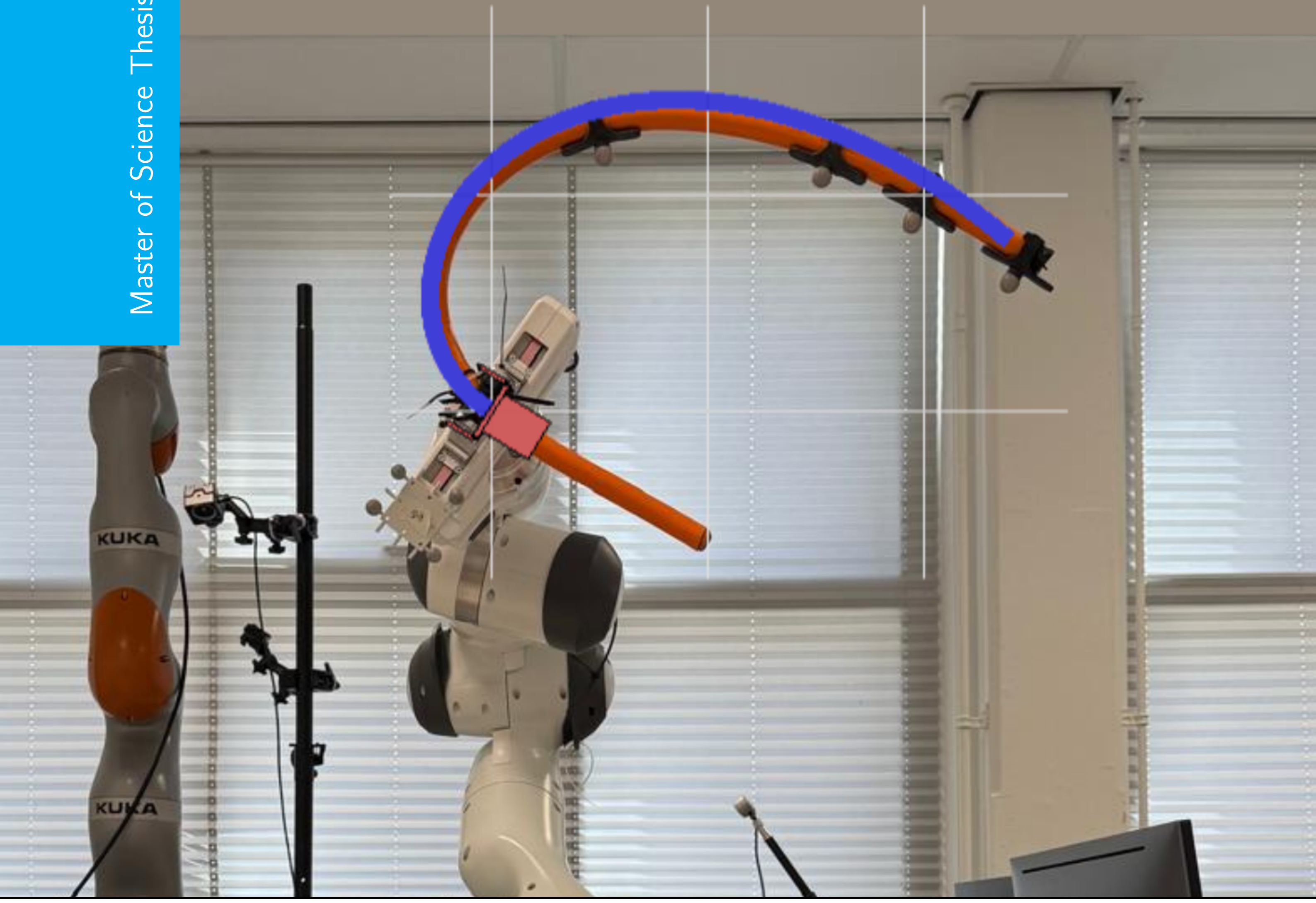
Optimal Control of Slender Soft Robots in Low-Stiffness Regimes

A Model Based Approach

Nikhil Nair

Master of Science Thesis



Delft Center for Systems and Control

Optimal Control of Slender Soft Robots in Low-Stiffness Regimes

A Model Based Approach

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Nikhil Nair
5997054

September 11, 2025

Supervisors:	Prof. Tamás Keviczky, Delft Center for Sytems and Control, TU Delft
	Prof. Cosimo Della Santina, Cognitive Robotics Department, TU Delft
Daily Supervisor:	Dr. Daniel Feliu Talegón, Cognitive Robotics Department, TU Delft

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

This study is a collaboration between the Delft Center for Systems and Control and the Cognitive Robotics Department at TU Delft.

Delft Center for
Systems and Control

Cognitive
Robotics

TU Delft
Delft
University of
Technology

Abstract

Continuum soft robots present significant opportunities for advancing robotics, but they also introduce substantial technical challenges. These systems are highly nonlinear, infinite-dimensional, and severely underactuated, making control particularly difficult. While recent advancements in model-based control have addressed some of these issues for soft robotics, numerical optimal control has shown strong potential, especially given its success in other severely underactuated domains such as bipedal and quadrupedal locomotion.

However, the application of optimal control in soft robotics has largely relied on simplified models, and its use with more accurate and geometrically consistent formulations remains underexplored, particularly for explicitly tackling underactuation. This thesis investigates the use of Differential Dynamic Programming (DDP) to control continuum soft robots modeled using the Geometric Variable Strain (GVS) framework. The focus is on the Soft Inverted Pendulum (SIP) as a template system to evaluate DDP's feasibility, robustness, and performance in underactuated settings, including low-stiffness regimes where collocated feedback strategies break down. The implementation leverages the use of analytical gradients computed via the Recursive Newton-Euler Algorithm (RNEA) to improve convergence and computational efficiency.

The results show that DDP outperforms traditional Partial Feedback Linearization (PFL) methods, both collocated and non-collocated, especially across challenging mass-stiffness combinations. This effectively extends control authority and stability into regimes previously considered difficult to handle. This thesis extends the method to more complex hybrid soft–rigid systems, examining real-time feasibility and practical implementation, thereby laying the foundation for a generalizable optimal control framework for soft robots.

Table of Contents

List of Figures

List of Tables

Acknowledgement

I would like to express my deepest gratitude to my supervisors, Prof. Tamás Keviczky, Prof. Cosimo Della Santina, and my daily supervisor Dr. Daniel Feliu Talegón, for their invaluable support and guidance throughout my thesis.

First, I would like to thank Prof. Tamás Keviczky for giving me the opportunity to pursue this project and for encouraging me to explore the field of soft robotics. His advice and direction at the beginning of my thesis were instrumental in shaping this work.

I am especially grateful to Prof. Cosimo Della Santina for welcoming me into the Physical Intelligence Lab within the Cognitive Robotics Department. Working in this environment has been a truly rewarding experience. The bi-weekly meetings, the stimulating discussions, and the exposure to the cutting-edge research in soft robotics have greatly enriched my academic journey. I am also thankful for his inspiring review article in IEEE Control Systems Magazine, which first introduced me to the field of soft robotics and sparked my interest in the control of such systems.

My heartfelt thanks also go to Dr. Daniel Feliu Talegón, my daily supervisor, for his consistent support, encouragement, and patience. His guidance in modeling approaches, his insights during our weekly meetings, and his dedication to helping me refine my ideas have been invaluable. His mentorship not only improved the quality of my thesis but also motivated me to pursue my best work.

I would also like to thank my friends at DCSC for sharing this journey with me. The challenges of the Master's program were made easier and more enjoyable thanks to your companionship, collaboration, and the diverse research we all pursued.

Most importantly, I wish to thank my family—my parents, Saji and Sangeetha, and my sister, Navya—for being my unwavering foundation. Your constant encouragement, countless sacrifices, and unconditional belief in me have carried me through every step of this journey.

Finally, I want to thank Varsha, my greatest source of strength and love. Thank you for standing by me through every high and low, for being my most honest critic and my loudest cheerleader, and for surrounding me with your endless patience, care, and encouragement. None of this would have been possible without you, and I am endlessly grateful for the love and support you have given me.

Chapter 1

Introduction

Soft robotics has grown rapidly over the past decade, driven by advances in materials, fabrication, and modeling [3]. Among these threads, control has seen major progress, yet remains challenging due to pronounced compliance, distributed deformation, and sensing–actuation constraints [10]. Model-based approaches typically yield superior performance to purely data-driven baselines when accurate control-oriented models are available, but their adoption in soft robotics has lagged because such models are difficult to derive and compute.

A distinctive feature of the field is its *reversed* trajectory of controller development: early successes relied on learning and data-driven methods, whereas model-based controllers matured later as tractable continuum models and efficient solvers emerged. This thesis focuses on *continuum* soft robots which are robotic analogues of invertebrate limbs (e.g., octopus arms, elephant trunks) where compliance confers safety and the ability to store/release elastic energy, enabling agile and energy-efficient behaviors that still elude rigid robots.

Optimal control techniques, although promising, have been relatively under explored in this domain. This limited adoption can be largely attributed to the difficulty in modeling soft robots and the preference for simpler controllers that are easier to implement on hardware.

This thesis investigates optimal control within a model based framework for soft robot control. It proposes and implements a nonlinear model predictive control and differential dynamic programming framework that extends stabilizability in underactuated settings, targeting low stiffness regimes where collocated methods fail. The work also develops an efficient formulation that uses analytical dynamics gradients, warm starts, and few solver iterations to achieve computation fast enough for real time closed loop control.

Challenges in the Control of Soft Robots

As highlighted in [10], soft robotic platforms have advanced rapidly in mechanical design and robustness, yet widespread deployment remains limited by the difficulty of achieving precise and reliable control. Unlike rigid robots, where control theory matured alongside accurate models, soft robotics initially faced a modeling bottleneck. This challenge led to early reliance on machine learning and other model-free strategies.

The viewpoint has shifted for two reasons. First, empirical and theoretical results show that feedback control can tolerate simplified or approximate models and often outperforms model-free methods in closed loop. Second, a growing set of reduced-order models now provides tractable and interpretable representations of soft-robot dynamics. Together these developments reopen the door to model-based control, including optimal control, for soft systems.

In practice, many implementations assume fully actuated approximations. This can work, but underactuation is fundamental in soft robotics and must be addressed rather than avoided. Not every equilibrium is attainable, and there may be no time-varying torque that reaches an arbitrary state from any initial condition. The design objective is therefore to harness underactuation, not to eliminate it, and to develop controllers that exploit elasticity and distributed compliance.

In the underactuated case, a common strategy stabilizes the actuated coordinates while relying on elasticity to stabilize the unactuated coordinates [10]. This coupling between actuation and intrinsic elasticity is central to controller design for soft systems.

Handling underactuation in general remains challenging. Most model-based approaches target specific subclasses where provable stability can be established, with collocated feedback as a prominent example. These methods provide valuable guarantees but operate in restricted parameter regions. This limitation motivates the exploration of optimal control with physics-based models, which can leverage full-state predictions to extend stabilizability beyond the regimes covered by collocated feedback.

1-1 Research Question

Building on the motivation outlined above, this report focuses on addressing the following research question:

Research Question

How can soft robots in low-stiffness regimes be controlled accurately and efficiently using optimal control?

Contributions

- A model-based NMPC formulation for continuum soft robots that exploits analytical dynamics gradients for fast DDP iterations.
- A feasibility study against PFL baselines, highlighting stabilization in mass–stiffness regions where zero-dynamics constraints preclude PFL.
- Robustness and disturbance-rejection evaluations, and generalization to additional soft-/hybrid systems within the same modeling–control framework.

- A hardware pipeline combining OptiTrack-based shape reconstruction with offline trajectory execution on a Franka Emika Panda manipulating a deformable object.

1-2 Report Structure

The report is organized as follows. Chapter 2 introduces the foundational concepts and tools for modeling and control of soft robots. Chapter 3 states the research problem, outlines the proposed modeling and control approach, and highlights the gaps in the literature that motivate the methodology. Chapter 4 presents the controller results and validation strategy. Chapter 5 concludes the work and outlines directions for future research.

Chapter 2

Soft Robotics – Modeling and Control

This chapter introduces the foundational concepts and tools necessary for understanding the modeling and control of soft robots. The chapter begins with an overview of soft robotics, highlighting key features such as compliance, continuum deformation, and the implications these have on traditional control strategies.

The discussion then transitions into mathematical modeling frameworks for soft robots, presenting a classification of common approaches with a focus on continuum mechanics. In particular, the chapter emphasizes the Director-based modeling paradigm and introduces the Geometric Variable Strain (GVS) approach as a physically consistent and control-oriented formulation for slender soft bodies. To provide full context, alternative models such as Piecewise Constant Curvature (PCC) and polynomial curvature models are also discussed, primarily to motivate the modeling choices made in this work.

The chapter then delves into the control of soft robots, beginning with a discussion of model-based control and an overview of the current state of the art. This is followed by a focused exploration of optimal control methods used in the control of soft robots.

By the end of this chapter, the reader will be equipped with the theoretical background necessary to understand the modeling assumptions, control formulations, and computational methods employed in the remainder of this thesis.

2-1 Soft Robotics: Motivation and Foundations

Soft robotics has developed as a response to the limitations inherent in traditional rigid-bodied robots, particularly in tasks involving unstructured environments, delicate interaction, or physical adaptability. While rigid robots are effective in industrial settings that prioritize precision and repeatability, they often lack the compliance and mechanical intelligence necessary for safe and robust operation in dynamic conditions. In contrast, biological organisms such as the octopus [19], cheetah [20], and elephant [21] display remarkable mobility and manipulation capabilities. These are largely attributed to the compliant nature of their musculoskeletal structures.

Biological softness, as exhibited by muscles, tendons, and ligaments, enables distributed compliance, efficient energy storage, and impact resilience. Such mechanical properties allow animals to traverse complex terrains, absorb shocks, and interact safely with their surroundings. Kim et al. [22] describe how the integration of soft tissues contributes to both robustness and energy efficiency in nature, inspiring analogous principles in robotics. The field of soft robotics, therefore, seeks to incorporate these advantages through the use of compliant materials and bioinspired designs.

The term “soft robot” was initially associated with pneumatic systems such as the McKibben artificial muscle [23]. Historical contributions to the field include the development of a wearable exoskeleton in 1957 [24], the Orm continuum manipulator in 1966 [25], and flexible microactuators in the early 1990s [26]. During the 2000s, interest in bioinspired robotics increased steadily, eventually leading to the formalization of soft robotics as a distinct subfield. A notable rise in publications began around 2011, reaching over 100 annually by 2015 [1]. Figure 2-1 illustrates the historical development and rapid growth of the field.

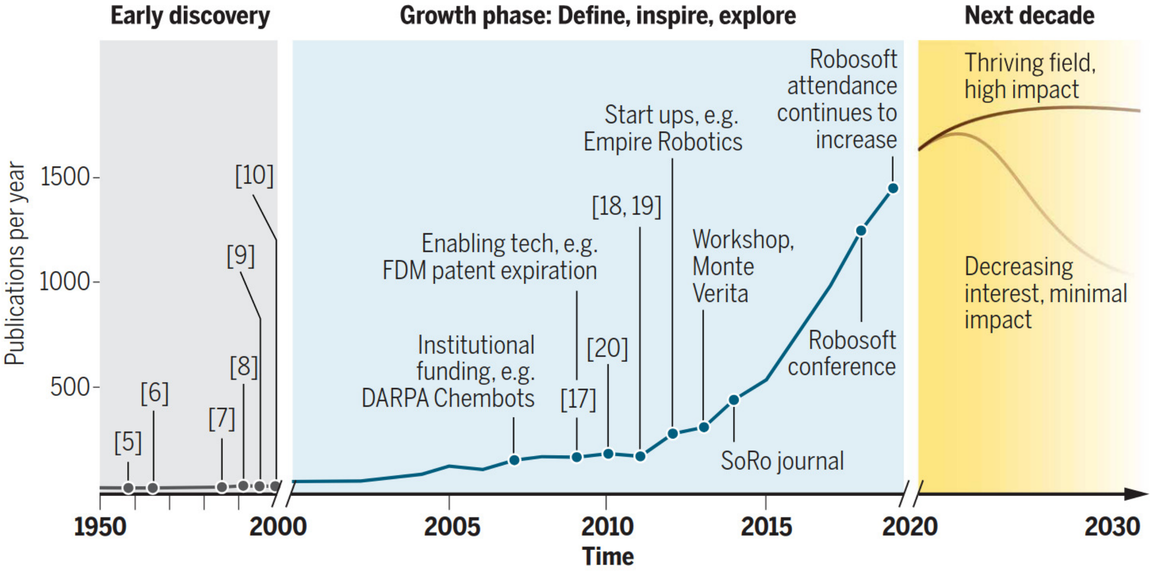


Figure 2-1: The growth of soft robotics.[1]

Soft robots have demonstrated potential across diverse domains, including automated handling in the food industry [27], marine exploration [28], and minimally invasive surgical procedures [29]. Their compliance allows for safe operation in human environments, adaptability to uncertain conditions, and functionality in tasks that demand dexterous manipulation.

2-1-1 Definition and Classification

According to the description in [2], soft robots are characterized by the deliberate integration of compliant components within their mechanical structure. These systems represent a shift in design philosophy from the traditional principle of "design for precision, control for safety" to "design for safety, control for performance."

Soft robots can be broadly categorized into two types. Articulated soft robots incorporate flexible elements at joints or segments and are typically inspired by vertebrate structures [30], including designs modeled after snakes, birds, and quadrupeds [31, 32, 33]. Continuum soft robots, on the other hand, draw inspiration from invertebrates and feature continuously deformable bodies capable of bending and twisting in multiple directions [34, 35, 36].

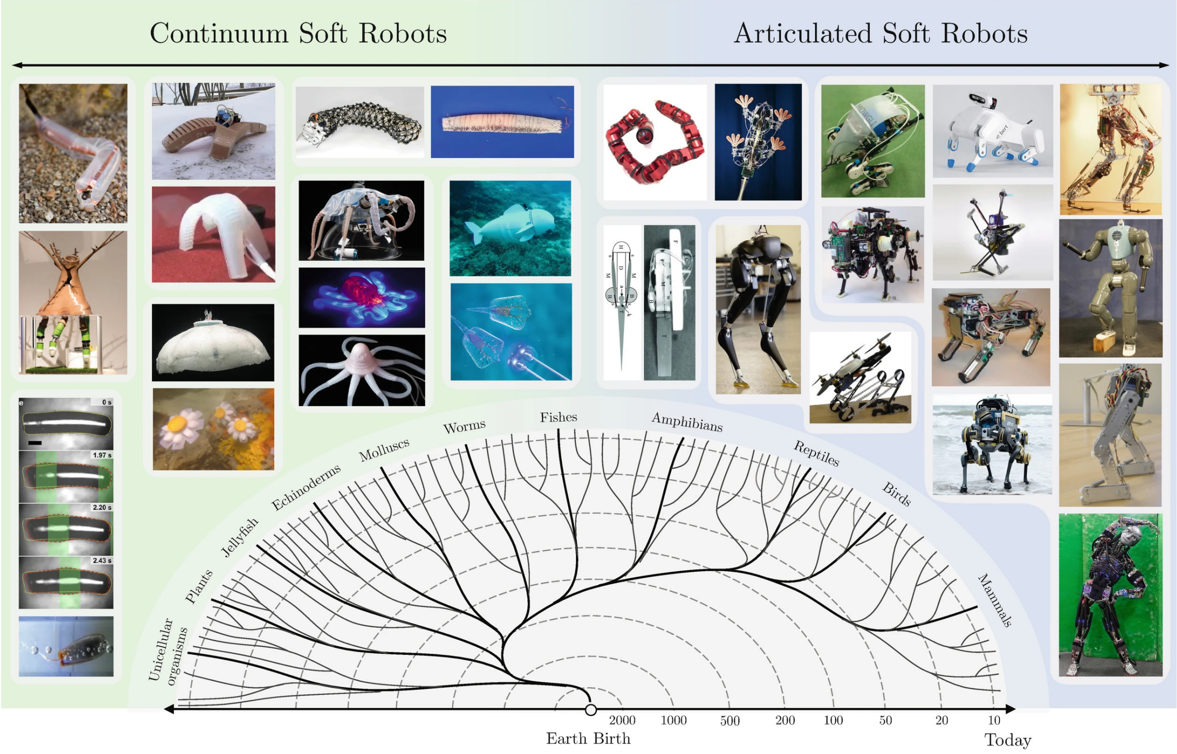


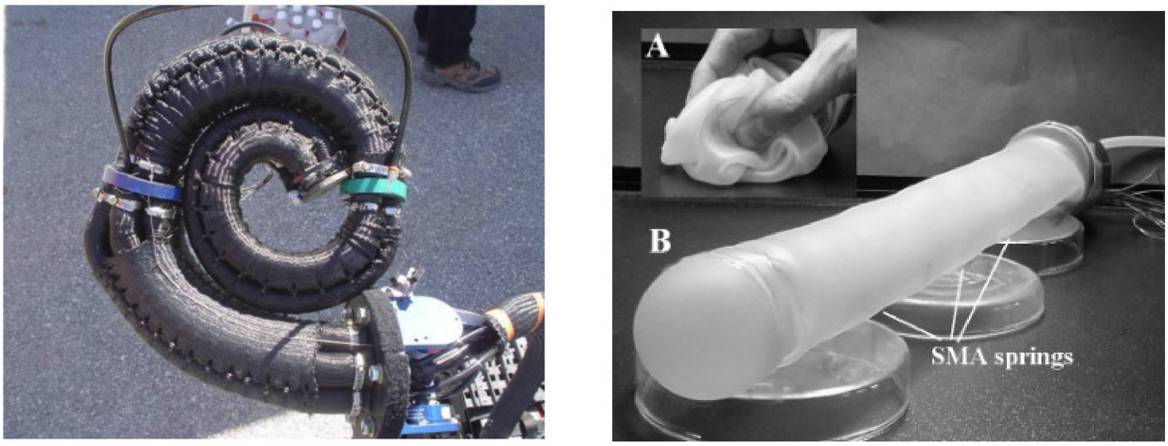
Figure 2-2: Classification of soft robots based on biological inspiration. [2]

2-1-2 Continuum Soft Robots

Continuum soft robots are defined by their ability to deform continuously along their bodies, in contrast to rigid-link mechanisms. These systems are constructed using materials with mechanical properties similar to biological tissues. As reported by Rus and Tolley [37], suitable materials typically exhibit Young's moduli in the range of 10^4 to 10^9 pascals. Examples include the OctArm manipulator [38], designed for hyper-redundant manipulation tasks, and the SoftBot [39], which emulates caterpillar-like crawling.

2-1-3 Materials, Actuation, and Sensing

Materials. Soft robots are typically composed of elastomers, hydrogels, or silicone-based materials [22, 37], which provide both flexibility and resilience. These materials can be fabricated using methods such as soft lithography [40], shape deposition manufacturing [41], and multimaterial 3D printing [42].



(a) OctArm [38] **(b)** SoftBot [39]

Figure 2-3: Early examples of continuum soft robots [3]

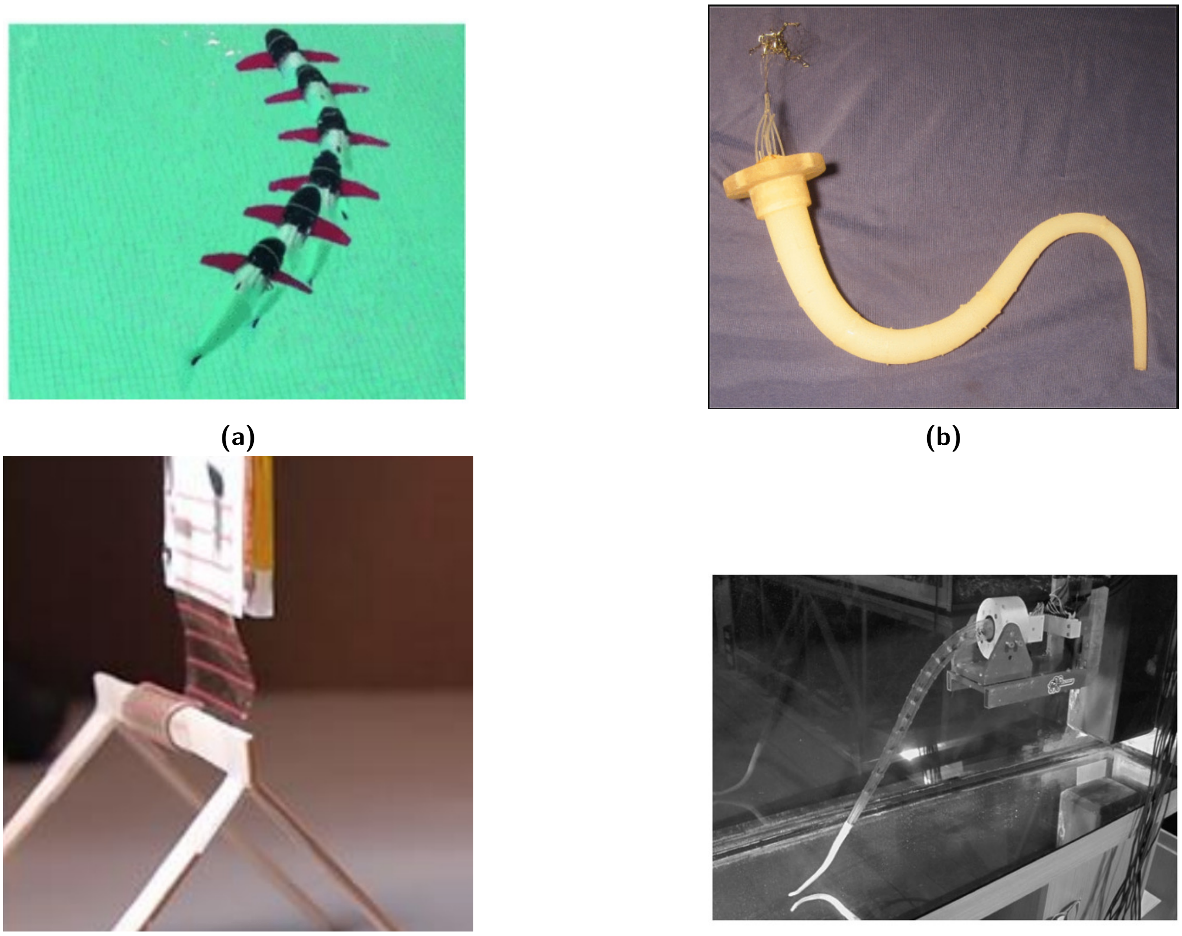
Actuation. Several actuation strategies are employed in soft robotic systems. Fluidic actuators use pneumatic or hydraulic pressure to inflate chambers, producing deformation [43, 4]. Tendon-driven mechanisms incorporate cables routed through soft substrates, which are actuated by external motors [5]. Electroactive polymers and dielectric elastomer actuators deform in response to electric fields [6], while shape memory alloys deform thermally in response to electrical current [7].

Sensing. The integration of sensing capabilities in soft robots remains a technical challenge due to the lack of rigid frames. Conventional sensors often restrict deformation, motivating the use of flexible alternatives. Resistive strain sensors [44], capacitive stretch sensors [45], and optical waveguides [46] have been embedded into soft structures to provide proprioception and tactile feedback. Advances in stretchable electronics have further enhanced the sensing capabilities of soft robotic platforms [47].

2-1-4 Applications

Soft robots are particularly well-suited to applications requiring compliance and adaptability. In manipulation tasks, they are able to conform to object shapes and modulate contact forces, allowing safe interaction with fragile or irregular items [48, 49]. For exploration, soft-bodied systems offer the ability to traverse narrow passages and withstand harsh environmental conditions [50, 51]. In healthcare, soft robotics has enabled developments in assistive devices [52], surgical instrumentation [53], and targeted drug delivery systems [54, 55].

This foundational overview establishes the context for the challenges addressed in this thesis. The next sections focus on the modeling and control of soft robots, particularly under conditions of underactuation.



(a) **(b)** **(c)** **(d)**

Figure 2-4: Examples of continuum soft robots based on actuation sources: **(a)** hydraulically actuated FEA tail for a soft-robotic fish [4], **(b)** continuum soft robot arm driven by 12 cables embedded within a silicone body [5], **(c)** dielectric elastomer gripper lifting a wood beam [6], **(d)** octopus-like arm using an SMA-based actuation system [7].

2-2 Modeling of Continuum Soft Robots

This section introduces the mathematical modeling of soft robots, a field that presents unique challenges due to the highly flexible and deformable nature of these systems. However, recent years have witnessed a surge in theoretical tools for modeling soft robots, leveraging on other scientific disciplines such as continuum mechanics, computational mechanics, and computer graphics.

The primary aim of this section is to provide an overview of the diverse theoretical approaches for modeling soft robots, with a particular focus on methods that are relevant to the proposed approach in the thesis. Specifically, the section highlights models relevant to continuum soft robots that can be represented as slender rods with high length-to-width aspect ratios.

In their exact form, continuum soft robots are infinite-dimensional systems, governed by highly nonlinear partial differential equations (PDEs) that are not analytically solvable [8].

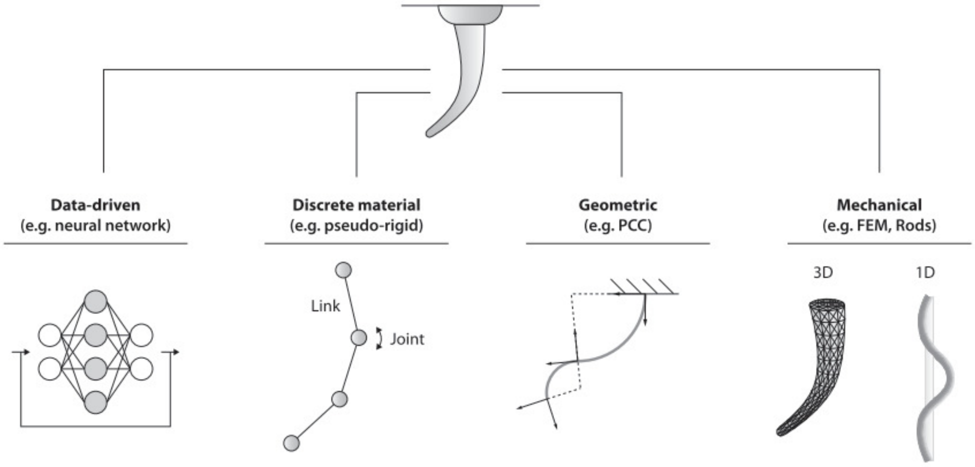


Figure 2-5: Overview of modeling techniques for continuum soft robots as described in [8]. Image from [9].

As emphasized in [8], a fundamental challenge in soft robot modeling is balancing accuracy with computational tractability, especially for control and optimization applications. While numerous reviews have addressed the mathematical modeling of soft robots, the comprehensive literature survey in [8] forms the basis for this discussion. The classification framework proposed in that paper, organized around mathematical techniques, is particularly insightful. Here, however, the primary focus will be on models specifically designed for slender soft robots. According to the classification scheme, soft robot models are categorized as follows:

Continuum Mechanics Models: These models are rooted in the classical theory of continuum mechanics and aim to describe the behavior of soft bodies by treating them as continuous deformable media. The underlying formulation involves defining strain measures, constitutive laws, and balance equations, resulting in partial differential equations (PDEs) that capture the system's nonlinear and distributed dynamics.

A general-purpose approach to solving these PDEs involves discretizing the problem domain using the **Finite Element Method (FEM)**. FEM approximates the configuration field over a mesh of finite elements and has been widely adopted in the soft robotics literature [56, 57]. While FEM provides a powerful simulation framework, its computational complexity can become prohibitive for real-time control, unless combined with model order reduction techniques [58].

For slender soft robots—systems with a high length-to-diameter ratio—a more specialized and efficient modeling paradigm is the **director-based approach**. Here, the soft body is modeled as a Cosserat rod: a spatial curve (the material axis) along which rigid cross-sections are distributed. Each cross-section is equipped with an orthonormal frame (directors) that tracks local orientation. The configuration of the rod is represented as a curve on the Lie group $SE(3)$:

$$\mathbf{g}(\cdot) : X \in [0, L] \mapsto \mathbf{g}(X) \in SE(3), \tag{2-1}$$

where $\mathbf{g}(X)$ encodes the pose of the cross-section at position X.

This formulation captures bending, twisting, shearing, and stretching within a geometrically exact framework. Recent work has recast this theory using Lie group methods to derive dynamic equations in a coordinate-invariant form. One of the most prominent methods in this class is the **Geometric Variable-Strain (GVS)** approach [59, 60], which parameterizes strain fields using modal basis functions and derives reduced-order Lagrangian dynamics. GVS preserves physical structure, enables efficient computation, and is particularly amenable to model-based control.

Geometrical Models: This category of models is based on the assumption that the deformed shape of a soft robot conforms to a specific geometric pattern. These models typically fall under the broader Cosserat rod framework, but unlike continuum mechanics models that directly solve the governing PDEs, geometrical models adopt a reduced-order approach by prescribing the system's kinematics through generalized coordinates.

Rather than discretizing continuous equations, these methods postulate a shape—often inspired by observations of how actuators deform soft structures—and build the model from geometric or variational principles such as the Euler–Lagrange equations. This makes them computationally efficient and particularly attractive for real-time control and planning.

Geometrical models can be broadly classified into two subgroups: *functional models* and *piecewise-constant curvature (PCC) models.*

1. **Functional Approaches:** These models describe the backbone of the robot using smooth, parameterized curves. A common method is the *modal approach* [61], which represents the shape through a set of basis functions weighted by time-varying coefficients. This yields a compact set of generalized coordinates suitable for control and planning.
2. **PCC Models:** PCC models approximate the soft body as a series of constant-curvature segments, each bending in a fixed plane. This assumption holds in many practical actuated systems and simplifies kinematic modeling. They are often used in a kinematics-based form, where actuator inputs are directly mapped to curvature and length parameters. However, care must be taken near zero curvature due to singularities, which can be resolved through improved parameterizations [62].

Discrete Models: These models represent soft robots using inherently discrete elements, without discretizing a continuous model. They are typically categorized into three types [8]:

1. **Lumped-Mass Models** use discrete masses, springs, and dampers to capture dynamics. They are flexible for modeling nonlinear effects but require many degrees of freedom (DOFs), making control and identification challenging.
2. **Pseudorigid Models** approximate soft bodies as serial chains of rigid links and joints. They leverage rigid-body theory but also demand high DOFs for fidelity.
3. **Discrete Rods** discretize a Cosserat rod into connected segments, capturing curvature through node positions and angles. While inspired by FEM, they are simpler and well-suited for simulation and control, especially in graphics and real-time applications.

Surrogate Models: These models represent a class of physics-agnostic approaches that rely on data-driven methods. In [8], they are classified into two categories: *neural networks* and *data-driven order reduction*. The first category employs neural networks to model the soft robot by learning its dynamics directly from data. The second approach uses data-driven methods to efficiently approximate the physical model. For instance, in [63], Koopman operator theory is used to enable the application of linear control methods to soft robots. This theory identifies a globally linear representation of the system, facilitating fast and efficient control design. Other methods include model order reduction of FEM models, where the finite element equations of motion are projected onto low-dimensional subspaces to achieve computational efficiency. Since this report focuses on physics-based modeling and control, these models will not be discussed in detail.

The classification scheme presented above provides a comprehensive framework to understand soft robot models from their theoretical foundations. This understanding offers valuable insights into their accuracy and computational feasibility. In [8], the uses of models from each category are summarized for simulation and control use cases. Some key insights include:

- Energetic approaches based on modal Ritz reduction, such as the **GVS approach** [59], provide highly reduced models in standard forms that are **well-suited for control**.
- Although **PCC approaches** are valid only under the circular arc assumption, they are ideal for **kinematic control**.
- Lumped mass and pseudorigid models are conceptually simple and allow the use of theory from rigid-body dynamics. However, they require high DOF to accurately model soft robots and need reduction techniques to be suitable for control.
- Although originally developed for fast interactive simulations, discrete rod approaches can be adapted for real-time simulation and control.
- Physics-agnostic approaches can enable fast simulation and control but are not generalizable and demand extensive data.

These insights motivate the modeling choice adopted in this thesis. While models such as PCC and polynomial curvature have been widely used in the literature due to their simplicity and computational efficiency, they rely on assumptions—such as constant curvature or planar deformation—that limit their generality and physical consistency. As noted in [8], both can be viewed as special cases of the more general GVS framework, obtained through specific choices of the strain basis. In light of this, the GVS model is presented as the unifying and physically consistent foundation for modeling slender soft robots in this work. It supports both theoretical rigor and practical applicability, making it well-suited for the control and optimization tasks addressed in the following chapters.

2-2-1 Configuration of a Soft Robot

Before proceeding to the models, it is worthwhile to establish how the configuration of a slender soft robot can be defined. One parameterization is already described in (2-1), which represents the configuration as a curve describing the full posture of the backbone of the soft

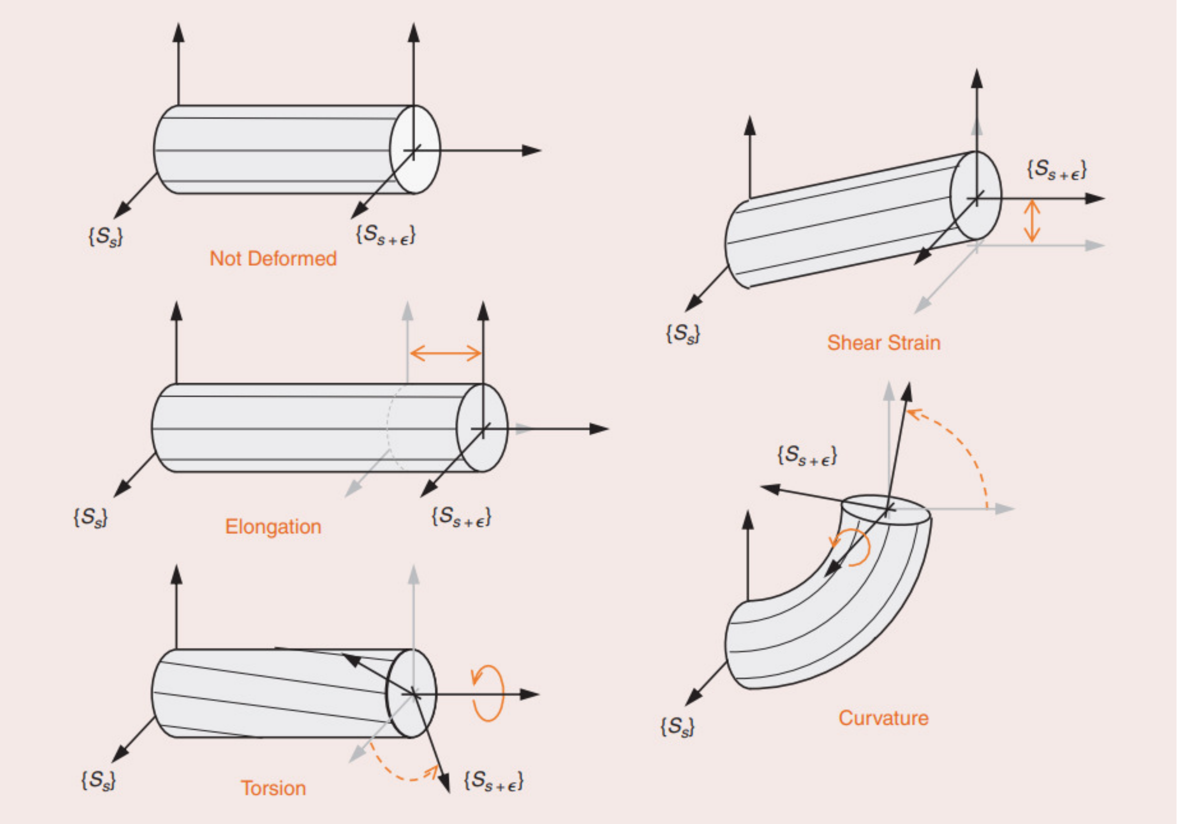


Figure 2-6: The six pure strains corresponding to $\xi(X) \in \mathbb{R}^6$. Image from [10].

robot at a given time. However, there is an alternative way to define the configuration of a soft robot, which will be used in the sections that follow.

As clarified in [10], local strains can effectively represent the configuration of a soft robot. These strains include curvature, twist, elongation, and shear. When considering the configuration of the soft robot as described in (2-1), these strains correspond to variations in g with respect to X. Consequently, the strain becomes a function of X, represented as $\xi(\cdot) : X \in [0, L] \mapsto \xi(X) \in \mathbb{R}^6$. The strains can be categorized into translational and rotational components. Translational strains include the elongation strain, which acts along the tangent to the backbone curve, and two shear strains, which act along axes orthogonal to the backbone curve. Rotational strains include the torsional strain about the backbone axis and two curvature or bending strains about axes perpendicular to the backbone.

2-2-2 Geometric Variable-Strain Approach

In this section, the GVS approach is discussed, which falls under the energetic methods within the director approaches class of continuum mechanics models. As mentioned earlier, this method relies on the Cosserat rod, a continuum mechanics object capable of modeling all six deformation modes of a slender body. The GVS approach is strain-based, where the strain field is parameterized along the length of the Cosserat rod. This methodology is introduced for both statics and dynamics in [64, 59, 65, 60]. The modeling process using the GVS approach is detailed in this section.

The configuration of the system, similar to (2-1), is given as:

$$\boldsymbol{g}(\cdot) : X \in [0, L] \to \boldsymbol{g}(X) = \begin{pmatrix} \boldsymbol{R} & \boldsymbol{r} \\ \boldsymbol{0} & 1 \end{pmatrix} \in SE(3), \tag{2-2}$$

where the transformation matrix $\boldsymbol{g}(X)$ maps a point in the body frame at X to the body frame at $X = 0$, implying that $\boldsymbol{g}(0) = \boldsymbol{I}_4$.

Strain State Representation. Considering the Lie algebra $\mathfrak{se}(3)$ of $SE(3)$, the spatial derivative of $\boldsymbol{g}$ can be expressed as:

$$\frac{\partial \boldsymbol{g}}{\partial X} = \boldsymbol{g}'(X) = \boldsymbol{g}(X)\hat{\boldsymbol{\xi}}, \tag{2-3}$$

where $\hat{\boldsymbol{\xi}}(\cdot) : X \to \mathfrak{se}(3)$ defines the strain state of the rod. This strain state is referenced against a strain configuration $\hat{\boldsymbol{\xi}}^*$. The components are written as:

$$\begin{aligned} \hat{\boldsymbol{\xi}}(X) &= \begin{pmatrix} \tilde{\boldsymbol{k}} & \boldsymbol{p} \\ \boldsymbol{0} & 0 \end{pmatrix} \in \mathfrak{se}(3), \\ \boldsymbol{\xi}(X) &= (\boldsymbol{k}^\top, \boldsymbol{p}^\top)^\top \in \mathbb{R}^6, \end{aligned} \tag{2-4}$$

where $\tilde{\boldsymbol{k}}(X) \in \mathfrak{so}(3)$, $\boldsymbol{k}(X) \in \mathbb{R}^3$, and $\boldsymbol{p}(X) \in \mathbb{R}^3$ represent angular and linear strains, respectively.

Temporal Derivative and Velocity Twist. The temporal derivative of $\boldsymbol{g}$ can similarly be written as:

$$\frac{\partial \boldsymbol{g}}{\partial t} = \dot{\boldsymbol{g}}(X) = \boldsymbol{g}(X)\hat{\boldsymbol{\eta}}, \tag{2-5}$$

where $\hat{\boldsymbol{\eta}}(\cdot) : X \to \mathfrak{se}(3)$ defines the velocity twist relative to $X = 0$ in the current body frame. This is expressed as:

$$\begin{aligned} \hat{\boldsymbol{\eta}}(X) &= \begin{pmatrix} \tilde{\boldsymbol{w}} & \boldsymbol{v} \\ \boldsymbol{0} & 0 \end{pmatrix} \in \mathfrak{se}(3), \\ \boldsymbol{\eta}(X) &= (\boldsymbol{w}^\top, \boldsymbol{v}^\top)^\top \in \mathbb{R}^6. \end{aligned} \tag{2-6}$$

Computing $\boldsymbol{g}(X)$ from $\boldsymbol{\xi}(X)$. Equation (2-3) is a matrix differential equation that can be integrated spatially to determine $\boldsymbol{g}(X)$. In the general case, this integration lacks a closed-form solution. However, for constant strain $\boldsymbol{\xi}(X) = \boldsymbol{\xi}$, the equation simplifies and can be straightforwardly integrated as shown in [64]. For the general case, the equation is of the form:

$$\boldsymbol{Y}' = \boldsymbol{Y}\boldsymbol{A}(X), \tag{2-7}$$

whose solution can be expressed using the Magnus expansion:

$$\boldsymbol{Y}(X) = \boldsymbol{Y}(0) \exp(\boldsymbol{\Omega}(X)), \tag{2-8}$$

where $\boldsymbol{\Omega}(X)$ is an integral series of Lie brackets of $\boldsymbol{\xi}(X)$ [66]. In [60], a fourth-order Zanna quadrature approximation of the Magnus expansion is employed. The domain $[0, L]$ is discretized into intervals of length h, and $\boldsymbol{g}(X)$ is evaluated recursively:

$$\boldsymbol{g}(X + h) = \boldsymbol{g}(X) \exp(\hat{\boldsymbol{\Omega}}(h)), \tag{2-9}$$

with the initial condition $\boldsymbol{g}(0) = \boldsymbol{I}$.

Strain Derivatives and Velocity Twist. The equality of mixed partial derivatives of $\boldsymbol{g}(X)$ yields:

$$\begin{aligned} \frac{\partial}{\partial X}\left(\frac{\partial \boldsymbol{g}}{\partial t}\right) &= \frac{\partial}{\partial t}\left(\frac{\partial \boldsymbol{g}}{\partial X}\right), \\ \boldsymbol{\eta}' &= \dot{\boldsymbol{\xi}} - \mathrm{ad}_{\boldsymbol{\xi}}\boldsymbol{\eta}, \\ \dot{\boldsymbol{\eta}}' &= \ddot{\boldsymbol{\xi}} - \mathrm{ad}_{\boldsymbol{\xi}}\dot{\boldsymbol{\xi}} - \mathrm{ad}_{\dot{\boldsymbol{\xi}}}\boldsymbol{\eta}. \end{aligned} \tag{2-10}$$

The velocity and acceleration twists can be computed as:

$$\begin{aligned} \boldsymbol{\eta}(X) &= \mathrm{Ad}_{\boldsymbol{g}}^{-1}\int_0^X \mathrm{Ad}_{\boldsymbol{g}}\dot{\boldsymbol{\xi}}\,ds, \\ \dot{\boldsymbol{\eta}}(X) &= \mathrm{Ad}_{\boldsymbol{g}}^{-1}\int_0^X \left(\dot{\mathrm{Ad}}_{\boldsymbol{g}}\dot{\boldsymbol{\xi}} + \mathrm{Ad}_{\boldsymbol{g}}\ddot{\boldsymbol{\xi}}\right) ds, \end{aligned} \tag{2-11}$$

where $\mathrm{ad}_{(\cdot)}$ and $\mathrm{Ad}_{(\cdot)}$ represent the adjoint operator of $\mathfrak{se}(3)$ and the adjoint map, respectively.

Strain Parameterization and Geometric Jacobian. The strain field is parameterized using generalized coordinates and basis functions:

$$\boldsymbol{\xi}(X) = \Phi_{\boldsymbol{\xi}}(X)\boldsymbol{q} + \boldsymbol{\xi}^*(X), \tag{2-12}$$

where $\boldsymbol{q} \in \mathbb{R}^n$ are the generalized coordinates, $\Phi_{\boldsymbol{\xi}}(X) \in \mathbb{R}^{6\times n}$ is a matrix of basis functions, and $\boldsymbol{\xi}^*(X)$ represents reference strains. Substituting these into (2-11), the geometric Jacobian can be obtained as:

$$\boldsymbol{\eta}(X, \boldsymbol{q}, \dot{\boldsymbol{q}}) = \underbrace{\mathrm{Ad}_{\boldsymbol{g}}^{-1}\int_0^X \mathrm{Ad}_{\boldsymbol{g}}\Phi_{\boldsymbol{\xi}} ds}_{\boldsymbol{J}(X,\boldsymbol{q})}\,\dot{\boldsymbol{q}}. \tag{2-13}$$

Using the Magnus expansion, $\boldsymbol{g}(X)$, $\boldsymbol{\eta}(X)$, and the geometric Jacobian can be recursively computed as shown earlier. The dynamics of the soft body can be obtained using D'Alembert's principle. The free-body dynamic equation in the local frame, along with its boundary condition, is given by the Cosserat equilibrium [60]:

$$\begin{aligned} &\boldsymbol{\mathcal{M}}\dot{\boldsymbol{\eta}} + \mathrm{ad}_{\boldsymbol{\eta}}^*\boldsymbol{\mathcal{M}}\boldsymbol{\eta} = \\ &\qquad (\boldsymbol{\mathcal{F}}_i - \boldsymbol{\mathcal{F}}_a)' + \mathrm{ad}_{\boldsymbol{\xi}}^*(\boldsymbol{\mathcal{F}}_i - \boldsymbol{\mathcal{F}}_a) + \boldsymbol{\mathcal{F}}_e, \\ &(\boldsymbol{\mathcal{F}}_i - \boldsymbol{\mathcal{F}}_a)(0) = -\boldsymbol{\mathcal{F}}_0, \\ &(\boldsymbol{\mathcal{F}}_i - \boldsymbol{\mathcal{F}}_a)(L) = \boldsymbol{\mathcal{F}}_L, \end{aligned} \tag{2-14}$$

where $\boldsymbol{\mathcal{M}}(X) \in \mathbb{R}^{6\times 6}$ is the screw inertia density matrix of the cross-section, given as:

$$\boldsymbol{\mathcal{M}}(X) = \rho\ \mathrm{diag}(J_x, J_y, J_z, A, A, A),$$

with ρ being the mass density, $A(X)$ the cross-sectional area, and J_x, J_y, J_z the second moments of area. The term $\boldsymbol{\mathcal{F}}_e(X) \in \mathbb{R}^6$ is the distributed external load, while $\boldsymbol{\mathcal{F}}_i(X), \boldsymbol{\mathcal{F}}_a(X) \in \mathbb{R}^6$ represent the internal wrenches due to material elasticity and distributed actuation, respectively.

Internal Elastic Wrench. The internal elastic wrench is derived using a Hooke-like linear elastic law:

$$\begin{aligned} \boldsymbol{\mathcal{F}}_i(X) &= \Sigma(\boldsymbol{\xi} - \boldsymbol{\xi}^*) + \Upsilon\dot{\boldsymbol{\xi}}, \\ &= \Sigma\Phi_{\boldsymbol{\xi}}\boldsymbol{q} + \Upsilon\Phi_{\boldsymbol{\xi}}\dot{\boldsymbol{q}}, \end{aligned} \tag{2-15}$$

where $\Sigma(X) \in \mathbb{R}^{6\times6}$ is the screw elasticity matrix:

$$\Sigma(X) = \text{diag}(GJ_x, EJ_y, EJ_z, EA, GA, GA),$$

with E being Young's modulus, G the shear modulus, and $\Upsilon(X) \in \mathbb{R}^{6\times6}$ the screw damping matrix:

$$\Upsilon(X) = \upsilon\ \text{diag}(J_x, 3J_y, 3J_z, 3A, A, A).$$

Distributed Actuation Wrench. The actuation can represent any distributed actuation wrench. In specific cases, it can be discretized using actuation bases (e.g., thread-like actuators such as tendons or embedded pneumatic chambers):

$$\boldsymbol{\mathcal{F}}_a(X) = \sum_{k=1}^{n_a} \begin{bmatrix} \tilde{\boldsymbol{d}}_k\boldsymbol{t}_k \\ \boldsymbol{t}_k \end{bmatrix} u_k = \Phi_a\boldsymbol{u}. \tag{2-16}$$

Generalized Dynamics of the Rod. The dynamics of the entire rod in generalized coordinates are obtained by integrating (2-14) over the length of the soft body and substituting the elastic, damping, and actuation laws. The resulting equations take the following form (in the case of independent bases):

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{K}\boldsymbol{q} + \boldsymbol{D}\dot{\boldsymbol{q}} = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{u}, \tag{2-17}$$

where the only external force considered here is gravity. The coefficients are computed as:

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{q}) &= \int_0^L \boldsymbol{J}^T\boldsymbol{\mathcal{M}}\boldsymbol{J}\,dX, \\ \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) &= \int_0^L \boldsymbol{J}^T(\text{ad}^*_{\boldsymbol{\eta}}\boldsymbol{\mathcal{M}}\boldsymbol{J} + \boldsymbol{\mathcal{M}}\dot{\boldsymbol{J}})\,dX, \\ \boldsymbol{G}(\boldsymbol{q}) &= \int_0^L \boldsymbol{J}^T\boldsymbol{\mathcal{M}}\left(\text{Ad}^{-1}_{\boldsymbol{g}^I}\boldsymbol{\mathcal{G}}\right)\,dX, \\ \boldsymbol{K} &= \int_0^L \Phi_{\boldsymbol{\xi}}^T\Sigma\Phi_{\boldsymbol{\xi}}\,dX, \\ \boldsymbol{D} &= \int_0^L \Phi_{\boldsymbol{\xi}}^T\Upsilon\Phi_{\boldsymbol{\xi}}\,dX, \\ \boldsymbol{B}(\boldsymbol{q}) &= \int_0^L \Phi_{\boldsymbol{\xi}}^T\Phi_a\,dX, \end{aligned} \tag{2-18}$$

where $\boldsymbol{\mathcal{G}} \in \mathbb{R}^6$ is the gravitational acceleration twist expressed in the inertial frame. Other terms are defined as previously described. The spatial integrals in equation (3-5) can be evaluated using numerical integration schemes, such as Gaussian-Legendre quadrature. This method evaluates the integrands at discrete points, providing an accurate numerical approximation of the integrals. To derive the generalized equations, a two-level nested quadrature scheme is employed, as outlined in [60]. Specifically, Zannah collocation is used to compute the approximate Magnus expansion of the strain twist, while the numerical scheme addresses the integration in equation (3-5).

The previously introduced GVS formulation is not limited to individual soft elements; it naturally extends to more complex morphologies, including serially connected soft links and hybrid soft–rigid structures, as outlined in [60]. A particularly appealing feature of the GVS framework is its generality: depending on the choice of basis functions used to parametrize the strain field, it can reproduce a variety of classical modeling approaches. These include the widely used Piecewise-Constant Curvature (PCC) model, the Piecewise-Constant Strain (PCS) model, and the Polynomial Curvature model.

Piecewise-Constant Curvature Model. The PCC model [62] can be recovered as a special case of the GVS formulation by selecting a spatially invariant strain basis. For instance, the following parametrization yields the classical planar PCC model with bending about the y- and z-axes and an axial elongation term:

$$\boldsymbol{\xi}(X, \boldsymbol{q}) = \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}}_{\boldsymbol{\Phi}_\xi} \underbrace{\begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix}}_{\boldsymbol{q}} + \underbrace{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}}_{\boldsymbol{\xi}^*} \tag{2-19}$$

This configuration assumes constant bending and axial strain over the entire segment, aligning with the assumptions of the PCC model. More generally, including additional nonzero rows in $\boldsymbol{\Phi}_\xi$ allows the model to capture shear and torsion, leading to the Piecewise-Constant Strain (PCS) formulation. These constant-strain representations benefit from analytical expressions for forward kinematics and do not require numerical integration techniques such as the Magnus expansion. However, they are only accurate under restrictive assumptions and are less general than the full GVS formulation.

Polynomial Curvature Model. Similarly, the Polynomial Curvature model proposed in [67] and analyzed in [68] can also be expressed within the GVS framework by using a monomial basis in the spatial variable X:

$$\boldsymbol{\xi}(X, \boldsymbol{q}) = \underbrace{\begin{pmatrix} 0 & 0 & \cdots & 0 \\ 1 & X & \cdots & X^{n-1} \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \end{pmatrix}}_{\boldsymbol{\Phi}_\xi(X)} \underbrace{\begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{pmatrix}}_{\boldsymbol{q}} + \underbrace{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}}_{\boldsymbol{\xi}^*} \tag{2-20}$$

This basis captures planar bending about the y-axis, with the curvature defined as a polynomial function of X. In the special case where $n = 2$, the curvature becomes affine, as used in the Soft Inverted Pendulum model [13], allowing closed-form expressions for the shape via

Fresnel integrals. However, these expressions are still numerically intensive and often require approximation in practice.

In summary, while models like PCC, PCS, and Polynomial Curvature offer useful closed-form expressions and simplified dynamics, they are all recoverable as specific instances of the GVS model with tailored strain basis functions. The GVS framework thus provides a unified and extensible foundation for modeling a wide variety of soft robotic systems while being well suited for model-based control.

2-3 Model-Based Control of Soft Robots

This section introduces the control challenges associated with continuum soft robots. As highlighted in [10], while soft robotic platforms have seen significant advances in mechanical design and robustness, their widespread deployment is still constrained by the difficulty of achieving precise, reliable control. Unlike rigid robots, where control theory has evolved in parallel with modeling capabilities, soft robotics initially faced a bottleneck in modeling complexity, leading to the widespread adoption of machine learning and model-free strategies.

This viewpoint has shifted in recent years due to two key developments. First, empirical and theoretical results have shown that feedback control can tolerate simplified or approximate models, often outperforming model-free methods. Second, as discussed in the previous section on the modeling, a number of reduced-order models have emerged—such as PCC, polynomial curvature, and GVS—that offer tractable and interpretable representations of soft robot dynamics. These developments have laid the groundwork for reintroducing model-based control methods into soft robotics, including geometric and optimal control.

Based on the unified perspective in [10], the control problem can now be formulated systematically. Assuming that the strain field $\boldsymbol{\xi}(X)$ is parameterized by a finite set of generalized coordinates $\boldsymbol{q} \in \mathbb{R}^n$, the governing dynamics of the soft robot can be written as:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{K}(\boldsymbol{q}) + \boldsymbol{D}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{A}(\boldsymbol{q})\boldsymbol{\tau}, \tag{2-21}$$

where:

- $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n \times n}$ is the inertia matrix,
- $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$ represents Coriolis and centrifugal effects,
- $\boldsymbol{G}(\boldsymbol{q})$ and $\boldsymbol{K}(\boldsymbol{q})$ denote gravitational and elastic forces,
- $\boldsymbol{D}(\boldsymbol{q})\dot{\boldsymbol{q}}$ accounts for damping,
- $\boldsymbol{A}(\boldsymbol{q}) \in \mathbb{R}^{n \times m}$ is the actuation matrix, and $\boldsymbol{\tau} \in \mathbb{R}^m$ is the control input.

The remainder of this section examines control strategies formulated around this dynamic structure. Beginning with fully actuated approximations and posture regulation, the discussion then addresses challenges in underactuation and a review of current methods and opportunities where optimal control techniques may provide practical and scalable solutions.

2-3-1 Control Problem

The primary control objective in soft robotics is to regulate the shape of the system. This involves controlling the strains, which are a function of the configuration vector $\boldsymbol{q}(t)$. The goal of posture regulation is to stabilize the configuration $\boldsymbol{q}(t)$ to a desired target $\bar{\boldsymbol{q}} \in \mathbb{R}^n$ using the control input $\boldsymbol{\tau} \in \mathbb{R}^m$, such that:

$$\lim_{t\to\infty} \boldsymbol{q}(t) = \bar{\boldsymbol{q}}. \tag{2-22}$$

The desired configuration is chosen to be an equilibrium configuration of the system (stable or unstable). The equilibrium configurations of the system in (2-21) for a given control input $\boldsymbol{\tau}_{eq}$ are the solutions to:

$$\boldsymbol{K}(\boldsymbol{q}) + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{A}(\boldsymbol{q})\boldsymbol{\tau}_{eq}. \tag{2-23}$$

As noted in [10], at least one solution exists for any given $\boldsymbol{\tau}_{eq}$ if $\boldsymbol{K}(\boldsymbol{q})$ is radially unbounded. This existence of at least one equilibrium for constant actuation is a notable departure from rigid robots, where constant actuation cannot result in equilibrium configurations unless gravity is a factor.

2-3-2 Fully actuated approximation

Following [10], control design can begin with a simplified case where the soft robot is fully actuated, i.e., $\boldsymbol{A}(\boldsymbol{q}) = \boldsymbol{I}$. In this case, a feedforward input that balances internal elastic and gravitational forces at a desired equilibrium $\bar{\boldsymbol{q}}$ can be written as:

$$\boldsymbol{\tau}(\bar{q}) = \boldsymbol{K}(\bar{q}) + \boldsymbol{G}(\bar{q}), \tag{2-24}$$

where $\boldsymbol{K}(\bar{q})$ and $\boldsymbol{G}(\bar{q})$ denote elastic and gravitational forces, respectively. This controller leverages soft robots' intrinsic dynamics and can stabilize configurations even with minimal feedback.

Theorem 1. [10] Let $(\bar{\boldsymbol{q}}, \boldsymbol{0})$ be an equilibrium of the system in (2-21) under (2-24). It is asymptotically stable if there exists a neighborhood $\mathcal{N}(\bar{\boldsymbol{q}})$ such that:

i. $U_K(\boldsymbol{q}) + U_G(\boldsymbol{q}) > U_K(\bar{\boldsymbol{q}}) + U_G(\bar{\boldsymbol{q}}) + (\boldsymbol{K}(\bar{\boldsymbol{q}}) + \boldsymbol{G}(\bar{\boldsymbol{q}}))^\top(\boldsymbol{q} - \bar{\boldsymbol{q}})$,

ii. $\boldsymbol{K}(\boldsymbol{q}) + \boldsymbol{G}(\boldsymbol{q}) \neq \boldsymbol{K}(\bar{\boldsymbol{q}}) + \boldsymbol{G}(\bar{\boldsymbol{q}})$ for $\boldsymbol{q} \in \mathcal{N}(\bar{\boldsymbol{q}}) \setminus \{\bar{\boldsymbol{q}}\}$.

These conditions reflect the *self-stabilizing* nature of soft robots due to their elastic fields. Condition (ii) can be interpreted as a convexity check via:

$$\left.\left(\frac{\partial \boldsymbol{K}}{\partial \boldsymbol{q}} + \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{q}}\right)\right|_{\boldsymbol{q}=\bar{\boldsymbol{q}}} \succ 0. \tag{2-25}$$

The feedforward law can be improved using feedback:

$$\boldsymbol{\tau}(\bar{\boldsymbol{q}}, \boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{K}(\bar{\boldsymbol{q}}) + \boldsymbol{G}(\bar{\boldsymbol{q}}) + \alpha(\bar{\boldsymbol{q}} - \boldsymbol{q}) - \beta\dot{\boldsymbol{q}}, \tag{2-26}$$

with gain matrices $\alpha, \beta \in \mathbb{R}^{n \times n}$. This introduces damping and enlarges the basin of attraction. For stability, it suffices that $\beta \succeq 0$ and:

$$\left(\frac{\partial \boldsymbol{K}}{\partial \boldsymbol{q}} + \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{q}} \right) \Bigg|_{\boldsymbol{q}=\bar{\boldsymbol{q}}} + \alpha \succ 0. \tag{2-27}$$

This control formulation demonstrates how soft robots' elastic properties reduce the complexity of the control problem and offer opportunities for stable, low-effort regulation.

2-3-3 Dealing with Underactuation

The system considered in the previous sections was based on fully actuated approximation ($\boldsymbol{A}(\boldsymbol{q}) = \boldsymbol{I}$). This works well in practice, but considering underactuation can be key to increasing the performance and reliability of controllers for soft robots, while even avoiding performance degradation and instability. The primary difficulty with underactuation lies in the limitations on the attainable equilibrium. For a nonsquare actuation matrix $\boldsymbol{A}(\boldsymbol{q}) \in \mathbb{R}^{n \times m}$ with $m < n$, for a given configuration $\bar{\boldsymbol{q}}$, there need not exist a $\boldsymbol{\tau}_{eq}$ for which the equilibrium equation (2-23) is satisfied because $\boldsymbol{K}(\bar{\boldsymbol{q}}) + \boldsymbol{G}(\bar{\boldsymbol{q}}) \notin \text{Span}(\boldsymbol{A}(\bar{\boldsymbol{q}}))$.

This also means that a time-varying torque may not exist such that an arbitrary state $(\boldsymbol{q}, \dot{\boldsymbol{q}})$ can be reached from any initial condition. However, for an attainable equilibrium, the equation can be solved using the left pseudo-inverse of $\boldsymbol{A}(\boldsymbol{q})$ as follows:

$$\boldsymbol{\tau} = (\boldsymbol{A}^T \boldsymbol{A})^{-1} \boldsymbol{A}^T (\boldsymbol{K}(\bar{\boldsymbol{q}}) + \boldsymbol{G}(\bar{\boldsymbol{q}})). \tag{2-28}$$

For a configuration-independent actuation matrix, this can lead to the same closed-loop behavior as with (2-24), and it can be proved using Theorem 1 if

$$(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^L)(\boldsymbol{K}(\boldsymbol{q}) + \boldsymbol{G}(\boldsymbol{q})) = 0, \tag{2-29}$$

which makes the closed-loop system equivalent to the closed loop with the fully actuated approximation, from which the proof follows.

The controller, similar to (2-26), can also be extended with $\alpha = \boldsymbol{A}\alpha\boldsymbol{A}^T$ and $\beta = \boldsymbol{A}\beta\boldsymbol{A}^T$. The proof follows directly from the fully actuated case as shown earlier. The conditions mentioned are specified under Corollary 2 in [10].

The modified condition (i) from Theorem 1 with the feedback controller (2-27) in this case can be written as

$$\left(\frac{\partial \boldsymbol{K}(\boldsymbol{q})}{\partial \boldsymbol{q}} + \frac{\partial \boldsymbol{G}(\boldsymbol{q})}{\partial \boldsymbol{q}} + \boldsymbol{A}\alpha\boldsymbol{A}^T \right) \Bigg|_{\boldsymbol{q}=\bar{\boldsymbol{q}}} \succ 0. \tag{2-30}$$

Where $\boldsymbol{A}\alpha\boldsymbol{A}^T \succeq 0$ if $\alpha \succ 0$, but $\text{Rank}(\boldsymbol{A}\alpha\boldsymbol{A}^T) \leq m$. This means that the condition is satisfied and $\bar{\boldsymbol{q}}$ is stabilized only when actuation is colocated on the directions in which the effective stiffness loses rank. As mentioned in [10], this can be interpreted as the controller stabilizing the actuated coordinates $\boldsymbol{A}^T\boldsymbol{q}$ while relying on the elasticity ($\boldsymbol{K}(\boldsymbol{q})$) to stabilize the unactuated coordinates. This interplay between actuation and inherent elasticity is a crucial factor that must be carefully considered when designing controllers for the underactuated case.

However, addressing underactuation in the general case is highly challenging. The literature on model-based controllers for underactuated systems primarily focuses on designing controllers for specific classes of underactuated systems where provably stable controllers can be formulated. The current body of work on model-based controllers using collocated feedback for soft robots can be broadly categorized into two main cases: the *Elastically Dominated Case* and the *Elastically Decoupled Case*.

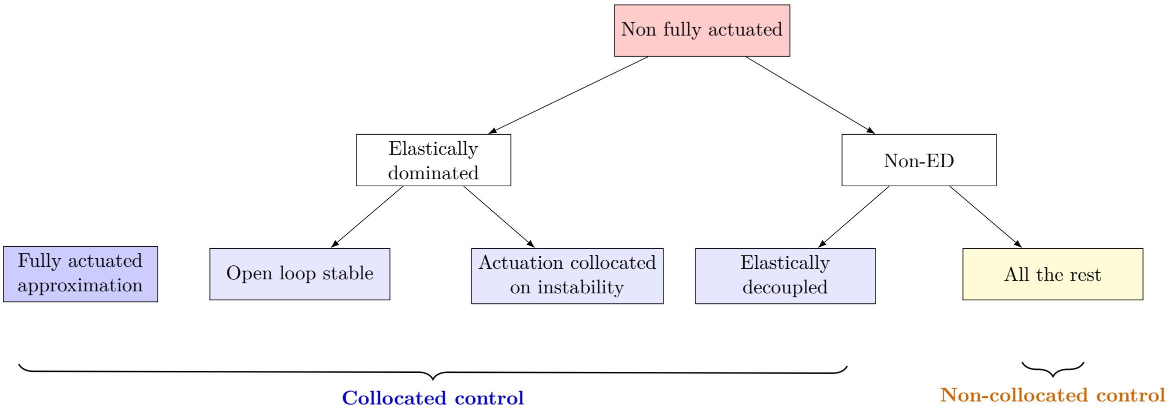


Figure 2-7: Taxonomy of soft robot control strategies, based on works in [10, 11, 12, 13]. Collocated control includes fully actuated and favorable underactuated cases, while non-collocated control encompasses more complex scenarios with increased control difficulty.

Both of these cases represent reasonably general classes of systems. The assumptions and control design methodologies for each of these cases are discussed as follows.

Elastically Dominated Case. In [11] a broad class of regulators are proposed to stabilize soft robots with dominant elasticity via collocated feedback along with simple verifiable conditions. In [11] the system dynamics is expressed in the Hamiltonian form as follows:

$$\begin{bmatrix} \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{p}} \end{bmatrix} = \begin{bmatrix} 0 & \boldsymbol{I}_n \\ -\boldsymbol{I}_n & -\boldsymbol{D} \end{bmatrix} \begin{bmatrix} \frac{\partial \boldsymbol{H}(\boldsymbol{q},\boldsymbol{p})}{\partial \boldsymbol{q}} \\ \frac{\partial \boldsymbol{H}(\boldsymbol{q},\boldsymbol{p})}{\partial \boldsymbol{p}} \end{bmatrix} + \begin{bmatrix} 0 \\ \boldsymbol{A} \end{bmatrix} \boldsymbol{\tau} \tag{2-31}$$

Where $\boldsymbol{q} \in \mathbb{R}^n$ are the generalized coordinates, and $\boldsymbol{p} \in \mathbb{R}^n$ are the generalized momenta as defined in (2-21). The Hamiltonian $\boldsymbol{H}(\boldsymbol{q},\boldsymbol{p}) = \boldsymbol{T}(\boldsymbol{q},\boldsymbol{p}) + \boldsymbol{V}(\boldsymbol{q})$ is the total energy, where the potential energy $\boldsymbol{V}(\boldsymbol{q})$ depends only on the generalized coordinates, and the kinetic energy $\boldsymbol{T}(\boldsymbol{q},\boldsymbol{p}) = \frac{1}{2}\boldsymbol{p}^T\boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{p}$ depends on both $\boldsymbol{q}$ and $\boldsymbol{p}$. The damping matrix $\boldsymbol{D}$, input mapping matrix $\boldsymbol{A}$, and other terms are as described in (2-21), with $\boldsymbol{A} \in \mathbb{R}^{n\times m}$, where $m < n$ in the underactuated case. The first term on the right-hand side describes the intrinsic dynamics of the system, while the second term captures the influence of external inputs $\boldsymbol{\tau}$ via $\boldsymbol{A}$.

This work assumes a constant actuation matrix $\boldsymbol{A}(\boldsymbol{q})$ with full column rank, ensuring that a linear transformation exists to decouple the input. This allows a change of coordinates that splits the generalized coordinates $\boldsymbol{q}$ into actuated and unactuated components (see Proposition 1 in [11]). However, this approach has been generalized to the case of configuration-dependent actuation matrices for which such a transformation still exists, as discussed in [69]. In this section, the dynamics in (2-31) is considered to already be input-decoupled, implying that $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{I}_m & \boldsymbol{0} \end{bmatrix}^T$ and $\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_a & \boldsymbol{q}_u \end{bmatrix}^T$, where $\boldsymbol{q}_a \in \mathbb{R}^m$ are the actuated coordinates and $\boldsymbol{q}_u \in \mathbb{R}^{n-m}$ are the unactuated coordinates.

Before stating the control law, the following two assumptions should be discussed.

Assumption 1: For a given desired equilibrium of the system $\bar{\boldsymbol{q}}$ (as in (2-23)), the elastic potential energy must satisfy:

$$\boldsymbol{K} \preccurlyeq \left(\frac{\partial^2 V_e}{\partial \boldsymbol{q}^2}\right)\bigg|_{\boldsymbol{q}=\bar{\boldsymbol{q}}}, \tag{2-32}$$

where $\boldsymbol{K} \in \mathbb{R}^{n\times n}$ is a positive semi-definite matrix. This assumption holds for a large class of soft robots and is typically satisfied in cases where the elasticity is linear, leading to a potential energy of the form $V_e = \frac{1}{2}\boldsymbol{q}^T\bar{\boldsymbol{K}}\boldsymbol{q}$, with $\bar{\boldsymbol{K}}$ being positive semi-definite. Additionally, $\bar{\boldsymbol{K}}$ can be written as:

$$\bar{\boldsymbol{K}} = \begin{bmatrix} \boldsymbol{K}_{aa} & \boldsymbol{K}_{au} \\ \boldsymbol{K}_{au}^T & \boldsymbol{K}_{uu} \end{bmatrix}, \tag{2-33}$$

where the diagonal matrices $\boldsymbol{K}_{aa} \in \mathbb{R}^{m\times m}$ and $\boldsymbol{K}_{uu} \in \mathbb{R}^{(n-m)\times(n-m)}$ are also positive semi-definite. This structure will become important for the proof and the next assumption.

Assumption 2: Elasticity must dominate the forces resulting from gravity in the unactuated coordinates:

$$\boldsymbol{K}_{uu} + \left(\frac{\partial^2 V_g}{\partial \boldsymbol{q}_u^2}\right)\bigg|_{\boldsymbol{q}=\bar{\boldsymbol{q}}} \succ 0. \tag{2-34}$$

This assumption is equivalent to assuming that the zero dynamics of (2-31) is asymptotically stable, considering $\boldsymbol{y} = \boldsymbol{q}_a$ as the output. Minimum-phase systems will be discussed further in the following sections. This assumption reinforces the idea that elasticity is relied upon to stabilize the unactuated coordinates.

In [11] the control design is expressed in Lemma 1 which is stated as follows:

Lemma 1. For the desired configuration $\bar{\boldsymbol{q}}$ of (2-31) satisfying Assumptions 1 and 2, let $\Phi : \mathbb{R}^m \to \mathbb{R}$ be a C^2 function such that:

$$\begin{aligned} &\left(\frac{\partial V}{\partial \boldsymbol{q}_a} + \frac{\partial \Phi}{\partial \boldsymbol{q}_a}\right)\bigg|_{\boldsymbol{q}=\bar{\boldsymbol{q}}} = 0, \\ &\left(\boldsymbol{H}_{ua}'^{\top}\left(\boldsymbol{H}_{uu}'\right)^{-1}\boldsymbol{H}_{ua}' - \boldsymbol{H}_{aa}'\right) \prec \left(\frac{\partial^2 \Phi}{\partial \boldsymbol{q}_a^2}\right)\bigg|_{\boldsymbol{q}=\bar{\boldsymbol{q}}}, \end{aligned} \tag{2-35}$$

where

$$\boldsymbol{H}_{aa}' := \left(\frac{\partial^2 V_g}{\partial \boldsymbol{q}_a^2}\right)\bigg|_{\boldsymbol{q}=\bar{\boldsymbol{q}}} + \boldsymbol{K}_{aa}, \quad \boldsymbol{H}_{ua}' := \left(\frac{\partial^2 V_g}{\partial \boldsymbol{q}_u \partial \boldsymbol{q}_a}\right)\bigg|_{\boldsymbol{q}=\bar{\boldsymbol{q}}} + \boldsymbol{K}_{au}^{\top}, \quad \boldsymbol{H}_{uu}' := \left(\frac{\partial^2 V_g}{\partial \boldsymbol{q}_u^2}\right)\bigg|_{\boldsymbol{q}=\bar{\boldsymbol{q}}} + \boldsymbol{K}_{uu}.$$

Hence, the controller

$$\boldsymbol{\tau} = -\frac{\partial \Phi(\boldsymbol{q}_a)}{\partial \boldsymbol{q}_a}, \tag{2-36}$$

(locally) stabilizes the system at $(\bar{\boldsymbol{q}}, \boldsymbol{0})$.

Many controllers can be designed based on Lemma 1. The paper introduces 4 choices. Thus, for the function $\varphi : \mathbb{R}^m \to \mathbb{R}$ such that $\frac{\partial \varphi(\boldsymbol{q}_a)}{\partial \boldsymbol{q}_a} = \left(\frac{\partial V}{\partial \boldsymbol{q}_a}\right)\Big|_{\boldsymbol{q}_u=\bar{\boldsymbol{q}}_u}$, $\kappa := \left(\frac{\partial V}{\partial \boldsymbol{q}_a}\right)\Big|_{\boldsymbol{q}=\bar{\boldsymbol{q}}}$, $\tilde{\boldsymbol{q}}_a := \boldsymbol{q}_a - \bar{\boldsymbol{q}}_a$, and gains $K_p, K_d, \alpha, \rho, \alpha_d, \rho_d$, we have:

(i) $u_{PD} = \kappa - K_p\tilde{\boldsymbol{q}}_a - K_D\dot{\boldsymbol{q}}_a$

(ii) $u_{PD_+} = \frac{\partial\varphi(\boldsymbol{q}_a)}{\partial\boldsymbol{q}_a} - K_p\tilde{\boldsymbol{q}}_a - K_D\dot{\boldsymbol{q}}_a$

(iii) $u_{sat} = \kappa - \alpha\tanh(\rho\tilde{\boldsymbol{q}}_a) - \alpha_d\tanh(\rho_d\dot{\boldsymbol{q}}_a)$

(iv) $u_{exp} = \frac{\partial\varphi(\boldsymbol{q}_a)}{\partial\boldsymbol{q}_a} - K_p\left(e^{\tilde{\boldsymbol{q}}_a} - e^{-\tilde{\boldsymbol{q}}_a}\right) - K_D\dot{\boldsymbol{q}}_a$

The limitations of the controller arise primarily from Assumption 2, which is a direct consequence of the system's underactuation. If this assumption is not satisfied, the controller can assign the desired equilibrium but may fail to stabilize the system.

Elastically Decoupled Case. In [12], a reasonably large class of underactuated soft robots is considered, referred to as *elastically decoupled* systems. These are characterized by the absence of direct elastic coupling between the actuated and unactuated variables. The considered class of systems is modeled by ODEs as in (2-21). However, similar to the previous case, this work assumes a constant actuation matrix $\boldsymbol{A}(\boldsymbol{q})$ with full column rank. This assumption allows for the existence of a linear change of coordinates such that the dynamic equations can be separated into actuated and unactuated variables (see Section II-A in [12]).

As previously mentioned, this approach can be generalized to the case of non-constant, configuration-dependent actuation matrices, as explored in [69] under necessary and sufficient conditions. Thus, in this section, the dynamics in (2-21) is considered to already be input-decoupled, implying that $\boldsymbol{A} = \begin{bmatrix}\boldsymbol{I}_m & \boldsymbol{0}\end{bmatrix}^T$ and $\boldsymbol{q} = \begin{bmatrix}\boldsymbol{q}_a & \boldsymbol{q}_u\end{bmatrix}^T$, where $\boldsymbol{q}_a \in \mathbb{R}^m$ are the actuated coordinates and $\boldsymbol{q}_u \in \mathbb{R}^{n-m}$ are the unactuated coordinates. After the coordinate transformation, the system dynamics can be rewritten as:

$$\underbrace{\begin{pmatrix}\boldsymbol{M}_{aa} & \boldsymbol{M}_{au}\\ \boldsymbol{M}_{ua} & \boldsymbol{M}_{uu}\end{pmatrix}}_{\boldsymbol{M}(\boldsymbol{q})}\begin{pmatrix}\ddot{\boldsymbol{q}}_a\\ \ddot{\boldsymbol{q}}_u\end{pmatrix} + \underbrace{\begin{pmatrix}\boldsymbol{C}_{aa} & \boldsymbol{C}_{au}\\ \boldsymbol{C}_{ua} & \boldsymbol{C}_{uu}\end{pmatrix}}_{\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})}\begin{pmatrix}\dot{\boldsymbol{q}}_a\\ \dot{\boldsymbol{q}}_u\end{pmatrix} + \underbrace{\begin{pmatrix}\boldsymbol{G}_a\\ \boldsymbol{G}_u\end{pmatrix}}_{\boldsymbol{G}(\boldsymbol{q})} + \underbrace{\begin{pmatrix}\boldsymbol{K}_{aa} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{K}_{uu}\end{pmatrix}}_{\boldsymbol{K}}\begin{pmatrix}\boldsymbol{q}_a\\ \boldsymbol{q}_u\end{pmatrix}$$
$$+ \underbrace{\begin{pmatrix}\boldsymbol{D}_{aa} & \boldsymbol{D}_{au}\\ \boldsymbol{D}_{ua} & \boldsymbol{D}_{uu}\end{pmatrix}}_{\boldsymbol{D}(\boldsymbol{q})}\begin{pmatrix}\dot{\boldsymbol{q}}_a\\ \dot{\boldsymbol{q}}_u\end{pmatrix} = \begin{pmatrix}\boldsymbol{\tau}\\ \boldsymbol{0}\end{pmatrix}. \tag{2-37}$$

The terms in the equation are defined as before. Notably, the stiffness term is linear, ensuring that the stiffness matrix takes an elastically decoupled form. Under a coordinate transformation, the transformed stiffness matrix should also retain this elastically decoupled structure.

In [12], the control design begins by analyzing the stability of the zero dynamics associated with the controlled output:$\boldsymbol{y} = \boldsymbol{q}_a - \boldsymbol{q}_{a,d}$ where $\boldsymbol{q}_{a,d}$ is a constant reference. This is referred to as the *collocated case*. The zero dynamics of a system represent the residual dynamics that remain when the controlled output is constrained to be zero at all times. A nonlinear control system is said to be *minimum phase* if the trajectories of its zero dynamics are bounded.

The zero dynamics of (2-37) can be obtained by setting $\boldsymbol{y}$, $\dot{\boldsymbol{y}}$, and $\ddot{\boldsymbol{y}}$ to zero, which implies $\boldsymbol{q}_a = \boldsymbol{q}_{a,d}$, $\dot{\boldsymbol{q}}_a = 0$, and $\ddot{\boldsymbol{q}}_a = 0$. Substituting these conditions into (2-37) yields:

$$\boldsymbol{M}_{uu}(\boldsymbol{q}_{a,d}, \boldsymbol{q}_u)\ddot{\boldsymbol{q}}_u + \boldsymbol{C}_{uu}(\boldsymbol{q}_{a,d}, \boldsymbol{q}_u, 0, \dot{\boldsymbol{q}}_u)\dot{\boldsymbol{q}}_u + \boldsymbol{G}(\boldsymbol{q}_{a,d}, \boldsymbol{q}_u) + \boldsymbol{K}_{uu}\boldsymbol{q}_u + \boldsymbol{D}_{uu}(\boldsymbol{q}_{a,d}, \boldsymbol{q}_u)\dot{\boldsymbol{q}}_u = 0. \tag{2-38}$$

The paper demonstrates that the above system is minimum phase. The intuition behind this result is that the damping term dissipates the energy stored in the unactuated subsystem. It is shown that for any initial state, the trajectories of (2-38) are bounded and converge to the equilibrium:

$$\boldsymbol{K}_{uu}\boldsymbol{q}_u + \boldsymbol{G}_u(\boldsymbol{q}_{a,d}, \boldsymbol{q}_u) = \boldsymbol{0}. \tag{2-39}$$

This result is proved using LaSalle's invariance principle by considering the Lyapunov candidate function:

$$V(\boldsymbol{q}_u, \dot{\boldsymbol{q}}_u) = \frac{1}{2}\dot{\boldsymbol{q}}_u^T \boldsymbol{M}_{uu}\dot{\boldsymbol{q}}_u + \frac{1}{2}\boldsymbol{q}_u^T \boldsymbol{K}_{uu}\boldsymbol{q}_u + \boldsymbol{U}(\boldsymbol{q}_{a,d}, \boldsymbol{q}_u). \tag{2-40}$$

Since the potential function is lower-bounded the function V is lower-bounded as well. The derivative $\dot{V}$ is shown to be negative semi-definite using the dynamics in (2-38) and the skew-symmetric property of $\dot{\boldsymbol{M}}_{uu} - 2\boldsymbol{C}_{uu}$. According to LaSalle's theorem, the trajectories will converge to the largest invariant set where $\dot{V} = 0$, which defines the set of equilibrium points $(\boldsymbol{q}_u, \boldsymbol{0})$, satisfying (2-39).

For this equilibrium to be unique, the condition:$\boldsymbol{K}_{uu} \succ -\frac{\partial^2 \boldsymbol{U}(\boldsymbol{q}_{a,d},\boldsymbol{q}_u)}{\partial \boldsymbol{q}_u^2}$ must hold. This is shown by considering a convex function: $P(\boldsymbol{q}_u) = \boldsymbol{U}(\boldsymbol{q}_{a,d}, \boldsymbol{q}_u) + \frac{1}{2}\boldsymbol{q}_u^T \boldsymbol{K}_{uu}\boldsymbol{q}_u$ whose gradient corresponds to (2-39). The given condition ensures that the Hessian is positive definite, making $(\boldsymbol{q}_{u,eq}, \boldsymbol{0})$ a unique minimum. This assumption on the stability of the zero dynamics is equivalent to the assumption 2 used in the *elastically dominated* case. This highlights how collocated control relies on the passive dynamics being stable because of the soft robot being stiff enough.

The paper proposes a PD regulator similar to the one discussed previously for the underactuated case in the beginning of the section. The controller takes the form:

$$\boldsymbol{\tau} = \boldsymbol{K}_P(\boldsymbol{q}_{a,d} - \boldsymbol{q}_a) - \boldsymbol{K}_D\dot{\boldsymbol{q}}_a + \boldsymbol{G}_a(\boldsymbol{q}) + \boldsymbol{K}_{aa}\boldsymbol{q}_{a,d} \tag{2-41}$$

Soft Inverted Pendulum

In [13], a novel underactuated template system called the soft inverted pendulum (SIP) with affine curvature is introduced. The SIP is modeled here based on the polynomial curvature approach from [67]. The purpose of this model, as outlined in [13], is to mirror the trajectory taken by the control community in understanding the control challenges of underactuated rigid robots, such as the cart-pole, pendubot, and acrobot. The SIP system is analogous to the acrobot, featuring two degrees of freedom with the control input applied at the farthest point from the base. This makes the SIP an important case study when investigating the initial stages of the research problem, as focusing on a small yet meaningful system allows for testing the proposed control strategies before extending them to higher-dimensional systems. Moreover, the SIP provides a controlled environment for isolating fundamental control challenges.

The SIP consists of a soft segment of length L and thickness D, undergoing only planar deformations and thus assumed to be inextensible. Following the modeling approach in [67], the kinematics of the SIP are derived, specialized for two degrees of freedom. The curvature can be written as

$$\alpha(v, t) = q_1(t) + q_2(t)\frac{v^2}{2}, \tag{2-42}$$

The resulting integrals are evaluated by expanding the arguments inside the sine and cosine functions, yielding

$$\begin{aligned} x(v,t) =& \sin\left(\frac{q_1^2}{2q_2}\right)\sqrt{\frac{\pi}{q_2}}\left[C\left(\frac{q_1+vq_2}{\sqrt{\pi q_2}}\right) - C\left(\frac{q_1}{\sqrt{\pi q_2}}\right)\right] \\ & - \cos\left(\frac{q_1^2}{2q_2}\right)\sqrt{\frac{\pi}{q_2}}\left[S\left(\frac{q_1+vq_2}{\sqrt{\pi q_2}}\right) - S\left(\frac{q_1}{\sqrt{\pi q_2}}\right)\right], \\ y(v,t) =& \cos\left(\frac{q_1^2}{2q_2}\right)\sqrt{\frac{\pi}{q_2}}\left[C\left(\frac{q_1+vq_2}{\sqrt{\pi q_2}}\right) - C\left(\frac{q_1}{\sqrt{\pi q_2}}\right)\right] \\ & + \sin\left(\frac{q_1^2}{2q_2}\right)\sqrt{\frac{\pi}{q_2}}\left[S\left(\frac{q_1+vq_2}{\sqrt{\pi q_2}}\right) - S\left(\frac{q_1}{\sqrt{\pi q_2}}\right)\right], \end{aligned} \tag{2-43}$$

where $S(w) = \int_0^w \sin\left(\frac{\pi}{2}t^2\right) dt$ and $C(w) = \int_0^w \cos\left(\frac{\pi}{2}t^2\right) dt$ are the Fresnel integrals. Additionally, using the curvature function, the Cartesian coordinates of any generic point (s, d) can be computed, where $d \in [-0.5, 0.5]$ parameterizes points in the cross-section at a given position s along the backbone curve. The SIP is illustrated in Figure 2-8.

The dynamics of the SIP are derived as in [67], leading to a system of equations in the form of (2-21) with the state vector

$$\boldsymbol{q} = \begin{pmatrix} q_1 & q_2 \end{pmatrix}^T. \tag{2-44}$$

The stiffness and damping terms are defined as

$$\boldsymbol{K}(\boldsymbol{q}) = k\boldsymbol{H}\boldsymbol{q}, \quad \boldsymbol{D}(\boldsymbol{q})\dot{\boldsymbol{q}} = \beta\boldsymbol{H}\dot{\boldsymbol{q}}, \tag{2-45}$$

where $k, \beta \in \mathbb{R}$ represent the segment stiffness and damping coefficients, respectively, and $\boldsymbol{H}$ is the Hankel matrix with elements $H_{i,j} = \frac{1}{i+j-1}$. The actuation matrix, derived as in [67], is given by

$$\boldsymbol{A} = \begin{pmatrix} 1 & 2 \end{pmatrix}^T, \tag{2-46}$$

confirming that the system is underactuated.

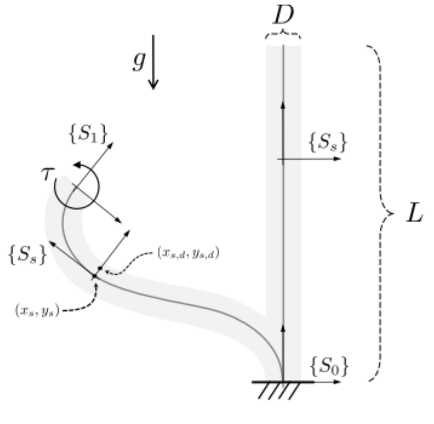


Figure 2-8: Soft Inverted Pendulum with Affine Curvature [13]

Many control strategies have been proposed for controlling the SIP system [13, 11, 70]. These control strategies include partial feedback linearization [71], energy based regulation and even a control law based on IDA-PBC [70]. These analytical controllers while simple to implement are often based on assumptions that might be challenging to verify across systems. The focus here will be on the partial feedback linearization controller derived in [13].

Partial Feedback Linearization

In [13] the author attempts to study the stabilization of the straight equilibrium using partial feedback linearization (PFL). For the same, a change of coordinates is required to clearly separate the system into an actuated portion and a completely under-actuated one, similar to [71]. Using $\boldsymbol{z}$ to denote the input-decoupled coordinates, we get:

$$\boldsymbol{z} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{3} \end{pmatrix}}_{\boldsymbol{H}} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} \tag{2-47}$$

Using this change of coordinates, the dynamics in (3-4) can be rewritten as:

$$\tilde{\boldsymbol{M}}(\boldsymbol{z})\ddot{\boldsymbol{z}} + \tilde{\boldsymbol{C}}(\boldsymbol{z}, \dot{\boldsymbol{z}})\dot{\boldsymbol{z}} + \tilde{\boldsymbol{G}}(\boldsymbol{z}) + \tilde{\boldsymbol{K}}\boldsymbol{z} + \tilde{\boldsymbol{D}}\dot{\boldsymbol{z}} = \begin{bmatrix} \boldsymbol{\tau} \\ 0 \end{bmatrix} \tag{2-48}$$

Here, $\tilde{\boldsymbol{M}}$, $\tilde{\boldsymbol{C}}$, etc., are the transformed system matrices via $\boldsymbol{H}^{-1}$ as detailed in [13].

In the next two sections, two control strategies will be discussed as in [13, 71]. In the first case, the output $y = z_1$ is considered. This output is collocated with the input $\boldsymbol{\tau}$ and is called collocated linearization. In the next case, the output $y = z_2$ is considered and is not collocated with the input. This strategy is called non-collocated linearization.

Collocated Linearization. In (2-48), $\ddot{z}_1$ can be extracted from the second row and substituted into the first to obtain a single input-single output (SISO) system in z_1:

$$\left(\tilde{M}_{11} - \frac{\tilde{M}_{12}^2}{\tilde{M}_{22}}\right)\ddot{z}_1 + \left(\tilde{h}_1 - \frac{\tilde{M}_{12}}{\tilde{M}_{22}}\tilde{h}_2\right) = \tau \tag{2-49}$$

where $\tilde{h} = \tilde{\boldsymbol{C}}\dot{\boldsymbol{z}} + \tilde{\boldsymbol{G}} + \tilde{\boldsymbol{K}}\boldsymbol{z} + \tilde{\boldsymbol{D}}\dot{\boldsymbol{z}}$.

This expression allows exact feedback linearization using the following control law:

$$\tau = \left(\tilde{h}_1 - \frac{\tilde{M}_{12}}{\tilde{M}_{22}}\tilde{h}_2\right) + \left(\tilde{M}_{11} - \frac{\tilde{M}_{12}^2}{\tilde{M}_{22}}\right)u \tag{2-50}$$

where u is a new virtual input for linearized dynamics. A simple PD law $u = -k_P z_1 - k_D \dot{z}_1$ achieves exponential convergence to the upright configuration.

To justify the use of this control law, two conditions must be satisfied:

1. The origin must be unstable in open loop, which corresponds to the composite stiffness and gravity term not being positive definite at the origin:

$$k\boldsymbol{H} + \left.\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{q}}\right|_{\boldsymbol{q}=0} \not\succ 0 \tag{2-51}$$

2. The zero dynamics must be locally stable, which is ensured if:

$$\tilde{K}_{22} + \left.\frac{\partial \tilde{G}_2}{\partial z_2}\right|_{z=0} > 0 \tag{2-52}$$

These conditions together imply that the system is underactuated and unstable in open loop (making control meaningful), while still admitting a locally stabilizable zero dynamics under collocated PFL.

Non-Collocated Linearization. For the non-collocated case, the output is $y = z_2$. The system is again rewritten to isolate $\ddot{z}_2$:

$$\left(\tilde{M}_{21} - \frac{\tilde{M}_{11}\tilde{M}_{22}}{\tilde{M}_{12}}\right)\ddot{z}_2 + \left(\tilde{M}_{11}\tilde{h}_2 - \tilde{M}_{12}\tilde{h}_1\right) = \tilde{M}_{12}\tau \tag{2-53}$$

This yields a feedback linearizing controller:

$$\tau = \left(\tilde{h}_1 - \frac{\tilde{M}_{11}}{\tilde{M}_{12}}\tilde{h}_2\right) + \left(\tilde{M}_{21} - \frac{\tilde{M}_{11}\tilde{M}_{22}}{\tilde{M}_{12}}\right)u \tag{2-54}$$

Here, the requirement $\tilde{M}_{12} \neq 0$ must be satisfied, ensuring inertial coupling. The closed-loop system becomes:

$$\ddot{z}_2 = u, \quad \tilde{M}_{21}\ddot{z}_1 + \tilde{h}_2 + \tilde{M}_{22}u = 0 \tag{2-55}$$

Since the zero dynamics have a marginally stable or even unstable mode (e.g., $\ddot{z}_1 = 0$), a local linearization around the equilibrium is considered and use pole placement or LQR to design u such that all eigenvalues of the linearized system lie in the left half-plane. The reachability condition is shown to hold under mild assumptions on the system parameters, ensuring controllability.

The non-collocated strategy successfully stabilizes the pendulum even for significantly lower stiffness values compared to the collocated strategy. However, beyond a certain stiffness threshold, even the non-collocated approach fails, despite the local equilibrium being attractive.

2-4 Optimal Control For Soft Robots

Optimal control offers a principled framework to govern complex dynamics by minimizing task-relevant objectives while respecting system constraints. In the context of soft robotics, it is particularly attractive due to its ability to harness elastic dynamics rather than override them, as some feedback strategies do [72, 73].

This section reviews the use of optimal control methods for soft robots. The focus is on identifying:

i the types of optimal control methods used (e.g., LQR, MPC, trajectory optimization),

ii the modeling approaches employed, and iii the extent to which underactuation is explicitly addressed.

Classical and modern techniques such as LQR, MPC, and nonlinear trajectory optimization have shown success in high-DOF and underactuated systems [74, 75, 76, 77, 78, 79]. These methods also support task-space control via embedded objectives [80].

Recent work applies optimal control to soft robots using data-driven models [63], reduced-order FEM models [81], and geometric representations such as PCC, polynomial curvature, and GVS-based parameterizations. This review emphasizes the latter category, which offers control-oriented formulations with tractable dynamics.

The section is organized as follows: first, learned and FEM-based model approaches are briefly introduced. Then, literature on model-based optimal control is presented in order of increasing complexity—from LQR to linear MPC, and finally nonlinear MPC and trajectory optimization. A critical discussion of their limitations follows.

Optimal Control Using Learned Models

Surrogate models such as neural networks, Koopman operators, and reduced-order data-driven representations have been increasingly used in conjunction with optimal control methods for soft robots. These approaches bypass analytical modeling by learning system dynamics from data.

Neural networks are used in [14, 82] to model soft robot kinematics and dynamics, enabling control via nonlinear MPC. AutoMPC [82] further automates controller design using Bayesian optimization and task-specific tuning. However, generalization remains a key challenge. Koopman operator theory provides linear embeddings of nonlinear dynamics, enabling the use of LQR and MPC in a lifted space [15]. K-MPC methods strike a balance between performance and real-time feasibility, while nonlinear Koopman NMPC achieves better accuracy at the cost of computational load. Other methods such as evolutionary NMPC [83] combine neural models with parallelized optimization to achieve real-time control on high-dimensional soft robots.

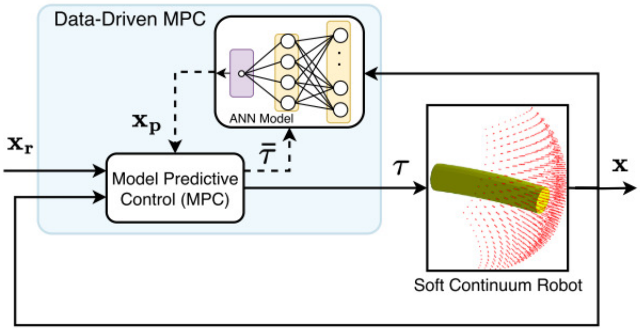


Figure 2-9: Neural network-based model predictive control for a soft robot [14].

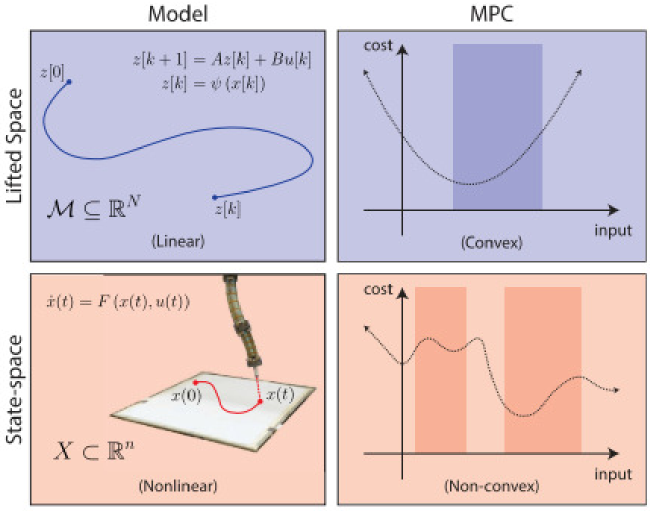


Figure 2-10: Koopman-based linear MPC on a lifted soft robot model [15].

Optimal Control Using Reduced-Order FEM Models

Finite Element Method (FEM) models offer high-fidelity, physics-based representations of soft robots, in contrast to data-driven methods. However, their high dimensionality hinders real-time control. Recent works [84, 16, 85] employ model-order reduction techniques to enable optimal control.

In [84], a ROM is derived for a soft elastomer robot using Proper Orthogonal Decomposition (POD). The full-order FEM model is projected onto a lower-dimensional subspace using data-driven snapshots. Further, to reduce computation, a piecewise affine ROM is created by linearizing around multiple operating points. A model predictive control (MPC) problem is formulated using this ROM and solved via Sequential Convex Programming (SCP). The approach supports real-time execution with local LQR feedback and outperforms Koopman-based methods in flexibility and frequency.

The work in [16] introduces an FEM-based trajectory optimization pipeline using Differential Dynamic Programming (DDP). The internal dynamics are condensed using implicit Euler integration and projected into task space. DDP solves the optimal control problem by backward and forward passes, iteratively updating control inputs. The method is applied in simulation to various soft robots, including the Stiff-Flop, and highlights DDP's advantage in producing feedback gains, although real-time MPC with DDP remains a challenge.

These works demonstrate that reduced-order FEM models can support advanced control strategies, bridging high-fidelity modeling with computational efficiency. The next sections examine optimal control with simplified models like PCC, polynomial curvature, and GVS-based representations.

Linear Quadratic Regulators (LQR)

LQR is a classical optimal control method used to regulate linear dynamical systems by minimizing a quadratic cost function over the system states and control inputs. Several

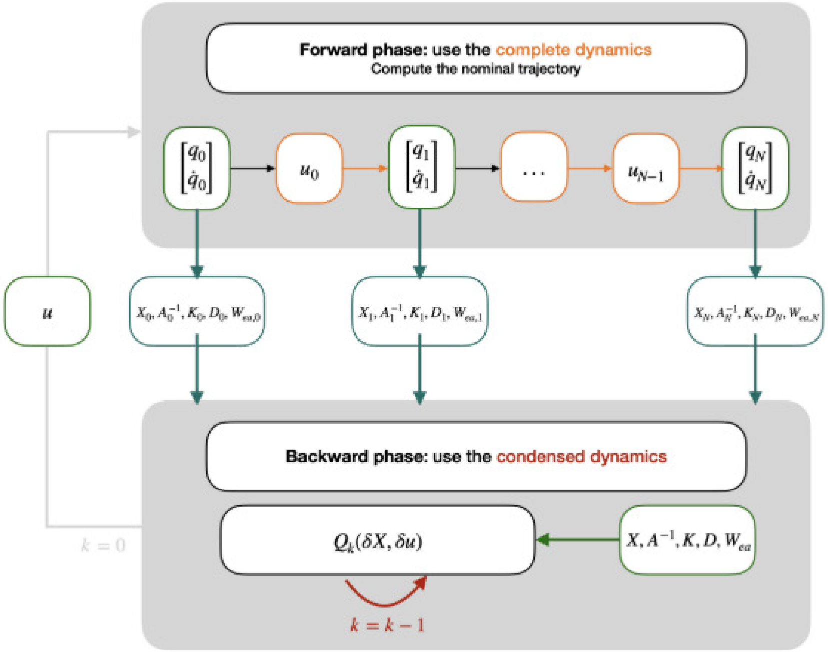


Figure 2-11: Trajectory optimization pipeline with FEM condensed dynamics and DDP [16].

works have explored its application to soft robotic systems [86, 87, 88].

In [86], a linear model of a soft pneumatic actuator and a syringe pump is developed, where the actuator is approximated as a cantilever beam undergoing small deformations, parameterized by a single bending angle θ. The system is modeled in state-space form, and an LQR controller is designed by solving the continuous-time Riccati equation:

$$A^T Y + YA - YBR^{-1}B^T Y + Q = 0, \tag{2-56}$$

where A, B represent the system matrices, and Q, R are weighting matrices. The optimal state-feedback law is then

$$u = -R^{-1}B^T Y\boldsymbol{x}. \tag{2-57}$$

In [87], LQR is applied following a feedback linearization of the soft actuator's second-order dynamics coupled with pressure dynamics. The linearized system is controlled using LQR with integral action, and Unscented Kalman Filtering (UKF) is used for state estimation. Comparative evaluation with PI and pole-placement controllers shows that the LQR-UKF combination yields superior performance.

In [88], an energy-based swing-up and stabilization strategy is proposed for a planar soft pendulum modeled using a variant of the polynomial curvature method. The swing-up is achieved by regulating the system to a target energy level using a Lyapunov-based approach. Once near the upright configuration, the dynamics are linearized, and an LQR controller is employed for stabilization.

These studies highlight the utility of LQR in soft robotics for both trajectory stabilization and regulation tasks, particularly when paired with state estimation or energy-based control strategies.

Linear Model Predictive Control (MPC)

Linear MPC extends the LQR framework by solving a finite-horizon optimal control problem at each sampling instant using the current system state as the initial condition. The resulting control input sequence optimizes a cost over state and input trajectories, with only the first input applied to the system. When the dynamics are linear and constraints are convex, the problem becomes a tractable convex optimization task [89, 90, 91, 92, 93, 94].

In [89, 90], linear MPC is applied to a 14-DoF soft humanoid robot ("King Louie") and a single-DoF soft joint ("grub"). Each joint is modeled as an independent inverted pendulum with simplified pressure dynamics, yielding a linear state-space model. The MPC is formulated with position, pressure, and rate constraints and executed at 300 Hz. Including pressure states led to improved tracking compared to earlier controllers.

In [91], the model is extended to include joint stiffness. Linearization is performed at each step, and MPC outperforms a sliding mode controller (SMC) in both position and stiffness tracking, particularly when priorities are weighted equally. MPC is also noted for easier tuning and generalization across multiple links.

Other linear MPC applications include beam-theoretic models for locomotion planning. In [92], a fish-like robot is modeled using the Euler–Bernoulli beam equation with a reduced linear model enabling task-level control. In [93], a segmented soft arm is modeled using central finite differences, and an H_∞ control problem is formulated for robust state feedback synthesis.

In [94], a linear MPC framework is implemented for a pneumatically actuated soft arm modeled using the Augmented Rigid Body formulation of PCC. A task-space controller is constructed via forward kinematics and validated in real hardware, running at 15 Hz. Robustness is enhanced via gain scheduling and offline-constrained MPC design [95].

These studies demonstrate that linear MPC, despite its reliance on simplifying assumptions, offers a powerful framework for real-time control of soft robots, particularly when paired with model reduction and real-time linearization strategies.

Nonlinear MPC and Trajectory Optimization

Soft robotic systems are inherently nonlinear, making linear control approximations insufficient for complex tasks. Nonlinear MPC (NMPC) and trajectory optimization (TO) provide more accurate frameworks, enabling dynamic behaviors and improved robustness. However, these approaches often come with significant computational demands and practical limitations.

Nonlinear MPC (NMPC)

Several works have implemented NMPC using physics-based models of soft robots. One example involves a quasi-static Cosserat rod model applied to a tendon-driven catheter, enabling hybrid position-force control but relying on precomputed trajectories, which limits adaptability [96]. A similar approach using the constant curvature (CC) assumption was used for visual servoing of an ablation catheter [97].

Robustness to uncertainty has been addressed through tube-based NMPC formulations that incorporate ancillary feedback terms, improving stability at the cost of implementation complexity [98]. An alternative method optimizes feedback gains directly using evolutionary strategies, offering resilience to model mismatch in lumped-parameter soft arms [99].

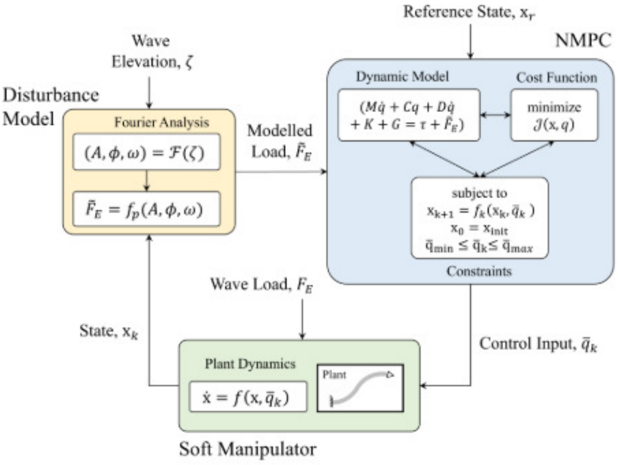


Figure 2-12: Overview of NMPC for underwater soft robot [17]

In underwater settings, NMPC has been used for disturbance rejection in multi-segment continuum robots, demonstrating robustness to actuator failure and wave-induced noise (Figure 2-12) [17]. More expressive models such as the polynomial curvature model have been employed to control a thrust-generating tentacle, though the resulting non-convex optimization problem required a heuristic genetic algorithm for real-time feasibility [100].

Trajectory Optimization (TO)

TO has proven effective for generating feasible trajectories in highly dynamic tasks. A prominent example uses a constant curvature model with direct collocation to enable dynamic grabbing motions. The trajectory is iteratively refined using an ILC framework to improve repeatability and performance [18].

Cosserat rod-based models have been used to control octopus-inspired arms, with optimized trajectories producing wave-like actuation patterns during reaching tasks [101]. Other approaches use discrete elastic rods to track reference paths for thermally actuated manipulators [102], while lumped-parameter models have been leveraged to track end-effector trajectories using nonlinear optimization [103].

Overall, NMPC offers real-time feedback and robustness but often faces challenges in model complexity and solver efficiency. TO enables highly dynamic open-loop behaviors but may suffer from sensitivity to model errors and lack of feedback. Both remain promising tools for addressing the challenges of soft robot control, especially under underactuation.

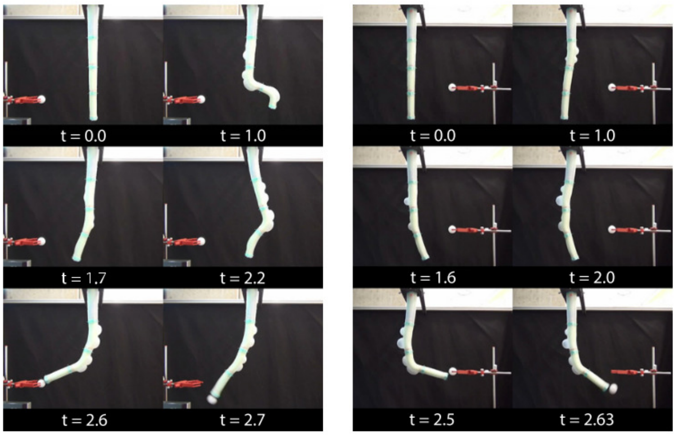


Figure 2-13: Dynamic grabbing via TO and ILC [18]

Research Gaps in Model-Based Optimal Control for Soft Robots

A wide range of model-based optimal control methods have been proposed for soft robots, spanning learned models, reduced-order FEM models, and simplified physics-based representations. Table 2-1 summarizes these contributions, organizing them by control strategy, modeling approach, and experimental validation.

Across the literature, two dominant modeling trends emerge. On one end, highly simplified models particularly those using the piecewise constant curvature (PCC) assumption enable tractable control formulations but neglect important dynamics. On the other end, complex models like discrete elastic rods (DER) or lumped-parameter rigid-body analogs offer richer dynamics but often introduce computational challenges. Only a few recent efforts, such as [100], attempt to bridge this gap by exploring intermediate representations like affine or polynomial curvature.

Another persistent issue is the treatment of underactuation. Although soft robots are often underactuated by nature, many studies either assume full actuation or fail to clearly specify the mapping between control inputs and generalized coordinates. Even when models appear underactuated, this is not always stated explicitly, making it difficult to assess the generality or robustness of the proposed controllers. For example, while [94] discusses task-space MPC and mentions handling underactuation, it remains unclear whether an underactuated model was actually employed.

Clear demonstrations of optimal control methods handling underactuation are rare. However, some notable exceptions do exist. The grabbing motions in [18] showcase the potential of trajectory optimization to exploit system compliance and inertia to achieve dynamic tasks beyond the quasi-static regime despite significant underactuation. Such examples highlight the promise of dynamics-aware optimal control formulations, especially when leveraging soft robots' natural compliance.

Ultimately, while the field has made significant progress in applying optimal control to soft robotic systems, major gaps remain in:

- Consistently modeling and stating underactuation.

- Balancing model expressiveness with computational feasibility.
- Exploiting compliance for dynamic behaviors through optimal control.

These gaps offer fertile ground for future work that aims to unify control theory with soft robot mechanics more rigorously.

Reference	**Control Method**	**Model Used**	**Underactuated**	**Control Task**	**Validation**
Yang et al. (2023)[86]	LQR	EBRT	No	Bending Angle Tracking	Experimental
Xavier et al. (2022)[87]	LQR	Lumped Parameter Model	No	Bending Angle Tracking	Experimental
Weerakoon et al. (2021)[88]	LQR	PCC	Yes	Shape Regulation	Simulation
Best et al. (2016,2021) [89, 90, 91]	MPC	Inverted Pendulum Model	No	Joint Angle Tracking	Experimental
Spinelli et al. (2022) [94]	MPC	PCC	-	Tip Pose Tracking	Experimental
Doroudchi et al. (2018) [93]	H_∞	EBRT	No	Decentralized State Feedback Control	Simulation
Barbosa et al. (2023) [92]	MPC	EBRT	No	Position Tracking	Simulation
Soltani et al. (2017) [96]	NMPC	CRT	-	Hybrid Position/Force Control	Experimental
Norouzi et al. (2021) [97]	NMPC	PCC	No	Visual Servoing	Experimental
Bastos et al. (2023) [98]	NMPC	PCC	Yes	Shape Tracking	Simulation
Jensen et al. (2024) [99]	NMPC	Lumped Parameter Model	-	Trajectory Tracking	Experimental
Walker et al. (2024) [17]	NMPC	PCC	No	Tip Pose Tracking	Simulation
Hachen et al. (2025) [104]	NMPC	PCC	No	Tip Pose Tracking	Experimental
Stella et al. (2022) [100]	NMPC	Affine Curvature Model	Yes	Thrust Generation	Experimental
Sanders et al. (2023) [103]	TrajOpt	Lumped Parameter Model	No	Tip Pose Tracking	Experimental
Marchese et al. (2015) [18]	TrajOpt	PCC	Yes	Grabbing	Experimental
Wang et al. (2021) [101]	TrajOpt	CRT	Yes	Reaching	Numerical
Wertz et al. (2022) [102]	TrajOpt	DER	Yes	Reaching	Experimental

Table 2-1: Summary of Model-Based Optimal Controllers for Soft Robots

Chapter 3

Problem Formulation and Proposed Approach

In this chapter, the research problem is formally stated, and the proposed approach to modeling and control is outlined. Before presenting the formulation, a concise discussion of the specific gaps in the literature that this work aims to address is provided, establishing the motivation and context for the proposed methodology.

3-1 Limitations of the State of the Art

Underactuation remains a fundamental challenge in the control of soft robots. Most model-based controllers in the literature rely on a *collocated* control paradigm, where actuators are placed to directly counteract the primary modes of instability. These methods assume that the robot possesses sufficiently high stiffness to passively stabilize its unactuated degrees of freedom. While such strategies effectively leverage the robot's natural impedance for control, they impose an inherent limitation: the robot must be stiff enough for the controller to work, thereby restricting operation in more compliant, low-stiffness regimes.

This thesis aims to address this limitation by investigating whether optimal control methods can extend the control of soft robots into these low-stiffness regimes, where traditional model-based controllers fail. In particular, the goal is to design controllers that explicitly stabilize the unactuated dynamics using the full system model, rather than relying on passive mechanical properties alone.

A second gap in the literature lies in the modeling choices used for optimal control. Existing works often either employ highly simplified models such as the Piecewise Constant Curvature (PCC) approximation or rely on computationally expensive high-dimensional models like reduced-order finite element models (FEM). In both cases, achieving real-time closed-loop control remains a significant challenge.

The objective of this thesis is thus twofold: first, to develop an optimal control framework capable of explicitly addressing underactuation; and second, to do so using reduced-order models that balance physical fidelity with computational efficiency. The long-term goal is to enable a generalizable framework that can be applied in real time across a broad range of soft robotic platforms.

3-2 Research Problem

The objective of this research is to develop a real-time feasible control strategy for underactuated soft robots by formulating a discrete-time nonlinear model predictive control (NMPC) problem. The goal is to stabilize the robot around unstable equilibria, particularly in low-stiffness regimes where traditional collocated control strategies fail.

Consider a slender soft robot modeled using a more accurate formulation (e.g., Polynomial Curvature or GVS-based models), resulting in a discretized dynamic system of the form:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_d(\boldsymbol{x}_k, \boldsymbol{u}_k)$$

where $\boldsymbol{x}_k \in \mathbb{R}^{2n}$ is the state vector (positions and velocities), $\boldsymbol{u}_k \in \mathbb{R}^m$ is the control input, and $\boldsymbol{f}_d$ represents the discretized system dynamics, with $m < n$ (underactuation).

The NMPC problem is formulated as:

$$\min_{\boldsymbol{u}_0,\ldots,\boldsymbol{u}_{N-1}} \quad \ell_f(\boldsymbol{x}_N) + \sum_{k=0}^{N-1} \ell(\boldsymbol{x}_k, \boldsymbol{u}_k)$$

subject to:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_d(\boldsymbol{x}_k, \boldsymbol{u}_k), \quad \forall k = 0, \ldots, N-1$$
$$\boldsymbol{x}_0 = \boldsymbol{x}_{\text{init}}, \quad \boldsymbol{u}_k \in \mathcal{U}, \quad \boldsymbol{x}_k \in \mathcal{X}$$

Here, ℓ is the stage cost penalizing deviation from the target state and control effort, and ℓ_f is a terminal cost. The goal is to regulate the robot to a desired unstable equilibrium $\bar{\boldsymbol{x}} = \begin{bmatrix} \bar{\boldsymbol{q}} & \boldsymbol{0} \end{bmatrix}^T$, where $\bar{\boldsymbol{q}}$ satisfies the static equilibrium condition.

The resulting optimal control policy $\boldsymbol{u}_0^*$ is applied in a receding horizon fashion, and the process is repeated at each timestep. Emphasis is placed on designing the controller to be computationally efficient to enable real-time execution.

This formulation enables the exploration of explicit unactuated DOF regulation, extending the operating envelope of model-based control into low-stiffness, dynamically unstable regimes.

3-3 Soft Inverted Pendulum System

To formulate the control problem properly, it is essential to define the system under consideration. As mentioned in Section 2-3-3, the soft inverted pendulum (SIP) [13] was introduced as a benchmark for studying control of soft robotic systems in a low-dimensional setting that nevertheless captures key challenges such as underactuation and compliance. In this work,

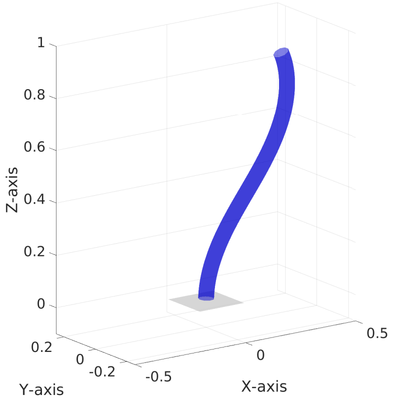


Figure 3-1: Soft Inverted Pendulum (SIP) system.

the SIP is modeled using the geometric variable strain (GVS) approach [60] rather than the polynomial curvature approach.

This choice is motivated by three reasons. First, GVS generalizes the polynomial curvature model in the planar case by using a polynomial strain basis, and therefore it is equivalent to modeling the SIP with polynomial curvature. Second, GVS provides a more flexible and unified mathematical framework: it allows the use of different strain bases, supports more strain modes and degrees of freedom, and also enables modeling of hybrid soft–rigid systems under the same formulation. While polynomial curvature can also handle the planar case, GVS is more systematic and extensible. Third, the proposed control approach (DDP) benefits from efficient computation of analytical gradients of the dynamics. Recent work [105] has demonstrated how such gradients can be computed for GVS models using the recursive Newton–Euler algorithm (RNEA). This makes GVS particularly suitable for implementing the control algorithm efficiently.

The next aspect to consider is the definition of the low-stiffness regime. As discussed in the previous chapter, many model-based soft robot control methods rely on assumptions ensuring that the zero dynamics are stable. This condition holds in elastically dominated cases, as captured in Assumption 2 ((2-34)). Even in elastically decoupled cases, uniqueness of the equilibrium requires the same condition to hold. The SIP provides a simple platform to investigate this assumption.

The SIP model is derived using the GVS approach using the paper [60] as a reference. The soft inverted pendulum is described just as in [13] but modeled using the GVS approach with an equivalent strain basis.

Strain Parameterization. The SIP is modeled with affine curvature with bending only about the $y-$axis. Thus the strain field is parameterized as follows:

$$\boldsymbol{\xi}(X,\boldsymbol{q}) = \underbrace{\begin{pmatrix} 0 & 0 \\ 1 & X \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}}_{\boldsymbol{\Phi}_\xi} \underbrace{\begin{pmatrix} q_1 \\ q_2 \end{pmatrix}}_{\boldsymbol{q}} + \underbrace{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}}_{\boldsymbol{\xi}^*} \tag{3-1}$$

Where $\boldsymbol{\xi}, \boldsymbol{\xi}^* \in \mathbb{R}^6, \boldsymbol{q} \in \mathbb{R}^2$. This monomial basis models linear bending about the $y-$axis.

Kinematics. As discussed in [60] the forward kinematics of the soft link is obtained by employing the Magnus expansion of the strain field $\boldsymbol{\xi}$ at X after which the exponential map can be applied as the matrix differential equation of the kinematics $\partial \boldsymbol{g}(X)/\partial X = \boldsymbol{g}\boldsymbol{\xi}(X)$ can be solved using the approximation.

$$\boldsymbol{g}(X) = \exp(\hat{\Omega}(X)) \tag{3-2}$$

A fourth-order Zannah quadrature approximation of Magnus expansion is used. As the Magnus expansion degrades for larger values of X the soft link is divided into smaller segments of lengths h and $\boldsymbol{g}(X)$ is evaluated recursively.

$$\boldsymbol{g}(X+h) = \boldsymbol{g}(X)\exp(\hat{\Omega}(h)) \tag{3-3}$$

Therefore for the forward kinematics the following steps are performed

i For a given $\boldsymbol{q}$ the strains $\boldsymbol{\xi}(X + (1/2 \mp \sqrt{3}/6)h)$ are evaluated at the first and second collocation points.

ii $\Omega(h)$ is evaluated using the Fourth-order Zannah quadrature approximation of Magnus expansion

iii $\boldsymbol{g}(X+h) = \boldsymbol{g}(X)\exp(\hat{\Omega}(h))$ is used to compute the $\boldsymbol{g}(X+h)$ that maps a point in the body-frame at $X+h$ to the body-frame at $X=0$ using $\boldsymbol{g}(X)$ computed from the previous iteration

Generalized Dynamics of the Rod. The dynamics of the entire rod in generalized coordinates are obtained by integrating free-body dynamic equations obtained from the Cosserat equilibrium, and projecting it on to the space of the generalized coordinates using the Jacobian, over the length of the soft body and substituting the elastic, damping, and actuation laws. The resulting equations take the following form:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{K}\boldsymbol{q} + \boldsymbol{D}\dot{\boldsymbol{q}} = \boldsymbol{A}(\boldsymbol{q})\boldsymbol{u}, \tag{3-4}$$

Where the SIP is subjected to gravity and an external torque at the tip. The terms are computed as:

$$\begin{aligned}
\boldsymbol{M}(\boldsymbol{q}) &= \int_0^L \boldsymbol{J}^T \boldsymbol{\mathcal{M}} \boldsymbol{J}\, dX, \\
\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) &= \int_0^L \boldsymbol{J}^T (\mathrm{ad}^*_{\boldsymbol{\eta}} \boldsymbol{\mathcal{M}} \boldsymbol{J} + \boldsymbol{\mathcal{M}} \dot{\boldsymbol{J}})\, dX, \\
\boldsymbol{G}(\boldsymbol{q}) &= \int_0^L \boldsymbol{J}^T \boldsymbol{\mathcal{M}} \left(\mathrm{Ad}_{\boldsymbol{g}}^{-1} \boldsymbol{\mathcal{G}} \right) dX, \\
\boldsymbol{K} &= \int_0^L \Phi_{\boldsymbol{\xi}}^T \Sigma \Phi_{\boldsymbol{\xi}}\, dX, \\
\boldsymbol{D} &= \int_0^L \Phi_{\boldsymbol{\xi}}^T \Upsilon \Phi_{\boldsymbol{\xi}}\, dX,
\end{aligned} \tag{3-5}$$

where $\boldsymbol{\mathcal{G}} \in \mathbb{R}^6$ is the gravitational acceleration twist expressed in the inertial frame. Other terms are defined as given below:

$$\begin{aligned}
\boldsymbol{\mathcal{M}} &= \rho \cdot \mathrm{diag}(J_x, J_y, J_z, A, A, A) \\
\boldsymbol{\mathcal{G}} &= (0\ 0\ 0\ 0\ 0\ 9.8) \\
\Sigma &= \mathrm{diag}(GJ_x, EJ_y, EJ_z, EA, GA, GA) \\
\Upsilon &= \upsilon \cdot \mathrm{diag}(J_x, 3J_y, 3J_z, 3A, A, A) \\
G &= \frac{E}{2(1+\nu)} \\
J_y &= \frac{\pi}{4} r^4,\ J_z = J_y,\ J_x = J_y + J_z
\end{aligned}$$

Input Matrix $\boldsymbol{A}(\boldsymbol{q})$. Since an external torque(about the $y-$axis)is applied it needs to be transformed to the local frame in order to calculate the integral for the input matrix. The external torque in the spatial frame is

$$\mathcal{F}_{global} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T \boldsymbol{u} \tag{3-6}$$

Where $\boldsymbol{\tau} \in \mathbb{R}$. To evaluate the effect of this external torque in the dynamics the force needs to be mapped to the local frame. Since the point of application of the force is same in the local and global frame only the rotation with respect to the body frame at $X = 0$ is required. Thus it is obtained as follows:

$$\boldsymbol{g}(X = L) = \begin{pmatrix} \boldsymbol{R} & r \\ \boldsymbol{0} & 1 \end{pmatrix},\ \boldsymbol{g}_\theta(X = L) = \begin{pmatrix} \boldsymbol{R} & \boldsymbol{0} \\ \boldsymbol{0} & 1 \end{pmatrix} \tag{3-7}$$

The force is mapped to the local frame using the Coadjoint map of $SE(3)$:

$$\mathcal{F}_{local} = \mathrm{Ad}^*_{\boldsymbol{g}_\theta^{-1}} \mathcal{F}_{global} = \mathrm{Ad}^*_{\boldsymbol{g}_\theta^{-1}} \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T \boldsymbol{u} \tag{3-8}$$

Finally the integral for the external force applied in the generalized coordinates is given as :

$$\boldsymbol{A}(\boldsymbol{q})\boldsymbol{u} = \int_0^L \boldsymbol{J}^T \mathcal{F}_{local}(X)\, dX = \underbrace{\boldsymbol{J}^T_{X=L} \mathrm{Ad}^*_{\boldsymbol{g}_\theta^{-1}} \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T}_{\boldsymbol{A}(\boldsymbol{q})} \boldsymbol{u} \tag{3-9}$$

The spatial integrals in equation (3-5) are evaluated using Gaussian-Legendre quadrature scheme. This method evaluates the integrands at discrete points, providing an accurate numerical approximation of the integrals.

$$\int_0^L f(X)dX = \sum_{k=1}^{n_p} w_k f(X_k) \tag{3-10}$$

Where n_p is the number of integration points on the backbone curve and w_k are their respective weights.

For the SIP, two conditions must be satisfied for collocated control via partial feedback linearization (PFL):

1. The origin must be unstable in open loop, which corresponds to the composite stiffness and gravity term not being positive definite at the origin:

$$k\boldsymbol{H} + \left.\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{q}}\right|_{\boldsymbol{q}=0} \not\succ 0 \tag{3-11}$$

2. The zero dynamics must be locally stable, which is ensured if:

$$\tilde{K}_{22} + \left.\frac{\partial \tilde{G}_2}{\partial z_2}\right|_{\boldsymbol{z}=0} > 0 \tag{3-12}$$

Together, these conditions imply that the system is underactuated and unstable in open loop (making the control problem meaningful), while still admitting locally stabilizable zero dynamics under collocated PFL.

These conditions also allow the definition of feasible regions in terms of the mass parameter m and stiffness parameter k, for which collocated PFL works. The goal of this work is then to compare the proposed NMPC approach with PFL, particularly in regions where collocated and non-collocated PFL fail to provide stability.

The collocated PFL law used is:

$$\tau = \left(\tilde{h}_1 - \frac{\tilde{M}_{12}}{\tilde{M}_{22}}\tilde{h}_2\right) + \left(\tilde{M}_{11} - \frac{\tilde{M}_{12}^2}{\tilde{M}_{22}}\right) u \tag{3-13}$$

where u is a virtual input for the linearized dynamics. A simple PD control law $u = -k_P z_1 - k_D \dot{z}_1$ achieves exponential convergence to the upright configuration. The gains k_P and k_D are chosen to match the closed-loop performance of the NMPC controller for a nominal mass and stiffness value, by shaping the second-order dynamics of the output variable. Specifically, we compare the linearized dynamics

$$\ddot{z}_1 + k_D \dot{z}_1 + k_P z_1 = 0$$

to the standard second-order form

$$\ddot{z}_1 + 2\zeta\omega_n \dot{z}_1 + \omega_n^2 z_1 = 0$$

and tune the gains to achieve the same settling time as the nominal NMPC case.

Similarly, the non-collocated PFL law is:

$$\tau = \left(\tilde{h}_1 - \frac{\tilde{M}_{11}}{\tilde{M}_{12}}\tilde{h}_2\right) + \left(\tilde{M}_{21} - \frac{\tilde{M}_{11}\tilde{M}_{22}}{\tilde{M}_{12}}\right)u \tag{3-14}$$

Since the zero dynamics in this case have a marginally stable or even unstable mode (e.g., $\ddot{z}_1 = 0$), a local linearization around the equilibrium is considered. Pole placement or LQR design is then used for u, ensuring that all eigenvalues of the linearized system lie in the left half-plane. Under mild assumptions on the system parameters, the reachability condition holds, guaranteeing controllability.

3-4 Differential Dynamic Programming for NMPC

Differential Dynamic Programming (DDP) [106, 107] is a widely used trajectory optimization method for nonlinear systems. Originally introduced by Mayne, DDP iteratively improves a nominal state-control trajectory using a backward pass based on Riccati-like recursions and a forward rollout of the system dynamics. The algorithm exhibits two key features that make it attractive for nonlinear model predictive control (NMPC): (i) linear complexity in the prediction horizon due to stagewise Riccati recursions, and (ii) local quadratic convergence when second-order information is retained. When only first-order approximations are considered, the method reduces to the iterative Linear Quadratic Regulator (iLQR) [108].

As a shooting-based method, DDP ensures that the system dynamics are satisfied at every iteration of the forward rollout. This contrasts with direct transcription approaches where dynamics are imposed as constraints and may be violated between collocation points. In addition, the backward pass of DDP yields, at no extra cost, a time-varying local state-feedback policy rather than a purely open-loop trajectory. This policy improves robustness to disturbances and model mismatch, and can be used to execute precomputed trajectories when online computation is limited. These properties have led to its widespread adoption in robotics, where fast NMPC solvers are required to operate under real-time constraints [109, 110].

In the broader landscape of optimization-based control, DDP belongs to a family of algorithms that exploit the time-induced sparsity of optimal control problems. Recent work has shown that multiple-shooting Sequential Quadratic Programming (SQP) solvers can also exploit this sparsity and achieve state-of-the-art performance in constrained MPC [111]. Nevertheless, DDP remains highly attractive for soft robotics applications due to its efficiency in unconstrained or softly constrained settings, its ability to generate stabilizing feedback, and its natural integration with existing dynamics libraries.

First, the shape regulation problem is considered with a quadratic cost. Therefore, the OCP in the problem statement is discretized as follows:

$$\begin{aligned}
\min_{\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_{N-1}} \quad & \frac{1}{2}\tilde{\boldsymbol{x}}_N^T \boldsymbol{W}_N \tilde{\boldsymbol{x}}_N + \sum_{k=1}^{N-1}\left(\frac{1}{2}\tilde{\boldsymbol{x}}_k^T \boldsymbol{W}_k \tilde{\boldsymbol{x}}_k + \frac{1}{2}\boldsymbol{u}_k^T \boldsymbol{R}_k \boldsymbol{u}_k\right) \\
& \text{s.t.} \\
& \boldsymbol{x}_{k+1} = \boldsymbol{f}_d(\boldsymbol{x}_k, \boldsymbol{u}_k), \quad \forall\ k = 1, 2, \cdots, N-1 \\
& \tilde{\boldsymbol{x}}_k = (\boldsymbol{x}_k - \boldsymbol{x}_g)
\end{aligned} \tag{3-15}$$

where $\boldsymbol{f}_d(\cdot)$ represents the discretized system dynamics. The discrete-time OCP is solved using the DDP algorithm. The cost weighting matrices $\boldsymbol{W}_k$ and $\boldsymbol{R}_k$ are, respectively, positive-semidefinite and positive-definite. The DDP algorithm requires defining the *optimal cost-to-go* $V_k^*(\boldsymbol{x})$ which is the total cost that will be accumulated between time steps k and N, starting in state $\boldsymbol{x}$, if the optimal policy is followed. Using Bellman's principle of optimality the following recurrence relation can be realized:

$$\begin{aligned} V_N^*(\boldsymbol{x}) &= \frac{1}{2}\tilde{\boldsymbol{x}}^T \boldsymbol{W}_N \tilde{\boldsymbol{x}} \\ V_k^*(\boldsymbol{x}) &= \min_{\boldsymbol{u}} \left(\frac{1}{2}\tilde{\boldsymbol{x}}^T \boldsymbol{W}_k \tilde{\boldsymbol{x}} + \frac{1}{2}\boldsymbol{u}^T \boldsymbol{R}_k \boldsymbol{u} + V_{k+1}^*(\boldsymbol{f}_d(\boldsymbol{x}, \boldsymbol{u})) \right) \\ &\equiv \min_{\boldsymbol{u}} \; Q_k^*(\boldsymbol{x}, \boldsymbol{u}) \end{aligned} \tag{3-16}$$

This is the classical dynamic programming update procedure. The cost-to-go function at a k is nonlinear due to the dynamics of the system and is difficult to solve in general. DDP overcomes this by starting with an initial guess trajectory and taking a second order approximation of the cost-to-go function near that trajectory. The algorithm proceeds by calculating the following approximate cost-to-go at each time step.

$$Q_k(\boldsymbol{x} + \delta\boldsymbol{x}, \boldsymbol{u} + \delta\boldsymbol{u}) \approx Q_k(\boldsymbol{x}, \boldsymbol{u}) + \frac{1}{2} \begin{bmatrix} \delta\boldsymbol{x} \\ \delta\boldsymbol{u} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{Q_{xx,k}} & \boldsymbol{Q_{ux,k}^T} \\ \boldsymbol{Q_{ux,k}} & \boldsymbol{Q_{uu,k}} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{x} \\ \delta\boldsymbol{u} \end{bmatrix} + \begin{bmatrix} \boldsymbol{Q_{x,k}} \\ \boldsymbol{Q_{u,k}} \end{bmatrix}^T \begin{bmatrix} \delta\boldsymbol{x} \\ \delta\boldsymbol{u} \end{bmatrix} \tag{3-17}$$

Where the Hessian block matrices and gradient vectors are:

- $\boldsymbol{Q_{x,k}} = \boldsymbol{W_k}\tilde{\boldsymbol{x}} + \left(\frac{\partial \boldsymbol{f_d}}{\partial \boldsymbol{x}}\right)^T \boldsymbol{g_{k+1}}$
- $\boldsymbol{Q_{u,k}} = \boldsymbol{R_k}\boldsymbol{u} + \left(\frac{\partial \boldsymbol{f_d}}{\partial \boldsymbol{u}}\right)^T \boldsymbol{g_{k+1}}$
- $\boldsymbol{Q_{xx,k}} = \boldsymbol{W_k} + \left(\frac{\partial \boldsymbol{f_d}}{\partial \boldsymbol{x}}\right)^T \boldsymbol{H_{k+1}} \left(\frac{\partial \boldsymbol{f_d}}{\partial \boldsymbol{x}}\right) + \left(\frac{\partial^2 \boldsymbol{f_d}}{\partial \boldsymbol{x}^2}\right) \cdot \boldsymbol{g_{k+1}}$
- $\boldsymbol{Q_{uu,k}} = \boldsymbol{R_k} + \left(\frac{\partial \boldsymbol{f_d}}{\partial \boldsymbol{u}}\right)^T \boldsymbol{H_{k+1}} \left(\frac{\partial \boldsymbol{f_d}}{\partial \boldsymbol{u}}\right) + \left(\frac{\partial^2 \boldsymbol{f_d}}{\partial \boldsymbol{u}^2}\right) \cdot \boldsymbol{g_{k+1}}$
- $\boldsymbol{Q_{ux,k}} = \left(\frac{\partial \boldsymbol{f_d}}{\partial \boldsymbol{u}}\right)^T \boldsymbol{H_{k+1}} \left(\frac{\partial \boldsymbol{f_d}}{\partial \boldsymbol{x}}\right) + \left(\frac{\partial^2 \boldsymbol{f_d}}{\partial \boldsymbol{u}\partial \boldsymbol{x}}\right) \cdot \boldsymbol{g_{k+1}}$

Minimize Q_k with respect to $\delta\boldsymbol{u}$:

$$\delta\boldsymbol{u}_k = -\boldsymbol{Q_{uu,k}^{-1}}(\boldsymbol{Q_{ux,k}}\delta\boldsymbol{x} + \boldsymbol{Q_{u,k}}) = -\boldsymbol{K_k}\delta\boldsymbol{x} - \boldsymbol{l_k} \tag{3-18}$$

- $\boldsymbol{l_k} = \boldsymbol{Q_{uu,k}^{-1}}\boldsymbol{Q_{u,k}}$ (feedforward term)
- $\boldsymbol{K_k} = \boldsymbol{Q_{uu,k}^{-1}}\boldsymbol{Q_{ux,k}}$ (feedback term)

Update $\boldsymbol{H_k}$ and $\boldsymbol{g_k}$:

- $\Delta V_k = -\frac{1}{2}\boldsymbol{Q_{u,k}^T}\boldsymbol{Q_{uu,k}^{-1}}\boldsymbol{Q_{u,k}}$

- $\boldsymbol{H_k} = \boldsymbol{Q_{xx,k}} - \boldsymbol{Q_{ux,k}^T} \boldsymbol{Q_{uu,k}^{-1}} \boldsymbol{Q_{ux,k}}$
- $\boldsymbol{g_k} = \boldsymbol{Q_{x,k}} - \boldsymbol{Q_{ux,k}^T} \boldsymbol{Q_{uu,k}^{-1}} \boldsymbol{Q_{u,k}}$

This recurrence relation and the second-order approximation are continued backward until $k = 1$. After that a forward pass is performed with the new corrected control law $\boldsymbol{u}_k(\boldsymbol{x})$ from (3-18) to compute a new trajectory $\{\boldsymbol{x}_k\}_{k=1}^N$. The backward and forward pass steps are repeated until convergence. The classical DDP algorithm is summarized in 1. Two more things are generally performed in practice to ensure good convergence properties. First, a regularization term must sometimes be added to $\boldsymbol{Q_{uu,k}}$ in equation (3-18) to ensure positive-definiteness. Second, a line search must be performed during the forward pass of the algorithm to ensure a sufficient decrease in cost is achieved.

Algorithm 1 Differential Dynamic Programming

```
procedure DDP($\boldsymbol{x}, \boldsymbol{u}, \epsilon$)
    repeat
        $\boldsymbol{K}, \boldsymbol{l}, \delta V \leftarrow$ backward pass to compute new corrected control law $\boldsymbol{u}_k(\boldsymbol{x})$
        $\boldsymbol{x}, \boldsymbol{u}, \delta J \leftarrow$ ForwardPass($\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{K}, \boldsymbol{l}, \delta V$)
    until $|\delta J| < \epsilon$
    return $\boldsymbol{x}, \boldsymbol{u}$
end procedure
function ForwardPass($\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{K}, \boldsymbol{l}, \delta V$)
    $\alpha = 1$
    repeat
        for $k = 1 \ldots N$ do
            $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{f_d}(\boldsymbol{x_k}, \boldsymbol{u_k} - \alpha \boldsymbol{l_k} - \boldsymbol{K_k} \delta \boldsymbol{x_k})$
        end for
        $J \leftarrow$ calculate cost from OCP
        $\alpha \leftarrow$ reduce according to line search update
    until line search conditions are satisfied
    return $\boldsymbol{x}, \boldsymbol{u}, \delta J$
end function
```

3-5 Implementation of Nonlinear Model Predictive Control

The DDP algorithm described in the previous section can be embedded into a receding-horizon control framework to implement Nonlinear Model Predictive Control (NMPC). At each sampling time t_k, the following finite-horizon OCP is solved:

$$\begin{aligned} \min_{\{\boldsymbol{u}_i\}_{i=k}^{k+N-1}} \quad & \frac{1}{2}\tilde{\boldsymbol{x}}_{k+N}^\top \boldsymbol{W}_N \tilde{\boldsymbol{x}}_{k+N} + \sum_{i=k}^{k+N-1} \left(\frac{1}{2}\tilde{\boldsymbol{x}}_i^\top \boldsymbol{W}_i \tilde{\boldsymbol{x}}_i + \frac{1}{2}\boldsymbol{u}_i^\top \boldsymbol{R}_i \boldsymbol{u}_i \right) \\ \text{s.t.} \quad & \boldsymbol{x}_{i+1} = \boldsymbol{f}_d(\boldsymbol{x}_i, \boldsymbol{u}_i), \quad \tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i - \boldsymbol{x}_g. \end{aligned} \tag{3-19}$$

Unlike offline trajectory optimization, NMPC must run in real time. Therefore, instead of iterating DDP to convergence, only a small number of backward–forward passes (1–3) are executed at each time step. To accelerate convergence, the solver is *warm-started* with

Algorithm 2 NMPC with Differential Dynamic Programming

```
Require: horizon $N$, short DDP iterations $n_{\text{iter}} \in \{1,2,3\}$, weights $\{\boldsymbol{W}_k, \boldsymbol{R}_k\}$, goal $\boldsymbol{x}_g$, dynamics $\boldsymbol{f}_d$, initial state $\boldsymbol{x}_0$, warm starts $\{\boldsymbol{x}_i^{(0)}\}, \{\boldsymbol{u}_i^{(0)}\}$, flag USE_FB $\in \{$true, false$\}$
for $k = 0, 1, 2, \dots$ do
    $\boldsymbol{x}_k \leftarrow$ current state
    Warm start: initialize trajectories
        $\{\boldsymbol{x}_i\}_{i=k}^{k+N} \leftarrow \{\boldsymbol{x}_i^{(0)}\}, \quad \{\boldsymbol{u}_i\}_{i=k}^{k+N-1} \leftarrow \{\boldsymbol{u}_i^{(0)}\}$
    Short DDP solve: run $n_{\text{iter}}$ backward–forward passes
        obtain $\{\boldsymbol{l}_i, \boldsymbol{K}_i\}_{i=k}^{k+N-1}$ and updated $\{\boldsymbol{x}_i^\star, \boldsymbol{u}_i^\star\}$
    Apply control:
    if USE_FB then
        $\boldsymbol{u}_k^{\text{apply}} \leftarrow \boldsymbol{u}_k^\star - \boldsymbol{K}_k(\boldsymbol{x}_k - \boldsymbol{x}_k^\star)$
    else
        $\boldsymbol{u}_k^{\text{apply}} \leftarrow \boldsymbol{u}_k^\star$
    end if
    Plant update:
        $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{f}_d(\boldsymbol{x}_k, \boldsymbol{u}_k^{\text{apply}})$
    Warm start shift:
    for $i = k, \dots, k+N-2$ do
        $\boldsymbol{x}_i^{(0)} \leftarrow \boldsymbol{x}_{i+1}^\star, \quad \boldsymbol{u}_i^{(0)} \leftarrow \boldsymbol{u}_{i+1}^\star$
    end for
    $\boldsymbol{x}_k^{(0)} \leftarrow \boldsymbol{x}_{k+1}, \quad \boldsymbol{x}_{k+N}^{(0)} \leftarrow \boldsymbol{x}_{k+N}^\star$
    $\boldsymbol{u}_{k+N-1}^{(0)} \leftarrow \boldsymbol{0}$
end for
```

the shifted solution from the previous NMPC iteration. That is, the nominal trajectories $\{\boldsymbol{x}_i^{(0)}, \boldsymbol{u}_i^{(0)}\}$ are updated by inserting the measured state $\boldsymbol{x}_k$ at the front, discarding the oldest element, and appending a fresh terminal guess.

The DDP backward pass produces feedforward and feedback terms $(\boldsymbol{l}_i, \boldsymbol{K}_i)$, yielding a local affine control law

$$\delta \boldsymbol{u}_i = -\boldsymbol{l}_i - \boldsymbol{K}_i \delta \boldsymbol{x}_i, \qquad i = k, \dots, k+N-1. \tag{3-20}$$

At each NMPC step, only the first input $\boldsymbol{u}_k$ is applied to the system, either in open-loop form

$$\boldsymbol{u}_k^{\text{apply}} = \boldsymbol{u}_k^\star,$$

or in closed-loop form using the local feedback gain

$$\boldsymbol{u}_k^{\text{apply}} = \boldsymbol{u}_k^\star - \boldsymbol{K}_k(\boldsymbol{x}_k - \boldsymbol{x}_k^\star).$$

The plant evolves forward according to $\boldsymbol{x}_{k+1} = \boldsymbol{f}_d(\boldsymbol{x}_k, \boldsymbol{u}_k^{\text{apply}})$, after which the warm start is updated and the process repeats.

This procedure ensures that NMPC retains the predictive and optimizing nature of DDP while remaining computationally tractable under real-time constraints. A summary is given in Algorithm 2.

3-6 Analytical Derivatives of GVS Dynamics

The backward pass of the DDP algorithm requires first- and second-order derivatives of the discretized dynamics

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_d(\boldsymbol{x}_k, \boldsymbol{u}_k). \tag{3-21}$$

In principle, these derivatives can be computed using finite differences or automatic differentiation. However, such approaches quickly become computationally expensive for high-dimensional soft-robot models, making them unsuitable for real-time NMPC. For DDP, it is therefore essential to obtain analytical gradients of the dynamics that are both accurate and fast to evaluate.

Consider the continuous-time dynamics of the soft robot written in strain coordinates:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \boldsymbol{FD}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{u}) \end{bmatrix}, \tag{3-22}$$

where $\boldsymbol{q}$ are the generalized strain coordinates, $\dot{\boldsymbol{q}}$ their velocities, and $\boldsymbol{FD}$ denotes the acceleration term obtained from the equations of motion.

Applying explicit Euler integration with time step Δt yields the discrete-time dynamics

$$\boldsymbol{f}_d(\boldsymbol{x}_k, \boldsymbol{u}_k) = \begin{bmatrix} \boldsymbol{q}_k \\ \dot{\boldsymbol{q}}_k \end{bmatrix} + \Delta t \begin{bmatrix} \dot{\boldsymbol{q}}_k \\ \boldsymbol{FD}(\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k, \boldsymbol{u}_k) \end{bmatrix}. \tag{3-23}$$

The Jacobians of $\boldsymbol{f}_d$ with respect to the state and control are then

$$\frac{\partial \boldsymbol{f}_d}{\partial \boldsymbol{x}} = \begin{bmatrix} \boldsymbol{I} & \Delta t\, \boldsymbol{I} \\ \Delta t \, \dfrac{\partial \boldsymbol{FD}}{\partial \boldsymbol{q}} & \boldsymbol{I} + \Delta t \, \dfrac{\partial \boldsymbol{FD}}{\partial \dot{\boldsymbol{q}}} \end{bmatrix}, \qquad \frac{\partial \boldsymbol{f}_d}{\partial \boldsymbol{u}} = \Delta t \begin{bmatrix} \boldsymbol{0} \\ \dfrac{\partial \boldsymbol{FD}}{\partial \boldsymbol{u}} \end{bmatrix}. \tag{3-24}$$

Thus, the problem of obtaining the DDP gradients reduces to computing the partial derivatives of the continuous dynamics function $\boldsymbol{FD}$ with respect to $\boldsymbol{q}, \dot{\boldsymbol{q}}$, and $\boldsymbol{u}$.

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} = \underbrace{\boldsymbol{A}(\boldsymbol{q})\boldsymbol{\tau} - \boldsymbol{K}\boldsymbol{q} - \boldsymbol{D}\dot{\boldsymbol{q}}}_{\bar{\boldsymbol{\tau}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\tau})} + \underbrace{\big(- \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{G}(\boldsymbol{q})\big)}_{\boldsymbol{F}(\boldsymbol{q}, \dot{\boldsymbol{q}})} = \bar{\boldsymbol{\tau}} + \boldsymbol{F}.$$

Thus the forward dynamics map is $\ddot{\boldsymbol{q}} = \boldsymbol{FD}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\tau})$ with $\boldsymbol{M}(\boldsymbol{q})\, \boldsymbol{FD} = \bar{\boldsymbol{\tau}} + \boldsymbol{F}$.

Differentiate w.r.t. $\boldsymbol{q}$ (treating $\ddot{\boldsymbol{q}}$ as $\boldsymbol{FD}$):

$$\frac{\partial \boldsymbol{M}}{\partial \boldsymbol{q}} \ddot{\boldsymbol{q}} + \boldsymbol{M} \frac{\partial \boldsymbol{FD}}{\partial \boldsymbol{q}} = \frac{\partial \bar{\boldsymbol{\tau}}}{\partial \boldsymbol{q}} + \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{q}}.$$

Introduce the inverse-dynamics residual

$$\boldsymbol{ID}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) := \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} - \boldsymbol{F}(\boldsymbol{q}, \dot{\boldsymbol{q}}), \quad \Rightarrow \quad \frac{\partial \boldsymbol{ID}}{\partial \boldsymbol{q}} = \frac{\partial \boldsymbol{M}}{\partial \boldsymbol{q}} \ddot{\boldsymbol{q}} - \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{q}}.$$

Rearranging gives the FD Jacobian:

$$\frac{\partial \boldsymbol{FD}}{\partial \boldsymbol{q}} = \boldsymbol{M}^{-1} \Big(\frac{\partial \bar{\boldsymbol{\tau}}}{\partial \boldsymbol{q}} - \frac{\partial \boldsymbol{ID}}{\partial \boldsymbol{q}} \Big) \tag{3-25}$$

Similarly, differentiating w.r.t. $\dot{\boldsymbol{q}}$ yields

$$\boldsymbol{M}\,\frac{\partial \boldsymbol{FD}}{\partial \dot{\boldsymbol{q}}} = \frac{\partial \bar{\boldsymbol{\tau}}}{\partial \dot{\boldsymbol{q}}} + \frac{\partial \boldsymbol{F}}{\partial \dot{\boldsymbol{q}}}, \qquad \frac{\partial \boldsymbol{ID}}{\partial \dot{\boldsymbol{q}}} = -\frac{\partial \boldsymbol{F}}{\partial \dot{\boldsymbol{q}}},$$

hence :

$$\frac{\partial \boldsymbol{FD}}{\partial \dot{\boldsymbol{q}}} = \boldsymbol{M}^{-1}\Big(\frac{\partial \bar{\boldsymbol{\tau}}}{\partial \dot{\boldsymbol{q}}} - \frac{\partial \boldsymbol{ID}}{\partial \dot{\boldsymbol{q}}}\Big) \tag{3-26}$$

Following [112], [105] uses The Recursive Newton Euler Algorithm (RNEA) which is the most efficient algorithm for the computation of inverse dynamics. It is a two-pass algorithm with a forward pass that calculates link kinematics and a backward pass that determines the joint wrench needed to produce this motion. This RNEA strategy is applied to the GVS rod in [105]: a *forward pass* recursively propagates poses and twists along the discretized pseudo-joint chain, and a *backward pass* accumulates spatial wrenches from the tip to the base and projects them onto the generalized coordinates.

For a soft body discretized into n_p computational points, the inverse dynamics is

$$ID \;=\; \sum_{\alpha=1}^{n_p-1} ID_\alpha \;=\; \sum_{\alpha=1}^{n_p-1} \boldsymbol{S}_\alpha^\top \boldsymbol{F}_\alpha^C, \tag{3-27}$$

with the cumulative wrench at point α given by the coadjoint-transported sum of distal point wrenches,

$$\boldsymbol{F}_\alpha^C \;=\; \sum_{k=\alpha+1}^{n_p} \mathrm{Ad}^*_{\boldsymbol{g}_{\alpha k}}\, \boldsymbol{F}_k, \tag{3-28}$$

and each local point wrench

$$\boldsymbol{F}_k \;=\; \boldsymbol{M}_k\, \dot{\boldsymbol{\eta}}_k \;+\; \mathrm{ad}^*_{\boldsymbol{\eta}_k}\, \boldsymbol{M}_k\, \boldsymbol{\eta}_k \;-\; \boldsymbol{M}_k\, \mathrm{Ad}^{-1}_{\boldsymbol{g}_k}\, \boldsymbol{G}. \tag{3-29}$$

Here $\boldsymbol{S}_\alpha$ is the joint motion subspace at α, $\boldsymbol{g}_{\alpha k}$ is the transform from α to k, $\boldsymbol{g}_k$ is the world-to-k transform, $\boldsymbol{\eta}_k$ and $\dot{\boldsymbol{\eta}}_k$ are the local twist and twist-rate, $\boldsymbol{M}_k$ is the spatial (screw) inertia at k, $\boldsymbol{G} = [\boldsymbol{0}^\top\ \boldsymbol{a}_g^\top]^\top$ encodes gravity, $\mathrm{Ad}(\cdot)$ / $\mathrm{Ad}^*(\cdot)$ are adjoint/coadjoint maps on $SE(3)$, and $\mathrm{ad}^*_{(\cdot)}$ is the coadjoint operator.

For point α (with n_p computational points), the partial derivatives of the local inverse-dynamics contribution are

$$\frac{\partial ID_\alpha}{\partial \boldsymbol{q}} = \frac{\partial \boldsymbol{S}_\alpha^\top}{\partial \boldsymbol{q}}\, \boldsymbol{F}_\alpha^C + \boldsymbol{S}_\alpha^\top\Big(\boldsymbol{N}_\alpha^C \boldsymbol{R}_\alpha^B + \boldsymbol{M}_\alpha^C \boldsymbol{Q}_\alpha^B + \boldsymbol{U}_\alpha^S + \boldsymbol{P}_\alpha^S\Big), \tag{3-30}$$

$$\frac{\partial ID_\alpha}{\partial \dot{\boldsymbol{q}}} = \boldsymbol{S}_\alpha^\top\Big(\boldsymbol{N}_\alpha^C \boldsymbol{S}_\alpha^B + \boldsymbol{M}_\alpha^C \boldsymbol{Y}_\alpha^B + \boldsymbol{V}_\alpha^S\Big). \tag{3-31}$$

The total derivatives follow by summation:

$$\frac{\partial ID}{\partial \boldsymbol{q}} \;=\; \sum_{\alpha=1}^{n_p-1} \frac{\partial ID_\alpha}{\partial \boldsymbol{q}}, \qquad \frac{\partial ID}{\partial \dot{\boldsymbol{q}}} \;=\; \sum_{\alpha=1}^{n_p-1} \frac{\partial ID_\alpha}{\partial \dot{\boldsymbol{q}}}. \tag{3-32}$$

The partial derivative of the forward dynamics with respect to the input can be computed as follows:

$$\frac{\partial \boldsymbol{FD}}{\partial \boldsymbol{u}} = \boldsymbol{M}^{-1}\left(\frac{\partial \bar{\boldsymbol{\tau}}}{\partial \boldsymbol{u}}\right) = \boldsymbol{M}^{-1}\, \boldsymbol{A}(\boldsymbol{q}), \tag{3-33}$$

The recursive update formulas for the forward- and backward-pass quantities, together with the derivatives of the motion-subspace (projection) matrix $\boldsymbol{S}$, follow [105] and are briefly summarized in Appendix A.

Chapter 4

Results

This chapter evaluates the proposed NMPC/DDP framework on the Soft Inverted Pendulum (SIP) and two underactuated benchmarks (R-SIP and soft Furuta), across simulation and hardware studies. The *simulation* part examines: (i) nominal closed-loop behavior (regulation/swing-up), (ii) robustness to external disturbances and model mismatch, (iii) comparisons against baseline controllers (e.g., collocated and non-collocated PFL), and (iv) scalability and numerical aspects, including per-step solve time versus `ndof` and the impact of analytical versus numerical dynamics gradients. The *hardware* part reports inverse-kinematics reconstruction, system identification outcomes, and control experiments.

4-1 Validation Strategy

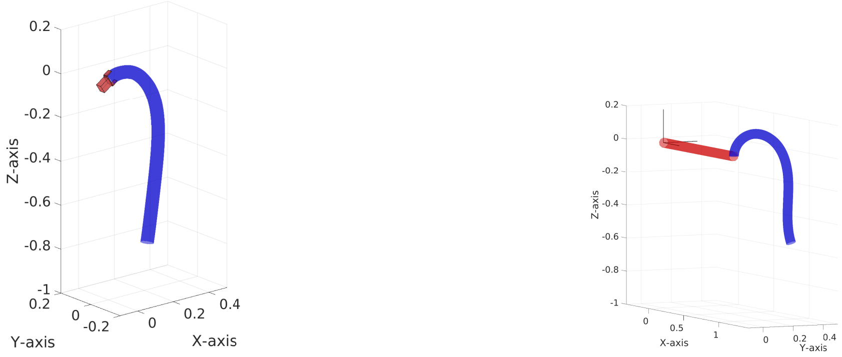
The proposed modeling and control framework is validated through numerical simulations. First, a nominal study verifies that the NMPC/DDP controller solves the swing up problem on the nominal plant with feasible runtimes. Next, robustness is assessed (i) to external disturbances by applying bounded forces at multiple locations along the length of the rod and (ii) to model mismatch by running the nominally tuned controller on plants with perturbed mass and stiffness parameters, thereby quantifying tolerance to mismatch. Finally, a comparative study evaluates NMPC in operating regimes where baseline strategies are known to work and in regimes where their assumptions fail; these regions are defined by the relative influence of gravitational and elastic effects as parameterized in (3-11) and (3-12).

For the comparison, the following compact scalars are used.

Elastic parameter.

$$\mathbf{K} = \int_0^L \Phi_\xi^\top \, \Sigma \, \Phi_\xi \, dX = \int_0^L \begin{bmatrix} 1 \\ X \end{bmatrix} EJ_y \begin{bmatrix} 1 & X \end{bmatrix} dX = EJ_y \int_0^L \begin{bmatrix} 1 & X \\ X & X^2 \end{bmatrix} dX.$$

$$\Rightarrow \quad \mathbf{K} = EJ_y \begin{bmatrix} L & \frac{L^2}{2} \\ \frac{L^2}{2} & \frac{L^3}{3} \end{bmatrix}$$



(a) R-Soft Inverted Pendulum (R-SIP). **(b)** Soft Furuta pendulum.

Figure 4-1: Soft underactuated benchmarks used for validation.

For $L = 1$,

$$\mathbf{K} = EJ_y \begin{bmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{3} \end{bmatrix} = k \begin{bmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{3} \end{bmatrix}, \qquad k := EJ_y.$$

Inertial parameter. The total mass of the rod is

$$m = \rho A L \xrightarrow{L=1} m = \rho A,$$

with A the cross-sectional area (for a circular rod $A = \pi r^2$).

Thus, k collects the bending material/geometry (E, J_y), while m collects inertia (ρ, A). These two scalars are used to index operating regimes and to report performance in the comparison section.

Beyond the SIP, the approach is validated on two underactuated soft systems: (i) an *R-Soft Inverted Pendulum (R-SIP)*, consisting of a base revolute joint with a soft link attached; its behavior parallels the SIP with a rigid base joint, and (ii) a *Soft Furuta pendulum*, a soft analogue of the rigid Furuta in which the horizontal driving link is rigid with a base revolute joint and a soft link attached at the distal end. A comparison of computation time versus the number of degrees of freedom (`ndof`) is also reported for the SIP model, together with the computation time difference between analytical versus numerical dynamics gradients.

For bending about the y-axis, the GVS strain basis is chosen :

$$\Phi_\xi(X) = \begin{bmatrix} \mathbf{0}_{1\times n} \\ \boldsymbol{\psi}(x)^\top \\ \mathbf{0}_{4\times n} \end{bmatrix}, \qquad \boldsymbol{\psi}(x) \in \mathbb{R}^n.$$

Two choices for $\boldsymbol{\psi}(X)$:

(i) Polynomial (monomial) basis

$$\boldsymbol{\psi}_{\text{poly}}(X) = \begin{bmatrix} 1 \\ X \\ X^2 \\ \vdots \\ X^{n-1} \end{bmatrix}.$$

(ii) Shifted Legendre basis on $[0,1]$ Let $\{p_k(X)\}_{k\geq 0}$ be the shifted Legendre polynomials, defined by

$$p_0(X) = 1, \qquad p_1(X) = 2X-1, \qquad p_{n+1}(X) = \frac{(2X-1)(2n+1)\,p_n(X) - n\,p_{n-1}(X)}{n+1} \quad (n \geq 1),$$

(equivalently, $p_n(X) = P_n(2X-1)$ with P_n the standard Legendre on $[-1,1]$). Then

$$\boldsymbol{\psi}_{\text{leg}}(X) = \begin{bmatrix} p_0(X) \\ p_1(X) \\ \vdots \\ p_{n-1}(X) \end{bmatrix}$$

4-2 Simulation Studies

This section evaluates the NMPC/DDP controller in simulation. Experiments use the SIP dynamics and cost from Section 3-5 with explicit–Euler discretization and short DDP iterations (1–3) per NMPC step with warm starts. Unless stated otherwise, the control applied is the time–varying feedback

$$\boldsymbol{u}_k^{\text{apply}} = \boldsymbol{u}_k^{\star} - \boldsymbol{K}_k(\boldsymbol{x}_k - \boldsymbol{x}_k^{\star}).$$

Performance is reported via: root–mean–square (RMS) and peak state error,

$$\text{RMS}(e) = \sqrt{\frac{1}{T}\sum_{k=1}^{T} \|\boldsymbol{x}_k - \boldsymbol{x}_g\|_2^2}, \qquad e_{\max} = \max_k \|\boldsymbol{x}_k - \boldsymbol{x}_g\|_\infty,$$

control effort $\sum_k \|\boldsymbol{u}_k\|_2^2$, settling time, and per–step solve time.

4-2-1 Baseline Performance: DDP on the Soft Inverted Pendulum

Using the physical model and NMPC parameters in Table 4-1, a 20 s NMPC run (step $h = 0.01$ s, 2000 steps) achieves precise swing-up and regulation on the SIP. As summarized in Table 4-2, terminal state errors are numerically zero at the solver tolerance, with a settling time of 1.70 s (5% band). Control remains moderate ($\|u\|_\infty \approx 9.77$, RMS 0.93), and the final cost vanishes. The average per-step solve time is 20.53 ms, consistent with short-iterate, warm-started DDP in NMPC. The accompanying plots (state and input evolution in Fig. 4-2, stroboscopic phase view in Fig. 4-3) illustrate rapid convergence to the upright equilibrium and bounded actuation throughout the maneuver.

Table 4-1: Baseline SIP model and OCP parameters for "Baseline Performance: DDP on the Soft Inverted Pendulum".

SIP model	
Length L	1 m
Radius r	0.03 m
Mass density ρ	10^3 kg m^{-3}
Material damping υ	10^4 Pa · s
Poisson's ratio ν	0.5
Young's modulus E	1×10^6 Pa
NMPC horizon & simulation	
Time step dt	0.01 s
Horizon length N	100 steps (1.00 s)
Total simulation T	2000 steps (20.00 s)
Cost weights	
Running state weight $\boldsymbol{W}$	$0.1\,\mathbf{I}_4$
Running control weight $\boldsymbol{R}$	diag(1)
Terminal state weight $\boldsymbol{W}_N$	diag(10^2, 10^2, 10^1, 10^1)
Initialization	
Initial state $\boldsymbol{x}_0$	$[8.5879,\ -11.5744,\ 0,\ 0]^\top$
Target state $\boldsymbol{x}_g$	$\mathbf{0}$

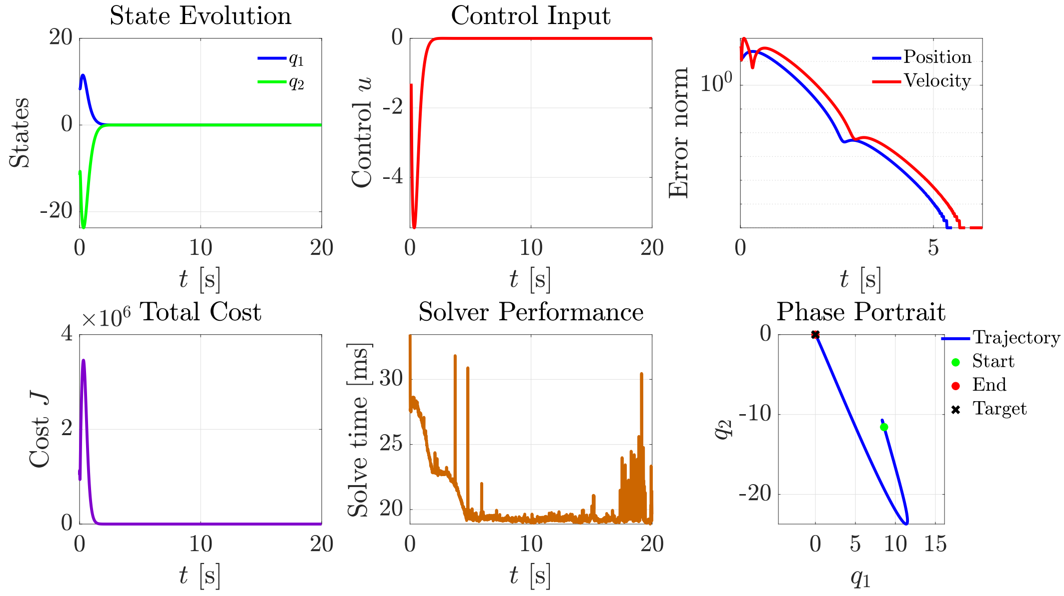


Figure 4-2: SIP nominal run under NMPC

Table 4-2: Nominal SIP simulation results (corresponding to the setup in Table 4-1).

Simulation duration	19.99 s
Number of time steps	2000
Time step	0.0100 s
Final position error	0.000000
Final velocity error	0.000000
Settling time (5% band)	1.70 s
Final cost	0.000
RMS control effort	0.931
Average solve time / step	20.53 ms

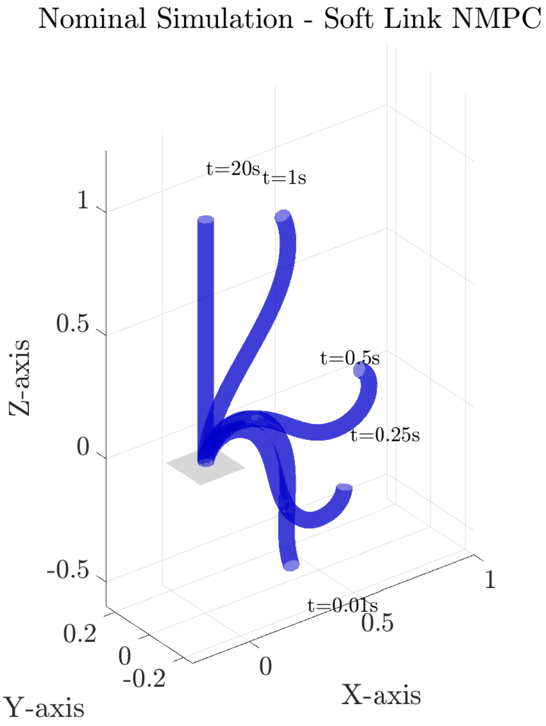


Figure 4-3: Stroboscopic plot of the SIP during swing-up and stabilization under NMPC.

Disturbance Rejection This section evaluates closed-loop robustness under exogenous wrenches applied at selected arclength grid points of the SIP (same setup as in Table 4-1). Ten cases are considered, varying disturbance *type* (force, moment, combined), *location* (base/middle/tip), *magnitude*, and *duration*. The disturbance program used in the simulations is summarized in Table 4-3. Disturbances are activated over the time window $[t_{\text{on}}, t_{\text{off}}]$ and applied at Gauss point index g_p of the soft link representing a point along the length of the soft link. A per-case performance table is provided in Table 4-4.

Table 4-3: Disturbance test cases (location by Gauss point g_p, type, magnitude, and activation window).

ID	Name	Location (g_p)	Type & Magnitude	Window [t_{on}, t_{off}] [s]
1	Small_Force_Tip	9	Force 3 N	[5.0, 5.5]
2	Medium_Force_Tip	9	Force 8 N	[5.0, 5.5]
3	Large_Force_Tip	9	Force 15 N	[5.0, 5.5]
4	Medium_Force_Middle	5	Force 8 N	[5.0, 5.5]
5	Medium_Force_Base	1	Force 8 N	[5.0, 5.5]
6	Moment_Tip	9	Moment 5 N · m	[5.0, 5.5]
7	Combined_Force_Moment_Tip	9	Force 6 N + Moment	[5.0, 5.5]
8	Long_Duration_Force	7	Force 10 N	[5.0, 6.0]
9	Multiple_Pulses	8	Force 12 N	[4.0, 4.2] (+ pulses)
10	Late_Disturbance	6	Force 8 N	[12.0, 12.5]

The disturbance–rejection results in Table 4-4, with state traces in Figures 4-5–4-6, show consistent robustness across all cases. Tip-applied forces induce the largest excursions (e.g., 8 N at the tip vs. base), while the same force at the base leads to much smaller deviations, reflecting leverage about the base joint. A mid-span application (case "Medium Force at Middle") produces the largest excursion among the 8 N cases, consistent with its effective moment arm. Recovery times are tightly clustered around $1.1-1.4$ s despite large transients (e.g., 15 N at the tip), and remain brief for long-duration forcing (0.82 s) once the input ceases. Multiple pulses extend recovery slightly due to repeated excitation. Control peaks scale with disturbance magnitude, while RMS effort increases with disturbance energy, yet final errors return to (numerical) zero in all cases and the overall settling time remains at 1.70 s, matching the nominal regime.

Table 4-4: Disturbance rejection summary under NMPC/DDP (showing only recovery time and peak control).

Case	RecTime [s]	MaxCtrl
Small Force at Tip (3N)	1.20	9.77
Medium Force at Tip (8N)	1.19	10.91
Large Force at Tip (15N)	1.38	15.22
Medium Force at Middle (8N)	1.15	9.77
Medium Force at Base (8N)	1.09	9.77
Moment at Tip (5Nm)	0.51	9.77
Combined Force+Moment	1.18	9.77
Long Duration (1s)	0.82	11.70
Multiple Pulses	1.26	9.77
Late Disturbance	1.18	10.10

RecTime: recovery time to 10% of the peak error after disturbance; MaxCtrl: $\max_t |u(t)|$.

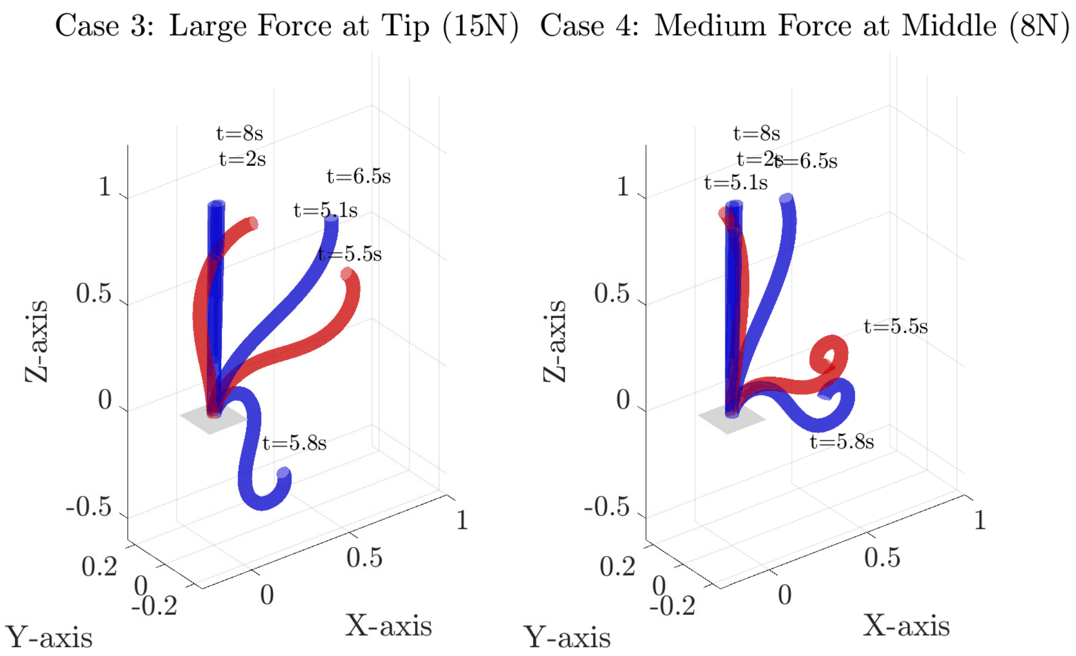


Figure 4-4: Stroboscopic plots for disturbance–rejection **Case 3** (Large Force at Tip, 15 N) and **Case 4** (Medium Force at Middle, 8 N). Red SIP segments denote the interval during which the disturbance is active.

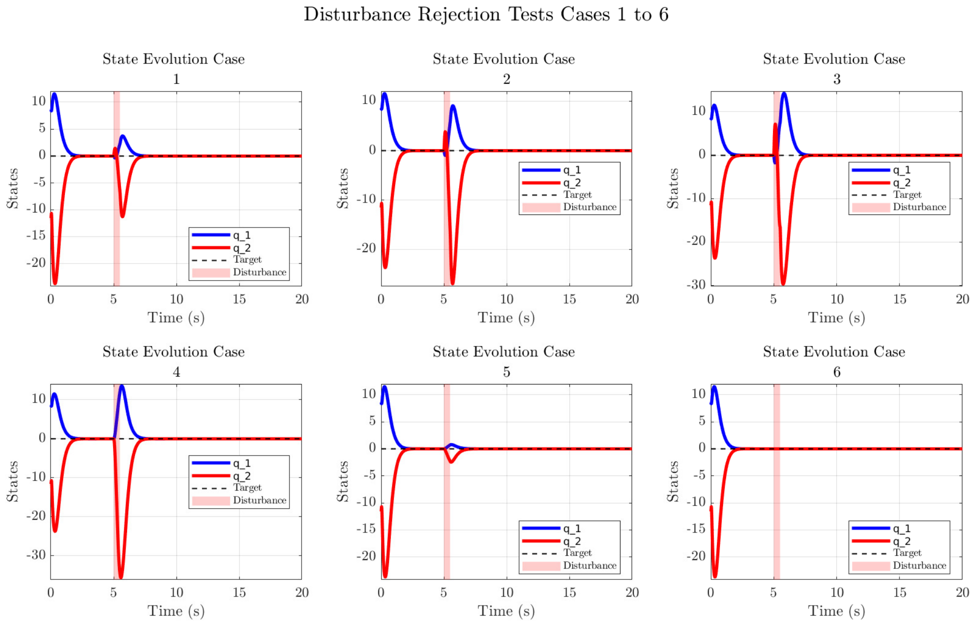


Figure 4-5: State evolution for disturbance-rejection cases 1–6.

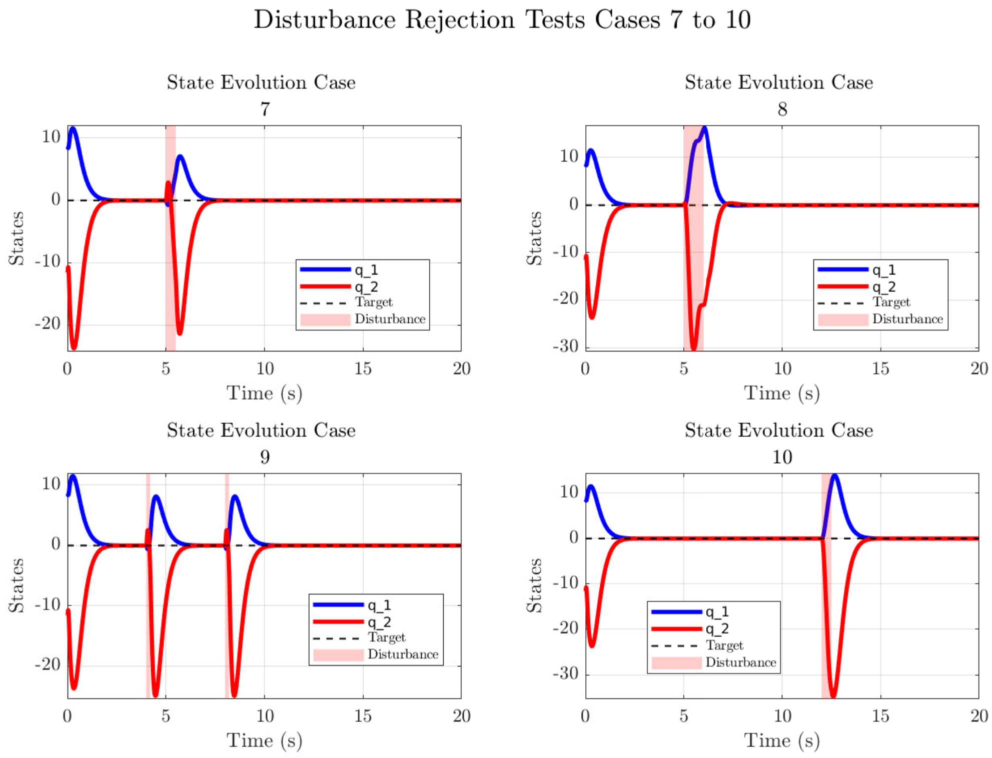


Figure 4-6: State evolution for disturbance-rejection cases 7–10.

Robustness to Model Mismatch The NMPC/DDP controller is kept *nominal* (parameters as in Table 4-1), and its closed-loop performance is evaluated on *plants* whose material properties are perturbed. The two perturbed parameters are the Young's modulus E and the mass density ρ. For each trial, one of $\{E, \rho\}$ is scaled while the other remains nominal. Let the plant parameters be

$$E_{\mathrm{p}} = (1 + \delta_E)\, E, \qquad \rho_{\mathrm{p}} = (1 + \delta_\rho)\, \rho,$$

with $\delta_E, \delta_\rho \in \{\pm 1\%, \pm 2\%, \pm 5\%, \pm 10\%, \pm 15\%, \pm 25\%, \pm 50\%\}$. Two one-parameter sweeps are considered: (i) E-sweep: $\delta_E \neq 0,\ \delta_\rho = 0$; (ii) ρ-sweep: $\delta_\rho \neq 0,\ \delta_E = 0$. For each perturbed plant, the static equilibrium $\boldsymbol{q}_{\mathrm{eq}}$ (zero velocity and input) is computed numerically and plotted against the nominal equilibrium. Figure 4-7 shows the distribution of equilibria over all (E, ρ) perturbations as well as the maximum deviation from the nominal case.

Figure 4-8 highlights a clear pattern consistent with the robustness table (Table 4-5): *mass decrease* ($\rho \downarrow$) and *stiffness increase* ($E \uparrow$) make the plant easier to regulate both settling time and RMS control effort remain small or even drop below nominal. Conversely, *mass increase* ($\rho \uparrow$) and *stiffness decrease* ($E \downarrow$) render the plant harder to control with the nominal controller, leading to longer settling times and larger RMS effort, with extreme cases approaching failure within the fixed horizon.

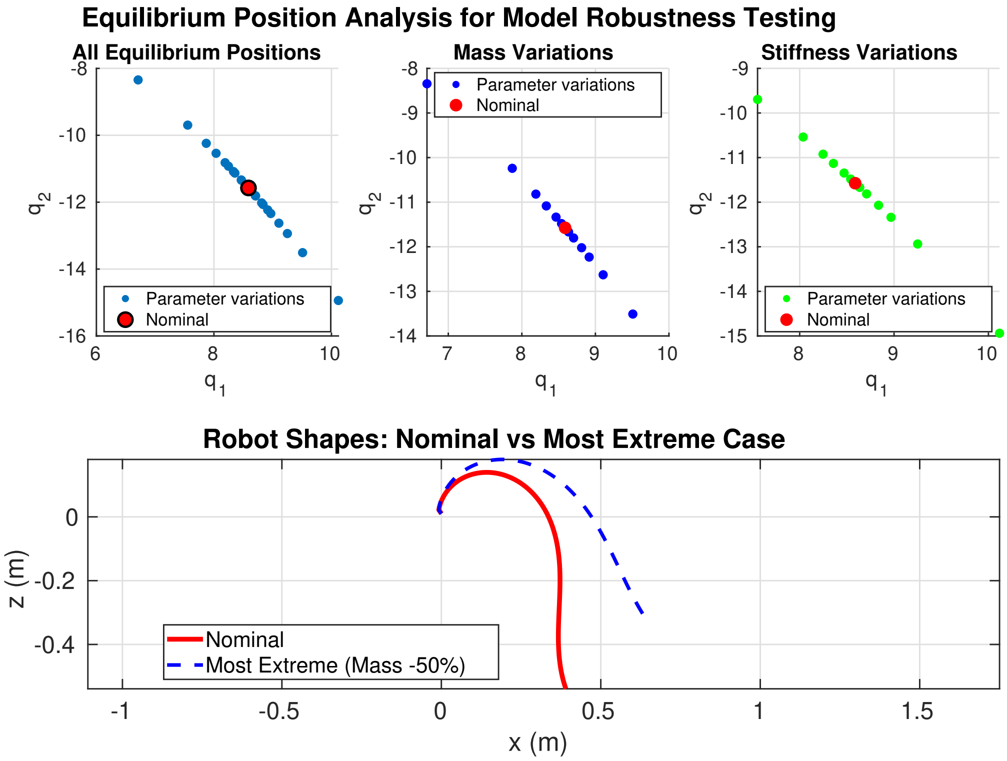


Figure 4-7: Equilibrium positions under parameter mismatch: scatter over all E and ρ perturbations; nominal equilibrium highlighted. The maximum deviation from nominal is indicated in the plot.

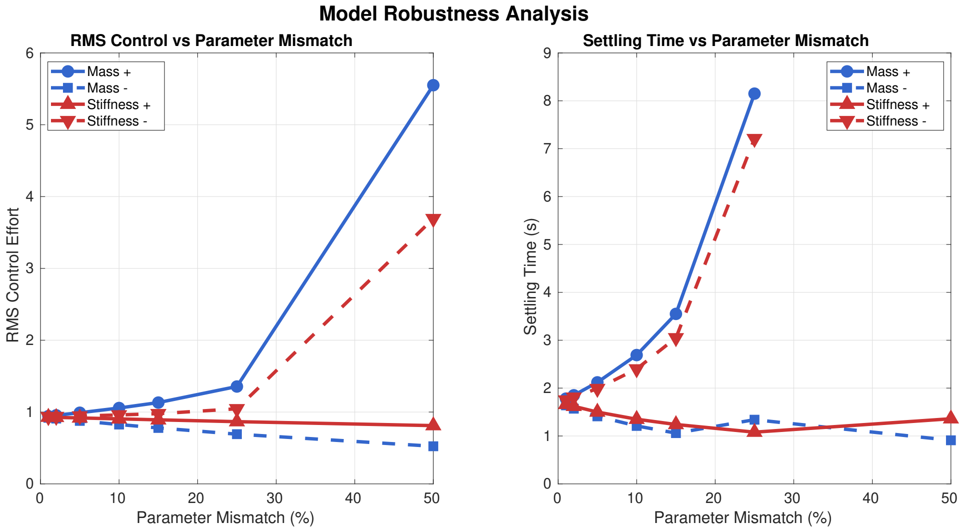


Figure 4-8: Effect of model–plant mismatch on settling time and RMS control effort. Categories on the abscissa: mass increase/decrease and stiffness (Young's modulus) increase/decrease. NMPC uses the nominal model (Table 4-1); plants vary ρ or E.

Table 4-5: Model robustness to parameter mismatch (controller uses nominal model). Left: mass density ρ variations; Right: Young's modulus E variations. Success $\equiv$ finite settling time within horizon.

(a) Mass density ρ

Case	RMSCtrl	Settle [s]	Success
Nominal (no error)	0.931	1.70	Success
Mass +1%	0.943	1.78	Success
Mass +2%	0.954	1.85	Success
Mass +5%	0.991	2.12	Success
Mass +10%	1.056	2.69	Success
Mass +15%	1.132	3.55	Success
Mass +25%	1.356	8.15	Success
Mass +50%	5.551	NaN	Fail
Mass −1%	0.920	1.64	Success
Mass −2%	0.909	1.57	Success
Mass −5%	0.877	1.41	Success
Mass −10%	0.826	1.21	Success
Mass −15%	0.779	1.06	Success
Mass −25%	0.692	1.34	Success
Mass −50%	0.524	0.91	Success

(b) Young's modulus E

Case	RMSCtrl	Settle [s]	Success
Stiffness +1%	0.929	1.66	Success
Stiffness +2%	0.926	1.62	Success
Stiffness +5%	0.918	1.50	Success
Stiffness +10%	0.905	1.35	Success
Stiffness +15%	0.892	1.24	Success
Stiffness +25%	0.866	1.08	Success
Stiffness +50%	0.811	1.36	Success
Stiffness −1%	0.934	1.75	Success
Stiffness −2%	0.937	1.81	Success
Stiffness −5%	0.945	1.99	Success
Stiffness −10%	0.960	2.40	Success
Stiffness −15%	0.977	3.05	Success
Stiffness −25%	1.045	7.21	Success
Stiffness −50%	3.693	NaN	Fail

4-2-2 Comparison with Partial Feedback Linearization (PFL) Controllers

This section compares the NMPC/DDP controller against two PFL baselines: the *collocated* PFL in (3-13) and the *non-collocated* PFL in (3-14). The objective is to demonstrate that NMPC can stabilize operating points in regions where these PFL controllers are not feasible.

The collocated PFL feasibility is characterized by the open-loop stability condition (3-11) and the zero-dynamics stability condition (3-12). The former demarcates regimes where the upright equilibrium becomes unstable (very high stiffness must exceed this threshold), while the latter specifies where the zero dynamics of the collocated output are stable. These two conditions define the *collocated PFL region* used in the feasibility plots.

The plant is swept over a 2-D grid of mass and stiffness values (m, k) previously defined:

$$m \in \{1, \dots, 9\}, \qquad \sigma := \log_{10}(1/k) \in \{0, 1, \dots, 100\},$$

to emphasize low-stiffness regimes.

Figure 4-9 summarizes control feasibility across mass–stiffness pairs (m, k). The plot is segmented into regions where open-loop behavior is stable and where feedback is required; within the latter, the analytically derived bounds (3-11) (blue curve) and (3-12) (red curve) are overlaid. The band between these two curves is the *collocated PFL feasible region* implied by (3-13) together with the zero-dynamics requirement. As m increases, both thresholds shift downward, enlarging the band of admissible controller choices. Outside this band, non-collocated PFL (3-14) may still succeed, and the NMPC/DDP controller is evaluated precisely in those more challenging regimes.

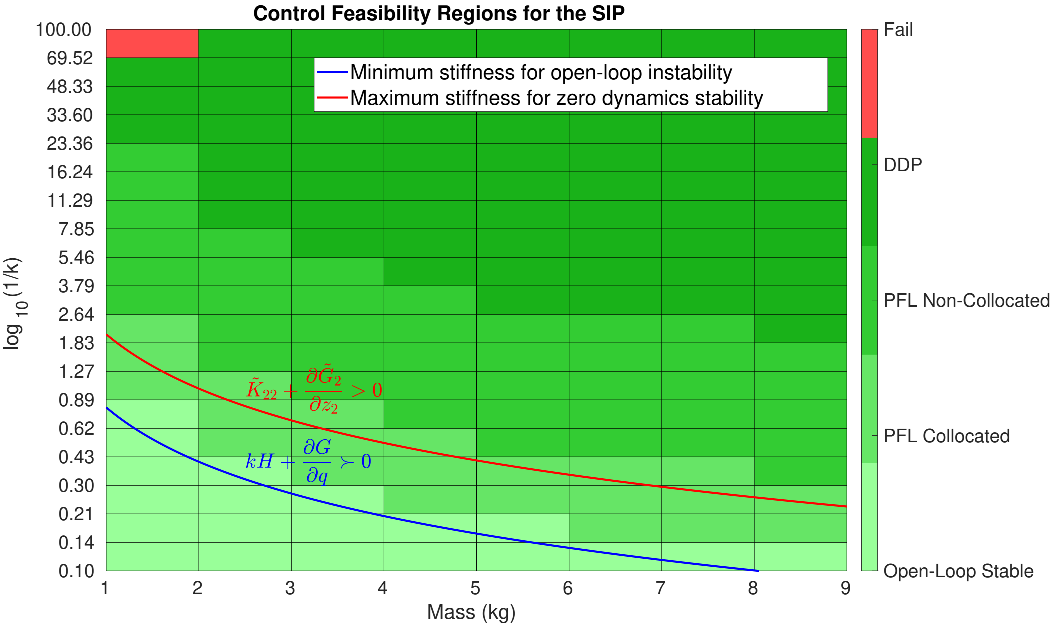


Figure 4-9: Feasibility regions over (m, k): open-loop stable , with analytic bounds from (3-11) (blue) and (3-12) (red). The collocated PFL band lies strictly between the two curves; non-collocated PFL and NMPC/DDP are evaluated outside/near these limits.

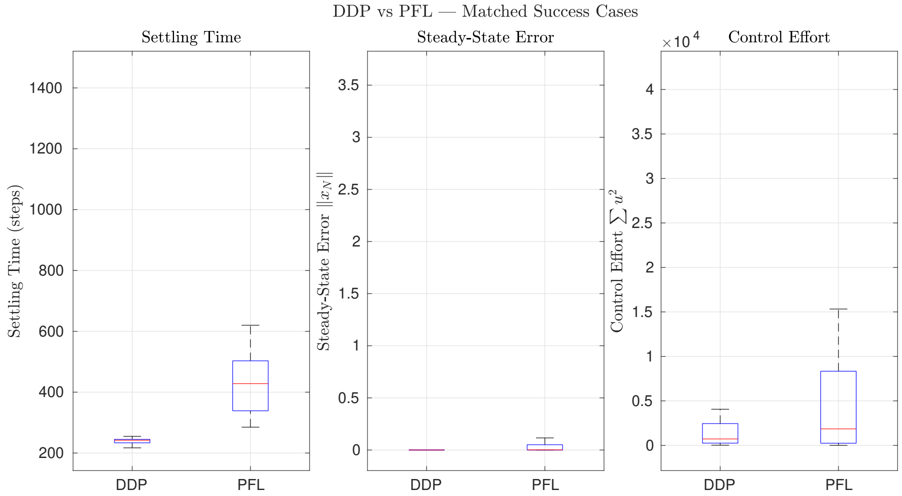


Figure 4-10: Collocated PFL ((3-13)): performance versus (m, k). Errors and effort are well behaved inside the feasibility band ((3-11)–(3-12)).

Figures 4-10–4-11 confirm the theoretical picture: collocated PFL performs well only inside the band delimited by (3-11) and (3-12), while non-collocated PFL stabilizes a broader slice of the plane at the cost of higher effort. In the subsequent plots, NMPC/DDP maintains regulation even beyond both PFL regions, particularly in low-stiffness/high-mass settings where the zero dynamics are unfavorable, illustrating its advantage in regimes that violate the PFL feasibility conditions.

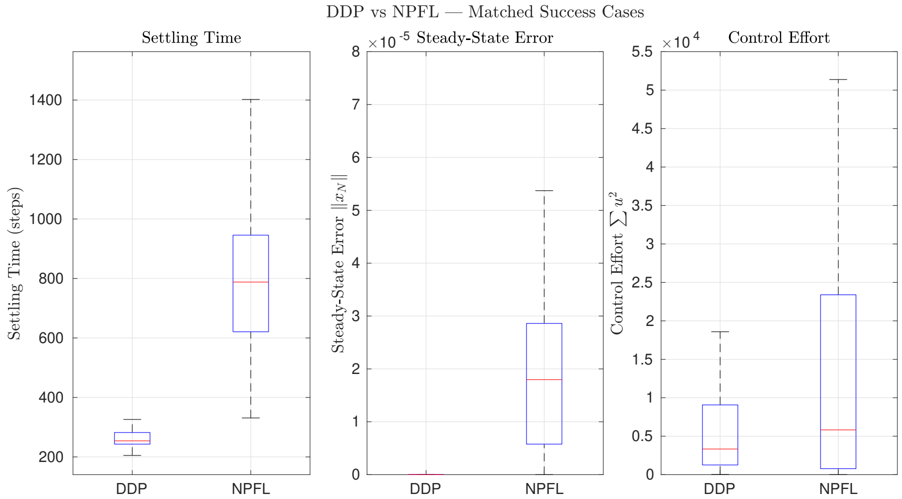
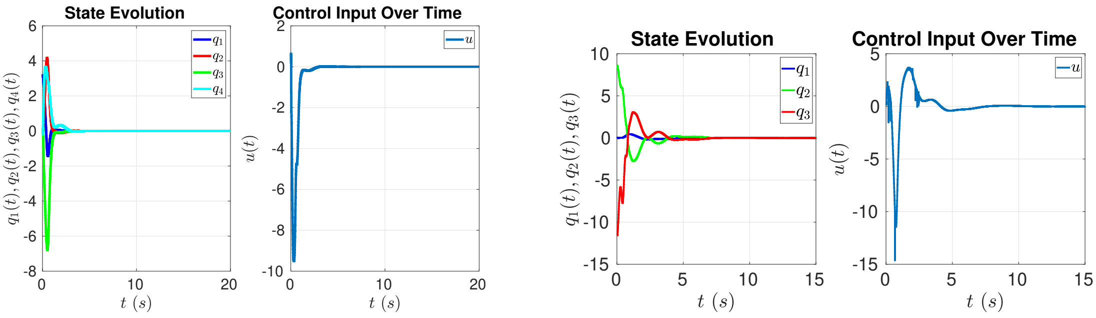


Figure 4-11: Non-collocated PFL ((3-14)).

4-2-3 Scalability and Generalization

Other Soft Underactuated Systems The NMPC/DDP framework transfers directly to other soft, underactuated robots without structural changes to the solver or cost design. Only the dynamics module and its analytical derivatives (GVS-based) are swapped to match the target system. The results below illustrate stabilization on two distinct platforms, confirming that the NMPC yields fast convergence and bounded effort beyond the SIP.

The **R-SIP** consists of a rigid revolute base joint driving a single soft link modeled with the GVS formulation. The soft segment is parameterized by a *second-order shifted Legendre basis* for the excited bending strain, i.e., $\boldsymbol{\psi}(x) = [P_0(x),\, P_1(x),\, P_2(x)]^\top$, which provides a low-order yet expressive shape space for swing-up and regulation. The **Soft Furuta** comprises a planar rigid driving arm with a distal soft appendage; its hybrid soft–rigid dynamics are likewise captured within the same GVS approach.

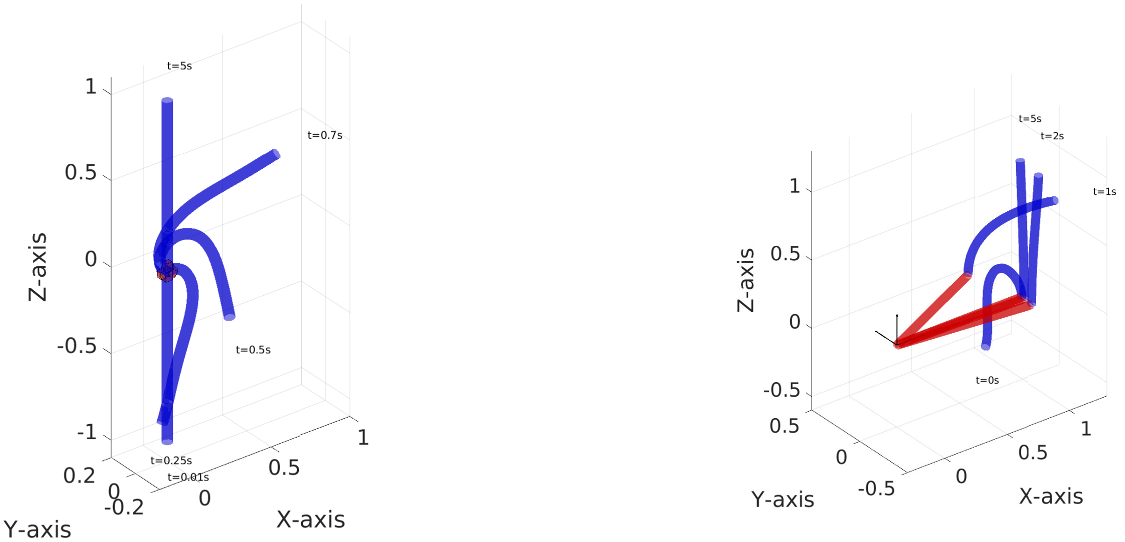


(a) R-SIP: states vs. time.

(b) Soft Furuta: states vs. time.

Figure 4-12: State trajectories: R-SIP (left) and Soft Furuta (right).

Figure 4-12 reports state trajectories for both systems. In each case, the controller performs swing-up and stabilizes the upright equilibrium with smooth transients and bounded input, despite low stiffness and underactuation. The complementary stroboscopic views in Fig. 4-13 show rapid convergence of the phase portrait to the upright fixed point.



(a) R-SIP stroboscopic plot.

(b) Soft Furuta stroboscopic plot.

Figure 4-13: Stroboscopic comparisons: R-SIP (left) and Soft Furuta (right).

Real-time Feasibility vs Degrees of Freedom Average per–step solve time grows moderately with the number of shape DOFs (ndof), while accuracy and closed-loop success remain high across all cases . The 2-DOF model attains the fastest runtime. Figure 4-14 show that average per–step solve time grows moderately with the number of shape DOFs: 7.05 ms (ndof= 2), 17.19 ms (ndof= 3), 25.53 ms (ndof= 4), and 27.61 ms (ndof= 5). The NMPC loop therefore sustains control rates from ∼140 Hz (ndof 2) down to ∼36–40 Hz (ndof 5).

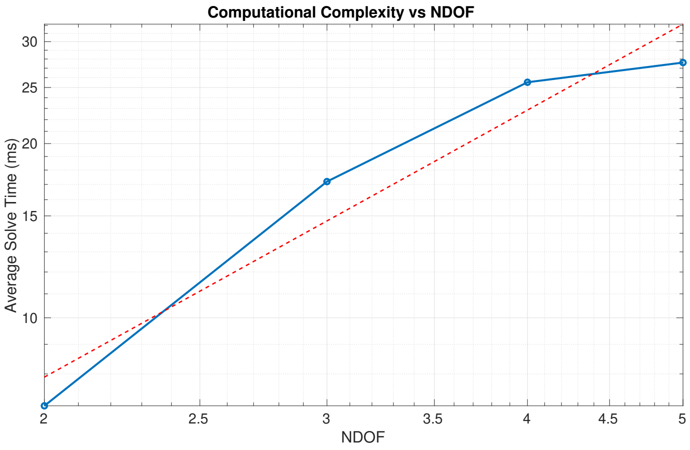


Figure 4-14: Average solve time per NMPC step versus number of degrees of freedom (ndof).

Analytical vs Numerical Gradients Analytical dynamics derivatives are decisively faster than numerical finite differences in the NMPC/DDP setupabout 4.4× lower per–step solve time (roughly 10.2 ms vs. 45.2 ms), enabling higher control rates (98 Hz vs. 22 Hz) with comparable final accuracy.

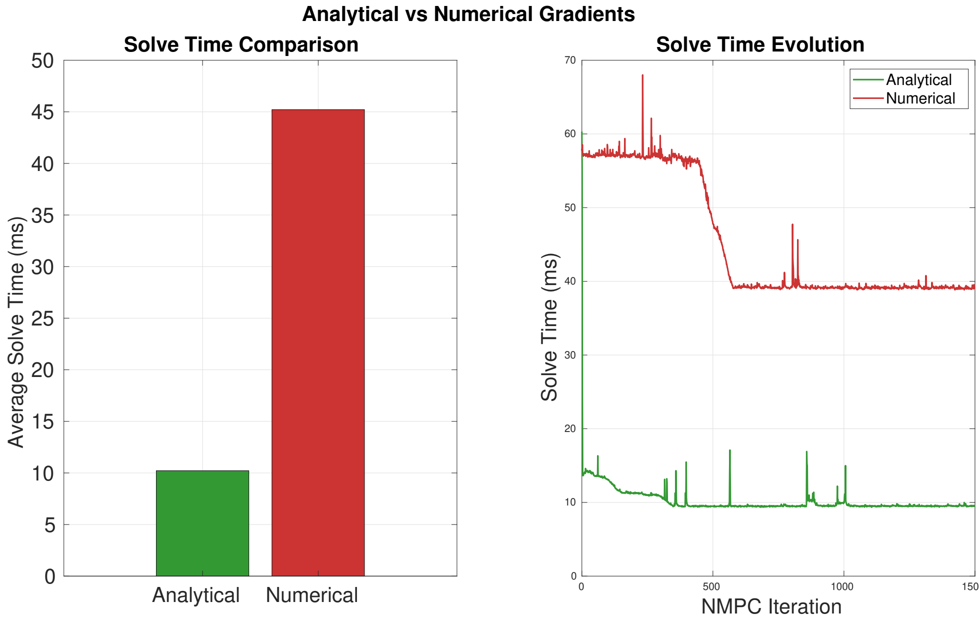


Figure 4-15: Analytical vs. numerical (finite-difference) dynamics gradients in NMPC/DDP. Analytical derivatives yield lower per-step solve time.

4-3 Hardware Setup

This section outlines the experimental platform used to validate the control framework for deformable-object manipulation. The setup consists of a 7-DoF Franka Emika Panda arm grasping a deformable cable at a single point along its length. The cable–gripper pair is modeled as a Revolute–Soft Inverted Pendulum (R-SIP): a rigid revolute joint at the gripper followed by a soft link representing the cable. The control objective is a swing-up from the downward stable equilibrium to the upright unstable equilibrium. A multi-camera OptiTrack motion-capture system provides state estimates of the cable, which are used by a closed-loop NMPC controller that commands the revolute attachment joint through the Panda. Implementation details are provided in the subsequent subsections: Inverse Kinematics, System Identification, and Control Framework.

4-3-1 Inverse Kinematics

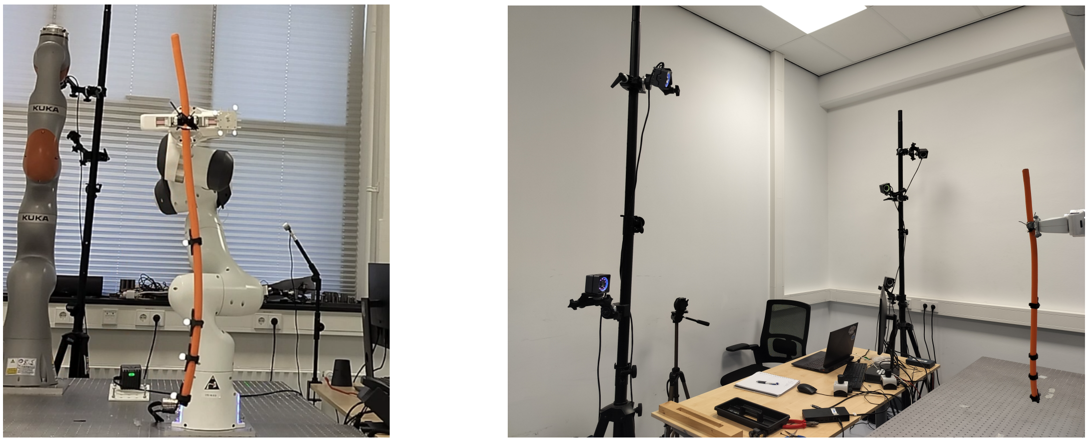
This section formulates how the inverse kinematics problem is solved using the measurements from the OptiTrack system.

OptiTrack measurements. The motion-capture system provides:

$$\text{Marker positions in the global frame } \{\mathcal{G}\}: \quad \boldsymbol{m}_j^{\mathcal{G}} \in \mathbb{R}^3, \qquad j = 1, \dots, 4, \tag{4-1}$$

$$\text{Rigid-body pose at the attachment joint:} \quad \boldsymbol{g}_{\mathrm{RB}}^{\mathcal{G}} = \begin{bmatrix} \boldsymbol{R}_{\mathrm{RB}}^{\mathcal{G}} & \boldsymbol{p}_{\mathrm{RB}}^{\mathcal{G}} \\ \boldsymbol{0}^\top & 1 \end{bmatrix} \in SE(3), \tag{4-2}$$

where $\boldsymbol{R}_{\mathrm{RB}}^{\mathcal{G}} \in SO(3)$ and $\boldsymbol{p}_{\mathrm{RB}}^{\mathcal{G}} \in \mathbb{R}^3$ denote orientation and position, respectively.



(a) Franka Emika Panda with deformable cable.

(b) Optical tracking setup (OptiTrack) to record marker positions

Figure 4-16: Hardware setup used in the experiments.

Marker position from a fixed local offset (with base calibration). Let $\boldsymbol{g}_{\mathrm{RB}}^{\mathcal{G}} \in SE(3)$ be the measured rigid–body pose at the attachment, and let $\boldsymbol{g}_{\mathcal{B}}^{\mathrm{RB}} \in SE(3)$ be the fixed calibration from the rigid–body marker to the SIP base. The base pose is

$$\boldsymbol{g}_{\mathcal{B}}^{\mathcal{G}} \;=\; \boldsymbol{g}_{\mathrm{RB}}^{\mathcal{G}}\, \boldsymbol{g}_{\mathcal{B}}^{\mathrm{RB}}.$$

For a given shape $\boldsymbol{q}$, the GVS forward kinematics in the base frame is $\boldsymbol{g}_{\mathcal{F}}^{\mathcal{B}}(X_j, \boldsymbol{q}) \in SE(3)$; hence the centerline pose in the global frame is

$$\boldsymbol{g}_{\mathcal{F}}^{\mathcal{G}}(X_j, \boldsymbol{q}) \;=\; \boldsymbol{g}_{\mathcal{B}}^{\mathcal{G}}\, \boldsymbol{g}_{\mathcal{F}}^{\mathcal{B}}(X_j, \boldsymbol{q}).$$

Assume each marker is located at the same fixed offset $\boldsymbol{r}_M \in \mathbb{R}^3$ from the centerline frame. In homogeneous coordinates $\bar{\boldsymbol{r}}_M := [\,\boldsymbol{r}_M^\top \;\; 1\,]^\top$, the predicted global position of marker j (placed at arclength X_j) is

$$\hat{\boldsymbol{m}}_j^{\mathcal{G}}(\boldsymbol{q}) \;=\; \Big[\boldsymbol{g}_{\mathcal{F}}^{\mathcal{G}}(X_j, \boldsymbol{q})\, \bar{\boldsymbol{r}}_M\Big]_{1:3} \;=\; \Big[\boldsymbol{g}_{\mathcal{B}}^{\mathcal{G}}\, \boldsymbol{g}_{\mathcal{F}}^{\mathcal{B}}(X_j, \boldsymbol{q})\, \bar{\boldsymbol{r}}_M\Big]_{1:3}, \qquad j = 1, \dots, 4,$$

where $[\cdot]_{1:3}$ extracts the translational part (entries 1–3 of the resulting 4-vector).

Inverse-kinematics estimation (least squares). Define residuals between predicted and measured marker positions

$$\boldsymbol{r}_j(\boldsymbol{q}) \;:=\; \hat{\boldsymbol{m}}_j^{\mathcal{G}}(\boldsymbol{q}) \;-\; \boldsymbol{m}_j^{\mathcal{G}} \in \mathbb{R}^3, \qquad j = 1, \dots, 4,$$

and stack them as $\boldsymbol{r}(\boldsymbol{q}) = [\boldsymbol{r}_1^\top, \ldots, \boldsymbol{r}_4^\top]^\top \in \mathbb{R}^{12}$. The IK problem is the unweighted nonlinear least-squares

$$\boldsymbol{q}^\star \;=\; \arg\min_{\boldsymbol{q}\in\mathbb{R}^n} \;\frac{1}{2}\sum_{j=1}^{4} \left\|\boldsymbol{r}_j(\boldsymbol{q})\right\|_2^2 \;=\; \arg\min_{\boldsymbol{q}} \;\frac{1}{2}\left\|\boldsymbol{r}(\boldsymbol{q})\right\|_2^2. \tag{4-3}$$

Let $\boldsymbol{J}(\boldsymbol{q}) := \dfrac{\partial \boldsymbol{r}(\boldsymbol{q})}{\partial \boldsymbol{q}} \in \mathbb{R}^{12\times n}$ be the stacked Jacobian (with $3 \times n$ blocks $\boldsymbol{J}_j(\boldsymbol{q}) = \partial \hat{\boldsymbol{m}}_j^{\mathcal{G}} / \partial \boldsymbol{q}$). The gradient of Φ is

$$\nabla\Phi(\boldsymbol{q}) \;=\; \boldsymbol{J}(\boldsymbol{q})^\top \boldsymbol{r}(\boldsymbol{q}).$$

A (unconstrained) gradient-descent solver updates

$$\boldsymbol{q}^{(k+1)} \;=\; \boldsymbol{q}^{(k)} \;-\; \alpha\,\nabla\Phi(\boldsymbol{q}^{(k)}) \;=\; \boldsymbol{q}^{(k)} \;-\; \alpha\,\boldsymbol{J}(\boldsymbol{q}^{(k)})^\top\,\boldsymbol{r}(\boldsymbol{q}^{(k)}),$$

where $\alpha > 0$ is a fixed stepsize. A practical stopping criterion is $\|\nabla\Phi(\boldsymbol{q}^{(k)})\|_\infty \le \varepsilon$ or $\|\boldsymbol{r}(\boldsymbol{q}^{(k)})\|_2 \le \varepsilon$ for a tolerance $\varepsilon > 0$. The analytical Jacobians from the GVS forward kinematics yield a closed-form Jacobian for the IK least-squares problem; the derivation and final expressions are provided in Appendix B.

4-3-2 System Identification

Building on the IK formulation above (same OptiTrack markers and base-pose calibration), the goal is to identify the material stiffness E of the soft link from a static (downward) configuration. All geometric and inertial quantities (L, r, $\rho, \ldots$) are taken as measured; only E is estimated.

Static equilibrium for a given E. At rest ($\dot{\boldsymbol{q}} = \boldsymbol{0}$, $\ddot{\boldsymbol{q}} = \boldsymbol{0}$), the GVS dynamics reduce to

$$\boldsymbol{F}_{\mathrm{eq}}(\boldsymbol{q}; E) \;:=\; \boldsymbol{K}(E)\,\boldsymbol{q} + \boldsymbol{G}(\boldsymbol{q}) \;=\; \boldsymbol{0}. \tag{4-4}$$

For a candidate E, the static shape is obtained by

$$\boldsymbol{q}_{\mathrm{eq}}(E) \;=\; \arg\min_{\boldsymbol{q}} \;\tfrac{1}{2}\left\|\boldsymbol{K}(E)\boldsymbol{q} + \boldsymbol{G}(\boldsymbol{q})\right\|_2^2, \tag{4-5}$$

solved numerically (e.g., `fmincon/fsolve`) with the same base pose and FK as in the IK section.

Marker cost. Using $\boldsymbol{q}_{\mathrm{eq}}(E)$, the predicted marker positions $\hat{\boldsymbol{m}}_j^{\mathcal{G}}(\boldsymbol{q}_{\mathrm{eq}}(E))$ are computed exactly as in the IK mapping. The residuals $\boldsymbol{r}_j = \hat{\boldsymbol{m}}_j^{\mathcal{G}} - \boldsymbol{m}_j^{\mathcal{G}}$ define the outer objective

$$J(E) \;=\; \tfrac{1}{2}\sum_{j=1}^{4} \|\hat{\boldsymbol{m}}_j^{\mathcal{G}}(\boldsymbol{q}_{\mathrm{eq}}(E)) - \boldsymbol{m}_j^{\mathcal{G}}\|_2^2, \tag{4-6}$$

and the stiffness estimate is

$$E^\star \;=\; \arg\min_{E\in[E_{\min},\,E_{\max}]} \;J(E), \tag{4-7}$$

solved with a 1D constrained optimizer (e.g., `fmincon` on $E > 0$).

4-3-3 Control Framework

The proposed NMPC framework is evaluated on a setup where a Franka Emika Panda manipulates a deformable cable attached at the last joint. The primary objective is to assess real-time feasibility on hardware. In the current iteration, two practical limitations prevented deploying the full closed-loop NMPC on the robot: (i) limited integration time for on-board implementation, and (ii) a velocity bound on the last Franka joint that would be frequently saturated by the aggressive inputs produced during swing-up to the upright equilibrium.

To nevertheless test the core control idea on the physical system, trajectories are generated *offline* using the same NMPC formulation and the RSIP model. Instead of the upright equilibrium, two lateral target equilibria are selected one on each side of the plane chosen to be challenging yet more attainable than the upright configuration. For each target, the NMPC computes an optimal state–control sequence that performs swing-up, local stabilization and swithcing to the other equilibrium so that the cable is swung to the first equilibrium, stabilized, and subsequently redirected to the second equilibrium.

Because the NMPC outputs an actuator torque signal, while the Franka is operated in a safety-minded *joint impedance* mode, execution proceeds via a reference-tracking layer: the optimized actuator coordinate trajectory (the joint to which the cable is attached) is extracted from the NMPC solution and used as the reference to the Franka's low-level joint controller. To satisfy the joint-speed limit, the reference is uniformly retimed (time-scaled) so that the commanded joint velocity remains within bounds, after which the joint-impedance loop tracks the reference.

In summary, the hardware validation consists of: (1) offline NMPC trajectory generation to lateral equilibria, (2) time-scaling of the actuator reference to respect joint limits, and (3) safe execution via the Franka's joint-impedance controller.

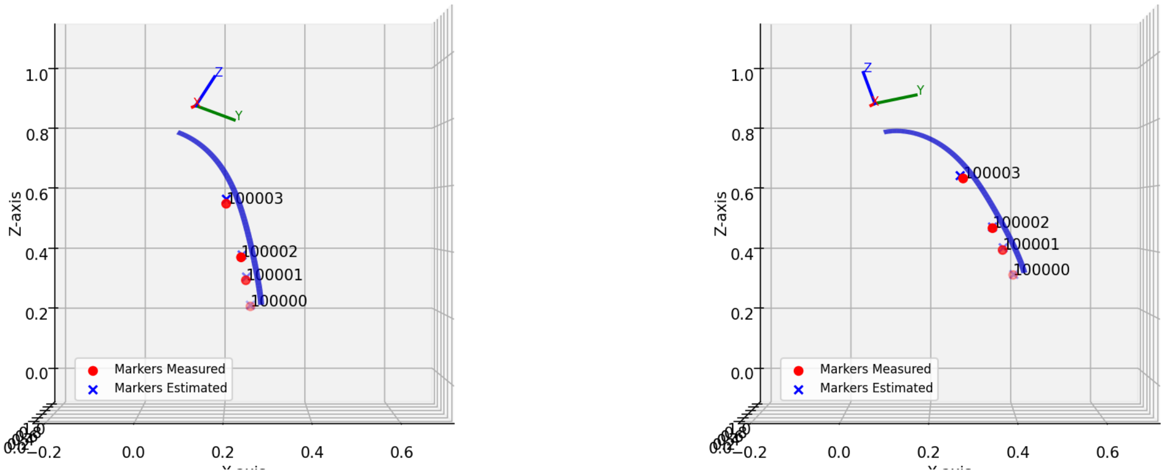
4-4 Implementation Details

The control framework was first prototyped and validated in simulation in MATLAB, where the DDP/NMPC pipeline, cost shaping, and model verification were developed. For real-time execution on hardware, the controller and the dynamics were reimplemented in C++ using the *Crocoddyl* library [110] for shooting-based optimal control and *Eigen* for linear algebra. The same GVS soft-link model was used in both stacks to ensure consistency. All nominal, disturbance-rejection, and mismatch studies were run in MATLAB with vectorized dynamics and cost routines. Numerical differentiation was used initially for debugging, then replaced by analytical derivatives to match the C++ pipeline.

4-5 Hardware Results

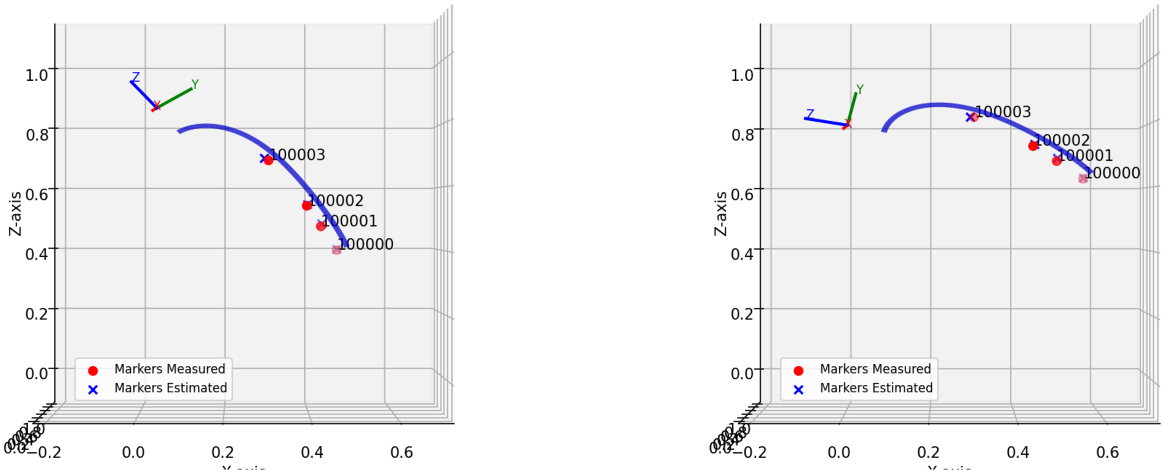
4-5-1 Inverse Kinematics

The IK formulation and mapping from OptiTrack measurements to shape coordinates were detailed in the previous chapter; here the goal is simply to verify that the pipeline works on hardware data. As shown in Fig. 4-17, the predicted marker locations from the GVS forward kinematics (using the identified base pose and fixed local offsets) align closely with the measured markers across multiple trials. Qualitatively, the overlays exhibit negligible reprojection error and no systematic bias along the rod, indicating that the base calibration and centerline-to-marker transformations are consistent. Convergence is reliable from nominal initializations, and the recovered shapes are physically plausible for the static poses considered. In summary the IK pipeline behaves as expected on real measurements and is suitable for downstream tasks (e.g., NMPC state estimation and stiffness identification). The IK solver exploits *analytical* Jacobians from the GVS kinematics (Appendix B), so each iteration reduces to lightweight matrix-vector operations without finite-difference passes. This markedly lowers computational overhead and improves numerical robustness, making the IK pipeline suitable for real-time use alongside the NMPC controller.



(a) IK solution (trial 1).

(b) IK solution (trial 2).

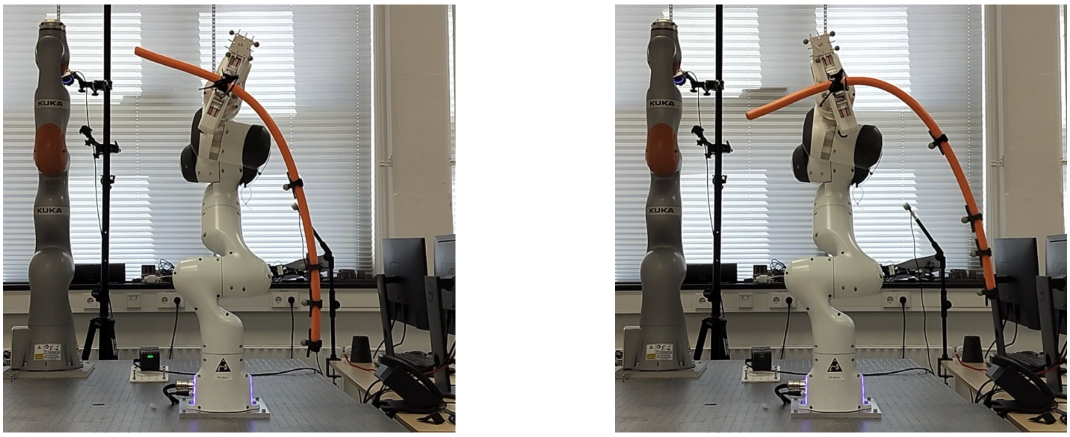


(c) IK solution (trial 3).

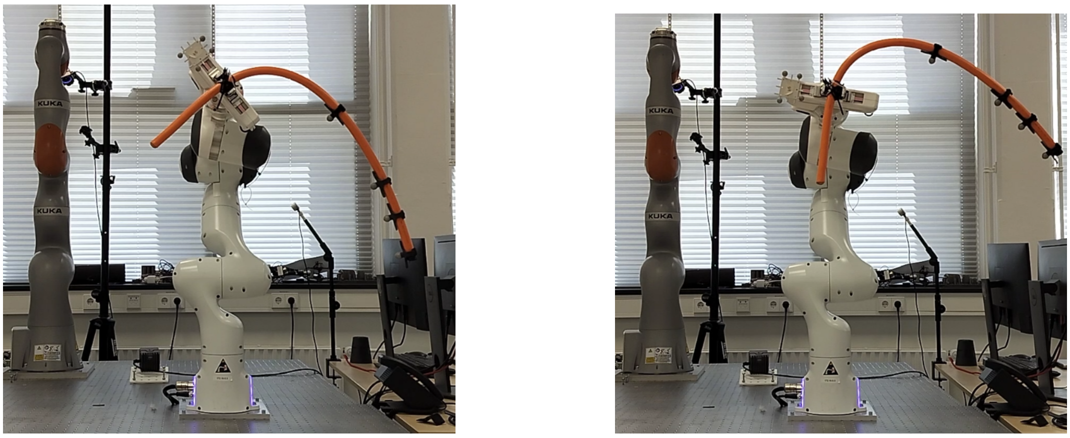
(d) IK solution (trial 4).

Figure 4-17: Inverse-kinematics reconstructions from OptiTrack measurements. Each panel shows the IK-estimated shape from the measured rigid-body pose and marker positions (Section 4-3-1).

Figure 4-18 shows the *real* snapshots (poses 1–4) of the Franka–cable setup, while Figure 4-17 shows the *IK reconstructions* for the same poses obtained from the OptiTrack rigid-body pose and marker measurements (using the same base calibration and fixed marker offsets as in Section 4-3-1). The reconstructed centerline and predicted marker locations visually match the images, indicating small error and confirming the fidelity of the IK pipeline on hardware data.



(a) Pose 1. **(b)** Pose 2.



(c) Pose 3. **(d)** Pose 4.

Figure 4-18: Franka Emika Panda manipulating a deformable cable (experimental snapshots).

4-5-2 System Identification

The estimation procedure is already detailed in Section 4-3-2; here only the outcomes are reported. All geometric/inertial quantities are measured (or derived), while the material stiffness E is identified by minimizing the marker reprojection error in the static configuration. The cable is highly damped; a sufficiently large material damping is *fixed* to match the observed decay and is not identified.

Table 4-6: Static identification summary (measured vs. identified).

Quantity	Value	Note
Length L	$0.6\,\mathrm{m}$	measured
Radius r	$0.009\,\mathrm{m}$	measured
Mass density ρ	$3546.6\ \mathrm{kg\,m^{-3}}$	measured
Material Damping	2×10^6	matched to decay
Young's modulus $E^\star$	$2.2069 \times 10^7\,\mathrm{Pa}$	Identified

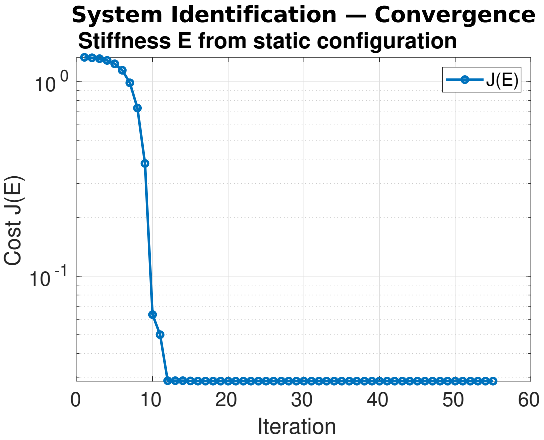


Figure 4-19: Cost convergence during stiffness identification.

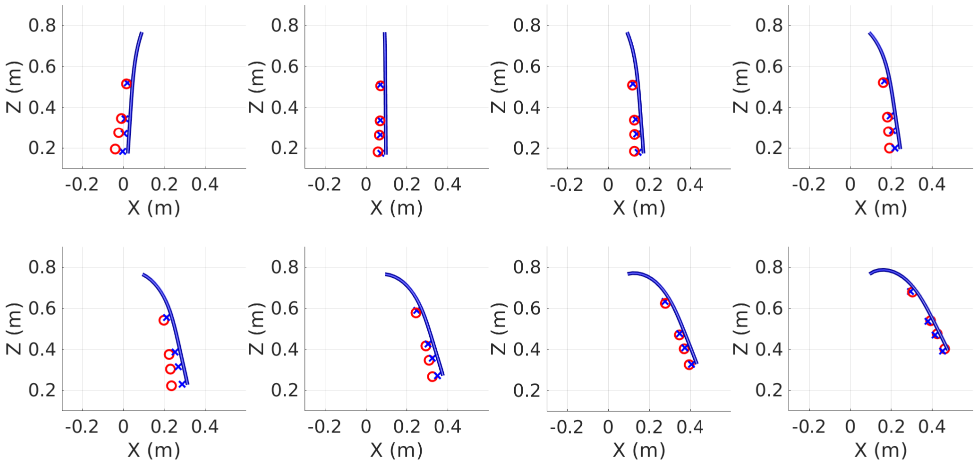
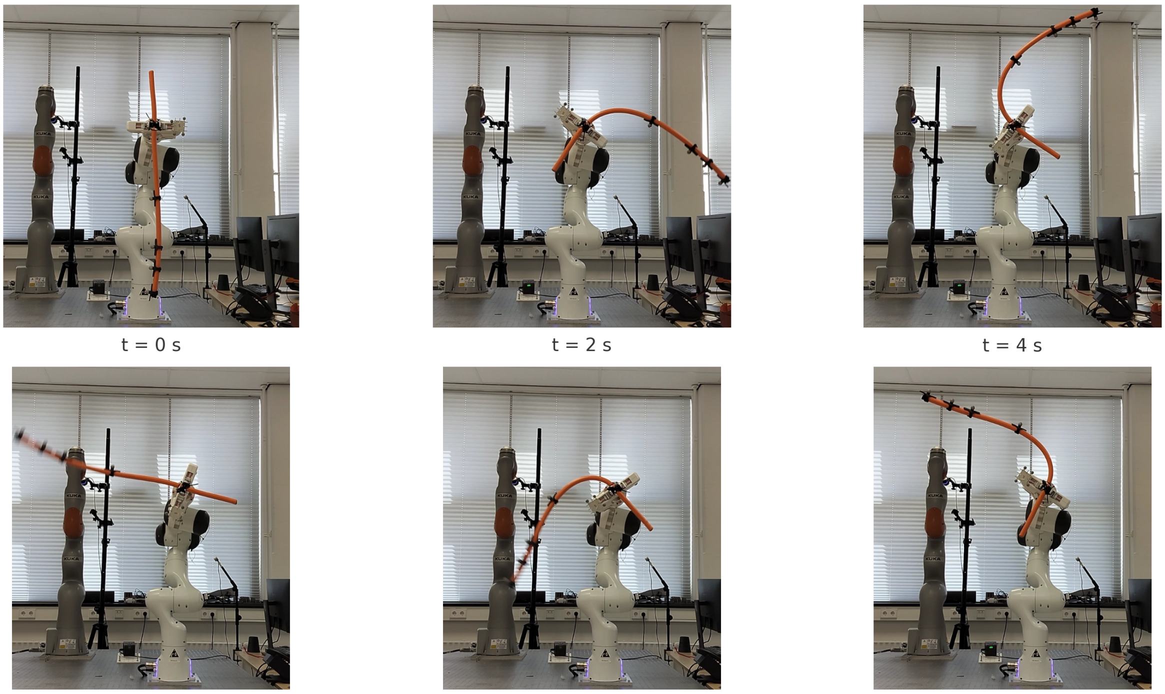


Figure 4-20: Statics solution with the identified stiffness $E^\star$ The configuration is obtained by solving the static equilibrium.

4-5-3 Control Experiments

The proposed control pipeline was evaluated on hardware by *offline* computing NMPC trajectories and streaming the base–joint reference to the Franka's low–level joint–impedance controller. As illustrated in Fig. 4-18, the robot executes a swing–up and switching maneuver between two lateral equilibria (chosen for practicality over the upright). The corresponding simulated shapes from the offline NMPC solution, rendered at the same timestamps, are shown in Fig. 4-22; the visual agreement confirms that the commanded motion reproduces the planned sequence.

For quantitative comparison, Fig. 4-23 reports the four configuration coordinates q_0–q_3 (base joint plus a second–order Legendre GVS shape model, totaling 4 coordinates) from (i) NMPC simulation and (ii) OptiTrack reconstruction during the experiment. The trajectories are close but not identical small biases appear due to modeling mismatch and cable hysteresis (left/right asymmetry). Nevertheless, the reference is tracked well enough to complete the task within hardware limits, demonstrating that the offline NMPC plans are feasible when executed via the Franka's impedance controller.



t = 0 s t = 2 s t = 4 s

t = 9 s t = 11 s t = 13 s

Figure 4-21: Franka–cable swing sequence at selected times.

To respect the velocity limit of the last joint, the NMPC reference was uniformly *time–scaled* prior to execution. Despite this retiming, the swing–up proceeds as intended: around $t \approx 4\,\mathrm{s}$ the cable settles near the first lateral equilibrium, and between $t \approx 4$–$13\,\mathrm{s}$ it transitions toward the opposite side, consistent with the snapshot sequence.

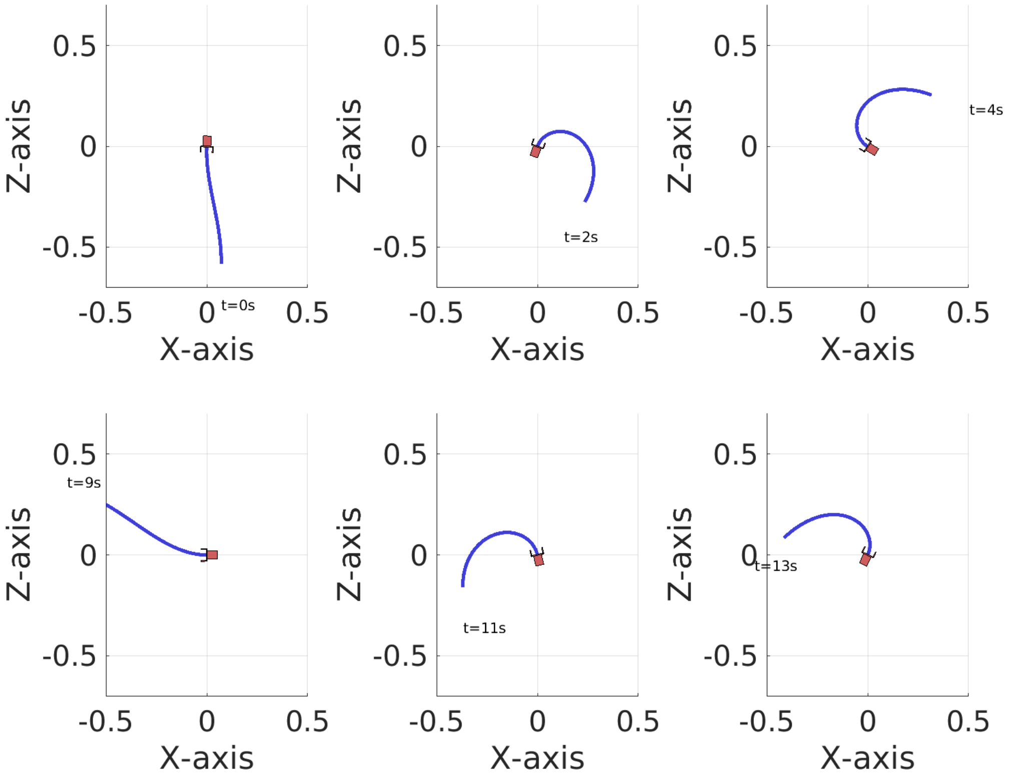


Figure 4-22: Visualization of the shapes obtained from the simulation of the system using the NMPC controller, shown at the same timestamps as the real images in Fig. 4-18.

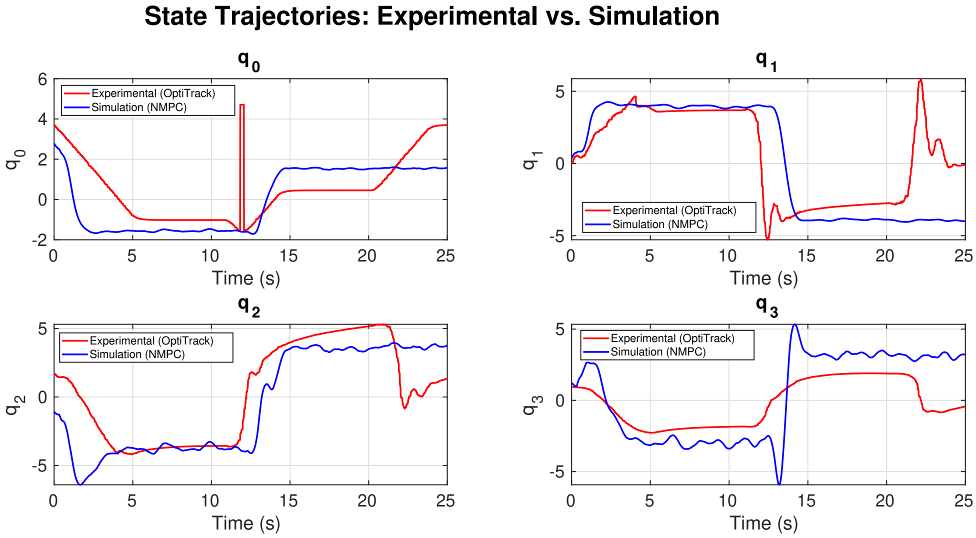


Figure 4-23: State trajectories (q_0–q_3): comparison between the NMPC simulation output and the OptiTrack measurements recorded during the real experiment.

4-6 Discussion of Results

The simulation studies establish baseline validity of the NMPC framework. On the nominal SIP model (Table 4-1), the controller regulates to the target with short settling time and modest control effort, indicating that the discretization, cost shaping, and DDP-based short solves are sufficient for accurate tracking. Disturbance-rejection tests (Figures 4-5–4-6) show bounded errors during force and moment pulses and rapid recovery after the disturbance interval. Robustness to parameter mismatch confirms the expected trend: plants that are lighter or stiffer than the nominal model remain easy to stabilize and often yield smaller settling times and reduced RMS effort. Heavier or softer plants become progressively harder to control, evidenced by longer settling times and higher effort, with failures appearing for very large deviations (about 50% or more), consistent with the physical intuition that increased inertia and reduced stiffness degrade controllability.

The comparison with partial feedback linearization highlights a qualitative advantage. Within the feasibility region defined by the open-loop stability and zero-dynamics stability conditions (Section 4-2-2), collocated PFL performs as expected while non-collocated PFL enlarges the stabilizable set. NMPC extends stabilization beyond both PFL variants for mass–stiffness combinations that violate the zero-dynamics stability requirement. This supports the view that model-based optimal control with full nonlinear dynamics can unlock performance in regimes where structure-based linearization controllers are not applicable.

Generality was examined by applying the same NMPC stack to other underactuated soft systems modeled with GVS. Both the R-SIP and the soft Furuta pendulum are stabilized (Figures 4-13a–4-13), with the R-SIP using a second-order Legendre basis for the soft link. This suggests that the formulation, together with the analytical dynamics derivatives, transfers across architectures without algorithmic changes.

Computational performance indicates practical real-time potential. Average per-iteration solve time is about 7 ms for $n = 2$ and increases to about 17–28 ms for $n = 3$–5 (Fig. 4-14), which is compatible with control loops in the 30–100 Hz range depending on model order and available compute. Using analytical derivatives accelerates solves by roughly 4.4× relative to numerical differentiation while achieving comparable final accuracy (Fig. 4-15), which is important for NMPC where only a few DDP iterations are executed per step.

Hardware experiments demonstrate feasibility of the sensing and planning pipeline. Marker-based IK with analytical Jacobians reconstructs the cable shape in real time, and a static-identification procedure estimates the effective stiffness from equilibrium fits while choosing a high damping ratio to match observed transients. Due to joint-velocity limits and the decision to operate the last joint in impedance mode rather than torque mode, the closed-loop NMPC was not executed on the robot within the available time. Instead, time-scaled trajectories computed offline were sent as joint references. The resulting motions match the planned swing-up to two lateral equilibria and the subsequent switching between them (Figures 4-18, 4-22, and 4-23). Small discrepancies are attributable to model mismatch and cable hysteresis, yet the task is completed reliably.

Overall, the results support three conclusions. First, NMPC with DDP and analytical derivatives is effective and fast enough for soft systems of moderate order. Second, the approach offers a wider stabilizable region than PFL-based baselines in mass–stiffness regimes that violate zero-dynamics stability. Third, the GVS-based modeling and derivative machinery provides a portable backbone across different soft and hybrid soft–rigid platforms. Future work should integrate closed-loop NMPC on hardware, relax the actuation limits by using torque control or a dedicated joint drive, and refine material models to capture hysteresis and rate effects.

Chapter 5

Conclusions and Future Work

This thesis has explored the main research question outlined in Chapter 1: the use of optimal control for soft robots, with a particular focus on their effectiveness in underactuated cases within low mass–stiffness regimes. To address this, an optimal control framework was developed based on Differential Dynamic Programming (DDP) for the implementation of a Nonlinear Model Predictive Controller (NMPC), where the soft robot dynamics are described using the Geometric Variable Strain (GVS) approach.

The results demonstrate the validity of the proposed framework. The combination of the modelling and control architecture is both generalizable and computationally efficient, particularly with the implementation of analytical gradients. Nonetheless, the framework can be further refined and extended in future work.

The proposed controller successfully addresses the problem of shape regulation in underactuated soft robotic systems, including the SIP, R-SIP, and soft Furuta pendulum. It exhibits robustness to disturbances and model mismatch. Most importantly, the controller is shown to stabilize systems in low-stiffness regimes regions beyond the applicability of collocated control. This finding highlights that stability of the zero dynamics is not a strict prerequisite for designing stabilizing controllers. Optimal control methods that exploit the complete system model therefore show strong promise for regulating a wide range of soft robotic systems that can be described with the GVS framework.

The use of analytical gradients has significantly improved the computational efficiency of the controller. Although real-time implementation was not achieved within the timeline of this thesis due to computational constraints, the results indicate that practical control rates of 50–100 Hz are achievable, provided that the dimensionality of the soft robot model is kept sufficiently low. The offline trajectory generation further supports the viability of deploying the controller on real systems.

Limitations of the Current Work

While the proposed framework demonstrates strong potential for controlling underactuated soft robotic systems, several limitations remain that must be acknowledged.

Limited validation on high-dimensional systems

The framework has not yet been evaluated on systems with very high DoFs, i.e. $n > 10$. Although the GVS modelling approach is scalable in principle, its computational burden grows rapidly with model dimension. The current validation focused on relatively low-dimensional systems such as the SIP, R-SIP, and soft Furuta pendulum. Extending the method to very complex high DoF models remains an open challenge, both in terms of computational feasibility and solver convergence.

Lack of hardware experiments in real time

All evaluations presented in this thesis were carried out in simulation, with offline trajectory generation used to assess feasibility. Real-time implementation on physical hardware was not achieved within the timeline of the thesis. Although the results indicate that practical control rates between 50–100 Hz may be possible for sufficiently low-dimensional models, this was not experimentally validated. Hardware implementation would introduce additional sources of uncertainty such as sensing noise, actuation delays, and unmodeled dynamics, which have not yet been addressed.

Absence of formal stability guarantees

The proposed controller demonstrates strong empirical performance, including robustness to disturbances and parameter mismatch. However, no formal stability proof has been derived. In particular, while the results show that stabilization can be achieved in regimes beyond the applicability of collocated PFL, the lack of rigorous guarantees means that the method must still be considered heuristic in nature. Future work should seek to provide theoretical analysis of stability, potentially through Lyapunov-based arguments or contraction analysis [113] adapted to the GVS formulation.

Recommendations for future work

Implicit Differential Dynamic Programming

One promising direction for future research is the integration of Implicit Differential Dynamic Programming (IDDP) [114] into the proposed control framework. Unlike classical DDP, which relies on explicit forward integration of the system dynamics, IDDP reformulates the optimal control problem using implicit integrators, including variational integrators. This formulation has two key advantages. First, implicit and variational integrators provide superior numerical stability and energy preservation properties, making them particularly well-suited for stiff or highly nonlinear systems such as soft robots. Second, IDDP allows for larger integration time steps without sacrificing stability, which directly reduces the number of shooting intervals required for trajectory optimization, thereby lowering computational cost.

The primal-dual proximal Lagrangian approach proposed in the IDDP framework enables handling both dynamical and path constraints in a unified and numerically stable manner. Importantly, the method relaxes the need to solve dynamics exactly at each iteration. Instead, the dynamics can be enforced inexactly and progressively refined as the solver converges, improving robustness and scalability to higher-dimensional systems. For soft robots modeled via the GVS approach, this could significantly extend the applicability of NMPC by enabling real-time control at practical rates even for more complex models.

Incorporating IDDP into soft robotic control therefore holds promise for achieving both faster and more robust online trajectory optimization, potentially overcoming current limitations related to real-time feasibility and scalability. Future work should explore the use of IDDP with variational integrators for soft continuum models, benchmark its performance against classical DDP-based NMPC, and evaluate its potential for experimental deployment on physical soft robotic platforms. The work in the paper the derives the analytical gradients [105] demonstrates the use of the derived gradients using implicit integration schemes.

Contact-Implicit Trajectory Optimization

Another promising research direction is the use of contact-implicit trajectory optimization methods, such as Hybrid iLQR (HiLQR) [115, 116], to extend the control framework to systems with frequent and complex contact interactions. Traditional NMPC approaches for soft robots often rely on fixed or predefined contact sequences, which can limit robustness and adaptability when unexpected perturbations or environmental changes occur. HiLQR addresses this by incorporating contact events directly into the trajectory optimization process, using the saltation matrix to accurately propagate gradients through mode transitions and allowing contact sequences to vary dynamically during optimization.

The key advantage of this approach is that it eliminates the need to prespecify contact timings, enabling the controller to seamlessly add, remove, or shift contacts in response to disturbances. By leveraging parallelized simulations and efficient analytical derivatives of the dynamics, HiLQR MPC has been demonstrated to replan whole-body motions in real time while maintaining dynamic feasibility. For soft robots, which inherently operate under hybrid dynamics due to interactions with their environment, this capability could be transformative.

Integrating contact-implicit optimization into the proposed framework would allow future controllers to not only stabilize soft robotic systems in low-stiffness regimes but also to robustly manage environmental contacts, collisions, or support changes. This could be particularly relevant for tasks such as locomotion, manipulation, and human–robot interaction, where adaptability to unplanned contacts is critical.

Appendix A

Analytical derivatives of Geometric Variable Strain Model

Inverse dynamics (ID) on the discretized chain. For n_p computational points,

$$ID \;=\; \sum_{\alpha=1}^{n_p-1} ID_\alpha \;=\; \sum_{\alpha=1}^{n_p-1} \boldsymbol{S}_\alpha^\top \boldsymbol{F}_\alpha^C, \qquad \boldsymbol{F}_\alpha^C \;=\; \sum_{k=\alpha+1}^{n_p} \mathrm{Ad}^*_{\boldsymbol{g}_{\alpha k}} \boldsymbol{F}_k, \tag{A-1}$$

where $\boldsymbol{S}_\alpha$ is the motion subspace at point α, Ad^* is the coadjoint map on $SE(3)$, and $\boldsymbol{F}_k$ is the local point wrench.

A.1 $\partial ID/\partial \boldsymbol{q}$ (Sec. 4.1)

For each α,

$$\frac{\partial ID_\alpha}{\partial \boldsymbol{q}} = \frac{\partial \boldsymbol{S}_\alpha^\top}{\partial \boldsymbol{q}} \boldsymbol{F}_\alpha^C + \boldsymbol{S}_\alpha^\top \Big(\boldsymbol{N}_\alpha^C \boldsymbol{R}_\alpha^B + \boldsymbol{M}_\alpha^C \boldsymbol{Q}_\alpha^B + \boldsymbol{U}_\alpha^S + \boldsymbol{P}_\alpha^S \Big), \tag{A-2}$$

and the total derivative is $\dfrac{\partial ID}{\partial \boldsymbol{q}} = \displaystyle\sum_{\alpha=1}^{n_p-1} \dfrac{\partial ID_\alpha}{\partial \boldsymbol{q}}$.

Forward-/backward-pass recursions (needed in (A-2)**).**

$$\boldsymbol{R}_\alpha^B = \mathrm{Ad}^{-1}_{\boldsymbol{g}_{\alpha-1,\alpha}} \Big(\boldsymbol{R}_{\alpha-1} + \boldsymbol{R}_{\alpha-1}^B \Big), \qquad \boldsymbol{Q}_\alpha^B = \mathrm{Ad}^{-1}_{\boldsymbol{g}_{\alpha-1,\alpha}} \Big(\boldsymbol{Q}_{\alpha-1} + \boldsymbol{Q}_{\alpha-1}^B \Big), \tag{A-3}$$

$$\boldsymbol{F}_\alpha^C = \mathrm{Ad}^*_{\boldsymbol{g}_{\alpha,\alpha+1}} \Big(\boldsymbol{F}_{\alpha+1} + \boldsymbol{F}_{\alpha+1}^C \Big), \qquad \boldsymbol{N}_\alpha^C = \mathrm{Ad}^*_{\boldsymbol{g}_{\alpha,\alpha+1}} \Big(\boldsymbol{N}_{\alpha+1} + \boldsymbol{N}_{\alpha+1}^C \Big) \mathrm{Ad}^{-1}_{\boldsymbol{g}_{\alpha,\alpha+1}}, \tag{A-4}$$

$$\boldsymbol{M}_\alpha^C = \mathrm{Ad}^*_{\boldsymbol{g}_{\alpha,\alpha+1}} \Big(\boldsymbol{M}_{\alpha+1} + \boldsymbol{M}_{\alpha+1}^C \Big) \mathrm{Ad}^{-1}_{\boldsymbol{g}_{\alpha,\alpha+1}}, \qquad \boldsymbol{U}_\alpha^S = \boldsymbol{N}_\alpha^C \boldsymbol{R}_\alpha + \boldsymbol{M}_\alpha^C \boldsymbol{Q}_\alpha + \mathrm{Ad}^*_{\boldsymbol{g}_{\alpha,\alpha+1}} \boldsymbol{U}_{\alpha+1}^S, \tag{A-5}$$

$$\boldsymbol{P}_\alpha^S = \mathrm{ad}^*_{\boldsymbol{F}_\alpha^C} \boldsymbol{S}_\alpha + \mathrm{Ad}^*_{\boldsymbol{g}_{\alpha,\alpha+1}} \boldsymbol{P}_{\alpha+1}^S. \tag{A-6}$$

Moreover,

$$\boldsymbol{N}_k = \mathrm{ad}^*_{\boldsymbol{M}_k \boldsymbol{\eta}_k} + \mathrm{ad}^*_{\boldsymbol{\eta}_k} \boldsymbol{M}_k - \boldsymbol{M}_k \, \mathrm{ad}_{\boldsymbol{\eta}_k}, \tag{A-7}$$

$$\boldsymbol{R}_\beta = \mathrm{ad}_{\boldsymbol{\eta}^+_\beta} \boldsymbol{S}_\beta + \frac{\partial \boldsymbol{S}_\beta}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}}, \tag{A-8}$$

$$\boldsymbol{Q}_\beta = \mathrm{ad}_{\dot{\boldsymbol{\eta}}^+_\beta} \boldsymbol{S}_\beta + \mathrm{ad}_{\boldsymbol{\eta}^+_\beta} \boldsymbol{R}_\beta + \mathrm{ad}_{\boldsymbol{\eta}_\beta} \frac{\partial \boldsymbol{S}_\beta}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}} + \frac{\partial \dot{\boldsymbol{S}}_\beta}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}} + \frac{\partial \boldsymbol{S}_\beta}{\partial \boldsymbol{q}} \ddot{\boldsymbol{q}} - \mathrm{ad}_{\mathrm{Ad}^{-1}_{\boldsymbol{g}_\beta} \boldsymbol{G}} \boldsymbol{S}_\beta. \tag{A-9}$$

All formulas above are exactly those in Eqs. (21)–(27) and (24a–c), with the forward/backward accumulation in (26a–g) from [105].

A.2 $\partial ID/\partial \dot{\boldsymbol{q}}$ (Sec. 4.2)

For each α,

$$\frac{\partial ID_\alpha}{\partial \dot{\boldsymbol{q}}} = \boldsymbol{S}^\top_\alpha \Big(\boldsymbol{N}^C_\alpha \boldsymbol{S}^B_\alpha + \boldsymbol{M}^C_\alpha \boldsymbol{Y}^B_\alpha + \boldsymbol{V}^S_\alpha \Big), \tag{A-10}$$

and $\dfrac{\partial ID}{\partial \dot{\boldsymbol{q}}} = \displaystyle\sum_{\alpha=1}^{n_p-1} \dfrac{\partial ID_\alpha}{\partial \dot{\boldsymbol{q}}}$.

Recursions (Sec. 4.2).

$$\boldsymbol{S}^B_\alpha = \mathrm{Ad}^{-1}_{\boldsymbol{g}_{\alpha-1,\alpha}} \Big(\boldsymbol{S}_{\alpha-1} + \boldsymbol{S}^B_{\alpha-1} \Big), \qquad \boldsymbol{Y}^B_\alpha = \mathrm{Ad}^{-1}_{\boldsymbol{g}_{\alpha-1,\alpha}} \Big(\boldsymbol{Y}_{\alpha-1} + \boldsymbol{Y}^B_{\alpha-1} \Big), \tag{A-11}$$

$$\boldsymbol{V}^S_\alpha = \boldsymbol{N}^C_\alpha \boldsymbol{S}_\alpha + \boldsymbol{M}^C_\alpha \boldsymbol{Y}_\alpha + \mathrm{Ad}^*_{\boldsymbol{g}_{\alpha,\alpha+1}} \boldsymbol{V}^S_{\alpha+1}, \tag{A-12}$$

with

$$\boldsymbol{Y}_\beta \; = \; \boldsymbol{R}_\beta + \mathrm{ad}_{\boldsymbol{\eta}_\beta} \boldsymbol{S}_\beta + \dot{\boldsymbol{S}}_\beta. \tag{A-13}$$

These are Eqs. (28)–(33) in the [105].

Remarks. Superscripts B denote forward-pass kinematic accumulations, while C/S denote backward-pass wrench/inertia accumulations; all have size $6 \times n_{\mathrm{dof}}$, and the projection by $\boldsymbol{S}^\top_\alpha$ yields $n_{\mathrm{dof}} \times n_{\mathrm{dof}}$ blocks. Full derivations and operator identities (Magnus/adjoint derivatives) are given in Appendices C–D of [105].

Appendix B

Analytical Jacobian of the Inverse Kinematics Problem

Kinematics of a marker pose. Let the marker pose be

$$\boldsymbol{g} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{r} \\ \boldsymbol{0}^\top & 1 \end{bmatrix} \in SE(3), \qquad \hat{\boldsymbol{\eta}} = \begin{bmatrix} \tilde{\boldsymbol{\omega}} & \boldsymbol{v} \\ \boldsymbol{0}^\top & 0 \end{bmatrix} \in \mathfrak{se}(3),$$

with $\tilde{\boldsymbol{\omega}}\boldsymbol{x} = \boldsymbol{\omega} \times \boldsymbol{x}$. Using the right–trivialized (body) twist,

$$\dot{\boldsymbol{g}} = \boldsymbol{g}\,\hat{\boldsymbol{\eta}} \implies \begin{bmatrix} \dot{\boldsymbol{R}} & \dot{\boldsymbol{r}} \\ \boldsymbol{0}^\top & 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}\tilde{\boldsymbol{\omega}} & \boldsymbol{R}\boldsymbol{v} \\ \boldsymbol{0}^\top & 0 \end{bmatrix},$$

hence

$$\dot{\boldsymbol{R}} = \boldsymbol{R}\tilde{\boldsymbol{\omega}}, \qquad \dot{\boldsymbol{r}} = \boldsymbol{R}\boldsymbol{v}.$$

GVS twist–to–position Jacobian. From GVS kinematics the marker body twist satisfies

$$\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix} = \boldsymbol{J}(\boldsymbol{q})\,\dot{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{J}_\omega(\boldsymbol{q}) \\ \boldsymbol{J}_v(\boldsymbol{q}) \end{bmatrix} \dot{\boldsymbol{q}}.$$

Using $\dot{\boldsymbol{r}} = \boldsymbol{R}\boldsymbol{v}$ and $\dot{\boldsymbol{r}} = (\partial \boldsymbol{r}/\partial \boldsymbol{q})\dot{\boldsymbol{q}}$,

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{q}} = \boldsymbol{R}\,\boldsymbol{J}_v(\boldsymbol{q})$$

(i.e., the translational Jacobian in the global frame equals the rotation $\boldsymbol{R}$ times the lower $3 \times n$ block of the body-twist Jacobian).

Jacobian of the IK residuals and gradient. For marker j with predicted position $\hat{\boldsymbol{m}}_j^{\mathcal{G}}(\boldsymbol{q})$ and residual $\boldsymbol{r}_j(\boldsymbol{q}) = \hat{\boldsymbol{m}}_j^{\mathcal{G}}(\boldsymbol{q}) - \boldsymbol{m}_j^{\mathcal{G}}$,

$$\frac{\partial \hat{\boldsymbol{m}}_j^{\mathcal{G}}}{\partial \boldsymbol{q}} = \boldsymbol{J}_j(\boldsymbol{q}) = \boldsymbol{R}_j(\boldsymbol{q})\, \boldsymbol{J}_{v,j}(\boldsymbol{q}) \in \mathbb{R}^{3\times n}.$$

Stacking four markers gives

$$\boldsymbol{r}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \\ \boldsymbol{r}_3 \\ \boldsymbol{r}_4 \end{bmatrix} \in \mathbb{R}^{12}, \qquad \boldsymbol{J}(\boldsymbol{q}) = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{q}} = \begin{bmatrix} \boldsymbol{R}_1 \boldsymbol{J}_{v,1} \\ \boldsymbol{R}_2 \boldsymbol{J}_{v,2} \\ \boldsymbol{R}_3 \boldsymbol{J}_{v,3} \\ \boldsymbol{R}_4 \boldsymbol{J}_{v,4} \end{bmatrix} \in \mathbb{R}^{12\times n}.$$

For the least–squares cost $\Phi(\boldsymbol{q}) = \frac{1}{2}\|\boldsymbol{r}(\boldsymbol{q})\|_2^2$, the gradient used by the solver is

$$\nabla\Phi(\boldsymbol{q}) \;=\; \boldsymbol{J}(\boldsymbol{q})^{\top}\, \boldsymbol{r}(\boldsymbol{q})$$

(e.g., for gradient-descent updates $\boldsymbol{q}^{+} = \boldsymbol{q} - \alpha\, \nabla\Phi(\boldsymbol{q})$).

Marker offset via adjoint mapping. If the physical marker is at a fixed local offset from the centerline at arclength X_j, encode it by $\boldsymbol{g}_{\mathcal{F}\to M} \in SE(3)$ (pure translation). With the centerline-frame body Jacobian $\boldsymbol{J}_j^{(\mathcal{F})} = [\boldsymbol{J}_{\omega,j}^{(\mathcal{F})};\, \boldsymbol{J}_{v,j}^{(\mathcal{F})}] \in \mathbb{R}^{6\times n}$, the marker-frame body Jacobian is obtained by the adjoint:

$$\boldsymbol{J}_j^{(M)}(\boldsymbol{q}) \;=\; \mathrm{Ad}_{\boldsymbol{g}_{\mathcal{F}\to M}^{-1}}\, \boldsymbol{J}_j^{(\mathcal{F})}(\boldsymbol{q}),$$

and the IK position Jacobian uses its translational block,

$$\frac{\partial \hat{\boldsymbol{m}}_j^{\mathcal{G}}}{\partial \boldsymbol{q}} \;=\; \boldsymbol{R}_j\, \boldsymbol{J}_{v,j}^{(M)}(\boldsymbol{q})\,, \qquad j = 1, \ldots, 4.$$

Bibliography

[1] Elliot W Hawkes, Carmel Majidi, and Michael T Tolley. Hard questions for soft robotics. *Science robotics*, 6(53):eabg6049, 2021.

[2] Cosimo Della Santina, Manuel G. Catalano, and Antonio Bicchi. Soft robots. *Encyclopedia of Robotics*, pages 1–15, 2021.

[3] Cecilia Laschi, Barbara Mazzolai, and Matteo Cianchetti. Soft robotics: Technologies and systems pushing the boundaries of robot abilities. *Science robotics*, 1(1):eaah3690, 2016.

[4] Robert K Katzschmann, Andrew D Marchese, and Daniela Rus. Hydraulic autonomous soft robotic fish for 3d swimming. In *Experimental Robotics: The 14th International Symposium on Experimental Robotics*, pages 405–420. Springer, 2015.

[5] Federico Renda, Michele Giorelli, Marcello Calisti, Matteo Cianchetti, and Cecilia Laschi. Dynamic model of a multibending soft robot arm driven by cables. *IEEE Transactions on Robotics*, 30(5):1109–1122, 2014.

[6] Samuel Shian, Katia Bertoldi, and David R Clarke. Dielectric elastomer based “grippers” for soft robotics. *Advanced Materials*, 2015.

[7] Matteo Cianchetti. Fundamentals on the use of shape memory alloys in soft robotics. *Interdisciplinary mechatronics*, pages 227–254, 2013.

[8] Costanza Armanini, Frédéric Boyer, Anup Teejo Mathew, Christian Duriez, and Federico Renda. Soft robots modeling: A structured overview. *IEEE Transactions on Robotics*, 39:1728–1748, 6 2023.

[9] Carlo Alessi, Camilla Agabiti, Daniele Caradonna, Cecilia Laschi, Federico Renda, and Egidio Falotico. Rod models in continuum and soft robot control: a review. *arXiv preprint arXiv:2407.05886*, 2024.

[10] Cosimo Della Santina, Christian Duriez, and Daniela Rus. Model-based control of soft robots: A survey of the state of the art and open challenges. *IEEE Control Systems Magazine*, 43(3):30–65, 2023.

[11] Pablo Borja, Azita Dabiri, and Cosimo Della Santina. Energy-based shape regulation of soft robots with unactuated dynamics dominated by elasticity. *2022 IEEE 5th International Conference on Soft Robotics, RoboSoft 2022*, pages 396–402, 2022.

[12] Pietro Pustina, Cosimo Della Santina, and Alessandro De Luca. Feedback regulation of elastically decoupled underactuated soft robots. *IEEE Robotics and Automation Letters*, 7:4512–4519, 4 2022.

[13] Cosimo Della Santina. The soft inverted pendulum with affine curvature. *Proceedings of the IEEE Conference on Decision and Control*, 2020-December:4135–4142, 12 2020.

[14] Hend Abdelaziz, Abdullah Ahmed, and Haitham El-Hussieny. Approximate neural network-based nonlinear model predictive control of soft continuum robots. In *2024 20th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, pages 1–7. IEEE, 2024.

[15] Daniel Bruder, Xun Fu, R. Brent Gillespie, C. David Remy, and Ram Vasudevan. Data-driven control of soft robots using koopman operator theory. *IEEE Transactions on Robotics*, 37:948–961, 6 2021.

[16] Etienne Menager, Alexandre Bilger, Wilson Jallet, Justin Carpentier, and Christian Duriez. Condensed semi-implicit dynamics for trajectory optimization in soft robotics. *2024 IEEE 7th International Conference on Soft Robotics, RoboSoft 2024*, pages 808–815, 2024.

[17] Kyle L. Walker, Cosimo Della Santina, and Francesco Giorgio-Serchi. Model predictive wave disturbance rejection for underwater soft robotic manipulators. *2024 IEEE 7th International Conference on Soft Robotics, RoboSoft 2024*, pages 40–47, 2024.

[18] Andrew D. Marchese, Russ Tedrake, and Daniela Rus. Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator. *Proceedings - IEEE International Conference on Robotics and Automation*, 2015-June:2528–2535, 6 2015.

[19] E. B.Lane Kennedy, Kendra C. Buresch, Preethi Boinapally, and Roger T. Hanlon. Octopus arms exhibit exceptional flexibility. *Scientific Reports 2020 10:1*, 10:1–10, 11 2020.

[20] A. M. Wilson, J. C. Lowe, K. Roskilly, P. E. Hudson, K. A. Golabek, and J. W. McNutt. Locomotion dynamics of hunting in wild cheetahs. *Nature 2013 498:7453*, 498:185–189, 6 2013.

[21] Jianing Wu, Yichao Zhao, Yunshu Zhang, David Shumate, Stephanie Braccini Slade, Scott V. Franklin, and David L. Hu. Elephant trunks form joints to squeeze together small objects. *Journal of The Royal Society Interface*, 15, 10 2018.

[22] Sangbae Kim, Cecilia Laschi, and Barry Trimmer. Soft robotics: A bioinspired evolution in robotics. *Trends in Biotechnology*, 31:287–294, 5 2013.

[23] Bertrand Tondu and Pierre Lopez. Modeling and control of mckibben artificial muscle robot actuators. *IEEE Control Systems*, 20:15–38, 2000.

[24] Buckingham Post. More help for polio victims. *Newsweek*, 1958. Historical reference on polio rehabilitation efforts. Exact issue details require archival access.

[25] International Centre for Mechanical Sciences, International Federation for the Theory of Machines, and Mechanisms. *On Theory and Practice of Robots and Manipulators*. Courses and Lectures - International Centre for Mechanical Sciences. Springer, 1974.

[26] Koichi Suzumori, Shoichi Iikura, and Hiroshisa Tanaka. Applying a flexible microactuator to robotic mechanisms. *IEEE Control systems magazine*, 12(1):21–27, 1992.

[27] Zhongkui Wang, Ryo Kanegae, and Shinichi Hirai. Circular shell gripper for handling food products. *Soft robotics*, 8(5):542–554, 2021.

[28] Shunichi Kurumaya, Brennan T. Phillips, Kaitlyn P. Becker, Michelle H. Rosen, David F. Gruber, Kevin C. Galloway, Koichi Suzumori, and Robert J. Wood. A modular soft robotic wrist for underwater manipulation. *https://home.liebertpub.com/soro*, 5:399–409, 8 2018.

[29] Haider Abidi, Giada Gerboni, Margherita Brancadoro, Jan Fras, Alessandro Diodato, Matteo Cianchetti, Helge Wurdemann, Kaspar Althoefer, and Arianna Menciassi. Highly dexterous 2-module soft robot for intra-organ navigation in minimally invasive surgery. *The international journal of medical robotics + computer assisted surgery : MRCAS*, 14, 2 2018.

[30] Alin Albu-Schaffer, Oliver Eiberger, Markus Grebenstein, Sami Haddadin, Christian Ott, Thomas Wimböck, Sebastian Wolf, and Gerd Hirzinger. Soft robotics. *IEEE Robotics and Automation Magazine*, 15:20–30, 2008.

[31] David Rollinson, Yigit Bilgen, Ben Brown, Florian Enner, Steven Ford, Curtis Layton, Justine Rembisz, Mike Schwerin, Andrew Willig, Pras Velagapudi, et al. Design and architecture of a series elastic snake robot. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4630–4636. IEEE, 2014.

[32] Mohammad Askari, Won Dong Shin, Damian Lenherr, William Stewart, and Dario Floreano. Avian-inspired claws enable robot perching or walking. *IEEE/ASME Transactions on Mechatronics*, 2023.

[33] Alexander T Spröwitz, Alexandre Tuleu, Mostafa Ajallooeian, Massimo Vespignani, Rico Möckel, Peter Eckert, Michiel D'Haene, Jonas Degrave, Arne Nordmann, Benjamin Schrauwen, et al. Oncilla robot: a versatile open-source quadruped research robot with compliant pantograph legs. *Frontiers in Robotics and AI*, 5:67, 2018.

[34] Stefano Palagi, Andrew G Mark, Shang Yik Reigh, Kai Melde, Tian Qiu, Hao Zeng, Camilla Parmeggiani, Daniele Martella, Alberto Sanchez-Castillo, Nadia Kapernaum, et al. Structured light enables biomimetic swimming and versatile locomotion of photoresponsive soft microrobots. *Nature materials*, 15(6):647–653, 2016.

[35] Jennifer Frame, Nick Lopez, Oscar Curet, and Erik D Engeberg. Thrust force characterization of free-swimming soft robotic jellyfish. *Bioinspiration & biomimetics*, 13(6):064001, 2018.

[36] Robert K Katzschmann, Joseph DelPreto, Robert MacCurdy, and Daniela Rus. Exploration of underwater life with an acoustically controlled soft robotic fish. *Science Robotics*, 3(16):eaar3449, 2018.

[37] Daniela Rus and Michael T. Tolley. Design, fabrication and control of soft robots. *Nature 2015 521:7553*, 521:467–475, 5 2015.

[38] Michael D Grissom, Vilas Chitrakaran, Dustin Dienno, Matthew Csencits, Michael Pritts, Bryan Jones, William McMahan, Darren Dawson, Chris Rahn, and Ian Walker. Design and experimental testing of the octarm soft robot manipulator. In *Unmanned systems technology VIII*, volume 6230, pages 491–500. SPIE, 2006.

[39] Barry A Trimmer, Ann E Takesian, Brian M Sweet, Chris B Rogers, Daniel C Hake, and Daniel J Rogers. Caterpillar locomotion: a new model for soft-bodied climbing and burrowing robots. In *7th international symposium on technology and the mine problem*, volume 1, pages 1–10. Monterey, CA: Mine Warfare Association, 2006.

[40] Younan Xia and George M. Whitesides. Soft lithography. *Annual Review of Materials Science*, 28:153–184, 1998.

[41] J. G. Cham, S. A. Bailey, J. E. Clark, R. J. Full, and M. R. Cutkosky. Fast and robust: hexapedal robots via shape deposition manufacturing. *International Journal of Robotics Research*, 21(10-11):869–882, 2002.

[42] Hod Lipson and Melba Kurman. *Fabricated: The New World of 3D Printing.* Wiley, Hoboken, NJ, 1st edition, 2013.

[43] Andrew D Marchese, Robert K Katzschmann, and Daniela Rus. A recipe for soft fluidic elastomer robots. *Soft robotics*, 2(1):7–25, 2015.

[44] Thomas George Thuruthel, Benjamin Shih, Cecilia Laschi, and Michael Thomas Tolley. Soft robot perception using embedded soft sensors and recurrent neural networks. *Science Robotics*, 4(26):eaav1488, 2019.

[45] Delin Hu, Francesco Giorgio-Serchi, Shiming Zhang, and Yunjie Yang. Stretchable e-skin and transformer enable high-resolution morphological reconstruction for soft robots. *nature machine intelligence*, 5(3):261–272, 2023.

[46] Zeyi Yang, Sheng Ge, Fang Wan, Yujia Liu, and Chaoyang Song. Scalable tactile sensing for an omni-adaptive soft robot finger. In *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 572–577. IEEE, 2020.

[47] Chidanand Hegde, Jiangtao Su, Joel Ming Rui Tan, Ke He, Xiaodong Chen, and Shlomo Magdassi. Sensing in soft robotics. *ACS Nano*, 17:15277–15307, 8 2023.

[48] Yasunori Toshimitsu, Ki Wan Wong, Thomas Buchner, and Robert Katzschmann. 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)*, pages 653–660. IEEE, 2021.

[49] Yuan-Fang Zhang, Ningbin Zhang, Hardik Hingorani, Ningyuan Ding, Dong Wang, Chao Yuan, Biao Zhang, Guoying Gu, and Qi Ge. Fast-response, stiffness-tunable soft actuator by hybrid multimaterial 3d printing. *Advanced Functional Materials*, 29(15):1806698, 2019.

[50] Robert K Katzschmann, Joseph DelPreto, Robert MacCurdy, and Daniela Rus. Exploration of underwater life with an acoustically controlled soft robotic fish. *Science Robotics*, 3(16):eaar3449, 2018.

[51] Michael Wehner, Ryan L Truby, Daniel J Fitzgerald, Bobak Mosadegh, George M Whitesides, Jennifer A Lewis, and Robert J Wood. An integrated design and fabrication strategy for entirely soft, autonomous robots. *nature*, 536(7617):451–455, 2016.

[52] Guoying Gu, Ningbin Zhang, Haipeng Xu, Shaoting Lin, Yang Yu, Guohong Chai, Lisen Ge, Houle Yang, Qiwen Shao, Xinjun Sheng, et al. A soft neuroprosthetic hand providing simultaneous myoelectric control and tactile feedback. *Nature biomedical engineering*, 7(4):589–598, 2023.

[53] Tommaso Ranzani, Matteo Cianchetti, Giada Gerboni, Iris De Falco, and Arianna Menciassi. A soft modular manipulator for minimally invasive surgery: design and characterization of a single module. *IEEE Transactions on Robotics*, 32(1):187–200, 2016.

[54] Stefano Fusco, Mahmut Selman Sakar, Stephen Kennedy, Christian Peters, Rocco Bottani, Fabian Starsich, Angelo Mao, Georgios A Sotiriou, Salvador Pané, Sotiris E Pratsinis, et al. An integrated microrobotic platform for on-demand, targeted therapeutic interventions. *Adv. Mater*, 26(6):952–957, 2014.

[55] Lamar O Mair, Georges Adam, Sagar Chowdhury, Aaron Davis, Dian R Arifin, Fair M Vassoler, Herbert H Engelhard, Jinxing Li, Xinyao Tang, Irving N Weinberg, et al. Soft capsule magnetic millirobots for region-specific drug delivery in the central nervous system. *Frontiers in Robotics and AI*, 8:702566, 2021.

[56] Matheus S Xavier, Andrew J Fleming, and Yuen K Yong. Finite element modeling of soft fluidic actuators: Overview and recent developments. *Advanced Intelligent Systems*, 3(2):2000187, 2021.

[57] Philip Moseley, Juan Manuel Florez, Harshal Arun Sonar, Gunjan Agarwal, William Curtin, and Jamie Paik. Modeling, design, and development of soft pneumatic actuators with finite element method. *Advanced engineering materials*, 18(6):978–988, 2016.

[58] Christian Duriez. Control of elastic soft robots based on real-time finite element method. In *2013 IEEE international conference on robotics and automation*, pages 3982–3987. IEEE, 2013.

[59] Federico Renda, Costanza Armanini, Vincent Lebastard, Fabien Candelier, and Frederic Boyer. A geometric variable-strain approach for static modeling of soft manipulators with tendon and fluidic actuation. *IEEE Robotics and Automation Letters*, 5:4006–4013, 7 2020.

[60] Anup Teejo Mathew, Daniel Feliu-Talegon, Abdulaziz Y. Alkayas, Frederic Boyer, and Federico Renda. Reduced order modeling of hybrid soft-rigid robots using global, local, and state-dependent strain parameterization. *International Journal of Robotics Research*, 7 2024.

[61] Gregory S Chirikjian. Conformational modeling of continuum structures in robotics and structural biology: A review. *Advanced Robotics*, 29(13):817–829, 2015.

[62] Cosimo Della Santina, Antonio Bicchi, and Daniela Rus. On an improved state parametrization for soft robots with piecewise constant curvature and its use in model based control. *IEEE Robotics and Automation Letters*, 5:1001–1008, 4 2020.

[63] David A. Haggerty, Michael J. Banks, Ervin Kamenar, Alan B. Cao, Patrick C. Curtis, Igor Mezić, and Elliot W. Hawkes. Control of soft robots with inertial dynamics. *Science Robotics*, 8, 8 2023.

[64] Federico Renda, Frederic Boyer, Jorge Dias, and Lakmal Seneviratne. Discrete cosserat approach for multisection soft manipulator dynamics. *IEEE Transactions on Robotics*, 34:1518–1533, 12 2018.

[65] Federico Renda, Costanza Armanini, Anup Mathew, and Frederic Boyer. Geometrically-exact inverse kinematic control of soft manipulators with general threadlike actuators' routing. *IEEE Robotics and Automation Letters*, 7:7311–7318, 7 2022.

[66] Ernst Hairer, Marlis Hochbruck, Arieh Iserles, and Christian Lubich. Geometric numerical integration. *Oberwolfach Reports*, 3(1):805–882, 2006.

[67] Cosimo Della Santina and Daniela Rus. Control oriented modeling of soft robots: The polynomial curvature case. *IEEE Robotics and Automation Letters*, 5:290–298, 4 2020.

[68] Daniele Caradonna, Michele Pierallini, Cosimo Della Santina, Franco Angelini, and Antonio Bicchi. Model and control of r-soft inverted pendulum. *IEEE Robotics and Automation Letters*, 9:5102–5109, 6 2024.

[69] Pietro Pustina, Cosimo Della Santina, Frédéric Boyer, Alessandro De Luca, and Federico Renda. Input decoupling of lagrangian systems via coordinate transformation: General characterization and its application to soft robotics. *IEEE Transactions on Robotics*, 2024.

[70] Giulia Pagnanelli, Michele Pierallini, Franco Angelini, and Antonio Bicchi. Assessing an energy-based control for the soft inverted pendulum in hamiltonian form. *IEEE Control Systems Letters*, 8:922–927, 2024.

[71] Mark W Spong. Partial feedback linearization of underactuated mechanical systems. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, volume 1, pages 314–321. IEEE, 1994.

[72] Sami Haddadin, Felix Huber, and Alin Albu-Schäffer. Optimal control for exploiting the natural dynamics of variable stiffness robots. In *2012 IEEE International Conference on Robotics and Automation*, pages 3347–3354, 2012.

[73] Cosimo Della Santina, Matteo Bianchi, Giorgio Grioli, Franco Angelini, Manuel Catalano, Manolo Garabini, and Antonio Bicchi. Controlling soft robots: Balancing feedback and feedforward elements. *IEEE Robotics Automation Magazine*, 24(3):75–83, 2017.

[74] Matthew J Powell, Eric A Cousineau, and Aaron D Ames. Model predictive control of underactuated bipedal robotic walking. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5121–5126. IEEE, 2015.

[75] Jean Pierre Sleiman, Farbod Farshidian, Maria Vittoria Minniti, and Marco Hutter. A unified mpc framework for whole-body dynamic locomotion and manipulation. *IEEE Robotics and Automation Letters*, 6:4688–4695, 7 2021.

[76] Ruben Grandia, Farbod Farshidian, Rene Ranftl, and Marco Hutter. Feedback mpc for torque-controlled legged robots. *IEEE International Conference on Intelligent Robots and Systems*, pages 4730–4737, 11 2019.

[77] Farbod Farshidian, Edo Jelavic, Asutosh Satapathy, Markus Giftthaler, and Jonas Buchli. Real-time motion planning of legged robots: A model predictive control approach. *IEEE-RAS International Conference on Humanoid Robots*, pages 577–584, 12 2017.

[78] Michael Neunert, Markus Stauble, Markus Giftthaler, Carmine D. Bellicoso, Jan Carius, Christian Gehring, Marco Hutter, and Jonas Buchli. Whole-body nonlinear model predictive control through contacts for quadrupeds. *IEEE Robotics and Automation Letters*, 3:1458–1465, 7 2018.

[79] Jared Di Carlo, Patrick M. Wensing, Benjamin Katz, Gerardo Bledt, and Sangbae Kim. Dynamic locomotion in the mit cheetah 3 through convex model-predictive control. *IEEE International Conference on Intelligent Robots and Systems*, pages 7440–7447, 12 2018.

[80] Maria Vittoria Minniti, Farbod Farshidian, Ruben Grandia, and Marco Hutter. Whole-body mpc for a dynamically stable mobile manipulator. *IEEE Robotics and Automation Letters*, 4(4):3687–3694, 2019.

[81] Sander Tonkens, Joseph Lorenzetti, and Marco Pavone. Soft robot optimal control via reduced order finite element models. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12010–12016, 2021.

[82] W. David Null, William Edwards, Dohun Jeong, Teodor Tchalakov, James Menezes, Kris Hauser, and Y. Zy. Automatically-tuned model predictive control for an underwater soft robot. *IEEE Robotics and Automation Letters*, 9:571–578, 1 2024.

[83] Phillip Hyatt and Marc D. Killpack. Real-time nonlinear model predictive control of robots using a graphics processing unit. *IEEE Robotics and Automation Letters*, 5:1468–1475, 4 2020.

[84] Sander Tonkens, Joseph Lorenzetti, and Marco Pavone. Soft robot optimal control via reduced order finite element models. *Proceedings - IEEE International Conference on Robotics and Automation*, 2021-May:12010–12016, 2021.

[85] James M Bern, Pol Banzet, Roi Poranne, and Stelian Coros. Trajectory optimization for cable-driven soft robot locomotion. In *Robotics: Science and Systems*, volume 1, 2019.

[86] Wu Te Yang, Burak Kürkçü, Motohiro Hirao, Lingfeng Sun, Xinghao Zhu, Zhizhou Zhang, Grace X. Gu, and Masayoshi Tomizuka. Control of soft pneumatic actuators with approximated dynamical modeling. *2023 IEEE International Conference on Robotics and Biomimetics, ROBIO 2023*, 2023.

[87] Matheus S. Xavier, Andrew J. Fleming, and Yuen Kuan Yong. Nonlinear estimation and control of bending soft pneumatic actuators using feedback linearization and ukf. *IEEE/ASME Transactions on Mechatronics*, 27:1919–1927, 8 2022.

[88] Lasitha Weerakoon and Nikhil Chopra. Swing up control of a soft inverted pendulum with revolute base. *Proceedings of the IEEE Conference on Decision and Control*, 2021-December:685–690, 2021.

[89] Charles M. Best, Morgan T. Gillespie, Phillip Hyatt, Levi Rupert, Vallan Sherrod, and Marc D. Killpack. A new soft robot control method: Using model predictive control for a pneumatically actuated humanoid. *IEEE Robotics and Automation Magazine*, 23:75–84, 9 2016.

[90] Charles M Best, Morgan T Gillespie, Phillip Hyatt, MD Killpack, L Rupert, and V Sherrod. Model predictive control for pneumatically actuated soft robots. *IEEE Robotics & Automation Magazine*, 2(9):31, 2016.

[91] Charles M. Best, Levi Rupert, and Marc D. Killpack. Comparing model-based control methods for simultaneous stiffness and position control of inflatable soft robots. *International Journal of Robotics Research*, 40:470–493, 1 2021.

[92] Arthur S. Barbosa, Lucas Z. Tahara, and Maíra M. da Silva. Motion planning of a fish-like piezoelectric actuated robot using model-based predictive control. *JVC/Journal of Vibration and Control*, 29:411–427, 1 2023.

[93] Azadeh Doroudchi, Sachin Shivakumar, Rebecca E. Fisher, Hamid Marvi, Daniel Aukes, Ximin He, Spring Berman, and Matthew M. Peet. Decentralized control of distributed actuation in a segmented soft robot arm. *Proceedings of the IEEE Conference on Decision and Control*, 2018-December:7002–7009, 7 2018.

[94] Filippo A. Spinelli and Robert K. Katzschmann. A unified and modular model predictive control framework for soft continuum manipulators under internal and external constraints. *IEEE International Conference on Intelligent Robots and Systems*, 2022-October:9393–9400, 2022.

[95] Cosimo Della Santina, Antonio Bicchi, and Daniela Rus. 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)*, pages 6622–6629. IEEE, 2019.

[96] Minou Kouh Soltani, Sohrab Khanmohammadi, Farzan Ghalichi, and Farrokh Janabi-Sharifi. A soft robotics nonlinear hybrid position/force control for tendon driven catheters. *International Journal of Control, Automation and Systems*, 15:54–63, 2 2017.

[97] Somayeh Norouzi-Ghazbi, Ali Mehrkish, Mostafa M.H. Fallah, and Farrokh Janabi-Sharifi. Constrained visual predictive control of tendon-driven continuum robots. *Robotics and Autonomous Systems*, 145:103856, 11 2021.

[98] Guaraci Bastos Jr and Enrico Franco. Dynamic tube model predictive control for a class of soft manipulators with fluidic actuation. *International Journal of Robust and Nonlinear Control*, 35(7):2780–2799, 2025.

[99] Spencer Jensen, John L. Salmon, and Marc D. Killpack. Model evolutionary gain-based predictive control (mega-pc) for soft robotics. *2024 IEEE 7th International Conference on Soft Robotics, RoboSoft 2024*, pages 816–823, 2024.

[100] Francesco Stella, Nana Obayashi, Cosimo Della Santina, and Josie Hughes. An experimental validation of the polynomial curvature model: Identification and optimal control of a soft underwater tentacle. *IEEE Robotics and Automation Letters*, 7:11410–11417, 10 2022.

[101] Tixian Wang, Udit Halder, Heng Sheng Chang, Mattia Gazzola, and Prashant G. Mehta. Optimal control of a soft cyberoctopus arm. *Proceedings of the American Control Conference*, 2021-May:4757–4764, 5 2021.

[102] Anthony Wertz, Andrew P. Sabelhaus, and Carmel Majidi. Trajectory optimization for thermally-actuated soft planar robot limbs. *2022 IEEE 5th International Conference on Soft Robotics, RoboSoft 2022*, pages 439–446, 2022.

[103] Haley P Sanders and Marc D Killpack. Dynamically feasible trajectory generation for soft robots. In *2023 IEEE International Conference on Soft Robotics (RoboSoft)*, pages 1–8. IEEE, 2023.

[104] Maximilian Hachen, Chengnan Shentu, Sven Lilge, and Jessica Burgner-Kahrs. A nonlinear model predictive task-space controller satisfying shape constraints for tendon-driven continuum robots. *IEEE Robotics and Automation Letters*, 2025.

[105] Anup Teejo Mathew, Frederic Boyer, Vincent Lebastard, and Federico Renda. Analytical derivatives for efficient mechanical simulations of hybrid soft rigid robots. *arXiv preprint arXiv:2411.04546*, 2024.

[106] David Mayne. A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems. *International Journal of Control*, 3(1):85–95, 1966.

[107] Yuval Tassa, Tom Erez, and Emanuel Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4906–4913, 2012.

[108] Emanuel Todorov and Weiwei Li. A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 300–306. IEEE, 2005.

[109] Wilson Jallet, Antoine Bambade, Etienne Arlaud, Sarah El-Kazdadi, Nicolas Mansard, and Justin Carpentier. Proxddp: Proximal constrained trajectory optimization. *IEEE Transactions on Robotics*, 2025.

[110] Carlos Mastalli, Rohan Budhiraja, Wolfgang Merkt, Guilhem Saurel, Bilal Hammoud, Maximilien Naveau, Justin Carpentier, Ludovic Righetti, Sethu Vijayakumar, and Nicolas Mansard. Crocoddyl: An efficient and versatile framework for multi-contact optimal control. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2536–2542. IEEE, 2020.

[111] Armand Jordana, Sébastien Kleff, Avadesh Meduri, Justin Carpentier, Nicolas Mansard, and Ludovic Righetti. Stagewise implementations of sequential quadratic programming for model-predictive control. *Preprint*, 4, 2023.

[112] Roy Featherstone. *Rigid body dynamics algorithms.* Springer, 2008.

[113] James Zhu, J Joe Payne, and Aaron M Johnson. Convergent ilqr for safe trajectory planning and control of legged robots. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8051–8057. IEEE, 2024.

[114] Wilson Jallet, Nicolas Mansard, and Justin Carpentier. Implicit differential dynamic programming. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1455–1461. IEEE, 2022.

[115] Vince Kurtz and Hai Lin. Contact-implicit trajectory optimization with hydroelastic contact and ilqr. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8829–8834, 2022.

[116] Nathan J. Kong, Chuanzheng Li, George Council, and Aaron M. Johnson. Hybrid ilqr model predictive control for contact implicit stabilization on legged robots. *IEEE Transactions on Robotics*, 39(6):4712–4727, 2023.