong stability guarantees (e.g., GAS, ISS), and ii) enables us to apply computationally efficient model-based control strategies established in the (soft) robotics community such as P-satI-D+potential shaping directly in latent space. Although the learned dynamics no longer map directly to a soft robot model as they did in Chapter 10, the fact that the latent dynamics retain a physical (in this case, mechanical) interpretation offers several advantages: (a) it enables us to enforce stability characteristics in the learned system similar to those observed in the original physical system, and (b) it allows the application of control strategies originally designed for fully physics-derived models. These attributes underscore the benefits of *greybox* approaches, particularly those that incorporate physical structure and provide stability guarantees.

The proposed approach of integrating an encoding into a latent space with a latent dynamical system that incorporates specific physical structures and stability guarantees proves valuable not only for learning models but also for motion planning. In recent years, there has been a growing trend in the research community to develop motion policies with stability guarantees by leveraging diffeomorphic encoders and latent dynamics represented by stable dynamical systems within the framework of Stable Motion Primitive (SMP)(Pérez-Dattari and Kober, 2023; Rana et al., 2020; Zhi et al., 2024). In Chapter 12, we expand this method to learn periodic motions and demonstrate its ability to generate natural, compliant, and safe behavior across various real (soft) robots.

12

Learning Stable Periodic Robot Motions from Demonstration

Foreword. This thesis has so far concentrated primarily on the use of learned models in shape sensing and control. However, a critical challenge in achieving fully autonomous robots lies in high-level decision-making. The controllers discussed in this work require either a setpoint or a trajectory to follow. In Chapter 7, we explored a method enabling users to guide the soft robot directly through their thoughts by manipulating setpoints in space, which a compliant impedance controller subsequently tracks. While this low-level HRI provides precise control over the robot's actions, it demands the user's full attention and can become exhausting, thereby limiting the efficiency gains we expect to gain by introducing intelligent, autonomous robots. An alternative approach involves the user specifying only high-level tasks, allowing the robot to plan its motions independently. Among potential methods, including RL and optimization-based strategies like MPC, Learning from Demonstration (LFD) stands out as particularly promising as it allows the learning of complex motions from humans or even from other biological creatures. As a special case of LFD, research on learning motion policies using dynamical systems has been well-established (Ijspeert et al., 2013). Being referred to as Dynamic Motion Primitive (DMP) or Stable Motion Primitive (SMP), this strategy exhibits interpretability, compliant behavior, and convergence guarantees. Recently, advances in deep learning and normalizing flows (Kobyzev et al., 2020) have given rise to frameworks that increase the expressiveness of the motion policy by leveraging diffeomorphic mappings into latent spaces, parametrized by neural networks, combined with latent dynamical systems (Pérez-Dattari and Kober, 2023; Rana et al., 2020; Zhi et al., 2024). These methods enable the learning of more complex behaviors while maintaining interpretability, stability, and convergence guarantees. In this chapter, we extend this approach to learn periodic motions with guaranteed orbital stability. This is achieved by integrating a bijective encoder based on Euclideanizing flows with latent dynamics modeled as supercritical Hopf bifurcations. The approach aligns in terms of vision with the work in Chapter 11, but here, it is applied to motion policy rather than dynamical model learning.

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q} =$$

Abstract. *Learning from demonstration provides a sample-efficient approach to acquiring complex behaviors, enabling robots to move robustly, compliantly, and with fluidity. In this context, Dynamic Motion Primitives offer built-in stability and robustness to disturbances but often struggle to capture complex periodic behaviors. Moreover, they are limited in their ability to interpolate between different tasks. These shortcomings substantially narrow their applicability, excluding a wide class of practically meaningful tasks such as locomotion and rhythmic tool use. In this work, we introduce Orbitally Stable Motion Policys (OSMPs)—a framework that combines a learned diffeomorphic encoder with a supercritical Hopf bifurcation in latent space, enabling the accurate acquisition of periodic motions from demonstrations while ensuring formal guarantees of orbital stability and transverse contraction. Furthermore, by conditioning the bijective encoder on the task, we enable a single learned policy to represent multiple motion objectives, yielding consistent zero-shot generalization to unseen motion objectives within the training distribution. We validate the proposed approach through extensive simulation and real-world experiments across a diverse range of robotic platforms—from collaborative arms and soft manipulators to a bio-inspired rigid–soft turtle robot—demonstrating its versatility and effectiveness in consistently outperforming state-of-the-art baselines such as diffusion policies, among others.*

This chapter is partly based on 🖹 ***M. Stölzle**, T.K. Rusch*, Z.J. Patterson*, R. Pérez Dattari, F. Stella, J. Hughes, C. Della Santina, and D. Rus (2025). Learning to Move in Rhythm: Task-Conditioned Motion Policies with Orbital Stability Guarantees. In Science Robotics Special issue on Transfer Learning, Foundation Models, and Emerging Capabilities.* **Under Review** (Stölzle et al., 2025b).

T.K.R. and Z.J.P. contributed equally to this work. M.S. conceived the project, and M.S., Z.J.P., T.K.R., C.D.S., and D.R. developed the research idea. M.S. developed the framework for training the orbital motion primitives. M.S. and C.D.S. derived the phase synchronization approach. M.S. and T.K.R. designed the encoder conditioning technique, and M.S. devised the smooth conditioning interpolation loss. Z.J.P. and M.S. performed the turtle robot experiments; M.S. conducted the UR5 robotic manipulator experiments; R.P.D. and M.S. executed the KUKA cobot experiments; M.S. and F.S. carried out the Helix soft robot experiments. M.S., Z.J.P., R.P.D., and F.S. performed the data analysis. M.S., Z.J.P., R.P.D., and F.S. wrote the manuscript. D.R., C.D.S., T.K.R., and Z.J.P. supervised the research project. D.R., C.D.S., and J.H. provided funding.

12.1 Introduction

Imitation Learning (Schaal, 1999; Zare et al., 2024) has regained substantial traction in recent years due to its superior sample and iteration efficiency in acquiring complex tasks compared to RL. Recent work has focused on improving the robustness, expressiveness, and generalization of motion policies learned from demonstration by leveraging modern ML architectures such as diffusion and flow matching models (Black et al., 2024; Chi et al., 2023); scaling up the number of demonstrations to increase robustness (Black et al., 2024; O'Neill et al., 2024; Team, 2025); training across multiple robot embodiments (e.g., different manipulators) to promote generalization (Black et al., 2024; O'Neill et al., 2024); and conditioning policies on semantic task instructions and environment context via embeddings from large VLMs (Black et al., 2024; Team, 2025). Among these, DMPs (Hu et al., 2024; Ijspeert et al., 2013, 2002; Saveriano et al., 2023) parametrize a motion policy through dynamical systems that predict the desired velocity or acceleration based on the system's current state. By grounding the formulation in dynamical systems, researchers can leverage established tools from nonlinear system theory (Khalil, 2002) to analyze and ensure-by-design convergence properties in motion primitive—such as global asymptotic stability (Ijspeert et al., 2013; Kober and Peters, 2009; Pérez-Dattari et al., 2024; Pérez-Dattari and Kober, 2023; Rana et al., 2020; Urain et al., 2020; Zhang et al., 2022b) or orbital stability (Abu-Dakka et al., 2024, 2021; Ijspeert et al., 2002; Khadivar et al., 2021; Kober and Peters, 2009; Nah et al., 2025; Urain et al., 2020; Wensing and Slotine, 2017; Zhi et al., 2024). This is not typically the case for other ML-based motion policies like RNNs or Diffusion Policys (DPs) (Black et al., 2024; Chi et al., 2023; O'Neill et al., 2024; Team, 2025). Such approaches—often referred to as SMPs—are robust to perturbations, disturbances, and model mismatches, as the motion policy continuously steers the system back to the desired reference. This also enhances data efficiency, a trait that is increasingly important as robots take on a broader range of tasks.

An important subclass of DMP strategies addresses tasks that require continuous, non-resting motion—those for which rest-to-rest trajectories are neither representative nor sufficient. Canonical examples include wiping a surface, swimming, or walking, where motion generation must produce sustained activity across cycles. These so-called rhythmic or periodic DMPs have spurred extensive research, both within the traditional dynamical systems formulation (Abu-Dakka et al., 2024; Hu et al., 2024; Ijspeert et al., 2013, 2002; Kober and Peters, 2009; Kramberger et al., 2018; Nah et al., 2025; Saveriano et al., 2023; Wensing and Slotine, 2017) and in more recent methods combining simple latent-space limit cycles with learned diffeomorphic mappings (Khadivar et al., 2021; Urain et al., 2020; Zhi et al., 2024). Still, despite these advances, existing approaches struggle to reproduce non-trivial trajectories—especially those with sharp transitions, high curvature, or discontinuous velocity profiles, which are common in real-world rhythmic tasks. Overcoming this typically requires many demonstrations, thus strongly limiting their applicability in practical settings.

Such limitations are exacerbated by the incapability of classical deterministic DMPs to generalize across tasks (Jaquier et al., 2025): a fresh or fine-tuned model must be trained for every new motion or task (Saveriano et al., 2023). Although several studies have introduced task-conditioned variants—such as conditioning on encoded visual observations (Bahl et al., 2020; Mohammadi et al., 2024) or adopting probabilistic DMPs formulations (Pekmezci et al.,

2024; Saveriano et al., 2023; Seker et al., 2019)—these methods often yield incoherent trajectories when presented with tasks they did not explicitly encounter during training (Jaquier et al., 2025), even when those tasks lie within the original training distribution.

So, despite their promise of being an alternative to data-intensive learning strategies, DMPs ultimately require a substantial amount of data and a complex training process when tasks are varied and trajectories are not straightforward. Instead, the ability to generate purposeful motions in zero-shot settings for unseen tasks will be essential on the path towards truly generalist autonomous robots in the future.

In this paper, we introduce OSMPs, a framework, visualized in Fig. 12.1, designed to address the limitations of existing rhythmic motion primitives by learning an expressive, orbitally stable limit cycle capable of capturing elaborate periodic behaviors. Our approach imposes a dynamic inductive bias by shaping the latent space according to a supercritical Hopf bifurcation oscillator; a well-studied system in nonlinear dynamics (Khadivar et al., 2021; Nah et al., 2025; Strogatz, 2018) that has remained unexplored in the context of machine learning. This core dynamical prior is complemented by a novel bijective Euclideanizing-flow encoder, extending the Real NVP architecture (Dinh et al., 2017; Rana et al., 2020).

Under mild architectural assumptions, we prove OSMPs are almost-globally transverse contracting, so every trajectory converges exponentially—not merely asymptotically—to the learned limit cycle. A tailored loss suite binds the cycle's shape and speed to the demonstration, eliminating the long-standing mismatch between a stable latent orbit and a highly curved, nonlinear sample. Thus, a single demonstration already yields an effective policy, drastically outperforming the data efficiency of DPs (Chi et al., 2023). A novel conditioning–interpolation loss then drives smooth, zero-shot transitions between related tasks: for example, the model continuously morphs between reverse and forward turtle-swimming gaits after seeing only those two exemplars. This ability sharply reduces the need to densely sample the motion task space and further increases data efficiency. We also introduce solutions for synchronizing multiple primitives in their phase and, without retraining, affinely scale, translate, or otherwise modulate the learned velocity field, turning rhythmic DMPs into a practical, data-efficient controller for complex robots.

We rigorously validate our approach in both simulation and on hardware. Quantitative and qualitative benchmarks compare it with classical neural models (MLPs, RNNs, NODEs), state-of-the-art motion policies like DP (Chi et al., 2023), and diffeomorphic-encoder methods with stable latent dynamics, including Imitation Flow (iFlow) (Urain et al., 2020) and Stable Periodic Diagrammatic Teaching (SPDT) (Zhi et al., 2024). Experiments show our OSMPs accurately track periodic trajectories on diverse robotic platforms—UR5 and KUKA arms, the Helix soft robot, and a hybrid underwater "crush turtle" robot. Expressed as autonomous dynamical systems, they deliver more compliant, natural behavior than time-indexed, error-based controllers. We further demonstrate in both simulations and real-world experiments in-phase synchronization of multiple OSMPs and, via encoder conditioning, smooth interpolation among several distinct motion tasks with a single motion policy.

12.1.1 Methodology in a Nutshell

Below, we provide a brief overview of the OSMP methodology and architecture, as depicted in Fig. 12.2. Further details are provided in Sec. 12.2. In the DMP framework, an OSMP

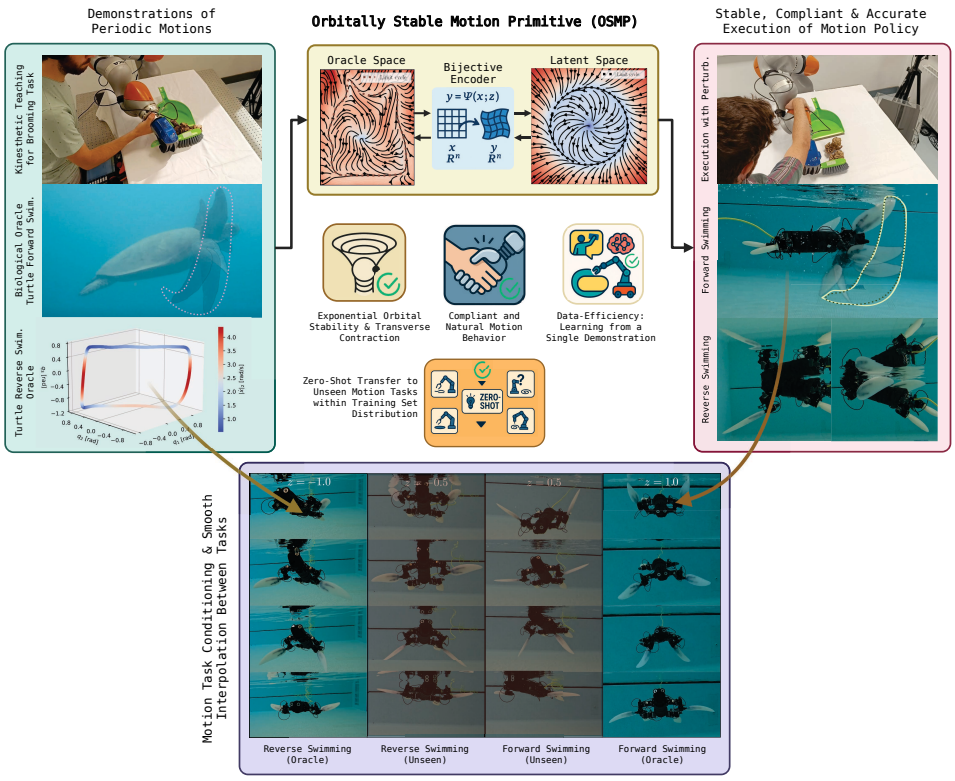


Figure 12.1: **Overview of Features and Characteristics of OSMPs.** OSMPs enable the learning of periodic motions from demonstrations while guaranteeing exponential orbital stability & transverse contraction, instilling compliant and natural motion behavior, and outstanding data efficiency as they are able to learn complex motion behaviors from just one demonstration. Furthermore, a motion task conditioning allows the same motion policy to exhibit multiple distinct behaviors while a specially crafted loss term encourages smooth interpolation between motion tasks seen during training leading to meaningful motion behaviors on task unseen during training (i.e., zero-shot transfer).

outputs the desired system velocity as $\dot{x} = f(x; z)$, with $x \in \mathbb{R}^n$ the configuration and z the motion-task conditioning. The computation proceeds in three steps: (i) map x into a latent coordinate $y \in \mathbb{R}^n$ using a z-conditioned bijective encoder (a learned diffeomorphism); (ii) evaluate the designed latent dynamics to obtain $\dot{y}$; and (iii) project $\dot{y}$ back to the original space via the encoder's inverse Jacobian. While the architecture can be trained under various regimes (e.g., reinforcement learning), this paper focuses on imitation learning—specifically, behavior cloning—where both the latent representation y and the predicted velocity $\dot{x}$ are supervised.

12.1.2 Related Work

While there is a long history of research on both discrete and rhythmic/periodic DMPs (Abu-Dakka et al., 2024; Hu et al., 2024; Ijspeert et al., 2013, 2002; Kober and Peters, 2009;

Kramberger et al., 2018; Nah et al., 2025; Saveriano et al., 2023; Wensing and Slotine, 2017), the expressive power of classical DMPs (Ijspeert et al., 2013, 2002; Kober and Peters, 2009; Wensing and Slotine, 2017) is limited, preventing them from learning highly complex and intricate trajectories.

Recently, however, an exciting research direction has emerged that combines diffeomorphisms into a latent space—learned using ML techniques—with relatively simple, analytically tractable (e.g., linear) latent space dynamics to enhance expressiveness while preserving stability and convergence guarantees (Pérez-Dattari and Kober, 2023; Rana et al., 2020; Urain et al., 2020; Zhang et al., 2022b). Most of these works focus on point-to-point motions and aim to ensure GAS (Pérez-Dattari et al., 2024; Pérez-Dattari and Kober, 2023; Rana et al., 2020; Zhang et al., 2022b), although there have also been several works combining diffeomorphic encoders with rhythmic latent dynamics for learning periodic motions from demonstration (Khadivar et al., 2021; Urain et al., 2020; Zhi et al., 2024). However, in the existing methods, either the chosen architecture for the bijective encoder lacks expressiveness (Khadivar et al., 2021; Urain et al., 2020), the method training is very sensitive to the initial neural network parameter (Urain et al., 2020), the method does not learn the demonstrated velocities but only the general direction of motion (Zhi et al., 2024), or cannot accurately learn very complex oracle shapes (Zhi et al., 2024).

Although the proposed model architecture is similar to that of Zhi et al. (2024), our training pipeline differs substantially: we incorporate an imitation loss that teaches the model the demonstrated velocities, replace the Hausdorff-distance objective with a limit-cycle matching loss better suited to complex or discontinuous paths, optionally guide latent polar angles with a time-alignment term to capture highly curved, possibly concave, contours, and regularize workspace velocities outside the demonstration to improve numerical stability during inference. Finally, we allow a parametrization of the polar angular velocity with a neural network, allowing the learning of complex velocity profiles along the limit cycle without compromising the strong contraction guarantees.

Furthermore, the above-mentioned methods do not offer solutions for many practical issues, such as synchronizing multiple systems—a common requirement in locomotion or bimanual manipulation (Gams et al., 2015)—or to shape the learned velocity online.

We provide a comparison with relevant existing methods in Table 12.1.

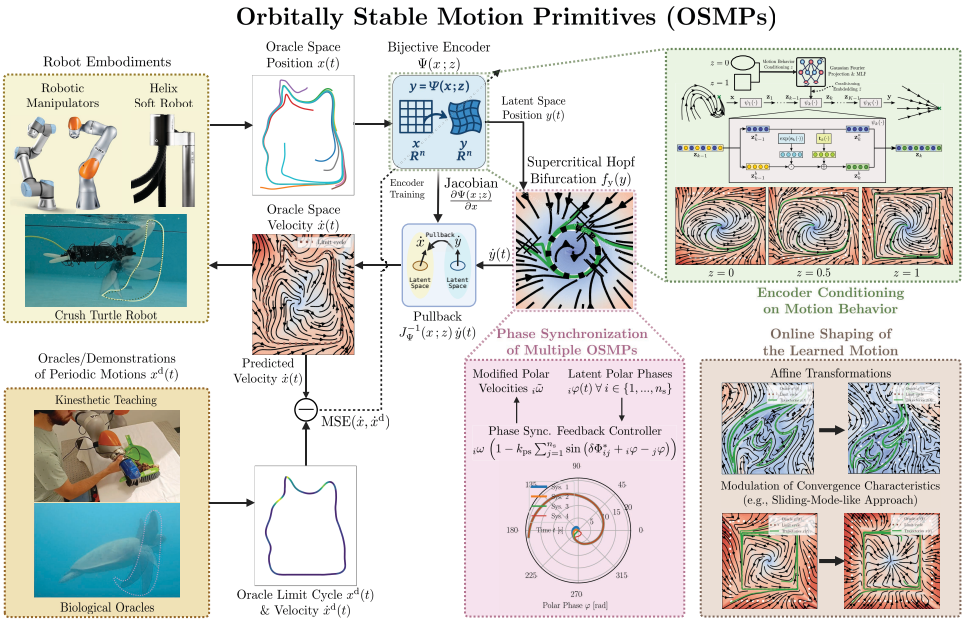


Figure 12.2: **Methodology of OSMPs.** Periodic motions can be learned from demonstration via a DMP that combines a bijective encoder with latent space dynamics defined by a supercritical Hopf bifurcation. After encoding the current oracle space position of the system into latent space and predicting the latent space velocity, the pullback operator projects the velocity back into oracle space and represents a motion reference for the various robot embodiments. During training, we enforce both the predicted velocity and the exhibited limit cycle to match the demonstrations that are, for example, provided via kinesthetic teaching or based on biological oracles. Multiple contributions increase the practical usefulness of the proposed methods, which include a technique to synchronize multiple OSMPs in phase, an approach to shape the learned motion online without requiring retraining via affine transformations or even modulating the convergence characteristics, and a methodology for conditioning the encoder on the oracle which allows the OSMP to capture multiple distinct motion behaviors and even smoothly interpolate between them. The depiction of the Euclideanizing flows encoder architecture is adapted from Rana et al. (2020).

Table 12.1: **Comparison of characteristics of proposed methods against relevant baseline methods.** In some cases, we denote with a x^* that the feature could (probably) be developed for the respective method, but was not presented in the original paper. *Note:* The CLF-CBF-NODE only guarantees convergence to a target trajectory as predicted by the learned NODE. However, the learned NODE is not guaranteed to exhibit a stable limit cycle behavior.

Method	Model Expressiveness	Orb. Stability Guarantees	Velocity Imitation	N-Policies Sync.	Task Conditioning	Smooth Task Interpolation
MLP & RNN	Moderate	x	Noisy	x	✓	x
NODE	Moderate	x	✓	x	✓	x
DP (Chi et al., 2023)	High	x	Noisy	x	✓	x
Classical Rhythmic DMPs	Limited	✓	✓	✓	7	x
HB-GMR (Khadivar et al., 2021)	Limited	✓	x	x*	x	x
Imitation Flow (iFlow) (Urain et al., 2020)	Moderate	✓	✓	x*	x*	x
CLF-CBF-NODE (Nawaz et al., 2024)	Moderate	x	✓	x	x*	x
SPDT (Zhi et al., 2024)	Moderate	✓	x	x*	x	x
OSMP (ours)	Moderate	✓	✓	✓	✓	✓

12.2 Methodology

In this work, we introduce OSMPs, which can be trained to capture complex periodic motions from demonstrations while ensuring convergence to a limit cycle that aligns with a predefined oracle. To accomplish this, we build on previous research (Rana et al., 2020; Zhi et al., 2024) that combines learned bijective encoders with a prescribed motion behavior in latent space. These latent dynamics generate velocities or accelerations that are subsequently mapped back into the oracle space via a pullback operator (Zhang et al., 2022b)—in the case of a bijective encoder, this operator is the inverse of the encoder's Jacobian. In this formulation, the motion in latent space exhibits key convergence properties, such as Global Asymptotic Stability (GAS) (Pérez-Dattari and Kober, 2023; Rana et al., 2020; Sochopoulos et al., 2024) or Orbital Stability (OS) (Zhi et al., 2024), while the learned encoder provides the necessary expressiveness to capture complex motions and transfers the convergence guarantees from latent to oracle space through the established diffeomorphism.

However, compared to existing work (Zhi et al., 2024), we introduce several crucial modifications that enhance both the performance and practical utility of the proposed method: (1) we develop a limit cycle matching loss to reduce the discrepancy between the learned limit cycle and the periodic oracle; (2) we design strategies to modulate the learned velocity field online without the need for retraining—for instance, to adjust the convergence behavior; (3) we establish a procedure to synchronize the phase of multiple OSMPs; and (4) we condition the encoder on a task, enabling a single OSMP to exhibit multiple distinct motion behaviors. Moreover, we introduce loss terms that allow the trained OSMP to smoothly interpolate between the learned motion behaviors—something that has not been possible before.

12.2.1 Orbitally Stable Motion Primitives

The dynamical motion policy $\dot{x} = f(x; z)$ is typically defined in task space but can also be defined in other Cartesian or generalized coordinates (e.g., joint space). Therefore, we will, in the following, refer to these coordinates as *Oracle space*. In this work, we are specifically interested in cases where we train $f(x)$ to learn periodic motions. Then, the smooth diffeomorphism between the oracle and latent space is made via a bijective encoder $\Psi : \mathbb{R}^n \to \mathbb{R}^n$, which maps positional states $x \in \mathbb{R}^n$ into the latent states $y \in \mathbb{R}^n$. Optionally, this encoding is conditioned on a continuous variable $z \in \mathbb{R}$ as a homotopy such that $y = \Psi(x; z)$. Furthermore, we construct Ψ such that it is invertible and a closed-form inverse function $\Psi^{-1} : y \mapsto x$ allows us to map from latent space back into oracle space. In latent space, we apply dynamics $\dot{y} = f_y(y)$ that exhibit a stable limit cycle behavior. In summary, the orbitally stable motion primitive is given as

$$\dot{x} = f(x; z) = f_s(x) J_\Psi^{-1}(x; z)\, f_y(\Psi(x; z)), \tag{12.1}$$

where $J_\Psi = \frac{\partial \Psi(x;z)}{\partial x}$ defines the Jacobian of the encoder. The function $f_s(x) : \mathbb{R}^n \to \mathbb{R}_{>0}$ scales the velocity magnitude and is implemented as $f_s(x) = e^{\mathrm{MLP}(x)} + \varepsilon$, where $\varepsilon \in \mathbb{R}_{>0}$ is a small value (Rana et al., 2020). As Ψ is a diffeomorphism (w.r.t. x and y), the motion policy is orbitally stable by construction (Rana et al., 2020; Zhi et al., 2024).

Diffeomorphic Encoder

We consider a bijective encoder $\Psi_\theta : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$, which maps positional states $x \in \mathbb{R}^n$ into the latent states $y \in \mathbb{R}^n$ conditioned on $z \in \mathbb{R}$, where we assume that $n \in \mathbb{N} \geq 2$. The encoder $y = \Psi_\theta(x; z)$ adopting the Euclideanizing flows (Dinh et al., 2017; Rana et al., 2020) architecture is parametrized by the learnable weights $\theta \in \mathbb{R}^{n_\theta}$ and consists of n_b blocks, where each block is analytically invertible. If a conditioning is used, z is first lifted into an embedding $\tilde{z}$, which is subsequently used to condition each block. More details on the encoder architecture can be found in the pioneering work (Dinh et al., 2017; Rana et al., 2020) and Section 12.2.2.

Latent Dynamics

In latent space, we consider the 1st-order dynamics of a supercritical Hopf bifurcation (Khadivar et al., 2021; Nah et al., 2025; Strogatz, 2018; Zhi et al., 2024)

$$\dot{y} = \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_{3:n} \end{bmatrix} = f_y(y) = \begin{bmatrix} -\omega(y)\, y_2 + \alpha \left(1 - \frac{y_1^2 + y_2^2}{R^2}\right) y_1 \\ +\omega(y)\, y_1 + \alpha \left(1 - \frac{y_1^2 + y_2^2}{R^2}\right) y_2 \\ -\beta\, y_{3:n} \end{bmatrix}, \tag{12.2}$$

where $\omega(y) = f_\omega(\mathrm{atan2}(y_2, y_1)) > 0$ determines the angular velocity. Here, the dynamics of y_1 and y_2 describe the Cartesian-space behavior of a simple limit cycle whose behavior in polar coordinates $y_\mathrm{pol} = \begin{bmatrix} r & \varphi & y_{3:n}^\top \end{bmatrix}^\top$ is expressed as

$$\dot{y}_\mathrm{pol} = \begin{bmatrix} \dot{r} \\ \dot{\varphi} \\ \dot{y}_{3:n} \end{bmatrix} = f_\mathrm{pol}(y_\mathrm{pol}) = \begin{bmatrix} \alpha \left(1 - \frac{r^2}{R^2}\right) r \\ f_\omega(\varphi) \\ -\beta\, y_{3:n} \end{bmatrix}, \tag{12.3}$$

where $f_\omega(\varphi) : [-\pi, \pi) \to \mathbb{R}_{>0}$ computes the positive angular velocity as a function of the polar angle. Often, in particular, when employing $f_\mathrm{s}(x) \neq 1$, it can also be set to a constant: $\omega = 1$. If not, we define in practice $\omega(y) = f_\omega(\bar{y}_{1:2}) = e^{\mathrm{MLP}(\bar{y}_{1:2})} + \epsilon_\omega$, where $\bar{y}_{1:2} = \begin{bmatrix} \frac{y_1}{\sqrt{y_1^2 + y_2^2}} & \frac{y_2}{\sqrt{y_1^2 + y_2^2}} \end{bmatrix}^\top \in \mathbb{R}^2$ with $\epsilon_\omega > 0$.

$\alpha > 0$ and $\beta > 0$ are positive gains that determine how fast the system converges onto the limit cycle. When learning the dynamics, we choose $\alpha = \beta = 1$. $R \in \mathbb{R}_{>0}$ expresses the radius of the limit cycle in latent space. Again, it is sufficient to choose $R = 1$ or $R = 0.5$.

Map from Polar to Cartesian Coordinates The map $h_\mathrm{p2y}(y_\mathrm{pol}) : \mathbb{R}^n \to \mathbb{R}^n$ from latent polar coordinates to latent Cartesian coordinates and its inverse $h_\mathrm{y2p}(y) = h_\mathrm{p2y}^{-1}(y)$ is given by

$$y = h_\mathrm{p2c}(y_\mathrm{pol}) = \begin{bmatrix} r\cos(\varphi) \\ r\sin(\varphi) \\ y_{3:n} \end{bmatrix}, \qquad y_\mathrm{pol} = h_\mathrm{c2p}(y) = \begin{bmatrix} \sqrt{y_1^2 + y_2^2} \\ \mathrm{atan2}(y_2, y_1) \\ y_{3:n} \end{bmatrix}. \tag{12.4}$$

Then, the Jacobian of the Polar-to-Cartesian map $\frac{\partial h_\mathrm{p2c}}{\partial y_\mathrm{pol}} \in \mathbb{R}^{n \times n}$ is given by

$$\frac{\partial h_\mathrm{p2c}}{\partial y_\mathrm{pol}} = \begin{bmatrix} \cos(\varphi) & -r\sin(\varphi) & 0_{1\times(n-2)} \\ \sin(\varphi) & r\cos(\varphi) & 0_{1\times(n-2)} \\ 0_{(n-2)\times 1} & 0_{(n-2)\times 1} & \mathbb{I}_{n-2} \end{bmatrix} \tag{12.5}$$

We can also substitute $y_{\text{pol}} = y$ into the Jacobian and determine its inverse

$$\frac{\partial h_{\text{p2c}}}{\partial y_{\text{pol}}} = \begin{bmatrix} \frac{y_1}{\sqrt{y_1^2+y_2^2}} & -y_2 & 0_{1\times(n-2)} \\ \frac{y_2}{\sqrt{y_1^2+y_2^2}} & y_1 & 0_{1\times(n-2)} \\ 0_{(n-2)\times 1} & 0_{(n-2)\times 1} & \mathbb{I}_{n-2} \end{bmatrix}, \quad \frac{\partial h_{\text{p2c}}}{\partial y_{\text{pol}}}^{-1} = \begin{bmatrix} \frac{y_1}{\sqrt{y_1^2+y_2^2}} & \frac{y_2}{\sqrt{y_1^2+y_2^2}} & 0_{1\times(n-2)} \\ -\frac{y_2}{y_1^2+y_2^2} & \frac{y_1}{y_1^2+y_2^2} & 0_{1\times(n-2)} \\ 0_{(n-2)\times 1} & 0_{(n-2)\times 1} & \mathbb{I}_{n-2} \end{bmatrix}, \tag{12.6}$$

which are both full-rank for $r = \sqrt{y_1^2 + y_2^2} > 0$.

12.2.2 Training

We consider a dataset $\langle T, X^{\text{d}}, \dot{X}^{\text{d}}, Z\rangle$ as a tuple between timestamps $T = \langle t(1), \dots, t(k), \dots, t(N)\rangle$ positions $X^{\text{d}} = \langle x^{\text{d}}(1), \dots, x^{\text{d}}(k), \dots, x^{\text{d}}(\text{N})\rangle$, the corresponding, demonstrated velocities $\dot{X}^{\text{d}} = \langle \dot{x}^{\text{d}}(1), \dots, \dot{x}^{\text{d}}(k), \dots, \dot{x}^{\text{d}}(\text{N})\rangle$ and an optional conditioning $Z = \langle z(1), \dots, z(k), \dots, z(\text{N})\rangle$, where $k \in \mathbb{N}_{\text{N}} = \{1, 2, \dots, N\}$.

Bijective Encoder

The bijective encoder based on Euclideanizing flow (Rana et al., 2020) / Real NVP (Dinh et al., 2017) uses coupling layers with the scaling and translation functions parametrized by Random Fourier Features Networkss (RFFNs) that integrate ELU activation functions and a hidden dimension of 100. The number of encoder layers/blocks varies by the complexity of the task and ranges from 10 for simple tasks to 25 for very complex tasks. At the start of the training, the encoder is initialized as an identity mapping.

(Angular) Velocity Scaling Network

The optional velocity scaling network $f_{\text{s}}(x)$ is parametrized by a three-to-five-layer MLP(x) (depending on the nonlinearities and discontinuities that the demonstration velocity profile exhibits) with a hidden dimension of 128 and LeakyReLU activation functions.

Similarly, the angular velocity network $\omega(y) = f_\omega(\bar{y}_{1:2}) = e^{\text{MLP}(\bar{y}_{1:2})} + \epsilon_\omega$, where $\bar{y}_{1:2} = \begin{bmatrix} \frac{y_1}{\sqrt{y_1^2+y_2^2}} & \frac{y_2}{\sqrt{y_1^2+y_2^2}} \end{bmatrix}^\top \in \mathbb{R}^2$ with $\epsilon_\omega = 10^{-6}$, is parametrized by a five-layer MLP with a hidden dimension of 128 and LeakyReLU activation functions.

Loss Functions

The total training loss function is then given by

$$\mathcal{L} = \underbrace{\zeta_{\text{vi}}\mathcal{L}_{\text{vi}}}_{\text{Vel. Imitation}} + \underbrace{\zeta_{\text{lcm}}\mathcal{L}_{\text{lcm}}}_{\text{Limit Cycle Matching}} + \underbrace{\zeta_{\text{tgd}}\mathcal{L}_{\text{tgd}}}_{\text{Time Guidance}} + \underbrace{\zeta_{\text{er}}\mathcal{L}_{\text{er}}}_{\text{Encoder Reg.}} + \underbrace{\zeta_{\text{vr}}\mathcal{L}_{\text{vr}}}_{\text{Vel. Reg.}} + \underbrace{\zeta_{\text{sci}}\mathcal{L}_{\text{sci}}}_{\text{Cond. Interpolation}} \tag{12.7}$$

where $\zeta_{\text{vi}}, \zeta_{\text{lcm}}, \zeta_{\text{tgd}}, \zeta_{\text{er}}, \zeta_{\text{vr}}, \zeta_{\text{sci}} \in \mathbb{R}$ are scalar loss weights. $\mathcal{L}_{\text{vi}}$ is a loss term that enforces that the velocity of the motion primitive matches the one demanded by the demonstration at all samples in the demonstration dataset, $\mathcal{L}_{\text{lcm}}$ ensures that the encoded demonstration positions lie on the latent limit cycle. The time-guidance loss $\mathcal{L}_{\text{tgd}}$ can support the limit cycle matching loss for highly curved demonstrations. The term $\mathcal{L}_{\text{er}}$ regularizes the encoder by penalizing deviations from the identity encoder. The optional $\mathcal{L}_{\text{sci}}$ gives rise to smooth interpolation between different encoder conditioning. The discretionary velocity regularization loss ζ_{vr} increases the numerical stability by penalizing very high velocities. In the following, we define each loss term formally.

Velocity Imitation Loss Analog to the literature on stable point-to-point motion primitives (Rana et al., 2020), the predicted oracle space velocity is supervised by a smooth ℓ_1 loss (Girshick, 2015), which computes a squared term if the absolute error falls below β_{ℓ_1} and the ℓ_1 term otherwise, and can be formally defined as

$$\mathcal{L}_{\text{vi}} = \frac{1}{N}\sum_{k=1}^{N}\begin{cases}\frac{\left(f(x^{\text{d}}(k);z(k))-\dot{x}^{\text{d}}(k)\right)^2}{2\beta_{\ell_1}}, & \text{if } \left\|f(x^{\text{d}}(k);z(k))-\dot{x}^{\text{d}}(k)\right\|_1 < \beta_{\ell_1} \\ \left\|f(x^{\text{d}}(k);z(k))-\dot{x}^{\text{d}}(k)\right\|_1 - \frac{\beta_{\ell_1}}{2}, & \text{otherwise}\end{cases}, \tag{12.8}$$

where we choose $\beta_{\ell_1} = 1$.

Limit Cycle Matching Loss Next, we consider a subset of the demonstrations $\mathcal{P} \subset \mathbb{N}_{\text{N}}$ that exhibit a periodic motion. To guarantee the OS of the learned system, we need to make sure that the learned limit cycle matches the periodic demonstration. For this purpose, we design a *limit cycle matching* loss $\mathcal{L}_{\text{lcm}}$ in latent space

$$y_{\text{p}}(k) = \begin{bmatrix}\sqrt{y_1^2(k)+y_2^2(k)} \\ y_{3:n}\end{bmatrix} \in \mathbb{R}^{n-1}, \quad y_{\text{p}}^{\text{d}}(k) = \begin{bmatrix}R \\ 0_{n-2}\end{bmatrix}, \quad \mathcal{L}_{\text{lcm}} = \sum_{k\in\mathcal{P}} \frac{\left\|y_{\text{p}}^{\text{d}}(k) - y_{\text{p}}(k)\right\|_2^2}{|\mathcal{P}|}, \tag{12.9}$$

where $|\mathcal{P}|$ is the cardinality of $\mathcal{P}$, and $y = \Psi(x^{\text{d}}; z) \in \mathbb{R}^n$ is the latent encoding.

Time Reference Guidance Loss For strongly curved oracles, we have found it advantageous to use the oracle's time parameterization to steer how its positions map onto the latent polar angle. Doing so spreads the oracle samples uniformly around the latent-space limit cycle, preventing the encoders from bunching up in one angular sector while leaving other portions of the cycle without corresponding oracle points.

First, we compute for each position contained in the rhythmic demonstration a desired latent polar angle based on the normalized time reference. Simultaneously, we evaluate the actual encoded polar angle of $x^{\text{d}}(k)$ as

$$\varphi^{\text{d}}(k) = \varphi_0 + 2\pi\frac{t(k)}{P}, \qquad \varphi(k) = \operatorname{atan2}\left(\Psi(x(k);z(k))_2, \{\Psi(x(k);z(k))\}_1\right), \tag{12.10}$$

where φ_0 is the polar angle anchor and P is the period of the rhythmic demonstration. Subsequently, we define a positive alignment loss between the

$$\mathcal{L}_{\text{tgd}} = \sum_{k\in\mathcal{P}} \frac{\max\left(\left|\operatorname{mod}\left(\varphi^{\text{d}}(k) - \varphi(k) + \pi,\ 2\pi\right) - \pi\right| - m_{\text{tgd}}, 0\right)^2}{|\mathcal{P}|}, \tag{12.11}$$

where $m_{\text{tgd}} \in \mathbb{R}_{>0}$ is the allowed margin between the time reference and the actual polar phase and the function $n_{e_\varphi}(e_\varphi) = \operatorname{mod}\left(e_\varphi(k)+\pi,\ 2\pi\right) - \pi$ normalizes the polar angle error $e_\varphi(k) = \varphi^{\text{d}}(k) - \varphi(k)$ into the interval $[-\pi, \pi)$.

Encoder Regularization Similar to Zhi et al. (2024), we employ an encoder regularization loss $\mathcal{L}_{\Psi,\text{reg}}$ that penalizes deviations from an identity encoder $y = \Psi(x) = x$. We draw

in each epoch N random positions samples $x(k) \sim \mathcal{U}(x_{\min}, x_{\max}) \forall\, k \in \mathbb{N}_N$ from a uniform distribution within the workspace $[x_{\min}, x_{\max}]$ of the system. Then, the loss is computed as

$$\mathcal{L}_{\text{er}} = \sum_{k=1}^{N} \frac{\|x - \Psi(x; z)\|}{N}. \tag{12.12}$$

Velocity Regularization The velocity-imitation loss $\mathcal{L}_{\text{vi}}$ constrains the predicted velocities only along the demonstration trajectory. When demonstrations are sparse and clustered, large regions of the workspace receive no direct supervision on velocity magnitude, even though orbital stability and transverse contraction are still guaranteed. However, in practice, large predicted velocities frequently lead to numerical instability. Therefore, it can be helpful to regularize the predicted velocities.

For this purpose, we draw in each epoch N random positions samples

$$x(k) \sim \mathcal{U}(x_{\min}, x_{\max}) \forall\, k \in \mathbb{N}_N \tag{12.13}$$

from a uniform distribution within the workspace $[x_{\min}, x_{\max}]$ of the system. Then, the loss is computed as

$$\mathcal{L}_{\text{vr}} = \sum_{k=1}^{N} \frac{\max(\|f(x(k); z)\|_2 - m_{\text{vr}}, 0_N)}{N}, \tag{12.14}$$

where $m \in \mathbb{R}_{\geq 0}$ is the margin. In practice, we choose m_{vr} to be 50% higher than the maximum velocity magnitude in the dataset in order not to conflict with the $\mathcal{L}_{\text{vi}}$ objective.

Smooth Conditioning Interpolation Loss Next, optionally, we can add a loss term that encourages a smooth interpolation of the learned limit cycle between conditionings z. We assume that all conditionings in the dataset $z(k) \in \mathcal{Z}$, where $\mathcal{Z} = \{z(1), \ldots, z(k), \ldots, z(N)\}$, are bounded in the interval $[z_{\min}, z_{\max}]$. Now, consider a convex hull

$$\text{conv}(\mathcal{Z}) = [\min(\mathcal{Z}), \min(\mathcal{Z})] = [z_{\min}, z_{\max}].$$

Next, we draw N_{sci} random conditionings from a uniform distribution: $\tilde{z}(j) \sim \mathcal{U}(\text{conv}(\mathcal{Z})) \in \mathbb{R}$ with $j \in \mathbb{N}_{N_{\text{sci}}}$. Additionally, we also generate N_{sci} samples on the latent limit cycle by uniformly sampling polar angles $\varphi(j) \sim [-\pi, \pi)$ and subsequently first map into Cartesian latent coordinates and then into oracle space using the inverse encoder

$$y(j) = \begin{bmatrix} R\cos(\varphi(j)) & R\sin(\varphi(j)) & 0_{n-2}^{\top} \end{bmatrix}^{\text{T}}, \quad \tilde{x}(j) = \Psi^{-1}(y(j); \tilde{z}(j)). \tag{12.15}$$

Now, we define the floor $\lfloor \cdot \rfloor$ and ceil $\lceil \cdot \rceil$ functions that round down, or up to the next conditioning $z \in \mathcal{Z}$ in the dataset

$$\lfloor \tilde{z} \rfloor = \max\{z \in \mathbb{Z} \mid z \leq \tilde{z}\}, \qquad \lceil \tilde{z} \rceil = \min\{z \in \mathbb{Z} \mid z \geq \tilde{z}\}. \tag{12.16}$$

Then, the target for $\tilde{x}(j)$ that represents a smooth linear interpolation between conditioning $\lfloor \tilde{z} \rfloor$ and $\lceil \tilde{z} \rceil$ is given by

$$\tilde{x}^*(j) = \lfloor \tilde{x}(j) \rfloor + \frac{\tilde{z}(j) - \lfloor \tilde{z}(j) \rfloor}{\lceil \tilde{z}(j) \rceil - \lfloor \tilde{z}(j) \rfloor} (\lceil \tilde{x}(j) \rceil - \lfloor \tilde{x}(j) \rfloor) \tag{12.17}$$

where

$$\lfloor \tilde{x}(j) \rfloor = \Psi^{-1}(y(j); \lfloor \tilde{z}(j) \rfloor), \qquad \lceil \tilde{x}(j) \rceil = \Psi^{-1}(y(j); \lceil \tilde{z}(j) \rceil). \tag{12.18}$$

Finally, the smooth conditioning interpolation loss can be formulated as

$$\mathcal{L}_{\text{sci}} = \sum_{j=1}^{N_{\text{sci}}} \frac{(\tilde{x}^*(j) - \tilde{x}(j))^2}{N_{\text{sci}}}. \tag{12.19}$$

12.2.3 Inference

Maintaining discrete-time stability demands that the OSMP—or any DMP—runs at sufficiently high control rates. This requirement becomes even tougher when computational resources are limited, as in our turtle-swimming setup where the OSMPs runs on a Raspberry Pi 5. To minimise latency, we sought to shorten the OSMP's inference time by exploiting PyTorch's compilation and export utilities. Unfortunately, most current PyTorch compilers/exporters are incompatible with autograd, which we still need at inference to obtain the encoder Jacobian $J_\Psi = \frac{\partial}{\partial x}\Psi(x; z)$. Consequently, we explored modern options in the `torch.func` namespace—including combinations of `vmap` with the forward and reverse functional Jacobian operators (`jacfwd`, `jacrev`) and the vector-Jacobian product (`vjp`). Our analysis, presented in the Sec. 12.3.7, shows that simple two-point finite-difference schemes for the Jacobian compile and export cleanly, run quickly, and yield Jacobians whose accuracy is very close to the analytic solution. In practice, we use an (absolute) step size $\delta_x = 5\,10^{-4}$ such that

$$J(x; z) \approx \left[\frac{\Psi(x+\delta_x\, e_1; z) - \Psi(x; z)}{\delta_x} \quad \dots \quad \frac{\Psi(x+\delta_x\, e_j; z) - \Psi(x; z)}{\delta_x} \quad \dots \quad \frac{\Psi(x+\delta_x\, e_n; z) - \Psi(x; z)}{\delta_x} \right], \tag{12.20}$$

where $e_j \in \mathbb{R}^n$ is the jth canonical basis vector in x-coordinates. This allows us to exploit *AOTInductor*, a specialized version of *TorchInductor*, to export a compiled executable, which runs at up to 15 kHz on the M4 Max CPU - a 9x increase over the standard eager inference mode.

In case the Jacobian of the encoder $J_\Psi(x; z)$ exhibits close-to-singular values, the numerical stability of the inference, can be improved by computing the inverse as $J_\Psi^{-1}(x; z) \equiv (J_\Psi(x; z) + \epsilon_{\text{inv}}\, \mathbb{I}_n)^{-1}$, where, for example, $\epsilon_{\text{inv}} = 10^{-6}$.

12.2.4 Online Shaping of the Learned Motion

In order to improve the practicality of using the learned orbital motion primitives, we introduce in this section approaches that allow us to modulate the learned velocity field to adjust the task or modify its characteristics without having to retrain the OSMP.

First, we introduce variables that allow us to spatially translate and scale the learned velocity field

$$\dot{x}(t) = \tilde{f}(x; z) := s_{\text{f}}\, f\left(\frac{x(t) - x^{\text{o}}}{s_{\text{f}}}; z \right). \tag{12.21}$$

Here, $s_{\text{f}} \in \mathbb{R}_{>0}$ controls the spatial scale of the velocity field. When $s_{\text{f}} = 1$, the executed motion primitive is equal to the learned motion primitive. $x^{\text{o}} \in \mathbb{R}^n$ defines the origin of the velocity field.

However, we are not limited to affine transformations such as translation and scaling. Additionally, we can adjust the period and convergence characteristics of the velocity field online. Specifically, by scaling the polar angular velocity $\omega(\varphi)$ with the factor $s_\omega \in \mathbb{R}_{>0}$, we can either slow-down ($0 < s_\omega < 1$) or speed-up ($s_\omega > 1$) the periodic motion. Furthermore, the convergence of trajectories onto the $\mathbb{S}^1$ limit cycle can be made more or less aggressive by adjusting the convergence gain $k_{\text{conv}} \in \mathbb{R}_{>0}$. Usually, we set $\alpha = \beta = k_{\text{conv}}\, s_\omega$.

Finally, constraints in the oracle or actuation space (e.g., joint limits, environment obstacles) might pose challenges to the deployment of the orbital motion primitive in real-world settings when the system is initialized (far) off the oracle. In these situations, we would not want to start our periodic motion directly, but instead, we would first converge into a neighborhood around the oracle that is collision-free. We devise a strategy that is able to accomplish such behavior by scaling the polar angular velocity $\tilde{\omega}$ as a function $\sigma : \mathbb{R}_{>0} \to \mathbb{R}$ of the distance from the limit cycle d_{lc}

$$d_{\text{lc}} = \sqrt{\frac{\left(\sqrt{y_1^2 + y_2^2} - R\right)^2 + \sum_{i=2}^{n} y_i^2}{n-1}}, \qquad \tilde{\omega} = \sigma(\varphi, d_{\text{lc}}) = \exp\left(-\frac{\max(d_{\text{lc}} - R_{\text{sm}}, 0)^2}{2\,\sigma_{\text{sm}}^2}\right) \omega(\varphi), \quad (12.22)$$

where $d_{\text{lc}} \in \mathbb{R}_{>0}$ the Euclidean distance of the latent state y from the limit cycle normalized by the DOF. The mapping $d_{\text{lc}} \mapsto \tilde{\omega}$ can be intuitively interpreted as follows: in a tube of radius R_{sm} around the limit cycle, we apply the nominal polar angular velocity $\omega(\varphi)$. Outside of that tube, we reduce the angular velocity using a Gaussian function with RMS width $\sigma_{\text{sm}} \in \mathbb{R}_{>0}$. In the limit $d_{\text{lc}} \to \infty$, the polar angular velocity is zero: $\lim_{d_{\text{lc}} \to \infty} \sigma(d_{\text{lc}}) = 0$.

12.2.5 Phase Synchronization of Multiple Motion Primitives

In many real-world applications, it is essential to synchronize the phases of multiple learned (orbital) motion primitives (Gams et al., 2015). For instance, in turtle swimming, the phases of the two limbs must align, while in (human) walking, the periodic movement of the two legs should be offset by π. To address this, we developed a controller that can synchronize the motion of two or more systems. Here, we consider that we trained n_{s} OSMPs. We refer to the latent state of the ith system, where $i \in \mathbb{N}_{n_{\text{s}}}$, as ${}_i y$. The polar phase of each system is given by ${}_i\varphi = \text{atan2}({}_i y_2, {}_i y_1)$. We then construct a symmetric matrix $\delta\Phi^* \in \mathbb{R}^{(n_{\text{s}}-1)\times(n_{\text{s}}-1)}$ that contains the desired phase offsets. For example, $\delta\Phi^*_{ij} = \delta\Phi^*_{ji} \in [-\pi, \pi)$ specifies the desired phase offset between the ith and the jth system. In the case of $\delta\Phi^* = 0^{(n_{\text{s}}-1)\times(n_{\text{s}}-1)}$, we ask the phase difference between all systems to be zero. We then adopt a technique from the field of network synchronization (Dörfler and Bullo, 2014) that allows the alignment of the OSMPs in phase. Namely, we define a feedback controller that acts on the angular velocity of the latent system

$${}_i\tilde{\omega}(\varphi) = {}_i\omega({}_i\varphi)\left(1 - k_{\text{ps}} \sum_{j=1}^{n_{\text{s}}} \sin\left(\delta\Phi^*_{ij} + {}_i\varphi - {}_j\varphi\right)\right), \quad (12.23)$$

where $\omega(\varphi), \tilde{\omega} \in \mathbb{R}$ are the default and modified polar angular velocities of the systems, respectively. $k_{\text{ps}} \in \mathbb{R}_{>0}$ is the phase synchronization gain that determines how quickly the systems synchronize.

12.3 Experiments and Results

12.3.1 OSMPs are Asymptotically Orbitally Stable and Transverse Contracting

In this section, we analyze the convergence characteristics of OSMPs from a theoretical perspective. Specifically, we show that OSMPs exhibit Asymptotic Orbital Stability (AOS) and under which conditions they are transverse contracting and lead to Exponential Orbital Stability (EOS)

Proof of Asymptotic Orbital Lyapunov Stability

In the general setting, we show that OSMPs possess AOS. This property was noted—but not fully proven—in earlier work (Urain et al., 2020; Zhi et al., 2024). Instead, in the following, we formally prove AOS.

Theorem 4. *Let $\alpha, \beta, R > 0$ and $f_s(x) : \mathbb{R}^n \to \mathbb{R}_{>0}$ and z be constant. Then, the dynamics $\dot{x} = f(x; z)$ are asymptotically orbitally stable under the transverse Lyapunov function*

$$V_x(x; z) = \frac{\alpha R^2}{4}\left(y_1^2 + y_2^2 - R^2\right)^2 + \frac{1}{2} y_{3:n}^\top \beta\, y_{3:n}\Big|_{y=\Psi(x;z)}. \tag{12.24}$$

Proof. **Step 1: Proof of Orbital Stability in Polar Latent Dynamics.** Consider the transverse Lyapunov candidate (Manchester, 2011) for the polar latent dynamics $\dot{y}_{\text{pol}} = f_{\text{pol}}(y_{\text{pol}})$

$$V_{\text{pol}}(y_{\text{pol}}) = \alpha \frac{R^2}{4}\left(r^2 - R^2\right)^2 + \frac{1}{2} y_{3:n}^\top \beta\, y_{3:n}. \tag{12.25}$$

We can demonstrate that V_{pol} is a valid Lyapunov candidate with respect to the limit cycle $\mathcal{O}_{\text{pol}} = \{r, \varphi \in \mathbb{R}, y_{3:n} \in \mathbb{R}^{n-2} | r = R, \varphi \in [-\pi, \pi), y_{3:n} = 0_{n-2}\}$ via

$$V_{\text{pol}}(y_{\text{pol}}) = \alpha \frac{R^2}{4}\left(R^2 - R^2\right)^2 = 0, \quad \forall\, y_{\text{pol}} \in \mathcal{O}, \tag{12.26}$$

and

$$V_{\text{pol}}(y_{\text{pol}}) = \frac{\alpha R^2}{4} \underbrace{\left(r^2 - R^2\right)^2}_{\geq 0} + \underbrace{\frac{1}{2} y_{3:n}^\top \beta\, y_{3:n}}_{\geq 0} > 0, \quad \forall y_{\text{pol}} \in \mathbb{R}^n \setminus \mathcal{O}_{\text{pol}}. \tag{12.27}$$

The time derivative of the Lyapunov candidate perpendicular to the orbital flow $\begin{bmatrix} 0 & f_\omega(\varphi) & 0 \end{bmatrix}^\top$ is given by

$$\begin{aligned} \dot{V}_{\text{pol}, f_{\text{pol},\perp}}(y_{\text{pol}}) &= \frac{\partial V_y}{\partial y_{\text{pol}}} f_{\text{pol},\perp} = \begin{bmatrix} -\alpha\left(1 - \frac{r^2}{R^2}\right) r & 0 & \beta\, y_{3:n} \end{bmatrix} \begin{bmatrix} \alpha\left(1 - \frac{r^2}{R^2}\right) r \\ 0 \\ -\beta\, y_{3:n} \end{bmatrix}, \\ &= -\underbrace{\alpha^2 \left(1 - \frac{r^2}{R^2}\right)^2 r^2}_{\geq 0} - \underbrace{y_{3:n}^\top \beta^2\, y_{3:n}}_{\geq 0} < 0, \quad \forall y_{\text{pol}} \in \mathbb{R}^n \setminus \mathcal{O}_{\text{pol}}. \end{aligned} \tag{12.28}$$

with $\dot{V}_{\text{pol},f_{\text{pol},\perp}}(y_{\text{pol}}) = 0 \,\forall y_{\text{pol}} \in \mathcal{O}_{\text{pol}}$.

Step 2: Proof of Orbital Stability in Cartesian Latent Dynamics. First, we define the limit cycle in latent space as $\mathcal{O}_{\text{y}} = \{y \in \mathbb{R}^n | \sqrt{y_1^2 + y_2^2} = R, y_{3:n} = 0_{n-2}\}$ The Lyapunov function in Cartesian latent coordinates $y = h_{\text{p2c}}(y_{\text{pol}})$ can be stated as

$$V_{\text{y}}(y) = V_{\text{pol}}(h_{\text{y2p}}(y)) = \frac{\alpha R^2}{4}\left(y_1^2 + y_2^2 - R^2\right)^2 + \frac{1}{2} y_{3:n}^\top \beta \, y_{3:n}. \tag{12.29}$$

Naturally, the two conditions on the Lyapunov candidate $V_{\text{y}}(y) = 0 \,\forall y \in \mathcal{O}_{\text{y}}$ and $V_{\text{y}}(y) > 0 \,\forall y \in \mathbb{R}^n \setminus \mathcal{O}_{\text{y}}$ still hold. As the flow orthogonal to the orbit is given by

$$f_{\text{y},\perp}(y) = \frac{\partial h_{\text{p2c}}}{\partial y_{\text{pol}}} f_{\text{pol},\perp}(h_{\text{c2p}}(y)) = \begin{bmatrix} \alpha \left(1 - \frac{y_1^2 + y_2^2}{R^2}\right) y_1 \\ \alpha \left(1 - \frac{y_1^2 + y_2^2}{R^2}\right) y_2 \\ -\beta \, y_{3:n} \end{bmatrix}, \tag{12.30}$$

the corresponding time derivative of the Lyapunov function can be computed as

$$\begin{aligned} \dot{V}_{\text{y},f_{\text{y},\perp}}(y) &= \frac{\partial V_{\text{y}}(y)}{\partial y} f_{\text{y},\perp}(y) = \frac{\partial V_{\text{pol}}(h_{c2p}(y))}{\partial y} \frac{\partial h_{\text{p2c}}(y_{\text{pol}})}{\partial y_{\text{pol}}} f_{\text{pol},\perp}(y_{\text{pol}})\Big|_{y_{\text{pol}}=h_{c2p}(y)}, \\ &= \frac{\partial V_{\text{pol}}(y_{\text{pol}})}{\partial y_{\text{pol}}} \underbrace{\frac{\partial h_{\text{c2p}}}{\partial y} \frac{\partial h_{\text{p2c}}}{\partial y_{\text{pol}}}}_{\mathbb{I}_n} f_{\text{pol},\perp}(y_{\text{pol}})\Big|_{y_{\text{pol}}=h_{c2p}(y)}, \\ &= \frac{\partial V_{\text{pol}}(y_{\text{pol}})}{\partial y_{\text{pol}}} f_{\text{pol},\perp}(y_{\text{pol}})\Big|_{y_{\text{pol}}=h_{c2p}(y)} < 0, \quad \forall y \in \mathbb{R}^n \setminus \mathcal{O}_{\text{y}}. \end{aligned} \tag{12.31}$$

Step 3: Proof of Asymptotic Orbital Stability of OSMP Dynamics. Similar to prior work (Rana et al., 2020; Urain et al., 2020; Zhi et al., 2024), we transfer the orbital stability guarantees back into oracle space. The orbit/limit cycle in oracle space is given by $\mathcal{O}_{\text{x}} = \{x \in \mathbb{R}^n | y = \Psi(x;z), \sqrt{y_1^2 + y_2^2} = R, y_{3:n} = 0_{n-2}\}$. Then, define the Lyapunov function by substituting $y = \Psi(x;z)$ into $V_{\text{y}}(y)$

$$V_{\text{x}}(x;z) = V_{\text{y}}(\Psi(x;z)) = \frac{\alpha R^2}{4}\left(y_1^2 + y_2^2 - R^2\right)^2 + \frac{1}{2} y_{3:n}^\top \beta \, y_{3:n}\Big|_{y=\Psi(x;z)}, \tag{12.32}$$

which still admits to the properties $V_{\text{x}}(x) = 0 \,\forall x \in \mathcal{O}_{\text{x}}$ and $V_{\text{x}}(x) > 0 \,\forall x \in \mathbb{R}^n \setminus \mathcal{O}_{\text{x}}$. The flow orthogonal to the limit cycle can be expressed as

$$f_{\text{x},\perp}(x) = f_{\text{s}}(x) J_\Psi^{-1}(x;z) \, f_{\text{y},\perp}(\Psi(x;z)). \tag{12.33}$$

Similar to Step 2, but now also considering the oracle-space velocity scaling $f_{\text{s}}(x)$, we

derive the time derivative of the Lyapunov function orthogonal to the limit cycle flow as

$$\begin{aligned}\dot{V}_{\mathrm{x},f_{\mathrm{x},\perp}}(x) &= \frac{\partial V_{\mathrm{x}}(x)}{\partial x} f_{\mathrm{x},\perp}(x) = \frac{\partial V_{\mathrm{y}}(\Psi(x;z))}{\partial x} f_{\mathrm{s}}(x) J_{\Psi}^{-1}(x;z)\, f_{\mathrm{y},\perp}(\Psi(x;z))\Big|_{y=\Psi(x;z)}, \\ &= f_{\mathrm{s}}(x) \frac{\partial V_{\mathrm{y}}(y)}{\partial y} \underbrace{J_{\Psi}(x;z) J_{\Psi}^{-1}(x;z)}_{\mathbb{I}_n} f_{\mathrm{y},\perp}(\Psi(x;z))\Big|_{y=\Psi(x;z)}, \\ &= \underbrace{f_{\mathrm{s}}(x)}_{>0} \underbrace{\frac{\partial V_{\mathrm{y}}(y)}{\partial y} f_{\mathrm{y},\perp}(y_{\mathrm{pol}})\Big|_{y=\Psi(x;z)}}_{\leq 0} < 0, \quad \forall x \in \mathbb{R}^n \setminus \mathcal{O}_{\mathrm{x}}.\end{aligned} \tag{12.34}$$

□

Proof of Transverse Contraction

Now, we move to analyzing the transverse contraction conditions. To our knowledge, earlier studies have not tackled EOS or contraction (Lohmiller and Slotine, 1998). When a velocity scaling is applied in the original coordinates—as in Euclideanizing flows (Rana et al., 2020)—such guarantees cannot be established unless the scaling factor is explicitly bounded. By contrast, if no velocity scaling is used in the x-coordinates (i.e., $f_{\mathrm{s}} = 1$), we can prove contraction in the directions orthogonal to the limit cycle, a property known as transverse contraction (Manchester and Slotine, 2014). Transverse contraction implies EOS, ensuring trajectories converge to the limit cycle at an exponential rate.

This section is organized as follows: First, we provide formal definitions of (transverse) contraction and EOS. Subsequently, we prove transverse contraction of the polar latent dynamics. Then, we transfer this proof into Cartesian latent coordinates and, finally, also into the original/oracle coordinates.

Definitions

Definition 1. *An autonomous system $\dot{x} = f(x)$ with is said to be transverse contracting in the region $x \in \mathcal{X} \subseteq \mathbb{R}^n$ with rate $\zeta \in \mathbb{R}_{>0}$ if a positive definite metric $M(x) \succ 0 \in \mathbb{R}^{n\times n}$ exists such that*

$$\delta_x^{\top} \left(\frac{\partial f(x)}{\partial x}^{\top} M(x) + M(x) \frac{\partial f(x)}{\partial x} + \dot{M}(x) + 2\,\zeta\, M(x) \right) \delta_x \leq 0 \quad \forall x \in \mathcal{X} \tag{12.35}$$

for all $\delta_x \neq 0$ orthogonal to the flow satisfying $\delta_x^{\top} M(x) f(x) = 0$ (Manchester and Slotine, 2014).

Definition 2. *Let $\dot{x} = f(x)$ be an autonomous, transverse contracting system in the region $x \in \mathcal{X} \subseteq \mathbb{R}^n$ with contracting rate $\zeta \in \mathbb{R}_{>0}$. Also, consider the non-trivial T-periodic solution $x_{\mathrm{lc}}(t) \in \mathcal{X}$ that defines the solution curve $X_{\mathrm{lc}} = \{x \in \mathbb{R}^n : \exists t \in [0, T) : x = x_{\mathrm{lc}}\}$. Then, the solution $x_{\mathrm{lc}}(t)$ is said to be exponentially orbitally stable as there exists a $k > 0$ such that for any $x_0 \in \mathcal{X}$*

$$\inf_{x_{\mathrm{lc}} \in X_{\mathrm{lc}}} \|x(t) - x_{\mathrm{lc}}\|_2 \leq k \inf_{x_{\mathrm{lc}} \in X_{\mathrm{lc}}} \|x_0 - x_{\mathrm{lc}}\|_2\, e^{-\zeta t}. \tag{12.36}$$

Please note that the notion of EOS, also referred to as transverse exponential stability, is stronger than the commonly used AOS as it guarantees **exponential** convergence to the orbit/limit cycle (Manchester, 2011).

Proof of Transverse Contraction of Latent Dynamics in Polar Coordinates First, we prove that the latent dynamics in polar coordinates $\dot{y}_{\text{pol}} = f_{\text{pol}}(y_{\text{pol}})$ are transversely contracting: i.e., they are not contracting along the polar phase variable φ, but contracting orthogonal to the flow $f_{\text{pol}}(y_{\text{pol}})$ (Manchester and Slotine, 2014). Inspired by a recent proof of transverse contraction for the Andronov-Hopf oscillator with state (r, φ) (Nah et al., 2025), we define the following Proposition.

Proposition 1. *Let $\alpha, \beta > 0$, $R > 0$, $f_\omega(r) : [-\pi, \pi) \to \mathbb{R}_{>0}$, and $y_{\text{pol}} = \begin{bmatrix} r & \varphi & y_{3:n}^\top \end{bmatrix}^\top \in \mathcal{Y}_{\text{pol}}$. Then, the latent dynamics in polar coordinates from* (12.3) *are transverse contracting under the metric*

$$M_{\text{pol}}(y_{\text{pol}}) = \begin{bmatrix} \frac{1}{r^2} & -\frac{\alpha\left(1-\frac{r^2}{R^2}\right)}{f_\omega(\varphi)\, r} & 0_{1\times(n-2)} \\ -\frac{\alpha\left(1-\frac{r^2}{R^2}\right)}{f_\omega(\varphi)\, r} & m_{\varphi\varphi}(r) & 0_{1\times(n-2)} \\ 0_{(n-2)\times 1} & 0_{(n-2)\times 1} & \mathbb{I}_{n-2} \end{bmatrix} \succ 0 \in \mathbb{R}^{n\times n}, \tag{12.37}$$

with the contraction rate $\zeta_{\text{pol}} \geq \left(\frac{2\alpha}{R^2} + \beta\right) \frac{r_\epsilon^2}{r_\epsilon^2 + 1}$ in the region

$$\mathcal{Y}_{\text{pol}} = \left\{ y_{\text{pol}} \in \mathbb{R}^n | r_\epsilon \in \mathbb{R}_{>0}, r \geq r_\epsilon, \varphi \in [-\pi, \pi) \right\}. \tag{12.38}$$

Proof. The proof consists of three steps: proof of positive-definiteness of the contraction metric, and the fulfillment of the orthogonality and contraction conditions in order to meet the conditions for transverse contraction stated in Theorem 3 of Manchester and Slotine (2014).

Step 1: Positive definite contraction metric. Positive definite contraction metric $M_{\text{pol}}(y_{\text{pol}})$. In order for $M_{\text{pol}}(y_{\text{pol}}) \in \mathbb{R}^{n\times n}$ to be a valid contraction metric, we need to ensure that it is positive definite (i.e., that the real part of its Eigenvalues is positive $\forall y_{\text{pol}} \in \mathcal{Y}_{\text{pol}}$). For this to be the case, the following condition, derived from the smallest Eigenvalue of the contraction metric $\zeta_{\text{m}}(M_{\text{pol}}(y_{\text{pol}}))$, must hold

$$\frac{m_{\varphi\varphi}(r)\, r^2 + 1}{2 r^2} - \frac{\sqrt{\left(R^4 f_\omega^2(\varphi) r^4\right) m_{\varphi\varphi}^2 + \left(-2R^4 f_\omega^2(\varphi) r^2\right) m_{\varphi\varphi}(r) + \left(4\alpha^2 r^6 - 8R^2\alpha^2 r^4 + 4R^2\alpha^2 r^2 + R^4 f_\omega^2(\varphi)\right)}}{2 R^2 f_\omega(\varphi)\, r^2} \geq 0 \tag{12.39}$$

which can be ensured if the following two conditions hold

$$m_{\varphi\varphi} \geq 0, \quad \left(R^4\right) m_{\varphi\varphi}^2(r) + \left(-2R^4 f_\omega^2(\varphi) r^2\right) m_{\varphi\varphi}(r) + \left(4\alpha^2 r^6 - 8R^2\alpha^2 r^4 + 4R^2\alpha^2 r^2 + R^4 f_\omega^2(\varphi)\right) \leq 0. \tag{12.40}$$

Solving the quadratic equation results in

$$0 \leq m_{\varphi\varphi}(r) \leq m_{\varphi\varphi}^{\text{ub}}(r) = \frac{R^2 f_\omega(\varphi) + 2\,\alpha\,\sqrt{-r^4 + 2R^2 r^2 - R^4}\,|r|}{R^2\, f_\omega(\varphi)\, r^2}. \tag{12.41}$$

Thus, for example, the choice of $m_{\varphi\varphi}(r) = m^{\text{ub}}_{\varphi\varphi}(r)$ admits to the stated condition. Then, the Eigenvalues of the $M_{\text{pol}}(y_{\text{pol}})$ are given by

$$\zeta_{1,2}(M_{\text{pol}}(r)) = \frac{R^2 f_\omega(\varphi) + \alpha\sqrt{-r^4 + 2R^2r^2 - R^4}\,|r|}{R^2\, f_\omega(\varphi)\, r^2}, \qquad \zeta_{3:n}M_{\text{pol}}(r)) = 1. \tag{12.42}$$

with $\text{Re}(\zeta_{1,2}(M_{\text{pol}}(r))) > 0\,\forall r \in R_{>0}$. Therefore, $M_{\text{pol}}(y_{\text{pol}}) \succ 0\,\forall r \in R_{>0}$.

Step 2: Orthogonality condition. Let $\delta_{y_{\text{pol}}} = c\begin{bmatrix}1 & 0 & 1_{n-2}\end{bmatrix}^\top$ with $c \in \mathbb{R}_+$ be the incremental motion orthogonal to the flow. Then, the contraction metric (12.37) fulfills the orthogonality condition for the stated choice of δy_{pol} (Manchester and Slotine, 2014)

$$\delta^\top_{y_{\text{pol}}} M_{\text{pol}}(y_{\text{pol}})\, f_{\text{pol}}(y_{\text{pol}}) = \begin{bmatrix}c & 0 & c\,1_{1\times(n-2)}\end{bmatrix}\begin{bmatrix}0\\ m_{\varphi\varphi}(r)\, f_\omega(\varphi) - \alpha^2\,\frac{(R^2-r^2)^2}{R^4\, f_\omega(\varphi)}\\ -\beta\, y_{3:n}\end{bmatrix} = -c\,\beta\, y_{3:n}. \tag{12.43}$$

Since $-c\,\beta\, y_{3:n}$ converges uniformly to zero, the orthogonality condition defined in (12.43) converges to zero and is, therefore, fulfilled[1].

Step 3: Transverse contraction condition. The transverse contraction condition is given by (Manchester and Slotine, 2014)

$$\begin{aligned}\delta^\top_{y_{\text{pol}}}\left(\frac{\partial f_{\text{pol}}}{\partial y_{\text{pol}}}^\top M_{\text{pol}}(y_{\text{pol}}) + M_{\text{pol}}(y_{\text{pol}})\frac{\partial f_{\text{pol}}}{\partial y_{\text{pol}}} + \dot{M}_{\text{pol}}(y_{\text{pol}}) + 2\,\zeta_{\text{pol}}\, M_{\text{pol}}(y_{\text{pol}})\right)\delta_{y_{\text{pol}}} \leq 0,\\ \frac{2\,c^2}{r^2}\left(\zeta_{\text{pol}}\left(r^2+1\right) - \left(\frac{2\,\alpha}{R^2}+\beta\right)r^2\right) \leq 0,\end{aligned} \tag{12.44}$$

where

$$\begin{aligned}\frac{\partial f_{\text{pol}}}{\partial y_{\text{pol}}} &= \begin{bmatrix}\alpha - 3\alpha\,\frac{r^2}{R^2} & 0 & 0_{1\times(n-2)}\\ 0 & \frac{f_\omega(\varphi)}{\partial\varphi} & 0_{1\times(n-2)}\\ 0_{(n-2)\times 1} & 0_{(n-2)\times 1} & -\beta\,\mathbb{I}_{n-2}\end{bmatrix},\\ \dot{M}_{\text{pol}} &= \begin{bmatrix}-\frac{2\alpha}{r^2} + \frac{2\alpha}{R^2} & \frac{\alpha\left(R^2-r^2\right)\left(\alpha(R^2+r^2)+R^2\,\frac{\partial f_\omega}{\partial\varphi}\right)}{R^4\, r\, f_\omega(\varphi)} & 0_{1\times(n-2)}\\ \frac{\alpha\left(R^2-r^2\right)\left(\alpha(R^2+r^2)+R^2\,\frac{\partial f_\omega}{\partial\varphi}\right)}{R^4\, r\, f_\omega(\varphi)} & \alpha\,\frac{R^2-r^2}{R^2}\, r\,\frac{\partial m_{\varphi\varphi}}{\partial r} & 0_{1\times(n-2)}\\ 0_{(n-2)\times 1} & 0_{(n-2)\times 1} & 0_{(n-2)\times(n-2)}\end{bmatrix}.\end{aligned} \tag{12.45}$$

We can simplify (12.44) to

$$\begin{aligned}2\,c^2\left(\zeta_{\text{pol}}\left(1+\frac{1}{r^2}\right) - \left(\frac{2\alpha}{R^2}+\beta\right)\right) \leq 0,\\ \zeta_{\text{pol}}\left(r^2+1\right) - \left(\frac{2\,\alpha}{R^2}+\beta\right)r^2 \leq 0,\\ \zeta_{\text{pol}} \leq \left(\frac{2\,\alpha}{R^2}+\beta\right)\frac{r^2}{r^2+1}.\end{aligned} \tag{12.46}$$

[1] See proof of Theorem 5 in (Manchester and Slotine, 2014).

Given $y_{\text{pol}} \in \mathcal{Y}_{\text{pol}}$ (i.e., $r \geq r_\epsilon$), we can, therefore, guarantee that the actual contraction rate admits to the lower bound

$$\zeta_{\text{pol}} \geq \left(\frac{2\,\alpha}{R^2} + \beta \right) \frac{r_\epsilon^2}{r_\epsilon^2 + 1}. \tag{12.47}$$

□

In practical robotics settings, the contraction rate is particularly relevant for the region $r \geq R^2$ (i.e., the system is outside the defined limit cycle contour). In such a setting with $r_\epsilon = R$, the contraction rate is given $\forall r \geq R$ by $\zeta_{\text{pol}} \geq \frac{2\alpha+\beta R^2}{R^2+1}$ and for $R = 1$ by $\zeta_{\text{pol}} \geq \alpha + \frac{\beta}{2}$. This illustrates well how the latent dynamics, α, β, allow us to modulate the contraction behavior of the system.

Proof of Transverse Contraction of Latent Dynamics in Cartesian Coordinates

Proposition 2. *Let $\omega \geq 0$, $\alpha, \beta > 0$, $R > 0$, and $y = \begin{bmatrix} y_1 & y_2 & y_{3:n}^\top \end{bmatrix}^\top \in \mathcal{Y}$. Then, the latent dynamics in Cartesian coordinates from* (12.2) *are transverse contracting under the metric*

$$M_y(y) = \frac{\partial h_{\text{p2c}}}{\partial y_{\text{pol}}}^{-\top} M_{\text{pol}} \frac{\partial h_{\text{p2c}}}{\partial y_{\text{pol}}}^{-1} \Bigg|_{y_{\text{pol}} = h_{\text{p2c}}^{-1}(y)} \succ 0 \in \mathbb{R}^{n \times n}, \tag{12.48}$$

with the contraction rate $\zeta_y \geq \left(\frac{2\alpha}{R^2} + \beta\right) \frac{r_\epsilon^2}{r_\epsilon^2+1}$ in the region $\mathcal{Y} = \left\{ y \in \mathbb{R}^n | r_\epsilon \in \mathbb{R}_{>0}, \sqrt{y_1^2 + y_2^2} \geq r_\epsilon \right\}$.

Proof. Again, the proof consists of three steps: proof of positive-definiteness of the contraction metric, and the fulfillment of the orthogonality and contraction conditions in order to meet the conditions for transverse contraction.

Step 1: Positive definite contraction metric. As shown in Proposition 1, $M_{\text{pol}}(y) \succ 0 \, \forall \, y \in \mathbb{R}^n$. As M_{pol} is square and rank $\left(\frac{\partial h_{\text{p2c}}}{\partial y_{\text{pol}}}^{-1} \right) = n \, \forall y \in \mathcal{Y}$, $M_y(y)$, as defined in (12.48), is positive definite (Petersen et al., 2008).

Step 2: Orthogonality condition. Let the incremental motion orthogonal to the flow be defined as

$$\delta_y = \frac{\partial h_{\text{p2c}}}{\partial y_{\text{pol}}} \delta_{y_{\text{pol}}} \Big|_{y_{\text{pol}} = h_{\text{p2c}}^{-1}(y)} = c \begin{bmatrix} \frac{y_1}{\sqrt{y_1^2+y_2^2}} & \frac{y_2}{\sqrt{y_1^2+y_2^2}} & 1 \end{bmatrix}^\top. \tag{12.49}$$

The orthogonality condition is then given by

$$\delta_y^\top M_y(y) f_y(y) = -c\,\beta\, y_{3:n}. \tag{12.50}$$

Since $-c\,\beta\, y_{3:n}$ converges uniformly to zero, the orthogonality condition defined in (12.43) converges to zero.

Step 3: Transverse contraction condition. The transverse contraction condition can be stated as

$$\begin{aligned} \delta_y^\top \left(\frac{\partial f_y}{\partial y}^\top M_y(y) + M_y(y) \frac{\partial f_y}{\partial y} + \dot{M}_y(y) + 2\,\zeta_y M_y(y) \right) \delta_y \leq 0 \\ \frac{2\,c^2}{R^2(y_1^2+y_2^2)} \left(-2\alpha(y_1^2+y_2^2) - \beta R^2(y_1^2+y_2^2) + \zeta_y R^2(y_1^2+y_2^2) + \zeta_y R^2 \right) \leq 0, \\ -(2\alpha + \beta R^2)(y_1^2+y_2^2) + \zeta_y R^2(y_1^2+y_2^2+1) \leq 0. \end{aligned} \tag{12.51}$$

Subsequently, we can isolate the contraction rate in the inequality condition

$$\zeta_y \leq \left(\frac{2\alpha}{R^2} + \beta \right) \frac{y_1^2 + y_2^2}{y_1^2 + y_2^2 + 1}. \tag{12.52}$$

Now, if we define the contraction region as $\sqrt{y_1^2 + y_2^2} \geq r_\epsilon > 0$, we can guarantee a transverse contraction rate $\zeta_y \geq \left(\frac{2\alpha}{R^2} + \beta \right) \frac{r_\epsilon^2}{r_\epsilon^2 + 1} \, \forall y \in \mathcal{Y}$. □

Proof of Transverse Contraction in Oracle/Original Coordinates

Theorem 5. *Let $\alpha, \beta > 0$, $R > 0$, and $z \in \mathbb{R}$ be constants. Also, choose $f_s(x) = 1$ and $\omega(\bar{y}_{1:2}) : \mathbb{R}^2 \to \mathbb{R}_{>0}$. Then, the OSMP dynamics $\dot{x} = f(x; z)$ defined in* (12.1) *are transverse contracting in the region $\mathcal{X} = \left\{ x \in \mathbb{R}^n | \sqrt{\{\Psi(x,z)\}_1^2 + \{\Psi(x,z)\}_2^2} > 0 \right\}$.*

Proof. The proof consists of the following steps:

1. Proposition 1 proves that the polar latent dynamics are transverse contracting in the region $\mathcal{Y}_{\text{pol}} = \left\{ y_{\text{pol}} \in \mathbb{R}^n | r > 0, \varphi \in [-\pi, \pi) \right\}$.
2. Proposition 2 shows that the same transverse contraction properties hold in the Cartesian latent coordinates with dynamics $\dot{y} = f_y(y)$.
3. Existing work (Beik-Mohammadi et al., 2024; Jaffe et al., 2024; Manchester and Slotine, 2017) demonstrates that such contraction properties also hold after a change of coordinates $x = \Psi^{-1}(y; z)$ that is defined by a smooth diffeomorphism, which is the case for our encoder based on Euclideanizing flows (Rana et al., 2020). This is equivalent to proving the transverse contraction of the oracle space dynamics as $f_s(x) = 1$.

□

Intuitively, Theorem 5 tells us that two trajectories starting from any initial conditions outside the exact center of the limit cycle will converge exponentially to the same periodic orbit (Manchester and Slotine, 2014), demonstrating almost-global contraction. This, in turn, implies almost-global exponential orbital stability: no matter the initial conditions (as long as they are outside the center of the limit cycle with $r = 0$), the trajectories will reach the stable limit cycle specified by the OSMP in exponential time.

12.3.2 OSMPs Exhibit a High Imitation Fidelity and Ensure Global Convergence to the Oracle

We conduct both quantitative and qualitative evaluations of OSMPs against several baselines. Specifically, we evaluate the transverse contracting/exponentially stable variant of OSMP with $f_s(x) = 1$. The baselines include classical neural motion policies—MLPs, RNNs, and LSTMs—that directly predict the next system position, plus NODEs (Chen et al., 2018), which instead predict the desired velocity. We also compare with state-of-the-art robotic imitation-learning methods such as DPs that predict system trajectory over a horizon and existing SMPs designed for periodic motion, namely Imitation Flows (iFlow) (Urain et al., 2020) and SPDT (Zhi et al., 2024), predicting system velocities.

Training

The OSMP motion policy is trained by a AdamW optimizer (Kingma and Ba, 2014; Loshchilov and Hutter, 2018) with $(\beta_1, \beta_2) = (0.9, 0.999)$ and a default weight decay of $\lambda = 10^{-10}$. We employ a learning rate scheduler that sequences a linear warmup phase (usually 10 epochs), with a relatively short constant learning rate period and subsequent long cosine annealing (Loshchilov and Hutter, 2016) period for the remaining epochs. We remark that we don't use a minibatch-based training strategy but instead process all given demonstrations in a single batch. Please note that we also apply the described training procedure to the baseline methods if not explicitly mentioned otherwise.

Datasets

In the following, we will introduce the datasets that we considered for the evaluation. We note that before training, all positions contained in the datasets are normalized to the interval $[-0.5, 0.5]$ with zero mean.

IROS Letters Originally published by Urain et al. (2020) and later adopted for benchmarking by Nawaz et al. (2024), the IROS Letters dataset provides several demonstrations for each character (*IShape*, *RShape*, *OShape*, and *SShape*), sometimes spanning multiple cycles. A noteworthy characteristic is that demonstrations of the same task are widely separated in state space, posing a challenge for deterministic policies. We smooth every trajectory with a Savitzky–Golay filter (order 3, window 8). Because the *IShape* sequence contains very few sample points and large gaps between consecutive states, we upsample it by a factor of 5. The duration of the demonstrations is chosen as 20 s.

Drawing2D The Drawing2D dataset introduced by Nawaz et al. (2024) offers four demonstrations of a kidney-bean-shaped periodic drawing. We train a separate motion policy for each demonstration, applying the same Savitzky–Golay filter (order 3, window 8) for smoothing. Trajectories are normalised to a SI20s duration.

Image Contour We contribute a new benchmark category based on image contours that range from simple shapes (Ellipse, Square) to highly intricate outlines such as the TU Delft flame logo, Bat, and Eagle. Compared to prior benchmarks like IROS Letters (Urain et al., 2020) and Drawing2D (Nawaz et al., 2024), these oracles introduce sharp corners (e.g., Star, Bat), strongly concave curves (e.g., TUD-Flame), and discontinuous velocity profiles (e.g., Star, Bat, Eagle), all of which are difficult for most DMP-based approaches. We extract each outline with OpenCV (Bradski, 2000) and lightly smooth it using a Savitzky–Golay filter (order 3, window 25). Every trajectory lasts 20 s. The list of image contours is: Ellipse, Square, Star, MIT-CSAIL, TUD-Flame, Doge, Bat, Dolphin, and Eagle.

Turtle Swimming This category contains four datasets. **(i)** The first two comprise Cartesian trajectories of the flipper tip of wild green sea turtles (Chelonia mydas) captured by marine biologists (van der Geest et al., 2022) and represented by Fourier series. We train on two variants: position only (3-D oracle) and position plus twist angle (4-D oracle), each with a period of 4.2 s. **(ii)** A subsequent dataset from the same authors applies inverse kinematics to those trajectories, yielding a three-joint-space oracle for bioinspired

robotic turtles (van der Geest et al., 2023). After smoothing with a 30th-degree polynomial, we compute velocities; this oracle has a 4.3 s period. **(iii)** Finally, we include a reverse-swimming template defined in joint space by sinusoidal functions, with a 4 s period. This template was designed by constructing waypoints to produce "reverse rowing" movement patterns, interpolating between them with a spline, and finally approximating the trajectory with a Fourier fit.

A key distinction from previous periodic-motion datasets is the pivotal influence of the velocity profile on swimming performance: if the velocity profile with which the trajectory is executed is wrong, the cost of transport rises, and in extreme cases, the turtle robot either stalls or even moves in the opposite direction.

Baseline Methods

Next, we formalize the baseline methods that we used during the evaluation.

Trajectory Tracking PD Controller A classical error-based feedback controller tracking a time-parametrized trajectory $(x^{\mathrm{d}}(t), \dot{x}^{\mathrm{d}}(t), \ddot{x}^{\mathrm{d}}(t)) \,\forall\, t \in [t_0, t_{\mathrm{f}}]$, referred to in this thesis as Trajectory Tracking (TT), is usually given in the form

$$\dot{x}(t) = \dot{x}^{\mathrm{d}}(t) + K_{\mathrm{p}}\,(x^{\mathrm{d}}(t) - x(t)), \tag{12.53}$$

where $K_{\mathrm{p}} \in \mathbb{R}^{n \times n}$ is a proportional feedback gain that operates on the error between the current position $x(t)$ and the desired position $x^{\mathrm{d}}(t)$. In practice, we choose a scalar $k_{\mathrm{p}} \in \mathbb{R}_+$ such that $K_{\mathrm{p}} = k_{\mathrm{p}}\,\mathbb{I}_n$.

Multilayer Perceptron (MLP) As the most basic neural motion policy, we consider an MLP that predicts the next position/state of the system according to the discrete transition function $x(k+1) = f_{\mathrm{MLP}}(x)$. We employ a five-layer MLP with a hidden dimension of 128 and a LeakyReLU activation function. During training, we randomly sample at the start of each epoch $N_{\mathrm{init}} = 64$ initial positions from the oracles contained in the dataset. Subsequently, we roll out each trajectory for $T = 25$ steps and enforce an MSE loss between the predicted $x_i(k)$ and demonstrated trajectory $x_i^{\mathrm{d}}(k)$

$$\mathcal{L}_{\mathrm{ro}} = \frac{\sum_{i=1}^{N_{\mathrm{init}}} \sum_{k=1}^{T} \left(x_i(k) - x_i^{\mathrm{d}}(k)\right)^2}{N_{\mathrm{init}}\,K}. \tag{12.54}$$

Recurrent Neural Networks (RNNs, LSTM) For the RNN-like networks (i.e., Elman RNN & LSTM), we employ a five-layer recurrent neural network with a hidden dimension of 128. For example, in the case of the Elman RNN, the transition function of the hidden state $h_j \in \mathbb{R}^{128}$ of the jth layer is given by

$$h_j(k) = \tanh\left(W_{\mathrm{xh}}\,u_j(k) + W_{\mathrm{hh}}\,h_j(k-1) + b\right). \tag{12.55}$$

Here, $u_j(k)$ is the input to the jth layer such that $u_1(k) = x(k)$ and $u_j(k) = h_{j-1}(k) \,\forall\, j \in 2, \ldots, 5$ and the output of the RNN (i.e., the next state prediction) is given by $x(k+1) = W_{\mathrm{o}}\,h_5(k)$, where $W_{\mathrm{o}} \in \mathbb{R}^{n \times 128}$ is a learned matrix. For training, we use the same rollout procedure and loss $\mathcal{L}_{\mathrm{ro}}$ as in the case of the discrete MLP motion policy, with the difference

that we initialize the RNN's concatenated hidden state $h = \begin{bmatrix} h_1^\top & \dots & h_5^\top \end{bmatrix}^\top$ as $h(1) = 0_{640}$ at the beginning of the trajectory and subsequently propagate through each of the 25 transitions.

Neural ODE (NODE) A NODE-based motion policy can be defined as $\dot{x} = f_{\text{NODE}}(x)$ where $f_{\text{NODE}}(x) : \mathbb{R}^n \to \mathbb{R}^n$ is parametrized by an MLP. Specifically, we choose a five-layer MLP with hidden dimensions of 128 and LeakyReLU activation functions. In addition to supervising the predicted velocity via $\mathcal{L}_{\text{vi}}$, we compute position-based losses based on rolled-out trajectories analogous to the MLP and RNN obtained via forward Euler integration of $f_{\text{NODE}}(x)$.

Diffusion Policy (DP) We use the official open-source implementation of DPs (Chi et al., 2023) and train them on the respective datasets for 250 epochs while employing a batch size of 256, an AdamW (Kingma and Ba, 2014; Loshchilov and Hutter, 2018) optimizer configured with a learning rate of 10^{-4}, a weight decay of 10^{-6}, and $(\beta_1, \beta_2) = (0.95, 0.999)$ and a cosine learning rate scheduler with 10 warmup steps. For this task, we define the observation as the current and last position of the system $o(k) = \begin{bmatrix} x^\top(k-1) & x^\top(k) \end{bmatrix}^\top \in \mathbb{R}^{2n}$ and the action as the positional state of the system $a = x \in \mathbb{R}^n$. For each observation $o(k)$, the DP is trained to predict a sequence of $h = 16$ actions $a(k+1), \dots a(k+h)$, of which during inference only eight actions are executed before replanning takes place. The sampled noise is denoised in a sequence of 100 steps via a Denoising Diffusion Probabilistic Models (DDPMs) `squaredcos_cap_v2` scheduler (Ho et al., 2020). The noise samples are clipped to the range $[-1, 1]$ and the noise prediction network is parametrized by a 1D UNet (Ronneberger et al., 2015) with dimension $(256, 512, 1024)$, kernel size 5 and global conditioning on the observation $o(k)$.

Imitation Flows (iFlow) We leverage the official iFlow (Urain et al., 2020) implementation for the training on the considered dataset. Specifically, we train the model with a normalizing flows (Rezende and Mohamed, 2015) bijective encoder consisting of 15 ResNet coupling layers for 1000 epochs with a batch size of 100 using an Adamax (Kingma and Ba, 2014) optimizer with learning rate 10^{-3} and $(\beta_1, \beta_2) = (0.9, 0.999)$. To define the dynamics in latent space, the method uses a stochastic variant of linear polar limit cycle dynamics transformed into Cartesian coordinates.

Stable Periodic Diagrammatic Teaching (SPDT) The model architecture of SPDT (Zhi et al., 2024) is very similar to OSMPs, apart from the parametric angular velocity. Furthermore, in addition to advanced features, such as online shaping of the learned motion, phase synchronization, and motion/task conditioning, that OSMPs exhibit, the main difference in training a SPDT lies in the loss function: Instead of employing a velocity imitation loss $\mathcal{L}_{\text{vi}}$, a limit cycle matching loss $\mathcal{L}_{\text{lcm}}$, a time reference guidance loss $\mathcal{L}_{\text{trg}}$, and velocity regularization loss $\mathcal{L}_{\text{vr}}$, SPDT in mainly relies on on the Hausdorff distance to ensure that the encoded demonstrations match the limit cycle defined in latent space

$$\mathcal{L}_{\text{haus}} := \max \left\{ \max_{k \in \mathbb{N}_N} \min_{\kappa \in \mathbb{N}_N} d_y(k, \kappa), \max_{\kappa \in \mathbb{N}_N} \min_{k \in \mathbb{N}_N} d_y(\kappa, k) \right\}, \tag{12.56}$$

where

$$d_y(k,\kappa) = \|\Psi(x(k);z(k)) - y^{\mathrm{d}}(\kappa)\|_2, \tag{12.57}$$

and $y^{\mathrm{d}}(k) = \begin{bmatrix} r\cos(\varphi(k)) & r\sin(\varphi(k)) & 0_{n-2}^\top \end{bmatrix}^\top \in \mathbb{R}^n \,\forall\, k \in \mathbb{N}_n$ is a sequence of length N of positions on the latent-space limit cycle obtained via arranging equally-spaced polar angles $\varphi^{\mathrm{d}}(k) \in [-\pi, \pi)$ and subsequently projecting them back into cartesian space with radius $r = R$. Additionally, the method also employs the encoder regularization loss $\mathcal{L}_{\mathrm{er}}$.

Imitation Metrics

We adopt several metrics from the literature that measure how well the motion policy is able to track the given demonstration, both along the time and spatial dimensions. For all metrics, we initialize the motion policy at the starting position of the demonstration (i.e., $x(1) = x^{\mathrm{d}}(1)$) and subsequently roll out the motion for N steps (i.e., as many steps as included in the demonstration). Subsequently, we evaluate the mismatch between the actual $x(k) \in \mathbb{R}^n \,\forall\, k \in \mathbb{N}_N$ and desired trajectory $x^{\mathrm{d}}(k) \in \mathbb{R}^n \,\forall\, k \in \mathbb{N}_N$, where we define $\mathbb{N}_N = 1, \ldots, N$. In case a dataset contains multiple demonstrations, we separately initialize the motion policy at the starting point of each demonstration and report the mean of the metrics across all demonstrations. Please note that we compute all evaluation metrics on the normalized datasets (i.e., with the positions normalized into the range $[-0.5, 0.5]$), which makes it easier to compare and aggregate metrics across several datasets within the same dataset category. For all imitation metrics, we rely on the Python implementation by Jekel et al. (2019) open-sourced in the `similaritymeasures` package.

Trajectory RMSE The *Trajectory Root Mean Square Error (Traj. RMSE)*, used, for example, by Khadivar et al. (2021), measures the deviation of the actual from the nominal trajectory in position space and is computed as

$$\mathrm{RMSE}_{\mathrm{x}}\left(\{x(k)\}_{k\in\mathbb{N}_N}, \{x^{\mathrm{d}}(k)\}_{k\in\mathbb{N}_N}\right) := \sum_{k=1}^{N}\sum_{i=1}^{n} \frac{\left(x_i^{\mathrm{d}}(k) - x_i(k)\right)^2}{N\,n}. \tag{12.58}$$

Normalized Trajectory DTW Dynamic Time Warping (DTW) (Sakoe and Chiba, 1978) searches across all allowable temporal warpings to identify the alignment that minimizes the Euclidean distance between two time series and is used frequently to benchmark imitation learning and behavioral cloning algorithms in robotics (Nawaz et al., 2024; Pérez-Dattari and Kober, 2023; Urain et al., 2020) is defined as

$$\mathrm{DTW}_{\mathrm{x}}\left(\{x(k)\}_{k\in\mathbb{N}_N}, \{x^{\mathrm{d}}(k)\}_{k\in\mathbb{N}_N}\right) := \min_{\pi} \sum_{(k,\kappa)\in\pi} \left\| x^d(k) - x(\kappa) \right\|_2. \tag{12.59}$$

where $\pi = ((k_1, \kappa_1), \ldots, (k_N, \kappa_N))$ is commonly referred to as the alignment path of length N that contains the sequence of index pairs[2]. In order for π to be a valid alignment path, it needs to fulfill the following constraints

$$\begin{aligned} \pi_1 = (k_1, \kappa_1) = (1,1), \qquad & \pi_N = (k_1, \kappa_1) = (N,N), \\ k_{\iota+1} - k_\iota \in \{0,1\}, \quad \kappa_{\iota+1} - \kappa_\iota \in \{0,1\}, \quad & k_{\iota+1} - k_\iota + \kappa_{\iota+1} - \kappa_\iota \geq 1, \end{aligned} \tag{12.60}$$

[2]Although DTW is formally not a valid mathematical metric and instead a similarity measure as the triangle inequality doesn't hold, we refer to it here for convenience as one of the evaluation metrics.

where the first row contains constraints that ensure that the beginning and the end of each sequence are connected, and the second row verifies that the indices are monotonically increasing in both k and κ and are contained in the sequence at least once. However, as can be seen in (12.59), the magnitude of the DTW is proportional to the demonstration length N, which makes it challenging to aggregate the measure across several datasets of varying demonstration lengths. Therefore, we instead compute a normalized version of the DTW, the *Normalized Trajectory Dynamic Time Warping* measure, as

$$\mathrm{NDTW}_{\mathrm{x}}\left(\{x(k)\}_{k\in\mathbb{N}_N},\{x^{\mathrm{d}}(k)\}_{k\in\mathbb{N}_N}\right)=\frac{1}{N}\,\mathrm{DTW}_{\mathrm{x}}\left(\{x(k)\}_{k\in\mathbb{N}_N},\{x^{\mathrm{d}}(k)\}_{k\in\mathbb{N}_N}\right). \tag{12.61}$$

Fréchet Distance Let the Minkowski distance between two points be given by

$$d_{\mathrm{m},p}(x(k),x^{\mathrm{d}}(\kappa))=\|x(k)-x^{\mathrm{d}}(\kappa)\|_p=\left(\sum_{i=1}^{n}\left|x_i(k)-x_i^{\mathrm{d}}(\kappa)\right|^p\right)^{\frac{1}{p}}\quad\forall\,k,\kappa\in\mathbb{N}_N, \tag{12.62}$$

where we choose the Minkowski order $p=2$. Then, for all monotone lattices paths σ from $(1,1)$ to (N,N) that move only right, up, or diagonally-up-right, the Fréchet distance is given by (Eiter et al., 1994)

$$D_{\mathrm{F}}\left(\{x(k)\}_{k\in\mathbb{N}_N},\{x^{\mathrm{d}}(k)\}_{k\in\mathbb{N}_N}\right):=\min_{\sigma}\max_{(k,\kappa)\in\sigma}d_{\mathrm{m},p}(x(k),x^{\mathrm{d}}(\kappa)). \tag{12.63}$$

Velocity RMSE The *Velocity Root Mean Square Error (Vel. RMSE)* measures the deviation of the actual from the nominal velocities along the trajectory and is computed as

$$\mathrm{RMSE}_{\dot{\mathrm{x}}}\left(\{\dot{x}(k)\}_{k\in\mathbb{N}_N},\{\dot{x}^{\mathrm{d}}(k)\}_{k\in\mathbb{N}_N}\right):=\sum_{k=1}^{N}\sum_{i=1}^{n}\frac{\left(\dot{x}_i^{\mathrm{d}}(k)-\dot{x}_i(k)\right)^2}{N\,n}, \tag{12.64}$$

where $\dot{x}(k)$ is either directly given by the motion policy (e.g., DMP, OSMP, NODE) or obtained via finite differences as $\dot{x}(k)=\frac{x(k)-x(k-1)}{\delta t}$ (e.g., DP). This metric is particularly relevant for tasks in which it is crucial that the demonstrated velocity, and not just trajectory, is accurately tracked, as in the case of turtle swimming.

Convergence Metrics

Our convergence study pursues two questions: (i) Does the method display orbital stability, converging to a stable limit cycle from any initial state in the workspace? and (ii) Does that limit cycle coincide with the desired oracle? Unlike the imitation metrics, we do not examine timing or velocity profiles; our attention is restricted to the geometric path of the motion. To the best of our knowledge, no standardized protocol yet exists for gauging the orbital-convergence behavior of rhythmic motion policies. Therefore, we devised a protocol based on several rollouts from randomly sampled initial conditions that verifies both the local and global convergence characteristics of the motion policy, which we describe in more detail in the next paragraph. We evaluate the convergence metrics on all rollouts and subsequently compute the mean value over the resulting trajectories.

Local vs. Global Convergence We evaluate both local and global convergence characteristics of the motion policies. In both cases, we sample randomly 25 positions $x^{\mathrm{d}}(\kappa)\,\forall \kappa \in 1,\dots,25$ from the set of positions contained in the demonstration $x^{\mathrm{d}}(k)\forall k \in 1,\dots,N$. Subsequently, we sample an offset $\Delta x_\kappa \sim \mathcal{N}(0,\sigma) \in \mathbb{R}^n$. Now, we consider $x_\kappa(1) = x^{\mathrm{d}}(\kappa) + \Delta x_\kappa$ as the initial position for the rollout of the motion policy. For the local convergence analysis, we initialize close to the demonstration by choosing $\sigma = 0.05$. For the global convergence analysis, we select instead $\sigma = 0.15$.

Subsequently, we roll out the motion policy for N and $2N$ steps for the local and global convergence analysis, respectively, resulting in the trajectory sequences $(x_\kappa(1),\dots,x_\kappa(N))$ and $(x_\kappa(1),\dots,x_\kappa(2N))$. For the global convergence, we strive to give the motion policy sufficient time to converge to its limit cycle, and therefore, only evaluate the metrics on the trajectory after the demonstration duration - i.e., we compute the metric on the trajectory sequence $(x_\kappa(N+1),\dots,x_\kappa(2N))$. For simplicity of notation, we assume in the following a reset of time indices such that $k = k - N$.

Directed Hausdorff Distance The directed Hausdorff distance (Hausdorff, 1914; Huttenlocher et al., 1993) computes the largest distance between closest-neighbor correspondences from the actual convergence trajectory to the desired limit cycle

$$h^{\rightarrow}\left(\{x(k)\}_{k\in\mathbb{N}_N},\{x^{\mathrm{d}}(\kappa)\}_{\kappa\in\mathbb{N}_N}\right) := \max_{k\in\mathbb{N}_N}\min_{\kappa\in\mathbb{N}_N}\left\|x(k)-x^{\mathrm{d}}(\kappa)\right\|_2. \tag{12.65}$$

The undirected/symmetric version of the Hausdorff distance was used by Zhi et al. (2024) for evaluating the similarity between the desired and actual trajectory shape.

Iterative Closest Point MED We use an Iterative Closest Point (ICP) algorithm (Besl and McKay, 1992) to identify the optimal alignment between the two sequences $x^{\mathrm{d}}(k)\,\forall\, k \in 1,\dots,N$ and $x(\kappa)\,\forall\,\kappa \in 1,\dots,N$ containing the desired limit cycle shape (i.e., the demonstration) and the actual (asymptotic) behavior generated by the motion policy, respectively, by iteratively estimating the transformation, consisting of a translation and a rotation, between the two shapes. After initializing the translation between the points as $p_0 = 0_n$ and the rotation as $R = \mathbb{I}_n$, each iteration of the ICP algorithm performs two steps

$$\textbf{(correspondence-search step)} \qquad c_m(k) := \arg\min_{\kappa\in N}\left\|x(k) - \left(R_{m-1}\,x^{\mathrm{d}}(\kappa) + p_{m-1}\right)\right\|_2,$$

$$\textbf{(alignment step)} \qquad (R_m, p_m) := \arg\min_{\substack{R\in SO(n)\\ p\in\mathbb{R}^n}}\sum_{k=1}^{n}\left\|x(k) - \left(R\,x^{\mathrm{d}}\left(c_m(k)\right) + p\right)\right\|_2^2, \tag{12.66}$$

where the nearest-neighborhood correspondence-search returns a correspondence map $c : N \to N$ that provides for each point $x(k)$ on the actual trajectory the closest point on the desired shape $x^{\mathrm{d}}(c(k))$. The *alignment-step* identifies the optimal transformation that aligns the two point sequences and is, in practice, implemented using a closed-form SVD step. After the algorithm has converged to $\hat{c} := c_{m^\star}$, $\hat{R} := R_{m^\star}$, and $\hat{p} := p_{m^\star}$, the Mean

Euclidean Distance (MED) between the two aligned shapes is computed as

$$\mathrm{MED}_{\mathrm{conv}}\left(\{x(k)\}_{k\in\mathbb{N}_N},\{x^{\mathrm{d}}(k)\}_{k\in\mathbb{N}_N}\right)=\sum_{k=1}^{N}\frac{\left\|x(k)\;-\;\left(\hat{R}\,x^{\mathrm{d}}\left(c_m(k)\right)+\hat{p}\right)\right\|_2}{N}. \tag{12.67}$$

Please note that ICP-based MED is designed to compare the shape of the desired and measured limit cycles. Because the algorithm also finds a single translation that minimizes the point-to-point distances, two identical shapes that are simply shifted in space will still produce an MED of zero, even though the measured limit cycle is displaced relative to the demonstration. Therefore, it is crucial to also always evaluate other metrics, such as the directed Hausdorff distance.

Table 12.2: **Orbitally Stable Motion Primitives (OSMPs) provide the best tradeoff between imitation accuracy, global convergence characteristics, and computational time when compared to classical neural motion policies (e.g., MLPs, RNNs, NODEs), current SOTA algorithms such as DPs (Chi et al., 2023), and prior work on period motion policies with stability guarantees (e.g., iFlow (Urain et al., 2020), SPDT (Zhi et al., 2024))**. We report the mean ± std across three random seeds for each dataset/method evaluation. Entries marked with a * diverged for some random seeds, and we only report the statistics for the seeds that converged. We label other cases where all random seeds diverged or exhibited extremely high errors using the ∞ symbol.

Dataset Category	Method	Imitation Metrics			Local Convergence Metrics		Global Convergence Metrics		Eval. Time per Step ↓
		Traj. RMSE ↓	Norm. Traj. RMSE ↓	Vel. RMSE ↓	Hausdorff Dist. ↓	ICP MED ↓	Hausdorff Dist. ↓	ICP MED ↓	
IROS Letters	MLP	0.257 ± 0.011	0.0745 ± 0.0042	0.630 ± 0.011	0.049 ± 0.003	0.011 ± 0.000	0.039 ± 0.003	0.011 ± 0.000	0.0017
	Elman RNN	0.267 ± 0.011	0.0935 ± 0.0286	0.680 ± 0.004	0.092 ± 0.011	0.013 ± 0.002	0.048 ± 0.014	0.013 ± 0.003	0.0009
	LSTM	0.406 ± 0.058	0.2719 ± 0.0473	0.777 ± 0.055	0.267 ± 0.104	0.032 ± 0.008	0.149 ± 0.050	0.019 ± 0.008	0.0010
	NODE	0.751 ± 0.573	0.5894 ± 0.5618	0.856 ± 0.062	∞	∞	∞	∞	0.0009
	DP	$\mathbf{0.255 \pm 0.004}$	$\mathbf{0.0891 \pm 0.0029}$	$\mathbf{0.648 \pm 0.004}$	0.103 ± 0.001	0.024 ± 0.001	0.101 ± 0.001	0.024 ± 0.001	0.0272
	iFlow	0.783 ± 0.481	0.6832 ± 0.6326	1.317 ± 0.895	1.248 ± 1.201	0.441 ± 0.449	1.242 ± 1.204	0.441 ± 0.449	**0.0007**
	SPDT	0.374 ± 0.002	0.1406 ± 0.0017	0.707 ± 0.004	0.103 ± 0.002	0.027 ± 0.001	0.098 ± 0.003	0.027 ± 0.001	0.0019
	OSMP (ours)	0.344 ± 0.007	0.1759 ± 0.0010	0.895 ± 0.007	$\mathbf{0.044 \pm 0.001}$	$\mathbf{0.010 \pm 0.000}$	$\mathbf{0.032 \pm 0.001}$	$\mathbf{0.009 \pm 0.000}$	0.0020
Drawing2D	MLP	$\mathbf{0.039 \pm 0.004}$	$\mathbf{0.0060 \pm 0.0002}$	$\mathbf{0.071 \pm 0.004}$	0.046 ± 0.003	$\mathbf{0.005 \pm 0.000}$	$\mathbf{0.015 \pm 0.001}$	$\mathbf{0.004 \pm 0.000}$	0.0008
	Elman RNN	0.109 ± 0.009	0.0457 ± 0.0382	0.199 ± 0.011	0.119 ± 0.002	0.012 ± 0.001	0.041 ± 0.003	0.009 ± 0.003	0.0007
	LSTM	0.384 ± 0.039	0.2891 ± 0.0911	0.482 ± 0.121	0.191 ± 0.021	0.017 ± 0.004	0.079 ± 0.043	0.006 ± 0.005	0.0008
	NODE	0.075 ± 0.039	0.0279 ± 0.0300	0.093 ± 0.028	∞	∞	∞	∞	0.0008
	DP	0.178 ± 0.028	0.0858 ± 0.0238	0.245 ± 0.015	0.156 ± 0.007	0.031 ± 0.001	0.154 ± 0.006	0.030 ± 0.001	0.0494
	iFlow	0.337 ± 0.264	0.2526 ± 0.3407	0.347 ± 0.296	0.921 ± 1.238	0.124 ± 0.166	0.907 ± 1.243	0.124 ± 0.166	**0.0007**
	SPDT	0.462 ± 0.001	0.2866 ± 0.0020	0.528 ± 0.019	$0.070^{*} \pm 0.022^{*}$	$0.014^{*} \pm 0.006^{*}$	$0.049^{*} \pm 0.025^{*}$	$0.014^{*} \pm 0.006^{*}$	0.0017
	OSMP (ours)	0.053 ± 0.002	0.0062 ± 0.0001	0.081 ± 0.000	$\mathbf{0.040 \pm 0.000}$	$\mathbf{0.005 \pm 0.000}$	$\mathbf{0.016 \pm 0.000}$	$\mathbf{0.004 \pm 0.000}$	0.0019
Image Cont.	MLP	0.124 ± 0.023	0.1049 ± 0.0237	0.106 ± 0.004	0.051 ± 0.003	0.006 ± 0.001	0.030 ± 0.005	0.004 ± 0.001	0.0012
	Elman RNN	0.318 ± 0.021	0.2793 ± 0.0197	0.397 ± 0.034	0.118 ± 0.013	0.010 ± 0.000	0.059 ± 0.020	0.004 ± 0.002	0.0012
	LSTM	0.494 ± 0.024	0.5415 ± 0.0071	0.719 ± 0.073	0.220 ± 0.023	0.012 ± 0.004	0.098 ± 0.032	0.003 ± 0.002	0.0385
	NODE	∞	∞	∞	∞	∞	∞	∞	**0.0009**
	DP	0.158 ± 0.016	0.0928 ± 0.0078	0.228 ± 0.008	0.113 ± 0.005	0.023 ± 0.002	0.125 ± 0.005	0.019 ± 0.001	0.0410
	iFlow	0.337 ± 0.114	0.2342 ± 0.1799	0.792 ± 0.280	0.688 ± 0.608	0.047 ± 0.045	0.679 ± 0.613	0.047 ± 0.045	0.0014
	SPDT	0.433 ± 0.000	0.3453 ± 0.0005	0.454 ± 0.009	0.114 ± 0.014	0.024 ± 0.000	$0.094^{*} \pm 0.002^{*}$	$0.023^{*} \pm 0.001^{*}$	0.0027
	OSMP (ours)	$\mathbf{0.033 \pm 0.009}$	$\mathbf{0.0129 \pm 0.0086}$	$\mathbf{0.050 \pm 0.001}$	$\mathbf{0.043 \pm 0.002}$	$\mathbf{0.004 \pm 0.001}$	$\mathbf{0.016 \pm 0.002}$	$\mathbf{0.003 \pm 0.000}$	0.0028
Turtle Swim.	MLP	0.138 ± 0.001	0.2089 ± 0.0003	0.294 ± 0.003	$\mathbf{0.084 \pm 0.001}$	$\mathbf{0.112 \pm 0.041}$	$\mathbf{0.036 \pm 0.009}$	$\mathbf{0.089 \pm 0.062}$	0.0008
	Elman RNN	0.174 ± 0.021	0.1756 ± 0.0286	1.165 ± 0.120	0.177 ± 0.037	0.126 ± 0.020	0.109 ± 0.056	0.120 ± 0.013	**0.0006**
	LSTM	0.476 ± 0.096	0.5712 ± 0.2284	2.748 ± 0.518	0.317 ± 0.095	0.042 ± 0.020	0.101 ± 0.070	0.008 ± 0.009	0.0008
	NODE	0.085 ± 0.022	0.0621 ± 0.0223	0.236 ± 0.016	5.133 ± 1.414	1.085 ± 0.247	3198 ± 1800	599 ± 332	0.0007
	DP	0.211 ± 0.014	0.1813 ± 0.0526	1.511 ± 0.175	0.183 ± 0.017	0.124 ± 0.022	0.190 ± 0.035	0.129 ± 0.004	0.0382
	iFlow	0.242 ± 0.144	0.2129 ± 0.2555	0.739 ± 0.329	$0.590^{*} \pm 0.323^{*}$	$0.183^{*} \pm 0.123^{*}$	0.391 ± 0.376	0.187 ± 0.096	0.0013
	SPDT	0.321 ± 0.009	0.2616 ± 0.0114	0.671 ± 0.029	0.331 ± 0.042	0.137 ± 0.005	0.233 ± 0.071	0.154 ± 0.027	0.0016
	OSMP (ours)	$\mathbf{0.009 \pm 0.000}$	$\mathbf{0.0056 \pm 0.0008}$	$\mathbf{0.052 \pm 0.000}$	0.135 ± 0.021	0.114 ± 0.015	0.045 ± 0.023	0.112 ± 0.011	0.0017

Results

Table 12.2 summarizes the quantitative benchmarking of OSMPs versus the baselines, assessing both imitation fidelity and convergence characteristics. To gauge imitation quality, we compute trajectory and velocity RMSE, together with the DTW distance, following prior work (Nawaz et al., 2024; Pérez-Dattari and Kober, 2023; Urain et al., 2020). Convergence is examined at two levels. For local convergence, the system is initialized near a demonstration; we roll out each policy for one estimated period and compare the resulting shapes using directed Hausdorff distance and MED after aligning the sequences with ICP. This scenario reflects small deviations from the desired limit cycle caused by low-level control errors or external disturbances. For global convergence, the system starts farther from the demonstrations. We roll out the motion policy for two full periods, then compute the same shape metrics on the second half of the rollout, allowing each policy sufficient time to settle into its limit cycle before measuring how closely that cycle matches the target demonstration.

Our benchmarks span several dataset categories. We include datasets used in earlier studies—such as the IROS letter drawings (Urain et al., 2020) and other 2-D shapes (Nawaz et al., 2024)—as well as particularly challenging 2-D image contours (e.g., Star, MIT CSAIL and TU Delft flame logos, Dolphin, Bat), whose tight curves and discontinuous velocity profiles test the methods limits. In addition, we employ turtle-swimming datasets generated from biologically inspired oracles; unlike previous work, these sequences require reproducing not only the positional trajectory but also its complex, nonlinear velocity profile.

Please note that we train a separate motion policy on each dataset contained in the dataset category and report the mean of all datasets and demonstrations contained in a dataset category. In order to give statistical relevance to the results, we conduct three training runs on each model+dataset combination, where we initialized the neural network weights in each run with a different random seed. Subsequently, we report the mean and standard deviation across the three random seeds.

The results in Tab. 12.2 indicate that, across most dataset categories and evaluation metrics, OSMPs outperform the baselines. They not only converge more reliably than neural policies without formal guarantees, but also imitate the demonstrations and align their limit cycles to intricate periodic shapes more accurately than other orbitally stable methods, including iFlow (Urain et al., 2020) and SPDT (Zhi et al., 2024). This conclusion is also supported by the qualitative benchmarking in Fig. 12.3, that shows while some of the baseline methods, such as MLPs, Neural ODEs, or SPDT (Zhi et al., 2024), might be sufficient for simpler oracles, such as the RShape (Urain et al., 2020), the planar drawing (Nawaz et al., 2024), or the Star oracle, but fail to track the periodic demonstration for more complex and highly curved oracles, such as the TUD-Flame or the Dolphin image contour.

Ablation Study Loss Functions

In order to quantify the impact of each loss term on the overall performance of OSMP, we conduct an ablation study and report the statistics of the evaluation metrics across three random seeds in Tab. 12.3 and qualitative results in Fig. 12.4. It can be seen that the combination of velocity imitation loss $\mathcal{L}_{\text{vi}}$, the encoder regularization loss $\mathcal{L}_{\text{er}}$, the limit cycle matching loss $\mathcal{L}_{\text{lcm}}$, and the time guidance loss $\mathcal{L}_{\text{tgd}}$ works the best for very complex

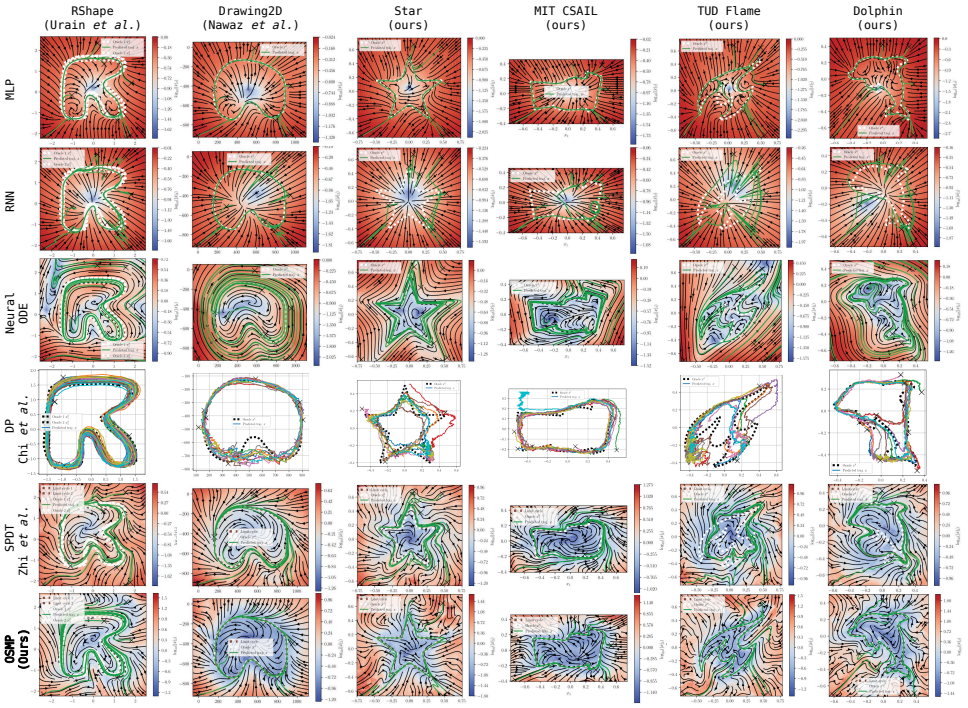


Figure 12.3: **Qualitative benchmarking of OSMP against baselines.** In this figure, we display the qualitative benchmarking results when comparing the proposed OSMP against baseline methods, such as MLPs, RNNs, NODEs, DPs (Chi et al., 2023), or SPDT (Zhi et al., 2024). The various columns represent oracles on which the motion policies were separately trained, shown as white dotted lines on the plots. The color map and the streamlines denote the velocity of the learned motion policy when evaluated on a grid. We initialize the trained motion policies at 10 different randomly sampled initial conditions and roll out their trajectory, visualized using solid green lines, for a duration of one period.

and curved oracles, such as the TUD-Flame. For "*easier*" oracles, such as the Ellipse and the Star, the encoder regularization loss and the time guidance loss can slightly degrade performance and can be left out of training. For very basic oracles, such as the Ellipse, even the limit cycle matching loss is not necessary, and it is sufficient to rely on the velocity imitation loss. Finally, while the Hausdorff (Hausdorff, 1914) loss is suitable for simple oracle shapes (Zhi et al., 2024), the limit cycle matching loss proposed in this paper is better suited for complex oracle shapes.

12.3.3 Stable and Accurate Tracking of Oracles Across Robot Embodiments

While the previous section focused on evaluating and benchmarking the learning of the OSMP, we now aim to demonstrate that the proposed OSMP can effectively control robot motion in real-world scenarios. To achieve this, we apply the method to a diverse range of robot embodiments, including robot manipulators (UR5), cobots (KUKA), continuum soft robots (Helix Soft Robot) (Guan et al., 2023), and prototypes of hybrid soft-rigid underwater

Table 12.3: **Quantitative results for ablation study on loss functions.** We report the mean ± std across three random seeds for each dataset evaluation. As a particular point of emphasis, we compare the limit cycle matching loss $\mathcal{L}_{\text{lcm}}$ proposed in this work with the Hausdorff distance loss as employed by Zhi et al. (2024) as they serve a similar purpose.

		Imitation Metrics			Local Convergence Metrics		Global Convergence Metrics	
Dataset	**Loss Config.**	**Traj. RMSE ↓**	**Norm. Traj. DTW ↓**	**Vel. RMSE ↓**	**Hausdorff Dist. ↓**	**ICP MED ↓**	**Hausdorff Dist. ↓**	**ICP MED ↓**
Ellipse	$\mathcal{L}_{\text{vi}}$	0.001 ± 0.000	0.0010 ± 0.0000	0.004 ± 0.000	**0.042 ± 0.000**	**0.002 ± 0.000**	0.002 ± 0.000	0.001 ± 0.000
	$\mathcal{L}_{\text{vi}} + \mathcal{L}_{\text{er}}$	0.001 ± 0.000	0.0007 ± 0.0001	0.004 ± 0.000	**0.042 ± 0.000**	**0.002 ± 0.000**	**0.001 ± 0.000**	**0.000 ± 0.000**
	$\mathcal{L}_{\text{vi}} + \mathcal{L}_{\text{er}} + \mathcal{L}_{\text{haus}}$	0.006 ± 0.002	0.0025 ± 0.0006	0.006 ± 0.000	**0.042 ± 0.000**	0.003 ± 0.000	0.005 ± 0.001	0.002 ± 0.001
	$\mathcal{L}_{\text{vi}} + \mathcal{L}_{\text{er}} + \mathcal{L}_{\text{lcm}}$	**0.000 ± 0.000**	**0.0006 ± 0.0001**	**0.004 ± 0.000**	**0.042 ± 0.000**	**0.002 ± 0.000**	**0.001 ± 0.000**	**0.000 ± 0.000**
	$\mathcal{L}_{\text{vi}} + \mathcal{L}_{\text{er}} + \mathcal{L}_{\text{lcm}} + \mathcal{L}_{\text{tgd}}$	**0.000 ± 0.000**	**0.0006 ± 0.0001**	**0.004 ± 0.000**	**0.042 ± 0.000**	**0.002 ± 0.000**	**0.001 ± 0.000**	**0.000 ± 0.000**
Star	$\mathcal{L}_{\text{vi}}$	0.153 ± 0.026	0.0817 ± 0.0193	0.124 ± 0.018	0.368 ± 0.065	0.068 ± 0.014	0.350 ± 0.055	0.062 ± 0.017
	$\mathcal{L}_{\text{vi}} + \mathcal{L}_{\text{er}}$	0.172 ± 0.041	0.1024 ± 0.0246	0.095 ± 0.008	0.393 ± 0.118	0.072 ± 0.018	0.340 ± 0.081	0.067 ± 0.014
	$\mathcal{L}_{\text{vi}} + \mathcal{L}_{\text{er}} + \mathcal{L}_{\text{haus}}$	0.077 ± 0.020	0.0144 ± 0.0040	0.089 ± 0.018	**0.041 ± 0.001**	0.006 ± 0.000	0.015 ± 0.001	0.005 ± 0.000
	$\mathcal{L}_{\text{vi}} + \mathcal{L}_{\text{er}} + \mathcal{L}_{\text{lcm}}$	**0.024 ± 0.006**	**0.0031 ± 0.0007**	**0.041 ± 0.007**	0.044 ± 0.002	**0.003 ± 0.000**	**0.013 ± 0.002**	**0.002 ± 0.000**
	$\mathcal{L}_{\text{vi}} + \mathcal{L}_{\text{er}} + \mathcal{L}_{\text{lcm}} + \mathcal{L}_{\text{tgd}}$	0.027 ± 0.007	0.0035 ± 0.0007	0.049 ± 0.008	0.049 ± 0.005	0.004 ± 0.000	0.017 ± 0.003	0.003 ± 0.000
TUD-Flame	$\mathcal{L}_{\text{vi}}$	0.116 ± 0.127	0.0879 ± 0.1198	0.107 ± 0.058	0.069 ± 0.042	0.009 ± 0.007	∞	∞
	$\mathcal{L}_{\text{vi}} + \mathcal{L}_{\text{er}}$	0.084 ± 0.070	0.0287 ± 0.0352	0.114 ± 0.058	0.082 ± 0.057	0.011 ± 0.010	0.066 ± 0.072	0.010 ± 0.011
	$\mathcal{L}_{\text{vi}} + \mathcal{L}_{\text{er}} + \mathcal{L}_{\text{haus}}$	0.375 ± 0.048	0.4067 ± 0.0289	0.180 ± 0.006	0.094 ± 0.020	0.016 ± 0.004	0.113 ± 0.030	0.010 ± 0.005
	$\mathcal{L}_{\text{vi}} + \mathcal{L}_{\text{er}} + \mathcal{L}_{\text{lcm}}$	0.113 ± 0.084	0.0249 ± 0.0282	0.114 ± 0.049	0.053 ± 0.018	0.006 ± 0.004	∞	∞
	$\mathcal{L}_{\text{vi}} + \mathcal{L}_{\text{er}} + \mathcal{L}_{\text{lcm}} + \mathcal{L}_{\text{tgd}}$	**0.033 ± 0.008**	**0.0025 ± 0.0002**	**0.073 ± 0.010**	**0.039 ± 0.003**	**0.003 ± 0.000**	**0.009 ± 0.001**	**0.002 ± 0.000**

robots (Crush turtle robot). Figure 12.5 illustrates the effectiveness of OSMPs across all tested robot embodiments.

First, we deploy the OSMPs trained on image contours on both the UR5 arm and the Helix soft robot, achieving accurate and stable contour tracking. The deviations and oscillations seen on the Helix stem not from the OSMP itself but from the low-level controller—particularly inverse-kinematics errors—as demonstrated in Fig. 12.6 and Movie S1[3], where we benchmark against a classical trajectory-tracking controller. Also, a quantitative evaluation of the imitation metrics and shape similarity between the actual system trajectory and the desired oracle shape is contained in Tab. 12.4.

We then target swimming behavior on the Crush Turtle robot using biologically inspired oracles collected by marine biologists. Our goal is for the OSMP to drive the two front flippers, the main propulsion surfaces. We use both a three-dimensional joint-space oracle (van der Geest et al., 2023) and a four-dimensional task-space oracle comprising flipper-tip position and twist (van der Geest et al., 2022), each derived from video recordings of green sea turtles (Chelonia mydas) (van der Geest et al., 2023, 2022). The resulting joint-space velocity commands are executed by the robot's actuators. Experiments show that OSMPs enable accurate tracking of the biological oracle at moderate speeds. Because of joint-motor velocity and acceleration limits, the system cannot perfectly track shape or speed at higher s_ω values; yet even when motion diverges slightly, stability is preserved, the trajectory rapidly reconverges to the oracle, and the turtle robot successfully swims. We observe that an OSMP trained on the joint-space swimming oracle yields more effective propulsion than one based on the task-space oracle—likely because the latter omits the full 3-D pose of the flippers. In addition, the joint-space OSMP avoids kinematic singularities that can destabilize task-space control.

Next, we test OSMP performance on kinesthetic-teaching demonstrations, which are typically jerkier and less smooth than the oracles above. In periodic demonstrations, the trajectory often fails to close exactly, leaving a spatial offset between start and end poses

[3]The movies are included in the Supplementary Material available at `https://doi.org/10.5061/dryad.4f4qrfjrd`.

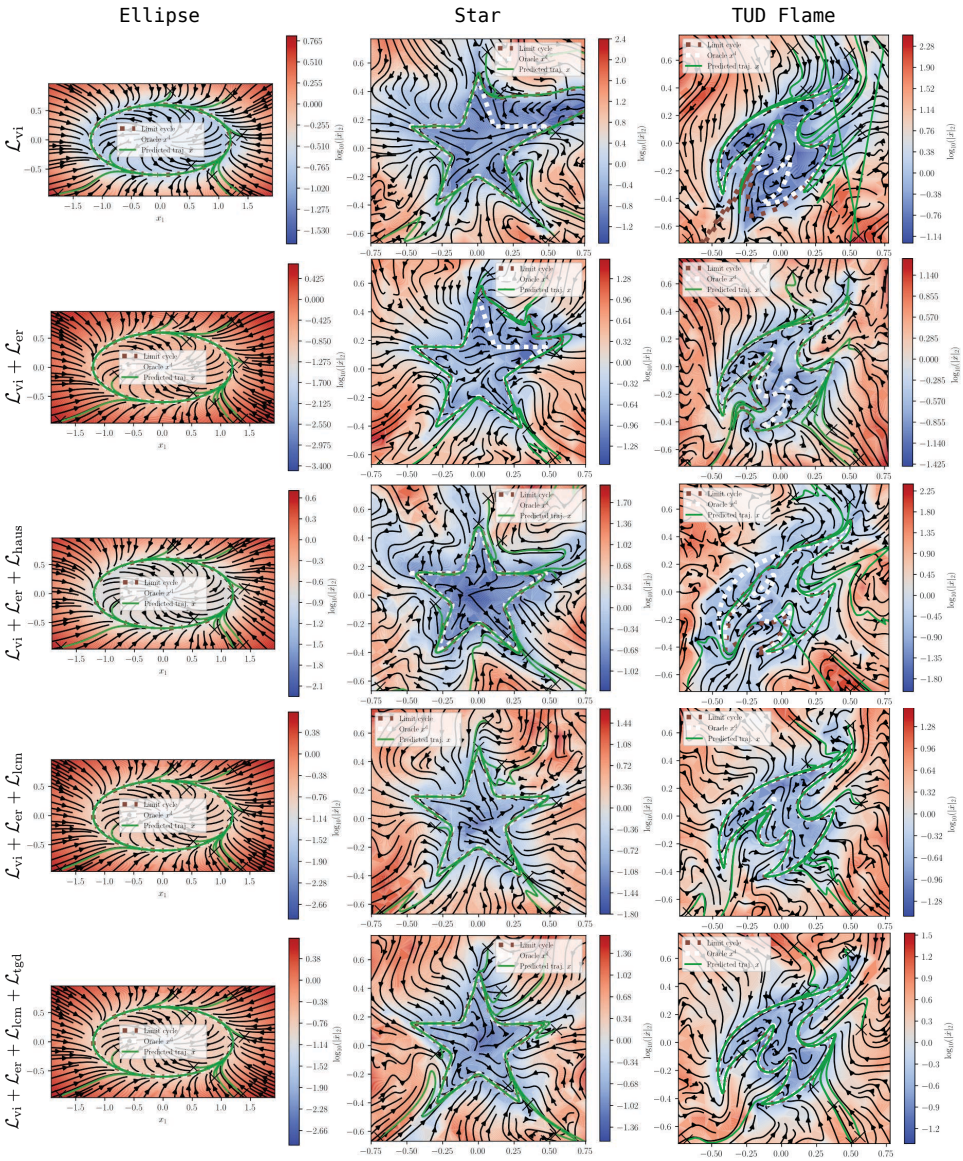


Figure 12.4: **Qualitative results for ablation study on loss functions.** This figure presents the qualitative results for the ablation study on the effect of the proposed loss functions on the imitation and convergence characteristics of the learned motion policy on a selection of datasets demonstrating that the full set of loss functions improves the quality of the learned velocity field on very complex and highly-curved oracles, such as the TUD-Flame logo dataset.

that complicates limit-cycle fitting. We investigate a whiteboard-erasure task on a UR5 manipulator and a brooming task on a KUKA cobot. For the UR5, we encode only the

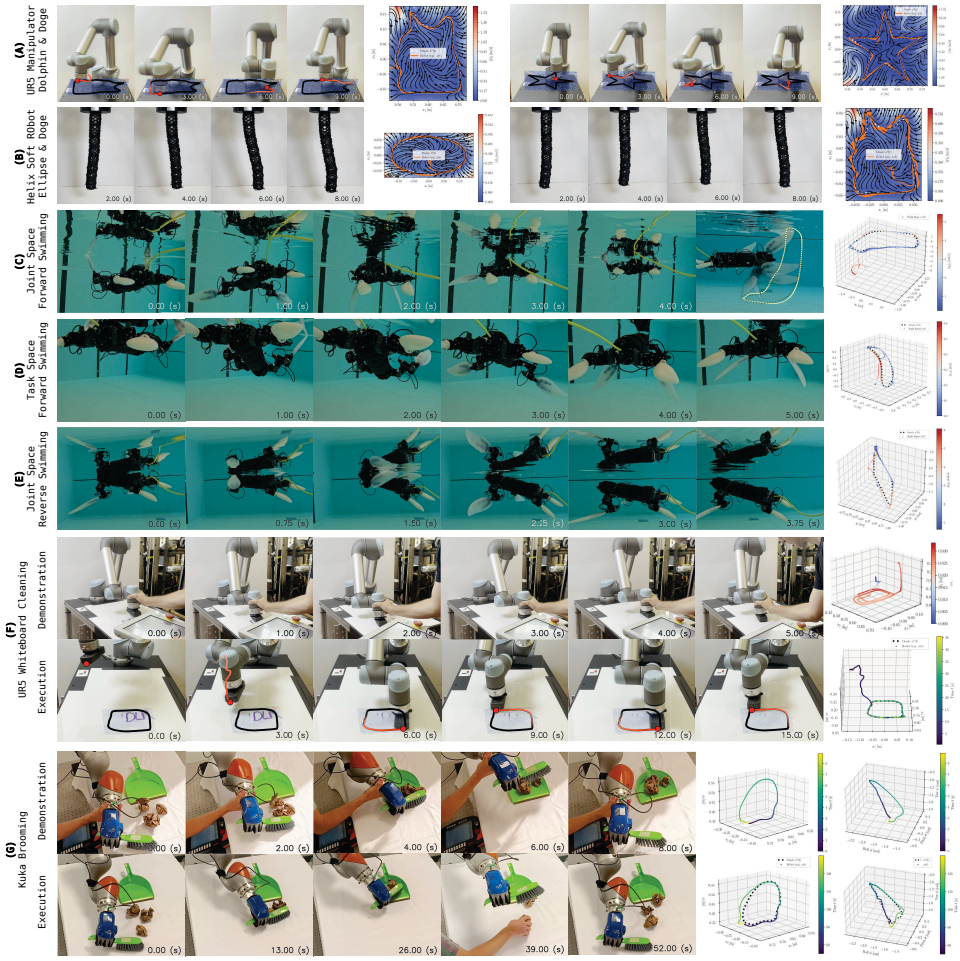


Figure 12.5: **Deployment of OSMPs on diverse robot embodiments.** We showcase the motion behavior generated by OSMPs deployed on various robot embodiments. **(A & B)** Tracking of image contours with the UR5 manipulator and the helix soft robot. The black line denotes the oracle shape and the red line the trajectory of the system over a horizon of the past 3 s. **(C-E)** Forward and reverse turtle robot swimming with the biological oracles defined either in joint or task space. **(F-G)** Demonstrations provided via kinesthetic teaching on whiteboard cleaning (UR5 manipulator) and brooming tasks (KUKA cobot) and subsequent execution of OSMPs trained on these demonstrations.

end-effector positions, whereas for the KUKA, we encode both position and orientation. On the UR5, we observe successful task completion (i.e., cleaning the writing from the whiteboard), rapid convergence to the limit cycle, and strong oracle tracking, with only minor errors where start and end points were fused. On the KUKA, tracking error is somewhat larger—likely due to the demanding six-DOF oracle and low feedback gains in the low-level controller—but the robot still completes the task reliably and repeats it with high consistency, even remaining robust to external disturbances and perturbations as seen in Movie S2.

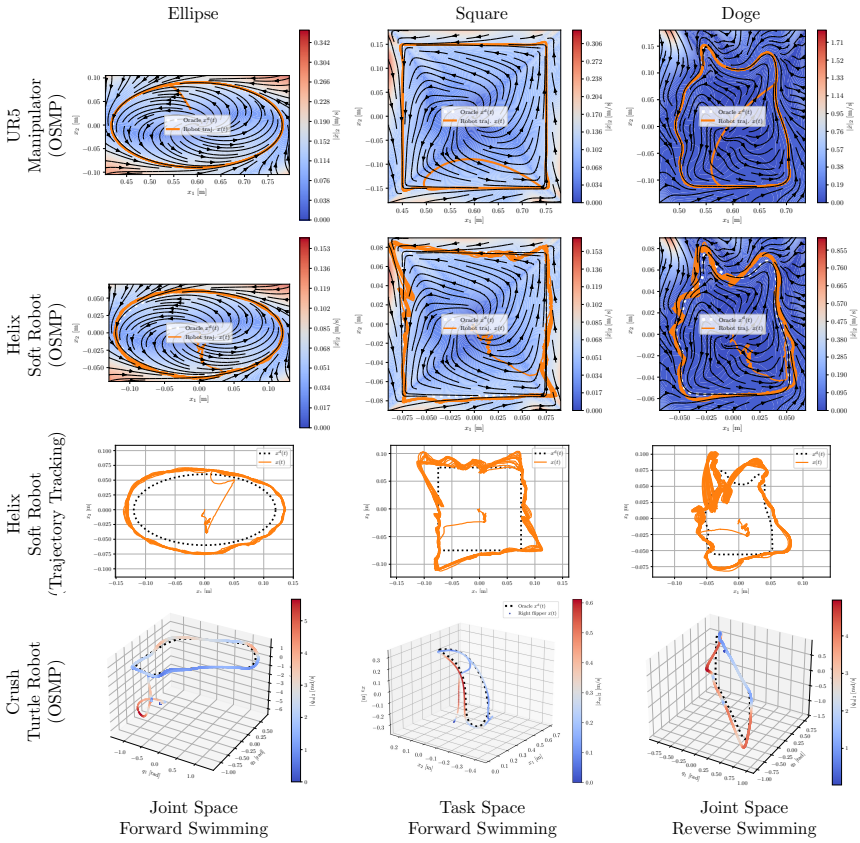


Figure 12.6: **Supplementary results for stable tracking of oracles across robot embodiments in the real world.** This figure shows the trajectories of various robot embodiments controlled with OSMPs based on data recorded during real-world experiments. The first row shows the behavior of a UR5 manipulator in task space where the OSMP is trained on various geometric shape oracles, including an ellipse, a square, and a doge. The black dotted lines denote the oracle, the orange line is the actual trajectory of the UR5's end-effector, and the velocity field is based on the learned OSMP. The second row considers the same trained OSMPs, but this time evaluates their behavior on a Helix soft robot. The third row presents the behavior of classical trajectory tracking controllers on the Helix soft robot for the same oracles. Finally, the fourth row contains measurements of the Crush turtle robot operated (in water) by OSMPs trained on biological oracles, where the black dotted line denotes the oracle and the colored line the actual behavior of the right flipper arm.

Table 12.4: **Imitation and shape similarity evaluation metrics for real-world robot experiments.** We report the imitation metrics that evaluate the similarity between the actual robot behavior and the oracle/demonstration. Furthermore, we report shape similarity metrics between the actual trajectory shape and the desired oracle. In order to compute the metrics, we first identify the alignment between the first oracle time step and the actual trajectory by identifying the time shift that minimizes the error between the two. Subsequently, we evaluate the similarity between the oracle and the system trajectory that was resampled to the same timings for the imitation metrics and to the same number of steps for the shape similarity metrics, respectively. For the Helix soft robot experiments, we also compare the performance of OSMP against a classical TT controller.

Robot	Task/Oracle	Speed-up Factor s_ω	Motion Policy	Imitation Metrics				Shape Similarity Metrics	
				Traj. RMSE ↓	Norm. Traj. DTW ↓	Traj. Frechet Dist. ↓	Vel. RMSE ↓	Hausdorff Dist. ↓	ICP MED ↓
UR5	Square	2.0	OSMP	0.0094 m	0.0014	0.016 m	0.0225 m/s	0.0077 m	0.0012 m
	Star	2.0	OSMP	0.0390 m	0.0083	0.111 m	0.0802 m/s	0.0102 m	0.0019 m
	Doge	2.0	OSMP	0.0090 m	0.0018	0.007 m	0.0230 m/s	0.0071 m	0.0016 m
	MIT-CSAIL	2.0	OSMP	0.0178 m	0.0027	0.041 m	0.0426 m/s	0.0063 m	0.0015 m
	Dolphin	0.5	OSMP	0.0152 m	0.0016	0.037 m	0.0166 m/s	0.0033 m	0.0007 m
	Whiteboard Cleaning	0.5	OSMP	0.0136 m	0.0024	0.031 m	0.0240 m/s	0.0142 m	0.0016 m
Turtle Robot	Joint Space Forw. Swim.	0.5	OSMP	0.2237 rad	0.0398	0.536 rad	0.3673 rad/s	0.0854 rad	0.0215 rad
	Joint Space Forw. Swim.	1.0	OSMP	0.2544 rad	0.0660	0.590 rad	0.8805 rad/s	0.1790 rad	0.0419 rad
	Joint Space Rev. Swim.	1.5	OSMP	0.6715 rad	0.3609	1.835 rad	2.1995 rad/s	0.3815 rad	1.4600 rad
	Task Space Forw. Swim.	0.5	OSMP	0.0620 m	0.0244	0.083 m	0.0609 m/s	0.0517 m	0.0125 m
Helix Soft Robot	Ellipse	2.0	TT	0.0096 m	0.0126	0.017 m	0.0119 m/s	0.0175 m	0.0122 m
	Ellipse	2.0	OSMP	0.0308 m	0.0066	0.055 m	0.0195 m/s	0.0061 m	0.0020 m
	Square	2.0	TT	0.0166 m	0.0123	0.039 m	0.0526 m/s	0.0388 m	0.0099 m
	Square	2.0	OSMP	0.0216 m	0.0073	0.067 m	0.0302 m/s	0.0112 m	0.0026 m
	Doge	2.0	TT	0.0128 m	0.0141	0.033 m	0.0514 m/s	0.0290 m	0.0101 m
	Doge	2.0	OSMP	0.0059 m	0.0039	0.011 m	0.0182 m/s	0.0108 m	0.0033 m

12.3.4 The Learned Policies Exhibit Compliant and Natural Motion Behavior

We aim for robots in human-centric environments to demonstrate robust, compliant, and predictable behavior. Specifically, *robustness* means that if a robot deviates from its intended path—perhaps due to a disturbance—it will always converge back to the desired motion. *Compliance* indicates that robots should exert only minimal forces when coming into contact with humans, and *predictability* ensures that their motions are sufficiently consistent for humans to anticipate their behavior and respond appropriately.

In this section, we compare the reaction upon disturbances and perturbations of OSMPs against classical trajectory tracking controllers that rely on a time-parametrized trajectory, given in the form

$$\dot{x}(t) = \dot{x}^{\mathrm{d}}(t) + k_{\mathrm{p}}\,(x^{\mathrm{d}}(t) - x(t)), \tag{12.68}$$

where $k_{\mathrm{p}} \in \mathbb{R}$ is a proportional feedback gain that operates on the error between the current position $x(t)$ and the desired position $x^{\mathrm{d}}(t)$. We stress here the reliance on a time-parametrized trajectory provided in the form $(x^{\mathrm{d}}(t), \dot{x}^{\mathrm{d}}(t))\,\forall\, t \in [t_0, t_{\mathrm{f}}]$. We evaluate three motion controllers: a pure feedforward trajectory tracking controller, which we gather by setting $k_{\mathrm{p}} = 0$, an error-based feedback controller with $k_{\mathrm{p}} > 0$, and the learned OSMP. In this setting, we are particularly interested in analyzing the behavior of the motion controllers upon encountering an external disturbance that perturbs the state of the system with respect to the time reference. For example, in simulation, we shift the time reference when initializing the system by half a period (i.e., a phase shift of π rad) and in the real world experiments with the KUKA robot running a low level impedance controller we apply external perturbations to the system that prevents or disturbs the nominal motion.

The results in Fig. 12.7 and Movie S3 show that the pure feedforward trajectory tracking controller entirely drifts off the desired trajectory. When adding an error-based feedback

term, the classical trajectory tracking controller is able to recover and rejoin the demonstrated trajectory after a bit. However, while doing so, the feedback term generates a very aggressive correction action, which could cause incompliant behavior and would not seem natural to humans. Instead, the OSMP, which is solely conditioned on the system state and not time, is not affected by the perturbation of the time reference and perfectly tracks the demonstration, immediately returning to the closest point on the limit cycle after a perturbation, while exhibiting compliant and natural behavior.

12.3.5 Achieving Phase Synchronization Across Multiple Motion Primitives

In many practical applications, such as locomotion or bimanual manipulation, synchronizing multiple motion policies is critical. In this section, we illustrate how our approach can synchronize multiple learned OSMPs by evaluating the polar phase of each and then aligning them via an error-based feedback controller (Dörfler and Bullo, 2014). Crucially, we only adjust the velocity magnitude without altering the system's spatial motion, thereby preserving the imitation and convergence properties of each learned motion policy.

In Fig. 12.8 and Movie S4, we show simulation and experimental results for synchronizing between two and six OSMPs. The simulation outcomes illustrate how the controller identifies the most efficient strategy to align the OSMPs, achieving rapid polar phase synchronization. A proportional gain determines the aggressiveness of the synchronization process. The simulation results confirm that the phase synchronization approach is effective not only for two systems but also for three or more. In complex systems with many DOFs, we have found it can be more effective to train separate OSMPs and synchronize them during execution rather than relying on one large OSMP that covers every DOF. With a joint OSMP, a disturbance in even a few DOFs can pull the rest off the limit cycle—and away from the oracle—until the system reconverges. By contrast, synchronized but independent OSMPs are insulated from such disturbances: if some DOFs are perturbed, the remaining DOFs managed by their own OSMPs can keep tracking the oracle accurately, and the phase synchronization ensures that all OSMPs stay locked in their phase.

Regarding experimental findings, we examined the swimming performance of the Crush turtle robot. Tests in a swimming pool revealed that the robot can swim effectively only when both front flippers—the primary means of locomotion in water (van der Geest et al., 2023, 2022)—are fully synchronized. In practice, even aside from external disturbances and inherent differences between the flippers, desynchronization occurs already during initialization when the flipper arms start in slightly different configurations with varying polar phases. Our results show that using our method, the two flipper arms synchronize, even with flippers initialized far from the oracle, within 4.72 s to less than 1° in phase error, and subsequently for the entire experiment exhibit a mean phase error of less than 0.2°, thereby enabling the turtle robot to swim effectively.

12.3.6 Smooth Interpolation Between Motion Behaviors via Encoder Conditioning

As robotics shifts toward generalist motion policies that choose among varied behaviors based on task, state, and perception, those policies must support multiple skills rather than one (Black et al., 2024; O'Neill et al., 2024; Team, 2025). Leveraging semantic cues—e.g.,

embeddings from vision–language models—could supply such conditioning, yet dynamic motion-primitive work seldom tackles it. Existing methods (Pérez-Dattari et al., 2024; Pérez-Dattari and Kober, 2023; Rana et al., 2020; Sochopoulos et al., 2024; Zhi et al., 2024) also lack smooth interpolation between trained behaviors. Examples include applications such as surface cleaning, where the robot must in the future seamlessly switch wiping motions as materials change. In locomotion, blending oracles for flat walking and stair climbing enables natural movement over moderately stepped terrain. Such cases underscore the need for motion policies that transfer to unseen tasks with few- or zero-shot generalization (Jaquier et al., 2025).

We introduce task conditioning in the bijective encoder through a scalar variable $z \in \mathbb{R}$, allowing the desired motion behavior to be selected online by simply setting z. To ensure the learned policy transitions smoothly across behaviors (e.g., for $z \in [-1, 1]$), we add a loss term $\mathcal{L}_{\text{sci}}$ during training. As illustrated in Fig. 12.9 and Movie S5, both simulation and hardware experiments with the turtle robot show that (a) a single OSMP faithfully reproduces all behaviors encountered during training, and (b) $\mathcal{L}_{\text{sci}}$ promotes smooth interpolation between oracles, enabling meaningful zero-shot performance on unseen tasks that fall within the training distribution. An ablation study—comparing against an OSMP trained without $\mathcal{L}_{\text{sci}}$—confirms this finding. Crucially, switching behaviors requires no elaborate sequence: once z is updated, the OSMP's convergence guarantees rapidly steer the system to the new behavior, preserving EOS for constant or slowly varying conditioning values.

12.3.7 Inference Time Benchmarking

We benchmarked multiple inference modes and Jacobian-estimation schemes for an OSMP trained with the CorneliaTurtleRobotJointSpace oracle; the outcomes are listed in Tab. 12.5. Here, "inference time" is the wall-clock duration—measured on an Apple MacBook M4 Max CPU—for a single forward pass (batch size = 1), averaged over 1000 runs.

Because numerical Jacobians reduce accuracy, we also computed the velocity RMSE with respect to the demonstrated velocities. In addition, starting from the oracle's initial states, we determined the trajectory RMSE. Unlike Section 12.3.2, we did not enforce the demonstration's fixed integration step; instead, each rollout used a step size equal to the model's measured inference time. Consequently, faster models are integrated with finer temporal resolution, potentially achieving higher numerical accuracy.

The results reveal an interesting trade-off: models with an exact Jacobian run more slowly, whereas those using approximate Jacobians support higher control rates and can sometimes track the trajectory more precisely thanks to the finer integration grid. Switching from analytical to numerical Jacobians increases the velocity RMSE by 12.5 %, yet the lowest trajectory RMSE is obtained with an Ahead-Of-Time (AOT)-compiled model that employs numerical Jacobians — 36 % lower than an eager-mode baseline with autograd Jacobians. This improvement stems from the roughly ninefold reduction in integration step size made possible by the compilation speed-up.

We note that this Trajectory RMSE tradeoff highly depends on the specific trained model, the oracle, and the available inference hardware. For example, the advantages of a high control rate are particularly pronounced where fast velocities are needed, such as in the case of a turtle swimming. Therefore, we recommend a separate analysis for each use case if losing accuracy via numerical Jacobians can be outweighed by a faster inference

Table 12.5: **Numerical Jacobians allow for compilation of the motion policy and inference rates of up to** 15 000 Hz **on modern CPUs.** We evaluated the OSMP model trained on the *CorneliaTurtleRobotJointSpace* dataset with $n = 3$ and report the inference time, the Velocity RMSE, and the Trajectory RMSE for various inference modes and methods for computing the Jacobian of the encoder J_Ψ. It is crucial to describe the evaluation procedure of the Trajectory RMSE. We begin by measuring the inference time for a single motion-policy step, from which we derive the control frequency at which the policy can run. We then execute the policy using a time step equal to this measured inference time, so a faster model, with its shorter step, can integrate the trajectory more finely and may, therefore, attain higher accuracy as long as it doesn't compromise the performance with a lowered velocity prediction accuracy.

Inference Mode	Jacobian Method	Inference Time [ms] ↓	Vel. RMSE ↓	Traj. RMSE ↓
Eager	Autograd	0.563	**0.00841**	0.0315
Eager	VJP	0.474	**0.00841**	0.0326
Compiled	VJP	0.116	**0.00841**	0.0270
Eager	Numerical	0.441	0.00946	0.0236
Compiled	Numerical	0.117	0.00946	0.0212
Export	Numerical	0.135	0.00946	0.0212
AOTInductor	Numerical	**0.064**	0.00946	**0.0202**

time.

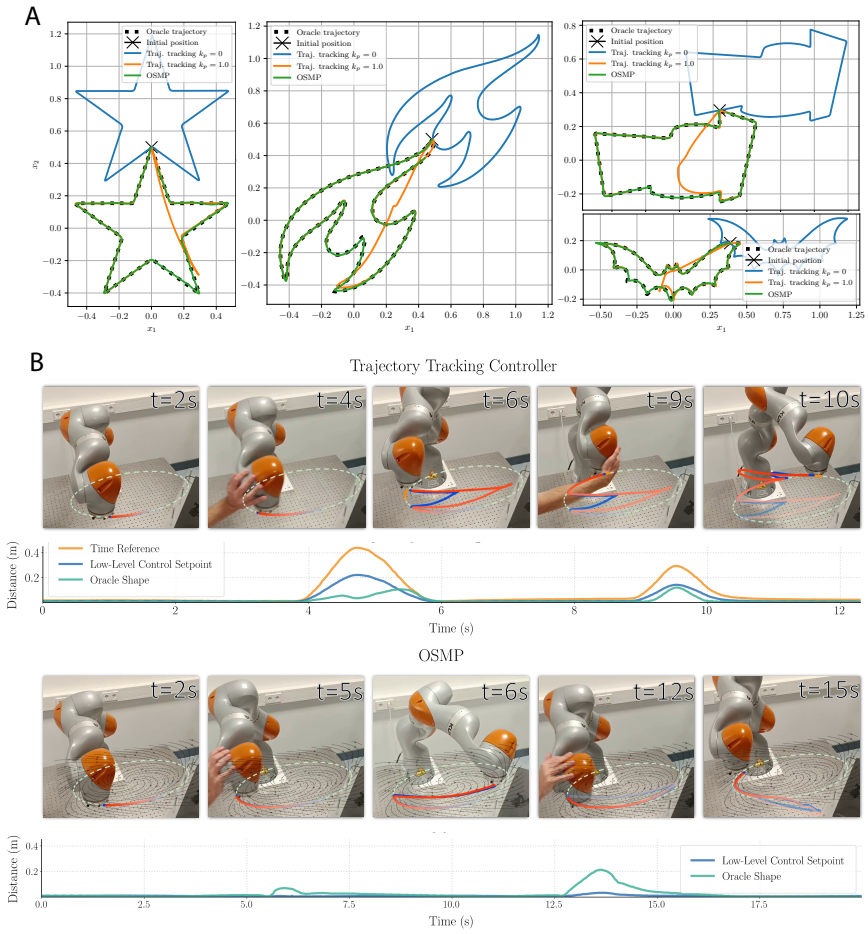


Figure 12.7: **OSMPs exhibit compliant and natural motion behavior.** In this figure, we analyze the motion behavior of OSMPs upon perturbations and interaction with humans and the environment. We compare the behavior of OSMPs against classical error-based trajectory tracking controllers $\dot{x}(t) = \dot{x}^{\mathrm{d}}(t) + k_{\mathrm{p}}(x^{\mathrm{d}}(t) - x(t))$ that operate on a time-parametrized reference $\dot{x}^{\mathrm{d}}(t)$, where $k_{\mathrm{p}} \in \mathbb{R}_{\geq 0}$ is the proportional control gain. **(A)** Simulations with time perturbations where we compare the behavior of traditional, time-parametrized trajectory tracking controllers against the OSMP. Here, the dashed black lines denote the oracles/demonstrations, the solid blue lines the behavior of a pure feedforward trajectory tracking controller with zero feedback gain $k_{\mathrm{p}} = 0$, the orange line the behavior of an error-based feedback trajectory tracking controller with $k_{\mathrm{p}} = 1$, and the green line the behavior of the learned OSMP. Compared to nominal scenarios, we perturb the time reference - i.e., the time reference exhibits a π offset in phase with respect to the initial position. **(B)** Experiments on a KUKA cobot that runs a compliant low-level impedance controller where a human interacts with the robot and exerts disturbances on the robot, comparing the behavior of the trajectory tracking controller with a feedback gain $k_{\mathrm{p}} = 1$. We also plot the Cartesian distance with respect to the time reference (trajectory tracking controller only), the next low-level control setpoint $x*(t)$, and the closest point on the oracle over time. Experiments on a KUKA cobot equipped with a compliant, low-level impedance controller while a human interacted with the robot and deliberately introduced disturbances/perturbations. The study compares the trajectory-tracking controller's behavior with a feedback gain of $k_{\mathrm{p}} = 1$ against that of the OSMP. We also plot, over time, the Cartesian distance to the time-indexed reference (trajectory-tracking controller only), the next low-level control setpoint $x^*(t)$, and the distance to the nearest point on the Ellipse oracle. In the snapshots, the red line represents the past trajectory, and the blue line the trajectory of past low-level control setpoints.

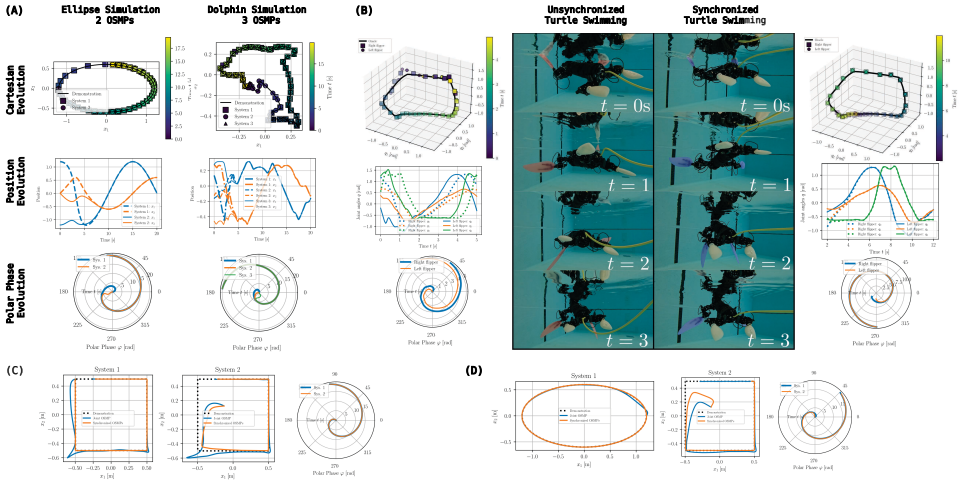


Figure 12.8: **The error-based feedback controller successfully synchronizes multiple OSMPs in their phase.** Results for the phase synchronization of multiple motion policies. **(A)** Simulation results for the phase synchronization of two and three OSMPs on the Ellipse and Dolphin oracles, respectively. The first row shows the Cartesian-space evolution of each OSMP, where the color of the markers communicates the time information. The second row shows the position vs. time, and the last row shows the polar phase φ of the systems over time. **(B)** Experimental results for phase synchronization of the robotic turtle flippers for turtle swimming based on the joint-space oracle. Naturally, the flippers are initialized at the start of the experiment at slightly different positions, causing a phase offset. If this phase offset is not corrected and the flippers remain unsynchronized (left side), the turtle robot doesn't swim or only very slowly. On the right side, the flippers rapidly synchronize, which leads to successful swimming. **(C & D)** Comparison of a "*large*", joined OSMP trained on two systems compared to "*small*" OSMPs trained on each system separately and then synchronized during inference. When one of the systems is initialized off the oracle/perturbed, this leads the other system controlled by the joined OSMP to drift off its own limit cycle as well. Contrarily, the synchronized separate OSMPs do not cause each other to drift off the limit cycle.

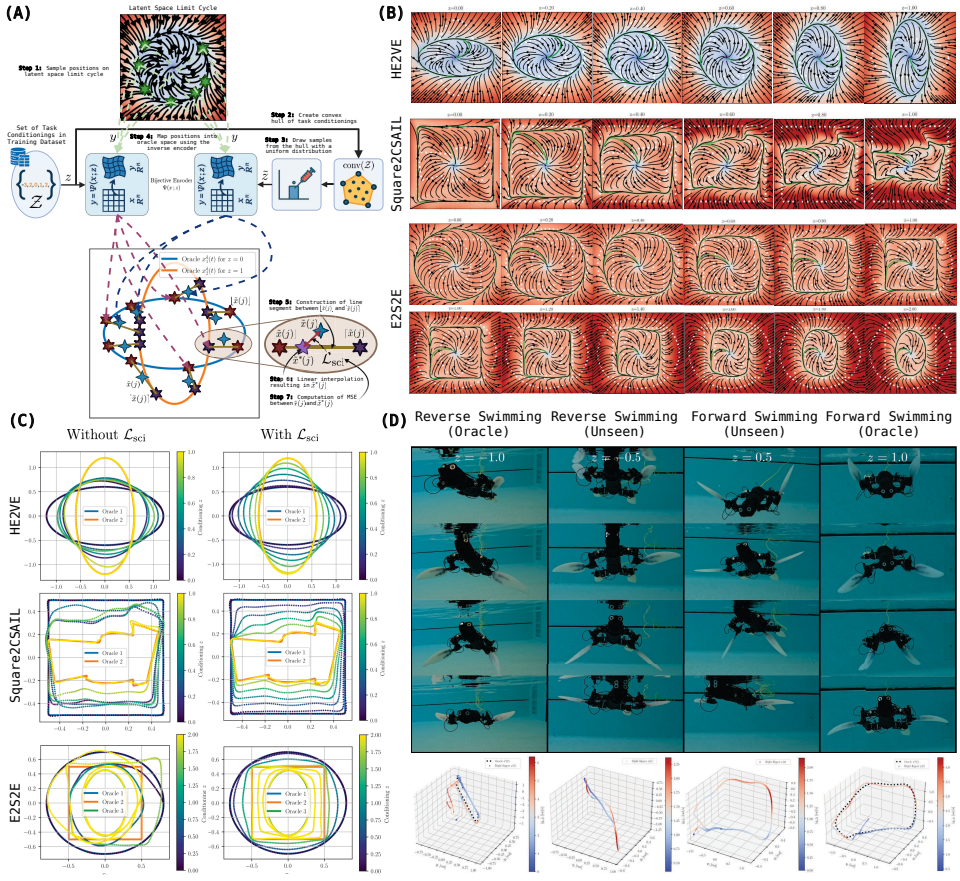


Figure 12.9: **Smooth interpolation between distinct motion behaviors via encoder conditioning.** Results demonstrating the conditioning of the encoder on multiple learned motion behaviors, including smooth interpolation between behaviors. **(A)** Graphic explaining the formulation of the $\mathcal{L}_{\text{sci}}$ loss. **(B)** OSMPs trained with $\mathcal{L}_{\text{sci}}$ on multiple image contour transition datasets - horizontal ellipse to vertical ellipse (HE2VE), square to CSAIL logo (Square2CSAIL), and horizontal ellipse to square to - evaluated for various, both seen and unseen during training, task conditioning values. **(C)** Ablation study for OSMPs trained without and with the smooth conditioning interpolation loss $\mathcal{L}_{\text{sci}}$, where we smoothly vary the conditioning value over the duration of the trajectory (e.g., $z = 0$ at $t = 0$ to $z = 1$ at $t = 140$ s) **(D)** Turtle swimming motion for various conditioning values, ranging from $z = -1$ (reverse swimming joint space oracle) to $z = 1$ (forward swimming joint space oracle), with $z = -0.5$ and $z = 0.5$ unseen during training.

12.4 Discussion

Convergence Guarantees The first result introduced in the previous section is theoretical in nature: we establish that the proposed OSMP guarantees almost-global asymptotic orbital stability. Under mild architectural conditions, this further strengthens into almost-global exponential orbital stability via transverse contraction. Yet this result is not merely of theoretical interest—it carries profound practical implications. Indeed, it ensures that, regardless of network weights or conditioning, any trajectory converges exponentially fast to the learned limit cycle. Exponential convergence prevents slow-dynamics plateaus where the system might otherwise stall, while the near-global basin drastically reduces the need for far-reaching training data coverage. Together, these properties yield significant gains in data efficiency, as they obviate the need to densely sample the state space. Beyond training, exponential convergence also facilitates system-level analysis and enhances modularity. Most critically, contractiveness plays a central role in enabling policy reuse and transfer learning: while interconnecting merely stable systems can lead to instability, contracting systems are provably composable and lend themselves naturally to transfer and hierarchical control (Angeli et al., 2025; Ofir et al., 2022).

In the same spirit, it is important to underscore that our stability results hold uniformly for any fixed conditioning value z, thereby reinforcing their relevance for transferability. This property suggests robustness not only to different instances within a task family but also to transitions across them. While outside the current scope, it is natural to ask whether convergence still holds under time-varying conditioning $z(t)$, such as during policy switching. Thanks to the global nature of the underlying result, the extension to the case where $z(t)$ eventually stabilizes (i.e., $z(t) \equiv z_\infty$ for $t > t'$) is straightforward. More ambitiously, we believe that the exponential convergence rate can be leveraged to establish stability even under piecewise-constant $z(t)$, including non-smooth transitions over finite horizons. This opens the door to formal tools for designing learning-based schedules of optimal conditioning patterns—a direction we leave for future investigation.

Quantitative Benchmark Quantitative benchmarks show that OSMPs surpass most baselines across the majority of dataset categories. They display superior convergence — especially when initialized far from the demonstration—compared with classic neural motion policies such as MLPs, RNNs, LSTMs, NODEs, and even state-of-the-art DPs (Chi et al., 2023). While other orbitally stable approaches like iFlow (Urain et al., 2020) and SPDT (Zhi et al., 2024) guarantee convergence to a limit cycle, they often struggle to (a) imitate highly curved shapes or discontinuous velocity profiles and (b) ensure that the resulting limit cycle accurately reproduces those complex demonstrations. For example, compared to SPDT (Zhi et al., 2024), OSMPs are able to imitate the velocity profile, and their limit cycle captures the oracle shape 5x more accurately, as the global convergence analysis shows. The few cases in which baselines outperform OSMPs highlight avenues for improvement. On the IROS letters dataset (Urain et al., 2020), DPs (Chi et al., 2023) achieve lower imitation errors, likely because widely separated demonstrations of the same letter must be captured by a single policy, which favors probabilistic methods like DPs and iFlow over deterministic ones, such as NODEs, SPDT, and OSMPs. For image-contour data, MLP policies slightly edge out OSMPs when starting near the demonstration but degrade markedly when initialized farther away. Regarding inference time/runtime, OSMPs demand

more computation than classical neural policies (which forgo gradient evaluations during inference) and iFlow (Urain et al., 2020) (given our more expressive bijective encoders), yet they strike a favorable performance-to-inference-time balance and are 10–20× faster than standard diffusion policies (Chi et al., 2023). Note, too, that the timings in Table 12.2 stem from unoptimized, eager-mode inference; with numerical gradients and PyTorch ahead-of-time compilation, OSMPs can run at up to 15 kHz on modern CPUs, as shown in Tab. 12.5.

Real-World Experiments When moving to real-world experiments, we quickly realized how the interpretable latent-dynamics structure allows us to inspect post-training how closely the learned cycle mirrors the periodic demonstration, promoting predictable behavior when the policy is deployed on a real robot. This aspect will require further quantification in future work.

On the UR5 robot, OSMPs are able to track the oracle with directed Hausdorff distance between 3 mm and 14 mm. As the Helix soft robot employs a less accurate low level controller, the shape tracking accuracy drops to $6-11$ mm. Still, OSMPs outperform classical trajectory tracking controller in terms of shape accuracy on the Helix soft robot by approximately $60-65$ %.

OSMPs enable highly successful locomotion in the turtle robot by leveraging biologically derived swimming oracles. Although oracle tracking is not perfectly precise (directed Hausdorff distance of 0.085-0.382 rad)—especially at higher speeds due to motor limits and external disturbances like water drag—the system remains stable, quickly reconverges to the limit cycle, and keeps the limbs nearly perfectly synchronized.

Next, we trained individual OSMPs on multiple kinesthetic-teaching demonstrations for the UR5 and KUKA manipulators. Although some demonstrations were jerky and uneven, the OSMPs accurately captured the intended motions: the UR5 achieved a 100 % success rate in cleaning the whiteboard using the learned policy. Similarly, the KUKA robot consistently completed the brooming task over many repetitions—despite limited execution speed and tracking accuracy imposed by the low feedback gains of its impedance controller—and even succeeded when its motion was perturbed by external disturbances.

Although orbital stability is always guaranteed, large deformations in the learned diffeomorphism can push the system far from the demonstrated path. On real robots, that drift is risky because joint position and velocity limits—or a finite task-space workspace—can be violated. We saw this on the Crush turtle robot, whose joint range and velocity/acceleration caps constrain what the low-level controller can track. Two design choices proved helpful: (i) encoder regularization during training and (ii) a sliding-mode–style motion modulation that first draws the system into a neighborhood of the oracle before advancing along the polar phase. Since motion directly on the oracle is usually feasible, these measures prevent most of the problems. Looking ahead, embedding CBFs could explicitly keep the system out of infeasible or unsafe regions in the oracle space, echoing recent advances in the DMP literature (Davoodi et al., 2022; Mohammadi et al., 2024; Nawaz et al., 2024; Simmoteit et al., 2025).

Phase synchronization is vital when deploying OSMPs for turtle swimming: without it, the limb controllers—lacking an explicit time parameter—would drift out of phase, sharply increasing the cost of transport. Instead, our phase synchronization strategy is able to keep

the mean phase error between the two flippers at less than 0.2°. The same synchronization strategy can be applied in the future to other platforms, such as bipedal or quadrupedal robots. We also demonstrate that training separate OSMPs and synchronizing their phases at runtime can outperform a single large OSMP, as disturbances affecting one subsystem do not pull the others away from their limit cycles.

Conditioning DMPs on task information enables them to produce purposeful motions for tasks never encountered during training, marking a paradigm shift that opens the door to zero-shot transfer of convergence-guaranteed motion policies to far more complex, unseen scenarios. Here, we take an initial step toward that vision: our training procedure enables smooth interpolation between motions observed in the dataset. For instance, a single OSMP trained on both forward and reverse turtle swimming can effortlessly blend the two behaviors into a continuum. Crucially, the orbital stability is always preserved, even when the shape of the stable limit cycle changes as a function of the conditioning z.

Limitations The proposed OSMP exhibits several limitations that highlight promising directions for future development. First, it presumes prior annotation of the demonstration's attractor regime, requiring that each trajectory be segmented to isolate the periodic portion and implicitly modeled as a limit cycle. While our approach accommodates sharp turns and discontinuities more robustly than existing baselines, such features still induce sizable deformations in the learned vector field. These deformations manifest as locally aggressive dynamics, where nearby trajectories may temporarily diverge markedly before reconverging toward the limit cycle.

Second, like other time-invariant dynamical motion primitive frameworks (Ijspeert et al., 2013, 2002; Pérez-Dattari and Kober, 2023; Rana et al., 2020), OSMP inherits the limitation of being ill-posed under intersecting demonstrations or oracles—situations that demand multi-valued flows at a single point in state space. In addition to existing ideas in literature (Sun et al., 2024), formulating the motion policy as second-order dynamics or augmenting the dynamics with an explicit notion of trajectory progress could resolve this ambiguity by lifting the state into a higher-dimensional phase space.

Looking ahead, we envision extending the framework to support multiple classes of attractors—such as point attractors with GAS, multistable basins with Multistability (MS), and limit cycles with Orbital Stability (OS)—within a single primitive. The supercritical Hopf bifurcation already captures a transition between equilibrium and periodic behavior, suggesting that the inclusion of an additional "attractor type" parameter could generalize the formalism to support richer behaviors (Strogatz, 2018). Finally, replacing the explicit conditioning variable z with implicit observation-based embeddings (e.g., task images or object states) would make OSMP compatible with vision-language-action models such as π_0 (Black et al., 2024) and SmolVLA (Shukor et al., 2025), offering a path toward more expressive and versatile control policies.

12.5 Conclusion

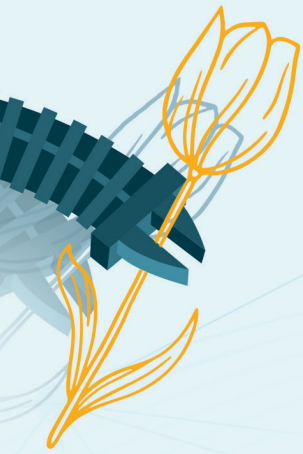
In this chapter, we introduced an approach for learning periodic/cyclical motions from demonstrations using OSMPs, where—similar to DMPs (Ijspeert et al., 2013, 2002)—the motion policy is parametrized by a dynamical system. We combined the advantages of ML

methods, such as their high expressiveness, with nonlinear system theory to ensure that the periodic motion is tracked via a limit cycle that exhibits AOS and transverse contraction. This is achieved by integrating a learned bijective encoder based on Euclideanizing flows (Dinh et al., 2016; Rana et al., 2020), which establishes a diffeomorphism between the oracle and latent space, with fixed, inspectable dynamics in latent space provided by a supercritical Hopf bifurcation (Strogatz, 2018). Compared to previous work (Zhi et al., 2024), our approach significantly improves the accuracy of matching the periodic demonstration to the generated limit cycle. Furthermore, benchmarking against standard ML techniques shows that OSMPs guarantee convergence, whereas traditional methods such as RNNs and NODEs (Zhi et al., 2024) often exhibit spurious attractors. We also validate our method extensively in real-world settings across various robotic platforms, including robotic manipulators and cobots, soft robots, and bioinspired hybrid soft-rigid underwater robots.

Moreover, we developed several techniques to enhance the capabilities of dynamic motion primitives. These include online reshaping of the velocity field to adjust convergence characteristics without retraining, synchronizing multiple OSMPs in phase via an error-based feedback term, and enabling the same motion policy to learn multiple distinct behaviors by conditioning the encoder on the task. With the addition of a tailored loss term, we can even achieve smooth interpolation between two motion behaviors.

III

Conclusion & Outlook



12

13

Conclusions & Future Work

This final chapter summarizes and examines the key findings of this dissertation. First, we revisit the various contributions of this thesis, evaluate their potential impact on the field, and acknowledge observed limitations. Next, we outline directions for future work. We begin with general recommendations for the field of soft robotics and then detail several specific avenues for future research, ranging from community-wide efforts to establish a safety metric to integrating SMPs with Vision-Language-Action (VLA) models, which aims to achieve high-performance, generalist motion control of (soft) robots while preserving robustness, compliance, and decision interpretability.

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q} = \mathcal{Q}$$

13.1 Conclusions

In the following, we will begin by summarizing the conclusions associated with each key contribution of this thesis. Next, as detailed in Sec. 13.1.7, we will synthesize overarching insights, with a particular focus on the core contribution.

13.1.1 Contribution I - Safety Metric

Addressing RQ 1, we introduced in Chapter 3 a quantitative safety metric for soft robots by modeling the collision dynamics between a soft robot and a human from first principles. Following the ISO standards for cobots (Standard, 2016), we use the maximum contact pressure/stress during a collision as a proxy for injury severity. To keep the evaluation computationally efficient, we derived a conservative upper bound that allows for a closed-form expression of the maximum contact pressure. Notably, the proposed SRISC accommodates collisions along the entire continuum of the soft robot and is built on validated dynamic models that incorporate its inertia, elasticity, and actuation. We then apply this metric to assess a soft robotic design by evaluating the maximum SRISC under its operating conditions.

We acknowledge that our safety metric has not yet been experimentally validated, which should be an emphasis for future work. Additionally, to ensure computational tractability, we made several simplifying assumptions—such as neglecting dissipation, using a linear spring contact model, modeling the human as a constrained point mass, and limiting the number of backbone segments—which result in a conservative injury severity estimate. This conservative bias might lead, for example when leveraging the metric for safety-aware control, to overly cautious robot behavior.

In Section 13.2, we give recommendations for future research and interesting applications for the proposed safety metric.

13.1.2 Contribution II - Leveraging Kinematic Models for Soft Robot Shape Sensing

Answering RQ 2, in Chapters 4 and 9, we demonstrated that kinematic models, such as PCC (Webster III and Jones, 2010), can be employed for shape sensing by solving a nonlinear optimization problem that aligns sensor measurements with the backbone shapes attainable by the chosen kinematic model. We proposed two approaches: one using cameras, where the robot perceives and tracks its environment combined with vSLAM algorithms, and another utilizing embedded magnets with magnetic sensors. Both methods were validated experimentally on a pneumatically-actuated soft segment.

Future work should evaluate the reliability and performance of the proposed method on multi-segment robots. In addition, since current proprioception techniques rarely incorporate dynamic model information when filtering state estimates, we recommend that future studies explore nonlinear observers (Shao et al., 2023) to enhance state estimation.

Connected to this contribution, we have observed promising recent work (Caroleo et al., 2025) that integrates commercial sensors—specifically, miniature TOF sensors—for soft robot proprioception by analyzing environmental motion within the soft body's local frame. This approach, akin to the vSLAM method presented in Chapter 4, could benefit from incorporating kinematic model knowledge to improve shape sensing.

13.1.3 Contribution III - Control with Advanced Physics-based Actuation Models for Soft Robots

Examining RQ 4, in Chapters 5 and 8, we derived physics-based actuation models for soft robots from first principles and demonstrated how to leverage these models for control—using methods like mapping into actuation coordinates (Pustina et al., 2024a) and backstepping (Khalil, 2002). For planar HSA robots, we further validated both the model and controller through extensive experiments.

However, the wide range of soft robot actuation modalities—from pneumatic and tendon-driven systems to dielectric elastomers, metamaterials (Zaidi et al., 2021), and shape memory alloys—poses a significant challenge for the community, as each actuation method requires its own tailored modeling approach and specialized controller - something we recommend future work to address (Copaci et al., 2020; Soleti et al., 2024, 2025).

13.1.4 Contribution IV - Integrating Physical Structure and Stability Guarantees into Learned Models

In this thesis, to tackle RQ 4, we proposed two methods to integrate physical structure and stability guarantees into learned soft robot models. By physical structure, we mean a formulation of the dynamics that explicitly defines potential and kinetic energy terms. Rather than modeling the entire dynamics with a black-box approach (e.g., a neural network), both methods leverage existing physics-based dynamic models by identifying their free parameters. This process can optionally include a coordinate transformation or encoding into a latent space to enhance expressiveness and reduce dimensionality.

The first method, presented in Chapter 10, introduces a data-driven strategy for deriving a PCS-based kinematic model. Our approach explicitly balances the trade-off between configuration dimensionality and backbone shape reconstruction accuracy. An Euler-Lagrange dynamic model is then derived, with its parameters identified in closed form. Stability for both open-loop and closed-loop systems can subsequently be analyzed using Lyapunov arguments (Khalil, 2002), similar to previous work on physics-based soft robotic models (Della Santina et al., 2023).

The second method, described in Chapter 11, focuses on learning the latent dynamics of soft robots through a network of nonlinearly coupled harmonic oscillators. This strategy provides a clear mechanical interpretation of the latent space, as each latent variable corresponds to the position of an oscillator. Moreover, the kinetic and potential energy terms inherent to this system enable us to prove its open-loop stability—specifically global asymptotic and input-to-state stability—using established Lyapunov techniques. A current limitation of the latent dynamics approach presented in Chapter 11 is that the input encoder-decoder—crucial for converting physical actuation into a forcing on the oscillators and vice versa—is not guaranteed to be a diffeomorphism. This may result in an incorrect mapping of the control input to the system's actuation. Future work could, therefore, explore bijective neural-network encoders, such as those based on normalizing flows (Kobyzev et al., 2020), to learn an analytically invertible mapping of the inputs into latent space.

Finally, we compare the two methods. A major advantage of the strain-model-based approach from Chapter 10 is its fully white-box nature, resulting in a fully interpretable, physics-based dynamical model with a strong inductive bias for soft robot dynamics. This bias improves data efficiency, reduces overfitting, and enhances extrapolation, but it may

struggle with dynamics such as hysteresis, nonlinear elasticity, deforming cross-sections, or significant environmental interactions. In contrast, the second approach, through its latent space encoding, is better equipped to learn a wider variety of dynamics, as demonstrated in the presented results. Where needed in the future, the network can be easily augmented with additional components (e.g., hysteresis or contact models, discussed in Sec. 13.2). The expressiveness added by the encoder sacrifices the direct interpretability of the latent variables as parameters of a kinematic model for the soft robot's backbone shape.

13.1.5 Contribution V - Exploiting Learned Models for Closed-Form Model-Based Control

This thesis, while exploring RQ 4, has shown that existing PID+energy shaping controllers (Della Santina et al., 2023; Kelly, 1995, 1998; Kelly and Carelli, 1996; Sciavicco and Siciliano, 2012) can be effectively combined with learned models that incorporate a physical structure. In particular, the regulation controller we examined employs an integral-saturated PID (P-satI-D) (Pustina et al., 2022a) as the feedback component alongside a potential shaping feedforward term. Here, the proportional term rejects disturbances, the saturated integral term compensates for modeling errors in a stable manner, and the derivative term adds extra damping as needed. Meanwhile, the feedforward term adjusts the potential energy of the learned dynamics so that its local or global minimum coincides with the desired state.

We applied this control strategy to both the identified PCS-based soft robot dynamics in Chapter 10 and the learned latent dynamics from the CON in Chapter 11. Simulation results indicate that the P-satI-D plus potential energy controller effectively steers the system toward a given setpoint. Notably, the characteristics of the learned potential field are quite interesting: for the ISS CON, the potential field is guaranteed to be convex, ensuring that the closed-loop latent space system—shaped by the feedforward term—is globally asymptotically stable, even if this might not hold for the actual soft robot. In contrast, the learned PCS-based dynamics can produce a non-convex potential field influenced by gravitational and elastic effects, which may result in only local asymptotic stability for the closed-loop system.

Several avenues for future research remain open: (1) experimental validation of the control strategy with learned models is yet to be performed; (2) while this thesis focused on setpoint regulation, extending the approach to trajectory tracking controllers would be both interesting and valuable; (3) we have so far considered only configurations where the learned model's dimensionality matches that of the actuation (i.e., the fully actuated case). In particular, for learned latent dynamics, exploring underactuated settings (Della Santina et al., 2023)—where the dynamics have a higher dimensionality than the available actuators—could enhance the model's expressiveness. A promising tool for addressing underactuation may be the mapping into actuation coordinates proposed by Pustina et al. (2024a), which we have already utilized for the physics-based HSA model in Chapter 6.

13.1.6 Contribution VI - Beyond Low-Level Control: Generating Compliant Motion Behaviors for Soft Robots

Finally, exploring RQ 5, in this thesis, we introduced two methods for achieving compliant motion in soft robots that allow guidance of low-level controllers with setpoints. In

Chapter 7, we developed a Brain-Machine Interface (BMI) that enables users to control soft robots through motor imagery. By binary classifying brain signals recorded with a wearable EEG device, this approach effectively and safely directs the robot's end-effector.

In Chapter 12, we presented a SMP-based method to parameterize periodic motion policies using a dynamical system that guarantees orbital stability. This technique allows for the learning of complex periodic motions from demonstrations without sacrificing stability or insight into the motion behavior, with the generated velocity commands—whether defined in task or configuration space—serving as references for low-level controllers. Our results show that this method produces more compliant and natural motions under environmental perturbations compared to traditional time-parametrized trajectory tracking.

Both approaches were experimentally validated on HSA and helicoid soft robots (Guan et al., 2023), respectively, illustrating that techniques originally developed for rigid manipulators can be adapted with minimal modifications for soft robots. However, despite some initial research (Goldman et al., 2014; Greer et al., 2020; Rao, 2024; Rao et al., 2024; Selvaggio et al., 2020), high-level motion planning and policy design tailored to soft robots remain underexplored. Unlike rigid manipulators that avoid unintended environmental contacts, soft robots should capitalize on contact along their entire bodies to fully leverage their embodied intelligence. This shift calls for new motion planning strategies that optimally exploit these interactions.

13.1.7 Core Contribution - Safe and Precise Soft Robots via Closed-form Control with Learned Models

In the following, we summarize our approach for achieving safe and precise motion in soft robots by leveraging control strategies based on learned models.

As discussed in Chapter 3, the inherent elasticity of both the structure and the surface of soft robots enhances their safety compared to traditional rigid manipulators. Nevertheless, further research is needed to experimentally validate the proposed safety metric and quantify the additional safety benefits relative to rigid robots.

Existing control methods struggle to deliver precise motion behavior without sacrificing safety. This is because (1) controllers based solely on physics-based models derived from first principles cannot fully capture the complex dynamics of soft robots—such as nonlinear elasticity and hysteresis—as well as their interactions with the environment, and (2) fully learned controllers (e.g., those based on RL) are sample inefficient, provide limited insight into their decision-making processes, and lack stability guarantees, potentially compromising safety.

Instead, this thesis, in its second part, proposes controlling soft robots with learned models, a strategy that combines the expressive power of modern ML techniques with enhanced data efficiency and improved transparency in decision-making. Although similar ideas have been explored in the literature on world models and reduced-order models, existing methods leveraging learned models for control through MPC, optimal control, or Model-Based Reinforcement Learning (MBRL) is computationally intensive. This computational burden restricts their control frequency and prevents full exploitation of soft robots' dynamic capabilities. In contrast, our approach embeds a physical structure into the learned model, enabling closed-form control and, therefore, significantly boosting computational efficiency. Specifically, we developed a model-based controller that inte-

grates an integral-saturated PID (P-satI-D) (Pustina et al., 2022a) with a potential-shaping feedforward term (Della Santina et al., 2023) for regulation. The key innovation was incorporating physical structure into the learned models, which allows us to compute the kinetic and potential energy of the system—something most existing dynamic model learning strategies, such as RNNs or NODEs, cannot do. In this thesis, we introduce two approaches (detailed in Chapter 10 and Chapter 11) that both leverage physics-based dynamic models. Our learning algorithm identifies the free parameters of these models, along with an optional coordinate transformation (e.g., an encoding into latent space). In both cases, we can express the system's kinetic and potential energy, allowing us to derive the potential shaping feedforward term in closed form. This framework also enables us to analyze the open- and closed-loop stability of the system using standard nonlinear systems theory tools, such as Lyapunov methods (Khalil, 2002).

There are, however, some limitations to this work. While we have been able to extensively experimentally validate Part I of this thesis, the model learning or the control strategies based on learned models have not been experimentally validated yet. We therefore recommend extensive experimental testing of both components in future work, preferably using various soft robot embodiments. Additionally, our current study has focused solely on setpoint regulation scenarios; future research should explore extending these strategies to trajectory tracking. In Chapter 11, we address control only for the case where the dimensionality of the latent dynamics matches that of the actuators (i.e., the fully actuated case). Future investigations might consider underactuated scenarios, where expanding the latent space could lead to more accurate dynamic modeling. Moreover, the current learned models do not account for environmental interactions—a crucial factor for soft robots intended to exploit their embodied intelligence through frequent contact with their surroundings. Finally, while the inherent softness of the robot enhances the safety of both open- and closed-loop systems, it would be worthwhile in future work to explicitly incorporate safety into the control strategy. One promising direction could be to use the safety metric proposed in Chapter 3 as a constraint within MPC or CBF (Ames et al., 2016) frameworks.

13.2 Future Work

First, we will introduce several overarching recommendations for the field moving into a future where soft robots are an essential contributor to our daily lives. Next, we outline several concrete avenues for future research building directly on this thesis.

13.2.1 General Recommendations

Historically, soft robotics research has primarily focused on design and actuation. Although the modeling and control of soft robots have garnered more attention in recent years, we find that cognitive intelligence for soft robots remains still underexplored. In particular, there is a notable lack of research on critical areas such as perception, high-level motion planning, task assignment, and reasoning—with tailored algorithms suited for soft robotic applications in human-centric environments or scenarios involving physical contact.

We believe one reason for this gap in high-level cognitive intelligence research is the scarcity of commercially available continuum soft robots compared to rigid manipulators,

cobots, and quadrupeds. This limitation makes it challenging for labs specializing in cognitive intelligence to venture into soft robotics, as developing a competitive, robust prototype requires years of expertise. We hypothesize that broader access to affordable continuum soft robots would catalyze rapid, exponential progress in the cognitive/computational intelligence of soft robots.

Furthermore, the emergence of more reliable, robust, and longer-lasting commercial soft robots would enable extensive field trials, moving beyond the breadboard and lab environments that have thus far dominated experimental validation. Demonstrating that soft robots outperform standard rigid manipulators or cobots in at least one measurable dimension is crucial. Such evidence is essential for sustaining and increasing investment in soft robotics research, fostering a vibrant field, and avoiding a potential "robotic winter" reminiscent of past challenges in Artificial Intelligence (AI) (Muthukrishnan et al., 2020).

In addition to field trials, we advocate for user studies that clearly establish soft robots as safer than cobots, along with defining explicit safety requirements—potentially tailored to various relevant applications, such as assisting the elderly with ADLs or collaborative manufacturing. More research is also needed to determine whether soft robots are perceived as safer and friendlier than traditional rigid robots. Essentially, further investigation into *social soft robots* is needed—mirroring the research already conducted on their rigid counterparts (Breazeal et al., 2016). Although initial findings suggest that soft robots may not be inherently perceived as more natural than rigid ones (Jørgensen et al., 2022), further investigations across diverse settings and designs are warranted (Pasquier et al., 2025).

In the following section, we outline several future research directions directly linked to the work presented in this thesis. These avenues focus on enhancing the physical intelligence of soft robots by improving their precision, dexterity, and capability to handle complex, sequential real-world tasks—especially in human-centric environments where safety is paramount. Ultimately, increasing the practical utility of soft robots by several orders of magnitude is necessary for soft robots to become an economically viable option, and this challenge should be a primary focus for the research community in the coming years.

13.2.2 Towards a Universally Accepted Safety Metric

For future work, we recommend experimentally validating the safety metric introduced in Chapter 3. This validation should involve a comprehensive experimental characterization of the collision properties of soft robots, covering various designs (e.g., silicon and helicoid soft robots) and different actuation methods (e.g., pneumatic, tendon-driven), and should verify whether the maximum contact force/pressure predicted by the metric corresponds with experimental measurements. Additionally, controlled user studies should determine appropriate thresholds for maximum force/pressure across different body areas, similar to the approach taken for collaborative robots (Muttray et al., 2014b). Such research could pave the way for establishing an internationally recognized standard (e.g., ISO) that certifies soft robots as sufficiently safe for specific applications. Moreover, we encourage exploring how the proposed safety metric can be integrated into safety-aware control strategies (Wong et al., 2025). Potential approaches might include leveraging MPC, CBFs (Ames et al., 2016; Wong et al., 2025), or developing conditions for safe operation (e.g., maximum velocity, torque, stiffness, reflected mass, etc.) through a safety map (Mansfeld et al., 2018).

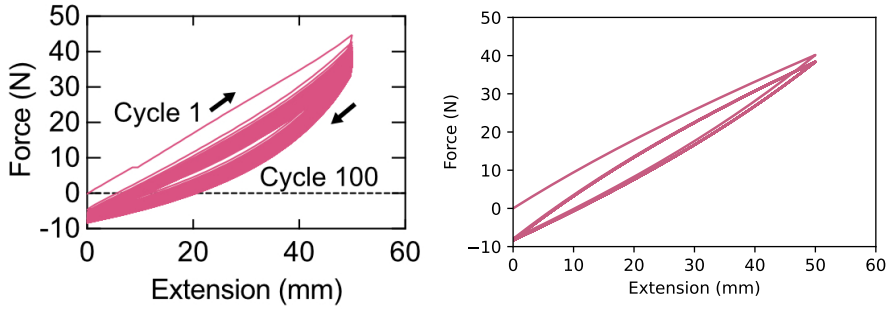
13.2.3 Towards Dexterous and Precise HSA Robots

In Chapter 6, we introduced several control strategies for planar HSA robots and demonstrated their effectiveness experimentally. Nonetheless, this control framework has several limitations: (a) it is confined to planar motions, (b) it applies only to single-segment HSA robots, (c) it is geared toward setpoint regulation rather than trajectory tracking, and (d) it does not explicitly account for the hysteresis inherent to HSA robots (Truby et al., 2021), necessitating a calibration of the rest strain before each experiment. In particular, future work should explore the twisting motion primitives illustrated in Fig. 5.2, which contribute to the unique behavior of HSA robots within soft robotics.

To address limitation (a), a kinematic parameterization that captures the three-dimensional motion of HSA robots is needed. Future research could build on the SPCS model introduced in Chapter 5—which describes the deformation of each HSA rod—by deriving closed-chain dynamics through the enforcement of Pfaffian constraints (Armanini et al., 2021; Tsounis et al., 2025) to eliminate drift in the redundant degrees of freedom.

Regarding limitation (b), recent work by Good et al. (2025) has shown that multi-segment HSA robots can be realized by actuating the distal segments with servo motors mounted at the robot base via Bendable Extendable Torque Resistant (BETR) shafts. This design reduces the inertia in motion compared to the more naive approach of placing motors at the proximal end of each segment. While extending existing models and controllers to accommodate multi-segment systems appears straightforward—as evidenced by work on standard continuum soft robots (Della Santina et al., 2020c)—new actuation models will likely be necessary to capture the unique properties of BETR shafts accurately.

Concerning limitation (c), the main challenge in developing trajectory-tracking controllers lies in the underactuated nature of the robot model. Although trajectory tracking controllers for fully actuated systems are well-established in the literature (Della Santina et al., 2020c), recent research by Soleti et al. (2025) on trajectory tracking for underactuated soft robots based on dielectric elastomers suggests that similar strategies could be adapted for HSA robots.



(a) HSA rod characterization by Truby et al. (2021) (b) HSA rod behavior simulated with a Bouc-Wen model

Figure 13.1: **Left:** Mechanical characterization of FPU HSA rods by Truby et al. (2021). **Right:** Preliminary simulation of a HSA rod based on the model derived in Chapter 5 augmented with a Bouc-Wen model to capture hysteretic behavior.

Finally, to overcome limitation (d), the hysteretic behavior of HSA robots could be modeled using the Bouc-Wen framework (Bouc, 1967; Wen, 1976) or related approaches. As shown in Fig. 13.1, preliminary results show that a Bouc-Wen model with $\alpha = 0.1$, $\beta = 0.25$, and $\gamma = -0.75$ representing strain hardening can capture the hysteretic behavior of HSA rods, as experimentally characterized by Truby et al. (2021) relatively well. A nonlinear observer (Shao et al., 2023) could then be developed to estimate the hidden hysteretic displacement in real-time based on the current state. Ultimately, incorporating an optimal control strategy—such as MPC—would allow the system to consider its hysteretic state when determining the next control action.

13.2.4 Towards Expressive Learned Models by Integrating Advanced Physical Priors

The CON model introduced in Chapter 11 is currently formulated to learn latent dynamics only for physical systems that display continuous dynamics, global asymptotic stability (i.e., they possess a unique, isolated fixed point attractor), and inherent dissipation. However, in practical applications, soft robots—especially when interacting with their environment—may not meet these criteria. For instance, depending on the gravitational field, a soft robot (much like a pendulum) might exhibit multiple equilibria; contact interactions could introduce hybrid dynamics, and time-dependent phenomena such as hysteresis or material degradation may occur. To address these issues, future work could (a) relax some of the network parameter constraints to accommodate non-convex potential energy landscapes and multistability and (b) enhance the expressiveness of the learned (latent) dynamics by incorporating additional components into the network, such as models for contact, Bouc-Wen (Bouc, 1967; Wen, 1976) hysteresis, or specialized actuation models.

13.2.5 Towards Efficient Co-Design of Computational and Embodied Intelligence

In this thesis, we assume a given soft robot morphology (e.g., an HSA or piston-driven pneumatic actuation) and develop the computational intelligence required for precise control of continuum soft robots. This involves creating accurate reduced-order models, designing efficient low-level controllers, and implementing compliant high-level motion policies. For instance, for the existing HSA robot design presented by Chin et al. (2018); Lipton et al. (2018); Truby et al. (2021), we developed the corresponding models (Chapter 5), controllers (Chapter 6), and BMIs (Chapter 7). Similarly, in Chapter 8, we developed a model capturing the coupled dynamics between soft robot and actuator for pneumatic piston-driven soft robots (Marchese et al., 2014; Marchese and Rus, 2016), which we then exploited for backstepping-based control. Finally, Chapter 9 demonstrates the need for co-optimization of sensor (and magnet) placement alongside the kinematic model and the proprioception algorithm.

These examples highlight how the current soft robotic development process typically follows a one-way street: a mechatronics engineer devises an innovative design (including actuator and sensor placement), then a modeling expert formulates a kinematic and dynamical model, followed by a perception specialist developing proprioception and exteroception methods, a control theorist implementing a low-level controller, and finally a motion planning specialist designing a high-level motion policy. This sequential approach

often results in designs that are very challenging to model and control.

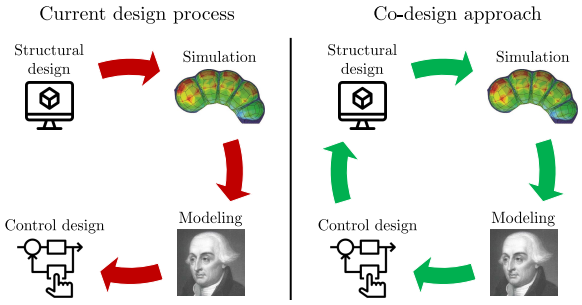


Figure 13.2: Traditional soft robot design follows a sequential process: first, the structure is created in Computer-aided Design (CAD); next, the robot's behavior is simulated (for example, using FEM); then, reduced-order kinematic and dynamic models are developed; and finally, the controller is designed based on these models. Moving forward, research should focus on a co-design approach that simultaneously optimizes the robot's body (e.g., structure, actuation, sensors) and its brain (modeling, perception, control).

Looking ahead, our goal is to co-design the body and the brain of the soft robot, fostering an optimal integration of embodied (Mengaldo et al., 2022) and computational intelligence, as illustrated in Fig.13.2. Although some initial work on soft robot co-design exists (Bhatia et al., 2021; van Diepen and Shea, 2022; Spielberg et al., 2019; Wang et al., 2024, 2022, 2023b), these methods are often too simplistic, discretizing the continuum into passive, actuated, and sensorized voxels or particles. As a result, the derived designs are rarely practical (Legrand et al., 2023; Wang et al., 2024). Additionally, most studies train a learning-based controller (e.g., using RL) from scratch for every design iteration, which creates a significant computational bottleneck in the co-design process (Bhatia et al., 2021; Wang et al., 2022, 2023b). We conclude that the high computational demands of co-design currently restrict its practical usefulness, limiting the scope of the design space that can be explored within available computing budgets.

Future research should develop and integrate computationally efficient metrics for manufacturability, observability, controllability, and safety to provide early feedback on the design to the optimizer—avoiding the need for costly RL training and FEM simulations for every design. While rigid manipulators benefit from fixed joints that simplify modeling, the deformability of soft robots complicates the creation of low-dimensional models (Armanini et al., 2023). Traditional trial-and-error methods for selecting backbone models (e.g., PCC, PCS, GVS) depend heavily on a robot's topology, actuation, and dynamics, making it essential to co-design reduced-order models with the morphology. Moreover, using model-based controllers rather than training a new RL policy from scratch for each design could further accelerate the process. We discuss our perspective on soft robot co-design in more detail in Appendix A.

13.2.6 Towards Generalist Policies for Robust, Compliant, and Interpretable Motion Behavior by Integrating Stable Motion Primitives with Vision-Language-Action Models

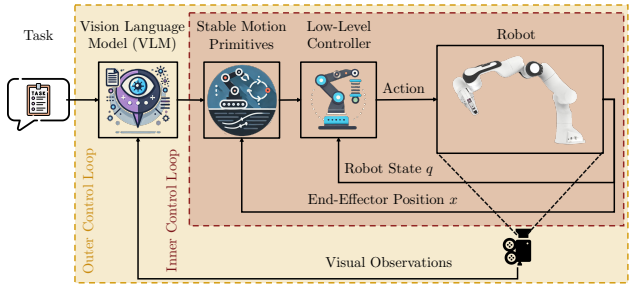
In Chapter 12, we introduced a motion policy based on SMP for learning periodic motions from demonstrations. By parameterizing the policy with an interpretable dynamical system, we can ensure both stability and compliant behavior. However, several limitations currently restrict its practicality: (a) most DMP/SMP methods are confined to a single attractor type (e.g., fixed-point, limit cycle, multistability), (b) a single DMP cannot readily capture intersecting trajectories, (c) information about the task setting and environment is not incorporated into decision-making, and (d) these methods typically learn only one or a few trajectories at a time.

To address these issues, future work might integrate SMPs with modern VLM/VLA models. As shown in Fig.13.3, a VLM could generate sequences of motion steps that are then tracked by an SMP combined with a low-level controller (e.g., an impedance controller). This setup forms a cascaded control scheme, illustrated in Fig.13.3(a): a low-level controller with configuration-space state feedback runs at high frequencies (around 1000 Hz), the SMP supplies end-effector motion references at approximately 100 Hz, and the VLM operates at about 10 Hz, processing visual feedback to issue motion steps to the SMP.

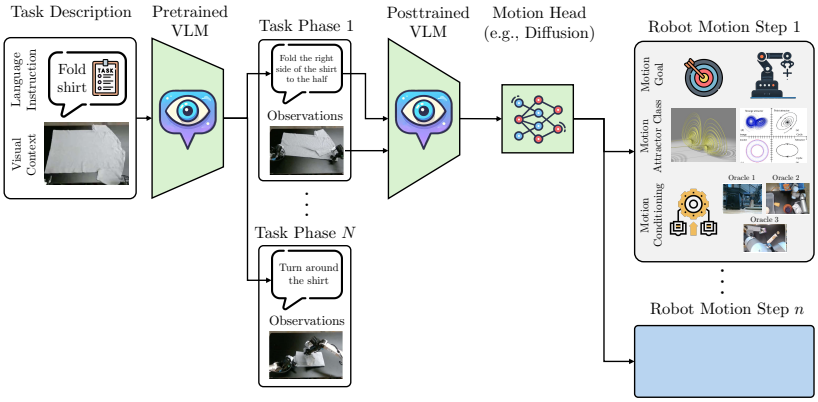
More specifically, as depicted in Fig.13.3(b), we envision a hierarchical approach (Haresh et al., 2024) that breaks down a high-level task—such as folding laundry—into task phases (e.g., folding the right side of a shirt halfway) and individual motion steps, utilizing both pretrained and posttrained VLMs[1]. The connection between the VLMs and the SMP is facilitated by a motion head network—potentially based on diffusion policies (Chi et al., 2023)—that predicts the necessary robot step, including a motion goal or waypoints, the corresponding SMP attractor type (e.g., fixed-point, limit cycle, multistability), and the conditioning of the SMP encoder on a motion demonstration from training. Figure 13.3(c) illustrates how this conditioning mechanism enhances the expressiveness of SMPs, enabling a single policy to represent a wide range of motions. During inference, both the motion goal x^{d} and the current end-effector position x are passed through a bijective encoder (Rana et al., 2020), which maps them into latent space coordinates z^{d} and z, respectively. The latent dynamics—governed by the latent goal and the selected attractor type—yield a latent velocity $\dot{z}$, which is then projected back into the original space using the encoder's inverse Jacobian. A low-level controller subsequently tracks this velocity reference by computing the required actuation commands.

This strategy has the potential to enhance the expressiveness and generalization of SMP-based policies by combining them with state-of-the-art models while still preserving the interpretability and stability guarantees inherent to SMPs. These features notably distinguish our approach from current VLA policies (Black et al., 2024; Kim et al., 2024), which generally do not provide insight into the action generation process or offer stability guarantees.

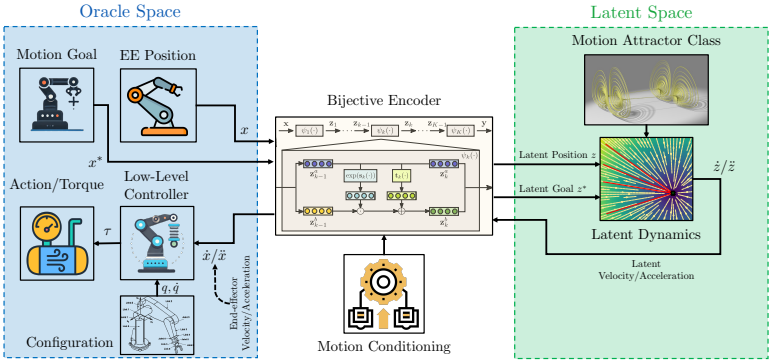
[1] Here, *pretrained* refers to a VLM trained on internet-scale data and large cross-embodiment datasets (Kim et al., 2024; O'Neill et al., 2024), while *posttrained* indicates a version fine-tuned on smaller, high-quality datasets specific to the robot embodiment and task (Black et al., 2024).



(a) Overview: Integration of Stable Motion Primitives (SMPs) with Vision-Language-Action (VLAs) Models



(b) Hierarchical Planning with VLMs

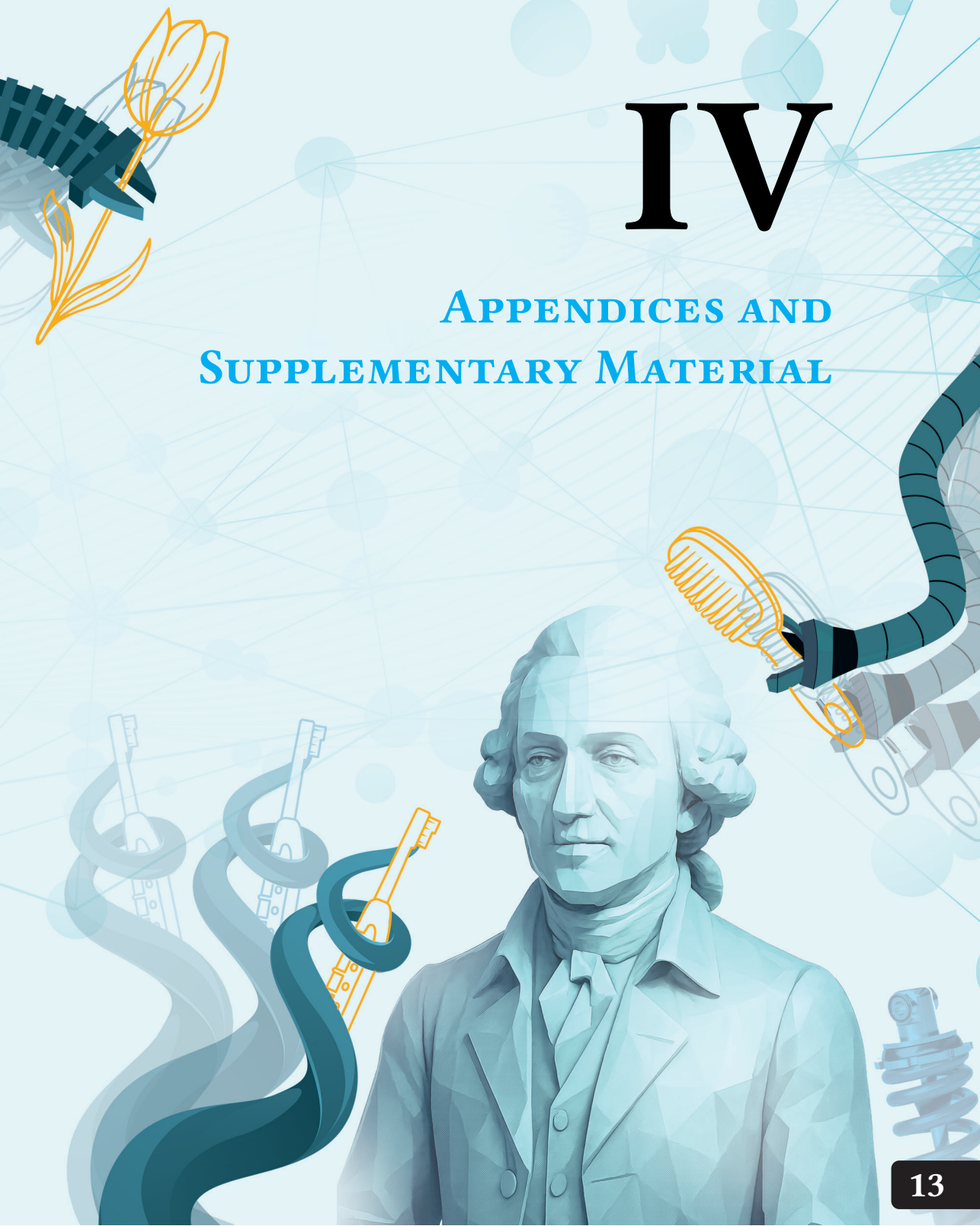


(c) Expressive SMPs

Figure 13.3: Concept for the Integrating SMPs with VLAs models: The VLM, composed of a pretrained and posttrained part, receives, possibly multimodal, task instructions, and its motion head translates them into robot motion steps instead of directly outputting actions. These robot motion steps, consisting of, for example, a motion goal, the motion attractor class, and conditioning on the demonstration/oracle, then shape the behavior of the motion policy parametrized by a dynamical system (i.e., a SMP).

IV

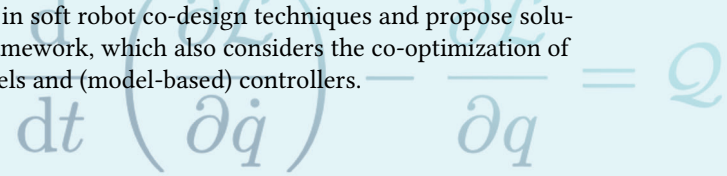
Appendices and Supplementary Material



A

Safe yet Effective Soft Robots via Holistic Co-Design

Foreword. In this thesis, we employed (so far) a traditional sequential design process: the soft robot's morphology—including its structural shape, actuation, and sensor placement—was predefined and assumed fixed. We then derived or learned reduced-order kinematic and control-oriented dynamic models, leveraging this model knowledge within controllers whose parameters (e.g., gains) were subsequently tuned. A notable example of this process is the design of HSA robots, extensively covered in this work. First, their design and actuation were developed at the Distributed Robotics Laboratory (DRL) at MIT (Chin et al., 2018; Lipton et al., 2018; Truby et al., 2021). Subsequently, we developed in this thesis modeling and control approaches for such HSA robots. In Chapter 5, we derived kinematic and dynamic models from first principles—relying heavily on our intuition and prior experience for the kinematic model—and determined their parameters via system identification. We then devised model-based controllers in Chapter 6 and experimentally tuned their gains, culminating in the development of a BMI interface in Chapter 7 to guide the low-level controller using EEG-measured brain signals. However, this sequential process prevents insights gained during later stages, such as closed-loop control experiments, from influencing the morphological design. This is a common challenge within the soft robotics community, where many designs prove difficult to model with reduced-order approaches and control accurately. In this appendix, we instead consider the co-design of the body (morphology) and the brain (control and perception algorithms) of soft robots, incorporating structured feedback mechanisms to iteratively optimize the entire design. Specifically, we review current gaps in soft robot co-design techniques and propose solutions through a holistic co-design framework, which also considers the co-optimization of control-oriented reduced-order models and (model-based) controllers.



A

Abstract. *Soft robots promise inherent safety via their material compliance for seamless interactions with humans or delicate environments. Yet, their development is challenging because it requires integrating materials, geometry, actuation, and autonomy into complex mechatronic systems. Despite progress, the field struggles to balance task-specific performance with broader factors like durability and manufacturability—a difficulty that we find is compounded by traditional sequential design processes with their lack of feedback loops. In this perspective, we review emerging co-design approaches that simultaneously optimize the body and brain, enabling the discovery of unconventional designs highly tailored to the given tasks. We then identify three key shortcomings that limit the broader adoption of such co-design methods within the soft robotics domain. First, many rely on simulation-based evaluations focusing on a single metric, while real-world designs must satisfy diverse criteria. Second, current methods emphasize computational modeling without ensuring feasible realization, risking sim-to-real performance gaps. Third, high computational demands limit the exploration of the complete design space. Finally, we propose a holistic co-design framework that addresses these challenges by incorporating a broader range of design values, integrating real-world prototyping to refine evaluations, and boosting efficiency through surrogate metrics and model-based control strategies. This holistic framework, by simultaneously optimizing functionality, durability, and manufacturability, has the potential to enhance reliability and foster broader acceptance of soft robotics, transforming human-robot interactions.*

This chapter is partly based on 🗎 ***M. Stölzle****, *N. Pagliarani**, *F. Stella, J. Hughes, C. Laschi, D. Rus, M. Cianchetti, C. Della Santina, and G. Zardini (2025). Soft yet Effective Robots via Holistic Co-Design. In Nature Machine Intelligence,* **Under Review** (Stölzle et al., 2025a).

*M.S. and N.P. contributed equally to this work. M.S. conceived and led the project. All authors contributed ideas to this perspective. M.S., N.P., and F.S. collaborated on the figures and graphics. M.S. wrote the majority, and N.P. and F.S. wrote parts of the manuscript. All authors revised the manuscript. G.Z., C.D.S., M.C., and J.H. supervised the project and provided funding.

A.1 Introduction

Robots have long been used in industrial tasks requiring precision and strength, such as assembly and manufacturing (Todd, 1996). However, societal challenges now call for robots designed for human-centered environments like homes and public spaces (Chibani et al., 2013; Nahavandi, 2019; Royakkers and Van Est, 2015). To operate safely in dynamic, unpredictable settings—and in line with Asimov's First Law to avoid harming humans (Asimov, 1941; Villani et al., 2018)—robots should incorporate inherent physical compliance. Modern systems use advanced controls like safety filters, compliant control, real-time collision detection, and predefined safety zones (Zhao et al., 2024b), relying on sophisticated sensors and algorithms to preempt hazards (Fragapane et al., 2021). Collaborative robots (cobots) are designed with decoupled actuators, reduced inertia, and compliant controllers for safer human interaction, yet their rigid components can still pose risks (Haddadin et al., 2009). Although collision detection can mitigate danger by slowing or stopping robots near humans (Haddadin, 2013), it does not completely eliminate injury risk.

Soft robotics offers a transformative approach by embedding safety directly into a robot's materials and structure, reducing reliance on complex computational algorithms (Laschi et al., 2016; Rus and Tolley, 2015). This inherent compliance enables safe human interaction and operation in sensitive settings such as personal assistance, caregiving, and handling delicate items (Abidi and Cianchetti, 2017; Pasquier et al., 2025). However, soft robot development is inherently complex, requiring the seamless integration of materials, geometry, actuation, sensing, compliant continuum dynamics, perception, and control systems. It is notably difficult to predict how morphological changes affect closed-loop motion, as the design space is much larger than for rigid robots, and the capabilities of the autonomy stack and control are inherently limited by the body. For some morphologies with very complex deformations, designing effective proprioception and control systems is even intractable. Consequently, we observe that traditional sequential design processes (van Boeijen et al., 2020)—which move from conceptual design through mechatronic design to the development of autonomy and control systems—struggle with this complexity, leading to robots that exhibit imprecise, oscillatory motions, limited payload capacity, and inadequate force output (Cianchetti et al., 2013; Hawkes et al., 2017; Iida and Laschi, 2011; Majidi, 2014; Mazzolai et al., 2022).

Related Work and Limitations Co-design strategies have proven effective in addressing multi-objective problems and accommodating the compositional, hierarchical nature of complex systems (Zardini, 2023). In this perspective, we review recent studies that have introduced algorithms for simultaneous optimization of soft robot morphology and control systems (Bhatia et al., 2021; Chen and Wang, 2020; Junge and Hughes, 2022; Legrand et al., 2023; Medvet et al., 2021; Navez, 2024; Spielberg et al., 2021, 2019; Van Diepen and Shea, 2018; Wang et al., 2024, 2022, 2023b). However, we identify several shortcomings that limit their broader application. First, computationally expensive optimization cycles hinder exploration of the full design space (Chen and Wang, 2020), due to high-dimensional discretizations (Legrand et al., 2023; Medvet et al., 2021, 2022; Spielberg et al., 2019; Wang et al., 2023a, 2024, 2022, 2023b), inefficient algorithms (e.g., evolutionary ones) (Bhatia et al., 2021; Chen and Wang, 2020; Hiller and Lipson, 2012; Medvet et al., 2021, 2022; Rieffel et al., 2014), costly RL-based control training (Bhatia et al., 2021; Wang et al.,

2023a, 2022, 2023b), and reliance on intensive simulations for fitness evaluation (Legrand et al., 2023; Medvet et al., 2021, 2022; Spielberg et al., 2019; Wang et al., 2023a, 2024, 2022, 2023b). Second, a narrow focus on easily computable evaluation metrics (e.g., locomotion speed (Wang et al., 2024), workspace (Guan et al., 2023)) often neglects other vital design values such as manufacturability (Kim et al., 2025), safety, cost, ecological impact, usability, and regulatory requirements (Junge and Hughes, 2022). Third, the actual realization of the design is rarely (Junge and Hughes, 2022) factored into the co-design optimization process, which limits the incorporation of insights gained from fabrication, prototyping, and lab/field testing. Similarly, the uncertainty associated with simulation-derived evaluation metrics is seldom considered (Chen and Wang, 2020), leading to designs that perform well in simulations but underperform in real-world scenarios. Finally, current co-design methods generally fail to incorporate diverse stakeholder input or account for (all) end-user requirements.

We recognize that parallel yet disconnected efforts exist to address these challenges. For instance, efficient category theory–based algorithms have recently been developed for the co-design of self-driving vehicles, considering diverse design values such as cost, compute requirements, vehicle mass, and power (Milojevic et al., 2025; Zardini et al., 2021b, 2022b). In ML, there is a strong emphasis on generative models for design generation (Vahdat et al., 2022), with recent applications in soft robot co-design (Song et al., 2024; Wang et al., 2024), enabling optimization in a reduced-order space while recovering the full design description. Furthermore, literature in ML and soft robotics explores how to learn reduced-order dynamic models for efficient model-based controller derivation (Alkayas et al., 2025; Alora et al., 2023a; Della Santina et al., 2023; Hewing et al., 2020; Stölzle and Della Santina, 2024) as an alternative to sample-inefficient RL. Additionally, concepts from nonlinear systems and robotics—such as observability (Griffith and Kumar, 1971), controllability (Zheng et al., 2019), and safety (Haddadin, 2013; Standard, 2016)—offer computationally inexpensive metrics to assess design fitness. Finally, a robust body of work in Bayesian optimization (Garnett, 2023; Hernández-Lobato et al., 2014), RL (Sutton et al., 1998), and co-design (Furter et al., 2025; Huang et al., 2025) has explored incorporating uncertainty in evaluation metrics during optimization and using exploration (i.e., realization) to reduce that uncertainty. The goal of this perspective is to synthesize these related yet isolated research directions into a holistic co-design framework that addresses the shortcomings of existing approaches.

Proposed Framework This perspective outlines a holistic co-design framework that integrates design components, stakeholder values, design processes, and optimization strategies, addressing key limitations of prior soft robotic co-design approaches through five core advances. First, the framework broadens the range of considered objectives and constraints to include safety, fabrication and operational costs, environmental impact, and regulatory compliance. Second, we introduce enhancements to computational co-design, boosting computational efficiency and allowing for global optimization over the design space. Specifically, this is achieved by (i) sampling from reduced-order design spaces decoded into full morphologies (Wang et al., 2024); (ii) co-optimizing reduced-order dynamical models, both physics-based (Armanini et al., 2023) and learned (Alkayas et al., 2025; Liu et al., 2024a; Navez et al., 2025; Stölzle and Della Santina, 2024; Valadas et al., 2025),

to capture task-relevant deformations with minimal complexity; (iii) using fast-to-compute surrogate metrics (e.g., controllability or observability) to guide the optimizer away from poor designs early in the process; and (iv) replacing costly RL training with efficient model-based control methodologies grounded in the same reduced-order models (Della Santina et al., 2023; Stölzle and Della Santina, 2024). Third, the framework incorporates purposeful physical prototyping to reduce uncertainty in computational evaluations. By treating co-design probabilistically, it accounts for uncertainties, such as the sim-to-real gap (Dubied et al., 2022), and uses high-fidelity simulation and prototyping across varying Technology Readiness Levels (TRLs) (Manning) to refine evaluation metric estimates. This enables formal trade-offs between computational "refinement" and physical "realization", taming the sim-to-real gap discrepancies in performances during the development cycle. Fourth, the integration of structured stakeholder engagement ensures that diverse values and requirements are reflected throughout the design process. Finally, the framework supports reproducibility (Baines et al., 2024) by maintaining an auditable design trail, which is critical for the deployment of soft robots in real-world contexts.

As illustrated in Fig. A.1, this co-design paradigm holds promise for enabling soft robots to take on vital roles in caregiving, education, and other human-centered domains. By ensuring both performance and safety, it can support societal well-being and increase public trust in robotic technologies.

A

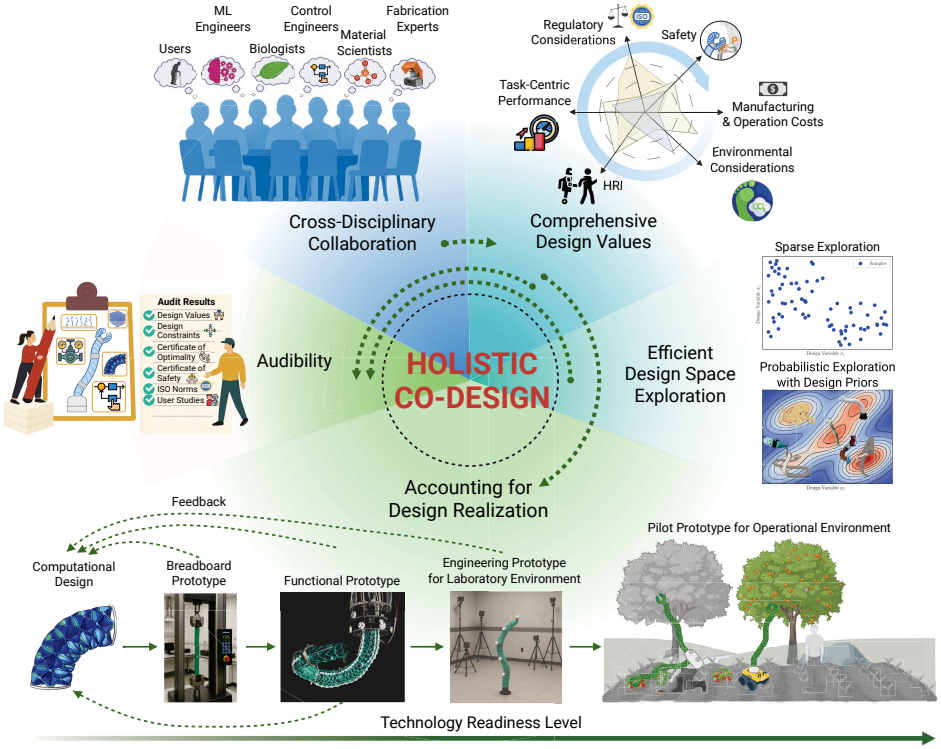


Figure A.1: **Holistic Co-Design of Soft Robots.** The five pillars of holistic co-design are (1) Incorporating comprehensive design requirements and values directly into multi-objective optimization; (2) Efficiently exploring the full design space by melding design priors (e.g., biological inspirations, existing solutions) with computationally efficient co-design routines; (3) Explicitly accounting for design realization (e.g., prototyping and testing) via a probabilistic treatment of evaluation metrics and by formalizing the refinement-vs-realization trade-off—enabling targeted prototyping to reduce metric uncertainty and narrow the sim-to-real gap; (4) Fostering cross-disciplinary collaboration and involving all relevant stakeholders in defining design values and providing iterative feedback; (5) Ensuring auditability to preserve design knowledge and guarantee reproducibility. These pillars are inherently interconnected (see green arrows)—for example, stakeholders and the design team co-define design requirements, certain design values (e.g., HRI) demand evaluation in the real world (i.e., enabled by realization), and both the exploration of the design space and the fulfillment of design requirements remain fully auditable.

A.2 The Past and Present Soft Robot Design Process

In this section, we review established design processes in the field of soft robotics. The discussion is organized into two parts: first, we examine the fundamental design cycle typically employed in soft robotics, covering the progression from initial task specifications to experimental prototyping and final design approval. Second, we explore recent work on the co-design of soft robots that showcases some initial ideas on how we could jointly design the body and brain.

Traditional Design Process of Soft Robots

The basic design cycle typically described in the literature involves several key steps (van Boeijen et al., 2020; Roozenburg and Eekels, 1995): (1) The cycle begins with defining criteria, where designers explicitly specify the values, requirements, and functions that the design must satisfy; (2) Next is the provisional design stage, in which solution ideas, concepts, and implementations are synthesized; (3) Following this, simulations can be used to anticipate the potential properties and performance of the design. These predictions are validated in step (4) by manufacturing prototypes at various TRLs (Manning); (5) Finally, if the prototypes meet the established criteria, the design is approved.

This traditional design process is also widely applied within soft robotics. As illustrated in Fig. A.2 (Left), it begins with task specifications and proceeds to a preliminary conceptual design, followed by detailed mechatronic development. The open-loop behavior of the resulting design is verified through simulation, fabrication, and testing on breadboard prototypes. Typically, control design and reduced-order modeling occur only after validating a viable soft robotic prototype, enabling verification of closed-loop behavior within laboratory settings. Furthermore, steps (2)-(4) often involve iterative cycles of diverging and converging phases (Feldhusen et al., 2013). During the *diverging* phase, new design ideas, concepts, and implementations are generated. Conversely, the *converging* phase evaluates these concepts through simulations or prototype tests to identify, for example, via selection matrices (Ulrich and Eppinger, 2016), promising designs worthy of further development.

However, the traditional design cycle has several notable deficiencies: (i) Although multiple iterations can theoretically "close the loop" of the design cycle, practical constraints—such as costs and complexity—often render iterative processes prohibitively expensive. Additionally, no formal or automated method typically exists to systematically incorporate insights from evaluated performance metrics back into improving the original design. (ii) Similarly, divergence and convergence cycles frequently lack effective feedback mechanisms, preventing valuable knowledge acquired during detailed design or high-fidelity prototyping from informing the generation of new design solutions. Indeed, for example, Suh's first axiom on design theory for systems argues that cycles should be avoided and functional requirements should be orthogonal to each other (Suh, 1998). However, numerous practical examples in product development have shown that iterative approaches can substantially enhance product quality and performance and decrease cost over time. Therefore, a design process for soft robots should embed such feedback mechanisms from the beginning. (iii) Constraints and metrics critical in later stages, such as manufacturability and compliance with regulatory standards, such as International Organization for Standardization (ISO) norms, are often inadequately considered during initial

stages. (iv) The sequential workflow between mechatronic design, perception, modeling, and control teams creates informational silos. For instance, modeling engineers typically transfer completed models to control engineers without sufficient feedback channels, thus restricting iterative enhancements. (v) A significant drawback of conventional design processes is the potential loss of "design history." Critical insights, rationale for decisions, and trade-offs are frequently undocumented, particularly when team members transition roles or leave the project. Consequently, revisiting earlier stages or building on previous knowledge becomes increasingly challenging. For example, a research team seeking to modify an earlier actuator design after identifying field performance issues may find the original design rationale inaccessible.

Co-Design of Soft Robots

Given the complexity of soft robots and the limitations of traditional design processes—especially the absence of structured and effective feedback cycles and the isolated design of components—the community has recently begun exploring how co-design algorithms could support soft robot development (Bhatia et al., 2021; Cianchetti, 2021; van Diepen and Shea, 2022; Junge and Hughes, 2022; Spielberg et al., 2019; Wang et al., 2024, 2022, 2023b). The term "*co*-design" highlights several distinctions from traditional design approaches, including a *co*mpositional and hierarchical nature - i.e., designing all system components, such as body and brain, together (Junge and Hughes, 2022); a *co*llaborative approach involving all teams and stakeholders, and a *co*ntinuous process of design improvement (Zardini, 2023). Co-design strategies have proven highly effective at solving complex, possibly multi-objective, optimization problems with a well-defined design space, cost function, and equality/inequality constraints. Notable examples stem from the fields of chemistry (Nørskov et al., 2009; Vaissier Welborn and Head-Gordon, 2018), construction engineering (Knippers et al., 2021), mobility systems (Zardini et al., 2022a, 2020), autonomous vehicles (Zardini, 2023; Zardini et al., 2021b), articulated robotics (Ha et al., 2018; Zhao et al., 2020).

Initial attempts have applied co-design methods to soft robots by simultaneously optimizing the body/morphology (i.e., structural shape, actuation, and sensing) and the brain (e.g., controller) (Bhatia et al., 2021; Cianchetti, 2021; van Diepen and Shea, 2022; Junge and Hughes, 2022; Navez, 2024; Spielberg et al., 2019; Wang et al., 2024, 2022, 2023b). Drawing on extensive research in mechanical and morphological design—particularly geometric design—these approaches can be classified (Chen and Wang, 2020) into three categories: (a) size optimization, which focuses on regular shapes defined by (hand-picked) geometric parameters (e.g., radii, segment lengths, pneumatic chamber dimensions) (Calisti et al., 2011; Dämmer et al., 2018; Guan et al., 2023; Junge and Hughes, 2022; Navez et al., 2024; Pagliarani et al., 2024; Polygerinos et al., 2015; Wang et al., 2018b); (b) shape optimization, which incrementally modifies the parametric surfaces of an initial design while preserving its connectivity or topology (Siéfert et al., 2019); and (c) full topology optimization, which rethinks the structure entirely by creating, repurposing, or removing elements from the topology (Caasenbrood et al., 2020; Jewett and Carstensen, 2019; Legrand et al., 2023; Pinskier et al., 2024; Sigmund and Maute, 2013; Spielberg et al., 2019; Wang et al., 2023a, 2024, 2022, 2023b; Zhang et al., 2018).

However, the existing co-design approaches exhibit several shortcomings: (i) The current topology optimization approaches often oversimplify the design process by discretizing the morphology spatially into passive, actuated, or sensorized voxels or parti-

cles (Legrand et al., 2023; Medvet et al., 2021, 2022; Spielberg et al., 2019; Wang et al., 2023a, 2024, 2022, 2023b), resulting in designs that rarely translate effectively to practical applications (Legrand et al., 2023; Wang et al., 2024). Additionally, (ii) most existing methods train learning-based controllers (e.g., using RL) from scratch for each iteration, creating a significant computational bottleneck in the overall co-design cycle (Bhatia et al., 2021; Wang et al., 2022, 2023b). Recent progress in differentiable physics-based simulation (Coevoet et al., 2017; Fang et al., 2020; Hu et al., 2019) suggests that integrating real-time gradient-based optimization of neural-network parametrized controllers could significantly reduce the computational overhead associated with iterative design-control loops, improving sample efficiency and convergence speed (Bächer et al., 2021; Spielberg et al., 2019; Wang et al., 2024). However, these methods assume smooth, end-to-end differentiability—which can fail under contact or discontinuities—and can be prone to local minima. Relatedly, (iii) these approaches typically require completing an entire cycle—from design generation, control policy training, and high-fidelity simulation through task-specific performance evaluation—before updating the design. This sequence can be highly computationally demanding and results in unnecessary resource use for designs that could have been earlier identified as suboptimal through intermediate metrics. Finally, (iv) most studies predominantly focus on computationally implementable aspects of the design process (i.e., *computational co-design*), neglecting critical downstream considerations (Junge and Hughes, 2022) such as breadboard testing, validation with prototypes, or regulatory compliance. Although topology optimization and multi-material design techniques (Chen and Wang, 2020) are gaining traction, their integration with fabrication constraints remains an open challenge. Considering manufacturability from the early stages of co-design—particularly in the context of additive manufacturing- could enhance the feasibility of computationally optimized soft robots (Navez, 2024). However, we conclude that the high computational demands of co-design caused by computationally expensive simulations and inefficient optimization routines currently restrict its practical usefulness, limiting the scope of the design space that can be explored within available computing budgets (Chen and Wang, 2020).

A.3 The Future: A Framework for Holistic Co-Design of Soft Robots

Holistic co-design marks a significant shift in soft robotics by adopting an integrated, application-driven strategy that emphasizes the seamless interaction between a robot's physical morphology, its control system, and user engagement. Unlike traditional design cycles—which remain prevalent—it provides a structured framework that concurrently develops all system components, refining the design iteratively through effective feedback loops. Moreover, this approach distinguishes itself from other co-design methods by embracing a broader set of design values, advocating for cross-disciplinary collaboration and active participation from diverse stakeholders, introducing techniques to enhance computational efficiency, and proposing a probabilistic view that acknowledges uncertainties in evaluation metrics while outlining strategies to mitigate them as the design evolves.

In detail, the proposed holistic co-design framework introduces several key innovations. First, it optimizes a range of design attributes simultaneously—such as task comple-

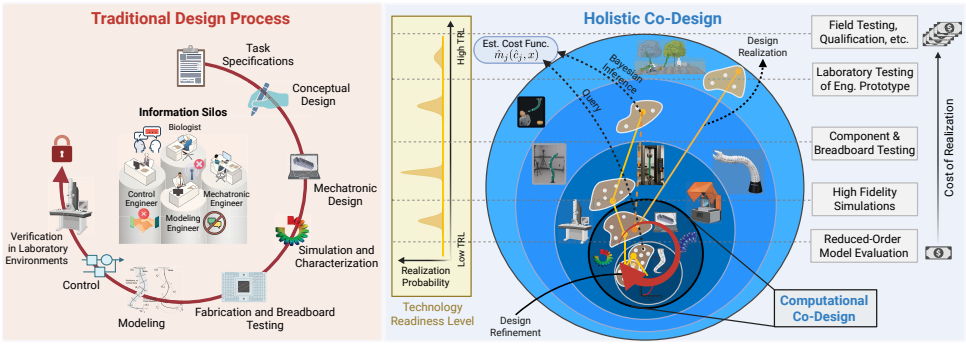


Figure A.2: *Left:* A traditional sequential design cycle (van Boeijen et al., 2020) applied to soft robots. This sequential workflow lacks iterative feedback loops, leading to information silos and preventing regular, bidirectional sharing of insights and data across the development team and the relevant stakeholders. *Right:* A holistic co-design framework that complements computational refinement with prototyping across various levels of design readiness - visualized as "onion" layers in this graphic. Solid yellow lines denote the prototyping pathway, while dashed orange arrows illustrate how empirical findings continuously inform and update the estimates $\hat{c}_j$ for the real-world design values c_j.

tion time, safety, material and fabrication costs, energy consumption, and environmental impact—instead of focusing on a single metric like locomotion speed, as most current approaches do. Second, it significantly boosts the efficiency of computational co-design by (i) establishing a reduced-order design space that avoids the complexity of directly optimizing a high-dimensional spatial morphology (as seen in voxel-based methods); (ii) developing a control-oriented reduced-order model within the co-design process; (iii) employing computationally inexpensive auxiliary metrics, such as controllability, observability (leveraging the reduced-order model), or manufacturability, which provide early feedback to the optimizer instead of relying solely on the outcome of resource-intensive closed-loop simulations; and (iv) efficient derivation/learning of controllers based on the reduced-order model (i.e., model-based control). Furthermore, while existing methods typically address only the fully computational aspects of design and evaluation, our framework explicitly accounts for the inherent uncertainties in these metrics—uncertainties that can only be properly identified through high-fidelity simulations or, ideally, extensive testing with physical prototypes. By introducing a probabilistic evaluation of a design's "fitness," we also formalize strategies to reduce such uncertainties through prototyping, as visualized in Fig. A.2 (Right). Additionally, we emphasize that collaboration among team members with varied expertise and backgrounds, as well as engagement with stakeholders (including customers, scientific advisors, and domain experts), is essential for developing a high-performing soft robot. Finally, we conclude that co-design approaches not only help preserve design knowledge but also enhance reproducibility, addressing a well-known challenge in soft robotics (Baines et al., 2024).

Multi-Objective Co-Design: Satisfying Comprehensive Requirements and Evaluation Metrics Current co-design approaches for soft robots typically consider only one (Medvet et al., 2022; Navez et al., 2024; Spielberg et al., 2021, 2019; Wang et al., 2024,

2023b), or at most a few (Navez et al., 2025), evaluation metrics[1] per task. For locomotion tasks, this often means measuring the velocity achieved or the distance traveled in a fixed time period (Spielberg et al., 2019; Wang et al., 2023b), while in manipulation tasks (e.g., pushing), the focus is on how far an object is safely moved (Wang et al., 2024). However, as soft robots evolve from research prototypes into commercial products, a broader set of design values becomes critical. These include, but are not limited to, safety (as discussed in Chapter 3), manufacturing costs (especially in relation to production batch sizes) (Junge and Hughes, 2022; Majidi, 2014; Miriyev et al., 2017; Schmitt et al., 2018), material supply, and the robot's operational lifetime—which depends on both the materials used and the stresses experienced—as well as ecological and recycling considerations (Mazzolai and Laschi, 2020). Additionally, traditional design processes tend to address regulatory standards only at the end or not at all. For example, medical soft robots must meet the FDA or EMA guidelines, which dictate acceptable materials, manufacturing methods, and performance criteria. Incorporating these regulatory factors into the design optimization loop helps prevent late-stage redesigns that could delay product deployment and significantly increase development costs. Holistic co-design integrates these values and constraints from the beginning, ensuring that the resulting designs are optimal across all dimensions.

Increasing the Efficiency of Computational Co-Design *Computational co-design* for soft robots refers to developing an automatic algorithm that optimizes the integrated design (encompassing both body and control components) to meet specified performance criteria and design values (Carlone and Pinciroli, 2019; Wang et al., 2023a). In its most general form, co-design can be formulated as a constrained nonlinear optimization problem (Zardini, 2023) over a parameterized design x, such that

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0 \\ & h(x) \leq 0, \end{aligned} \tag{A.1}$$

where $f(x) : \mathcal{X} \to \mathcal{C}$ represents the cost or loss function, and $g(x) : \mathcal{X} \to \mathbb{R}^{n_{\text{eq}}}$ and $h(x) : \mathcal{X} \to \mathbb{R}^{n_{\text{ineq}}}$ denote the equality and inequality constraints, respectively. For example, in multi-objective optimization, we can define $f(x) = \left[\hat{c}_1^\top(x), \ldots, \hat{c}_{n_{\text{obj}}}^\top(x)\right]^\top$, with each $f_j(x) = \hat{c}_j(x)$ for $j = 1, \ldots, n_{\text{obj}}$ corresponding to design objectives implemented via the (estimated) evaluation metrics $\hat{c}_j(x) : \mathcal{X} \to \mathcal{C}_j$. These objectives might include task performance, manufacturing, and operational costs, or environmental impact. Equality constraints typically capture system dynamics, while inequality constraints ensure physical feasibility (such as non-negative volume, adherence to manufacturing tolerances, or minimum geometric dimensions for manufacturability) and guarantee that the design satisfies essential requirements (like minimum safety levels or regulatory standards). Please note that such (inequality) constraints can also be a function of an estimated evaluation metric $\hat{c}_j(x)$.

In our view, one key reason co-design approaches are not yet widely adopted in soft robotics is their computational inefficiency and high cost, which severely restrict

[1] Commonly, such evaluation metrics are also referred to as costs, losses, or rewards

the design space that can be explored and diminish the chances of finding the optimal design, thereby reducing their practical utility (Chen and Wang, 2020). We identify three primary sources of computational inefficiency in current methods: (i) they often operate in high-dimensional design spaces—for instance, by discretizing the soft robotic geometry into voxels—which makes it extremely challenging, if not impossible, to locate the global optimum; (ii) the optimization loop is closed via performance metrics obtained from one or multiple closed-loop system simulations, and using high-fidelity simulators makes this evaluation process computationally demanding; (iii) assessing closed-loop performance requires access to a controller. In principle, there are two ways to address this: one can train a controller over a set of different designs (Boekel, 2025; Zardini et al., 2021a), though this means the controller may not be optimized for a specific design, so the evaluation might not reflect the true performance achievable with a specialized controller. Alternatively, training a controller tailored to the proposed design can fully exploit its kinematics and dynamics to achieve optimal task performance, but this approach is computationally very intensive—especially when using RL controllers trained from scratch (Bhatia et al., 2021; Wang et al., 2023a, 2022, 2023b) or control policies trained via gradient descent using a differentiable simulation (Bächer et al., 2021; Spielberg et al., 2019; Wang et al., 2023a, 2024)—which often struggle with complex hybrid dynamics such as contact.

Our framework, depicted in Fig. A.3, paves the way for significantly more efficient computational co-design of soft robots by introducing four key modifications that address the previously mentioned challenges: (1) Building on previous work (Spielberg et al., 2019; Wang et al., 2024), we advocate for performing optimization in a reduced-order design space, which can be either entirely learned or partially defined manually (e.g., specifying the length and diameter of a soft segment). In practice, reduced-order designs x are sampled from a distribution, and then a design decoder reconstructs the full design description (such as a volumetric mesh with sensor and actuator placement). After evaluating the design, the optimizer updates the posterior belief of the sampling distribution to bias it towards effective designs. We elaborate on reduced-order design in Section A.4. (2) A control-oriented reduced-order model can greatly assist in analyzing the system's motion characteristics and behavior (Alkayas et al., 2025; Alora et al., 2023a; Armanini et al., 2023; Bruder et al., 2020; Ménager et al., 2023; Stölzle and Della Santina, 2024; Valadas et al., 2025). While deriving such models for rigid manipulators is generally straightforward (Siciliano et al., 2010; Zhao et al., 2020), soft robotics presents an inherent interplay between design, actuation, and task (e.g., payload, gravitational forces) on one side and the resulting deformations on the other. This interdependence makes it essential to jointly synthesize the design (including actuation) and the kinematic model. We discuss this further in Sec. A.6. (3) To reduce the reliance on performance metrics derived from resource-intensive closed-loop simulations, we introduce several auxiliary metrics that are computationally cheaper to evaluate. These metrics provide early feedback to the optimizer, allowing for the early discarding of designs with very low predicted fitness. Examples include metrics assessing observability and controllability based on the reduced-order model (which directly links actuator and sensor placement to the structural design), open-loop compliance (Guan et al., 2023), safety (as discussed in Chapter. 3), embodied intelligence (Cianchetti, 2021; Mengaldo et al., 2022; Vihmar et al., 2023), or heuristics estimating manufacturability. We further discuss controllability and observability metrics in Sec. A.6. (4) Finally, we propose leveraging the

reduced-order model to derive the control law in a model-based fashion. Recent advancements provide a solid foundation for exploiting dynamical models—whether physics-based or data-driven—for control (Della Santina et al., 2023; Laschi et al., 2023). For example, using Koopman theory to learn a linear model (or linearizing a nonlinear model around equilibrium) enables the design of optimal controllers in closed form via LQR (Bruder et al., 2020). For nonlinear models with a physical structure (i.e., with well-defined kinetic and potential energy terms) (Alkayas et al., 2025; Armanini et al., 2023; Liu et al., 2024a; Stölzle and Della Santina, 2024; Valadas et al., 2025), PID+Feedforward (Della Santina et al., 2023; Stölzle and Della Santina, 2024; Stölzle et al., 2024b) or similar closed-form controllers, such as PD+ (Della Santina et al., 2020c) or computed torque, can be designed. When dynamics are modeled as generic nonlinear transition functions, such as RNNs (Thuruthel et al., 2017) or NODEs (Kasaei et al., 2023), optimal control techniques such as MPC (Alora et al., 2023a) or model-based RL (Thuruthel et al., 2018b) can compute the control input. These model-based approaches to deriving a control law are dramatically more computationally efficient than training an RL control policy from scratch for each design (Bhatia et al., 2021; Wang et al., 2023a, 2022, 2023b).

Probabilistic Evaluation Metrics Another key tenet of holistic co-design is the explicit recognition of uncertainty when evaluating a design computationally across multiple dimensions (Chen and Wang, 2020). In contrast to many existing approaches (Wang et al., 2024) that rely on one or a few metrics derived from simulations as stand-ins for a design's actual performance, our perspective acknowledges that such performance metrics estimated via simulation are only proxies. For example, the DiffuseBot study (Wang et al., 2024) assesses various robotic tasks—such as balancing, landing, crawling, gripping, and box manipulation—using performance metrics based on closed-loop simulation outcomes. However, the well-known sim-to-real gap (Dubied et al., 2022) means that a design's simulation performance often deviates from its real-world behavior (Junge and Hughes, 2022). Moreover, important factors like manufacturing costs (Junge and Hughes, 2022), ecological sustainability, or the operational lifetime of a soft robot are difficult to evaluate solely through simulation and instead require prototyping, verification, and real-world testing, often with human involvement. As a result, the "optimal" design identified through computational co-design may not be optimal in practice.

To address these challenges, we propose a framework that adopts a probabilistic view of evaluation metrics—treating them as probabilistic beliefs about expected performance conditioned on the design specifications. This approach enables us to (a) explicitly account for uncertainty in the metrics during the design optimization process, (b) optimize the design also based on metrics that cannot be directly evaluated in simulation (e.g., manufacturability), and (c) progressively build confidence in the metrics and reduce uncertainty through high-fidelity simulations and/or prototyping. In our framework, the optimization process leverages the current probabilistic estimates of the metrics (i.e., refinement) to assess and improve designs computationally while using prototyping and experimental validation to refine these metric estimates (i.e., realization). This balance draws on well-established techniques from Bayesian optimization and reinforcement learning to manage the trade-off between refinement and realization effectively. In Fig. A.2 (Right), refinement is depicted as the iterative design cycles within the innermost layer, while realization is

A

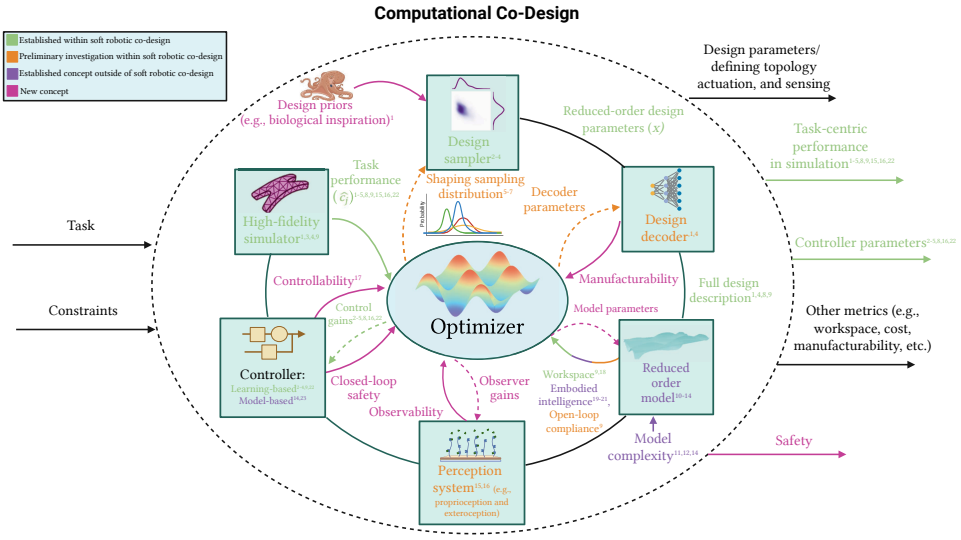


Figure A.3: **A Framework for Efficient Computational Co-Design of Soft Robots:** We sample reduced-order design parameters x from an initial distribution that can include design priors from biological inspiration (Chen and Wang, 2020; Laschi and Mazzolai, 2024; Mazzolai and Laschi, 2020), known mechanisms, or existing soft robot designs. This design space—either learned or explicitly defined by physical or geometric values (i.e., size optimization)—is translated by a design decoder into a detailed robot body description (such as a 3D mesh with sensor and actuator placements) that provides immediate manufacturability feedback. A reduced-order model is then derived (Alkayas et al., 2025; Alora et al., 2023a; Stölzle and Della Santina, 2024; Valadas et al., 2025) or learned to efficiently assess workspace, open-loop compliance, and embodied intelligence (Cianchetti, 2021; Mengaldo et al., 2022; Vihmar et al., 2023), offering inexpensive feedback to the optimizer. Similarly, perception and control systems can be developed using this model (via model-based control (Della Santina et al., 2023), MPC (Alora et al., 2023a), or differentiable physics (Bächer et al., 2021; Spielberg et al., 2019; Wang et al., 2023a)), while observability and controllability are evaluated without costly simulations. Finally, closed-loop simulations (low-fidelity using the reduced-order model or high-fidelity FEM-based (Coevoet et al., 2017)) assess integrated performance metrics, which in turn optimize the sampling distribution, decoder, and all relevant system parameters. *References in graphic:* (1): (Navez, 2024), (2): (Bhatia et al., 2021), (3): (Wang et al., 2023a), (4): (Wang et al., 2024), (5): (Song et al., 2024), (6): (Sutton et al., 1998), (7): (Garnett, 2023), (8): (Medvet et al., 2022), (9): (Guan et al., 2023), (10): (Armanini et al., 2023), (11): (Valadas et al., 2025), (12): (Alkayas et al., 2025), (13): (Ménager et al., 2023), (14): (Alora et al., 2023a), (15): (Spielberg et al., 2021), (16): (Junge and Hughes, 2022), (17): (Zheng et al., 2019), (18): (Amehri, 2022), (19): (Cianchetti, 2021), (20): (Mengaldo et al., 2022), (21): (Vihmar et al., 2023), (22) (Spielberg et al., 2019), (23): (Della Santina et al., 2023).

shown by selected designs transitioning to the outer layers—achieved through building prototypes with varying TR-Ls to enhance confidence in both the evaluation metrics and the designs. Ultimately, this methodology ensures that resources are allocated efficiently by prioritizing prototyping and testing in areas where predictive confidence is low but the potential for outstanding real-world performance is high. We discuss this approach in greater detail in Sec. A.5.

Synergistic Cross-Disciplinary Collaboration Collaboration is another pillar of this co-design process. The approach actively involves diverse stakeholders, engineers, end-users, material scientists, and domain experts from the outset. This collective input ensures

that requirements are practical and adaptable to real-world constraints. For instance, when designing a robotic arm for harvesting, growers provide insights into crop fragility and harvesting techniques, guiding the development of soft end-effectors that minimize damage and maximize yield.

In traditional design workflows, this kind of collaboration is often sequential: the modeling engineer finishes their work and hands it over to the control engineer, who then applies control strategies without sufficient feedback loops. This hand-off model creates information silos and limits opportunities for iterative improvement. In contrast, co-design promotes continuous communication between these roles. For instance, the modeling engineer and the control engineer work in tandem, iterating on the design as new insights emerge, resulting in a more refined final product.

Preserving Design Knowledge and Enabling Reproducibility A significant drawback of conventional design processes is the potential loss of the "design history". Key insights, decisions, and trade-offs often remain undocumented, especially when team members leave or shift roles. As a result, revisiting earlier design stages or learning from past iterations becomes challenging or even impossible. For instance, if a research team wants to tweak a previously tested actuator geometry after realizing performance limitations in the field, they might discover that the rationale behind the original design choices is no longer accessible.

Holistic co-design naturally addresses this issue by serving as a transparent log of the development process. Design discussions, iterations, and parameter choices are recorded throughout the project, creating an accessible "audit trail". If a team needs to revisit an earlier design step, they can do so with clarity, regardless of team turnover or changing project priorities. Furthermore, the audibility significantly eases the certification process, which we already discussed previously.

In a co-design approach, however, the process remains inherently flexible. Since parameter definitions, performance assessments, and stakeholder inputs are continuously documented, engineers can backtrack when needed to refine or adjust the design by recalibrating safety margins based on updated risk analyses, testing alternative hardware configurations to optimize for cost or power consumption, or revisiting software algorithms in response to emerging performance data. This iterative workflow inherently aligns technical feasibility, regulatory compliance, and evolving business objectives in a transparent manner, ensuring that every modification is grounded in both empirical evidence and strategic considerations.

A.4 Reduced-Order Design Parametrizations

Since direct optimization of soft robots requires parameterization, we represent the design via n_x variables $x \in \mathcal{X}$, where $\mathcal{X} = \mathbb{R}^{n_x}$ defines an n_x-dimensional continuous design space.[2] In soft robot co-design, these parameters typically encode essential characteristics, including the spatial geometry of the robot's body, actuator (Wang et al., 2024) and sensor (Junge and Hughes, 2022; Spielberg et al., 2021) placement, material selection, and other structural attributes. Traditionally, two methods have been employed to parameterize soft robot

[2]While a continuous space is assumed here, discrete design choices can be naturally embedded into this framework.

designs: (1) size optimization (Chen and Wang, 2020), where (morphological) design optimization parameters such as the number of segments, radii, segment lengths, and materials are chosen explicitly (Calisti et al., 2011; Guan et al., 2023; Junge and Hughes, 2022; Navez et al., 2024; Pagliarani et al., 2024; Polygerinos et al., 2015), or (2) discretization-based shape/topology[3] optimization (Chen and Wang, 2020), which partition the continuous design into two- or three-dimensional voxels or particles (Bhatia et al., 2021; Caasenbrood et al., 2020; Medvet et al., 2022; Nadizar et al., 2022; Pinskier et al., 2024; Wang et al., 2022), each categorized as empty (removed), passive (unactuated), active (actuated), or sensorized.

However, these traditional approaches have distinct drawbacks. Manually selected parameters are often suboptimal, can introduce redundancies, or might include parameters that have minimal or no impact on design objectives, unnecessarily complicating the co-design process. Discretization-based parameterizations, on the other hand, create high-dimensional optimization spaces that are computationally expensive to explore effectively. Furthermore, these discretizations typically oversimplify practical constraints, such as interactions between actuators and structural components, and frequently yield designs difficult to realize in practice (Legrand et al., 2023).

To overcome these challenges, recent studies (Navez, 2024; Song et al., 2024; Wang et al., 2024) have proposed optimizing within a reduced-order design space and then using a (potentially learned) generative decoder to reconstruct the complete design—such as generating a 3D mesh of the soft robot that includes sensor and actuator placements. In this context, we incorporate a design sampler in the reduced-order space along with a design decoder into our comprehensive co-design framework. Specifically, we define a probabilistic design policy $\vartheta(x) : \mathcal{X} \to [0,1]$ that maps the design parameters x to a probability $\Pr(x)$. During co-design, the generator samples parameters from this distribution, $x \sim \vartheta(\cdot)$, which are then expanded into a full design description $d \in \boldsymbol{D}$ by the decoder $d = \psi(x)$. The decoder may be either model-based - e.g., utilizing parametric CAD - or implemented as a learned generative model, such as VAEs (Navez, 2024; Song et al., 2024), Generative Adversarial Networks (GANs) (Hu et al., 2022), or Diffusion models (Wang et al., 2024). Design priors—for example, proven soft robot designs (Navez, 2024), or biological inspiration (Chen and Wang, 2020; Laschi and Mazzolai, 2024; Mazzolai and Laschi, 2020)—can inform the initial guess (i.e., the prior) for the distribution $\vartheta(x)$. Throughout the co-design process, the optimizer iteratively refines the posterior belief of $\vartheta(x)$ via update steps (Song et al., 2024; Sutton et al., 1998).

This flexible framework supports both model-based and learning-based approaches while enabling optimization in a reduced-order space, thereby making the co-design process more computationally tractable without sacrificing the detailed design information required for accurate evaluation and subsequent fabrication.

Sensitivity analysis can play a crucial role within the aforementioned "design generator" by reducing the dimensionality of the design parameters we sample and improving the conditioning of the design decoder (Chen and Wang, 2020; Guan et al., 2023; Navez, 2024). When choosing geometrical parameters, manufacturing constraints, and actuation variables,

[3]Shape optimization refers to adjusting the boundaries or surfaces of an existing design while maintaining the overall connectivity (or topology) of the structure. On the other hand, topology optimization rethinks the structure from the ground up while potentially adjusting the connectivity (e.g., creating or removing elements) of the structure (Chen and Wang, 2020).

sensitivity analysis is vital for pinpointing and eliminating parameters that exert minimal influence on performance. This involves systematically varying parameters—such as wall thickness, segment lengths, material stiffness, and actuator placement—to evaluate their impact on key performance metrics like gripping force, range of motion, and energy consumption. Formally, for a given metric $m_j(x) : \mathbb{R}^{n_x} \to \mathbb{R}^{n_{c_j}}$ that maps design parameters to an optimization objective c_i, the local sensitivity of the ith design parameter x_i at a design x is given by $s_i = \left|\frac{\partial m_j(x)}{\partial x_i}\right| \in \mathbb{R}^{n_{c_j}}$, which measures the magnitude of its influence. For instance, in a soft robotic gripper designed for fruit harvesting, sensitivity analysis may show that finger curvature and material elasticity have a significant effect on grasping delicate produce (yielding a high s_i), whereas small variations in actuator placement may have a negligible impact (a low s_i). Consequently, the optimization process can focus computational resources on refining the most critical parameters—either by (a) employing a curriculum approach that defers optimization of less sensitive parameters until later stages or (b) completely fixing these parameters to reduce the search space.

Encoding deformations in a low-dimensional subspace improves computational efficiency for both design optimization and control. Rather than tackling high-dimensional, resource-intensive simulations, designers can employ reduced-order models that encapsulate the primary deformation modes of soft robotic morphology.

A.5 Probabilistic Co-Design Metrics: Formalizing the Refinement vs. Realization Trade-Off

As discussed earlier, our holistic co-design framework inherently balances a trade-off between refinement and realization when evaluation metrics—such as manufacturability or task-centric performance—are treated probabilistically. In this setting, refinement involves running a computational co-design cycle that evaluates new design candidate(s) and updates the posterior design sampling policy based on the current evaluation metric prior. Conversely, realization seeks to enhance our confidence in the metric by performing high-fidelity simulations and/or fabricating prototypes with progressively higher TRLs (Junge and Hughes, 2022).

To formalize this further, consider an evaluation metric $c_j = m_j(x)$ that assigns a cost $c_j \in \mathcal{C}_j$ to a design $x \in \mathcal{X}$, which we aim to minimize during the co-design process. Here, the subscript j indicates the jth metric and its associated cost c_j. As noted earlier, the soft robot co-design problem is typically addressed in a multi-objective context where various metrics capture aspects such as task performance (e.g., task completion time, energy consumption), safety, controllability, observability, manufacturing costs, and more. In practice, however, accurately evaluating these metrics is typically very expensive since it involves building a prototype and conducting extensive field trials or user studies. Instead, we may have access to a simplified computational model or a current probabilistic belief about the metric, denoted as $\hat{m}_j(\hat{c}_j, x) : \mathcal{C}_j \times \mathcal{X} \to [0,1]$, which maps a design x and an estimated cost[4] $\hat{c}_j \in \mathcal{C}_j$ to a probability $\Pr(\hat{c}_j, x)$. We now detail the definitions of refinement and realization in this framework.

Refinement. During refinement, we adopt a Monte Carlo procedure by querying

[4]The cost is equivalent to a loss and may also be viewed as a negative reward.

A

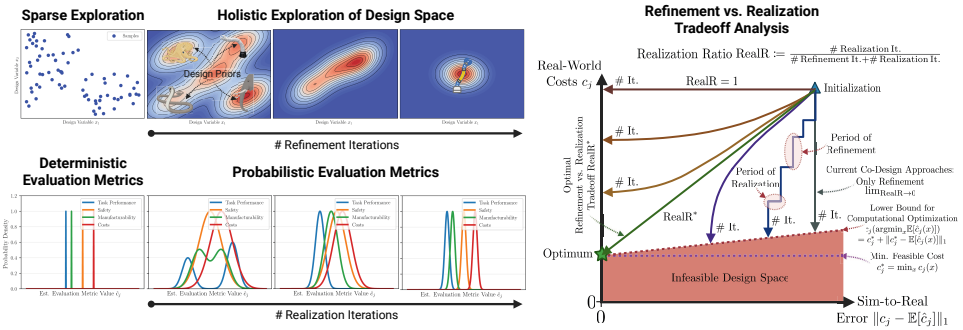


Figure A.4: **Refinement vs. Realization.** *Top Left:* This panel shows how refinement affects the design sampling distribution $\vartheta(x)$. Unlike (most) conventional co-design methods that sparsely explore the design space, we condition $\vartheta(x)$ on design priors (e.g., biological inspiration and existing soft robot designs) and iteratively update it via the co-design optimizer until the optimum is reached. *Bottom Left:* Here, we demonstrate how iterative realization refines probabilistic evaluation metrics. Rather than using deterministic metrics that ignore inherent simulation uncertainty, we define a probabilistic distribution $\hat{m}_j(\hat{c}_j, x)$ for each metric, with purposeful prototyping reducing the uncertainty over iterations. *Right:* This panel examines the tradeoff between refinement and realization. We define the realization ratio RealR as the fraction of realization iterations relative to the total iterations and the sim-to-real error as $\|c_j - \mathbb{E}[\hat{c}_j]\|$. We then plot the evolution of real-world cost c_j versus sim-to-real error as iterations increase for a fixed RealR: when $\lim \text{RealR} \to 0$, designs are optimized computationally with realization occurring only at the end, while RealR = 1 means only the confidence in the evaluation metrics is increased. The optimal ratio RealR* efficiently minimizes the real-world cost to its optimum $\min_x c_j(x)$ via refinement and reduces the sim-to-real error to zero by updating $\hat{m}_j(\hat{c}_j, x)$ via realization.

the metric conditioned on the design, i.e., $\hat{c}_j \sim m_j(\cdot, x)$. The resulting estimated cost $\hat{c}_j$ is then employed either as an optimization objective or as part of a constraint. Crucially, the optimizer leverages this estimated cost to update the posterior distribution of the design variables x, adjust any free parameters of the decoder $\vartheta(x)$, and tune other pertinent parameters such as control or observer gains or hyperparameters of the reduced-order model. This process represents one iteration of computational co-design, which can be visualized as one cycle in Fig. A.3 and as a cycle within the innermost layer in Fig. A.2 (Right). Drawing an analogy to traditional product development cycles, refinement resembles the *diverging* phase, where new concepts and designs are explored and assessed. Similarly, during *exploitation* phases in RL, where a fixed action policy $\pi(a, s)$ is followed during inference without necessarily updating the policy's parameters. In our context, instead of a reward for following the action policy, our outcome is a posterior design generator distribution along with updated design decoder parameters.

Realization. In realization, our goal is to boost our confidence in the defined metrics. Specifically, we aim to maximize the expected information gain regarding the metrics for the current best design—effectively reducing the entropy (or uncertainty) of the metric's posterior around the optimal design. To do this, we can employ methods like Predictive Entropy Search (PES) from Bayesian optimization (Hernández-Lobato et al., 2014), which assists in pinpointing the global minimizer x^* of the jth cost c_j. The acquisition function $\alpha_{\text{PES}}(x) : \mathcal{X} \to C_j$ then enables us to select the next candidate point $x_{\text{n}} = \arg\min_{x \in \mathcal{X}} \alpha_{\text{PES}}(x)$ that most effectively reduces the uncertainty regarding the location of x^* (Hernández-

Lobato et al., 2014), where

$$\alpha_{\text{PES}}(x) = H[\hat{m}_j(x)] - \mathbb{E}_{\hat{m}_j(x^*)}\left[H\left(\Pr(\hat{c}_j \mid x, x^*)\right)\right], \tag{A.2}$$

and $H[p(x)] = -\int p(x)\log(p(x))\,\mathrm{d}x$ denotes the differential entropy. Another challenge in holistic co-design is that our goal is not only to reduce uncertainty about the optimum of a single metric through realization but to do so for all metrics involved in the multi-objective optimization problem. Consequently, future research should focus on developing methods that effectively select the next sample point x_{n} for realization, simultaneously minimizing the uncertainty across multiple metric estimates $\hat{m}_j(x_{\text{n}})$ while also accounting for the cost, time, and effort required to obtain accurate ground-truth measurements. We conclude that in holistic co-design, realization aligns with the *converging* phases of traditional product development, where concepts and designs are validated through simulations or prototypes to reduce risk and uncertainty, thereby informing better future decisions (for instance, by ruling out designs that are suboptimal based on more certain metrics). Similarly, just as RL utilizes random policies to explore unseen state-action pairs (Sutton et al., 1998) within *exploration* phases[5], we advocate leveraging high-fidelity simulators and prototypes to verify and enhance metrics for previously untested designs.

In practice, it is crucial to strike a balance between refinement and realization, similar to the exploitation vs. exploration tradeoff in RL. Refinement can generate (potentially) better designs at the cost of computational time, and the resulting design may not be optimal due to uncertainties in the predicted costs of the various metrics—essentially, a discrepancy between model predictions and reality. In contrast, realization does not directly enhance the design; instead, it bolsters our confidence in the metrics through high-fidelity simulations and prototyping tests at different TRLs, often incurring significant expense, particularly when human designers are involved in fabricating and validating the prototypes. To manage this trade-off, established techniques from RL (Sutton et al., 1998) and Bayesian optimization (Garnett, 2023) can be applied. For example, one viable approach is to assess the Expected Improvement (EI) (Jones et al., 1998) when sampling a new design and building its prototype while accounting for the uncertainty inherent in the metric under consideration.

Even when a balance between refinement and realization is achieved, realization—through prototyping and real-world validation—remains both resource- and cost-intensive. To address this challenge, we propose that the community create a shared database where researchers and engineers can contribute soft robotic designs along with their performance benchmarks. In line with efforts to enhance soft robot benchmarking (Bhatia et al., 2021; Wang et al., 2023a) and reproducibility (Baines et al., 2024), such a repository would improve the initial probabilistic beliefs in the metrics and reduce uncertainty in computational evaluations. Ultimately, this would lessen the reliance on expensive realization processes, which is especially important for aspects like serial production costs that are impractical to repeat multiple times during a given design cycle.

[5] A key difference between exploration in RL and realization in co-design is that in product development, the final optimized system must be implemented, whereas in RL, further exploration may be unnecessary if the policy already meets performance expectations.

A

A.6 Co-Design of Physical Intelligence

For rigid robotic manipulators, the joint and link topology is well-defined, which simplifies deriving a kinematic model that describes the robot's state using a limited number of configuration variables (e.g., joint angles) (Siciliano et al., 2010; Zhao et al., 2020). Such models are crucial for downstream tasks like model-based control, model-based RL, and motion planning. In contrast, the continuous deformability of continuum soft robots makes it challenging to establish such reduced-order models. Traditionally, the soft robotics community has relied on expert intuition and trial-and-error to determine finite-dimensional descriptions of the robot's backbone shape, such as PCC (Webster III and Jones, 2010), PCS (Renda et al., 2018), GVS (Renda et al., 2020). However, the appropriateness of these choices depends heavily on the robot's topology, actuation, and dynamic modes. Consequently, it is vital to co-design the reduced-order model alongside the soft robot hardware. In this context, our focus is on applying algorithms capable of automatically identifying a suitable, potentially control-oriented model without requiring user intervention. One promising direction involves modern machine learning methods to derive reduced-order models in a data-driven fashion (Alora et al., 2023a; Bern et al., 2020; Chen et al., 2024; Ménager et al., 2023; Thuruthel et al., 2017), with recent applications emerging in co-design/design optimization (Navez, 2024). Alternatively, to address the sample inefficiency and limited extrapolation of fully learned models, the community has recently proposed methods to automatically adapt existing finite-dimensional strain approximations to a given soft robot design using data-driven approaches (Alkayas et al., 2025; Navez et al., 2025; Valadas et al., 2025).

As noted in Sec. A.2, most current co-design approaches indirectly assess the system's controllability and observability by running simulations or real-world experiments with a learned controller. However, this process provides a weak signal because it is difficult to determine whether poor control performance arises from an inadequate controller or from inherent challenges in controlling or observing the system. We contend that integrating direct metrics for controllability and observability into the co-design process could accelerate convergence and reduce the reliance on computationally expensive simulations and experiments. For example, the controllability and observability of a nonlinear soft robot dynamic model can be evaluated using established techniques (Griffith and Kumar, 1971; Zheng et al., 2019). Alternatively, one may linearize the system around an equilibrium (e.g., the straight configuration) and then use the resulting state-space description to assess the well-known linear properties. Moreover, it is important to consider the stability of the closed-loop system dynamics. When guiding the soft robot towards a desired configuration q^{d}, previous work has shown that common control strategies, such as PD+Feedforward, are locally stable within a region where the potential field is convex (e.g., when $\frac{\partial^2 \mathcal{U}}{\partial q^2} + K_{\mathrm{p}} \succ 0$). Given that we typically aim to keep proportional feedback gains as low as possible to enhance phase margins and reduce control effort, it follows that achieving closed-loop stability over a larger region—or even global asymptotic stability—requires optimizing the soft robot design so that $\frac{\partial^2 \mathcal{U}}{\partial q^2} \succ 0$ throughout the entire desired workspace.

A.7 Exploiting Safety vs. Performance Trade-off via Multi-Objective Optimization

Safety and performance are fundamental components in multi-objective co-design, often in conflict. Specifically, safety can be treated as either a maximization objective—continuously optimizing for compliance—or a design/control constraint, ensuring a minimum threshold is met. Furthermore, unlike rigid robotics, which predominantly ensures safety through computational methods, or conventional soft robotics, which relies primarily on material compliance, we argue in this chapter that the safety of soft robots should be addressed by jointly considering both the characteristics of the robot's *body* and its *brain*.

In our proposed holistic approach, we treat the trade-off between safety and performance as a multi-objective optimization problem during the design of the soft robot's *body* (i.e., morphology) and *brain* (i.e., controller). Specifically, we simultaneously maximize both objectives while enforcing a constraint to ensure that the worst-case safety of the closed-loop system always meets or exceeds the specified task requirements. During deployment (i.e., the control phase), adaptive strategies dynamically balance safety and performance according to real-time conditions, again ensuring safety always remains above the required task threshold. This layered approach broadens the design space by reducing constraints traditionally imposed on structural components. It also increases resilience, as the robot remains safe even in scenarios where control performance is compromised.

A.8 Conclusion and Open Challenges

A.8.1 Conclusion

In this chapter, we introduced an approach that leverages holistic co-design to develop soft robots that are both safe and high-performing. First, we reviewed existing soft robot design approaches, covering both traditional design processes and early research on co-design. This review revealed shortcomings in current co-design methods, such as computational inefficiencies and a tendency to focus solely on aspects that can be fully simulated while neglecting critical downstream factors like prototyping, real-world evaluation, manufacturability, safety, and regulatory issues. We subsequently proposed a framework for holistic co-design that incorporates several essential modifications and additions. This framework emphasizes the optimization of broader design values—such as manufacturability, safety, and cost considerations in both manufacturing and operation—rather than relying solely on a single, simulation-based performance metric. Furthermore, we identified several avenues to boost the computational efficiency of co-design, including the use of reduced-order design spaces, the co-learning of reduced-order models, and auxiliary metrics that provide early, cost-effective feedback on controllability, observability, and open-loop compliance, as well as the model-based derivation of controllers. We also presented a probabilistic perspective on co-design metrics, explaining how high-fidelity simulations and a strategic selection of prototypes can build confidence and reduce uncertainty in evaluation metrics. Looking ahead, exploiting the trade-off between design refinement and realization may lead to solutions that are not only optimal in simulation but also effective in the real world. Finally, we stressed the importance of synergistic cross-disciplinary collaboration, noting that a holistic co-design approach can help preserve design knowledge and enhance reproducibility within soft robotics.

This holistic co-design approach would also be instrumental in formalizing the trade-off between safety and performance, allowing us to pinpoint effective soft robotic designs—integrating both morphology and control—that excel at the task while satisfying a broad range of values and requirements without compromising safety.

A.8.2 Next Steps and Open Challenges

As the next steps, we suggest taking initial steps to enable its realization. This includes but is not limited to: (1) deriving and verifying metrics such as manufacturability, fabrication and operation costs, controllability, and observability in a model-based fashion so they can be computationally evaluated and integrated into co-design methodologies—with an emphasis on making these metrics openly accessible and easily combinable; (2) investigating effective methods to incorporate design priors, such as existing soft robotic designs (Navez et al., 2024) or biological inspirations from invertebrates (Chen and Wang, 2020; Krieg et al., 2015; Laschi et al., 2012; Laschi and Mazzolai, 2024), into the design sampling process. Generative and foundation models—such as VLMs trained on internet-scale data (Grattafiori et al., 2024; Hurst et al., 2024)—offer promising avenues for accelerating the development of novel soft robotic architectures. Initial studies should explore how these models can serve as design samplers by proposing bioinspired or unconventional morphologies that might not arise from traditional optimization routines. For instance, a VLM could be prompted to suggest soft robot designs for specific tasks and constraints (Ghasemi and Moghaddam, 2025; Stella et al., 2023a), potentially conditioned on renderings or descriptions of previously evaluated designs - i.e., through in-context learning (Brown et al., 2020), which could then be mapped into a reduced-order design space x via a learned neural network. Connected to this, (3) VLMs could also be effective as design critics (Ghasemi and Moghaddam, 2025)—assessing factors such as perceived human safety, environmental impact, feasibility, ergonomics, and functional versatility, aspects that are challenging to model from first principles. (4) As discussed in Sec. A.5, it is crucial to investigate how to effectively exploit the trade-off between design refinement and realization. The ideal approach should account for the resources (e.g., effort, cost) required for each refinement and realization cycle, allocating them optimally to enhance design improvement efficiency. (5) As emphasized in Sec. A.6, a major challenge in the co-design of soft robots is optimizing effective control policies in a computationally tractable manner. One potential alternative to the model-based control approach presented here is to draw inspiration from emerging X-Embodiment policies (O'Neill et al., 2024) developed for rigid manipulators and to develop controllers that work across a variety of soft robotic designs, thereby avoiding the need to train a specialized controller for each design from scratch. Moreover, (6) characterizing the Pareto front between safety and performance will enable a structured trade-off analysis, particularly in common tasks such as pick-and-place, where payload capacity, speed, and compliance must be optimized simultaneously.

B

Supplementary Material to Chapter 11 - Coupled Oscillator Networks

This appendix provides supplementary material for Chapter 11. In particular, we offer further details on the training and evaluation process, including the mathematical definitions of all baseline methods and evaluation metrics considered. Additionally, we present extra results on learning latent dynamics—such as a complete set of evaluation metrics and the number of trainable parameters for every evaluated method—as well as sequences of stills illustrating the prediction results of CON across all datasets. Finally, we include additional findings on model-based latent space control, featuring baselines like a pure feedback controller based on a NODE model and one based on the learned CON dynamics.

This appendix is partly based on 🗎 🏆 ***M. Stölzle**, and C. Della Santina (2024). Input-to-State Stable Coupled Oscillator Networks for Closed-form Model-based Control in Latent Space. In Proceedings of Advances in Neural Information Processing Systems (NeurIPS) 37, **Spotlight** (Stölzle and Della Santina, 2024).* M.S. and C.D.S conceived the project, developed the formulation of the coupled oscillator network, and derived the stability proof. M.S. devised the closed-form approximation to the CON dynamics, came up with the model-based control approach, and implemented the training and control pipeline. M.S. planned and conducted all experiments and wrote the manuscript. C.D.S. revised the manuscript, provided funding, and supervised the project.

B.1 Appendix on Experimental Setup

B.1.1 Autoencoder Architecture

For the encoder and decoder, we rely on a vanilla CNNs implemented as a β-VAE (Kingma and Welling, 2014).

B

Encoder. The encoder consists of two convolutional layers with kernel size $(3,3)$ and stride $(1,1)$ mapping to 16, 32, respectively. The features are flattened and then passed to two linear layers with hidden dimensions 256 and n_z. Each layer (except for the last) is followed by a layer norm (Ba et al., 2016) and a LeakyReLU nonlinearity.

Decoder. The decoder first uses two linear layers to map to hidden dimensions of 256 and 32768, respectively. We then apply two 2D transposed convolutions (Dumoulin and Visin, 2016), reducing the number of channels first to 16, and then to 1. Each layer (except for the last linear and last convolutional) is followed by a layer norm (Ba et al., 2016) and a LeakyReLU nonlinearity. Finally, we apply a sigmoid function to clip the output into the range $[-1,1]$.

B.1.2 Latent Dynamic Models

In the following section, we provide implementation details for the latent dynamic models that we evaluated as part of this work.

Coupled Oscillator Network (CON)

We leverage the CON in $\mathcal{W}$-coordinates given by (11.3) for learning latent space dynamics. Specifically, we consider the input-to-force mapping $g(u)) = B(u)\,u(t)$, where $B(u) \in \mathbb{R}^{n\times m}$ is parametrized by few-layer MLP. We report results for two different sizes of the MLP: one medium-sized variant consisting of five layers with a hidden dimension of 30 and a small variant with two layers and a hidden dimension of 12. In both cases, we use a hyperbolic tangent as a nonlinearity.

When training the model, we jointly optimize $M_{\mathrm{w}}^{-1}, K_{\mathrm{w}}, D_{\mathrm{w}}, b$ and $g(u)$. However, we also need to make sure that we adhere to the stability constraints $M_{\mathrm{w}}^{-1}, K_{\mathrm{w}}, D_{\mathrm{w}} \succ 0$. For this, we leverage the Cholesky decomposition (Petersen et al., 2008). Instead of directly learning the full matrix $A \in \mathbb{R}^{n_z\times n_z}$, we designate the elements of an upper triangular matrix $U \in \mathbb{R}^{n_z\times n_z}$ as the trainable parameters. The Cholesky decomposition demands that $\mathrm{diag}(U_{11},\dots,U_{n_z n_z}) > 0$. Therefore, we apply the operation

$$U_{ii} = \log\left(1 + e^{\check{U}_{ii}+\epsilon_1}\right) + \epsilon_2, \tag{B.1}$$

where $\check{U}$ is the learned upper triangular matrix, and $\epsilon_1 = 1\times 10^{-6}$ and $\epsilon_2 = 2\times 10^{-6}$ are two small, positive values. The positive-definite matrix A is now given by $A = U^\top U \succ 0$.

Neural ODEs

We consider two kinds of NODEs (Chen et al., 2018): the vanilla $f_{\mathrm{NODE}} : \xi(t)\times u(t) \mapsto \dot{\xi}(t)$ maps latent state and system actuation directly into a time derivative of the latent state. In contrast, for the *MECH-NODE*, we enforce the latent dynamics to have a mechanical structure

$$\dot{\xi}(t) = \begin{bmatrix} \frac{\mathrm{d}z}{\mathrm{d}t} \\ \frac{\mathrm{d}\dot{z}}{\mathrm{d}t} \end{bmatrix} = \begin{bmatrix} \dot{z}(t) \\ f_{\mathrm{MECH-NODE}}(\xi(t), u(t)) \end{bmatrix}. \tag{B.2}$$

We represent both f_{NODE} and $f_{\text{MECH–NODE}}$ as MLPs consisting of 5 layers, a hidden dimension of 30, and a hyperbolic tangent nonlinearity.

Autoregressive models

For the below stated autoregressive models, we divide the integration between two (latent) samples $\xi(t_k)$ and $\xi(t_{k+1})$ into N_{int} integration steps $\xi(t_k + \delta t), \dots, \xi(t_k + k'\delta t), \dots, \xi(t_k + N_{\text{int}}\delta t)$ where δt is the integration step size and $t_{k+1} = t_k + N_{\text{int}}\delta t$. The autoregressive model now describes the transition $\xi(t_{k'+1}) = f_{\text{ar}}(\xi(t_{k'}), u(t_k))) \, \forall k' \in 1, \dots, N_{\text{int}}$.

RNN. We implement a standard, single-layer Elman RNN with `tanh` nonlinearity. The hidden state captures the latent state of the system. The latent state transition functions are given by

$$\xi(t_{k'+1}) = \tanh(W_{\text{hh}}\, \xi(t_{k'}) + b_{\text{hh}} + W_{\text{ih}}\, u(t_k) + b_{\text{ih}}), \tag{B.3}$$

where $W_{\text{hh}} \in \mathbb{R}^{2n_z \times 2n_z}$, $b_{\text{hh}} \in \mathbb{R}^{2n_z}$, $W_{\text{ih}} \in \mathbb{R}^{2n_z \times m}$, and $b_{\text{ih}} \in \mathbb{R}^{2n_z}$.

GRU. We implement a standard, single-layer GRU (Cho et al., 2014) with `sigmoid` activation function, where we interpret the latent state of the system as the hidden state of the cell. The latent state transition functions are given by

$$\begin{aligned} r &= \sigma\left(W_{\text{hr}}\, \xi(t_{k'}) + b_{\text{hr}} + W_{\text{ir}}\, u(t_k) + b_{\text{ir}}\right) \\ p &= \sigma\left(W_{\text{hp}}\, \xi(t_{k'}) + b_{\text{hp}} + W_{\text{ip}}\, u(t_k) + b_{\text{ip}}\right) \\ n &= \tanh\left(r \odot \left(W_{\text{hn}}\, \xi(t_{k'}) + b_{\text{hn}}\right) + W_{\text{in}}\, u(t_k) + b_{\text{in}}\right) \\ \xi(t_{k'+1}) &= (1 - p) \odot n + p \odot \xi(t_{k'}) \end{aligned} \tag{B.4}$$

where σ is the sigmoid function, $\odot$ the Hadamard product, $W_{\text{hr}}, W_{\text{hp}}, W_{\text{hn}} \in \mathbb{R}^{2n_z \times 2n_z}$, $W_{\text{ir}}, W_{\text{ip}}, W_{\text{in}} \in \mathbb{R}^{2n_z \times m}$, and $b_{\text{hr}}, b_{\text{ir}}, b_{\text{ip}}, b_{\text{in}} \in \mathbb{R}^{2n_z}$.

coRNN. A time-discrete coRNN is defined by the transition function

$$\xi(t_{k'+1}) = \begin{bmatrix} z(t_{k'+1}) \\ \dot{z}(t_{k'+1}) \end{bmatrix} = \begin{bmatrix} z(t_{k'}) + \delta t\, \dot{z}(t_{k'}) \\ \dot{z}(t_{k'}) + \delta t\, (-\gamma z(t_{k'}) - \varepsilon \dot{z}(t_{k'}) + \tanh(W \xi(t_{k'}) + V u(t_k) + b)) \end{bmatrix} \tag{B.5}$$

where $\gamma, \varepsilon \in \mathbb{R}^+$ are positive, scalar hyperparameters representing the stiffness and damping coefficients, respectively. The term $\tanh(W \xi(t_{k'}) + V u(t_k) + b)$ with $W \in \mathbb{R}^{2n_z \times 2n_z}$, $V \in \mathbb{R}^{n_z \times m}$, and $b \in \mathbb{R}^{n_z}$ contributes nonlinear state-to-state connections. It is implemented with a linear layer operating on $(\xi(t_{k'}), u(t_k))$ followed by a hyperbolic tangent nonlinearity.

CFA-CON. We adapt the Alg. 3 for predicting the time evolution in latent-space

$$\xi(t_{k'+1}) = f_{\text{CFA–CON}}(\xi(t'_k), u(t_k)), \tag{B.6}$$

where $f_{\text{CFA–CON}}$ describe the autoregressive state transition by the CFA-CON model as introduced in Eq. 11.45.

B.1.3 First-Order Variants of Dynamical Models

For learning (latent) dynamics of 1st-order systems (e.g., the reaction-diffusion dataset *R-D*), it might also be beneficial to formulate the dynamical model to be of 1st-order. While this is straightforward for some dynamics that do not explicitly take the order into account (e.g., RNN, GRU, NODE), for other models such as coRNN, CON, and CFA-CON, more adjustments are necessary. Namely, we substitute the $\frac{\mathrm{d}z}{\mathrm{d}t}$ component of the ODE with the expression for $\frac{\mathrm{d}\dot{z}}{\mathrm{d}t}$. Furthermore, we remove any terms that depend on the velocity $\dot{z}$ (e.g., damping effects). Below, we report in detail the adapted, 1st-order formulations for the coRNN, CON, and CFA-CON models.

CON. In the 1st-order version, we adapt the standard, 2nd-order ODE of the CON network as defined in (11.3) to

$$\dot{\xi}(t) = \dot{z}(t) = M_{\mathrm{w}}^{-1}\left(g(u(t)) - K_{\mathrm{w}} z(t) - \tanh(z(t) + b)\right). \tag{B.7}$$

coRNN. In the 1st-order version, we define the transition function as

$$\xi(t_{k'+1}) = z(t_{k'+1}) = z(t_{k'}) - \delta t\, \gamma\, z(t_{k'}) + \delta t \tanh\left(W \xi(t_{k'}) + V u(t_k) + b\right). \tag{B.8}$$

CFA-CON. We adapt a 1st-order version of Alg. 3 for predicting the time evolution in latent-space

$$\begin{aligned} \xi(t_{k'+1}) = z(t_{k'+1}) &= z(t_{k'}) + \int_{t_{k'}}^{t_{k'}+\delta t} F(t_{k'}) - \kappa\, z(t')\,\mathrm{d}t', \\ F(t_{k'}) &= g(u(t_k)) - (K - \kappa) z(t_{k'}) - \tanh(W z(t_{k'}) + b), \end{aligned} \tag{B.9}$$

where the closed-form solution for the integral is given by

$$\int_{t_{k'}}^{t_{k'}+\delta t} F(t_{k'}) - \kappa\, z(t')\,\mathrm{d}t' = \frac{F(t_{k'})}{\kappa}\left(1 - e^{-\kappa\,\delta t}\right). \tag{B.10}$$

B.1.4 Training

We implement the network dynamics and the neural networks (e.g., encoder, decoder, and MLPs) in JAX (Bradbury et al., 2018) and Flax (Heek et al., 2023), respectively. When training or inferring time-continuous dynamical models (e.g., NeuralODE, CON), we rely on Diffrax (Kidger, 2021) for numerical integration of the ODE using the Dormand-Prince's 5/4 method (Dormand and Prince, 1980) (Dopri5). For the numerical integration of both the time-continuous and the time-discrete models (e.g., RNN, coRNN, CFA-CON), we use an integration time-step δt of 0.025 s and 0.01 s for the *Toy Physics* (Botev et al., 2021) and soft robotic datasets, respectively.

Because of the GPU memory constraints, we limit ourselves to a batch size of 30 and 80 trajectories for the *Toy Physics* (Botev et al., 2021) and soft robotic datasets, respectively. We implement a learning rate schedule consisting of a warm-up (5 epochs) and a cosine annealing (Loshchilov and Hutter, 2016) period (remaining epochs). We employ an AdamW

optimizer (Kingma and Ba, 2014; Loshchilov and Hutter, 2018) with $\beta_1 = 0.9$, $\beta_2 = 0.999$ for updating both neural network weights (e.g., encoder, decoder) and parameters of the dynamical model (e.g., $K, D_\mathrm{w}, M_\mathrm{w}$, etc.).

B.1.5 Evaluation Metrics

Similar to other publications in the field (Liu et al., 2018a; Stölzle et al., 2022; Yu et al., 2018), we state the RMSE, the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index Measure (SSIM) (Wang et al., 2004) between the ground-truth image $o \in \mathbb{R}^{h_\mathrm{o} \times w_\mathrm{o}}$ and the predicted image image $\hat{o} \in \mathbb{R}^{h_\mathrm{o} \times w_\mathrm{o}}$. We use the separated test set for all evaluation results.

Root Mean-Square Error

The RMSE between the two images is given by

$$\mathrm{RMSE}(o, \hat{o}) = \sqrt{\sum_{u=1}^{h_\mathrm{o}} \sum_{v=1}^{w_\mathrm{o}} \frac{(o_{uv} - \hat{o}_{uv})^2}{h_\mathrm{o}\, w_\mathrm{o}}}. \tag{B.11}$$

Peak Signal-to-Noise Ratio

The PSNR is a function of the total MSE loss and the maximum dynamic range of the image L.

$$\mathrm{PSNR}(o, \hat{o}) = 20 \log_{10}(L) - 10 \log_{10} \left(\sum_{u=1}^{h_\mathrm{o}} \sum_{v=1}^{w_\mathrm{o}} \frac{(o_{uv} - \hat{o}_{uv})^2}{h_\mathrm{o}\, w_\mathrm{o}} \right). \tag{B.12}$$

As we work with normalized images with pixels in the interval $[-1, 1]$, the dynamic range is $L = 2$.

Structural Similarity Index Measure

As simple pixel-by-pixel metrics such as RMSE or PSNR tend to average out any encountered errors, this could lead to a situation in which a significant reconstruction error in a part of the image is not seen in the RMSE metric but has a huge impact on the visual appearance of the reconstruction. SSIM (Wang et al., 2004) incorporates not just the *absolute errors*, but also the strong inter-dependencies between pixels, especially when they are spatially close. The SSIM metric between two observations o and $\hat{o}$ is given by

$$\mathrm{SSIM}(o, \hat{o}) = l^\alpha(o, \hat{o})\, c^\beta(o, \hat{o})\, s^\gamma(o, \hat{o}), \tag{B.13}$$

where

$$l(o, \hat{o}) = \frac{2\mu_o \mu_{\hat{o}} + C_1}{\mu_o^2 + \mu_{\hat{o}}^2 + C_1}, \quad c(o, \hat{o}) = \frac{2\sigma_o \sigma_{\hat{o}} + C_2}{\sigma_o^2 + \sigma_{\hat{o}}^2 + C_2}, \quad s(o, \hat{o}) = \frac{\sigma_{o\hat{o}} + C_3}{\sigma_o \sigma_{\hat{o}} + C_3}. \tag{B.14}$$

We use the constants $C_1 = (k_1 L)^2$, $C_2 = (k_2 L)^2$ and $C_3 = C_2/2$, where L signifies the dynamic range as previously used for the PSNR metric, and $k_1 = 0.01$ and $k_2 = 0.03$. The average μ and the variance σ^2 are computed with a Gaussian filter with a 1D kernel of size 11 and sigma 1.5. We set the weight exponents α, β, and γ for the luminance, contrast, and structure comparisons all to one. We rely on the PIX library (DeepMind et al., 2020) for efficiently computing the SSIM metric.

B.1.6 Compute Resources

We trained the models on several desktop workstations for a total duration of roughly 150 h. In total, we relied on 10x RTX 3090/4090 GPUs, each with 24 GB of VRAM, training the models in parallel. Each workstation contained between 64 and 128 GB of RAM, and we used roughly 100 GB of total storage. Training each model on one random seed took between 45 min and 4 h depending on the model type, the integration time constant, and the number of trainable parameters. The hyperparameter tuning we conducted beforehand (only on one random seed) took roughly the same time and computational resources as generating the final results.

For the control experiments, we additionally used a laptop with a 16-core Intel Core i7-10870H CPU and 32 GB RAM. We did not need to use a GPU for evaluating the model during closed-loop control.

B.2 Appendix on Learning Latent Dynamics

We report the full set of quantitative results, including the additional evaluation metrics PSNR and SSIM, in Tables B.4 to B.7. For *PCC-NS-2* in Tab. B.5, we additionally recorded the training steps per second on an Nvidia RTX 3090 GPU with a batch size of 80 (which leads to 8080 images per batch). We plot the results of a sweep across the latent dimensions for the additional evaluation metrics PSNR and SSIM in Fig. B.1. Correspondingly, we visualize the number of trainable parameters of each model vs. the latent dimension in Fig. B.2. In Figs. B.3- B.5, we present sequences of stills for the rollout of the trained *CON*(-M) models on the various datasets.

Model	RMSE ↓	PSNR ↑	SSIM ↑	# Params. ↓
RNN	0.2739 ± 0.0057	4.16 ± 0.02	0.6958 ± 0.0122	88
GRU (Cho et al., 2014)	0.0267 ± 0.0033	$\mathbf{6.13 \pm 0.09}$	0.9861 ± 0.0022	248
coRNN (Rusch and Mishra, 2020)	$\mathbf{0.0265 \pm 0.0002}$	$\mathbf{6.13 \pm 0.01}$	$\mathbf{0.9853 \pm 0.0006}$	40
NODE (Chen et al., 2018)	$\mathbf{0.0264 \pm 0.0010}$	$\mathbf{6.14 \pm 0.03}$	$\mathbf{0.9858 \pm 0.0009}$	3368
MECH-NODE	0.0328 ± 0.0034	5.99 ± 0.07	0.9821 ± 0.0024	3244
CON (our)	0.0303 ± 0.0053	6.05 ± 0.13	0.9847 ± 0.0027	**34**
CFA-CON (our)	0.0313 ± 0.0026	6.02 ± 0.06	0.9843 ± 0.0008	**34**

Table B.1: Benchmarking of CON and CFA-CON at learning latent dynamics on the ***M-SP+F* (mass-spring with friction) dataset**. For all models, a latent dimension of $n_z = 4$ is chosen. As this dataset does not consider any inputs, we remove all parameters in the RNN, GRU, coRNN, CON, and CFA-CON models related to the input mapping. *MECH-NODE* is a NODE with prior knowledge about the mechanical structure of the system (i.e., $\frac{\mathrm{d}x}{\mathrm{d}t} = \dot{x}$). We report the mean and standard deviation over three different random seeds and the number of parameters of each latent dynamics model.

Model	RMSE ↓	PSNR ↑	SSIM ↑	# Params. ↓
RNN	0.2378 ± 0.0352	4.31 ± 0.15	0.7568 ± 0.0350	88
GRU (Cho et al., 2014)	0.1457 ± 0.0078	4.78 ± 0.05	0.9168 ± 0.0093	248
coRNN (Rusch and Mishra, 2020)	0.1333 ± 0.0044	4.86 ± 0.03	0.9194 ± 0.0055	40
NODE (Chen et al., 2018)	$\mathbf{0.1260 \pm 0.0013}$	$\mathbf{4.91 \pm 0.01}$	$\mathbf{0.9379 \pm 0.0009}$	3368
MECH-NODE	0.1650 ± 0.0205	4.67 ± 0.12	0.8985 ± 0.0153	3244
CON (our)	0.1303 ± 0.0064	4.88 ± 0.04	0.9175 ± 0.0095	**34**
CFA-CON (our)	0.1352 ± 0.0073	4.85 ± 0.05	0.9133 ± 0.0052	**34**

Table B.2: Benchmarking of CON and CFA-CON at learning latent dynamics on the ***S-P+F* (single pendulum with friction) dataset**. For all models, a latent dimension of $n_z = 4$ is chosen. As this dataset does not consider any inputs, we remove all parameters in the RNN, GRU, coRNN, CON, and CFA-CON models related to the input mapping. *MECH-NODE* is a NODE with prior knowledge about the mechanical structure of the system (i.e., $\frac{\mathrm{d}x}{\mathrm{d}t} = \dot{x}$). We report the mean and standard deviation over three different random seeds and the number of parameters of each latent dynamics model.

Model	RMSE ↓	PSNR ↑	SSIM ↑	# Params. ↓
RNN	0.1694 ± 0.0004	4.631 ± 0.002	0.7082 ± 0.0032	672
GRU (Cho et al., 2014)	0.1329 ± 0.0005	4.858 ± 0.003	$\mathbf{0.8340 \pm 0.0021}$	1968
coRNN (Rusch and Mishra, 2020)	0.1324 ± 0.0016	4.862 ± 0.012	0.8229 ± 0.0039	348
NODE (Chen et al., 2018)	0.1324 ± 0.0024	4.861 ± 0.016	0.8101 ± 0.0024	4404
MECH-NODE	0.1710 ± 0.0111	4.624 ± 0.063	0.7170 ± 0.0439	4032
CON (our)	0.1323 ± 0.0018	4.862 ± 0.013	0.8067 ± 0.0038	**246**
CFA-CON (our)	$\mathbf{0.1307 \pm 0.0012}$	$\mathbf{4.873 \pm 0.008}$	0.8147 ± 0.0034	**246**

Table B.3: Benchmarking of CON and CFA-CON at learning latent dynamics on the ***D-P+F* (double pendulum with friction) dataset**. For all models, a latent dimension of $n_z = 12$ is chosen. As this dataset does not consider any inputs, we remove all parameters in the RNN, GRU, coRNN, CON, and CFA-CON models related to the input mapping. *MECH-NODE* is a NODE with prior knowledge about the mechanical structure of the system (i.e., $\frac{\mathrm{d}x}{\mathrm{d}t} = \dot{x}$). We report the mean and standard deviation over three different random seeds and the number of parameters of each latent dynamics model.

Model	RMSE ↓	PSNR ↑	SSIM ↑	# Params. ↓
RNN	$\mathbf{0.1011 \pm 0.0009}$	$\mathbf{25.92 \pm 0.08}$	$\mathbf{0.9777 \pm 0.0004}$	696
GRU (Cho et al., 2014)	0.1125 ± 0.0100	24.99 ± 0.74	0.9730 ± 0.0040	2040
coRNN (Rusch and Mishra, 2020)	0.2537 ± 0.0018	17.93 ± 0.06	0.8820 ± 0.0024	**336**
NODE (Chen et al., 2018)	0.2415 ± 0.0021	18.36 ± 0.08	0.8946 ± 0.0023	4374
MECH-NODE	0.2494 ± 0.0028	18.08 ± 0.10	0.8898 ± 0.0016	4002
CON-S (our)	0.1993 ± 0.0646	20.03 ± 2.44	0.9218 ± 0.0380	1386
CON-M (our)	0.1063 ± 0.0027	25.49 ± 0.22	0.9758 ± 0.0011	8568
CFA-CON (our)	0.1462 ± 0.0211	22.72 ± 1.17	0.9573 ± 0.0103	8568

Table B.4: Benchmarking of CON and CFA-CON at learning latent dynamics on the ***CS* (soft robot with one constant strain segment) dataset**. For all models, a latent dimension of $n_z = 12$ is chosen. *CON-S* and *CON-M* are small and medium-sized versions of the CON model, respectively. *MECH-NODE* is a NODE with prior knowledge about the mechanical structure of the system (i.e., $\frac{\mathrm{d}x}{\mathrm{d}t} = \dot{x}$). We report the mean and standard deviation over three different random seeds and the number of parameters of each latent dynamics model.

Model	RMSE ↓	PSNR ↑	SSIM ↑	# Params. ↓	$\frac{\text{Train. steps}}{\text{second}}$ ↑	Inf. time [ms] ↑
RNN	0.1373 ± 0.0185	23.27 ± 1.10	0.9643 ± 0.0077	320	1.87	02.6
GRU (Cho et al., 2014)	0.0951 ± 0.0021	$\mathbf{26.45 \pm 0.19}$	0.9730 ± 0.0040	928	1.83	03.2
coRNN (Rusch and Mishra, 2020)	0.2504 ± 0.0899	18.05 ± 2.66	0.9814 ± 0.0006	**152**	**1.89**	02.7
NODE (Chen et al., 2018)	0.1867 ± 0.0561	20.60 ± 2.28	0.8774 ± 0.0857	3856	0.79	50.2
MECH-NODE	0.1035 ± 0.0012	25.07 ± 0.06	0.9778 ± 0.0004	3062	0.79	50.3
CON-S (our)	0.0996 ± 0.0012	26.05 ± 0.11	$\mathbf{0.9792 \pm 0.0007}$	676	0.78	50.2
CON-M (our)	0.1008 ± 0.0006	25.95 ± 0.05	0.9786 ± 0.0003	7048	0.60	60.1
CFA-CON (our)	0.1124 ± 0.0025	25.01 ± 0.19	0.9734 ± 0.0012	7048	1.12	13.6

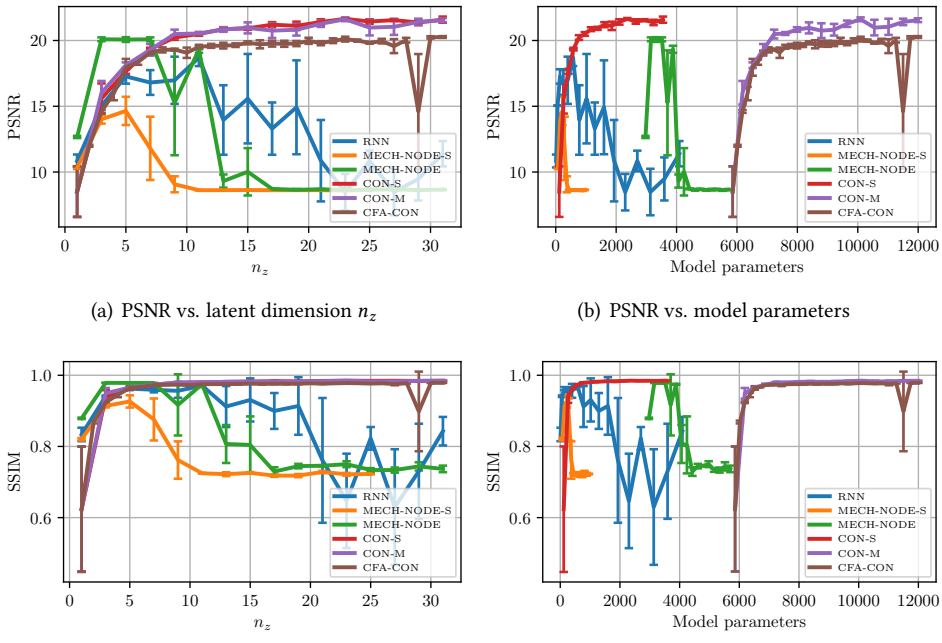
Table B.5: Benchmarking of CON and CFA-CON at learning latent dynamics on the ***PCC-NS-2* (soft robot with two constant curvature segments) dataset**. For all models, a latent dimension of $n_z = 8$ is chosen. *CON-S* and *CON-M* are small and medium-sized versions of the CON model, respectively. *MECH-NODE* is a NODE with prior knowledge about the mechanical structure of the system (i.e., $\frac{\mathrm{d}x}{\mathrm{d}t} = \dot{x}$). We report the mean and standard deviation over three different random seeds. Furthermore, we state the number of parameters of each latent dynamics model and the training steps per second on an Nvidia RTX 3090 GPU. Each batch contains 80 trajectories and 8080 images of resolution 32x32px in total. Finally, we report the inference time averaged over 5000 runs for performing a rollout of 2.02 s (while encoding and decoding all images along the trajectory) on an Nvidia RTX 3090 GPU with a batch size of 1.

Model	RMSE ↓	PSNR ↑	SSIM ↑	# Params. ↓
RNN	0.2232 ± 0.0075	19.05 ± 0.29	0.8955 ± 0.0083	696
GRU (Cho et al., 2014)	0.2148 ± 0.0196	19.38 ± 0.76	0.9039 ± 0.0223	2040
coRNN (Rusch and Mishra, 2020)	0.2474 ± 0.0018	18.15 ± 0.06	0.8877 ± 0.0011	**336**
NODE (Chen et al., 2018)	0.3373 ± 0.0565	15.46 ± 1.34	0.7432 ± 0.0935	4374
MECH-NODE	0.1900 ± 0.0024	20.45 ± 0.11	0.9315 ± 0.0011	4002
CON-S (our)	$\mathbf{0.1792 \pm 0.0038}$	$\mathbf{20.96 \pm 0.18}$	$\mathbf{0.9392 \pm 0.0023}$	1386
CON-M (our)	$\mathbf{0.1785 \pm 0.0023}$	$\mathbf{20.99 \pm 0.11}$	$\mathbf{0.9395 \pm 0.0018}$	8568
CFA-CON (our)	0.1803 ± 0.0003	20.90 ± 0.01	0.9366 ± 0.0004	8568

Table B.6: Benchmarking of CON and CFA-CON at learning latent dynamics on the ***PCC-NS-3* (soft robot with three constant curvature segments) dataset**. For all models, a latent dimension of $n_z = 12$ is chosen. *CON-S* and *CON-M* are small and medium-sized versions of the CON model, respectively. *MECH-NODE* is a NODE with prior knowledge about the mechanical structure of the system (i.e., $\frac{\mathrm{d}x}{\mathrm{d}t} = \dot{x}$). We report the mean and standard deviation over three different random seeds and the number of parameters of each latent dynamics model.

Model	RMSE ↓	PSNR ↑	SSIM ↑	# Params. ↓
RNN	0.3763 ± 0.0374	3.82 ± 0.12	0.4463 ± 0.1358	**20**
GRU (Cho et al., 2014)	0.3232 ± 0.0368	$\mathbf{3.99 \pm 0.13}$	0.6798 ± 0.0949	52
1^{st}-order coRNN (Rusch and Mishra, 2020)	$\mathbf{0.0741 \pm 0.0001}$	$\mathbf{5.35 \pm 0.00}$	$\mathbf{0.9724 \pm 0.0014}$	**20**
NODE (Chen et al., 2018)	$\mathbf{0.0738 \pm 0.0007}$	$\mathbf{5.36 \pm 0.01}$	$\mathbf{0.9683 \pm 0.0022}$	3064
CON (our)	0.1110 ± 0.0160	5.03 ± 0.12	0.9372 ± 0.0109	24
CFA-CON (our)	0.1068 ± 0.0059	5.05 ± 0.05	0.9418 ± 0.0026	24

Table B.7: Benchmarking of CON and CFA-CON at learning latent dynamics on the ***R-D* (reaction-diffusion) dataset**. For all models, a latent dimension of $n_z = 4$ is chosen. As this dataset does not consider inputs, we remove all parameters in the RNN, GRU, coRNN, CON, and CFA-CON models related to the input mapping. Also, as the *reaction-diffusion* system is governed by 1^{st}-order PDE dynamics, we use specialized, 1^{st}-order version of the *CON*, *CFA-CON*, and *coRNN* dynamics. We report the mean and standard deviation over three different random seeds and the number of parameters of each latent dynamics model.



(a) PSNR vs. latent dimension n_z

(b) PSNR vs. model parameters

(c) SSIM vs. latent dimension n_z

(d) SSIM vs. model parameters

Figure B.1: Evaluation of prediction performance of the various models vs. the dimension of their latent representation n_z and the number of trainable parameters of the dynamics model, respectively. We optimize the hyperparameters for the case of $n_z = 8$, and execute the tuning separately for each model and dataset.

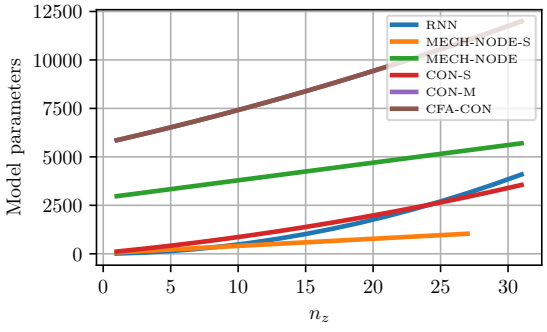
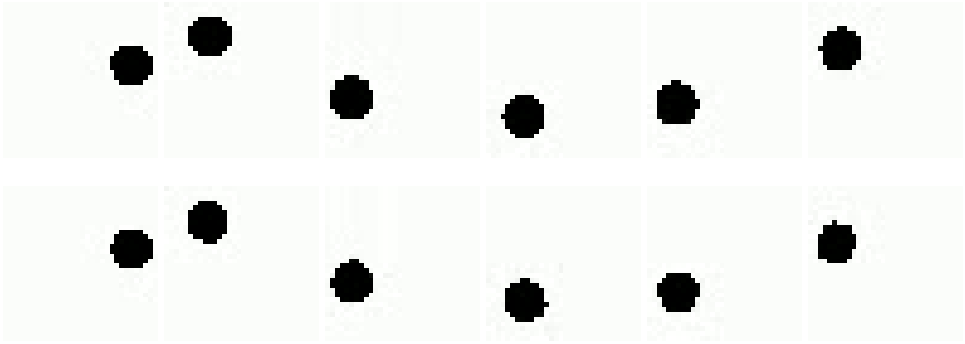
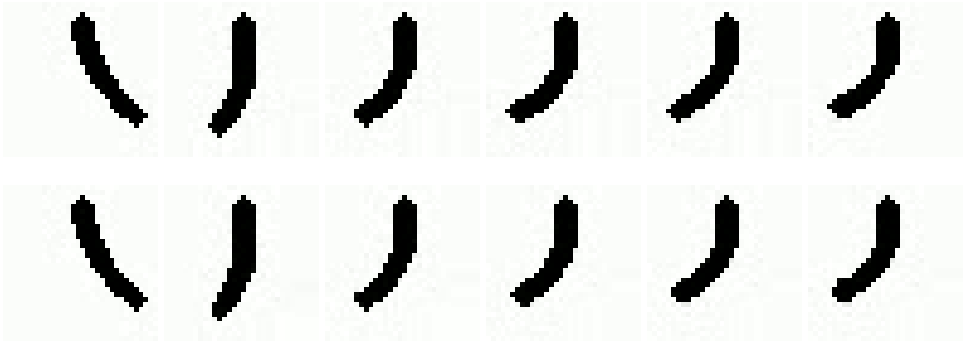


Figure B.2: Plot of number of trainable parameters vs. the latent dimension n_z of various models trained on the *PCC-NS-2* dataset. As we have configured them, *CON-M* and *CFA-CON* always have the same number of parameters (i.e., overlaying lines).



(a) t=0.0 s (b) t=0.55 s (c) t=1.10 s (d) t=1.65 s (e) t=2.20 s (f) t=2.75 s

Figure B.3: Prediction sequence of a CON model with latent dimension $n_z = 4$ trained on the single pendulum with friction (*S-P+F*) dataset (Botev et al., 2021). **Top row:** Ground-truth evolution of the system. **Bottom row:** Predictions of the *CON* model.
The prediction model is given three images centered around $t = 0$ for encoding the initial latent $z(0)$ and estimation of the initial latent velocity $\dot{z}(0)$. Subsequently, we roll out the autonomous network dynamics (i.e., unforced) and compare the decoded predictions with the ground-truth evolution of the system.



(a) t=0.00 s (b) t=0.04 s (c) t=0.08 s (d) t=0.12 s (e) t=0.16 s (f) t=0.20 s

Figure B.4: Prediction sequence of a forced CON model with latent dimension $n_z = 12$ trained on the soft robotic *CS* dataset containing trajectories of a simulated constant strain robot with one segment. **Top row:** Ground-truth evolution of the system. **Bottom row:** Predictions of the *CON-M* model.
The prediction model is given three images centered around $t = 0$ for encoding the initial latent $z(0)$ and estimation of the initial latent velocity $\dot{z}(0)$. Subsequently, we roll out the autonomous network dynamics (i.e., unforced) and compare the decoded predictions with the ground-truth evolution of the system.

(a) t=0.0 s (b) t=0.3 s (c) t=0.6 s (d) t=0.9 s (e) t=1.2 s

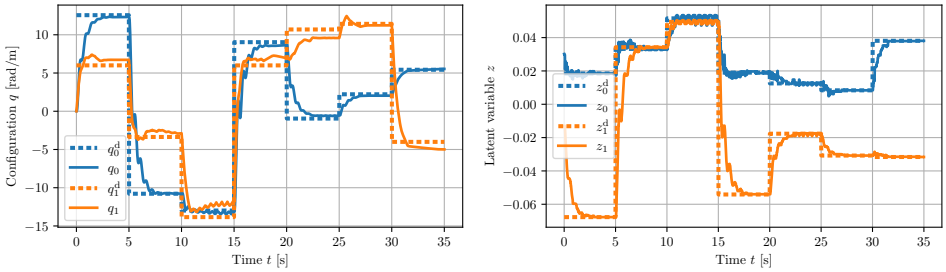
Figure B.5: Prediction sequence of an unforced CON model with latent dimension $n_z = 8$ trained on the *PCC-NS-2* dataset. **Top row:** Ground-truth evolution of the system. **Bottom row:** Predictions of the *CON-M* model. The prediction model is given three images centered around $t = 0$ for encoding the initial latent $z(0)$ and estimation of the initial latent velocity $\dot{z}(0)$. Subsequently, we roll out the autonomous network dynamics (i.e., unforced) and compare the decoded predictions with the ground-truth evolution of the system.

(a) t=0.0 s (b) t=0.3 s (c) t=0.6 s (d) t=0.9 s (e) t=1.2 s

Figure B.6: Prediction sequence of a forced CON model with latent dimension $n_z = 8$ trained on the *PCC-NS-2* dataset. **Top row:** Ground-truth evolution of the system. **Bottom row:** Predictions of the *CON-M* model. The prediction model is given three images centered around $t = 0$ for encoding the initial latent $z(0)$ and estimation of the initial latent velocity $\dot{z}(0)$. Subsequently, we provide the same constant input u to both the simulator and the network dynamics (i.e., unforced) and compare the decoded predictions with the ground-truth evolution of the system.

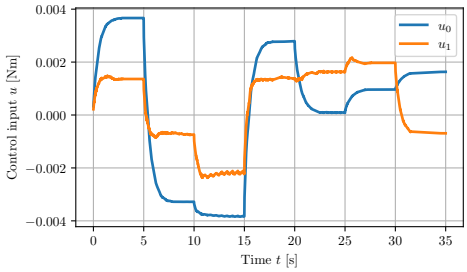
B.3 Appendix on Latent Space Control

Below, we present supplementary results for the model-based, latent-space control on the two-segment PCC soft robot (*PCC-NS-2*).



(a) Configuration $q(t) \in \mathbb{R}^2$

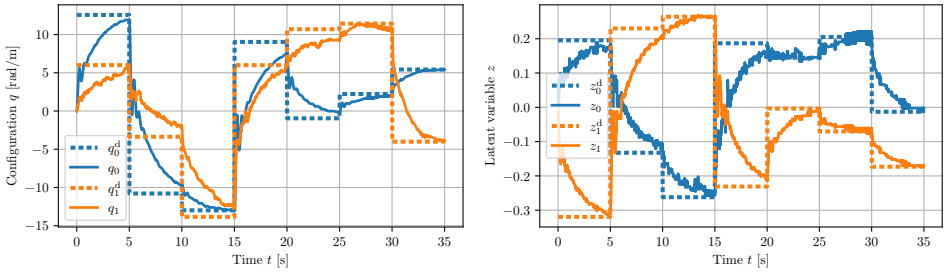
(b) Latent representation $z(t) \in \mathbb{R}^2$



(c) Control input $u(t) \in \mathbb{R}^2$

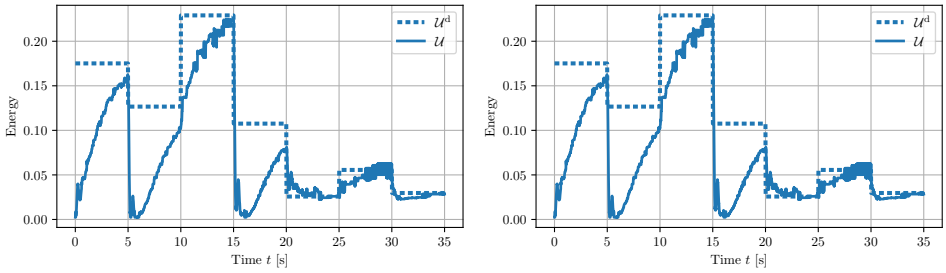
Figure B.7: Latent-space control of a continuum soft robot (simulated using two PCC segments) following a sequence of setpoints with a pure **P-satI-D** feedback controller operating in a 2D latent space learned with the **MECH-NODE** model. The CON model weights are initialized using a **random seed of 0**. The dotted and solid lines show the reference and actual values, respectively. For each setpoint, we randomly sample a desired shape q^d and render the corresponding image o^d. This image is then encoded to a target latent z^d. The controller then computes a latent-space torque F^d, which is decoded to an input u. Finally, we provide this input to the simulator, which performs a roll-out of the closed-loop dynamics. Important: The robot's configuration (i.e., the first-principle, minimal-order state) is solely used for generating a target image and simulating the closed-loop system.

B



(a) Configuration $q(t) \in \mathbb{R}^2$

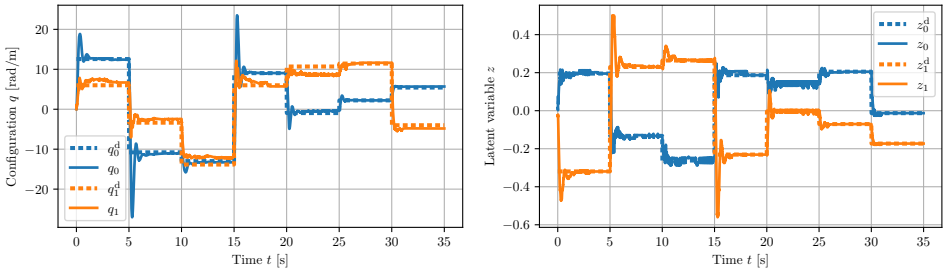
(b) Latent representation $z(t) \in \mathbb{R}^2$



(c) Control input $u(t) \in \mathbb{R}^2$

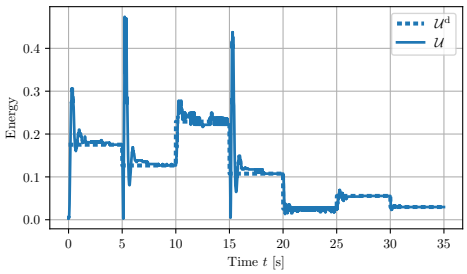
(d) Potential energy $U(t) \in \mathbb{R}$

Figure B.8: Latent-space control of a continuum soft robot (simulated using two PCC segments) following a sequence of setpoints with a pure **P-satI-D** feedback controller operating in a 2D latent space learned with the **CON** model. The CON model weights are initialized using a random seed of 0. The dotted and solid lines show the reference and actual values, respectively. For each setpoint, we randomly sample a desired shape q^d and render the corresponding image o^d. This image is then encoded to a target latent z^d. The controller then computes a latent-space torque F^d, which is decoded to an input u. Finally, we provide this input to the simulator, which performs a roll-out of the closed-loop dynamics. Important: The robot's configuration (i.e., the first-principle, minimal-order state) is solely used for generating a target image and simulating the closed-loop system.

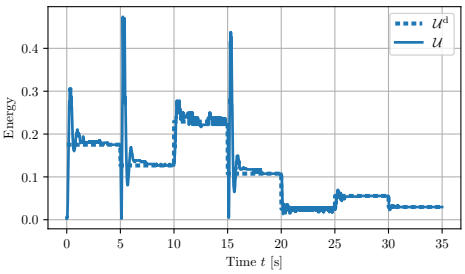


(a) Configuration $q(t) \in \mathbb{R}^2$

(b) Latent representation $z(t) \in \mathbb{R}^2$



(c) Control input $u(t) \in \mathbb{R}^2$



(d) Potential energy $U(t) \in \mathbb{R}$

Figure B.9: Latent-space control of a continuum soft robot (simulated using two PCC segments) following a sequence of setpoints with a pure **P-satI-D+FF** feedback & feedforward controller operating in a 2D latent space learned with the **CON** model. The CON model weights are initialized using a random seed of 0. The dotted and solid lines show the reference and actual values, respectively. For each setpoint, we randomly sample a desired shape q^d and render the corresponding image o^d. This image is then encoded to a target latent z^d. The controller then computes a latent-space torque F^d, which is decoded to an input u. Finally, we provide this input to the simulator, which performs a roll-out of the closed-loop dynamics. Important: The robot's configuration (i.e., the first-principle, minimal-order state) is solely used for generating a target image and simulating the closed-loop system.

C

Software Libraries & Lab Infrastructure

In this appendix, we present multiple software libraries and packages that were developed as part of the Ph.D and are all openly hosted within the `tud-phi` *organization on GitHub* [1]*. Furthermore, we report on essential components of the lab infrastructure that the Ph.D. candidate significantly contributed to. Although these contributions are technical rather than scientific, their development required considerable effort and was crucial to achieving the scientific results presented in this thesis. We believe that sharing these technical resources will be valuable for the community and help advance research in this domain. For instance, we anticipate that the ROS2 package for Optitrack MCS, along with the Festo pressure regulator documentation—including its pneumatic circuitry schematic and ROS2 drivers—will be valuable for soft robotics practitioners setting up their own experimental systems. Additionally, our JAX implementation of soft robotic dynamic models marks a step toward modernizing the algorithmic software stack, enabling researchers to leverage techniques such as automatic differentiation, Just-In-Time (JIT)-compilation, and Graphics Processing Unit (GPU) deployment, which are essential for integrating these models into ML approaches. Finally, our fully-fledged HSA software ecosystem will enable the community to rapidly bootstrap their HSA robot control research and inspire the creation of similar ROS2 packages for other soft robots.*

[1] `https://github.com/tud-phi`

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q} = \mathcal{Q}$$

This appendix is organized as follows: First, we detail the setup of our motion capture system used to acquire ground-truth data on the shape evolution of soft robots. Next, we describe the configuration and interfacing with a Festo pressure regulator that actuates pneumatic soft robots. Third, we document the fabrication process for a multi-segment pneumatic soft arm made of silicon. Fourth, we introduce the *JSRM* package, which implements control-oriented soft robot models in JAX. Finally, we provide an overview of the ROS2 ecosystem developed for the control and operation of HSA robots.

C

C.1 Motion Capture System

Our soft robot experiments demand precise state information. For example, tasks like closed-loop control (as demonstrated in Chapter 6) and the quantitative evaluation of proprioceptive shape sensing approaches (as outlined in Chapters 4 and 9) rely on accurate soft robot backbone shape information. In line with common practice in both research and industry, we use an Optitrack MCS system equipped with eight Infrared (IR)-illuminated cameras (Optitrack Prime X13). To ensure that our trials are reproducible while minimizing setup effort, we built a cubical cage with a 750 mm side length using standard aluminum T-beams. The cameras are flexibly mounted to the beams with clamps, which keep their relative positions fixed and eliminate the need for frequent, time-consuming recalibrations whenever a camera's pose might change.

To enable closed-loop control using the MCS measurements, we implemented a ROS2 package built on the NatNet SDK. This package, available on GitHub[2], streams the poses of tracked rigid bodies from a Windows workstation running the Optitrack Motive software and publishes them as ROS messages. Subsequently, the package transforms the poses from the MoCap world frame to a soft robot base coordinate system, with the origin at the robot's base and the z-axis aligned with the backbone direction.

[2] `https://github.com/tud-phi/ros2-mocap_optitrack`

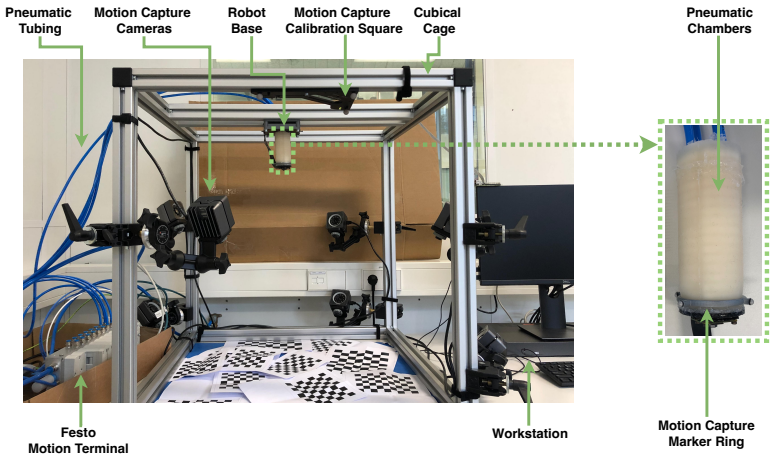


Figure C.1: Experimental Setup with a Pneumatic Soft Robot Segment: A soft robotic segment is attached to the top of a cage equipped with motion capture cameras. Pneumatic actuation is achieved using a pressure regulator (Festo VTEM Motion Terminal), while a workstation manages communication via Ethernet with both the motion capture system and the pressure regulator.

C.2 Pneumatic Pressure Regulator

In this thesis, we consider at various times (e.g., Chapters 4, 2, 9) pneumatically-actuated robots. Two primary methods exist for implementing such actuation. The first involves fluidic drive cylinders, where the robot's air chambers are connected to an array of cylinders (Marchese et al., 2014; Taheri et al., 2014). In this setup, moving the pistons adjusts the volume of the closed system, causing the pressure to change inversely according to the ideal gas law (Stölzle and Della Santina, 2021) (see Chapter 8). Alternatively, pneumatic valves can be used to manage airflow: they control the entry of air from a high-pressure source into the chamber and facilitate pressure release by allowing air to escape to the atmosphere (Falkenhahn et al., 2016).

We opted for the valve-based approach and selected Festo's *Motion Terminal VTEM*[3], which can regulate airflow into 16 independent chambers. Although pressure regulation is our primary focus, the VTEM also supports flow control and additional tasks. To connect the motion terminal with the air supply unit, we developed a sophisticated pneumatic circuitry (see Fig. C.2). Pressure set-points are transmitted via Modbus / Transmission Control Protocol (TCP) over a Local Area Network (LAN), and the regulator's built-in low-level PID controller adjusts the valve positions to achieve these targets. Simultaneously, current pressure readings from the valves are made available for purposes such as system identification. For this communication, we developed drivers in both C++ and MATLAB based on the VTEM protocol[4], which are distributed together with a ROS2 package on GitHub[5]. These tools allow users to both publish pressure commands and subscribe to the

[3] https://www.festo.com/de/en/p/motion-terminal-id_VTEM/
[4] https://www.festo.com/net/SupportPortal/Files/684271/VTEM_PLC_QuickReference_V1-4.pdf
[5] https://github.com/tud-phi/ros2-vtem_control

valve pressure data.

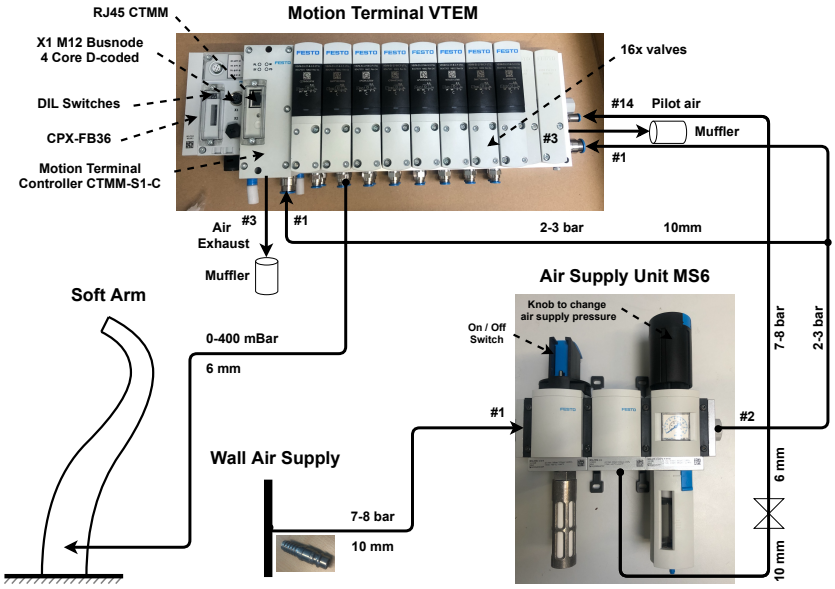


Figure C.2: Pneumatic Circuitry: The air supply unit reduces the input pressure to approximately 3–4 bars, which is then fed into the VTEM pressure regulator through two ports. In parallel, a higher-pressure pilot air (around 7–8 bars) is provided by the same unit to actuate the VTEM's valves. This configuration enables independent pressure control across 16 chambers. Moreover, a workstation can interface with the system via Modbus/TCP, allowing it to monitor the current valve pressures and update the pressure set points for the regulator's low-level PID controller.

C.3 Pneumatic Soft Robot Arm

To kick off PhI-Lab's experimental soft robotics research, we adopted a pneumatic soft robotic arm made up of a chain of identical segments. Each segment contains three to four inflatable cavities, evenly distributed along the radial direction from its centerline (Marchese et al., 2015). By controlling the air pressure within these cavities, the segment can extend and bend in three-dimensional Cartesian space.

The fabrication process of the silicone body is quite intricate. It begins by casting wax negatives for the air chambers—beeswax is poured into a silicon mold to form the initial shape. Once hardened, these wax inserts are placed into a 3D-printed mold designed for the segment. After mixing and degassing the silicone in a vacuum chamber, it is slowly poured into the mold. Following curing, the segment is removed and then heated in an oven to melt away the wax, thereby leaving behind the inflatable chambers (Marchese et al., 2015). Although the CAD design was already available, we had to source materials from Dutch suppliers, develop a detailed manufacturing procedure, and learn the practical dos and don'ts of producing these pneumatic soft segments.

Figure C.1 illustrates our laboratory setup for controlling this pneumatic soft robot, as utilized in Chapters 4 and 9.

C.4 An Implementation of Soft Robot Models in JAX

In recent years, several open-source soft robot simulators have emerged—such as SOFA (Coevoet et al., 2017), SimSOFT (Grazioso et al., 2019), ChainQueen (Hu et al., 2019), TMT-Dyn (Sadati et al., 2021), Elastica (Naughton et al., 2021), SoRoSim (Mathew et al., 2022), and Sorotoki (Caasenbrood et al., 2024)—yet control-oriented dynamic models (i.e., low-dimensional, control-affine Euler-Lagrange formulations) based on kinematic models like the PCC (Webster III and Jones, 2010), PCS (Renda et al., 2018), and GVS (Boyer et al., 2020; Renda et al., 2020) remain seldom available. Moreover, since most of these models are implemented in MATLAB, integrating them into ML applications is challenging and prevents the straightforward application of modern techniques offered by Python frameworks (e.g., PyTorch (Paszke et al., 2019), JAX (Bradbury et al., 2018)), such as automatic differentiation, JIT-compilation, and GPU/Tensor Processing Unit (TPU) inference. Consequently, we recognized the need for an open-source implementation of widely adopted dynamic soft robot models within a modern, high-performance Python framework.

We chose to implement the soft robotic models in JAX (Bradbury et al., 2018) for several key reasons. First, JAX provides JIT compilation for fast execution after an initial build phase while allowing for rapid Pythonic development instead of more cumbersome and time-consuming C++ coding. Second, it offers first-class vectorization support, which is invaluable for tasks encountered in this thesis—such as evaluating forward kinematics across many points along the backbone. Third, unlike many ML frameworks that focus on differentiating only the trainable parameters of neural networks, JAX makes it very easy to perform auto differentiation with respect to any function input. This capability enables us to differentiate through the forward kinematic model for iterative inverse kinematics (i.e., the Jacobian inverse method) as shown in Section 5.3 or to implement a latent dynamic model within an approach where we jointly optimize both dynamic and autoencoder parameters via gradient descent, as demonstrated in Chapter 11. Finally, JAX allows us to deploy and parallelize dynamic model rollouts on GPUs and TPUs without any code changes—a significant advantage over existing soft robotic simulators like Elastica (Naughton et al., 2021) or SoRoSim (Mathew et al., 2022), which are limited to CPU execution, unlike modern rigid robotics simulators such as Mujoco's MJX (Todorov et al., 2012) or Nvidia Isaac Sim (Makoviychuk et al., 2021) that support parallel GPU-based simulations.

Below, we provide further details on the implementation. We chose to separate the model derivation and implement it in Sympy (Meurer et al., 2017) for three main reasons: (a) a symbolic derivation of the kinematics and dynamics allows us to inspect and correct the individual terms as needed; (b) several steps in the derivation require integrating terms along the robot or segment length (Della Santina et al., 2023; Renda et al., 2018), and symbolic integration offers a closed-form expression instead of the inherent tradeoff between integration accuracy and computational complexity that numerical integration at runtime exhibits; (c) the *simplify* functions available in most symbolic computing libraries often reduce the computational load significantly compared to a naive algorithmic implementation.

Accordingly, we derive symbolically the forward kinematics $\pi(q, s)$, the Jacobian $J(q, s)$,

the inertia matrix $B(q)$, the Coriolis matrix $C(q, \dot{q})$, the gravitational forces $G(q)$, the kinetic energy $\mathcal{T}(q, \dot{q})$, the potential energy $\mathcal{U}(q)$, and several other terms as outlined in the literature (Della Santina et al., 2023; Stölzle et al., 2024b). We then store these symbolic expressions in binary format (i.e., we pickle them) using the `dill` package. Next, we *lambdify* the expressions into a JAX function that accepts inputs such as the robot parameters (e.g., geometric and material characteristics), the robot state, and, for the forward dynamics, the control input. Crucially, this approach enables automatic differentiation of the kinematics and dynamics with respect to the robot's state $(q, \dot{q})$, the control input τ, and various model parameters, including the robot length L, backbone radius R, and the material's elastic modulus, among others.

In case the model should be used as a simulator, we can formulate an ODE of the dynamics, as referenced in Chapter 2, and integrate numerically in time with an established ODE solver. In JAX (Bradbury et al., 2018), a suitable choice for this task is the Diffrax library (Kidger, 2021), which implements several popular solvers such as forward Euler, Tsitouras' 5/4 method (Tsitouras, 2011), or Dormand-Prince's 5/4 method (Dormand and Prince, 1980).

Currently, our package supports (elastic) N-link pendula, planar soft robots based on the PCS kinematic model, and planar HSA robots. It is available as open-source on GitHub[6]. In the future, we intend to add additional soft robotic models to the package, including 3D PCS (Renda et al., 2018), PAC (Stella et al., 2023c), and GVS (Boyer et al., 2020). Furthermore, we have discovered that the current implementation strategy does not scale well with the number of soft robot segments. In particular, the time required for symbolically integrating the kinematics and dynamic terms along the segment, simplifying these expressions with Sympy (Meurer et al., 2017), and JIT-compiling the (unsimplified) symbolic expressions in JAX (Bradbury et al., 2018) increases exponentially with each additional segment. As a result, it is currently only feasible to simulate soft robots with up to three PCS segments. To address this issue, we plan to eliminate the symbolic derivation steps and perform all necessary computations directly in JAX, including numerical integration along the backbone to compute the mass matrix, among other quantities.

This package has been integral to multiple chapters of this thesis. In Chapter 3, for instance, we derive the soft robotic safety metric—the Soft Robot Injury Severity Criterion (SRISC)—using the PCS model from the package to compute the necessary Jacobians, Cartesian inertias, stiffnesses, and more. In Chapter 5, we validate the package' planar HSA model against experimental data. Chapter 6 leverages the implemented forward and inverse kinematics, along with dynamic terms (e.g., potential forces), within our proposed HSA robot controllers. In Chapter 10, the proposed kinematic fusion algorithm is based on both the inverse and forward kinematics of the planar PCS model. Additionally, we derive the dynamic basis functions using symbolic expressions from Sympy and verify the learned dynamics against the simulated model. Finally, in Chapter 11, we utilize the planar PCS model to generate the soft robot datasets.

[6] `https://github.com/tud-phi/jax-soft-robot-modeling`

C.5 ROS2 Ecosystem for HSA Robots

To facilitate the experimental validation of the HSA robot models, controllers, and BMIs introduced in Chapters 5, 6, and 7, we established a comprehensive ROS2 ecosystem. This open-source framework, available on GitHub[7], comprises the following packages and nodes:

- **Dynamixel Control Package.** Based on the Dynamixel SDK, this package implements a ROS2 service that sends commands to and receives encodings from the four Dynamixel MX-28 motors of the HSA robot. It offers two modes of operation: individual read & write for single motors and synchronized read & write for all four.

- **HSA Actuation Package.** This package features a ROS2 node that publishes the current motor angles at a frequency of 100 Hz and provides a subscriber for sending motor angle setpoints via the Dynamixel control service. Key steps include converting discrete motor positions into continuous angles and mapping planar control inputs (i.e., two positive angles ϕ) into four motor angles and vice-versa while accounting for the handedness of the HSA rods.

- **HSA Inverse Kinematics Package.** Subscribing to the 200 Hz pose messages of the HSA published by the ROS2 Optitrack package (see Section C.1), this node applies the CS inverse kinematics for the planar case as detailed in Chapter 5 and then publishes the resulting configuration of the planar HSA robot.

- **HSA Velocity Estimation Package.** This ROS2 node subscribes to both the MCS pose measurements and the configuration values from the inverse kinematics node, estimating the time derivatives of the pose and configuration at a rate of 200 Hz. For this purpose, we use Scipy's (Virtanen et al., 2020) implementation of the Savitzky-Golay filter (Savitzky and Golay, 1964) with a window length of 20 and a third-order polynomial.

- **HSA Simulation Package.** We wrapped the planar HSA model from Chapter 5 together with the *JAX Soft Robot Modeling* package in a ROS2 node, creating a digital twin for the physical robot. This node subscribes to control inputs (i.e., two HSA rod twist angles in the planar case) and simulates the system at a control frequency of 100 Hz with a simulation time step of 0.1 ms, leveraging the Dormand-Prince 5/4 method (Dormand and Prince, 1980; Shampine, 1986) for numerical integration.

- **HSA Visualization Package.** Implemented in OpenCV (Bradski, 2000), this package provides a visualization of the kinematic state of planar HSA robots. A ROS2 wrapper node publishes the rendered images at 20 Hz. Additionally, it can optionally display the end-effector workspace, the attractor of the low-level impedance controller, and the setpoint/goal of a control task. This visualization was used to verify both open-loop and closed-loop behaviors in simulation, as illustrated in Fig. 7.5(b) and Chapter 7.

[7] `https://github.com/tud-phi/ros2-hsa`

- **HSA Joy Operation.** To support both keyboard- and BMI-based operation of the HSA robot, we implemented a joy interface. This interface interprets signals (e.g., from keyboard arrows or a joystick, as defined in the ROS2 message `sensor_msg/msg/Joy`) as values in the set $\{-1, 0, 1\}$ along two axes—where $+1$ corresponds to up/right, 0 indicates no input, and -1 corresponds to down/left. These signals are then translated into an incremental movement of the setpoint by 0.1 mm in the planar Cartesian space, with the updated setpoint subsequently used as an attractor by the operational space impedance controller (see Sec. 6.3).

- **HSA Kinematic Control Package.** Serving as a baseline, we implemented a shape controller for HSA robots using a PID law operating on the configuration values.
- **HSA Model-Based Control Package.** The `hsa_planar_control` package[8] comprises several nodes that support model-based control of planar HSA robots:
  - **Calibration Node.** This node identifies the rest strains of the HSA robot based on MCS measurements, configuration estimates, motor actuation angles, and payload mass. Assuming steady-state conditions, it solves the static inversion problem for the rest strain ξ^0 using linear least-squares, as further detailed in Chapter 5.
  - **Model-Based Control Node.** Subscribing to end-effector pose measurements, configuration data, and their time derivatives, this node computes the control input for planar HSA robots. It supports several controllers used in Chapter 6: (1) a basic operational-space PID controller, (2) a configuration-space *P-satI-D+FF* controller that compensates static forces at the desired configuration via a steady-state actuation ϕ^{ss} (see Section 6.2), (3) a configuration-space *P-satI-D+GC* controller that cancels gravitational forces, (4) an operational-space impedance controller (Khatib, 1987) as described in Sec 6.3, (5) an operational-space *PD+* controller (Della Santina et al., 2020c; Ott, 2008; Paden and Panja, 1988) that cancels both gravitational and elastic forces at the current state, among other variations. All controllers run at frequencies between 40 Hz and 50 Hz. Notably, we utilize JAX's JIT functionality to compile the controller at the experiment's outset, enabling these (relatively) high control rates.
  - **Static Planning Node.** This node addresses the static inversion problem by determining a steady-state actuation ϕ^{ss} and a feasible orientation θ^{d} that keep the end-effector at a desired position x^{d}. After applying the closed-form CS-based inverse kinematics, it also computes the corresponding configuration q^{d} for the end-effector pose $\chi^{\text{d}} = \begin{bmatrix} x^{\text{d}} \\ \theta^{\text{d}} \end{bmatrix}$, which then serves as a setpoint for the configuration-space controllers. As described in Sec. 6.2, we implemented two strategies: (a) for FPU (Car, 2020b) material, we solve the static inversion as a root-finding problem—balancing actuation with static forces (elastic and gravitational)—using Scipy's implementation of the Levenberg-Marquardt algorithm (Levenberg, 1944; Marquardt, 1963), and (b) for EPU (Car, 2020a) material, where the force balance problem is ill-conditioned, we employ a method

[8] `https://github.com/tud-phi/hsa-planar-control`

based on forward rollouts of the dynamics (similar to the shooting method in numerical analysis and as proposed by Amehri (2022) for identifying soft robot workspaces). In this approach, the planar HSA robot starts at its rest configuration, the optimizer proposes an actuation ϕ^{ss}, and the dynamics are forward-simulated until steady-state (i.e., until $\dot{q}$ is below a threshold). The steady-state configuration q^{ss} is then used to evaluate the forward kinematics to obtain the end-effector pose χ^{ss}, with the cost function defined as the squared residual between the desired position x^{d} and the achieved position x^{ss}. This nonlinear least-squares problem is solved using Scipy's (Virtanen et al., 2020) Levenberg-Marquardt algorithm (Levenberg, 1944; Marquardt, 1963). Importantly, we JIT-compile the dynamics to expedite the rollouts.

References

References

[Abidi and Cianchetti 2017] Abidi, Haider ; Cianchetti, Matteo: On intrinsic safety of soft robots. In: *Frontiers in Robotics and AI* 4 (2017), S. 5 2, 9, 57, 58, 62, 70, 323

[Abu-Dakka et al. 2024] Abu-Dakka, Fares ; Saveriano, Matteo ; Peternel, Luka: Learning periodic skills for robotic manipulation: Insights on orientation and impedance. In: *Robotics and Autonomous Systems* 180 (2024), S. 104763 261, 263

[Abu-Dakka et al. 2021] Abu-Dakka, Fares J. ; Saveriano, Matteo ; Peternel, Luka: Periodic DMP formulation for quaternion trajectories. In: *2021 20th International Conference on Advanced Robotics (ICAR)* IEEE, 2021, S. 658–663 261

[Ahn et al. 2024] Ahn, Michael ; Dwibedi, Debidatta ; Finn, Chelsea ; Arenas, Montse G. ; Gopalakrishnan, Keerthana ; Hausman, Karol ; Ichter, Brian ; Irpan, Alex ; Joshi, Nikhil ; Julian, Ryan et al.: Autort: Embodied foundation models for large scale orchestration of robotic agents. In: *arXiv preprint arXiv:2401.12963* (2024) 2

[Ajoudani et al. 2018] Ajoudani, Arash ; Zanchettin, Andrea M. ; Ivaldi, Serena ; Albu-Schäffer, Alin ; Kosuge, Kazuhiro ; Khatib, Oussama: Progress and prospects of the human–robot collaboration. In: *Autonomous robots* 42 (2018), S. 957–975 2

[Albu-Schäffer et al. 2007] Albu-Schäffer, Alin ; Haddadin, Sami ; Ott, Christian ; Stemmer, Andreas ; Wimböck, Thomas ; Hirzinger, Gerhard: The DLR lightweight robot: design and control concepts for robots in human environments. In: *Industrial Robot: an international journal* 34 (2007), Nr. 5, S. 376–385 2, 58, 60

[Aldini et al. 2019] Aldini, Stefano ; Akella, Ashlesha ; Singh, Avinash K. ; Wang, Yu-Kai ; Carmichael, Marc ; Liu, Dikai ; Lin, Chin-Teng: Effect of mechanical resistance on cognitive conflict in physical human-robot collaboration. In: *2019 international conference on robotics and automation (ICRA)* IEEE, 2019, S. 6137–6143 139

[Alessi et al. 2024a] Alessi, Carlo ; Agabiti, Camilla ; Caradonna, Daniele ; Laschi, Cecilia ; Renda, Federico ; Falotico, Egidio: Rod models in continuum and soft robot control: a review. In: *arXiv preprint arXiv:2407.05886* (2024) 4, 15, 31

[Alessi et al. 2024b] Alessi, Carlo ; Bianchi, Diego ; Stano, Gianni ; Cianchetti, Matteo ; Falotico, Egidio: Pushing with soft robotic arms via deep reinforcement learning. In: *Advanced Intelligent Systems* 6 (2024), Nr. 8, S. 2300899 4, 5

[Alkayas et al. 2025] Alkayas, Abdulaziz Y. ; Mathew, Anup T. ; Feliu-Talegon, Daniel ; Deng, Ping ; Thuruthel, Thomas G. ; Renda, Federico: Soft Synergies: Model Order Reduction of Hybrid Soft-Rigid Robots via Optimal Strain Parameterization. In: *IEEE Transactions on Robotics* (2025), S. 1–20. http://dx.doi.org/10.1109/TRO.2024.3522182. – DOI 10.1109/TRO.2024.3522182. – Accepted 195, 196, 324, 332, 333, 334, 340

[Almanzor et al. 2023] Almanzor, Elijah ; Ye, Fan ; Shi, Jialei ; Thuruthel, Thomas G. ; Wurdemann, Helge A. ; Iida, Fumiya: Static shape control of soft continuum robots using deep visual inverse kinematic models. In: *IEEE Transactions on Robotics* (2023) 242

[Alora et al. 2023a] Alora, John I. ; Cenedese, Mattia ; Schmerling, Edward ; Haller, George ; Pavone, Marco: Data-driven spectral submanifold reduction for nonlinear optimal control of high-dimensional robots. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)* IEEE, 2023, S. 2627–2633 5, 6, 195, 222, 324, 332, 333, 334, 340

[Alora et al. 2023b] Alora, John I. ; Pabon, Luis A. ; Köhler, Johannes ; Cenedese, Mattia ; Schmerling, Ed ; Zeilinger, Melanie N. ; Haller, George ; Pavone, Marco: Robust Nonlinear Reduced-Order Model Predictive Control. In: *2023 62nd IEEE Conference on Decision and Control (CDC)* IEEE, 2023, S. 4798–4805 5, 6, 11, 221

[Amehri 2022] Amehri, Walid: *Workspace Estimation and Design Optimization of Soft Robots*, Centrale Lille Institut, Diss., 2022 334, 367

[Ames et al. 2016] Ames, Aaron D. ; Xu, Xiangru ; Grizzle, Jessy W. ; Tabuada, Paulo: Control barrier function based quadratic programs for safety critical systems. In: *IEEE Transactions on Automatic Control* 62 (2016), Nr. 8, S. 3861–3876 2, 59, 62, 76, 312, 313

[Angeli et al. 2025] Angeli, David ; Martini, Davide ; Innocenti, Giacomo ; Tesi, Alberto: An LMI formulation of small-gain theorems for 2-contraction of nonlinear interconnected systems. In: *IEEE Transactions on Automatic Control* (2025) 300

[Aouaj et al. 2021] Aouaj, Ilias ; Padois, Vincent ; Saccon, Alessandro: Predicting the post-impact velocity of a robotic arm via rigid multibody models: An experimental study. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)* IEEE, 2021, S. 2264–2271 65

[Aracri et al. 2024] Aracri, Simona ; Hughes, Josie ; Della Santina, Cosimo ; Jovanova, Jovana ; Hoh, Sam ; Garcia Morales, Ditzia S. ; Barcaro, Rosangela ; Tan, Yu J. ; Kortman, Vera G. ; Sakes, Aimée et al.: Soft Robotics: A Route to Equality, Diversity, and Inclusivity in Robotics. In: *Soft Robotics* (2024) 195

[Armanini et al. 2017] Armanini, C ; Dal Corso, F ; Misseroni, D ; Bigoni, D: From the elastica compass to the elastica catapult: an essay on the mechanics of soft robot arm. In: *Proc. R. Soc. A* Bd. 473 The Royal Society, 2017, S. 20160870 4

[Armanini et al. 2023] Armanini, Costanza ; Boyer, Frédéric ; Mathew, Anup T. ; Duriez, Christian ; Renda, Federico: Soft robots modeling: A structured overview. In: *IEEE*

Transactions on Robotics 39 (2023), Nr. 3, S. 1728–1748 3, 4, 10, 25, 26, 31, 62, 64, 65, 99, 174, 195, 201, 242, 316, 324, 332, 333, 334

[Armanini et al. 2021] Armanini, Costanza ; Hussain, Irfan ; Iqbal, Muhammad Z. ; Gan, Dongming ; Prattichizzo, Domenico ; Renda, Federico: Discrete cosserat approach for closed-chain soft robots: Application to the fin-ray finger. In: *IEEE Transactions on Robotics* 37 (2021), Nr. 6, S. 2083–2098 101, 314

[Arpaia et al. 2022] Arpaia, Pasquale ; Coyle, Damien ; Donnarumma, Francesco ; Esposito, Antonio ; Natalizio, Angela ; Parvis, Marco: Non-immersive versus immersive extended reality for motor imagery neurofeedback within a brain-computer interfaces. In: *International Conference on Extended Reality* Springer, 2022, S. 407–419 17, 139

[Asimov 1941] Asimov, Isaac: Three laws of robotics. In: *Asimov, I. Runaround* 2 (1941), Nr. 3 2, 323

[Aswani et al. 2013] Aswani, Anil ; Gonzalez, Humberto ; Sastry, S S. ; Tomlin, Claire: Provably safe and robust learning-based model predictive control. In: *Automatica* 49 (2013), Nr. 5, S. 1216–1226 6

[Aubin et al. 2022] Aubin, Cameron A. ; Gorissen, Benjamin ; Milana, Edoardo ; Buskohl, Philip R. ; Lazarus, Nathan ; Slipher, Geoffrey A. ; Keplinger, Christoph ; Bongard, Josh ; Iida, Fumiya ; Lewis, Jennifer A. et al.: Towards enduring autonomous robots via embodied energy. In: *Nature* 602 (2022), Nr. 7897, S. 393–402 3

[Ba et al. 2016] Ba, Jimmy L. ; Kiros, Jamie R. ; Hinton, Geoffrey E.: Layer normalization. In: *arXiv preprint arXiv:1607.06450* (2016) 344

[Baaij et al. 2023] Baaij, Thomas ; Holkenborg, Marn K. ; Stölzle, Maximilian ; Tuin, Daan van d. ; Naaktgeboren, Jonatan ; Babuška, Robert ; Della Santina, Cosimo: Learning 3D shape proprioception for continuum soft robots with multiple magnetic sensors. In: *Soft Matter* 19 (2023), Nr. 1, S. 44–56 31, 172, 430, 433

[Bächer et al. 2021] Bächer, Moritz ; Knoop, Espen ; Schumacher, Christian: Design and control of soft robots using differentiable simulation. In: *Current Robotics Reports* 2 (2021), Nr. 2, S. 211–221 329, 332, 334

[Bahl et al. 2020] Bahl, Shikhar ; Mukadam, Mustafa ; Gupta, Abhinav ; Pathak, Deepak: Neural dynamic policies for end-to-end sensorimotor learning. In: *Advances in Neural Information Processing Systems* 33 (2020), S. 5058–5069 261

[Baines et al. 2024] Baines, Robert ; Shah, Dylan ; Marvel, Jeremy ; Case, Jennifer ; Spielberg, Andrew: The need for reproducible research in soft robotics. In: *Nature Machine Intelligence* (2024), S. 1–2 325, 330, 339

[Behrens et al. 2022] Behrens, Roland ; Pliske, G ; Umbreit, M ; Piatek, S ; Walcher, F ; Elkmann, Norbert: A statistical model to determine biomechanical limits for physically safe interactions with collaborative robots. In: *Frontiers in Robotics and AI* 8 (2022), S. 667818 76

[Beik-Mohammadi et al. 2024] Beik-Mohammadi, Hadi ; Hauberg, Søren ; Arvanitidis, Georgios ; Figueroa, Nadia ; Neumann, Gerhard ; Rozo, Leonel: Neural Contractive Dynamical Systems. In: *The Twelfth International Conference on Learning Representations*, 2024 279

[Bern et al. 2020] Bern, James M. ; Schnider, Yannick ; Banzet, Pol ; Kumar, Nitish ; Coros, Stelian: Soft robot control with a learned differentiable model. In: *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)* IEEE, 2020, S. 417–423 340

[Bertino et al. 2023] Bertino, Alexander ; Naseradinmousavi, Peiman ; Krstić, Miroslav: Prescribed-time safety filter for a 7-DOF robot manipulator: experiment and design. In: *IEEE Transactions on Control Systems Technology* 31 (2023), Nr. 4, S. 1762–1773 58, 59, 62

[Besl and McKay 1992] Besl, Paul J. ; McKay, Neil D.: A Method for Registration of 3-D Shapes. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14 (1992), Nr. 2, S. 239–256. http://dx.doi.org/10.1109/34.121791. – DOI 10.1109/34.121791 285

[Bhatia et al. 2021] Bhatia, Jagdeep ; Jackson, Holly ; Tian, Yunsheng ; Xu, Jie ; Matusik, Wojciech: Evolution gym: A large-scale benchmark for evolving soft robots. In: *Advances in Neural Information Processing Systems* 34 (2021), S. 2201–2214 316, 323, 328, 329, 332, 333, 334, 336, 339

[Bhattacharyya et al. 2017] Bhattacharyya, Saugat ; Konar, Amit ; Tibarewala, DN: Motor imagery and error related potential induced position control of a robotic arm. In: *IEEE/CAA Journal of Automatica Sinica* 4 (2017), Nr. 4, S. 639–650 139

[Bicchi et al. 2008] Bicchi, Antonio ; Bavaro, Michele ; Boccadamo, Gianluca ; De Carli, Davide ; Filippini, Roberto ; Grioli, Giorgio ; Piccigallo, Marco ; Rosi, Alessandro ; Schiavi, Riccardo ; Sen, Soumen et al.: Physical human-robot interaction: Dependability, safety, and performance. In: *2008 10th IEEE International Workshop on Advanced Motion Control* IEEE, 2008, S. 9–14 60

[Bicchi and Tonietti 2004] Bicchi, Antonio ; Tonietti, Giovanni: Fast and" soft-arm" tactics [robot arm design]. In: *IEEE Robotics & Automation Magazine* 11 (2004), Nr. 2, S. 22–33 58, 60, 61

[Bischoff et al. 2010] Bischoff, Rainer ; Kurth, Johannes ; Schreiber, Günter ; Koeppe, Ralf ; Albu-Schäffer, Alin ; Beyer, Alexander ; Eiberger, Oliver ; Haddadin, Sami ; Stemmer, Andreas ; Grunwald, Gerhard et al.: The KUKA-DLR Lightweight Robot arm-a new reference platform for robotics research and manufacturing. In: *ISR 2010 (41st international symposium on robotics) and ROBOTIK 2010 (6th German conference on robotics)* VDE, 2010, S. 1–8 58, 60

[Black et al. 2024] Black, Kevin ; Brown, Noah ; Driess, Danny ; Esmail, Adnan ; Equi, Michael ; Finn, Chelsea ; Fusai, Niccolo ; Groom, Lachy ; Hausman, Karol ; Ichter, Brian et al.: π_0: A Vision-Language-Action Flow Model for General Robot Control. In: *arXiv preprint arXiv:2410.24164* (2024) 261, 294, 302, 317

[Bloch et al. 2001] Bloch, Anthony M. ; Chang, Dong E. ; Leonard, Naomi E. ; Marsden, Jerrold E.: Controlled Lagrangians and the stabilization of mechanical systems. II. Potential shaping. In: *IEEE Transactions on Automatic Control* 46 (2001), Nr. 10, S. 1556–1571 221, 222, 248

[Blondel et al. 2021] Blondel, Mathieu ; Berthet, Quentin ; Cuturi, Marco ; Frostig, Roy ; Hoyer, Stephan ; Llinares-López, Felipe ; Pedregosa, Fabian ; Vert, Jean-Philippe: Efficient and Modular Implicit Differentiation. In: *arXiv preprint arXiv:2105.15183* (2021) 133

[van Boeijen et al. 2020] Boeijen, Annemiek G. ; Daalhuizen, Jaap ; Zijlstra, Jelle: *Delft design guide: Perspectives, models, approaches, methods.* bis Publishers, 2020 323, 327, 330

[Boekel 2025] Boekel, Fabio: *Learning-based Co-design for Bio-inspired Quadrupeds.* Delft, The Netherlands, Delft University of Technology, Master Thesis, Jan 2025 332

[Borja et al. 2022] Borja, Pablo ; Dabiri, Azita ; Della Santina, Cosimo: Energy-based shape regulation of soft robots with unactuated dynamics dominated by elasticity. In: *2022 IEEE 5th international conference on soft robotics (RoboSoft)* IEEE, 2022, S. 396–402 16, 37, 42, 43

[Botev et al. 2021] Botev, Aleksandar ; Jaegle, Andrew ; Wirnsberger, Peter ; Hennes, Daniel ; Higgins, Irina: Which priors matter? Benchmarking models for learning latent dynamics. In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021 221, 239, 240, 242, 246, 346, 353

[Bouc 1967] Bouc, R: Forced vibrations of mechanical systems with hysteresis. In: *Proc. of the Fourth Conference on Nonlinear Oscillations, Prague, 1967*, 1967 315

[Box 1976] Box, George E.: Science and statistics. In: *Journal of the American Statistical Association* 71 (1976), Nr. 356, S. 791–799 39

[Boyd and Vandenberghe 2004] Boyd, Stephen P. ; Vandenberghe, Lieven: *Convex optimization.* Cambridge university press, 2004 228, 229

[Boyer et al. 2020] Boyer, Frédéric ; Lebastard, Vincent ; Candelier, Fabien ; Renda, Federico: Dynamics of continuum and soft robots: A strain parameterization based approach. In: *IEEE Transactions on Robotics* 37 (2020), Nr. 3, S. 847–863 3, 30, 155, 156, 162, 195, 205, 363, 364

[Boyer et al. 2006] Boyer, Frédéric ; Porez, Mathieu ; Khalil, Wisama: Macro-continuous computed torque algorithm for a three-dimensional eel-like robot. In: *IEEE transactions on robotics* 22 (2006), Nr. 4, S. 763–775 51

[Bradbury et al. 2018] Bradbury, James ; Frostig, Roy ; Hawkins, Peter ; Johnson, Matthew J. ; Leary, Chris ; Maclaurin, Dougal ; Necula, George ; Paszke, Adam ; VanderPlas, Jake ; Wanderman-Milne, Skye ; Zhang, Qiao: *JAX: composable transformations of Python+NumPy programs.* 2018 71, 76, 243, 346, 363, 364

[Bradski 2000] Bradski, G.: The OpenCV Library. In: *Dr. Dobb's Journal of Software Tools* (2000) 243, 280, 365

[Breazeal et al. 2016] Breazeal, Cynthia ; Dautenhahn, Kerstin ; Kanda, Takayuki: Social robotics. In: *Springer handbook of robotics* (2016), S. 1935–1972 313

[Bristow et al. 2006] Bristow, Douglas A. ; Tharayil, Marina ; Alleyne, Andrew G.: A survey of iterative learning control. In: *IEEE control systems magazine* 26 (2006), Nr. 3, S. 96–114 5

[Brown et al. 2020] Brown, Tom B. ; Mann, Benjamin ; Ryder, Nick ; Subbiah, Melanie ; Kaplan, Jared ; Dhariwal, Prafulla ; Neelakantan, Arvind ; Shyam, Pranav ; Sastry, Girish ; Askell, Amanda et al.: Language Models are Few-Shot Learners. In: *Advances in Neural Information Processing Systems* 33 (2020), S. 1877–1901 342

[Bruder et al. 2024] Bruder, Daniel ; Bombara, David ; Wood, Robert J.: A Koopman-based residual modeling approach for the control of a soft robot arm. In: *The International Journal of Robotics Research* (2024), S. 02783649241272114 5, 6, 195

[Bruder et al. 2020] Bruder, Daniel ; Fu, Xun ; Gillespie, R B. ; Remy, C D. ; Vasudevan, Ram: Data-driven control of soft robots using Koopman operator theory. In: *IEEE Transactions on Robotics* 37 (2020), Nr. 3, S. 948–961 5, 6, 11, 195, 332, 333

[Brunton et al. 2016] Brunton, Steven L. ; Brunton, Bingni W. ; Proctor, Joshua L. ; Kutz, J N.: Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. In: *PloS one* 11 (2016), Nr. 2, S. e0150171 221

[Buriani et al. 2025] Buriani, Gioele ; Liu, Jingyue ; Stölzle, Maximilian ; Della Santina, Cosimo ; Ding, Jiatao: Symbolic Learning of Interpretable Reduced-order Models for Jumping Legged Robots. (2025). – Under review 433

[Caasenbrood and Nijmeijer 2021] Caasenbrood, Brandon ; Nijmeijer, Henk: Energy-based Control for Soft Manipulators using Cosserat-beam Models. In: *ICINCO*, 2021, S. 311–319 4, 37

[Caasenbrood et al. 2020] Caasenbrood, Brandon ; Pogromsky, Alexander ; Nijmeijer, Henk: A computational design framework for pressure-driven soft robots through nonlinear topology optimization. In: *2020 3rd IEEE international conference on soft robotics (RoboSoft)* IEEE, 2020, S. 633–638 328, 336

[Caasenbrood et al. 2023] Caasenbrood, Brandon ; Pogromsky, Alexander ; Nijmeijer, Henk: Control-oriented models for hyperelastic soft robots through differential geometry of curves. In: *Soft Robotics* 10 (2023), Nr. 1, S. 129–148 4, 7

[Caasenbrood et al. 2024] Caasenbrood, Brandon J. ; Pogromsky, Alexander Y. ; Nijmeijer, Henk: Sorotoki: A Matlab toolkit for design, modeling, and control of soft robots. In: *IEEE Access* (2024) 363

[Calisti et al. 2011] Calisti, Marcello ; Giorelli, Michele ; Levy, Guy ; Mazzolai, Barbara ; Hochner, B ; Laschi, Cecilia ; Dario, Paolo: An octopus-bioinspired solution to movement and manipulation for soft robots. In: *Bioinspiration & biomimetics* 6 (2011), Nr. 3, S. 036002 328, 336

[Calzolari et al. 2020] Calzolari, Davide ; Della Santina, Cosimo ; Albu-Schäffer, Alin: Exponential convergence rates of nonlinear mechanical systems: The 1-DoF case with configuration-dependent inertia. In: *IEEE Control Systems Letters* 5 (2020), Nr. 2, S. 445–450 224

[Car 2020a] Carbon 3D Printing (Hrsg.): *EPU 40 Technical Datasheet.* 1st. 1089 Mills Way, Redwood City, CA 94063, USA: Carbon 3D Printing, 6 2020. (103208-00) . – Rev. E 113, 366

[Car 2020b] Carbon 3D Printing (Hrsg.): *FPU 50 Technical Datasheet.* 1st. 1089 Mills Way, Redwood City, CA 94063, USA: Carbon 3D Printing, 6 2020. (103215-02) . – Rev. A 113, 366

[Carlone and Pinciroli 2019] Carlone, Luca ; Pinciroli, Carlo: Robot co-design: beyond the monotone case. In: *2019 International Conference on Robotics and Automation (ICRA)* IEEE, 2019, S. 3024–3030 331

[Caroleo et al. 2025] Caroleo, Giammarco ; Albini, Alessandro ; Maiolino, Perla: Soft Robot Localization Using Distributed Miniaturized Time-of-Flight Sensors. (2025), S. 1–8 95, 308

[Carrasco-Zevallos et al. 2018] Carrasco-Zevallos, Oscar M. ; Viehland, Christian ; Keller, Brenton ; McNabb, Ryan P. ; Kuo, Anthony N. ; Izatt, Joseph A.: Constant linear velocity spiral scanning for near video rate 4D OCT ophthalmic and surgical imaging with isotropic transverse sampling. In: *Biomedical optics express* 9 (2018), Nr. 10, S. 5052–5070 188

[Cenedese et al. 2022] Cenedese, Mattia ; Axås, Joar ; Bäuerlein, Bastian ; Avila, Kerstin ; Haller, George: Data-driven modeling and prediction of non-linearizable dynamics via spectral submanifolds. In: *Nature communications* 13 (2022), Nr. 1, S. 872 5, 221

[Ceni et al. 2024] Ceni, Andrea ; Cossu, Andrea ; Stölzle, Maximilian ; Liu, Jingyue ; Della Santina, Cosimo ; Bacciu, Davide ; Gallicchio, Claudio: Random Oscillators Network for Time Series Processing. In: *International Conference on Artificial Intelligence and Statistics* PMLR, 2024, S. 4807–4815 221, 222, 223, 225, 434

[Champion et al. 2019] Champion, Kathleen ; Lusch, Bethany ; Kutz, J N. ; Brunton, Steven L.: Data-driven discovery of coordinates and governing equations. In: *Proceedings of the National Academy of Sciences* 116 (2019), Nr. 45, S. 22445–22451 221, 222, 240, 242, 244

[Chang and Khatib 1995] Chang, Kyong-Sok ; Khatib, Oussama: Manipulator control at kinematic singularities: A dynamically consistent strategy. In: *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on* Bd. 3 IEEE, 1995, S. 84–88 33, 35

[Chen and Wang 2020] Chen, Feifei ; Wang, Michael Y.: Design optimization of soft robots: A review of the state of the art. In: *IEEE Robotics & Automation Magazine* 27 (2020), Nr. 4, S. 27–43 323, 324, 328, 329, 332, 333, 334, 336, 342

[Chen et al. 2018] Chen, Ricky T. ; Rubanova, Yulia ; Bettencourt, Jesse ; Duvenaud, David K.: Neural ordinary differential equations. In: *Advances in neural information processing systems* 31 (2018) 10, 221, 222, 239, 240, 279, 344, 349, 350, 351

[Chen et al. 2024] Chen, Zixi ; Renda, Federico ; Le Gall, Alexia ; Mocellin, Lorenzo ; Bernabei, Matteo ; Dangel, Théo ; Ciuti, Gastone ; Cianchetti, Matteo ; Stefanini, Cesare: Data-driven methods applied to soft robot modeling and control: A review. In: *IEEE Transactions on Automation Science and Engineering* (2024) 5, 6, 10, 195, 340

[Cheng et al. 2020] Cheng, Hao ; Liu, Houde ; Wang, Xueqian ; Liang, Bin: Approximate Piecewise Constant Curvature Equivalent Model and Their Application to Continuum Robot Configuration Estimation. In: *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* IEEE, 2020, S. 1929–1936 82

[Chi et al. 2023] Chi, Cheng ; Xu, Zhenjia ; Feng, Siyuan ; Cousineau, Eric ; Du, Yilun ; Burchfiel, Benjamin ; Tedrake, Russ ; Song, Shuran: Diffusion policy: Visuomotor policy learning via action diffusion. In: *The International Journal of Robotics Research* (2023), S. 02783649241273668 261, 262, 265, 282, 286, 288, 300, 301, 317

[Chibani et al. 2013] Chibani, Abdelghani ; Amirat, Yacine ; Mohammed, Samer ; Matson, Eric ; Hagita, Norihiro ; Barreto, Marcos: Ubiquitous robotics: Recent challenges and future trends. In: *Robotics and Autonomous Systems* 61 (2013), Nr. 11, S. 1162–1172 2, 323

[Chin et al. 2018] Chin, Lillian ; Lipton, Jeffrey ; MacCurdy, Robert ; Romanishin, John ; Sharma, Chetan ; Rus, Daniela: Compliant electric actuators based on handed shearing auxetics. In: *2018 IEEE International Conference on Soft Robotics (RoboSoft)* IEEE, 2018, S. 100–107 14, 99, 315, 321

[Chin et al. 2019] Chin, Lillian ; Lipton, Jeffrey ; Yuen, Michelle C. ; Kramer-Bottiglio, Rebecca ; Rus, Daniela: Automated recycling separation enabled by soft robotic material classification. In: *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)* IEEE, 2019, S. 102–107 99

[Cho et al. 2014] Cho, Kyunghyun ; Van Merriënboer, Bart ; Gulcehre, Caglar ; Bahdanau, Dzmitry ; Bougares, Fethi ; Schwenk, Holger ; Bengio, Yoshua: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: *arXiv preprint arXiv:1406.1078* (2014) 221, 239, 240, 345, 349, 350, 351

[Chulliat et al. 2020] Chulliat, Arnaud ; Brown, William ; Alken, Patrick ; Beggan, Ciaran ; Nair, Manoj ; Cox, Grace ; Woods, Adam ; Macmillan, Susan ; Meyer, Brian ; Paniccia, Michael: The US/UK world magnetic model for 2020-2025. In: *British Geological Survey* (2020) 188

[Cianchetti 2021] Cianchetti, Matteo: Embodied intelligence in soft robotics through hardware multifunctionality. In: *Frontiers in Robotics and AI* 8 (2021), S. 724056 328, 332, 334

[Cianchetti et al. 2013] Cianchetti, Matteo ; Ranzani, Tommaso ; Gerboni, Giada ; De Falco, Iris ; Laschi, Cecilia ; Menciassi, Arianna: STIFF-FLOP surgical manipulator: Mechanical design and experimental characterization of the single module. In: *2013 IEEE/RSJ international conference on intelligent robots and systems* IEEE, 2013, S. 3576–3581 57, 323

[Coevoet et al. 2017] Coevoet, Eulalie ; Morales-Bieze, Thor ; Largilliere, Frederick ; Zhang, Zhongkai ; Thieffry, Maxime ; Sanz-Lopez, Mario ; Carrez, Bruno ; Marchal, Damien ; Goury, Olivier ; Dequidt, Jeremie et al.: Software toolkit for modeling, simulation, and control of soft robots. In: *Advanced Robotics* 31 (2017), Nr. 22, S. 1208–1224 26, 329, 334, 363

[Copaci et al. 2020] Copaci, Dorin-Sabin ; Blanco, Dolores ; Martin-Clemente, Alejandro ; Moreno, Luis: Flexible shape memory alloy actuators for soft robotics: Modelling and control. In: *International Journal of Advanced Robotic Systems* 17 (2020), Nr. 1, S. 1729881419886747 309

[Cosserat 1909] Cosserat, Eugene: *Théorie des corps déformables.* Librairie Scientifique A. Hermann et Fils, 1909 4, 26

[Coyle et al. 2007] Coyle, Shirley M. ; Ward, Tomás E ; Markham, Charles M.: Brain–computer interface using a simplified functional near-infrared spectroscopy system. In: *Journal of neural engineering* 4 (2007), Nr. 3, S. 219 139

[Cranmer et al. 2020] Cranmer, Miles ; Greydanus, Sam ; Hoyer, Stephan ; Battaglia, Peter ; Spergel, David ; Ho, Shirley: Lagrangian neural networks. In: *arXiv preprint arXiv:2003.04630* (2020) 7, 221

[Dämmer et al. 2018] Dämmer, Gabriel ; Gablenz, Sven ; Hildebrandt, Alexander ; Major, Zoltan: Design and shape optimization of PolyJet bellows actuators. In: *2018 IEEE International Conference on Soft Robotics (RoboSoft)* IEEE, 2018, S. 282–287 328

[Davoodi et al. 2022] Davoodi, Mohammadreza ; Iqbal, Asif ; Cloud, Joseph M. ; Beksi, William J. ; Gans, Nicholas R.: Rule-based safe probabilistic movement primitive control via control barrier functions. In: *IEEE Transactions on Automation Science and Engineering* 20 (2022), Nr. 3, S. 1500–1514 301

[De Luca et al. 2006] De Luca, Alessandro ; Albu-Schaffer, Alin ; Haddadin, Sami ; Hirzinger, Gerd: Collision detection and safe reaction with the DLR-III lightweight manipulator arm. In: *2006 IEEE/RSJ international conference on intelligent robots and systems* IEEE, 2006, S. 1623–1630 58

[De Santis et al. 2008] De Santis, Agostino ; Siciliano, Bruno ; De Luca, Alessandro ; Bicchi, Antonio: An atlas of physical human–robot interaction. In: *Mechanism and Machine Theory* 43 (2008), Nr. 3, S. 253–270 58, 60

[DeepMind et al. 2020] DeepMind ; Babuschkin, Igor ; Baumli, Kate ; Bell, Alison ; Bhupatiraju, Surya ; Bruce, Jake ; Buchlovsky, Peter ; Budden, David ; Cai, Trevor ; Clark, Aidan ; Danihelka, Ivo ; Dedieu, Antoine ; Fantacci, Claudio ; Godwin,

Jonathan ; Jones, Chris ; Hemsley, Ross ; Hennigan, Tom ; Hessel, Matteo ; Hou, Shaobo ; Kapturowski, Steven ; Keck, Thomas ; Kemaev, Iurii ; King, Michael ; Kunesch, Markus ; Martens, Lena ; Merzic, Hamza ; Mikulik, Vladimir ; Norman, Tamara ; Papamakarios, George ; Quan, John ; Ring, Roman ; Ruiz, Francisco ; Sanchez, Alvaro ; Sartran, Laurent ; Schneider, Rosalia ; Sezener, Eren ; Spencer, Stephen ; Srinivasan, Srivatsan ; Stanojević, Miloš ; Stokowiec, Wojciech ; Wang, Luyu ; Zhou, Guangyao ; Viola, Fabio: *The DeepMind JAX Ecosystem.* 2020 347

[Della Santina 2020] Della Santina, Cosimo: The soft inverted pendulum with affine curvature. In: *2020 59th IEEE Conference on Decision and Control (CDC)* IEEE, 2020, S. 4135–4142 31, 174, 176, 185

[Della Santina 2021] Della Santina, Cosimo: Flexible Manipulators. In: *Encyclopedia of Robotics* (2021) 158

[Della Santina 2025] Della Santina, Cosimo: Pushing the boundaries of actuators-to-task kinematic inversion: from fully actuated to underactuated (soft) robots. In: *The International Journal of Robotics Research* (2025). – In Preparation 14, 34, 40

[Della Santina et al. 2017] Della Santina, Cosimo ; Bianchi, Matteo ; Grioli, Giorgio ; Angelini, Franco ; Catalano, Manuel ; Garabini, Manolo ; Bicchi, Antonio: Controlling soft robots: balancing feedback and feedforward elements. In: *IEEE Robotics & Automation Magazine* 24 (2017), Nr. 3, S. 75–83 42, 74, 76, 139

[Della Santina et al. 2019a] Della Santina, Cosimo ; Bicchi, Antonio ; Rus, Daniela: Dynamic control of soft robots with internal constraints in the presence of obstacles. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* IEEE, 2019, S. 6622–6629 189

[Della Santina et al. 2020a] Della Santina, Cosimo ; Bicchi, Antonio ; Rus, Daniela: On an improved state parametrization for soft robots with piecewise constant curvature and its use in model based control. In: *IEEE Robotics and Automation Letters* 5 (2020), Nr. 2, S. 1001–1008 25, 29, 30, 84, 187, 188

[Della Santina et al. 2020b] Della Santina, Cosimo ; Catalano, Manuel G. ; Bicchi, Antonio ; Ang, M ; Khatib, O ; Siciliano, B: Soft robots. In: *Encyclopedia of Robotics* 489 (2020) 23, 139

[Della Santina et al. 2023] Della Santina, Cosimo ; Duriez, Christian ; Rus, Daniela: Model-Based Control of Soft Robots: A Survey of the State of the Art and Open Challenges. In: *IEEE Control Systems Magazine* 43 (2023), Nr. 3, S. 30–65 3, 4, 7, 8, 10, 11, 16, 19, 25, 26, 28, 31, 32, 37, 42, 43, 44, 45, 48, 62, 64, 65, 74, 76, 99, 113, 133, 155, 156, 195, 196, 197, 201, 205, 210, 243, 309, 310, 312, 324, 325, 333, 334, 363, 364

[Della Santina et al. 2020c] Della Santina, Cosimo ; Katzschmann, Robert K. ; Bicchi, Antonio ; Rus, Daniela: Model-based dynamic feedback control of a planar soft robot: trajectory tracking and interaction with the environment. In: *The International Journal of Robotics Research* 39 (2020), Nr. 4, S. 490–513 3, 4, 7, 16, 25, 32, 33, 37, 38, 43, 44, 46, 51, 52, 66, 131, 155, 162, 165, 314, 333, 366

[Della Santina et al. 2019b] Della Santina, Cosimo ; Pallottino, Lucia ; Rus, Daniela ; Bicchi, Antonio: Exact task execution in highly under-actuated soft limbs: an operational space based approach. In: *2019 IEEE International Conference on Soft Robotics (RoboSoft)* IEEE, 2019 66, 131

[Della Santina and Rus 2019] Della Santina, Cosimo ; Rus, Daniela: Control oriented modeling of soft robots: The polynomial curvature case. In: *IEEE Robotics and Automation Letters* 5 (2019), Nr. 2, S. 290–298 31, 99, 176, 205

[van Diepen and Shea 2022] Diepen, Merel van ; Shea, Kristina: Co-design of the morphology and actuation of soft robots for locomotion. In: *Journal of Mechanical Design* 144 (2022), Nr. 8, S. 083305 316, 328

[Ding et al. 2021] Ding, Ze Y. ; Loo, Junn Y. ; Baskaran, Vishnu M. ; Nurzaman, Surya G. ; Tan, Chee P.: Predictive uncertainty estimation using deep learning for soft robot multimodal sensing. In: *IEEE Robotics and Automation Letters* 6 (2021), Nr. 2, S. 951–957 174

[Dinh et al. 2016] Dinh, Laurent ; Sohl-Dickstein, Jascha ; Bengio, Samy: Density estimation using real nvp. In: *arXiv preprint arXiv:1605.08803* (2016) 18, 22, 303

[Dinh et al. 2017] Dinh, Laurent ; Sohl-Dickstein, Jascha ; Bengio, Samy: Density estimation using Real NVP. In: *International Conference on Learning Representations*, 2017 262, 267, 268

[Dörfler and Bullo 2014] Dörfler, Florian ; Bullo, Francesco: Synchronization in complex networks of phase oscillators: A survey. In: *Automatica* 50 (2014), Nr. 6, S. 1539–1564 272, 294

[Dormand and Prince 1980] Dormand, J. R. ; Prince, P. J.: A family of embedded Runge–Kutta formulae. In: *J. Comp. Appl. Math* 6 (1980), S. 19–26 346, 364, 365

[Dubied et al. 2022] Dubied, Mathieu ; Michelis, Mike Y. ; Spielberg, Andrew ; Katzschmann, Robert K.: Sim-to-real for soft robots using differentiable fem: Recipes for meshing, damping, and actuation. In: *IEEE Robotics and Automation Letters* 7 (2022), Nr. 2, S. 5015–5022 325, 333

[Dumoulin and Visin 2016] Dumoulin, Vincent ; Visin, Francesco: A guide to convolution arithmetic for deep learning. In: *arXiv preprint arXiv:1603.07285* (2016) 344

[Eiter et al. 1994] Eiter, Thomas ; Mannila, Heikki et al.: Computing discrete Fréchet distance. (1994) 284

[El Zaatari et al. 2019] El Zaatari, Shirine ; Marei, Mohamed ; Li, Weidong ; Usman, Zahid: Cobot programming for collaborative industrial tasks: An overview. In: *Robotics and Autonomous Systems* 116 (2019), S. 162–180 2

[Epstein and Xu 2016] Epstein, Irving R. ; Xu, Bing: Reaction–diffusion processes at the nano-and microscales. In: *Nature nanotechnology* 11 (2016), Nr. 4, S. 312–319 222, 249

[EuroNCAP 2004] EuroNCAP: European Protocol New Assessment Programme—Frontal Impact Testing Protocol / EuroNCAP. 2004. – Protocol. – European Protocol for frontal impact testing under the New Assessment Programme 60

[Falkenhahn et al. 2016] Falkenhahn, Valentin ; Hildebrandt, Alexander ; Neumann, Rüdiger ; Sawodny, Oliver: Dynamic control of the bionic handling assistant. In: *IEEE/ASME Transactions on Mechatronics* 22 (2016), Nr. 1, S. 6–17 155, 361

[Fang et al. 2020] Fang, Guoxin ; Matte, Christopher-Denny ; Scharff, Rob B. ; Kwok, Tsz-Ho ; Wang, Charlie C.: Kinematics of soft robots by geometric computing. In: *IEEE Transactions on Robotics* 36 (2020), Nr. 4, S. 1272–1286 329

[Farrell et al. 2020] Farrell, David T. ; McGinn, Conor ; Bennett, Gareth J.: Extension twist deformation response of an auxetic cylindrical structure inspired by deformed cell ligaments. In: *Composite Structures* 238 (2020), S. 111901 99

[Faure et al. 2012] Faure, François ; Duriez, Christian ; Delingette, Hervé ; Allard, Jérémie ; Gilles, Benjamin ; Marchesseau, Stéphanie ; Talbot, Hugo ; Courtecuisse, Hadrien ; Bousquet, Guillaume ; Peterlik, Igor et al.: Sofa: A multi-model framework for interactive physical simulation. In: *Soft tissue biomechanical modeling for computer assisted surgery*. Springer, 2012, S. 283–321 26, 155, 156

[Feldhusen et al. 2013] Feldhusen, Jörg ; Grote, Karl-Heinrich et al.: *Pahl/Beitz Konstruktionslehre*. Springer, 2013 327

[Ferraguti et al. 2020] Ferraguti, Federica ; Bertuletti, Mattia ; Landi, Chiara T. ; Bonfè, Marcello ; Fantuzzi, Cesare ; Secchi, Cristian: A control barrier function approach for maximizing performance while fulfilling to iso/ts 15066 regulations. In: *IEEE Robotics and Automation Letters* 5 (2020), Nr. 4, S. 5921–5928 58, 59, 62, 76

[Fragapane et al. 2021] Fragapane, Giuseppe ; De Koster, Rene ; Sgarbossa, Fabio ; Strandhagen, Jan O.: Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda. In: *European Journal of Operational Research* 294 (2021), Nr. 2, S. 405–426 2, 323

[Franco et al. 2021] Franco, Enrico ; Ayatullah, Tutla ; Sugiharto, Arif ; Garriga-Casanovas, Arnau ; Virdyawan, Vani: Nonlinear energy-based control of soft continuum pneumatic manipulators. In: *Nonlinear Dynamics* (2021), S. 1–25 155

[Franco et al. 2024] Franco, Enrico ; Borvorntanajanya, Korn ; Leung, Fung F.: Model based control of an affordable pressure regulator for pneumatic soft robots. In: *2024 IEEE Conference on Control Technology and Applications (CCTA)* IEEE, 2024, S. 747–752 168

[Fuentes-Pacheco et al. 2015] Fuentes-Pacheco, Jorge ; Ruiz-Ascencio, José ; Rendón-Mancha, Juan M.: Visual simultaneous localization and mapping: a survey. In: *Artificial intelligence review* 43 (2015), Nr. 1, S. 55–81 82

[Furter et al. 2025] Furter, Marius ; Huang, Yujun ; Zardini, Gioele: Composable Uncertainty in Symmetric Monoidal Categories for Design Problems. In: *arXiv preprint arXiv:2503.17274* (2025) 324

[Gams et al. 2015] Gams, Andrej ; Ude, Aleš ; Morimoto, Jun: Accelerating synchronization of movement primitives: Dual-arm discrete-periodic motion of a humanoid robot. In: *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)* IEEE, 2015, S. 2754–2760 264, 272

[Gao et al. 2024] Gao, Junpeng ; Michelis, Mike Y. ; Spielberg, Andrew ; Katzschmann, Robert K.: Sim-to-real of soft robots with learned residual physics. In: *IEEE Robotics and Automation Letters* (2024) 195

[Garg et al. 2022] Garg, Aman ; Good, Ian ; Revier, Daniel ; Airis, Kevin ; Lipton, Jeffrey: Kinematic Modeling of Handed Shearing Auxetics via Piecewise Constant Curvature. In: *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)* IEEE, 2022, S. 423–430 14, 99, 112, 118, 195

[Garnett 2023] Garnett, Roman: *Bayesian optimization*. Cambridge University Press, 2023 324, 334, 339

[Gazzola et al. 2018] Gazzola, M ; Dudte, LH ; McCormick, AG ; Mahadevan, L: Forward and inverse problems in the mechanics of soft filaments. In: *Royal Society open science* 5 (2018), Nr. 6, 171628. http://dx.doi.org/10.1098/rsos.171628. – DOI 10.1098/rsos.171628 4, 26, 31, 99, 101, 102, 105, 195, 243

[van der Geest et al. 2023] Geest, Nick van d. ; Garcia, Lorenzo ; Borret, Fraser ; Nates, Roy ; Gonzalez, Alberto: Soft-robotic green sea turtle (Chelonia mydas) developed to replace animal experimentation provides new insight into their propulsive strategies. In: *Scientific Reports* 13 (2023), Nr. 1, S. 11983 281, 289, 294

[van der Geest et al. 2022] Geest, Nick van d. ; Garcia, Lorenzo ; Nates, Roy ; Godoy, Daniel A.: New insight into the swimming kinematics of wild Green sea turtles (Chelonia mydas). In: *Scientific Reports* 12 (2022), Nr. 1, S. 18151 280, 289, 294

[Gelada et al. 2019] Gelada, Carles ; Kumar, Saurabh ; Buckman, Jacob ; Nachum, Ofir ; Bellemare, Marc G.: Deepmdp: Learning continuous latent space models for representation learning. In: *International conference on machine learning* PMLR, 2019, S. 2170–2179 221

[George Thuruthel et al. 2017] George Thuruthel, Thomas ; Falotico, Egidio ; Manti, Mariangela ; Pratesi, Andrea ; Cianchetti, Matteo ; Laschi, Cecilia: Learning closed loop kinematic controllers for continuum manipulators in unstructured environments. In: *Soft robotics* 4 (2017), Nr. 3, S. 285–296 4

[Ghasemi and Moghaddam 2025] Ghasemi, Parisa ; Moghaddam, Mohsen: Vision-Language Models for Design Concept Generation: An Actor-Critic Framework. In: *Journal of Mechanical Design* (2025), S. 1–28 342

[Gillespie et al. 2018] Gillespie, Morgan T. ; Best, Charles M. ; Townsend, Eric C. ; Wingate, David ; Killpack, Marc D.: Learning nonlinear dynamic models of soft robots for model predictive control with neural networks. In: *2018 IEEE International Conference on Soft Robotics (RoboSoft)* IEEE, 2018, S. 39–45 5, 6, 11

[Girshick 2015] Girshick, Ross: Fast R-CNN. In: *2015 IEEE International Conference on Computer Vision (ICCV)* IEEE, 2015, S. 1440–1448 269

[Goldman et al. 2014] Goldman, Roger E. ; Bajo, Andrea ; Simaan, Nabil: Compliant motion control for multisegment continuum robots with actuation force sensing. In: *IEEE Transactions on Robotics* 30 (2014), Nr. 4, S. 890–902 311

[Golub and Van Loan 2013] Golub, Gene H. ; Van Loan, Charles F.: *Matrix computations*. JHU press, 2013 227, 229, 230

[Gómez et al. 2021] Gómez, Pablo ; Toftevaag, Håvard Hem ; Meoni, Gabriele: torchquad: Numerical Integration in Arbitrary Dimensions with PyTorch. In: *Journal of Open Source Software* 6 (2021), Nr. 64, S. 3439 177

[Good et al. 2025] Good, Ian ; Balaji, Srivatsan ; Oh, David ; Thomas, Sawyer ; Lipton, Jeffrey I.: *Torque Responsive Metamaterials Enable High Payload Soft Robot Arms*. https://arxiv.org/abs/2501.09819. Version: 2025 14, 314

[Good et al. 2022] Good, Ian ; Brown-Moore, Tosh ; Patil, Aditya ; Revier, Daniel ; Lipton, Jeffrey I.: Expanding the Design Space for Electrically-Driven Soft Robots Through Handed Shearing Auxetics. In: *2022 International Conference on Robotics and Automation (ICRA)* IEEE, 2022, S. 10951–10957 14, 99, 102, 103, 104, 113, 115, 118, 135, 426

[Got et al. 1978] Got, C ; Patel, A ; Fayon, A ; Tarriere, C ; Walfisch, G: Results of experimental head impacts on cadavers: the various data obtained and their relations to some measured physical parameters / SAE Technical Paper. 1978. – Forschungsbericht 60

[Grattafiori et al. 2024] Grattafiori, Aaron ; Dubey, Abhimanyu ; Jauhri, Abhinav ; Pandey, Abhinav ; Kadian, Abhishek ; Al-Dahle, Ahmad ; Letman, Aiesha ; Mathur, Akhil ; Schelten, Alan ; Vaughan, Alex et al.: The llama 3 herd of models. In: *arXiv preprint arXiv:2407.21783* (2024) 64, 342

[Gravert et al. 2024] Gravert, Stephan-Daniel ; Varini, Elia ; Kazemipour, Amirhossein ; Michelis, Mike Y. ; Buchner, Thomas ; Hinchet, Ronan ; Katzschmann, Robert K.: Low-voltage electrohydraulic actuators for untethered robotics. In: *Science Advances* 10 (2024), Nr. 1, S. eadi9319 3

[Grazioso et al. 2019] Grazioso, Stanislao ; Di Gironimo, Giuseppe ; Siciliano, Bruno: A geometrically exact model for soft continuum robots: The finite element deformation space formulation. In: *Soft robotics* 6 (2019), Nr. 6, S. 790–811 99, 155, 156, 195, 363

[Greer et al. 2020] Greer, Joseph D. ; Blumenschein, Laura H. ; Alterovitz, Ron ; Hawkes, Elliot W. ; Okamura, Allison M.: Robust navigation of a soft growing robot by exploiting contact with the environment. In: *The International Journal of Robotics Research* 39 (2020), Nr. 14, S. 1724–1738 311

[Greydanus et al. 2019] Greydanus, Samuel ; Dzamba, Misko ; Yosinski, Jason: Hamiltonian neural networks. In: *Advances in neural information processing systems* 32 (2019) 7, 221

[Griffith and Kumar 1971] Griffith, Ernest W. ; Kumar, Kadaba S.: On the observability of nonlinear systems: I. In: *Journal of Mathematical Analysis and Applications* 35 (1971), Nr. 1, S. 135–147 324, 340

[Guan et al. 2023] Guan, Qinghua ; Stella, Francesco ; Della Santina, Cosimo ; Leng, Jinsong ; Hughes, Josie: Trimmed helicoids: an architectured soft structure yielding soft robots with high precision, large workspace, and compliant interactions. In: *npj Robotics* 1 (2023), Nr. 1, S. 4 3, 13, 18, 23, 151, 288, 311, 324, 328, 332, 334, 336

[Guo et al. 2019] Guo, Hao ; Ju, Feng ; Cao, Yanfei ; Qi, Fei ; Bai, Dongming ; Wang, Yaoyao ; Chen, Bai: Continuum robot shape estimation using permanent magnets and magnetic sensors. In: *Sensors and Actuators A: Physical* 285 (2019), S. 519–530 82, 173

[Ha and Schmidhuber 2018] Ha, David ; Schmidhuber, Jürgen: World models. In: *arXiv preprint arXiv:1803.10122* (2018) 5, 221

[Ha et al. 2018] Ha, Sehoon ; Coros, Stelian ; Alspach, Alexander ; Bern, James M. ; Kim, Joohyung ; Yamane, Katsu: Computational design of robotic devices from high-level motion specifications. In: *IEEE Transactions on Robotics* 34 (2018), Nr. 5, S. 1240–1251 328

[Haddadin 2013] Haddadin, Sami: *Towards safe robots: approaching Asimov's 1st law*. Bd. 90. Heidelberg : Springer Berlin, 2013 2, 58, 59, 60, 61, 62, 70, 323, 324

[Haddadin et al. 2008a] Haddadin, Sami ; Albu-Schaffer, Alin ; De Luca, Alessandro ; Hirzinger, Gerd: Collision detection and reaction: A contribution to safe physical human-robot interaction. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems* IEEE, 2008, S. 3356–3363 58, 59

[Haddadin et al. 2011a] Haddadin, Sami ; Albu-Schaffer, Alin ; Haddadin, Fahed ; Rosmann, Jurgen ; Hirzinger, Gerd: Study on soft-tissue injury in robotics. In: *IEEE Robotics & Automation Magazine* 18 (2011), Nr. 4, S. 20–34 58, 59

[Haddadin et al. 2007] Haddadin, Sami ; Albu-Schäffer, Alin ; Hirzinger, Gerd: Safety evaluation of physical human-robot interaction via crash-testing. In: *Robotics: Science and systems* Bd. 3, 2007, S. 217–224 60, 61

[Haddadin et al. 2008b] Haddadin, Sami ; Albu-Schaffer, Alin ; Hirzinger, Gerd: The role of the robot mass and velocity in physical human-robot interaction-Part I: Non-constrained blunt impacts. In: *2008 IEEE International Conference on Robotics and Automation* IEEE, 2008, S. 1331–1338 60

[Haddadin et al. 2009] Haddadin, Sami ; Albu-Schäffer, Alin ; Hirzinger, Gerd: Requirements for safe robots: Measurements, analysis and new insights. In: *The International Journal of Robotics Research* 28 (2009), Nr. 11-12, S. 1507–1527 59, 60, 61, 62, 64, 65, 66, 67, 70, 71, 75, 323

[Haddadin et al. 2010a] Haddadin, Sami ; Albu-Schäffer, Alin ; Hirzinger, Gerd: Soft-tissue injury in robotics. In: *2010 IEEE International Conference on Robotics and Automation* IEEE, 2010, S. 3426–3433 59

[Haddadin et al. 2011b] Haddadin, Sami ; Albu-Schäffer, Alin ; Hirzinger, Gerd: Safe physical human-robot interaction: measurements, analysis and new insights. In: *Robotics Research: The 13th International Symposium ISRR* Springer, 2011, S. 395–407 59, 60, 61, 64, 66

[Haddadin et al. 2017] Haddadin, Sami ; De Luca, Alessandro ; Albu-Schäffer, Alin: Robot collisions: A survey on detection, isolation, and identification. In: *IEEE Transactions on Robotics* 33 (2017), Nr. 6, S. 1292–1312 2

[Haddadin et al. 2010b] Haddadin, Sami ; Urbanek, Holger ; Parusel, Sven ; Burschka, Darius ; Rossmann, Jürgen ; Albu-Schäffer, Alin ; Hirzinger, Gerd: Real-time reactive motion generation based on variable attractor dynamics and shaped velocities. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* IEEE, 2010, S. 3109–3116 59

[Hafner et al. 2019a] Hafner, Danijar ; Lillicrap, Timothy ; Ba, Jimmy ; Norouzi, Mohammad: Dream to Control: Learning Behaviors by Latent Imagination. In: *International Conference on Learning Representations*, 2019 221

[Hafner et al. 2019b] Hafner, Danijar ; Lillicrap, Timothy ; Fischer, Ian ; Villegas, Ruben ; Ha, David ; Lee, Honglak ; Davidson, James: Learning latent dynamics for planning from pixels. In: *International conference on machine learning* PMLR, 2019, S. 2555–2565 221

[Hafner et al. 2023] Hafner, Danijar ; Pasukonis, Jurgis ; Ba, Jimmy ; Lillicrap, Timothy: Mastering diverse domains through world models. In: *arXiv preprint arXiv:2301.04104* (2023) 221

[Haggerty et al. 2023] Haggerty, David A. ; Banks, Michael J. ; Kamenar, Ervin ; Cao, Alan B. ; Curtis, Patrick C. ; Mezić, Igor ; Hawkes, Elliot W.: Control of soft robots with inertial dynamics. In: *Science robotics* 8 (2023), Nr. 81, S. eadd6864 76, 221

[Haresh et al. 2024] Haresh, Sanjay ; Dijkman, Daniel ; Bhattacharyya, Apratim ; Memisevic, Roland: ClevrSkills: Compositional Language And Visual Reasoning in Robotics. In: *Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2024 317

[Hausdorff 1914] Hausdorff, Felix: *Grundzüge der mengenlehre*. Bd. 7. von Veit, 1914 285, 288

[Hawkes et al. 2017] Hawkes, Elliot W. ; Blumenschein, Laura H. ; Greer, Joseph D. ; Okamura, Allison M.: A soft robot that navigates its environment through growth. In: *Science Robotics* 2 (2017), Nr. 8, S. eaan3028 3, 57, 323

[Hawkes et al. 2021] Hawkes, Elliot W. ; Majidi, Carmel ; Tolley, Michael T.: Hard questions for soft robotics. In: *Science robotics* 6 (2021), Nr. 53, S. eabg6049 3, 57

[He et al. 2021] He, Hongmei ; Gray, John ; Cangelosi, Angelo ; Meng, Qinggang ; McGinnity, T M. ; Mehnen, Jörn: The challenges and opportunities of human-centered AI for trustworthy robots and autonomous systems. In: *IEEE Transactions on Cognitive and Developmental Systems* 14 (2021), Nr. 4, S. 1398–1412 2

[Heek et al. 2023] Heek, Jonathan ; Levskaya, Anselm ; Oliver, Avital ; Ritter, Marvin ; Rondepierre, Bertrand ; Steiner, Andreas ; Zee, Marc van: *Flax: A neural network library and ecosystem for JAX*. 2023 346

[Hernández-Lobato et al. 2014] Hernández-Lobato, José M. ; Hoffman, Matthew W. ; Ghahramani, Zoubin: Predictive entropy search for efficient global optimization of black-box functions. In: *Advances in neural information processing systems* 27 (2014) 324, 338

[Hewing et al. 2020] Hewing, Lukas ; Wabersich, Kim P. ; Menner, Marcel ; Zeilinger, Melanie N.: Learning-based model predictive control: Toward safe learning in control. In: *Annual Review of Control, Robotics, and Autonomous Systems* 3 (2020), S. 269–296 6, 59, 62, 221, 324

[Higgins et al. 2017] Higgins, Irina ; Matthey, Loic ; Pal, Arka ; Burgess, Christopher P. ; Glorot, Xavier ; Botvinick, Matthew M. ; Mohamed, Shakir ; Lerchner, Alexander: beta-vae: Learning basic visual concepts with a constrained variational framework. In: *ICLR (Poster)* 3 (2017) 238

[Hiller and Lipson 2012] Hiller, Jonathan ; Lipson, Hod: Automatic Design and Manufacture of Soft Robots. In: *IEEE Transactions on Robotics* 28 (2012), Nr. 2, S. 457–466. http://dx.doi.org/10.1109/TRO.2011.2172702. – DOI 10.1109/TRO.2011.2172702 323

[Ho et al. 2020] Ho, Jonathan ; Jain, Ajay ; Abbeel, Pieter: Denoising diffusion probabilistic models. In: *Advances in neural information processing systems* 33 (2020), S. 6840–6851 282

[Hochreiter and Schmidhuber 1997] Hochreiter, Sepp ; Schmidhuber, Jürgen: Long short-term memory. In: *Neural computation* 9 (1997), Nr. 8, S. 1735–1780 221

[Hofer et al. 2019] Hofer, Matthias ; Spannagl, Lukas ; D'Andrea, Raffaello: Iterative learning control for fast and accurate position tracking with an articulated soft robotic arm. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* IEEE, 2019, S. 6602–6607 4, 5

[Homberg et al. 2019] Homberg, Bianca S. ; Katzschmann, Robert K. ; Dogar, Mehmet R. ; Rus, Daniela: Robust proprioceptive grasping with a soft robot hand. In: *Autonomous Robots* 43 (2019), Nr. 3, S. 681–696 82

[Hu et al. 2022] Hu, Jiaheng ; Whitman, Julian ; Travers, Matthew ; Choset, Howie: Modular robot design optimization with generative adversarial networks. In: *2022 International Conference on Robotics and Automation (ICRA)* IEEE, 2022, S. 4282–4288 336

[Hu et al. 2024] Hu, Yingbai ; Abu-Dakka, Fares J. ; Chen, Fei ; Luo, Xiao ; Li, Zheng ; Knoll, Alois ; Ding, Weiping: Fusion dynamical systems with machine learning in imitation learning: A comprehensive overview. In: *Information Fusion* (2024), S. 102379 261, 263

[Hu et al. 2019] Hu, Yuanming ; Liu, Jiancheng ; Spielberg, Andrew ; Tenenbaum, Joshua B. ; Freeman, William T. ; Wu, Jiajun ; Rus, Daniela ; Matusik, Wojciech: Chainqueen: A real-time differentiable physical simulator for soft robotics. In: *2019 International conference on robotics and automation (ICRA)* IEEE, 2019, S. 6265–6271 329, 363

[Huang and Varma 1991] Huang, Ming Z. ; Varma, Hareendra: Optimal rate allocation in kinematically-redundant manipulators-the dual projection method. In: *Proceedings. 1991 IEEE International Conference on Robotics and Automation* IEEE Computer Society, 1991, S. 702–703 34

[Huang et al. 2025] Huang, Yujun ; Furter, Marius ; Zardini, Gioele: On Composable and Parametric Uncertainty in Systems Co-Design. In: *arXiv preprint arXiv:2504.02766* (2025) 324

[Hughes et al. 2020] Hughes, Josie ; Stella, Francesco ; Santina, Cosimo D. ; Rus, Daniela: Sensing Soft Robot Shape Using IMUs: An Experimental Investigation. In: *International Symposium on Experimental Robotics* Springer, 2020, S. 543–552 82

[Hunt 1975] Hunt, KH: Coefficient of Restitution Interpreted as Damping in Vibroimpact. In: *Journal of Applied Mechanics* 10 (1975), Nr. 1.3423596, S. 01333795 65

[Hurst et al. 2024] Hurst, Aaron ; Lerer, Adam ; Goucher, Adam P. ; Perelman, Adam ; Ramesh, Aditya ; Clark, Aidan ; Ostrow, AJ ; Welihinda, Akila ; Hayes, Alan ; Radford, Alec et al.: Gpt-4o system card. In: *arXiv preprint arXiv:2410.21276* (2024) 342

[Huttenlocher et al. 1993] Huttenlocher, Daniel P. ; Klanderman, Gregory A. ; Rucklidge, William J.: Comparing images using the Hausdorff distance. In: *IEEE Transactions on pattern analysis and machine intelligence* 15 (1993), Nr. 9, S. 850–863 285

[Iida and Laschi 2011] Iida, Fumiya ; Laschi, Cecilia: Soft robotics: Challenges and perspectives. In: *Procedia Computer Science* 7 (2011), S. 99–102 57, 323

[Ijspeert et al. 2013] Ijspeert, Auke J. ; Nakanishi, Jun ; Hoffmann, Heiko ; Pastor, Peter ; Schaal, Stefan: Dynamical movement primitives: learning attractor models for motor behaviors. In: *Neural computation* 25 (2013), Nr. 2, S. 328–373 17, 18, 22, 259, 261, 263, 264, 302

[Ijspeert et al. 2002] Ijspeert, Auke J. ; Nakanishi, Jun ; Schaal, Stefan: Learning rhythmic movements by demonstration using nonlinear oscillators. In: *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS2002)*, 2002, S. 958–963 261, 263, 264, 302

[ISO 10218-1: 2011 2011] ISO 10218-1: 2011, S: *Robots and robotic devices—Safety requirements for industrial robots—Part 1: Robots.* 2011 58

[Izmailov et al. 2018] Izmailov, P ; Wilson, AG ; Podoprikhin, D ; Vetrov, D ; Garipov, T: Averaging weights leads to wider optima and better generalization. In: *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, 2018, S. 876–885 182, 186

[Jaffe et al. 2024] Jaffe, Sean ; Davydov, Alexander ; Lapsekili, Deniz ; Singh, Ambuj K. ; Bullo, Francesco: Learning neural contracting dynamics: Extended linearization and global guarantees. In: *Advances in Neural Information Processing Systems* 37 (2024), S. 66204–66225 279

[Jaquier et al. 2025] Jaquier, Noémie ; Welle, Michael C. ; Gams, Andrej ; Yao, Kunpeng ; Fichera, Bernardo ; Billard, Aude ; Ude, Aleš ; Asfour, Tamim ; Kragic, Danica: Transfer learning in robotics: An upcoming breakthrough? A review of promises and challenges. In: *The International Journal of Robotics Research* 44 (2025), Nr. 3, S. 465–485 261, 262, 295

[Jekel et al. 2019] Jekel, Charles F. ; Venter, Gerhard ; Venter, Martin P. ; Stander, Nielen ; Haftka, Raphael T.: Similarity measures for identifying material parameters from hysteresis loops using inverse analysis. In: *International Journal of Material Forming* (2019), may. http://dx.doi.org/10.1007/s12289-018-1421-8. – DOI 10.1007/s12289–018–1421–8. – ISSN 1960–6214 283

[Jewett and Carstensen 2019] Jewett, Jackson L. ; Carstensen, Josephine V.: Topology-optimized design, construction and experimental evaluation of concrete beams. In: *Automation in Construction* 102 (2019), S. 59–67 328

[Jitosho et al. 2023] Jitosho, Rianna ; Lum, Tyler Ga W. ; Okamura, Allison ; Liu, Karen: Reinforcement learning enables real-time planning and control of agile maneuvers for soft robot arms. In: *Conference on Robot Learning* PMLR, 2023, S. 1131–1153 3, 4, 5

[Johnson 1987] Johnson, Kenneth L.: *Contact mechanics*. Cambridge university press, 1987 65

[Jones and Walker 2006] Jones, Bryan A. ; Walker, Ian D.: Kinematics for multisection continuum robots. In: *IEEE Transactions on Robotics* 22 (2006), Nr. 1, S. 43–55 162

[Jones et al. 1998] Jones, Donald R. ; Schonlau, Matthias ; Welch, William J.: Efficient global optimization of expensive black-box functions. In: *Journal of Global optimization* 13 (1998), S. 455–492 339

[Jørgensen et al. 2022] Jørgensen, Jonas ; Bojesen, Kirsten B. ; Jochum, Elizabeth: Is a soft robot more "natural"? Exploring the perception of soft robotics in human–robot interaction. In: *International Journal of Social Robotics* 14 (2022), Nr. 1, S. 95–113 3, 313

[Junge and Hughes 2022] Junge, Kai ; Hughes, Josie: Leveraging Embodied Intelligence for Dexterous Robotic Manipulators Through Iterative Co-design. In: *IOP Conference Series: Materials Science and Engineering* Bd. 1261 IOP Publishing, 2022, S. 012009 323, 324, 328, 329, 331, 333, 334, 335, 336, 337

[Kabzan et al. 2019] Kabzan, Juraj ; Hewing, Lukas ; Liniger, Alexander ; Zeilinger, Melanie N.: Learning-based model predictive control for autonomous racing. In: *IEEE Robotics and Automation Letters* 4 (2019), Nr. 4, S. 3363–3370 6

[Kaiser et al. 2018] Kaiser, Eurika ; Kutz, J N. ; Brunton, Steven L.: Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. In: *Proceedings of the Royal Society A* 474 (2018), Nr. 2219, S. 20180335 196

[Kaptanoglu et al. 2022] Kaptanoglu, Alan A. ; Silva, Brian M. ; Fasel, Urban ; Kaheman, Kadierdan ; Goldschmidt, Andy J. ; Callaham, Jared ; Delahunt, Charles B. ; Nicolaou, Zachary G. ; Champion, Kathleen ; Loiseau, Jean-Christophe ; Kutz, J. N. ; Brunton, Steven L.: PySINDy: A comprehensive Python package for robust sparse system identification. In: *Journal of Open Source Software* 7 (2022), Nr. 69, 3994. http://dx.doi.org/10.21105/joss.03994. – DOI 10.21105/joss.03994 121

[Kasaei et al. 2023] Kasaei, Mohammadreza ; Babarahmati, Keyhan K. ; Li, Zhibin ; Khadem, Mohsen: A Data-efficient Neural ODE Framework for Optimal Control of Soft Manipulators. In: *The Conference on Robot Learning 2023* PMLR, 2023, S. 1–14 6, 333

[Katzschmann et al. 2019] Katzschmann, Robert K. ; Della Santina, Cosimo ; Toshimitsu, Yasunori ; Bicchi, Antonio ; Rus, Daniela: Dynamic motion control of multi-segment soft robots using piecewise constant curvature matched with an augmented rigid body model. In: *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)* IEEE, 2019, S. 454–461 13, 91

[Katzschmann et al. 2018] Katzschmann, Robert K. ; DelPreto, Joseph ; MacCurdy, Robert ; Rus, Daniela: Exploration of underwater life with an acoustically controlled soft robotic fish. In: *Science Robotics* 3 (2018), Nr. 16. http://dx.doi.org/10.1126/scirobotics.aar3449. – DOI 10.1126/scirobotics.aar3449 3

[Kelly 1995] Kelly, Rafael: A tuning procedure for stable PID control of robot manipulators. In: *Robotica* 13 (1995), Nr. 2, S. 141–148 11, 310

[Kelly 1997] Kelly, Rafael: PD control with desired gravity compensation of robotic manipulators: a review. In: *The International Journal of Robotics Research* 16 (1997), Nr. 5, S. 660–672 16, 41

[Kelly 1998] Kelly, Rafael: Global positioning of robot manipulators via PD control plus a class of nonlinear integral actions. In: *IEEE Transactions on Automatic Control* 43 (1998), Nr. 7, S. 934–938 11, 16, 310

[Kelly and Carelli 1996] Kelly, Rafael ; Carelli, Ricardo: A class of nonlinear PD-type controllers for robot manipulators. In: *Journal of Robotic Systems* 13 (1996), Nr. 12, S. 793–802 11, 16, 45, 46, 310

[Kelly and Salgado 1994] Kelly, Rafael ; Salgado, Ricardo: PD control with computed feedforward of robot manipulators: A design procedure. In: *IEEE Transactions on Robotics and Automation* 10 (1994), Nr. 4, S. 566–571 42, 45

[Khadivar et al. 2021] Khadivar, Farshad ; Lauzana, Ilaria ; Billard, Aude: Learning dynamical systems with bifurcations. In: *Robotics and Autonomous Systems* 136 (2021), S. 103700 261, 262, 264, 265, 267, 283

[Khalil 2002] Khalil, Hassan K.: Nonlinear systems third edition. In: *Patience Hall* 115 (2002) 6, 8, 12, 15, 16, 21, 22, 32, 39, 43, 44, 45, 222, 226, 230, 231, 232, 261, 309, 312

[Khan et al. 2020] Khan, Muhammad A. ; Das, Rig ; Iversen, Helle K. ; Puthusserypady, Sadasivan: Review on motor imagery based BCI systems for upper limb post-stroke neurorehabilitation: From designing to application. In: *Computers in biology and medicine* 123 (2020), S. 103843 139

[Khatib 1987] Khatib, Oussama: A unified approach for motion and force control of robot manipulators: The operational space formulation. In: *IEEE Journal on Robotics and Automation* 3 (1987), Nr. 1, S. 43–53 2, 25, 33, 35, 40, 51, 61, 66, 120, 366

[Kidger 2021] Kidger, P: *On neural differential equations*, University of Oxford, Diss., 2021 10, 233, 346, 364

[Kim et al. 2025] Kim, Byungchul ; Wang, Tsun-Hsuan ; Rus, Daniela: Generative-AI-Driven Jumping Robot Design Using Diffusion Models. In: *2025 International Conference on Robotics and Automation (ICRA)* IEEE, 2025 324

[Kim et al. 2021] Kim, Daekyum ; Kim, Sang-Hun ; Kim, Taekyoung ; Kang, Brian B. ; Lee, Minhyuk ; Park, Wookeun ; Ku, Subyeong ; Kim, DongWook ; Kwon, Junghan ; Lee, Hochang et al.: Review of machine learning methods in soft robotics. In: *Plos one* 16 (2021), Nr. 2, S. e0246102 10, 195

[Kim et al. 2024] Kim, Moo J. ; Pertsch, Karl ; Karamcheti, Siddharth ; Xiao, Ted ; Balakrishna, Ashwin ; Nair, Suraj ; Rafailov, Rafael ; Foster, Ethan ; Lam, Grace ; Sanketi, Pannag et al.: OpenVLA: An Open-Source Vision-Language-Action Model. In: *arXiv preprint arXiv:2406.09246* (2024) 317

[Kingma and Ba 2014] Kingma, Diederik P. ; Ba, Jimmy: Adam: A method for stochastic optimization. In: *arXiv preprint arXiv:1412.6980* (2014) 280, 282, 347

[Kingma and Welling 2014] Kingma, Diederik P. ; Welling, Max: Auto-encoding variational bayes. In: *Proceedings of the International Conference on Learning Representations*, 2014 16, 221, 238, 344

[Kirschner et al. 2021a] Kirschner, Robin J. ; Mansfeld, Nico ; Abdolshah, Saeed ; Haddadin, Sami: Experimental analysis of impact forces in constrained collisions according to iso/ts 15066. In: *2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR)* IEEE, 2021, S. 1–5 60

[Kirschner et al. 2021b] Kirschner, Robin J. ; Mansfeld, Nico ; Peña, Guillermo G. ; Abdolshah, Saeed ; Haddadin, Sami: Notion on the correct use of the robot effective mass in the safety context and comments on ISO/TS 15066. In: *2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR)* IEEE, 2021, S. 6–9 61, 71

[Knippers et al. 2021] Knippers, Jan ; Kropp, Cordula ; Menges, Achim ; Sawodny, Oliver ; Weiskopf, Daniel: Integrative computational design and construction: Rethinking architecture digitally. In: *Civil engineering design* 3 (2021), Nr. 4, S. 123–135 328

[Kober and Peters 2009] Kober, Jens ; Peters, Jan: Learning motor primitives for robotics. In: *2009 IEEE International Conference on Robotics and Automation* IEEE, 2009, S. 2112–2118 261, 263, 264

[Kobyzev et al. 2020] Kobyzev, Ivan ; Prince, Simon J. ; Brubaker, Marcus A.: Normalizing flows: An introduction and review of current methods. In: *IEEE transactions on pattern analysis and machine intelligence* 43 (2020), Nr. 11, S. 3964–3979 259, 309

[Kokotovic 1992] Kokotovic, Petar V.: The joy of feedback: nonlinear and adaptive. In: *IEEE control systems magazine* 12 (1992), Nr. 3, S. 7–17 15, 21

[Kramberger et al. 2018] Kramberger, Aljaž ; Shahriari, Erfan ; Gams, Andrej ; Nemec, Bojan ; Ude, Aleš ; Haddadin, Sami: Passivity based iterative learning of admittance-coupled dynamic movement primitives for interaction with changing environments. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* IEEE, 2018, S. 6023–6028 261, 264

[Kramer et al. 2011] Kramer, Rebecca K. ; Majidi, Carmel ; Sahai, Ranjana ; Wood, Robert J.: Soft curvature sensors for joint angle proprioception. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems* IEEE, 2011, S. 1919–1926 173

[Krieg et al. 2015] Krieg, Michael ; Sledge, Isaac ; Mohseni, Kamran: Design considerations for an underwater soft-robot inspired from marine invertebrates. In: *Bioinspiration & biomimetics* 10 (2015), Nr. 6, S. 065004 342

[Kroell et al. 1974] Kroell, Charles K. ; Schneider, Dennis C. ; Nahum, Alan M.: Impact tolerance and response of the human thorax II. In: *SAE Transactions* (1974), S. 3724–3762 60

[Lacevic and Rocco 2011] Lacevic, Bakir ; Rocco, Paolo: Safety-oriented control of robotic manipulators–a kinematic approach. In: *IFAC Proceedings Volumes* 44 (2011), Nr. 1, S. 11508–11513 58, 59

[Lacevic et al. 2022] Lacevic, Bakir ; Zanchettin, Andrea M. ; Rocco, Paolo: Safe human-robot collaboration via collision checking and explicit representation of danger zones. In: *IEEE Transactions on Automation Science and Engineering* 20 (2022), Nr. 2, S. 846–861 58, 59

[Lanthaler et al. 2024] Lanthaler, Samuel ; Rusch, T K. ; Mishra, Siddhartha: Neural oscillators are universal. In: *Advances in Neural Information Processing Systems* 36 (2024) 221, 223, 225

[Larson et al. 2016] Larson, Christina ; Peele, B ; Li, S ; Robinson, S ; Totaro, M ; Beccai, L ; Mazzolai, B ; Shepherd, R: Highly stretchable electroluminescent skin for optical signaling and tactile sensing. In: *science* 351 (2016), Nr. 6277, S. 1071–1074 3

[Laschi and Cianchetti 2014] Laschi, Cecilia ; Cianchetti, Matteo: Soft robotics: new perspectives for robot bodyware and control. In: *Frontiers in bioengineering and biotechnology* 2 (2014), S. 3 57

[Laschi et al. 2012] Laschi, Cecilia ; Cianchetti, Matteo ; Mazzolai, Barbara ; Margheri, Laura ; Follador, Maurizio ; Dario, Paolo: Soft robot arm inspired by the octopus. In: *Advanced Robotics* 26 (2012), Nr. 7, S. 709–727 3, 342

[Laschi and Mazzolai 2024] Laschi, Cecilia ; Mazzolai, Barbara: Bioinspiration. In: *Handbook on Soft Robotics*. Springer, 2024, S. 13–33 334, 336, 342

[Laschi et al. 2016] Laschi, Cecilia ; Mazzolai, Barbara ; Cianchetti, Matteo: Soft robotics: Technologies and systems pushing the boundaries of robot abilities. In: *Sci. Robot.* 1 (2016), Nr. 1, S. eaah3690 2, 323

[Laschi et al. 2023] Laschi, Cecilia ; Thuruthel, Thomas G. ; Lida, Fumiya ; Merzouki, Rochdi ; Falotico, Egidio: Learning-based control strategies for soft robots: Theory, achievements, and future challenges. In: *IEEE Control Systems Magazine* 43 (2023), Nr. 3, S. 100–113 333

[Lau and Viano 1986] Lau, Ian V. ; Viano, David C.: The viscous criterion—bases and applications of an injury severity index for soft tissues. In: *SAE transactions* (1986), S. 672–691 61

[Lee et al. 2017] Lee, Kyuhwa ; Liu, Dong ; Perroud, Laetitia ; Chavarriaga, Ricardo ; Millán, José del R: A brain-controlled exoskeleton with cascaded event-related desynchronization classifiers. In: *Robotics and Autonomous Systems* 90 (2017), S. 15–23 139

[Lee et al. 2024] Lee, Sharon ; Zhang, Ruohan ; Hwang, Minjune ; Hiranaka, Ayano ; Wang, Chen ; Ai, Wensi ; Tan, Jin Jie R. ; Gupta, Shreya ; Hao, Yilun ; Levine, Gabrael et al.: NOIR: Neural Signal Operated Intelligent Robots for Everyday Activities. In: *Bridging the Gap between Cognitive Science and Robot Learning in the Real World: Progresses and New Directions*, 2024 17, 139

[Legrand et al. 2023] Legrand, Julie ; Terryn, Seppe ; Roels, Ellen ; Vanderborght, Bram: Reconfigurable, multi-material, voxel-based soft robots. In: *IEEE Robotics and Automation Letters* 8 (2023), Nr. 3, S. 1255–1262 316, 323, 324, 328, 329, 336

[Lenz et al. 2015] Lenz, Ian ; Knepper, Ross A. ; Saxena, Ashutosh: DeepMPC: Learning deep latent features for model predictive control. In: *Robotics: Science and Systems* Bd. 10 Rome, Italy, 2015, S. 25 221

[Lepri et al. 2023] Lepri, Marco ; Bacciu, Davide ; Della Santina, Cosimo: Neural Autoencoder-Based Structure-Preserving Model Order Reduction and Control Design for High-Dimensional Physical Systems. In: *IEEE Control Systems Letters* (2023) 221

[Levenberg 1944] Levenberg, Kenneth: A method for the solution of certain non-linear problems in least squares. In: *Quarterly of Applied Mathematics* 2 (1944), S. 164–168 13, 89, 366, 367

[Li et al. 2021] Li, Shuguang ; Awale, Samer A. ; Bacher, Katharine E. ; Buchner, Thomas J. ; Della Santina, Cosimo ; Wood, Robert J. ; Rus, Daniela: Scaling Up Soft Robotics: A Meter-Scale, Modular, and Reconfigurable Soft Robotic System. In: *Soft Robotics* (2021) 82, 173

[Li et al. 2023] Li, Yucheng ; Myszka, David H. ; Murray, Andrew: The kinematics of constant curvature continuum robots through three segments. In: *IEEE Robotics and Automation Letters* 8 (2023), Nr. 11, S. 7631–7638 29

[Lin et al. 2020] Lin, Xi ; Willemet, Laurence ; Bailleul, Alexandre ; Wiertlewski, Michaël: Curvature sensing with a spherical tactile sensor using the color-interference of a marker array. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)* IEEE, 2020, S. 603–609 82

[Lindenroth et al. 2016] Lindenroth, Lukas ; Back, Junghwan ; Schoisengeier, Adrian ; Noh, Yohan ; Würdemann, Helge ; Althoefer, Kaspar ; Liu, Hongbin: Stiffness-based modelling of a hydraulically-actuated soft robotics manipulator. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* IEEE, 2016, S. 2458–2463 155

[Liou et al. 2014] Liou, Cheng-Yuan ; Cheng, Wei-Chen ; Liou, Jiun-Wei ; Liou, Daw-Ran: Autoencoder for words. In: *Neurocomputing* 139 (2014), S. 84–96 221

[Lipton et al. 2018] Lipton, Jeffrey I. ; MacCurdy, Robert ; Manchester, Zachary ; Chin, Lillian ; Cellucci, Daniel ; Rus, Daniela: Handedness in shearing auxetics creates rigid and compliant structures. In: *Science* 360 (2018), Nr. 6389, S. 632–635 3, 13, 14, 99, 315, 321

[Liu et al. 2018a] Liu, Guilin ; Reda, Fitsum A. ; Shih, Kevin J. ; Wang, Ting-Chun ; Tao, Andrew ; Catanzaro, Bryan: Image inpainting for irregular holes using partial convolutions. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, S. 85–100 347

[Liu et al. 2024a] Liu, Jingyue ; Borja, Pablo ; Della Santina, Cosimo: Physics-Informed Neural Networks to Model and Control Robots: A Theoretical and Experimental Investigation. In: *Advanced Intelligent Systems* 6 (2024), Nr. 5, S. 2300385 6, 7, 195, 197, 208, 216, 324, 333

[Liu et al. 2024b] Liu, Sichao ; Wang, Lihui ; Gao, Robert X.: Cognitive neuroscience and robotics: Advancements and future research directions. In: *Robotics and Computer-Integrated Manufacturing* 85 (2024), S. 102610 139

[Liu et al. 2018b] Liu, Yiliang ; Su, Wenbin ; Li, Zhijun ; Shi, Guangming ; Chu, Xiaoli ; Kang, Yu ; Shang, Weiwei: Motor-imagery-based teleoperation of a dual-arm robot performing manipulation tasks. In: *IEEE Transactions on Cognitive and Developmental Systems* 11 (2018), Nr. 3, S. 414–424 139

[Lobdell et al. 1973] Lobdell, TE ; Kroell, CK ; Schneider, DC ; Hering, WE ; Nahum, AM: Impact response of the human thorax. In: *Human impact response: Measurement and simulation* (1973), S. 201–245 60

[Lohmiller and Slotine 1998] Lohmiller, Winfried ; Slotine, Jean-Jacques E.: On contraction analysis for non-linear systems. In: *Automatica* 34 (1998), Nr. 6, S. 683–696 275

[Loshchilov and Hutter 2016] Loshchilov, Ilya ; Hutter, Frank: SGDR: Stochastic Gradient Descent with Warm Restarts. In: *International Conference on Learning Representations*, 2016 182, 280, 346

[Loshchilov and Hutter 2018] Loshchilov, Ilya ; Hutter, Frank: Decoupled Weight Decay Regularization. In: *International Conference on Learning Representations*, 2018 280, 282, 347

[Lotte 2014] Lotte, Fabien: A tutorial on EEG signal-processing techniques for mental-state recognition in brain–computer interfaces. In: *Guide to brain-computer music interfacing* (2014), S. 133–161 145

[Lotze and Halsband 2006] Lotze, Martin ; Halsband, Ulrike: Motor imagery. In: *Journal of Physiology-paris* 99 (2006), Nr. 4-6, S. 386–395 141

[Lozano et al. 1992] Lozano, Rogelio ; Brogliato, Bernard et al.: Adaptive control of robot manipulators with flexible joints. In: *IEEE Transactions on Automatic Control* 37 (1992), Nr. 2, S. 174–181 15, 21

[Lucci et al. 2020] Lucci, Niccolò ; Lacevic, Bakir ; Zanchettin, Andrea M. ; Rocco, Paolo: Combining speed and separation monitoring with power and force limiting for safe collaborative robotics applications. In: *IEEE Robotics and Automation Letters* 5 (2020), Nr. 4, S. 6121–6128 2

[Luo et al. 2017] Luo, Ming ; Skorina, Erik H. ; Tao, Weijia ; Chen, Fuchen ; Ozel, Selim ; Sun, Yinan ; Onal, Cagdas D.: Toward Modular Soft Robotics: Proprioceptive Curvature Sensing and Sliding-Mode Control of Soft Bidirectional Bending Modules. In: *Soft robotics* 4 (2017), Nr. 2, S. 117–125 173

[Lutter et al. 2019] Lutter, M ; Ritter, C ; Peters, Jan: Deep Lagrangian Networks: Using Physics as Model Prior for Deep Learning. In: *International Conference on Learning Representations (ICLR 2019)* OpenReview. net, 2019 7, 11, 15, 221, 430

[Madea and Brinkmann 2019] Madea, Burkhard ; Brinkmann, Bernd: *Handbuch gerichtliche Medizin: Band 2*. Bd. 2. Springer-Verlag, 2019 60

[Mahmud et al. 2020] Mahmud, Saifuddin ; Lin, Xiangxu ; Kim, Jong-Hoon: Interface for human machine interaction for assistant devices: A review. In: *2020 10th Annual computing and communication workshop and conference (CCWC)* IEEE, 2020, S. 0768–0773 147

[Maithripala and Berg 2015] Maithripala, DH S. ; Berg, Jordan M.: An intrinsic PID controller for mechanical systems on Lie groups. In: *Automatica* 54 (2015), S. 189–200 16

[Majidi 2014] Majidi, Carmel: Soft robotics: a perspective—current trends and prospects for the future. In: *Soft robotics* 1 (2014), Nr. 1, S. 5–11 3, 57, 323, 331

[Makoviychuk et al. 2021] Makoviychuk, Viktor ; Wawrzyniak, Lukasz ; Guo, Yunrong ; Lu, Michelle ; Storey, Kier ; Macklin, Miles ; Hoeller, David ; Rudin, Nikita ; Allshire, Arthur ; Handa, Ankur et al.: Isaac gym: High performance gpu-based physics simulation for robot learning. In: *arXiv preprint arXiv:2108.10470* (2021) 363

[Malas et al. 2024] Malas, Dionysios ; Wang, Shuai ; Huang, Wei ; Lindenroth, Lukas ; Xia, Wenfeng ; Liu, Hongbin: A Novel Pneudraulic Actuation Method to Enhance Soft Robot Control. In: *Soft Robotics* (2024) 14

[Mamakoukas et al. 2019] Mamakoukas, Giorgos ; Castano, Maria ; Tan, Xiaobo ; Murphey, Todd: Local Koopman operators for data-driven control of robotic systems. In: *Robotics: science and systems*, 2019 221

[Manchester 2011] Manchester, Ian R.: Transverse dynamics and regions of stability for nonlinear hybrid limit cycles. In: *IFAC Proceedings Volumes* 44 (2011), Nr. 1, S. 6285–6290 273, 276

[Manchester and Slotine 2014] Manchester, Ian R. ; Slotine, Jean-Jacques E.: Transverse contraction criteria for existence, stability, and robustness of a limit cycle. In: *Systems & Control Letters* 63 (2014), S. 32–38 275, 276, 277, 279

[Manchester and Slotine 2017] Manchester, Ian R. ; Slotine, Jean-Jacques E.: Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design. In: *IEEE Transactions on Automatic Control* 62 (2017), Nr. 6, S. 3046–3053 279

[Manning] Manning, Catherine G.: *Technology Readiness Levels*. https://www.nasa.gov/directorates/somd/space-communications-navigation-program/technology-readiness-levels/, . – Accessed: 2025-03-13 325, 327

[Mansfeld et al. 2018] Mansfeld, Nico ; Hamad, Mazin ; Becker, Marvin ; Marin, Antonio G. ; Haddadin, Sami: Safety map: A unified representation for biomechanics impact data and robot instantaneous dynamic properties. In: *IEEE Robotics and Automation Letters* 3 (2018), Nr. 3, S. 1880–1887 58, 313

[Marchese et al. 2015] Marchese, Andrew D. ; Katzschmann, Robert K. ; Rus, Daniela: A recipe for soft fluidic elastomer robots. In: *Soft Robotics* 2 (2015), Nr. 1, S. 7–25 14, 187, 362

[Marchese et al. 2014] Marchese, Andrew D. ; Komorowski, Konrad ; Onal, Cagdas D. ; Rus, Daniela: Design and control of a soft and continuously deformable 2d robotic manipulation system. In: *2014 IEEE international conference on robotics and automation (ICRA)* Ieee, 2014, S. 2189–2196 14, 157, 315, 361

[Marchese and Rus 2016] Marchese, Andrew D. ; Rus, Daniela: Design, kinematics, and control of a soft spatial fluidic elastomer manipulator. In: *The International Journal of Robotics Research* 35 (2016), Nr. 7, S. 840–869 14, 155, 315

[Marino 1985] Marino, Riccardo: High-gain feedback in non-linear control systems. In: *International Journal of Control* 42 (1985), Nr. 6, S. 1369–1385 41

[Marquardt 1963] Marquardt, Donald W.: An algorithm for least-squares estimation of nonlinear parameters. In: *Journal of the society for Industrial and Applied Mathematics* 11 (1963), Nr. 2, S. 431–441 13, 89, 366, 367

[Marques Monteiro et al. 2024] Marques Monteiro, Richard ; Shi, Jialei ; Wurdemann, Helge ; Iida, Fumiya ; George Thuruthel, Thomas: Visuo-dynamic self-modelling of soft robotic systems. In: *Frontiers in Robotics and AI* 11 (2024), S. 1403733 6

[Mathew et al. 2025] Mathew, Anup T. ; Feliu-Talegon, Daniel ; Alkayas, Abdulaziz Y. ; Boyer, Frederic ; Renda, Federico: Reduced order modeling of hybrid soft-rigid robots using global, local, and state-dependent strain parameterization. In: *The International Journal of Robotics Research* 44 (2025), Nr. 1, S. 129–154 26, 29, 30, 31

[Mathew et al. 2022] Mathew, Anup T. ; Hmida, Ikhlas Mohamed B. ; Armanini, Costanza ; Boyer, Frederic ; Renda, Federico: SoRoSim: A MATLAB Toolbox for Hybrid Rigid-Soft Robots Based on the Geometric Variable-Strain Approach. In: *IEEE Robotics & Automation Magazine* (2022) 101, 363

[Matsuo et al. 2022] Matsuo, Yutaka ; LeCun, Yann ; Sahani, Maneesh ; Precup, Doina ; Silver, David ; Sugiyama, Masashi ; Uchibe, Eiji ; Morimoto, Jun: Deep learning, reinforcement learning, and world models. In: *Neural Networks* 152 (2022), S. 267–275 221

[Mazzolai and Laschi 2020] Mazzolai, Barbara ; Laschi, Cecilia: A vision for future bioinspired and biohybrid robots. In: *Science robotics* 5 (2020), Nr. 38, S. eaba6893 331, 334, 336

[Mazzolai et al. 2022] Mazzolai, Barbara ; Mondini, Alessio ; Del Dottore, Emanuela ; Margheri, Laura ; Carpi, Federico ; Suzumori, Koichi ; Cianchetti, Matteo ; Speck, Thomas ; Smoukov, Stoyan K. ; Burgert, Ingo et al.: Roadmap on soft robotics: multifunctionality, adaptability and growth without borders. In: *Multifunctional Materials* 5 (2022), Nr. 3, S. 032001 3, 57, 323

[Medvet et al. 2021] Medvet, Eric ; Bartoli, Alberto ; Pigozzi, Federico ; Rochelli, Marco: Biodiversity in evolved voxel-based soft robots. In: *Proceedings of the Genetic and Evolutionary Computation Conference*, 2021, S. 129–137 323, 324, 329

[Medvet et al. 2022] Medvet, Eric ; Nadizar, Giorgia ; Pigozzi, Federico: On the impact of body material properties on neuroevolution for embodied agents: The case of voxel-based soft robots. In: *Proceedings of the genetic and evolutionary computation conference companion*, 2022, S. 2122–2130 323, 324, 329, 330, 334, 336

[Ménager et al. 2023] Ménager, Etienne ; Navez, Tanguy ; Goury, Olivier ; Duriez, Christian: Direct and inverse modeling of soft robots by learning a condensed FEM model. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)* IEEE, 2023, S. 530–536 332, 334, 340

[Mengaldo et al. 2022] Mengaldo, Gianmarco ; Renda, Federico ; Brunton, Steven L. ; Bächer, Moritz ; Calisti, Marcello ; Duriez, Christian ; Chirikjian, Gregory S. ;

Laschi, Cecilia: A concise guide to modelling the physics of embodied intelligence in soft robotics. In: *Nature Reviews Physics* 4 (2022), Nr. 9, S. 595–610 9, 316, 332, 334

[Meurer et al. 2017] Meurer, Aaron ; Smith, Christopher P. ; Paprocki, Mateusz ; Čertík, Ondřej ; Kirpichev, Sergey B. ; Rocklin, Matthew ; Kumar, AMiT ; Ivanov, Sergiu ; Moore, Jason K. ; Singh, Sartaj ; Rathnayake, Thilina ; Vig, Sean ; Granger, Brian E. ; Muller, Richard P. ; Bonazzi, Francesco ; Gupta, Harsh ; Vats, Shivam ; Johansson, Fredrik ; Pedregosa, Fabian ; Curry, Matthew J. ; Terrel, Andy R. ; Roučka, Štěpán ; Saboo, Ashutosh ; Fernando, Isuru ; Kulal, Sumith ; Cimrman, Robert ; Scopatz, Anthony: SymPy: symbolic computing in Python. 3 (2017), Januar, e103. http://dx.doi.org/10.7717/peerj-cs.103. – DOI 10.7717/peerj–cs.103. – ISSN 2376–5992 363, 364

[Milojevic et al. 2025] Milojevic, Dejan ; Zardini, Gioele ; Elser, Miriam ; Censi, Andrea ; Frazzoli, Emilio: CODEI: Resource-Efficient Task-Driven Co-Design of Perception and Decision Making for Mobile Robots Applied to Autonomous Vehicles. In: *IEEE Transactions on Robotics* (2025), S. 1–20. http://dx.doi.org/10.1109/TRO.2025.3552347. – DOI 10.1109/TRO.2025.3552347 324

[Miriyev et al. 2017] Miriyev, Aslan ; Stack, Kenneth ; Lipson, Hod: Soft material for soft actuators. In: *Nature communications* 8 (2017), Nr. 1, S. 596 331

[Mitchell et al. 2021] Mitchell, Mason D. ; Hurley, Forrest E. ; Onal, Cagdas D.: Fast Probabilistic 3-D Curvature Proprioception with a Magnetic Soft Sensor. In: *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)* IEEE, 2021, S. 215–220 173

[Mohammadi et al. 2024] Mohammadi, Hadi B. ; Hauberg, Søren ; Arvanitidis, Georgios ; Neumann, Gerhard ; Rozo, Leonel: Extended Neural Contractive Dynamical Systems: On Multiple Tasks and Riemannian Safety Regions. In: *arXiv preprint arXiv:2411.11405* (2024) 261, 301

[Morimoto et al. 2021] Morimoto, Ryota ; Nishikawa, Satoshi ; Niiyama, Ryuma ; Kuniyoshi, Yasuo: Model-free reinforcement learning with ensemble for a soft continuum robot arm. In: *2021 IEEE 4th International Conference on Soft Robotics (RoboSoft)* IEEE, 2021, S. 141–148 4, 5

[Mur-Artal and Tardós 2017] Mur-Artal, Raul ; Tardós, Juan D.: Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. In: *IEEE Transactions on Robotics* 33 (2017), Nr. 5, S. 1255–1262 82, 83, 86, 88, 89, 91

[Muthukrishnan et al. 2020] Muthukrishnan, Nikesh ; Maleki, Farhad ; Ovens, Katie ; Reinhold, Caroline ; Forghani, Behzad ; Forghani, Reza et al.: Brief history of artificial intelligence. In: *Neuroimaging Clinics of North America* 30 (2020), Nr. 4, S. 393–399 313

[Muttray et al. 2014a] Muttray, A ; Melia, M ; Geissler, B ; König, J ; Letzel, S: Collaborative robots—investigation of pain sensibility at the man-machine-interface. In: *Mainz, Germany. Retrieved March* 28 (2014), S. 2021 76

[Muttray et al. 2014b] Muttray, A ; Melia, M ; Geissler, B ; König, J ; Letzel, S: Kollaborierende Roboter–Ermittlung der Schmerzempfindlichkeit an der Mensch-Maschine-Schnittstelle: Wissenschaftlicher Schlussbericht zum Vorhaben FP-0317 / Tech. rep. Mainz: Johannes-Gutenberg-Universität Mainz. 2014 (1). – Forschungsbericht. – - 60, 313

[Nadizar et al. 2022] Nadizar, Giorgia ; Medvet, Eric ; Miras, Karine: On the schedule for morphological development of evolved modular soft robots. In: *European conference on genetic programming (Part of EvoStar)* Springer, 2022, S. 146–161 336

[Nah et al. 2025] Nah, Moses C. ; Lachner, Johannes ; Hogan, Neville ; Slotine, Jean-Jacques: Combining Movement Primitives with Contraction Theory. In: *arXiv preprint arXiv:2501.09198* (2025) 261, 262, 264, 267, 276

[Nahavandi 2019] Nahavandi, Saeid: Industry 5.0—A human-centric solution. In: *Sustainability* 11 (2019), Nr. 16, S. 4371 2, 323

[Naughton et al. 2021] Naughton, Noel ; Sun, Jiarui ; Tekinalp, Arman ; Parthasarathy, Tejaswin ; Chowdhary, Girish ; Gazzola, Mattia: Elastica: A compliant mechanics environment for soft robotic control. In: *IEEE Robotics and Automation Letters* 6 (2021), Nr. 2, S. 3389–3396 14, 20, 100, 101, 363

[Navez 2024] Navez, Tanguy: *Contributions to the Concept of Embodied Intelligence in Soft Robotics through Control and Design Co-Optimization*, Université de Lille, Diss., 2024 23, 76, 323, 328, 329, 334, 336, 340

[Navez et al. 2024] Navez, Tanguy ; Liévin, Baptiste ; Peyron, Quentin ; Navarro, Stefan E. ; Goury, Olivier ; Duriez, Christian: Design optimization of a soft gripper using self-contacts. In: *2024 IEEE 7th International Conference on Soft Robotics (RoboSoft)* IEEE, 2024, S. 1054–1060 328, 330, 336, 342

[Navez et al. 2025] Navez, Tanguy ; Ménager, Etienne ; Chaillou, Paul ; Goury, Olivier ; Kruszewski, Alexandre ; Duriez, Christian: Modeling, Embedded Control and Design of Soft Robots using a Learned Condensed FEM Model. In: *IEEE Transactions on Robotics* (2025) 324, 331, 340

[Nawaz et al. 2024] Nawaz, Farhad ; Li, Tianyu ; Matni, Nikolai ; Figueroa, Nadia: Learning Complex Motion Plans using Neural ODEs with Safety and Stability Guarantees. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)* IEEE, 2024, S. 17216–17222 265, 280, 283, 287, 301

[Nørskov et al. 2009] Nørskov, Jens K. ; Bligaard, Thomas ; Rossmeisl, Jan ; Christensen, Claus H.: Towards the computational design of solid catalysts. In: *Nature chemistry* 1 (2009), Nr. 1, S. 37–46 328

[Ofir et al. 2022] Ofir, Ron ; Margaliot, Michael ; Levron, Yoash ; Slotine, Jean-Jacques: A sufficient condition for k-contraction of the series connection of two systems. In: *IEEE Transactions on Automatic Control* 67 (2022), Nr. 9, S. 4994–5001 300

[Ortega et al. 2021] Ortega, Romeo ; Romero, José G. ; Borja, Pablo ; Donaire, Alejandro: *PID passivity-based control of nonlinear systems with applications.* John Wiley & Sons, 2021 221, 222, 248

[Ortner and Coliado Bandeira 2020] Ortner, Michael ; Coliado Bandeira, Lucas G.: Magpylib: A free Python package for magnetic field computation. In: *SoftwareX* (2020). http://dx.doi.org/10.1016/j.softx.2020.100466. – DOI 10.1016/j.softx.2020.100466 181, 185

[Ott 2008] Ott, Christian: *Cartesian impedance control of redundant and flexible-joint robots.* Springer, 2008 33, 34, 52, 131, 366

[Ozel et al. 2015] Ozel, Selim ; Keskin, Nehir A. ; Khea, Darien ; Onal, Cagdas D.: A precise embedded curvature sensor module for soft-bodied robots. In: *Sensors and Actuators A: Physical* 236 (2015), S. 349–356 173

[O'Neill et al. 2024] O'Neill, Abby ; Rehman, Abdul ; Maddukuri, Abhiram ; Gupta, Abhishek ; Padalkar, Abhishek ; Lee, Abraham ; Pooley, Acorn ; Gupta, Agrim ; Mandlekar, Ajay ; Jain, Ajinkya et al.: Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)* IEEE, 2024, S. 6892–6903 261, 294, 317, 342

[Paden and Panja 1988] Paden, Brad ; Panja, Ravi: Globally asymptotically stable 'PD+'controller for robot manipulators. In: *International Journal of Control* 47 (1988), Nr. 6, S. 1697–1712 45, 366

[Pagliarani et al. 2024] Pagliarani, Niccolo ; Arleo, Luca ; De Luca, Giovanna ; Pozzi, Jasmine ; Cianchetti, Matteo: Variable stiffness structure inspired by seashells. In: *Smart Materials and Structures* 33 (2024), Nr. 2, S. 025004 328, 336

[Park et al. 2011] Park, Jung-Jun ; Haddadin, Sami ; Song, Jae-Bok ; Albu-Schäffer, Alin: Designing optimally safe robot surface properties for minimizing the stress characteristics of human-robot collisions. In: *2011 IEEE International Conference on Robotics and Automation* IEEE, 2011, S. 5413–5420 58, 65

[Parlikar et al. 2024] Parlikar, Soham ; Zielińska, Teresa ; Gumowski, Konrad: Concept of Pneumatic Soft Robot: Suction-Driven Locomotion. In: *Climbing and Walking Robots Conference* Springer, 2024, S. 179–190 14

[Pas 2023] Pas, Pieter: *The Damped Harmonic Oscillator.* https://tttapa.github.io/Pages/Arduino/Audio-and-Signal-Processing/VU-Meters/Damped-Harmonic-Oscillator.html. Version: Mar 2023 233, 234

[Pasquier et al. 2025] Pasquier, Cosima d. ; Grannen, Jennifer ; Pan, Chuer ; Huber, Serin L. ; Smith, Aliyah ; Kennedy, Monroe ; Song, Shuran ; Sadigh, Dorsa ; Okamura, Allison M.: A Study of Perceived Safety for Soft Robotics in Caregiving Tasks. In: *arXiv preprint arXiv:2503.20916* (2025) 313, 323

[Paszke et al. 2019] Paszke, Adam ; Gross, Sam ; Massa, Francisco ; Lerer, Adam ; Bradbury, James ; Chanan, Gregory ; Killeen, Trevor ; Lin, Zeming ; Gimelshein, Natalia ; Antiga, Luca ; Desmaison, Alban ; Kopf, Andreas ; Yang, Edward ; DeVito, Zachary ; Raison, Martin ; Tejani, Alykhan ; Chilamkurthy, Sasank ; Steiner, Benoit ; Fang, Lu ; Bai, Junjie ; Chintala, Soumith: PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Wallach, H. (Hrsg.) ; Larochelle, H. (Hrsg.) ; Beygelzimer, A. (Hrsg.) ; Alché-Buc, F. d'(Hrsg.) ; Fox, E. (Hrsg.) ; Garnett, R. (Hrsg.): *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, S. 8024–8035 363

[Patterson et al. 2024] Patterson, Zach J. ; Sologuren, Emily ; Della Santina, Cosimo ; Rus, Daniela: Design and Control of Modular Soft-Rigid Hybrid Manipulators with Self-Contact. In: *arXiv preprint arXiv:2408.09275* (2024) 45

[Pekmezci et al. 2024] Pekmezci, Mehmet ; Ugur, Emre ; Oztop, Erhan: Coupled Conditional Neural Movement Primitives. In: *Neural Computing and Applications* 36 (2024), Nr. 30, S. 18999–19021 261

[Pérez-Dattari et al. 2024] Pérez-Dattari, Rodrigo ; Della Santina, Cosimo ; Kober, Jens: PUMA: Deep Metric Imitation Learning for Stable Motion Primitives. In: *Advanced Intelligent Systems* 6 (2024), Nr. 11, S. 2400144 261, 264, 295

[Pérez-Dattari and Kober 2023] Pérez-Dattari, Rodrigo ; Kober, Jens: Stable motion primitives via imitation and contrastive learning. In: *IEEE Transactions on Robotics* (2023) 18, 258, 259, 261, 264, 266, 283, 287, 295, 302

[Petersen et al. 2008] Petersen, Kaare B. ; Pedersen, Michael S. et al.: The matrix cookbook. In: *Technical University of Denmark* 7 (2008), Nr. 15, S. 510 278, 344

[Pfurtscheller and Da Silva 1999] Pfurtscheller, Gert ; Da Silva, FH L.: Event-related EEG/MEG synchronization and desynchronization: basic principles. In: *Clinical neurophysiology* 110 (1999), Nr. 11, S. 1842–1857 148

[Phillips et al. 2021] Phillips, Trevor ; Stölzle, Maximilian ; Turricelli, Erick ; Achermann, Florian ; Lawrance, Nicholas ; Siegwart, Roland ; Chung, Jen J.: Learn to Path: Using neural networks to predict Dubins path characteristics for aerial vehicles in wind. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)* IEEE, 2021, S. 1073–1079 434

[Pierallini et al. 2023] Pierallini, Michele ; Stella, Francesco ; Angelini, Franco ; Deutschmann, Bastian ; Hughes, Josie ; Bicchi, Antonio ; Garabini, Manolo ; Della Santina, Cosimo: A provably stable iterative learning controller for continuum soft robots. In: *IEEE Robotics and Automation Letters* (2023) 4, 5

[Pinskier et al. 2024] Pinskier, Josh ; Wang, Xing ; Liow, Lois ; Xie, Yue ; Kumar, Prabhat ; Langelaar, Matthijs ; Howard, David: Diversity-Based Topology Optimization of Soft Robotic Grippers. In: *Advanced Intelligent Systems* 6 (2024), Nr. 4, S. 2300505 328, 336

[Polygerinos et al. 2017] Polygerinos, Panagiotis ; Correll, Nikolaus ; Morin, Stephen A. ; Mosadegh, Bobak ; Onal, Cagdas D. ; Petersen, Kirstin ; Cianchetti, Matteo ; Tolley, Michael T. ; Shepherd, Robert F.: Soft Robotics: Review of Fluid-Driven Intrinsically Soft Devices; Manufacturing, Sensing, Control, and Applications in Human-Robot Interaction. In: *Advanced Engineering Materials* (2017) 82

[Polygerinos et al. 2015] Polygerinos, Panagiotis ; Wang, Zheng ; Overvelde, Johannes T. ; Galloway, Kevin C. ; Wood, Robert J. ; Bertoldi, Katia ; Walsh, Conor J.: Modeling of soft fiber-reinforced bending actuators. In: *IEEE Transactions on Robotics* 31 (2015), Nr. 3, S. 778–789 328, 336

[Popovic et al. 2002] Popovic, RS ; Drljaca, PM ; Schott, C: Bridging the gap between AMR, GMR, and Hall magnetic sensors. In: *2002 23rd International Conference on Microelectronics* Bd. 1 IEEE, 2002, S. 55–58 187

[Pratt and Williamson 1995] Pratt, Gill A. ; Williamson, Matthew M.: Series elastic actuators. In: *Proceedings 1995 IEEE/RSJ international conference on intelligent robots and systems. Human robot interaction and cooperative robots* Bd. 1 IEEE, 1995, S. 399–406 2

[Pupa and Secchi 2024] Pupa, Andrea ; Secchi, Cristian: Efficient ISO/TS 15066 Compliance through Model Predictive Control. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)* IEEE, 2024, S. 17358–17364 58, 59, 62

[Pustina 2025] Pustina, Pietro: *Analysis and control of the underactuation in continuum soft robots: a kinematic independent approach*, Università degli Studi di Roma" La Sapienza", Diss., 2025 32, 37, 38, 42, 43, 45, 46, 47, 48, 49

[Pustina et al. 2022a] Pustina, Pietro ; Borja, Pablo ; Della Santina, Cosimo ; De Luca, Alessandro: P-satI-D Shape Regulation of Soft Robots. In: *IEEE Robotics and Automation Letters* 8 (2022), Nr. 1, S. 1–8 8, 11, 14, 16, 19, 25, 37, 41, 118, 119, 210, 248, 310, 312

[Pustina et al. 2024a] Pustina, Pietro ; Della Santina, Cosimo ; Boyer, Frédéric ; De Luca, Alessandro ; Renda, Federico: Input Decoupling of Lagrangian Systems via Coordinate Transformation: General Characterization and its Application to Soft Robotics. In: *IEEE Transactions on Robotics* (2024) 3, 25, 32, 34, 35, 37, 40, 41, 42, 118, 120, 309, 310

[Pustina et al. 2022b] Pustina, Pietro ; Della Santina, Cosimo ; De Luca, Alessandro: Feedback regulation of elastically decoupled underactuated soft robots. In: *IEEE Robotics and Automation Letters* 7 (2022), Nr. 2, S. 4512–4519 37, 40, 42

[Pustina et al. 2024b] Pustina, Pietro ; Della Santina, Cosimo ; De Luca, Alessandro: Unified Inverse Dynamics of Modular Serial Mechanical Systems with Application to Soft Robotics. In: *arXiv preprint arXiv:2402.07037* (2024) 51

[Rana et al. 2020] Rana, Muhammad A. ; Li, Anqi ; Fox, Dieter ; Boots, Byron ; Ramos, Fabio ; Ratliff, Nathan: Euclideanizing flows: Diffeomorphic reduction for learning stable dynamical systems. In: *Learning for Dynamics and Control* PMLR, 2020, S. 630–639 18, 22, 258, 259, 261, 262, 264, 265, 266, 267, 268, 269, 274, 275, 279, 295, 302, 303, 317

[Rao 2024] Rao, Priyanka: *Tendon Driven Continuum Robots: Modeling and Motion Planning for Contact Aided Navigation and Shape Locking*, University of Toronto (Canada), Diss., 2024 311

[Rao et al. 2024] Rao, Priyanka ; Salzman, Oren ; Burgner-Kahrs, Jessica: Towards Contact-Aided Motion Planning for Tendon-Driven Continuum Robots. In: *IEEE Robotics and Automation Letters* (2024) 311

[Renda et al. 2020] Renda, Federico ; Armanini, Costanza ; Lebastard, Vincent ; Candelier, Fabien ; Boyer, Frederic: A geometric variable-strain approach for static modeling of soft manipulators with tendon and fluidic actuation. In: *IEEE Robotics and Automation Letters* 5 (2020), Nr. 3, S. 4006–4013 3, 13, 29, 30, 65, 99, 107, 340, 363

[Renda et al. 2018] Renda, Federico ; Boyer, Frédéric ; Dias, Jorge ; Seneviratne, Lakmal: Discrete Cosserat approach for multisection soft manipulator dynamics. In: *IEEE Transactions on Robotics* 34 (2018), Nr. 6, S. 1518–1533 3, 14, 15, 20, 25, 28, 29, 65, 71, 75, 83, 84, 99, 100, 106, 107, 112, 115, 195, 196, 198, 340, 363, 364

[Renda et al. 2016] Renda, Federico ; Cacucciolo, Vito ; Dias, Jorge ; Seneviratne, Lakmal: Discrete Cosserat approach for soft robot dynamics: A new piece-wise constant strain model with torsion and shears. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* IEEE, 2016, S. 5495–5502 28, 243

[Rezende and Mohamed 2015] Rezende, Danilo ; Mohamed, Shakir: Variational inference with normalizing flows. In: *International conference on machine learning* PMLR, 2015, S. 1530–1538 282

[Ribeiro et al. 2025] Ribeiro, Lucas N. ; Borja, Pablo ; Santina, Cosimo D. ; Deutschmann, Bastian: Singular-Perturbation Control of a Tendon-Driven Soft Robot: Theory and Experiments. In: *IEEE Transactions on Control Systems Technology* (2025), Nr. 99, S. 1–8 168

[Rieffel et al. 2014] Rieffel, John ; Knox, Davis ; Smith, Schuyler ; Trimmer, Barry: Growing and evolving soft robots. In: *Artificial life* 20 (2014), Nr. 1, S. 143–162 323

[Roc et al. 2021] Roc, Aline ; Pillette, Lea ; Mladenovic, Jelena ; Benaroch, Camille ; N'Kaoua, Bernard ; Jeunet, Camille ; Lotte, Fabien: A review of user training methods in brain computer interfaces based on mental tasks. In: *Journal of Neural Engineering* 18 (2021), Nr. 1, S. 011002 146

[Ronneberger et al. 2015] Ronneberger, Olaf ; Fischer, Philipp ; Brox, Thomas: U-net: Convolutional networks for biomedical image segmentation. In: *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18* Springer, 2015, S. 234–241 282

[Roozenburg and Eekels 1995] Roozenburg, Norbert F. ; Eekels, Johannes: *Product Design: Fundamentals and Methods*. Wiley, 1995 327

[Rosi et al. 2022] Rosi, Emanuele R. ; Stölzle, Maximilian ; Solari, Fabio ; Della Santina, Cosimo: Sensing soft robots' shape with cameras: an investigation on kinematics-aware SLAM. In: *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)* IEEE, 2022, S. 795–801 81, 173, 198, 244, 430, 434

[Roth et al. 2025] Roth, Pascal ; Frey, Jonas ; Cadena, Cesar ; Hutter, Marco: Learned Perceptive Forward Dynamics Model for Safe and Platform-aware Robotic Navigation. In: *arXiv preprint arXiv:2504.19322* (2025) 221

[Royakkers and Van Est 2015] Royakkers, Lambèr ; Van Est, Rinie: A literature review on new robotics: automation from love to war. In: *International journal of social robotics* 7 (2015), S. 549–570 2, 323

[Ruder 2016] Ruder, Sebastian: An overview of gradient descent optimization algorithms. In: *arXiv preprint arXiv:1609.04747* (2016) 182

[Rumelhart et al. 1986] Rumelhart, David E. ; Hinton, Geoffrey E. ; Williams, Ronald J.: Learning representations by back-propagating errors. In: *nature* 323 (1986), Nr. 6088, S. 533–536 7

[Rus and Tolley 2015] Rus, Daniela ; Tolley, Michael T.: Design, fabrication and control of soft robots. In: *Nature* 521 (2015), Nr. 7553, S. 467–475 2, 9, 57, 62, 139, 323

[Rusch and Mishra 2020] Rusch, T K. ; Mishra, Siddhartha: Coupled Oscillatory Recurrent Neural Network (coRNN): An accurate and (gradient) stable architecture for learning long time dependencies. In: *International Conference on Learning Representations*, 2020 221, 222, 223, 225, 239, 240, 349, 350, 351

[Rusch and Mishra 2021] Rusch, T K. ; Mishra, Siddhartha: Unicornn: A recurrent model for learning very long time dependencies. In: *International Conference on Machine Learning* PMLR, 2021, S. 9168–9178 221, 223, 225

[Rusch and Rus 2024] Rusch, T K. ; Rus, Daniela: Oscillatory State-Space Models. In: *arXiv preprint arXiv:2410.03943* (2024) 221

[Sabelhaus and Majidi 2021] Sabelhaus, Andrew P. ; Majidi, Carmel: Gaussian process dynamics models for soft robots with shape memory actuators. In: *2021 IEEE 4th International Conference on Soft Robotics (RoboSoft)* IEEE, 2021, S. 191–198 10

[Sadati et al. 2021] Sadati, SM H. ; Naghibi, S E. ; Shiva, Ali ; Michael, Brendan ; Renson, Ludovic ; Howard, Matthew ; Rucker, Caleb D. ; Althoefer, Kaspar ; Nanayakkara, Thrishantha ; Zschaler, Steffen et al.: TMTDyn: A Matlab package for modeling and control of hybrid rigid–continuum robots based on discretized lumped systems and reduced-order models. In: *The International Journal of Robotics Research* 40 (2021), Nr. 1, S. 296–347 99, 155, 156, 363

[Sakoe and Chiba 1978] Sakoe, H. ; Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26 (1978), Nr. 1, S. 43–49. http://dx.doi.org/10.1109/TASSP.1978.1163055. – DOI 10.1109/TASSP.1978.1163055 283

[Saveriano et al. 2023] Saveriano, Matteo ; Abu-Dakka, Fares J. ; Kramberger, Aljaž ; Peternel, Luka: Dynamic movement primitives in robotics: A tutorial survey. In: *The International Journal of Robotics Research* 42 (2023), Nr. 13, S. 1133–1184 17, 261, 262, 264

[Savitzky and Golay 1964] Savitzky, Abraham ; Golay, Marcel J.: Smoothing and differentiation of data by simplified least squares procedures. In: *Analytical chemistry* 36 (1964), Nr. 8, S. 1627–1639 365

[Schaal 1999] Schaal, Stefan: Is imitation learning the route to humanoid robots? In: *Trends in cognitive sciences* 3 (1999), Nr. 6, S. 233–242 261

[Schäfke et al. 2024] Schäfke, Hendrik ; Habich, Tim-Lukas ; Muhmann, Christian ; Ehlers, Simon F. ; Seel, Thomas ; Schappler, Moritz: Learning-based Nonlinear Model Predictive Control of Articulated Soft Robots using Recurrent Neural Networks. In: *IEEE Robotics and Automation Letters* (2024) 10, 11

[Schiatti et al. 2017] Schiatti, Lucia ; Tessadori, Jacopo ; Barresi, Giacinto ; Mattos, Leonardo S. ; Ajoudani, Arash: Soft brain-machine interfaces for assistive robotics: A novel control approach. In: *2017 International Conference on Rehabilitation Robotics (ICORR)* IEEE, 2017, S. 863–869 139, 148

[Schmitt et al. 2018] Schmitt, François ; Piccin, Olivier ; Barbé, Laurent ; Bayle, Bernard: Soft robots manufacturing: A review. In: *Frontiers in Robotics and AI* 5 (2018), S. 84 331

[Schwarting et al. 2021] Schwarting, Wilko ; Seyde, Tim ; Gilitschenski, Igor ; Liebenwein, Lucas ; Sander, Ryan ; Karaman, Sertac ; Rus, Daniela: Deep Latent Competition: Learning to Race Using Visual Control Policies in Latent Space. In: *Conference on Robot Learning* PMLR, 2021, S. 1855–1870 221

[Sciavicco and Siciliano 2012] Sciavicco, Lorenzo ; Siciliano, Bruno: *Modelling and control of robot manipulators*. Springer Science & Business Media, 2012 11, 16, 38, 310

[Scimeca et al. 2019] Scimeca, Luca ; Hughes, Josie ; Maiolino, Perla ; Iida, Fumiya: Model-free soft-structure reconstruction for proprioception using tactile arrays. In: *IEEE Robotics and Automation Letters* 4 (2019), Nr. 3, S. 2479–2484 82, 173

[Seker et al. 2019] Seker, Muhammet Y. ; Imre, Mert ; Piater, Justus H. ; Ugur, Emre: Conditional Neural Movement Primitives. In: *Robotics: Science and Systems* Bd. 10, 2019 262

[Selvaggio et al. 2020] Selvaggio, Mario ; Ramirez, LA ; Naclerio, Nicholas D. ; Siciliano, Bruno ; Hawkes, Elliot W.: An obstacle-interaction planning method for navigation of actuated vine robots. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)* IEEE, 2020, S. 3227–3233 311

[Sermanet et al. 2025] Sermanet, Pierre ; Majumdar, Anirudha ; Irpan, Alex ; Kalashnikov, Dmitry ; Sindhwani, Vikas: *Generating Robot Constitutions & Benchmarks for Semantic Safety*. https://arxiv.org/abs/2503.08663. Version: 2025 2

[Shampine 1986] Shampine, Lawrence F.: Some Practical Runge-Kutta Formulas. In: *Mathematics of Computation* 46 (1986), Nr. 173, S. 135–150. http://dx.doi.org/https://doi.org/10.2307/2008219. – DOI https://doi.org/10.2307/2008219 365

[Shao et al. 2023] Shao, Xiangyu ; Pustina, Pietro ; Stölzle, Maximilian ; Sun, Guanghui ; De Luca, Alessandro ; Wu, Ligang ; Della Santina, Cosimo: Model-based control for soft robots with system uncertainties and input saturation. In: *IEEE Transactions on Industrial Electronics* (2023) 195, 308, 315, 433

[She et al. 2020a] She, Yu ; Liu, Sandra Q. ; Yu, Peiyu ; Adelson, Edward: Exoskeleton-covered soft finger with vision-based proprioception and tactile sensing. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)* IEEE, 2020, S. 10075–10081 82

[She et al. 2020b] She, Yu ; Song, Siyang ; Su, Hai-Jun ; Wang, Junmin: A comparative study on the effect of mechanical compliance for a safe physical human–robot interaction. In: *Journal of Mechanical Design* 142 (2020), Nr. 6, S. 063305 65

[Shepherd et al. 2013] Shepherd, Robert F. ; Stokes, Adam A. ; Freake, Jacob ; Barber, Jabulani ; Snyder, Phillip W. ; Mazzeo, Aaron D. ; Cademartiri, Ludovico ; Morin, Stephen A. ; Whitesides, George M.: Using explosions to power a soft robot. In: *Angewandte Chemie International Edition* 52 (2013), Nr. 10, S. 2892–2896 3

[Shih et al. 2019] Shih, Benjamin ; Christianson, Caleb ; Gillespie, Kyle ; Lee, Sebastian ; Mayeda, Jason ; Huo, Zhaoyuan ; Tolley, Michael T.: Design considerations for 3D printed, soft, multimaterial resistive sensors for soft robotics. In: *Frontiers in Robotics and AI* 6 (2019), S. 30 173

[Shin et al. 2008] Shin, Dongjun ; Sardellitti, Irene ; Khatib, Oussama: A hybrid actuation approach for human-friendly robot design. In: *2008 IEEE international conference on robotics and automation* IEEE, 2008, S. 1747–1752 60

[Shintake et al. 2018] Shintake, Jun ; Piskarev, Yegor ; Jeong, Seung H. ; Floreano, Dario: Ultrastretchable strain sensors using carbon black-filled elastomer composites and comparison of capacitive versus resistive sensors. In: *Advanced Materials Technologies* 3 (2018), Nr. 3, S. 1700284 173

[Sholokhov et al. 2023] Sholokhov, Aleksei ; Liu, Yuying ; Mansour, Hassan ; Nabi, Saleh: Physics-informed neural ODE (PINODE): embedding physics into models using collocation points. In: *Scientific Reports* 13 (2023), Nr. 1, S. 10166 221

[Shukor et al. 2025] Shukor, Mustafa ; Aubakirova, Dana ; Capuano, Francesco ; Kooijmans, Pepijn ; Palma, Steven ; Zouitine, Adil ; Aractingi, Michel ; Pascal, Caroline ; Russi, Martino ; Marafioti, Andres et al.: Smolvla: A vision-language-action model for affordable and efficient robotics. In: *arXiv preprint arXiv:2506.01844* (2025) 302

[Siciliano et al. 2010] Siciliano, Bruno ; Sciavicco, Lorenzo ; Villani, Luigi ; Oriolo, Giuseppe: *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010 26, 27, 33, 332, 340

[Siéfert et al. 2019] Siéfert, Emmanuel ; Reyssat, Etienne ; Bico, José ; Roman, Benoît: Bio-inspired pneumatic shape-morphing elastomers. In: *Nature materials* 18 (2019), Nr. 1, S. 24–28 328

[Sigmund and Maute 2013] Sigmund, Ole ; Maute, Kurt: Topology optimization approaches: A comparative review. In: *Structural and multidisciplinary optimization* 48 (2013), Nr. 6, S. 1031–1055 328

[Simmoteit et al. 2025] Simmoteit, Ken-Joel ; Schillinger, Philipp ; Rozo, Leonel: Diffeomorphic Obstacle Avoidance for Contractive Dynamical Systems via Implicit Representations. In: *arXiv preprint arXiv:2504.18860* (2025) 301

[Slotine and Li 1987] Slotine, Jean-Jacques E. ; Li, Weiping: On the Adaptive Control of Robot Manipulators. In: *The International Journal of Robotics Research* 6 (1987), Nr. 3, 49-59. http://dx.doi.org/10.1177/027836498700600303. – DOI 10.1177/027836498700600303 50

[Slotine et al. 1991] Slotine, Jean-Jacques E. ; Li, Weiping et al.: *Applied nonlinear control*. Prentice hall Englewood Cliffs, NJ, 1991 46

[Sochopoulos et al. 2024] Sochopoulos, Andreas ; Gienger, Michael ; Vijayakumar, Sethu: Learning deep dynamical systems using stable neural ODEs. In: *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* IEEE, 2024, S. 11163–11170 266, 295

[Soleti et al. 2024] Soleti, Giovanni ; Massenio, Paolo R. ; Kunze, Julian ; Rizzello, Gianluca: Nonlinear Coordinate Transformation and Trajectory Tracking Control of an Underactuated Soft Robot Driven by Dielectric Elastomers. In: *2024 IEEE 7th International Conference on Soft Robotics (RoboSoft)* IEEE, 2024, S. 228–234 309

[Soleti et al. 2025] Soleti, Giovanni ; Massenio, Paolo R. ; Kunze, Julian ; Rizzello, Gianluca: Model-Based Robust Position Control of an Underactuated Dielectric Elastomer Soft Robot. In: *IEEE Transactions on Robotics* (2025), S. 1–18. http://dx.doi.org/10.1109/TRO.2025.3539184. – DOI 10.1109/TRO.2025.3539184 4, 135, 309, 314

[Song et al. 2024] Song, Junru ; Yang, Yang ; Peng, Wei ; Zhou, Weien ; Wang, Feifei ; Yao, Wen: MorphVAE: advancing morphological design of voxel-based soft robots with variational autoencoders. In: *Proceedings of the AAAI Conference on Artificial Intelligence* Bd. 38, 2024, S. 10368–10376 324, 334, 336

[Song et al. 2015] Song, Shuang ; Li, Zheng ; Yu, Haoyong ; Ren, Hongliang: Electromagnetic positioning for tip tracking and shape sensing of flexible robots. In: *IEEE Sensors Journal* 15 (2015), Nr. 8, S. 4565–4575 173

[Soter et al. 2018] Soter, Gabor ; Conn, Andrew ; Hauser, Helmut ; Rossiter, Jonathan: Bodily aware soft robots: integration of proprioceptive and exteroceptive sensors. In: *2018 IEEE international conference on robotics and automation (ICRA)* IEEE, 2018, S. 2448–2453 174

[Spielberg et al. 2021] Spielberg, Andrew ; Amini, Alexander ; Chin, Lillian ; Matusik, Wojciech ; Rus, Daniela: Co-learning of task and sensor placement for soft robotics. In: *IEEE Robotics and Automation Letters* 6 (2021), Nr. 2, S. 1208–1215 323, 330, 334, 335

[Spielberg et al. 2019] Spielberg, Andrew ; Zhao, Allan ; Hu, Yuanming ; Du, Tao ; Matusik, Wojciech ; Rus, Daniela: Learning-in-the-loop optimization: End-to-end control and co-design of soft robots through learned deep latent representations. In: *Advances in Neural Information Processing Systems* 32 (2019) 23, 76, 316, 323, 324, 328, 329, 330, 331, 332, 334

[Spong et al. 2020] Spong, Mark W. ; Hutchinson, Seth ; Vidyasagar, Mathukumalli: *Robot modeling and control*. John Wiley & Sons, 2020 50

[Standard 2016] Standard, ISO: ISO/TS 15066: 2016: Robots and robotic devices–collaborative robots. In: *International Organization for Standardization: Geneva, Switzerland* (2016) 2, 12, 57, 58, 59, 60, 61, 62, 64, 65, 66, 67, 68, 69, 70, 72, 75, 76, 308, 324

[Standard 2023] Standard, ISO: ISO/PAS 5672: 2023: Robotics — Collaborative applications — Test methods for measuring forces and pressures in human-robot contacts. In: *International Organization for Standardization: Geneva, Switzerland* (2023) 61

[Stella et al. 2023a] Stella, Francesco ; Della Santina, Cosimo ; Hughes, Josie: How can LLMs transform the robotic design process? In: *Nature machine intelligence* 5 (2023), Nr. 6, S. 561–564 342

[Stella et al. 2023b] Stella, Francesco ; Della Santina, Cosimo ; Hughes, Josie: Soft robot shape estimation with IMUs leveraging PCC kinematics for drift filtering. In: *IEEE Robotics and Automation Letters* (2023) 10

[Stella et al. 2023c] Stella, Francesco ; Guan, Qinghua ; Della Santina, Cosimo ; Hughes, Josie: Piecewise affine curvature model: a reduced-order model for soft robot-environment interaction beyond pcc. In: *2023 IEEE International Conference on Soft Robotics (RoboSoft)* IEEE, 2023, S. 1–7 31, 174, 176, 185, 202, 205, 364

[Stella and Hughes 2023] Stella, Francesco ; Hughes, Josie: The science of soft robot design: A review of motivations, methods and enabling technologies. In: *Frontiers in Robotics and AI* 9 (2023), S. 1059026 4

[Stella et al. 2023d] Stella, Francesco ; Hughes, Josie ; Rus, Daniela ; Della Santina, Cosimo: Prescribing cartesian stiffness of soft robots by co-optimization of shape and segment-level stiffness. In: *Soft Robotics* 10 (2023), Nr. 4, S. 701–712 39

[Stella et al. 2022] Stella, Francesco ; Obayashi, Nana ; Della Santina, Cosimo ; Hughes, Josie: An experimental validation of the polynomial curvature model: identification and optimal control of a soft underwater tentacle. In: *IEEE Robotics and Automation Letters* 7 (2022), Nr. 4, S. 11410–11417 31, 185, 203, 205

[Stölzle et al. 2024a] Stölzle, Maximilian ; Baberwal, Sonal ; Rus, Daniela ; Coyle, Shirley ; Della Santina, Cosimo: Guiding Soft Robots with Motor-Imagery Brain Signals and

Impedance Control. In: *2024 IEEE 7th International Conference on Soft Robotics (RoboSoft)* IEEE, 2024, S. 1–8. – Received the Best Paper Award 66, 117, 138, 243, 428, 434

[Stölzle et al. 2023] Stölzle, Maximilian ; Chin, Lillian ; Truby, Ryan L. ; Rus, Daniela ; Della Santina, Cosimo: Modelling Handed Shearing Auxetics: Selective Piecewise Constant Strain Kinematics and Dynamic Simulation. In: *2023 IEEE International Conference on Soft Robotics (RoboSoft)* IEEE, 2023, S. 1–8 98, 113, 118, 139, 434

[Stölzle and Della Santina 2021] Stölzle, Maximilian ; Della Santina, Cosimo: Piston-driven pneumatically-actuated soft robots: Modeling and backstepping control. In: *IEEE Control Systems Letters* 6 (2021), S. 1837–1842 153, 222, 252, 361, 433

[Stölzle and Della Santina 2024] Stölzle, Maximilian ; Della Santina, Cosimo: Input-to-State Stable Coupled Oscillator Networks for Closed-form Model-based Control in Latent Space. In: Globerson, A. (Hrsg.) ; Mackey, L. (Hrsg.) ; Belgrave, D. (Hrsg.) ; Fan, A. (Hrsg.) ; Paquet, U. (Hrsg.) ; Tomczak, J. (Hrsg.) ; Zhang, C. (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 37, Curran Associates, Inc., 2024, 82010–82059. – Spotlight (top 10% of accepted papers) 12, 195, 208, 209, 210, 216, 220, 324, 325, 332, 333, 334, 343, 434

[Stölzle et al. 2022] Stölzle, Maximilian ; Miki, Takahiro ; Gerdes, Levin ; Azkarate, Martin ; Hutter, Marco: Reconstructing occluded elevation information in terrain maps with self-supervised learning. In: *IEEE Robotics and Automation Letters* 7 (2022), Nr. 2, S. 1697–1704 347, 433

[Stölzle et al. 2024b] Stölzle, Maximilian ; Rus, Daniela ; Della Santina, Cosimo: An Experimental Study of Model-based Control for Planar Handed Shearing Auxetics Robots. In: *Experimental Robotics*. Cham : Springer Nature Switzerland, 2024. – ISBN 978–3–031–63596–0, S. 153–167 38, 71, 98, 117, 133, 139, 145, 195, 197, 203, 210, 243, 248, 333, 364, 434

[Strogatz 2018] Strogatz, Steven H.: *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. CRC press, 2018 262, 267, 302, 303

[Stölzle et al. 2025a] Stölzle, Maximilian ; Pagliarani, Niccolò ; Hughes, Josie ; Stella, Francesco ; Laschi, Cecilia ; Rus, Daniela ; Cianchetti, Matteo ; Della Santina, Cosimo ; Zardini, Gioele: Soft yet Effective Robots via Holistic Co-Design. In: *Nature Machine Intelligence* (2025). – Under review 322, 433

[Stölzle et al. 2025b] Stölzle, Maximilian ; Rusch, T. K. ; Patterson, Zach J. ; Pérez-Dattari, Rodrigo ; Stella, Francesco ; Hughes, Josie ; Della Santina, Cosimo ; Rus, Daniela: Learning to Move in Rhythm: Task-Conditioned Motion Policies with Orbital Stability Guarantees. In: *Science Robotics Special Issue on Transfer Learning, Foundation Models, and emerging capabilities* (2025). – Under review 260, 433

[Suh 1998] Suh, Nam P.: Axiomatic design theory for systems. In: *Research in engineering design* 10 (1998), S. 189–209 327

[Sun et al. 2024] Sun, Sunan ; Gao, Haihui ; Li, Tianyu ; Figueroa, Nadia: Directionality-aware mixture model parallel sampling for efficient linear parameter varying dynamical system learning. In: *IEEE Robotics and Automation Letters* (2024) 302

[Sun et al. 2022] Sun, Wentao ; Akashi, Nozomi ; Kuniyoshi, Yasuo ; Nakajima, Kohei: Physics-informed recurrent neural networks for soft pneumatic actuators. In: *IEEE Robotics and Automation Letters* 7 (2022), Nr. 3, S. 6862–6869 6

[Sutton et al. 1998] Sutton, Richard S. ; Barto, Andrew G. et al.: *Reinforcement learning: An introduction*. MIT press Cambridge, 1998 324, 334, 336, 339

[Tacca et al. 2022] Tacca, Nicholas ; Nassour, John ; Ehrlich, Stefan K. ; Berberich, Nicolas ; Cheng, Gordon: Neuro-cognitive assessment of intentional control methods for a soft elbow exosuit using error-related potentials. In: *Journal of NeuroEngineering and Rehabilitation* 19 (2022), Nr. 1, S. 124 139

[Taheri et al. 2014] Taheri, Behzad ; Case, David ; Richer, Edmond: Force and stiffness backstepping-sliding mode controller for pneumatic cylinders. In: *IEEE/ASME Transactions on Mechatronics* 19 (2014), Nr. 6, S. 1799–1809 361

[Taylor et al. 2020] Taylor, Andrew ; Singletary, Andrew ; Yue, Yisong ; Ames, Aaron: Learning for safety-critical control with control barrier functions. In: *Learning for dynamics and control* PMLR, 2020, S. 708–717 6

[Team 2025] Team, Gemini R.: Gemini Robotics: Bringing AI into the Physical World / Google DeepMind. 2025. – Forschungsbericht. – Available at `https://storage.googleapis.com/deepmind-media/gemini-robotics/gemini_robotics_report.pdf` 261, 294

[Terryn et al. 2017] Terryn, Seppe ; Brancart, Joost ; Lefeber, Dirk ; Van Assche, Guy ; Vanderborght, Bram: Self-healing soft pneumatic robots. In: *Science Robotics* 2 (2017), Nr. 9, S. eaan4268 3

[Thuruthel et al. 2018a] Thuruthel, Thomas G. ; Ansari, Yasmin ; Falotico, Egidio ; Laschi, Cecilia: Control Strategies for Soft Robotic Manipulators: A Survey. In: *Soft robotics* (2018) 4

[Thuruthel et al. 2017] Thuruthel, Thomas G. ; Falotico, Egidio ; Renda, Federico ; Laschi, Cecilia: Learning dynamic models for open loop predictive control of soft robotic manipulators. In: *Bioinspiration & biomimetics* 12 (2017), Nr. 6, S. 066003 10, 195, 333, 340

[Thuruthel et al. 2018b] Thuruthel, Thomas G. ; Falotico, Egidio ; Renda, Federico ; Laschi, Cecilia: Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators. In: *IEEE Transactions on Robotics* 35 (2018), Nr. 1, S. 124–134 3, 5, 6, 11, 333

[Thuruthel and Iida 2023] Thuruthel, Thomas G. ; Iida, Fumiya: Multi-modal Sensor Fusion for Learning Rich Models for Interacting Soft Robots. In: *2023 IEEE International Conference on Soft Robotics (RoboSoft)* IEEE, 2023, S. 1–6 242

[Thuruthel et al. 2019] Thuruthel, Thomas G. ; Shih, Benjamin ; Laschi, Cecilia ; Tolley, Michael T.: Soft robot perception using embedded soft sensors and recurrent neural networks. In: *Science Robotics* 4 (2019), Nr. 26, S. eaav1488 3, 9, 82, 174

[Tiburzio et al. 2025] Tiburzio, Sebastien ; Coleman, Tomás ; Feliu-Talegon, Daniel ; Della Santina, Cosimo: Controlling Deformable Objects with Non-negligible Dynamics: a Shape-Regulation Approach to End-Point Positioning. In: *IEEE Transactions on Robotics* (2025). – Under Review 31

[Timoshenko 1921] Timoshenko, Stephen P.: LXVI. On the correction for shear of the differential equation for transverse vibrations of prismatic bars. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 41 (1921), Nr. 245, S. 744–746 62

[Todd 1996] Todd, David J.: *Fundamentals of robot technology: An introduction to industrial robots, teleoperators and robot vehicles*. Dordrecht : Springer, 1996 2, 323

[Todorov et al. 2012] Todorov, Emanuel ; Erez, Tom ; Tassa, Yuval: Mujoco: A physics engine for model-based control. In: *2012 IEEE/RSJ international conference on intelligent robots and systems* IEEE, 2012, S. 5026–5033 363

[Tolley et al. 2014] Tolley, Michael T. ; Shepherd, Robert F. ; Mosadegh, Bobak ; Galloway, Kevin C. ; Wehner, Michael ; Karpelson, Michael ; Wood, Robert J. ; Whitesides, George M.: A resilient, untethered soft robot. In: *Soft robotics* 1 (2014), Nr. 3, S. 213–223 3

[Toshimitsu et al. 2021] Toshimitsu, Yasunori ; Wong, Ki W. ; Buchner, Thomas ; Katzschmann, Robert: Sopra: Fabrication & dynamical modeling of a scalable soft continuum robotic arm with integrated proprioceptive sensing. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* IEEE, 2021, S. 653–660 195

[Touvron et al. 2023] Touvron, Hugo ; Lavril, Thibaut ; Izacard, Gautier ; Martinet, Xavier ; Lachaux, Marie-Anne ; Lacroix, Timothée ; Rozière, Baptiste ; Goyal, Naman ; Hambro, Eric ; Azhar, Faisal et al.: Llama: Open and efficient foundation language models. In: *arXiv preprint arXiv:2302.13971* (2023) 64

[Truby et al. 2021] Truby, Ryan L. ; Chin, Lillian ; Rus, Daniela: A recipe for electrically-driven soft robots via 3d printed handed shearing auxetics. In: *IEEE Robotics and Automation Letters* 6 (2021), Nr. 2, S. 795–802 13, 14, 99, 110, 113, 121, 140, 314, 315, 321

[Truby et al. 2020] Truby, Ryan L. ; Della Santina, Cosimo ; Rus, Daniela: Distributed proprioception of 3D configuration in soft, sensorized robots via deep learning. In: *IEEE Robotics and Automation Letters* 5 (2020), Nr. 2, S. 3299–3306 3, 82, 174

[Tsitouras 2011] Tsitouras, Ch: Runge–Kutta pairs of order 5 (4) satisfying only the first column simplifying assumption. In: *Computers & Mathematics with Applications* 62 (2011), Nr. 2, S. 770–775 364

[Tsounis et al. 2025] Tsounis, Vassilios ; Grandia, Ruben ; Bächer, Moritz: On Solving the Dynamics of Constrained Rigid Multi-Body Systems with Kinematic Loops. In: *arXiv preprint arXiv:2504.19771* (2025) 314

[Tulleken 1990] Tulleken, Herbert J.: Generalized binary noise test-signal concept for improved identification-experiment design. In: *Automatica* 26 (1990), Nr. 1, S. 37–49 115

[Ulrich and Eppinger 2016] Ulrich, Karl T. ; Eppinger, Steven D.: *Product design and development*. McGraw-hill, 2016 327

[Urain et al. 2020] Urain, Julen ; Ginesi, Michele ; Tateo, Davide ; Peters, Jan: Imitationflow: Learning deep stable stochastic dynamic systems by normalizing flows. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* IEEE, 2020, S. 5231–5237 261, 262, 264, 265, 273, 274, 279, 280, 282, 283, 286, 287, 300, 301

[Vahdat et al. 2022] Vahdat, Arash ; Williams, Francis ; Gojcic, Zan ; Litany, Or ; Fidler, Sanja ; Kreis, Karsten et al.: Lion: Latent point diffusion models for 3d shape generation. In: *Advances in Neural Information Processing Systems* 35 (2022), S. 10021–10039 324

[Vaissier Welborn and Head-Gordon 2018] Vaissier Welborn, Valerie ; Head-Gordon, Teresa: Computational design of synthetic enzymes. In: *Chemical reviews* 119 (2018), Nr. 11, S. 6613–6630 328

[Valadas et al. 2025] Valadas, Ricardo ; Stölzle, Maximilian ; Liu, Jingyue ; Della Santina, Cosimo: Learning Low-Dimensional Strain Models of Soft Robots by Looking at the Evolution of Their Shape with Application to Model-Based Control. In: *2025 IEEE 8th International Conference on Soft Robotics (RoboSoft)* IEEE, 2025, S. 1–8 194, 324, 332, 333, 334, 340, 429, 434

[Van Diepen and Shea 2018] Van Diepen, Merel ; Shea, Kristina: A spatial grammar method for the computational design synthesis of virtual soft robots. In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* Bd. 51753 American Society of Mechanical Engineers, 2018, S. V02AT03A012 58, 323

[Van Erp et al. 2012] Van Erp, Jan ; Lotte, Fabien ; Tangermann, Michael: Brain-computer interfaces: beyond medical applications. In: *Computer* 45 (2012), Nr. 4, S. 26–34 139

[Van Hoof et al. 2016] Van Hoof, Herke ; Chen, Nutan ; Karl, Maximilian ; Smagt, Patrick van d. ; Peters, Jan: Stable reinforcement learning with autoencoders for tactile and visual data. In: *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)* IEEE, 2016, S. 3928–3934 221

[Vanneste et al. 2021] Vanneste, Félix ; Goury, Olivier ; Duriez, Christian: Enabling the control of a new degree of freedom by using anisotropic material on a 6-DOF parallel soft robot. In: *2021 IEEE 4th International Conference on Soft Robotics (RoboSoft)* IEEE, 2021, S. 636–642 99

[Vasiljevic et al. 2016] Vasiljevic, Gabriel Alves M. ; De Miranda, Leonardo C. ; De Miranda, Erica Esteves C. et al.: A case study of MasterMind Chess: Comparing mouse/keyboard interaction with Kinect-based gestural interface. In: *Advances in Human-Computer Interaction* 2016 (2016) 147

[Vasios et al. 2020] Vasios, Nikolaos ; Gross, Andrew J. ; Soifer, Scott ; Overvelde, Johannes T. ; Bertoldi, Katia: Harnessing viscous flow to simplify the actuation of fluidic soft robots. In: *Soft robotics* 7 (2020), Nr. 1, S. 1–9 3

[Versace 1971] Versace, John: A review of the severity index. (1971) 60

[Viano and Lau 1985] Viano, David C. ; Lau, Ian V.: Thoracic impact: a viscous tolerance criterion / SAE Technical Paper. 1985. – Forschungsbericht 61

[Vihmar et al. 2023] Vihmar, Marie ; Valdur, Kadri-Ann ; Banerji, Saoni ; Must, Indrek: How to measure embodied intelligence? In: *IOP Conference Series: Materials Science and Engineering* Bd. 1292 IOP Publishing, 2023, S. 012002 332, 334

[Villani et al. 2018] Villani, Valeria ; Pini, Fabio ; Leali, Francesco ; Secchi, Cristian: Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. In: *Mechatronics* 55 (2018), S. 248–266 2, 323

[Virtanen et al. 2020] Virtanen, Pauli ; Gommers, Ralf ; Oliphant, Travis E. ; Haberland, Matt ; Reddy, Tyler ; Cournapeau, David ; Burovski, Evgeni ; Peterson, Pearu ; Weckesser, Warren ; Bright, Jonathan ; van der Walt, Stéfan J. ; Brett, Matthew ; Wilson, Joshua ; Millman, K. J. ; Mayorov, Nikolay ; Nelson, Andrew R. J. ; Jones, Eric ; Kern, Robert ; Larson, Eric ; Carey, C J. ; Polat, İlhan ; Feng, Yu ; Moore, Eric W. ; VanderPlas, Jake ; Laxalde, Denis ; Perktold, Josef ; Cimrman, Robert ; Henriksen, Ian ; Quintero, E. A. ; Harris, Charles R. ; Archibald, Anne M. ; Ribeiro, Antônio H. ; Pedregosa, Fabian ; van Mulbregt, Paul ; SciPy 1.0 Contributors: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. In: *Nature Methods* 17 (2020), S. 261–272. http://dx.doi.org/10.1038/s41592-019-0686-2. – DOI 10.1038/s41592–019–0686–2 365, 367

[Vo-Minh et al. 2010] Vo-Minh, Tri ; Tjahjowidodo, Tegoeh ; Ramon, Herman ; Van Brussel, Hendrik: A new approach to modeling hysteresis in a pneumatic artificial muscle using the Maxwell-slip model. In: *IEEE/ASME transactions on mechatronics* 16 (2010), Nr. 1, S. 177–186 168

[Wahlström et al. 2015] Wahlström, Niklas ; Schön, Thomas B. ; Deisenroth, Marc P.: Learning deep dynamical models from image pixels. In: *IFAC-PapersOnLine* 48 (2015), Nr. 28, S. 1059–1064 221

[Wall et al. 2017] Wall, Vincent ; Zöller, Gabriel ; Brock, Oliver: A method for sensorizing soft actuators and its application to the RBO hand 2. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)* IEEE, 2017, S. 4965–4970 82

[Wang et al. 2013] Wang, Hesheng ; Chen, Weidong ; Yu, Xiaojin ; Deng, Tao ; Wang, Xiaozhou ; Pfeifer, Rolf: Visual servo control of cable-driven soft robotic manipulator.

In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* IEEE, 2013, S. 57–62 82

[Wang et al. 2018a] Wang, Hongbo ; Totaro, Massimo ; Beccai, Lucia: Toward perceptive soft robots: Progress and challenges. In: *Advanced Science* 5 (2018), Nr. 9, S. 1800541 82

[Wang et al. 2019] Wang, Tao ; Zhang, Yunce ; Chen, Zheng ; Zhu, Shiqiang: Parameter identification and model-based nonlinear robust control of fluidic soft bending actuators. In: *IEEE/ASME Transactions on Mechatronics* 24 (2019), Nr. 3, S. 1346–1355 155

[Wang et al. 2018b] Wang, Tianyu ; Ge, Lisen ; Gu, Guoying: Programmable design of soft pneu-net actuators with oblique chambers can generate coupled bending and twisting motions. In: *Sensors and Actuators A: Physical* 271 (2018), S. 131–138 328

[Wang et al. 2023a] Wang, Tsun-Hsuan ; Ma, Pingchuan ; Spielberg, Andrew E. ; Xian, Zhou ; Zhang, Hao ; Tenenbaum, Joshua B. ; Rus, Daniela ; Gan, Chuang: Softzoo: A soft robot co-design benchmark for locomotion in diverse environments. In: *The Eleventh International Conference on Learning Representations*, 2023 323, 324, 328, 329, 331, 332, 333, 334, 339

[Wang et al. 2024] Wang, Tsun-Hsuan J. ; Zheng, Juntian ; Ma, Pingchuan ; Du, Yilun ; Kim, Byungchul ; Spielberg, Andrew ; Tenenbaum, Josh ; Gan, Chuang ; Rus, Daniela: Diffusebot: Breeding soft robots with physics-augmented generative diffusion models. In: *Advances in Neural Information Processing Systems* 36 (2024) 23, 76, 195, 316, 323, 324, 328, 329, 330, 331, 332, 333, 334, 335, 336

[Wang et al. 2022] Wang, Yuxing ; Wu, Shuang ; Fu, Haobo ; Fu, Qiang ; Zhang, Tiantian ; Chang, Yongzhe ; Wang, Xueqian: Curriculum-based co-design of morphology and control of voxel-based soft robots. In: *The Eleventh International Conference on Learning Representations*, 2022 316, 323, 324, 328, 329, 332, 333, 336

[Wang et al. 2023b] Wang, Yuxing ; Wu, Shuang ; Zhang, Tiantian ; Chang, Yongzhe ; Fu, Haobo ; Fu, Qiang ; Wang, Xueqian: PreCo: Enhancing Generalization in Co-Design of Modular Soft Robots via Brain-Body Pre-Training. In: *Conference on Robot Learning* PMLR, 2023, S. 478–498 316, 323, 324, 328, 329, 331, 332, 333

[Wang et al. 2004] Wang, Zhou ; Bovik, Alan C. ; Sheikh, Hamid R. ; Simoncelli, Eero P.: Image quality assessment: from error visibility to structural similarity. In: *IEEE transactions on image processing* 13 (2004), Nr. 4, S. 600–612 347

[Ward-Cherrier et al. 2018] Ward-Cherrier, Benjamin ; Pestell, Nicholas ; Cramphorn, Luke ; Winstone, Benjamin ; Giannaccini, Maria E. ; Rossiter, Jonathan ; Lepora, Nathan F.: The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies. In: *Soft robotics* 5 (2018), Nr. 2, S. 216–227 82

[Wassink and Stramigioli 2007] Wassink, Martin ; Stramigioli, Stefano: Towards a novel safety norm for domestic robotics. In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems* IEEE, 2007, S. 3354–3359 60, 74

[Watter et al. 2015] Watter, Manuel ; Springenberg, Jost ; Boedecker, Joschka ; Riedmiller, Martin: Embed to control: A locally linear latent dynamics model for control from raw images. In: *Advances in neural information processing systems* 28 (2015) 221

[Weber et al. 2012] Weber, Bernhard ; Zeller, Paul ; Kühnlenz, Kolja: Multi-camera based real-time configuration estimation of continuum robots. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* IEEE, 2012, S. 3350–3355 82

[Webster III and Jones 2010] Webster III, Robert J. ; Jones, Bryan A.: Design and kinematic modeling of constant curvature continuum robots: A review. In: *The International Journal of Robotics Research* 29 (2010), Nr. 13, S. 1661–1683 13, 25, 28, 29, 65, 83, 99, 174, 195, 196, 244, 308, 340, 363

[Wehner et al. 2016] Wehner, Michael ; Truby, Ryan L. ; Fitzgerald, Daniel J. ; Mosadegh, Bobak ; Whitesides, George M. ; Lewis, Jennifer A. ; Wood, Robert J.: An integrated design and fabrication strategy for entirely soft, autonomous robots. In: *nature* 536 (2016), Nr. 7617, S. 451–455 3

[Wen 1976] Wen, Yi-Kwei: Method for random vibration of hysteretic systems. In: *Journal of the engineering mechanics division* 102 (1976), Nr. 2, S. 249–263 315

[Wensing and Slotine 2017] Wensing, Patrick M. ; Slotine, Jean-Jacques: Sparse control for dynamic movement primitives. In: *IFAC-PapersOnLine* 50 (2017), Nr. 1, S. 10114–10121 261, 264

[Werner et al. 2020] Werner, Peter ; Hofer, Matthias ; Sferrazza, Carmelo ; D'Andrea, Raffaello: Vision-based proprioceptive sensing: Tip position estimation for a soft inflatable bellow actuator. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* IEEE, 2020, S. 8889–8896 82

[White et al. 2023] White, Alistair ; Kilbertus, Niki ; Gelbrecht, Maximilian ; Boers, Niklas: Stabilized Neural Differential Equations for Learning Dynamics with Explicit Constraints. In: *Advances in Neural Information Processing Systems* 36 (2023), S. 12929–12950 221

[Wong et al. 2025] Wong, Kiwan ; Stölzle, Maximilian ; Xiao, Wei ; Della Santina, Cosimo ; Rus, Daniela ; Zardini, Gioele: Contact-Aware Safety in Soft Robots Using High-Order Control Barrier and Lyapunov Functions. In: *IEEE Robotics and Automation Letters* (2025). – Accepted 313, 429, 433

[Wu et al. 2022] Wu, Xuwei ; Ott, Christian ; Albu-Schäffer, Alin ; Dietrich, Alexander: Passive decoupled multitask controller for redundant robots. In: *IEEE Transactions on Control Systems Technology* 31 (2022), Nr. 1, S. 1–16 224, 226, 230

[Xavier et al. 2020] Xavier, Matheus S. ; Fleming, Andrew J. ; Yong, Yuen K.: Modelling and simulation of pneumatic sources for soft robotic applications. In: *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)* IEEE, 2020, S. 916–921 14

[Xie et al. 2023] Xie, Qing ; Zhang, Yunce ; Wang, Tao ; Zhu, Shiqiang: Dynamic response prediction of hydraulic soft robotic arms based on LSTM neural network. In: *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* 237 (2023), Nr. 7, S. 1251–1265 6

[Yamada et al. 2023] Yamada, Jun ; Hung, Chia-Man ; Collins, Jack ; Havoutis, Ioannis ; Posner, Ingmar: Leveraging scene embeddings for gradient-based motion planning in latent space. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)* IEEE, 2023, S. 5674–5680 221

[Yamada et al. 1997] Yamada, Yoji ; Suita, Kazutsugu ; Ikeda, H ; Sugimoto, N ; Miura, H ; Nakamura, H: Evaluation of pain tolerance based on a biomechanical method for human-robot coexistence. In: *Transactions of the Japan Society of Mechanical Engineers, Series C* 63 (1997), Nr. 612, S. 2814–2819 76

[Yasa et al. 2023] Yasa, Oncay ; Toshimitsu, Yasunori ; Michelis, Mike Y. ; Jones, Lewis S. ; Filippi, Miriam ; Buchner, Thomas ; Katzschmann, Robert K.: An overview of soft robotics. In: *Annual Review of Control, Robotics, and Autonomous Systems* 6 (2023), Nr. 1, S. 1–29 3, 5, 25, 57

[Yu et al. 2018] Yu, Jiahui ; Lin, Zhe ; Yang, Jimei ; Shen, Xiaohui ; Lu, Xin ; Huang, Thomas S.: Generative image inpainting with contextual attention. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, S. 5505–5514 347

[Yumbla et al. 2021] Yumbla, Emiliano Q. ; Qiao, Zhi ; Tao, Weijia ; Zhang, Wenlong: Human assistance and augmentation with wearable soft robotics: a literature review and perspectives. In: *Current Robotics Reports* (2021), S. 1–15 2

[Zaidi et al. 2021] Zaidi, Shadab ; Maselli, Martina ; Laschi, Cecilia ; Cianchetti, Matteo: Actuation technologies for soft robot grippers and manipulators: A review. In: *Current Robotics Reports* 2 (2021), Nr. 3, S. 355–369 14, 168, 309

[Zardini et al. 2021a] Zardini, Enrico ; Zappetti, Davide ; Zambrano, Davide ; Iacca, Giovanni ; Floreano, Dario: Seeking quality diversity in evolutionary co-design of morphology and control of soft tensegrity modular robots. In: *Proceedings of the Genetic and Evolutionary Computation Conference*, 2021, S. 189–197 332

[Zardini 2023] Zardini, Gioele: *Co-design of complex systems: From autonomy to future mobility systems*, ETH Zurich, Diss., 2023 76, 323, 328, 331

[Zardini et al. 2021b] Zardini, Gioele ; Censi, Andrea ; Frazzoli, Emilio: Co-design of autonomous systems: From hardware selection to control synthesis. In: *2021 European Control Conference (ECC)* IEEE, 2021, S. 682–689 324, 328

[Zardini et al. 2022a] Zardini, Gioele ; Lanzetti, Nicolas ; Censi, Andrea ; Frazzoli, Emilio ; Pavone, Marco: Co-design to enable user-friendly tools to assess the impact of future mobility solutions. In: *IEEE Transactions on Network Science and Engineering* 10 (2022), Nr. 2, S. 827–844 328

[Zardini et al. 2020] Zardini, Gioele ; Lanzetti, Nicolas ; Salazar, Mauro ; Censi, Andrea ; Frazzoli, Emilio ; Pavone, Marco: On the co-design of AV-enabled mobility systems. In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)* IEEE, 2020, S. 1–8 328

[Zardini et al. 2022b] Zardini, Gioele ; Suter, Zelio ; Censi, Andrea ; Frazzoli, Emilio: Task-driven Modular Co-design of Vehicle Control Systems. In: *2022 61th IEEE Conference on Decision and Control (CDC)*. Cancun, Mexico, Dezember 2022, S. 2196–2203 324

[Zare et al. 2024] Zare, Maryam ; Kebria, Parham M. ; Khosravi, Abbas ; Nahavandi, Saeid: A survey of imitation learning: Algorithms, recent developments, and challenges. In: *IEEE Transactions on Cybernetics* (2024) 261

[Zhang et al. 2022a] Zhang, Annan ; Truby, Ryan L. ; Chin, Lillian ; Li, Shuguang ; Rus, Daniela: Vision-Based Sensing for Electrically-Driven Soft Actuators. In: *IEEE Robotics and Automation Letters* 7 (2022), Nr. 4, S. 11509–11516 14, 99, 118

[Zhang et al. 2018] Zhang, Hongying ; Kumar, A S. ; Chen, Feifei ; Fuh, Jerry Y. ; Wang, Michael Y.: Topology optimized multimaterial soft fingers for applications on grippers, rehabilitation, and artificial hands. In: *IEEE/ASME Transactions on Mechatronics* 24 (2018), Nr. 1, S. 120–131 328

[Zhang et al. 2022b] Zhang, Jiechao ; Mohammadi, Hadi B. ; Rozo, Leonel: Learning Riemannian stable dynamical systems via diffeomorphisms. In: *6th Annual Conference on Robot Learning*, 2022 261, 264, 266

[Zhang et al. 2019] Zhang, Jinhua ; Wang, Baozeng ; Zhang, Cheng ; Xiao, Yanqing ; Wang, Michael Y.: An EEG/EMG/EOG-based multimodal human-machine interface to real-time control of a soft robot hand. In: *Frontiers in neurorobotics* 13 (2019), S. 7 139

[Zhao et al. 2020] Zhao, Allan ; Xu, Jie ; Konaković-Luković, Mina ; Hughes, Josephine ; Spielberg, Andrew ; Rus, Daniela ; Matusik, Wojciech: Robogrammar: graph grammar for terrain-optimized robot design. In: *ACM Transactions on Graphics (TOG)* 39 (2020), Nr. 6, S. 1–16 328, 332, 340

[Zhao et al. 2024a] Zhao, Wei ; Jiang, Xiaolu ; Zhang, Baocan ; Xiao, Shixiao ; Weng, Sujun: CTNet: a convolutional transformer network for EEG-based motor imagery classification. In: *Scientific Reports* 14 (2024), Nr. 1, S. 20237 151

[Zhao et al. 2024b] Zhao, Xiangmo ; Fang, Yukun ; Min, Haigen ; Wu, Xia ; Wang, Wuqi ; Teixeira, Rui: Potential sources of sensor data anomalies for autonomous vehicles: An overview from road vehicle safety perspective. In: *Expert Systems with Applications* 236 (2024), S. 121358 2, 323

[Zheng et al. 2019] Zheng, Gang ; Goury, Olivier ; Thieffry, Maxime ; Kruszewski, Alexandre ; Duriez, Christian: Controllability pre-verification of silicone soft robots based on finite-element method. In: *2019 International Conference on Robotics and Automation (ICRA)* IEEE, 2019, S. 7395–7400 324, 334, 340

[Zheng et al. 2024] Zheng, Hehui ; Pinzello, Sebastian ; Cangan, Barnabas G. ; Buchner, Thomas J. ; Katzschmann, Robert K.: Vision-Based Online Key Point Estimation of Deformable Robots. In: *Advanced Intelligent Systems* 6 (2024), Nr. 10, S. 2400105 198

[Zhi et al. 2024] Zhi, Weiming ; Tang, Haozhan ; Zhang, Tianyi ; Johnson-Roberson, Matthew: Teaching Periodic Stable Robot Motion Generation Via Sketch. In: *IEEE Robotics and Automation Letters* (2024) 258, 259, 261, 262, 264, 265, 266, 267, 269, 273, 274, 279, 282, 285, 286, 287, 288, 289, 295, 300, 303

[Zhong and Leonard 2020] Zhong, Yaofeng D. ; Leonard, Naomi: Unsupervised learning of lagrangian dynamics from images for prediction and control. In: *Advances in Neural Information Processing Systems* 33 (2020), S. 10741–10752 221

[Zinn et al. 2004a] Zinn, Michael ; Khatib, Oussama ; Roth, Bernard ; Salisbury, J K.: Playing it safe [human-friendly robots]. In: *IEEE Robotics & Automation Magazine* 11 (2004), Nr. 2, S. 12–21 58, 60

[Zinn et al. 2004b] Zinn, Michael ; Roth, Bernard ; Khatib, Oussama ; Salisbury, J K.: A new actuation approach for human friendly robot design. In: *The international journal of robotics research* 23 (2004), Nr. 4-5, S. 379–398 58, 60

[Zöller et al. 2018] Zöller, Gabriel ; Wall, Vincent ; Brock, Oliver: Acoustic sensing for soft pneumatic actuators. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* IEEE, 2018, S. 6986–6991 82

Abbreviations

AC Affine Curvature.

ADL Activities of Daily Living.

AI Artificial Intelligence.

AOS Asymptotic Orbital Stability.

AOT Ahead-Of-Time.

BA Bundle Adjustment.

BCI Brain-Computer Interface.

BMI Brain-Machine Interface.

CAD Computer-aided Design.

CBF Control Barrier Function.

CC Constant Curvature.

CFA-CON Closed-Form Approximation of CON.

CFA-UDCON Closed-Form Approximation of Underdamped CON.

CLF Control Lyapunov Function.

CNN Convolutional Neural Network.

cobot collaborative robot.

COM Center of Mass.

CON Coupled Oscillator Network.

coRNN Coupled Oscillatory Recurrent Neural Network.

CS Constant Strain.

CV Computer Vision.

DCM Discretized Cosserat Rod Model.

DMP Dynamic Motion Primitive.

DNN Deep Neural Network.

DOF Degrees of Freedom.

DP Diffusion Policy.

DTW Dynamic Time Warping.

EEG Electroencephalography.

EOM Equation of Motion.

EOS Exponential Orbital Stability.

ERD Event-Related Desynchronization.

ERS Event-Related Synchronization.

FEM Finite Element Method.

GAN Generative Adversarial Network.

GAS Global Asymptotic Stability.

GBN Generalized Binary Noise.

GP Gaussian Process.

GPU Graphics Processing Unit.

GRU Gated Recurrent Unit.

GVS Geometric Variable Strain.

HIC Head Injury Criterion.

HMI Human Machine Interface.

HNN Hamiltonian Neural Network.

HRI Human Robot Interface.

HSA Handed Shearing Auxetic.

ICP Iterative Closest Point.

ILC Iterative Learning Control.

IMU Inertial Measurement Unit.

IR Infrared.

ISO International Organization for Standardization.

ISS Input-to-State Stability.

JIT Just-In-Time.

LAN Local Area Network.

LAS Local Asymptotic Stability.

LDA Linear Discriminant Analysis.

LFD Learning from Demonstration.

LNN Lagrangian Neural Network.

LQR Linear-Quadratic Regulator.

LSTM Long Short-Term Memory.

MAE Mean Absolute Error.

MBRL Model-Based Reinforcement Learning.

MCS Motion Capture System.

MED Mean Euclidean Distance.

ML Machine Learning.

MLP Multilayer Perceptron.

MPC Model Predictive Control.

MRS Magnetoresistive Sensor.

MS Multistability.

MSE Mean Squared Error.

NN Neural Network.

NODE Neural ODE.

ODE Ordinary Differential Equation.

OS Orbital Stability.

OSMP Orbitally Stable Motion Policy.

PAC Piecewise Affine Curvature.

PCB Printed Circuit Board.

PCC Piecewise Constant Curvature.

PCS Piecewise Constant Strain.

PDE Partial Differential Equation.

POD Proper Orthogonal Decomposition.

PSNR Peak Signal-to-Noise Ratio.

RFFN Random Fourier Features Networks.

RL Reinforcement Learning.

RMSE Root Mean Squared Error.

RNN Recurrent Neural Network.

RQ Research Question.

SGD Stochastic Gradient Descent.

SINDy Sparse Identification of Nonlinear Dynamics.

SLAM Simultaneous Localization and Mapping.

SMA Shape-Memory Alloy.

SMP Stable Motion Primitive.

SOTA State of the Art.

SPCS Selective Piecewise Constant Strain.

SPDT Stable Periodic Diagrammatic Teaching.

SRDHC Soft Robot Design Hazardousness Criterion.

SRISC Soft Robot Injury Severity Criterion.

SSIM Structural Similarity Index Measure.

SWA Stochastic Weight Averaging.

TCP Transmission Control Protocol.

TOF Time of Flight.

TPU Tensor Processing Unit.

TRL Technology Readiness Level.

TT Trajectory Tracking.

VAE Variational Autoencoder.

VLA Vision-Language-Action.

VLM Vision-Language Model.

vSLAM visual SLAM.

Acknowledgments

First and foremost, I want to thank my advisor and copromotor, Dr. Cosimo Della Santina, for believing in me and giving me the chance to pursue a Ph.D. Initially, you entrusted me with a well-scaffolded project that let me dive into soft robotics and advanced nonlinear control step by step, smoothing the learning curve. Over time, you progressively empowered me to design and manage research projects on my own. Your technical guidance was always spot-on whenever I hit a roadblock, and your candid yet constructive feedback on my research and draft papers—serving as the ”feared second reviewer“ long before paper submission—pushed the work to a higher standard. Your coaching on presentation structure and slide style elevated my presentation skills, and your early confidence in my ability to mentor students broadened my professional growth. Thank you for connecting me with leading collaborators across Europe and beyond, and for gradually entrusting me with *senior* responsibilities like leading the ICS practical assignment and EU EMERGE deliverables. Above all, thank you for your steadfast and effective support throughout this entire Ph.D. journey.

I am deeply grateful to my promotor, Professor Robert Babuška, for always being available and for responding so quickly to every request. His critical assessments of my progress, streamlined coordination, fast feedback on thesis drafts, and our engaging discussions about the propositions were invaluable.

Professors Daniela Rus and Gioele Zardini, the many impactful collaborations with you fundamentally shaped this Ph.D. research, and my visiting period in your groups significantly contributed to my personal development—thank you.

I appreciate the independent committee members—Junior Professor Gianluca Rizzello, Dr. Moritz Bächer, Professors Tom Ooomen, Guido de Croon, and Gioele Zardini—for graciously accepting the invitation, accommodating our scheduling constraints, and dedicating significant time to reviewing the thesis and attending the defense.

I am indebted to all scientific collaborators whose contributions enliven the chapters of this thesis. In particular, Ryan Truby and Lillian Chin, thank you for inviting me to work on the HSA robot platform—a project that was both enjoyable and pivotal to this thesis. Thank you, Sonal Baberwal, for a highly interdisciplinary and almost daring collaboration on brain-controlled soft robots—bold, harmonious, and ultimately very successful. Working on this project with you was a joy, and I’m grateful for the friendship we’ve built along the way.

Thank you to all my students—whether you joined for a research internship, a Bachelor End Project (BEP), or a master’s thesis—for trusting my project ideas and guidance. Your motivation, dedication, and creativity made our work both productive and inspiring, and many of your projects became publications that form the backbone of this dissertation. I had the privilege of co-mentoring several M.S. students—Emanuele Rosi, Ricardo Valadas, Gabriele Di Marzo, Riccardo Sepe, Gioele Buriani, and Iván López Brocéno—as well as the 2023 BEP team of Bendert de Roij van Zuidewijn, Hannah Gielen, Quirijn Bos, and

Floris Cuperus. A special shout-out goes to the 2021 BEP team—Thomas Baaij, Marn Klein Holkenborg, Daan van der Tuin, and Jonathan Naaktgeboren. From my first month as a Ph.D. student, I had the privilege to be involved in your BEP project, which subsequently inspired an ambitious research project that you embraced, even though it was (initially) well beyond the techniques you knew at the time. Your perseverance over the 1.5 years turned that idea into a joint publication, and our collaboration was filled with unforgettable moments—not least your "Lord of the Rings"-inspired animation of our journey together.

My master's mentors at ETH Zürich—Julian Zilly (IDSC), Takahiro Miki (RSL), Martin Azkarate and Levin Gerdes (PRL @ ESA), Florian Achermann, Nicholas Lawrance, and Jen Jen Chung (ASL)—ignited my passion for robotics and helped me build the foundation that made this Ph.D. possible.

Beginning in the CoR department during COVID, when on-site work was limited to one day a week, was unusual. I am especially grateful to department veterans Bruno Brito and Padmaja Kulkarni, and at the time senior master student Francesco Stella, for their warm welcome. Shortly after I began my Ph.D., postdocs Pablo Borja and Sagar Joshi joined the group. Thank you both for fostering a friendly lab culture and for the many enjoyable dinners and drinks we shared. To the PhI-Lab teammates who joined later—Pietro Pustina, Tomás Coleman, Anton Bredenbeck, Jiatao Ding, Chuhan Zhang, Jingyue Liu, Ebrahim Shahabi, Giovanni Franzeso, Rodrigo Pérez-Dattari, Zhaoting Li, Mariano Ramirez Montero, Daniel Feliu Talegon, Kirsten Lussenburg, and Semanur Küçük—thank you for the great times and for our successful collaborations on research, the ICS course, and student mentoring.

I also appreciate our visiting researchers—Kyle Walker, Xiang-Yu Shao, Maja Trumić, Francesco Piqué, Michele Pierallini (a fantastic Pisa tour guide), Sonal Baberwal (an exceptionally kind collaborator), Domenico Donà, Michele Martini, and Lorenzo Paiola—for bringing fresh perspectives and for the memorable moments we shared.

Special thanks to Pietro Pustina for always being ready to help whenever I (yet again) had a question about control theory or actuation coordinates.

To my TUD-EMERGE teammates Jingyue Liu, Mariano Ramirez Montero, and Ebrahim Shahabi: thank you for your effective teamwork on our many time-consuming reports and deliverables. I also thank the entire EMERGE consortium for our productive collaboration on cutting-edge interdisciplinary topics—especially our Pisa partners Andrea Ceni, Andrea Cossu, Claudio Gallichio, and Davide Bacciu.

I'm grateful to all past and present CoR colleagues for making the department such a supportive place, cultivating a spirit of collaboration, and always being willing to lend expertise or equipment. Bas van der Heijden, Giovanni Franzeso, Rodrigo Pérez-Dattari, and Lasse Peters—thank you for the many insightful research discussions. My sincere thanks go as well to the CoR secretaries for their kindness, assistance, and for shielding us from excessive university bureaucracy.

My time at MIT was truly transformative. I thank Daniela Rus and Gioele Zardini for making the visit possible, and my many peers—Zach Patterson, Konstantin Rusch, Annan Zhang, Erfan Aasi, Emily Sologuren, Pascal Spino, Joseph DelPreto, Alaa Maalouf, Shiva Sreeram, Wei Xiao, Kiwan Wong, Xinling Li, Yujun Huang, Jiarui Li, Marius Furter, and Riccardo Fiorista—for their warm welcome and for including me in every group activity. I am especially grateful to Konstantin Rusch and Zach Patterson for their impactful collabora-

tion and effective mentorship, and to Kiwan Wong for trusting my guidance—despite initial reluctance to revisit soft-robot projects. Although my Boston schedule was hectic, the experiences and time spent with friends were essential to maintaining a healthy work-life balance. Among many others, thank you, Niccolò Pagliarani, Francesco Stella, Laurence Willemet, Luzia Knödler, Alberto Comoretto, Viola Del Bono, and Deniz Albayrak for the memorable gatherings, dinners, parties, and good times.

Organizing the Priors4Robots workshop at RSS 2024 was both insightful and fun. I thank all co-organizers—especially John Alora, Luis Pabon, and Roshan Kaundinya—for the great camaraderie (and I'm sorry I couldn't guide you around Delft in person). I also appreciate every speaker who accepted our invitation, and Michael Lutter for the fascinating private tour of Boston Dynamics.

Conferences and Ph.D. schools were among the most enjoyable parts of my doctoral journey, offering the chance to meet researchers in related fields and build lasting friendships. Niccolò Pagliarani, Burcu Seyidoglu, Francesco Stella, Brandon Caasenbrood, Nana Obayashi, Kai Junge, Zach Patterson, Annan Zhang, Lillian Chin, Ian Good, Davide Calzolari, Mariano Ramirez Montero, Daniel Feliu Talegon, and many others: thank you for the unforgettable RoboSoft adventures and the connected trips to Bali, Joshua Tree National Park, Los Angeles, Portofino, and beyond. My thanks go to the Dutch soft-robotics community for organizing outstanding symposia and Ph.D. schools, and especially to Brandon Caasenbrood, Philip Mitterbach, Benn Proper, Krishna Kommuri, Mostafa Atalla, Nick Willemstein, and Vera Kortman for the great experiences. It was a pleasure meeting Enrico Donato, Niccolò Pagliarani, Burcu Seyidoglu, Elisa Setti, and many others at the Dutch Soft-Robotics Summer School—thank you all. Enrico Donato, Elisa Setti, Philip Mitterbach, Benn Proper, Michele Martini, Pietro Pustina, and Daniele Caradonna—thank you for the exceptional culinary adventures in Rome during the soft-robotics modeling and control graduate course. Despite occasionally feeling like an outsider in the control-systems community, the mostly DLR crew—Davide Calzolari, George Pollayil, and Xuming Meng—made ACC 2022 in Atlanta a blast. ISER 2023 in Chiang Mai was equally enjoyable, thanks to Christopher Bradley, Adrian Piedra, Rafael Papallas, and Nathaniel Simon. Mónika Farsang, thank you for teaming up at DRL and for navigating the kilometer-long row of posters with me at NeurIPS 2024 in Vancouver.

To my colleagues and friends in Delft: you made this period special and joyful. Many of my original office mates—Luzia Knödler, Elia Trevision, Yujie Tang, Anna Mészáros, Saray Baker, Khaled Mustafa, and Nils Wilde—started around the same time I did, and together we navigated the highs and lows of Ph.D. and research life. I cherished our lunchtime conversations, tea breaks, and social outings—dinners, karting, skiing, bouldering, tennis, and more. Julian Schumann, thank you for supplying so many delicious cakes for our "cake breaks" and for the fun tennis matches. Tomás Coleman, Anton Bredenbeck, and Italo Belli—thank you for hosting countless barbecues and dinners that united the department. I also appreciate the many other wonderful people at CoR—Corrado Pezzato, Giovanni Franzese, Rodrigo Pérez-Dattari, Mariano Ramirez Montero, Gustavo Rezende, Alvaro Serra Gómez, Jelle Luijkx, Julian Schumann, Max Lodel, Oscar de Groot, Andreu Matoses Gimenez, Max Spahn, Lorenzo Lyons, Dennis Benders, Anastasios (Tasos) Tsolakis, Linda van der Spaan, Bas van der Heijden, Bruno Brito, Alex Ratschat, Ashwin George, Ekaterina (Katy) Karmanova, and Fiorella Sibona—for cultivating such a positive department

atmosphere.

My social support system has been vital—lifting my spirits, providing welcome distractions from research stress, and accepting the time demands of a Ph.D. I thank my Swiss friends Loris, Philip, Laurens, Daniele, and Claudio for always making me feel at home when I visited. I'm sorry the Netherlands and its beach bars didn't fully win you over a few years ago; I hope your visit for my defense changes that.

Above all, I am deeply grateful to my family—especially my mom, Sabine, and my brother, Johannes—for being the rock I can always lean on and for inspiring me to become the best version of myself.

To my life partner over this period, Léa: thank you for sharing these years with me and for the many sacrifices you've made—most notably moving to Delft / the Netherlands, to join me on this journey. I appreciate your understanding of the long workdays, weekend hours, deadline-driven vacation schedules, and the period abroad in the US that were part of this Ph.D. Your unwavering love and belief in me have been an incredible source of strength and support.

I'm sure I have inadvertently overlooked others who made meaningful contributions to this Ph.D. journey—please know you have my deepest thanks as well.

Acknowledgements Relating to Specific Chapters

- **Samenvatting** I would like to thank Saray Bakker for proofreading and revising the Dutch translation of the thesis summary.
- **Chapter 5.** The authors would like to thank Ian Good and Jeffrey Lipton from the University of Washington, U.S., for sharing the mechanical characterization published in (Good et al., 2022). We also acknowledge Sagar Joshi from the Delft University of Technology, the Netherlands, for their valuable guidance on attaching reflective markers to the HSA.
- **Chapter 6.** We would like to acknowledge Pietro Pustina for the valuable insights into a coordinate transformation into collocated variables, control of underactuated soft robots, and his help in revising Chapter 6.
- **Chapter 7.** The authors thank Dr. Fabien Lotte for his suggestions concerning the protocol, Dr. Tomas Ward and the Neuroconcise team for their support with the FlexEEG device, and J.K. Balasubramanian for his assistance with the EEG setup.
- **Chapter 9.** The authors acknowledge Ehsan Hoseini, Jasper Insinger, and Tom Salden from Delft University of Technology, the Netherlands, for their guidance in designing the PCB.
- **Appendix A.** The authors would like to acknowledge Yujun Huang and Marius Furter from the Zardini Lab at MIT and Andrew Fletcher from UC San Diego for reviewing the manuscript draft.

Maximilian Stölzle
Delft, August 2025

Curriculum Vitæ

Maximilian Stölzle

Education

03/2021 - 02/2025	Ph.D. Candidate at the Physical Intelligence (PhI) Lab, Department of Cognitive Robotics, Faculty of Mechanical Engineering, Delft University of Technology, The Netherlands. Supervisor: Prof. Cosimo Della Santina Promotor: Prof. Robert Babuška Research visits: Prof. Daniela Rus, MIT CSAIL, Cambridge, USA (two months); Prof. Gioele Zardini, MIT LIDS, Cambridge, USA (four months).
01/2019 - 02/2021	Master of Science ETH in Mechanical Engineering, ETH Zürich, Switzerland. Graduated with Distinction. Major: Robotics, Systems & Control Tutor: Prof. Emilio Frazzoli Exchange Semester: University College London (UCL), UK. Thesis: *Solving Occlusion in Digital Elevation Maps Using Neural Networks* as a joint project between RSL at ETH and the PRL at ESA.
09/2014 - 08/2018	Bachelor of Science ETH in Mechanical Engineering, ETH Zürich, Switzerland. Thesis: Control of all-wheel drives for motorcycles
08/2010 - 06/2014	Swiss Upper Secondary School Diploma, Kantonsschule Romanshorn, Switzerland. Exchange Year: Sentinel Secondary School, West Vancouver, Canada. Major: Physics and Applied Mathematics Minor: Economics & Law Matura Thesis: Adaptation to Climate Change in Northern Canada concerning Settlement Structure

Experience

04/2025 - 10/2025	Postdoctoral Researcher at the Physical Intelligence (PhI) Lab, Department of Cognitive Robotics, Faculty of Mechanical Engineering, Delft University of Technology, The Netherlands.
Since 01/2021	Technical Advisor at Quick Technologies AG, Hünenberg, Switzerland.
09/2024 - 12/2024	Visiting Researcher at the Zardini Lab, Laboratory for Information & Decision Systems (LIDS), Massachusetts Institute of Technology, Cambridge, USA.
07/2024 - 08/2024	Visiting Researcher at the Distributed Robotics Lab, Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology, Cambridge, USA.
07/2024 - 08/2024	Visiting Researcher at the Planetary Robotics Lab, ESTEC, European Space Agency.
09/2019 - 10/2020	Team-Lead Simulations & Control at Academic Space Initiative Switzerland (ARIS Space), Zürich, Switzerland.
09/2019 - 10/2020	Teaching Assistant at IWF, ETH Zürich, Switzerland.
06/2019 - 09/2019	Research Assistant at Department of Mechanical Engineering, University College London, UK.
09/2018 - 12/2018	Intern in Product Development at Zühlke Engineering, Schlieren, Switzerland.
02/2018 - 05/2018	Teaching Assistant at IWF, ETH Zürich, Switzerland.
09/2015 - 12/2020	Co-Lead Development & IT at Quap GmbH, Zürich, Switzerland.
03/2016 - 06/2026	Engineering Intern at Bühler, Uzwil, Switzerland.
08/2014 - 09/2014	Workshop Trainee at Stadler Rail, Bussnang, Switzerland.
10/2011 - 10/2011	Student Intern at GFL Consult, Dresden, Germany.

Awards

- Named as a **Rising Star** at the 2025 IEEE International Conference on Soft Robotics (**RoboSoft 2025**) with a 18 % selection rate.[9]
- RoboSoft 2024 **Best Paper Award Winner** for publication *M. Stölzle*, S. S. Baberwal*, D. Rus, S. Coyle, and C. Della Santina (2024). Guiding Soft Robots with Motor-Imagery Brain Signals and Impedance Control. In Proceedings of the 2024 IEEE 7th International Conference on Soft Robotics (RoboSoft) (pp. 1-8). IEEE* (Stölzle et al., 2024a).
- **Best Poster Award** at the RoboSoft 2025 Workshop on *Bridging Computational Disciplines for Motion Generation in Soft Robotics and Deformable Systems.*
- Cultuurfonds Wetenschapsbeurzen 2024 supporting six-month research visit at Massachusetts Institute of Technology (MIT), Cambridge, USA[10].

[9] https://robosoft2025.org/spotlight/
[10] https://www.cultuurfonds.nl/cultuurfondsbeurzen/wetenschap-beurzen

- Qualcomm Innovation Fellowship Europe 2024 finalist[11].
- MathWorks poster award at the RoboSoft 2023 workshop on *Reduced-Order Modelling for Soft Robots*.

Mentoring

Wessel Wolterbeek Muller	Research intern working on *Massively Parallel Deep Reinforcement Learning for Soft Robots*.
Nicola Visentin	Visiting master student from the Politecnico di Milano, working on his MSc. thesis with the topic *Physical Reservoir Computing with Soft Robots: Towards More Efficient GenAI via Soft Robot Design Optimization*.
Michele Martini	Visiting PhD student from the Italian Institute of Technology (IIT) working on the co-design of soft robots using differentiable simulators.
Lorenzo Paiola	Visiting PhD student from the University of Pisa, working on instilling stability guarantees into flow matching-based robot motion policies.
Iván López Broceño	MSc Robotics student at TU Delft. Master's thesis on the manipulation of ropes and other deformable objects.
Solange Gribonval	Visiting student from École Polytechnique, Paris. Research internship on the implementation of control-oriented soft robot models in JAX.
Kiwan Wong	Ph.D. student at MIT Mechanical Engineering with Prof. Gioele Zardini and Prof. Daniela Rus. Mentored the student during the visiting period at LIDS with research projects on the co-design of robotic hands and safety-aware control of soft robots using CBFs and Control Lyapunov Functions (CLFs). Resulted in paper submitted to RA-L (Wong et al., 2025).
Gabriele di Marzo	Visiting MSc student from Sapienza Università di Roma. Master thesis on *Multistable Motion Primitives*.
Ricardo Valadas	MSc Robotics student at TU Delft. Master's thesis on *Low-Dimensional Kinematic and Dynamic Model Identification for Planar Continuum Soft Robots from Image Pixels*. Resulted in a RoboSoft conference paper (Valadas et al., 2025)
Riccardo Sepe	Visiting MSc student from the University of Turin. Master thesis on *Physics-Informed Model-Based Reinforcement Learning for Soft Robot control*.
BEP 2023	Supervised a group of four TU Delft BSc Mechanical Engineering students working on *Integration of Planning, Control, and Simulation for Manipulation with a Soft Continuum Robot using Power Grasping*.

[11] https://www.qualcomm.com/research/university-relations/innovation-fellowship/2024-europe

BEP & Applied AI Project 2021	Supervised a group of four TU Delft BSc Mechanical Engineering students working on *Proprioception for Continuum Soft Robots with Magnetic Sensors*. Resulted in Soft Matter journal paper (Baaij et al., 2023).
Emanuele Rosi	Visiting MSc student from the University of Genoa. Master's thesis on *Sensing soft robots' shape with cameras: an investigation on kinematics-aware SLAM*. Resulted in a RoboSoft conference paper (Rosi et al., 2022).

Teaching

Intelligent Control Systems 2025	Onboarding of the new ICS Practical Assignment (PA) TA team.
Intelligent Control Systems 2024	Lead the creation & revision of the ICS Practical Assignment (PA), TA for the Q&A sessions, answered questions on the Brightspace forum, graded the submitted PAs.
Intelligent Control Systems 2023	Lead the creation of the ICS Practical Assignment (PA), contributed new exercises to the PA (e.g., learning a model with Lagrangian Neural Networks (Lutter et al., 2019), Iterative Learning Control, etc.), TA for the Q&A sessions, answered questions on the Brightspace forum, graded the submitted PAs.
Intelligent Control Systems 2022	Assisted with the ICS literature assignment (e.g., proposing topics, chairing the symposium, grading submitted assignments).

Academic service

Organizer	RSS 2024 Workshop on Structural Priors as Inductive Biases for Learning Robot Dynamics
Reviewer	EAAI 2022 ERF 2024 EIC Pathfinder Challenge WS Humanoids 2024 ICRA 2022-2024 IJRR 2023, 2025 IROS 2023 L-CSS 2025 NeurIPS 2025 npj robotics 2024 RA-L 2021-2025 RAM 2022 RoboSoft 2022-2025 ROPM 2022 RSS 2024 Priors4Robots WS T-RO 2021-2022

Other Activities

EMERGE	Led the effort and writing of deliverables D3.1[12] & D3.2[13] and contributed to deliverables D3.3 and D4.1.

[12] https://edu.nl/qnny4
[13] https://edu.nl/97kqc

List of Publications

Referred journals

1. **M. Stölzle**, N. Pagliarani*, D. Caradonna*, D. Rus, E. Falotico, M. Cianchetti, C. Della Santina*, and G. Zardini*. A Metric for Quantifying the Safety of Soft Robots. ***In Preparation***.

2. (Buriani et al., 2025) G. Buriani*, J. Liu*, **M. Stölzle**, C. Della Santina, and J. Ding (2025). Symbolic Learning of Interpretable Reduced-Order Models for Jumping Legged Robots. IFAC Journal of Systems and Control, ***Under Review***.

3. (Stölzle et al., 2025b) **M. Stölzle**, T.K. Rusch*, Z.J. Patterson*, R. Pérez Dattari, F. Stella, J. Hughes, C. Della Santina, and D. Rus (2025). Learning to Move in Rhythm: Task-Conditioned Motion Policies with Orbital Stability Guarantees. In Science Robotics Special issue on Transfer Learning, Foundation Models, and Emerging Capabilities, ***Under Review***.

4. (Wong et al., 2025) K. Wong, **M. Stölzle**, W. Xiao, C. Della Santina, D. Rus*, and G. Zardini* (2025). Contact-Aware Safety in Soft Robots Using High-Order Control Barrier and Lyapunov Functions. In Robotics and Automation Letters (RA-L) Special Issue on Interdisciplinarity and Widening Horizons in Soft Robotics, ***Accepted***.

5. (Stölzle et al., 2025a) **M. Stölzle***, N. Pagliarani*, F. Stella, J. Hughes, C. Laschi, D. Rus, M. Cianchetti, C. Della Santina, and G. Zardini (2025). Soft yet Effective Robots via Holistic Co-Design. In Nature Machine Intelligence, ***Under Review***.

6. (Shao et al., 2023) X. Shao, P. Pustina*, **M. Stölzle***, G. Sun, A. De Luca, L. Wu, and C. Della Santina (2023). Model-based control for soft robots with system uncertainties and input saturation. IEEE Transactions on Industrial Electronics, 71(7), 7435-7444.

7. (Baaij et al., 2023) T. Baaij*, M. K. Holkenborg*, **M. Stölzle***, D. van der Tuin*, J. Naaktgeboren, R. Babuška, and C. Della Santina (2023). Learning 3D shape proprioception for continuum soft robots with multiple magnetic sensors. Soft Matter, 19(1), 44-56.

8. (Stölzle et al., 2022) **M. Stölzle**, T. Miki, L. Gerdes, M. Azkarate, and M. Hutter (2022). Reconstructing occluded elevation information in terrain maps with self-supervised learning. IEEE Robotics and Automation Letters, 7(2), 1697-1704.

9. (Stölzle and Della Santina, 2021) **M. Stölzle**, C. Della Santina (2021). Piston-driven pneumatically-actuated soft robots: Modeling and backstepping control. IEEE Control Systems Letters, 6, 1837-1842.

Referred conference proceedings

1. (Valadas et al., 2025) R. Valadas*, **M. Stölzle***, J. Liu, and C. Della Santina (2025). Learning Low-Dimensional Strain Models of Soft Robots by Looking at the Evolution of Their Shape with Application to Model-Based Control. In Proceedings of the 2025 IEEE 8th International Conference on Soft Robotics (RoboSoft) (pp. 1-8). IEEE.

2. (Stölzle and Della Santina, 2024) **M. Stölzle**, and C. Della Santina (2024). Input-to-State Stable Coupled Oscillator Networks for Closed-form Model-based Control in Latent Space. In Proceedings of Advances in Neural Information Processing Systems (NeurIPS) 37, **Spotlight (top 10% of accepted papers)**.

3. (Stölzle et al., 2024a) **M. Stölzle***, S. S. Baberwal*, D. Rus, S. Coyle, and C. Della Santina (2024). Guiding Soft Robots with Motor-Imagery Brain Signals and Impedance Control. In Proceedings of the 2024 IEEE 7th International Conference on Soft Robotics (RoboSoft) (pp. 1-8). IEEE. **Best Paper Award**.

4. (Stölzle et al., 2024b) **M. Stölzle**, D. Rus, and C. Della Santina (2024). An Experimental Study of Model-based Control for Planar Handed Shearing Auxetics Robots. In Experimental Robotics: The 18th International Symposium (ISER) (pp. 153-167). Springer.

5. (Ceni et al., 2024) A. Ceni*, A. Cossu*, **M. Stölzle**, J. Liu, C. Della Santina, D. Bacciu, and C. Gallicchio (2024). Random Oscillators Network for Time Series Processing. In Proceedings of The 27th International Conference on Artificial Intelligence and Statistics (AISTATS) (pp. 4807-4815). PMLR.

6. (Stölzle et al., 2023) **M. Stölzle**, L. Chin, R. L. Truby, D. Rus, and C. Della Santina (2023, April). Modelling Handed Shearing Auxetics: Selective Piecewise Constant Strain Kinematics and Dynamic Simulation. In 2023 IEEE International Conference on Soft Robotics (RoboSoft) (pp. 1-8). IEEE.

7. (Rosi et al., 2022) E. Rosi*, **M. Stölzle***, F. Solari, and C. Della Santina (2022, April). Sensing Soft Robots' Shape with Cameras: an Investigation on Kinematics-Aware SLAM. In 2022 IEEE 5th International Conference on Soft Robotics (RoboSoft) (pp. 795-801). IEEE.

8. (Phillips et al., 2021) T. Phillips*, **M. Stölzle***, E. Turricelli*, F. Achermann, N. Lawrance, R. Siegwart, and J. J. Chu (2021, May). Learn to Path: Using Neural Networks to Predict Dubins Path Characteristics for Aerial Vehicles in Wind. In 2021 IEEE International Conference on Robotics and Automation (ICRA) (pp. 1073-1079). IEEE.

Included in this thesis.
* indicates equal contribution.
Won a best paper award.

Invited Talks and Dissemination

27/04/2025 *Soft yet Effective Robots via Holistic Co-Design* at 25-Years of Embodied Intelligence Workshop, Lausanne, Switzerland. Contributed breakout talk.

25/04/2025 *Safe Control of Soft Robots: Bridging Physics and Learned Models* at 2025 IEEE International Conference on Soft Robotics (RoboSoft 2025), Lausanne, Switzerland. **Rising Star** talk.

24/04/2025 *Closed-Form Control of Continuum Soft Robots with Learned Latent Oscillatory Models* at the RoboSoft 2025 Workshop on Bridging Computational Disciplines for Motion Generation in Soft Robotics and Deformable Systems, Lausanne, Switzerland. Poster presentation, **Best Poster Award**.

02/04/2025 *Safe yet Effective Soft Robots via Holistic Co-Design* at 5th International Conference on Embodied Intelligence, Online. Contributed masterclass talk.

04/12/2024 *Model-based Control of Planar Handed Shearing Auxetics Robots* at MRS Fall Meeting, Boston, USA. Contributed talk.

01/10/2024 *Integrating Physical Structure and Stability Guarantees into the Learning of Robot Models & Motion Policies* at MIT LIDS Autonomy Tea Talk, Cambridge, USA. Invited talk.

12/09/2024 *Designing Compact Models for the Control of Soft Robots* at MIT Zardini Lab Group Meeting, Cambridge, USA. Invited talk.

20/08/2024 *Deriving and Learning Minimalist Models for the Control of Soft Robots* at MIT Distributed Robotics Lab (DRL) Group Meeting, Cambridge, USA. Invited talk.

15/07/2024 *Leveraging Coupled Oscillator Networks as a Structural Prior when Learning Latent Dynamics from Pixels* at RSS 2024 Workshop on Structural Priors as Inductive Biases for Learning Robot Dynamics, Delft, Netherlands. Spotlight talk (oral). 15 % acceptance rate (3/27), peer-reviewed, **rated by reviewers as the best submission**.

25/06/2024 *Input-to-State Stable Coupled Oscillator Networks for Closed-form Model-based Control in Latent Space* at TU Delft AI PhD Poster Day 2024, Delft, Netherlands. Poster presentation.

05/2024 *Model-based Control of Planar Handed Shearing Auxetics Robots* at Dutch Soft Robotics Symposium 2024, Eindhoven, Netherlands. Invited talk.

05/2023 *Modelling Handed Shearing Auxetics: Selective Piecewise Constant Strain Kinematics and Dynamic Simulation* at Dutch Soft Robotics Symposium 2023, Enschede, Netherlands. Poster presentation.

04/2023 *Learning 3D Shape Proprioception for Continuum Soft Robots with Multiple Magnetic Sensors* at RoboSoft 2023 Workshop on Reduced-Order Modelling for Soft Robots, Singapore. Poster presentation. **Poster presentation award (3rd-place).**

04/2023 *Learning 3D Shape Proprioception for Continuum Soft Robots with Multiple Magnetic Sensors* at 3rd International Conference on Embodied Intelligence, Online. Contributed talk.

09/2022 *Learning 3D Shape Proprioception for Continuum Soft Robots with Multiple Magnetic Sensors* at 15th International Workshop on Human-Friendly Robotics (HFR), Delft, Netherlands. Poster presentation.

05/2022 *Learning 3D Shape Proprioception for Continuum Soft Robots with Multiple Magnetic Sensors.* at ICRA 2022 Workshop on Compliant Manipulation, Philadelphia, USA. Poster presentation.

04/2022 *Learning 3D Shape Proprioception for Continuum Soft Robots with Multiple Magnetic Sensors* at RoboSoft 2022 Workshop on Soft Sensing: Environment, Morphology, Brain in Biology and Robotics, Edinburgh, United Kingdom. Poster presentation.

04/2022 *Piston-Driven Pneumatically-Actuated Soft Robots: modeling and backstepping control* at RoboSoft 2022 Workshop on Soft Robotics modeling: what are we missing?, Edinburgh, United Kingdom. Poster presentation.

12/2021 *Solving Occlusion in Terrain Mapping with Neural Networks* at NeurIPS 2021 4th Robot Learning Workshop: Self-Supervised and Lifelong Learning, Online. Poster presentation.

05/2021 *Piston-Driven Pneumatically-Actuated Soft Robots: modeling and backstepping control* at Benelux Workshop on Systems and Control 2021, Rotterdam, Netherlands. Contributed talk.

Propositions

accompanying the dissertation

Safe yet Precise Soft Robots

Incorporating Physics into Learned Models for Control

by

Maximilian Stölzle

1. Integrating physical priors into learned dynamical models enables computationally efficient, provably stable control while also enhancing model expressiveness through data-driven learning. [Core Contribution]

2. A key limitation in the soft robotic literature is that the tradeoff between performance and safety remains insufficiently quantified and exploited. [Chapter 3]

3. Substantial progress in soft robot proprioception can be achieved by optimizing the hardware and software integration of established commercial sensing modalities—such as cameras or IMUs—instead of relying on novel, yet unproven, technologies. [Chapter 4]

4. We need to strive for compliance in both body and brain when operating robots close to humans. [Chapter 7]

5. A primary barrier to advancing soft robotics research is the absence of integrated benchmarks and baselines.

6. Simply increasing the quantity and diversity of data is not enough to develop effective, robust robot models and motion policies unless we also integrate physical structure and stability guarantees.

7. Although motion policies jointly learned across different robots can improve generalization, specialized policies for each robot are essential to fully exploit its characteristics and maximize performance on highly dynamic tasks.

8. Universities expanding non-managerial career paths solely dedicated to research would enhance research quality and its societal impact.

9. Restructuring large tech companies as worker cooperatives would promote balanced decision-making, enhance inclusivity, and reduce societal division.

10. Unless international treaties that establish guidelines and guardrails for deploying robots and AI are implemented within this decade, the societal drawbacks of robotics are likely to outweigh its benefits.

These propositions are regarded as opposable and defendable and have been approved as such by the promotor Prof. Dr. R. Babuška and the copromotor Dr. C. Della Santina.


$\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q} = \mathcal{Q}$
$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q} = \mathcal{Q}$