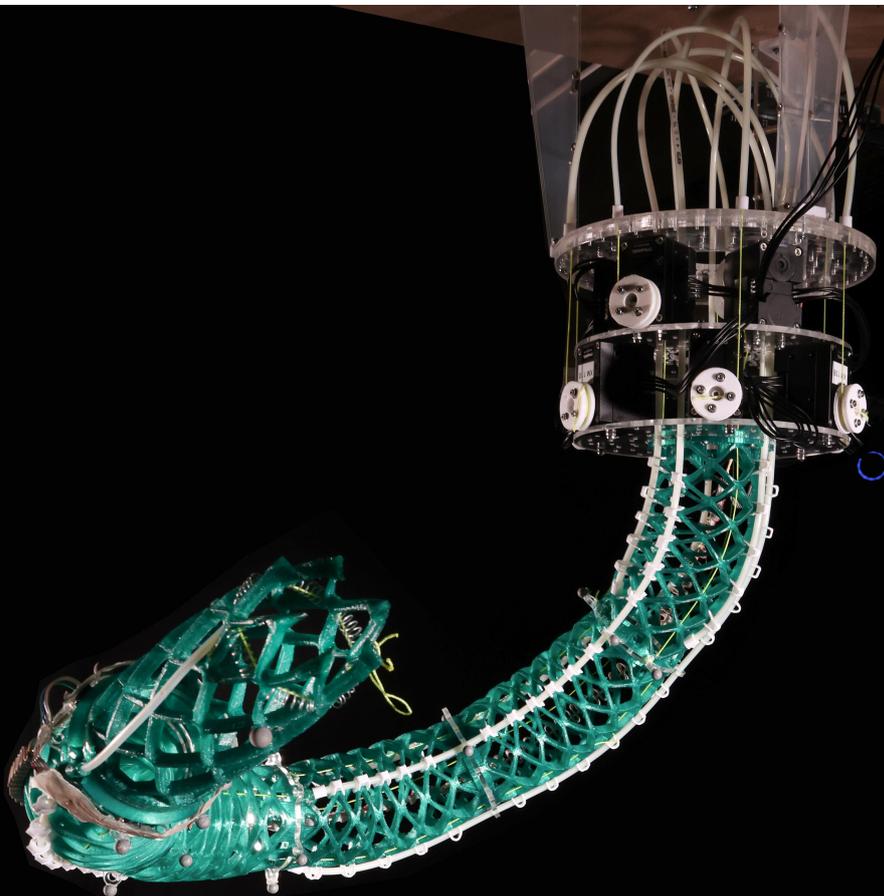


A-PCC: Adaptive inverse kinematic control with decoupled sensing- and actuation inputs

L.R. Dalenberg



A-PCC: Adaptive inverse kinematic control with decoupled sensing- and actuation inputs

Master of Science Thesis

To obtain the degree of Master of Science in robotics,
at the Delft University of Technology.

by

L.R. Dalenberg

Student number:	4933451	
Thesis supervisors:	Dr. C. Della Santina	TU Delft, (ME-CoR)
	Dr. P. Pustina	Sapienza, University of Rome, DIAG
Thesis committee:	Dr. C. Della Santina	TU Delft, (ME-CoR)
	Dr. C. Pek	TU Delft, CoR
	Dr. P. Pustina	Sapienza, University of Rome, DIAG

Cover: Michael David Mitchell. Soft, elephant trunk-like robot for close interaction with humans. © 2023 EPFL/Alain Herzog - (CC-BY-SA 4.0)

A-PCC: Adaptive inverse kinematic control with decoupled sensing- and actuation inputs

1st Lennard Dalenberg
Cognitive Robotics (CoR)
Delft University of Technology
Delft, The Netherlands

Abstract—Soft continuum robots provide a compelling solution for safe interactions with unknown and unstructured environments, owing to their compliance and infinite degrees of freedom. However, the non-linear deformation and inherent underactuation pose a persistent challenge for real-time control and state estimation. To date, continuum structures are typically discretized using a fixed-parameter kinematic model, introducing a trade-off between model accuracy and computational efficiency.

This letter presents an adaptive kinematic modeling framework to address the challenge of underactuation in a different way - not by increasing model resolution or complexity - but by making the model parametric with respect to a set of parameters that are dynamically adapted using a novel inverse kinematic adaptive controller.

We formally prove stability of the adaptive controller and validate its performance through simulations and experiments, considering setpoint reaching tasks with variable end-effector payloads. Even though the approach introduces additional challenges, in comparison to the conventional fixed-parameter models presented in literature, the proposed solution enhances shape representation, redundancy resolution, and state inference while mitigating model complexity.

Index Terms—Soft robotics, adaptive control, kinematic models, underactuation, simulation, experimental validation

I. INTRODUCTION

The growing demand for safer human-robot interaction, careful handling of delicate objects, and exploration of unknown, unstructured environments instigated a shift within the field of robotics. The nascent sub-domain of soft robotics has witnessed significant progress in recent years regarding design, modeling, and control [1]–[3]. To suit the absence of structural rigidity, design principles are adapted to confer inherent compliance on these types of robots; uniquely suited to manipulation in uncertain and dynamic environments such as minimally invasive surgery [4], crop harvesting [5], and search-and-rescue applications [6].

While their conformability is advantageous, the highly non-linear nature and inherent underactuation introduce additional challenges regarding modeling and control. The influence of actuator design has become more pronounced as they dictate the finite and controllable degrees of freedom, through actuation mechanisms that are integrated and distributed throughout the structure [7].

Moreover, conventional sensing modalities are to be re-considered as state feedback acquisition is compromised - sensor measurements must be transformed into the selected

state representation to recover the generalized coordinates that describe the system’s state within a virtual model. A synergy that explains how the chosen sensing and actuation modality influences the performance of soft continuum manipulators via modeling inaccuracies.

Researchers are developing new models to capture the intricate kinematics and complex dynamics, governed by the principles of continuum mechanics instead of those of rigid-body dynamics [8]. Within the literature, the conventional modeling methods rely on fixed-parameter models, thereby assuming a static representation of the system’s behavior in both kinematics and dynamics.

Depending on the underlying assumptions, they occupy various levels of accuracy. Lower-fidelity, computationally efficient, and analytically tractable models such as the Piecewise Constant Curvature (PCC) are widely adopted in a vast range of control applications, from inverse kinematic control [9] to closed-loop dynamic feedback control [3]. However, suffers from the rigorous assumptions of no external loads and the general validity of the constant curvature, which is too conservative for dynamically loaded tasks. Higher-fidelity models, on the other hand, based on Cosserat rod theory [10], [11] or Finite Element Methods (FEM) [12], [13], provide high accuracy at the cost of run-time complexity.

Even though feedback control is one approach to mitigate modeling inaccuracies, at the current state, it still asks for advancements in sensory feedback to assist in the development of more advanced closed-loop control envelopes. As such, a fundamental contradiction exists that a higher gain is needed to obtain good accuracy, which eradicates the desired softness [14]. In recent years, adaptive control has been explored to capture the intricacies of soft robotic morphology, environmental interactions, and task execution.

A notable trend within this literature emerges as the focus is exclusively on uncertainty in the dynamics [15]–[17]; however, this approach proves insufficient in the context of dynamically loaded tasks, where the discrepancies between modeled and actual behavior are more pronounced. The fidelity of the underlying kinematic model remains critical for accurate state estimation and effective control, as inaccuracies in the kinematics further exacerbate uncertainties in the dynamics [18]. Therefore, to this day, kinematic controllers remain the most commonly utilized controllers within the domain of model-

based control [19].

The inherent compliance and infinite degrees of freedom thus challenge traditional control and modeling methods. Yet, conventional approaches to overcome the issue of underactuation rely on increasing the complexity and resolution of the used model, revealing a gap within the literature: the absence of a modeling approach that retains simplicity while adapting to real-world deformation online to overcome the issue of underactuation.

In this work, we therefore propose an adaptive kinematic modeling framework. By making the model parametric with respect to a set of parameters, we can dynamically adapt to the system's behavior and compensate for the underactuation while preserving computational tractability. The contributions of this work can be summarized as follows:

- A novel inverse kinematic adaptive controller that dynamically adapts the parameters of the kinematic model, inherently enhancing the shape representation of the controlled structure as a by-product of the primary control objective.
- A decoupling algorithm to detach both sensing- and actuation inputs from the real-system and the corresponding virtual model; as part of a new decoupled state feedback algorithm to infer the virtual model state using an arbitrary number of sensors and model sections at once - independent of the underlying discretization as estimated by the adaptive controller.

Although the work is developed alongside the widely used PCC model due to its popularity and analytical simplicity, in principle, the approach generalizes to any more complex kinematic model that allows for a linearly parametrized structure. In that regard, the contribution is not only in dynamically improving the shape representation, state inference, or task performance, but in offering a new perspective on dealing with the inherent under-actuation in soft continuum robots.

II. THE KINEMATIC MODEL

In this work, we adopt the Piecewise Constant Curvature (PCC) model to define the kinematics of our soft continuum manipulator. Using this model, we can present the proposed solution with a reduced level of abstraction, providing a more intuitive understanding of the system's behavior.

Said model neglects all strains but one curvature strain and hypothesizes that the continuous structure can be approximated through a discrete set of extensible tangential sections with constant curvature, which are continuously differentiable [3] (see fig. 1). While different sets of parameters¹ have been used to describe the state of a constant curvature section, the adopted state parameterization in this work is based on the Δ -parametrization [20] void of unwanted discontinuities and singularities associated with the unified- and standardized frameworks.

¹Different sets of parameters have been used throughout the literature to describe a constant curvature section, for example using Denavit-Hartenburg parameters, Frenet-Serret frames, and exponential coordinates.

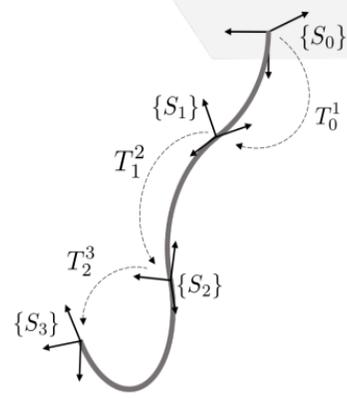


Fig. 1: Schematic overview of an arbitrary 3D curve discretized into three CC sections. The frames $\{S_i\}$ represent the attached reference frames and base frame $\{S_0\}$, with $\{T_{i-1}^i\}$ being the homogeneous transformation mappings. Image taken from [20]

Hence, in the remainder of this letter, the configuration of the i th section is thus defined by $q_i = [\Delta_{x_i} \ \Delta_{y_i} \ \delta L_i]^T \in \mathbb{R}^3$ where the vector $q \in \mathbb{R}^n$ collects the individual terms associated with each body; obtaining the homogeneous transformation from the $(i-1)$ th frame to frame i as defined in (1). In this case, $\Delta_{x,i} = \theta_i d_i \cos(\phi_i)$, $\Delta_{y,i} = \theta_i d_i \sin(\phi_i)$ and $\Delta_i = \sqrt{\Delta_{x,i}^2 + \Delta_{y,i}^2}$ where ϕ_i is the direction of bending and θ_i the angle of curvature. It is worth pointing out that the translational component of (1) is a function of both the configuration vector q_i as well as the fixed kinematic parameter $L_{0,i}$.

III. ADAPTIVE KINEMATIC CONTROLLER

To introduce adaptivity into the framework, we propose to make the model parametric with respect to unknown kinematic parameters $L_{0,i}$, i.e., the rest length of each considered body, estimated by an adaptive inverse kinematic controller. This approach presents a unique trait in that it inherently enhances the shape representation as a by-product of the primary control objective. This is particularly desirable in the context of soft robotics for accurate state inference and performing secondary objectives besides precise end-effector positioning.

Even though this set of parameters is the most intuitive and introduces unique challenges on its own, in principle, this framework can be extended to different models that allow for a linearly parametrized structure to compensate for the underactuation.

A. Linear parametrization

For the derivation of the adaptive inverse kinematic controller, we establish the following theorem:

Theorem 1. *The relationship between the task-space velocity \dot{x} and the joint-space velocity \dot{q} , defined by $\dot{x} = J(q)\dot{q}$, is linear in the set of constant kinematic parameters:*

$$\pi = \{L_{0_1}, \dots, L_{0_{n_b}}\}$$

such that we can write

$$\dot{x} = J(q)\dot{q} = Y(q, \dot{q})\pi + \Phi(q, \dot{q}) \quad (2)$$

where $Y(q, \dot{q}) \in \mathbb{R}^{3 \times n_b}$ is a known regressor matrix and $\Phi(q, \dot{q}) \in \mathbb{R}^3$ is a known function independent of π for n_b discretized sections.

Proof. Let us collect all the kinematic parameters L_{0_i} , into a vector $\pi \in \mathbb{R}^{n_b}$ to define the end-effector position $x(q, \pi) \in \mathbb{R}^3$ and orientation, given by the rotation matrix $R(q, \pi) \in SO(3)$ such that:

$$\begin{bmatrix} R(q, \pi) & x(q, \pi) \\ [0]_3 & 1 \end{bmatrix} = \prod_{i=1}^{n_b} T_{i-1}^i \quad (3)$$

where each T_{i-1}^i is to be right multiplied. Then, considering that the linear- and angular velocities of the end-effector can be defined as [21]:

$$\dot{x}(q, \dot{q}, \pi) = D_q(x(q, \pi))\dot{q} \quad (4)$$

$$S(\omega(q, \dot{q}, \pi)) = \left(\sum_{i=1}^n D_{q_i}(R(q, \pi))\dot{q}_i \right) R^T(q, \pi) \quad (5)$$

with $S(\cdot)$ denoting the skew-symmetric operator. It follows that the end-effector twist is given by the function of joint velocities:

$$\eta(q, \dot{q}, \pi) = \omega(q, \dot{q}, \pi) \oplus \dot{x}(q, \dot{q}, \pi) = J(q, \pi)\dot{q} \quad (6)$$

where $J(q, \pi) \in \mathbb{R}^{6 \times n}$ is the Jacobian.

Exercising the theory in [21], we can state that $\eta(q, \dot{q}, \pi)$ is linearly parametrizable if and only if all elements in (3) are linearly parametrizable $\forall i \in \{1, \dots, n_b\}$ - as linearly parametrizable functions are closed with respect to all the operators used to compute $\eta(q, \dot{q}, \pi)$.

For the sake of brevity, the remainder of the proof considers a single-section manipulator, i.e., $i = 1$. To that end, provided that the linear end-effector velocity is given by (4), linear parametrization is applicable if all elements in $x(q, \pi)$ can be linearly parametrized with respect to π . By expanding Δ_1 in (1) and setting $d_1 = 1$, simple algebraic steps then show

that a linear parametrization of $x(q, \pi)$ with respect to L_{0_1} can thus readily be found as:

$$x(q, \pi) = \begin{bmatrix} \frac{\delta L \Delta_x (1 - \cos(\Delta))}{\Delta^2} & \frac{\Delta_x (1 - \cos(\Delta))}{\Delta^2} \\ \frac{\delta L \Delta_y (1 - \cos(\Delta))}{\Delta^2} & \frac{\Delta_y (1 - \cos(\Delta))}{\Delta^2} \\ \frac{\delta L \sin(\Delta)}{\Delta} & \frac{\sin(\Delta)}{\Delta} \end{bmatrix} \begin{bmatrix} 1 \\ L_0 \end{bmatrix}, \quad (7)$$

concluding that (2) must hold.

The Jacobian $J(q, \pi)$ relates the rate of change of the configuration variables q to the end-effector velocities, and is obtained by taking the partial derivatives of $x(q, \pi)$ with respect to q . By inspection of $J(q, \pi)$, and the concurrent multiplication with \dot{q} ; expansion and collection of terms provides a linear factorization for each element with respect to L_{0_i} , $\forall i \in \{1, \dots, n_b\}$. To obtain the following structure for $Y(q, \dot{q})$ and π :

$$Y(q, \dot{q}) = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n_\pi} \\ a_{21} & a_{22} & \dots & a_{2n_\pi} \\ a_{31} & a_{32} & \dots & a_{3n_\pi} \end{bmatrix} \in \mathbb{R}^{3 \times n_\pi}, \quad (8)$$

with $\pi = [1 \quad L_{0_1} \quad \dots \quad L_{0_{n_b}}]^T \in \mathbb{R}^{n_b+1}$.

We can recognize that the first entry of π is constant, as the first column of $Y(q, \dot{q})$ is a known term independent of L_{0_i} . Hence, only the second element and onward - or true kinematic parameters - are estimated online to obtain:

$$\hat{\pi} = [\hat{L}_{0_1} \quad \dots \quad \hat{L}_{0_{n_b}}]^T \quad (9)$$

Accordingly, we can deconstruct the regressor matrix $Y(q, \dot{q})$ such that:

$$\dot{x} = J(q)\dot{q} = Y(q, \dot{q})\pi + \Phi(q, \dot{q}) \quad (10)$$

For the sake of space, the explicit expression for $Y(q, \dot{q})$ and $\Phi(q, \dot{q})$ are not reported here. \square

B. Derivation of the adaptive controller

By invocation of Theorem 1, we can thus write:

$$\dot{x} = J(q)\dot{q} = Y(q, \dot{q})\pi + \Phi(q, \dot{q}) \quad (11)$$

where π , $Y(q, \dot{q})$ and $\Phi(q, \dot{q})$ have been defined. Amidst kinematic uncertainty, however, the parameters of the Jacobian are uncertain and therefore:

$$\hat{\dot{x}} = \hat{J}(q, \hat{\pi})\dot{q} = Y(q, \dot{q})\hat{\pi} + \Phi(q, \dot{q}) \quad (12)$$

where $\hat{\dot{x}} \in \mathbb{R}^3$, $\hat{J}(q, \hat{\pi}) \in \mathbb{R}^{3 \times n}$ and $\hat{\pi} \in \mathbb{R}^{n_b}$ denote the estimates of the task-space velocity, the Jacobian and the set

$$T_{i-1}^i = \begin{bmatrix} 1 + \frac{\Delta_{x,i}^2}{\Delta_i^2} (\cos(\frac{\Delta_i}{d_i}) - 1) & \frac{\Delta_{x,i} \Delta_{y,i}}{\Delta_i^2} (\cos(\frac{\Delta_i}{d_i}) - 1) & -\frac{\Delta_{x,i}}{\Delta_i} \sin(\frac{\Delta_i}{d_i}) & \frac{d_i(L_{0,i} + \delta L_i)}{\Delta_i^2} \Delta_{x,i} (1 - \cos(\frac{\Delta_i}{d_i})) \\ \frac{\Delta_{x,i} \Delta_{y,i}}{\Delta_i^2} (\cos(\frac{\Delta_i}{d_i}) - 1) & 1 + \frac{\Delta_{y,i}^2}{\Delta_i^2} (\cos(\frac{\Delta_i}{d_i}) - 1) & -\frac{\Delta_{y,i}}{\Delta_i} \sin(\frac{\Delta_i}{d_i}) & \frac{d_i(L_{0,i} + \delta L_i)}{\Delta_i^2} \Delta_{y,i} (1 - \cos(\frac{\Delta_i}{d_i})) \\ \frac{\Delta_{x,i}}{\Delta_i} \sin(\frac{\Delta_i}{d_i}) & \frac{\Delta_{y,i}}{\Delta_i} \sin(\frac{\Delta_i}{d_i}) & \cos(\frac{\Delta_i}{d_i}) & \frac{d_i(L_{0,i} + \delta L_i)}{\Delta_i^2} \Delta_i \sin(\frac{\Delta_i}{d_i}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \in SE(3) \quad (1)$$

of kinematic parameters respectively. Note, indeed, that no uncertainty exists in $\Phi(q, \dot{q})$ as it strictly depends on q and \dot{q} ; so no estimation is imposed.

Then, if we let $x_d(t)$ and $\hat{\pi}(t)$ be our desired end-effector trajectory and time-varying estimate of π respectively, we can deduce the error dynamics from the position error $e = x_d - x$ and parameter estimation error $\tilde{\pi} = \pi - \hat{\pi}$:

$$\begin{aligned} \dot{e} &= \dot{x}_d - Y(q, \dot{q})(\tilde{\pi} + \hat{\pi}) - \Phi(q, \dot{q}) \\ &= \dot{x}_d - Y(q, \dot{q})\tilde{\pi} - (\hat{J}(q, \hat{\pi})\dot{q} - \Phi(q, \dot{q})) - \Phi(q, \dot{q}) \\ &= \dot{x}_d - \hat{J}(q, \hat{\pi})\dot{q} - Y(q, \dot{q})\tilde{\pi} \end{aligned} \quad (13)$$

Based on the derived error dynamics, we can now formulate an appropriate kinematic control and parameter update law:

$$\dot{q}(q, e, \hat{\pi}, t) = \hat{J}^\dagger(\dot{x}_d + Ke) \quad (14)$$

$$\dot{\hat{\pi}}(q, e, \hat{\pi}, t) = -\Gamma Y^T e \quad (15)$$

where $\hat{J}(q, \hat{\pi})$ is assumed to be full row rank with $K \succ 0$ and $\Gamma \succ 0$.

Remark 1. In the remainder of this work, the derived adaptive controller is strictly concerned with the linear component of $Y(q, \dot{q})$. However, a quick inspection of (1) shows that (5) can be linearly parametrized with respect to π due to its independence; the orientation regressor would yield a zero matrix. Hence, the adaptive controller can easily be extended to account for the end-effector orientation as well due to the independence of the adaptive law (15) with respect to the end-effector orientation.

Remark 2. To address redundancy of the the manipulator with respect to the task-vector - prevalent within the domain of soft robotics - the null-space of the $\hat{J}(q, \hat{\pi})$ can be used to optimize a secondary objective, provided that the gradient of the cost-function is of convex nature [22]:

$$\dot{q}(q, e, \hat{\pi}, t) = \hat{J}^\dagger(\dot{x}_d + Ke) + (I_n - \hat{J}^\dagger \hat{J}) \nabla_\Psi \quad (16)$$

with ∇_Ψ the negative convex function gradient.

C. Convergence and stability

While the derivation of $Y(q, \dot{q})$ introduces the known time-varying term $\Phi(q, \dot{q})$ - an attribute of the used kinematic model and consequent linear parametrization - we can observe that it holds no influence over the error dynamics in (13).

Hence, introducing the control- and adaptive law, as defined in (14) and (15), in the error dynamics (13) further simplifies the expression to obtain:

$$\dot{e} + Ke + Y\tilde{\pi} = 0, \quad (17)$$

where the set $e = 0$ can be shown to be positively invariant as (17) simplifies to $\dot{e} + Y\tilde{\pi} = 0$. More specifically, $e = 0$ implies that (15) becomes zero and therefore $\hat{\pi}$ remains constant. Since π is also constant, from $\tilde{\pi} = \pi - \hat{\pi}$ we know that $\tilde{\pi}$ and \dot{e} therefore must be too. Hence, e remains zero.

To ensure global asymptotic stability of this positively invariant set, we can consider the following Lyapunov function candidate:

$$V = \frac{1}{2} e^T e + \frac{1}{2} \tilde{\pi}^T \Gamma^{-1} \tilde{\pi} \quad (18)$$

and its time derivative:

$$\dot{V} = e^T \dot{e} + \tilde{\pi}^T \Gamma^{-1} \dot{\tilde{\pi}}, \quad (19)$$

which can be further simplified by substitution of $\dot{e} = -Y\tilde{\pi} - Ke$ and $\dot{\tilde{\pi}} = -\dot{\hat{\pi}}$ to obtain:

$$\dot{V} = -e^T Ke \quad (20)$$

The remainder of the proof then naturally follows conform Theorem I in [21], however, will be reported here for the sake of clarity. The fact that V is lower bounded implies that e and $\tilde{\pi}$ are to be bounded as $\dot{V} \leq 0$. Therefore, as $x_d(t)$ is finite, x has to be bounded. Under the assumption that the structure of the robot is such that if x is bounded, $\hat{J}(q, \pi)$ must be bounded, then so is $\hat{J}^\dagger(q, \pi)$. Hence, (14) tells us that \dot{q} is also bounded and subsequently so are $Y(q, \dot{q})$ and $\Phi(q, \dot{q})$.

Then, given that:

$$\ddot{V} = 2e^T K^2 e + 2e^T KY\tilde{\pi}, \quad (21)$$

as \ddot{V} is finite and \dot{V} is uniformly continuous, Barbalat's lemma allows us to conclude that $\lim_{t \rightarrow \infty} \dot{V} = 0$. Showing that the control- and parameter update law in conjunction with the parameter dynamics provide $\lim_{t \rightarrow \infty} e = 0$.

IV. DECOUPLING SENSING AND ACTUATION

The domain of soft robotics poses unique challenges that often prevent conventional sensing methodologies from being applied directly to infer the system's state. A common challenge is that acquired sensor measurements must be transformed into the chosen state representation to obtain the generalized coordinates that describe the system's state in a virtual model.

In addition, the adaptive discretization of the virtual model - proposed in this letter - introduces a new challenge in that it relieves the model from a direct, i.e. fixed, mathematical definition of the sensor- and actuator positions along the backbone of the structure, as is the case with fixed-parameter models presented in literature.

To infer the state of our adaptive model through sensor measurements and resolve the computation of actuation inputs, it is thus required to decouple the real system from the virtual model.

A. System decoupling

Given the continuous nature of the real system and the associated (adaptive) discretization of our virtual model, we can infer the relative locations of the sensor- and/or actuators - along the backbone of our virtual model - using a set of curvilinear material abscissae as defined on the real-system in the strain-free state.

If we assume the sensor- and/or actuator locations to be reasonably measurable, we can construct a set of curvilinear material abscissae $\mathcal{S}_{(\cdot)}$:

$$\mathcal{S}_{(\cdot)} = \{X_i \mid X_i \in [0, L_r], i \in I_{(\cdot)}\} \quad (22)$$

where L_r denotes the rest length of the real system and $I_{(\cdot)}$ the considered index set of sensors or actuators $I =$

$\{1, 2, \dots, n_{s/a}\}$. Then, to transform the elements of $\mathcal{S}_{(\cdot)}$ into a normalized space and acquiring the relative positions within the virtual model at any given time t , we utilize the time-varying estimates $\hat{\pi}(t)$ of π to construct cumulative intervals.

Hence, we obtain:

$$\hat{\pi}_j^{(c)} = \sum_{m=1}^j \hat{\pi}_m, \quad \forall j \in \{1, \dots, n_b\} \quad (23)$$

with $(\cdot)^{(c)}$ denoting the cumulative values and $\hat{\pi}_0^{(c)} = 0$. These cumulative values partition the interval in (22), such that for each $X_i \in \mathcal{S}_{(\cdot)}$ we can identify a discretized section index:

$$j_i = \min\{m \mid \hat{\pi}_m^{(c)} \geq X_i\} \quad (24)$$

where X_i lies on the interval $\hat{\pi}_{j_i-1}^{(c)} < X_i \leq \hat{\pi}_{j_i}^{(c)}$. Given the identified index j_i , we can then transform each element $X_i \in \mathcal{S}_{(\cdot)}$ into a normalized curvilinear material abscissa \tilde{X}_i using:

$$\tilde{X}_i = \frac{X_i - \hat{\pi}_{j_i-1}^{(c)}}{\hat{\pi}_{j_i}^{(c)} - \hat{\pi}_{j_i-1}^{(c)}}, \quad \text{for } \hat{\pi}_{j_i-1}^{(c)} < X_i \leq \hat{\pi}_{j_i}^{(c)}. \quad (25)$$

Here, \tilde{X}_i denotes the relative position of X_i within the interval $[\hat{\pi}_{j_i-1}^{(c)}, \hat{\pi}_{j_i}^{(c)}]$ and j_i represents the corresponding model section to which X_i belongs. Imposing this mapping on all elements in $\mathcal{S}_{(\cdot)}$ provides the transformed set:

$$\tilde{\mathcal{S}} = \{(\tilde{X}_i, j_i) \mid X_i \in \mathcal{S}_{(\cdot)}, \hat{\pi}_{j_i-1}^{(c)} < X_i \leq \hat{\pi}_{j_i}^{(c)}\} \quad (26)$$

The mapping $\mathcal{S} \mapsto \tilde{\mathcal{S}}$ establishes an eloquent method to normalize the elements in $\mathcal{S}_{(\cdot)}$ within the corresponding estimated virtual model sections $\hat{\pi}$ at time t ; decoupling the position of the sensors and actuators on the real system from the virtual model (see algorithm 1).

Algorithm 1 Decoupling algorithm

Input: $\mathcal{S}_{s/a}, \hat{\pi}$

Require: $\mathcal{S}_{s/a} \neq \emptyset, \hat{\pi} \neq \emptyset$

- 1: $\hat{\pi}^{(c)} \leftarrow [0]$ Initialize a zero-list $\hat{\pi}^{(c)}$
 - 2: **for** $j = 1$ **to** n_b **do**
 - 3: $\hat{\pi}_j^{(c)} = \sum_{m=1}^j \hat{\pi}_m$ eq. (23)
 - 4: $\hat{\pi}^{(c)} \leftarrow \hat{\pi}_j^{(c)}$ Update $\hat{\pi}^{(c)}$ for each j
 - 5: **end for**
 - 6: $\tilde{\mathcal{S}}_{s/a} \leftarrow \emptyset$ Initialize an empty set $\tilde{\mathcal{S}}_{s/a}$
 - 7: **for** $X_i \in \mathcal{S}_{(\cdot)}$ **do**
 - 8: $j_i = \min\{m \mid \hat{\pi}_m^{(c)} \geq X_i\}$ eq. (24)
 - 9: $\tilde{X}_i = \frac{X_i - \hat{\pi}_{j_i-1}^{(c)}}{\hat{\pi}_{j_i}^{(c)} - \hat{\pi}_{j_i-1}^{(c)}}$ eq. (25)
 - 10: $\tilde{\mathcal{S}}_{s/a} \leftarrow (\tilde{X}_i, j_i)$ Update $\tilde{\mathcal{S}}_{s/a}$ for each tuple
 - 11: **end for**
 - 12: **return** $\tilde{\mathcal{S}}_{s/a}$
-

Remark 3. In case any of the elements in the set $\mathcal{S}_{(\cdot)}$ extend beyond the cumulative sum $\hat{\pi}_{n_b}^{(c)}$ of the frozen estimates $\hat{\pi}$, the normalized curvilinear abscissa \tilde{X}_i will be attached to the tip of the last section.

B. State feedback

Building upon the assumption that the set \mathcal{S}_s , as defined in (22), is to be reasonably measurable for n_s -sensor(s); and the chosen sensing modality to provide the Cartesian pose $\mathcal{T}_i \in SE(3)$ for each sensor $i \in \{1, \dots, n_s\}$. Then, at any given time t , by invocation of algorithm 1 we are to obtain the transformed set (26). Enabling us to infer the Cartesian sensor poses inside our virtual model via the forward kinematic map $h(\cdot)$:

$$\bar{\mathcal{T}}_i = h(\bar{q}, \tilde{X}_i, \hat{\pi}), \quad \forall i \in \{1, \dots, n_s\} \quad (27)$$

in any generic configuration \bar{q} for the current (frozen) estimates $\hat{\pi}$ as provided by the adaptive controller at time t (see fig. 2).

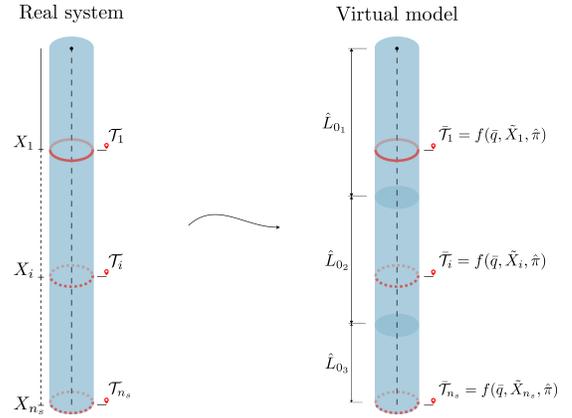


Fig. 2: Visualization of the real system, i.e., a continuum with infinite degrees of freedom, with an arbitrary number of sensors n_s along the backbone (left). The continuum presented as the proposed decoupled virtual model, comprised of discrete bodies whose kinematic parameters are adapted over time. To illustrate the state of decoupling, the section tips of the virtual model do not coincide with the sensor locations, which would be a natural choice for the fixed-parameter models found in literature.

Noting the equivalence between $\bar{\mathcal{T}}_i$ and the real-world sensor readings $\mathcal{T}_i \forall i \in \{1, \dots, n_s\}$ thus allows us to infer the new state of our virtual model through decoupled numerical inverse kinematics; driven by the estimated Jacobian matrix $\hat{J}_i(\bar{q}_k, \tilde{X}_i, \hat{\pi})$ to each sensor along the backbone of the virtual model, the 'inferred' state at the previous iteration \bar{q}_{k-1} , the current estimates $\hat{\pi}$ and the task-error of each sensor \mathcal{V}_{b_i} .

More specifically, we seek to iteratively obtain the optimal virtual model state, i.e., the generalized coordinates q for which the norm of the task error is minimized at the measured locations. The need for the decoupling becomes evident as the adaptive discretization continuously influences the relative sensor locations along the backbone.

By leveraging the set $\tilde{\mathcal{S}}_s$, we can thus retrieve the discretized section index j_i and corresponding normalized curvilinear material abscissa \tilde{X}_i for each individual sensor to directly evaluate its pose $\bar{\mathcal{T}}_i$ as defined in (3). This method can be conveniently formulated to infer the virtual model state using

an arbitrary number of sensors and model sections at once, independent of the underlying discretization $\hat{\pi}$ as estimated by the adaptive controller (see fig. 3).

Using composition, both a single task error vector and estimated Jacobian matrix can be constructed; each element corresponding to a relative sensor pose inside the virtual model:

$$\hat{J}_b(\bar{q}_k, \tilde{X}, \hat{\pi}) = \begin{bmatrix} \hat{J}_{b_1}(\bar{q}_k, \tilde{X}_1, \hat{\pi}) \\ \vdots \\ \hat{J}_{b_{n_s}}(\bar{q}_k, \tilde{X}_{n_s}, \hat{\pi}) \end{bmatrix} \in \mathbb{R}^{6n_s \times n} \quad (28)$$

for n_s -sensors. Here, we adopt the body Jacobian to each relative sensor defined as:

$$J_{b_i}(\bar{q}_k, \tilde{X}_i, \hat{\pi}) = [\omega \quad v]^T \in \mathbb{R}^{6 \times n}, \quad (29)$$

and the task-error vector:

$$\mathcal{V}_{b_i} = \log(\bar{\mathcal{T}}_i^{-1}(\bar{q}_k, \tilde{X}_i, \hat{\pi})\mathcal{T}_i) \in \mathbb{R}^6. \quad (30)$$

Note that we compute the Jacobian with respect to all configuration variables q to ensure a consistent dimensional structure across all sensors. Hence, we can solve for the virtual model state considering all sensor pose(s) at once through [23]:

$$\bar{q}_{k+1} = \bar{q}_k + \hat{J}_b^{-1}(\bar{q}_k, \tilde{X}, \hat{\pi})\mathcal{V}_b \quad (31)$$

While the decoupled state feedback algorithm is presented considering *pose* feedback (see algorithm 2) - to alleviate the inferred model state of orientational ambiguity - it can easily be reformulated to consider the translational component t_i of each sensor \mathcal{T}_i only:

$$\bar{q}_{k+1} = \bar{q}_k + \hat{J}_p^\dagger(\bar{q}_k, \tilde{X}, \hat{\pi})(t - \bar{t}(\bar{q}_k, \tilde{X}, \hat{\pi})) \quad (32)$$

providing a more computationally friendly approach.

An observant reader may point out that besides the inferred configuration \bar{q} , the kinematic regressor matrix (8) exhibits a dependency on \dot{q} . Indeed, our adaptive law requires us to approximate the configuration velocity \dot{q} . To do so, we impose a Savitzky-Golay type filter, convolving a fitted polynomial over the previous N -number of measured configurations - including the newly inferred configuration \bar{q} , taking its time derivative and evaluating it at time t . A visual representation of this approach is provided in fig. 4.

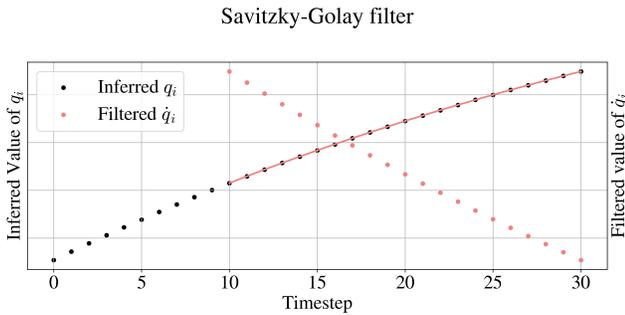


Fig. 4: Twin-axis plot highlighting the Savitzky-Golay filter to approximate \dot{q}_i via the last N inferred states q_i associated with the previous t timesteps.

Algorithm 2 Decoupled state feedback algorithm

Input: $\hat{\pi}, \bar{q}_{t-1}, \mathcal{S}_s, \mathcal{T}, N, LinThresh, AngThresh$
Require: $\mathcal{S}_s \neq \emptyset, \hat{\pi} \neq \emptyset, \mathcal{T} \in \mathbb{R}^{4n_s \times 4}$

```

1:  $\tilde{\mathcal{S}}_s \leftarrow \mathcal{S}_s$  Run algorithm (1)
2:  $\hat{J}_b \leftarrow [0] \in \mathbb{R}^{6n_s \times n}$  Initialize a zero matrix
3:  $\mathcal{V}_b \leftarrow [0] \in \mathbb{R}^{6n_s}$  Initialize a zero vector
4: while ( $e_{ang} > Thresh_1$  &  $e_{lin} > Thresh_2$ ) &  $k \leq N$ 
   do:
5:   if  $k = 0$  then
6:      $\bar{q}_k \leftarrow \bar{q}_{t-1}$  Seed numerical IK
7:   else
8:      $\bar{q}_k \leftarrow \bar{q}_{k-1}$  Seed numerical IK
9:   end if
10:  for  $(\tilde{X}_i, j) \in \tilde{\mathcal{S}}_s$  do
11:     $\bar{\mathcal{T}}_i = h(\bar{q}_k, \tilde{X}_i, \hat{\pi})$ 
12:     $\hat{J}_{b_i} = \hat{J}_b(\bar{q}_k, \tilde{X}_i, \hat{\pi})$ 
13:     $\mathcal{V}_{b_i} = \log(\bar{\mathcal{T}}_i^{-1}(\bar{q}_k, \tilde{X}_i, \hat{\pi})\mathcal{T}_i)$ 
14:     $\hat{J}_b \leftarrow \hat{J}_{b_i}$  Update  $\hat{J}_b$  for each sensor
15:     $\mathcal{V}_b \leftarrow \mathcal{V}_{b_i}$  Update  $\mathcal{V}_b$  for each sensor
16:  end for
17:   $\bar{q}_{k+1} \leftarrow \bar{q}_k + \hat{J}_b^{-1}(\bar{q}_k, \tilde{X}, \hat{\pi})\mathcal{V}_b$  eq. (31)
18: end while
19: return  $\bar{q}, e_{ang}, e_{lin}$ 

```

Remark 4. The numerical IK is initialized at time t with \bar{q}_{t-1} as the initial guess such that $\bar{q}_0 = \bar{q}_{t-1}$.

C. The dynamic model

To build the equations of motion (EOM) we consider the introduced kinematic model and the Euler-Lagrange equation:

$$\frac{d}{dt} \left(\frac{\partial L(q, \dot{q})}{\partial \dot{q}} \right)^T - \left(\frac{\partial L(q, \dot{q})}{\partial q} \right)^T = \tau(q, u) \quad (33)$$

with the Lagrangian and generalized forces acting on the body defined as:

$$\begin{cases} \mathcal{L}(q, \dot{q}) & := \mathcal{T}(q, \dot{q}) - \mathcal{U}(q) \\ \tau(q, u) & := B(q)u + \tau_{ext} \end{cases} \quad (34)$$

where $\mathcal{T}(q, \dot{q})$ and $\mathcal{U}(q)$ are the total kinetic, and potential energy respectively. The actuation inputs u and external forces f_{ext} perform work on q through the generalized actuation matrix $B(q) \in \mathbb{R}^{n \times n_a}$ and Jacobian $J(q)$ respectively, where $J(q)$ has been defined. It can be shown that this formulation, in coherence with the introduced kinematic model, yields the equation of motion [3]:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + D(q)\dot{q} + K(q) = \tau(q, u) \quad (35)$$

where $M(q) \in \mathbb{R}^{n \times n}$ and $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ are the inertia matrix and the Coriolis/centrifugal terms respectively. $G(q) \in \mathbb{R}^n$ is the gravitational force, and $D(q)\dot{q} \in \mathbb{R}^{n \times n}$ and $K(q) \in \mathbb{R}^{n \times n}$ collect the damping and stiffness forces.

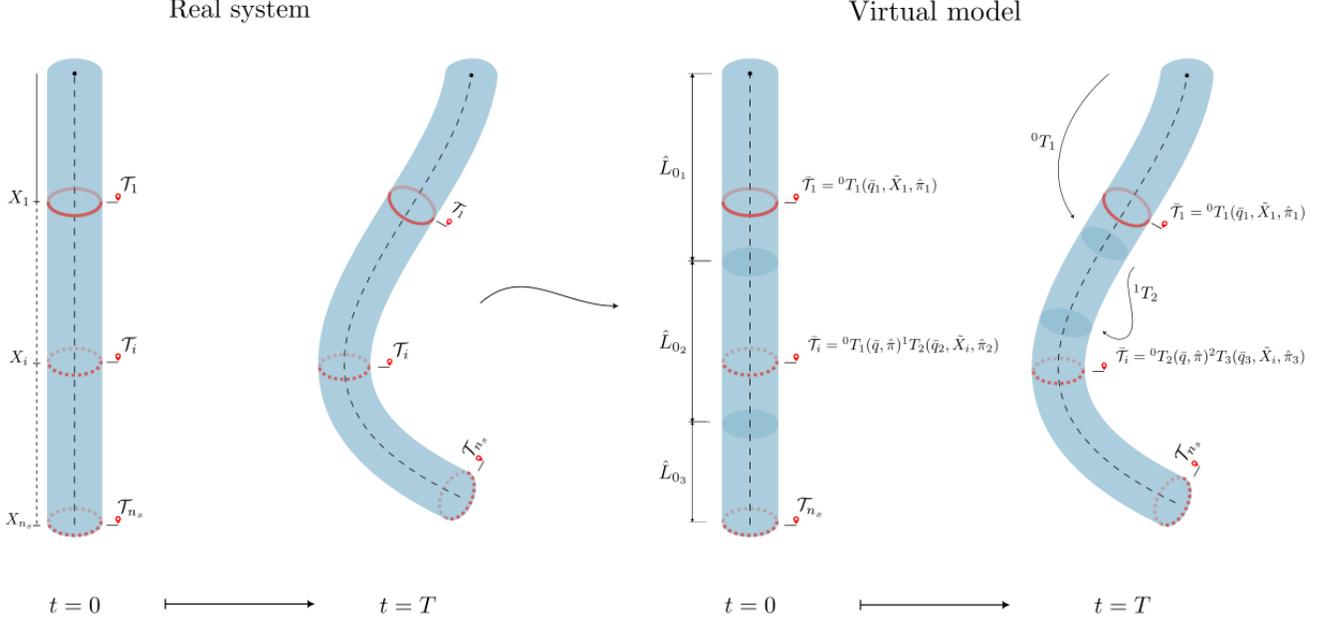


Fig. 3: The time-wise evolution of the real system, i.e., a continuum with infinite degrees of freedom, with an arbitrary number of sensors (or actuators) along the backbone. (left) The continuum represented as the proposed decoupled virtual model, highlighting how the discretization evolves over time, opposing the conventional fixed-parameter models, and the influence on state inference. Note that for the sake of interpretation, the sensor (or actuator) locations do not coincide with the section tips of the virtual model in the initial state. (right)

The Coriolis matrix $C(q, \dot{q})$ is acquired through the mass matrix using Christoffel symbols, where the element of $C(q, \dot{q})$ at index $C_{i,j}$ is given by:

$$[C(q, \dot{q})]_{ij} = \sum_{k=1}^n \Gamma_{ijk}(q) \dot{q}_k \quad (36)$$

with Γ defined as Christoffel symbols of the first kind:

$$\Gamma_{ijk}(q) = \frac{1}{2} \left[\frac{\partial M_{ij}}{\partial q_k} + \frac{\partial M_{ik}}{\partial q_j} - \frac{\partial M_{jk}}{\partial q_i} \right], \quad (37)$$

The gravitational- and stiffness forces naturally follow from the associated potential energy governing the system as:

$$G(q) = \frac{\partial U_g}{\partial q}, \quad K(q) = \frac{\partial U_k}{\partial q} \quad (38)$$

D. Model of actuation

To decouple the generalized actuation matrix $B(q)$ - based on the Geometric Variable-Strain approach - let us first briefly introduce the Cosserat theory². Here, a structure is modeled as a continuous stack of infinitesimal microsolds, parametrized through the familiar curvilinear material abscissa $X \in [0, L]$ with L the length of the structure.

²A generalization of the Kirchoff rod theory to address all modes of deformation that present itself in slender structures subject to external loading such as bending, torsion, stretching and shearing - based on an analogy between continuum mechanics and rigid-body kinematics.

At any given time t , the configuration of a microsoid is then defined by the position vector to a moving frame of directors $t(X) \in \mathbb{R}^3$ with an orientation $R(X) \in SO(3)$ relative to the world frame. Hence, we can define the configuration of the manipulator via a curve:

$$g(\cdot) : X \rightarrow g(X) = \begin{pmatrix} R(X) & t(X) \\ [0]_3 & 1 \end{pmatrix} \in SE(3), \quad (39)$$

where the partial derivative of (39) with respect to X , hereinafter denoted by $(\cdot)'$, provides the strain state in the form of a twist vector field along $g(\cdot)$ [11]:

$$X \rightarrow \hat{\xi}(X) = g^{-1} \frac{\partial g}{\partial X} = g^{-1} g' \in \mathfrak{se}(3). \quad (40)$$

The strain twist of each cross-section in the local body frame is then defined by:

$$\hat{\xi} = \begin{pmatrix} \tilde{k} & p \\ [0]_3 & 0 \end{pmatrix} \in \mathfrak{se}(3), \quad \xi = (\hat{\xi})^\vee = (k^T, p^T)^T \in \mathbb{R}^6 \quad (41)$$

with $\widehat{(\cdot)}$ the isomorphism between the twist vector and matrix representation, and $(\cdot)^\vee$ serving as its inverse. Furthermore, the operator (\cdot) refers to the isomorphism between 3D vectors and their corresponding skew-symmetric matrix representations, with $k(X) \in \mathbb{R}^3$ and $p(X) \in \mathbb{R}^3$ the angular and linear strains respectively.

Building upon these definitions, for a general routing of n_a threadlike actuators, it has been shown [11] that by projection

of the continuous model onto the space of generalized coordinates, the following expression for the generalized actuation matrix can be derived:

$$B(q) = \int_0^L \Phi_\xi^T \Phi_a(q, X) dX \in \mathbb{R}^{n \times n_a}. \quad (42)$$

which relies on the actuation matrix $\Phi_a(q, X) \in \mathbb{R}^{6 \times n_a}$ and the discretization of the strain field onto a basis of n -vector functions:

$$\xi(X) = \Phi_\xi(X)q + \xi^*(X) \quad (43)$$

Here, $\Phi_\xi(X) \in \mathbb{R}^{6 \times n}$ describes a matrix function whose columns form the basis of the strain field, $q \in \mathbb{R}^n$ is the familiar vector of generalized coordinates in the considered base and $\xi^*(X)$ is the reference strain computed in the stress-free configuration. For the sake of space, the explicit definitions of $\Phi_a(q, X) \in \mathbb{R}^{6 \times n_a}$ and $\Phi_\xi(X) \in \mathbb{R}^{6 \times n}$ have been omitted as the focus of this work is not on the derivation of the model of actuation, but rather on decoupling it.

From here, the decoupled model of actuation can be derived through the superposition of the individual bodies that compose our manipulator. Based on this notion, it follows that each discretized body has an associated local strain field $\xi_i(X_i)$, basis $\Phi_{\xi_i}(X_i)$ and actuation matrix $\Phi_{a_i}(q_i, X_i)$ which describes the influence of each tendon traversing the considered body. Allowing us to rewrite (42) as:

$$B(q) = \underbrace{\int_0^{L_i} \Phi_{\xi_i}^T \Phi_{a_i} dX_i}_{B_i} + \dots + \underbrace{\int_0^{L_n} \Phi_{\xi_n}^T \Phi_{a_n} dX_n}_{B_n} \quad (44)$$

where $\forall i \in \{1, \dots, n_b\}$ and $L_i \in \mathbb{R}^+$ the length of the considered body.

The adaptive discretization introduces, however, another caveat as the structure of $B(q)$ is unknown beforehand; the underlying discretization is time dependent (see fig. 3). To alleviate this issue, we decouple tendon routing by reintroducing the normalized set \mathcal{S}_a , as defined in (22). Providing direct access to the desired integration intervals to obtain the estimated generalized actuation mapping matrix at any given time t :

$$\hat{B}(q) = [\hat{B}_1(q) \quad \dots \quad \hat{B}_{n_b}(q)]^T \quad (45)$$

where

$$B_i = \int_0^{L_i} \dots dX_i \longrightarrow \hat{B}_i = \int_0^{\tilde{X}_i \hat{L}_i} \dots dX_i \quad (46)$$

In short, the decoupled generalized actuation matrix can thus be constructed by considering the body-dependent contributions with appropriate integration intervals, scaled with the normalized curvilinear material abscissae to properly define the relative position of the tendons.

By making a preemptive symbolic assumption that all tendons traverse the entire structure, we can efficiently evaluate (45) via nullification of the matrix elements using the acquired set \mathcal{S}_a online (see fig. 5). It is worth pointing out that under the PCC-hypothesis, $B(q)$ becomes constant. However, due to

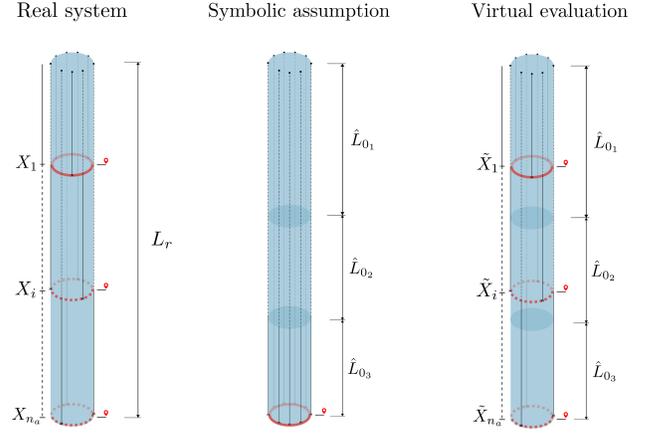


Fig. 5: Visualization of the real system, i.e. a continuum with infinite degrees of freedom, for an arbitrary number of tendon-racks k along the backbone and the corresponding curvilinear material abscissa set $\mathcal{S}_{(\cdot)} = \{X_i | X_i \in [0, L_r], i \in I_{(\cdot)}\}$ (left). The consequent pre-emptive symbolic virtual model assumption of all tendons traversing the entire structure with $\tilde{\mathcal{S}} = \{(1, 3) | X_i \in \mathcal{S}_{(\cdot)}, \hat{\pi}_2^{(c)} < X_i \leq \hat{\pi}_3^{(c)}\}$ (middle) and an example of the corresponding online evaluated decoupled state (right).

the adaptive discretization, this no longer holds as it becomes a function of the estimated parameters, i.e. $\hat{B}(\hat{\pi})$.

Now given that the length of a threadlike actuator L_{c_i} can be expressed as [11]:

$$L_{c_i} = \int_0^L \Phi_{a_i}^T \left(\xi + \begin{bmatrix} 0_3 \\ d' \end{bmatrix} \right) dX, \quad (47)$$

with Φ_{a_i} the column associated with the i th actuator, we obtain a tendon-length differential by comparing L_{c_i} to the reference length in the unstressed state. Which, by using the generalized mapping matrix $\hat{B}(\hat{\pi}_k)$ can be expressed as:

$$\begin{aligned} \hat{y}_i &= \int_0^{\tilde{X}_i \hat{L}_i} \Phi_{a_i}^T \Phi_\xi dX q + \int_0^{\tilde{X}_i \hat{L}_i} (\Phi_{a_i} - \Phi_{a_i}^*)^T \left(\xi^* \begin{bmatrix} 0_3 \\ d'_i \end{bmatrix} \right) dX \\ \hat{y}_i &= \hat{B}(\hat{\pi})_i^T q \end{aligned} \quad (48)$$

for $d'_i = 0$ due to structural design. Collecting the individual differentials in a single vector simplifies to the following actuator kinematics:

$$\hat{y} = \hat{B}(\hat{\pi})^T q \quad (49)$$

V. SIMULATION RESULTS

In this section, we validate the efficacy of the proposed solution through both kinematic and dynamic simulations. As the manipulator is redundant to the task, a secondary objective is introduced through the null-space term, substantiated in remark 2, to constrain the solution space of the (non-)adaptive controller below base, i.e., $z \leq 0$. Hence, the null-space cost

function is formulated as the Euclidean norm of the section-wise elongations:

$$\Psi(q) = \|\delta_L\|_2 = \sqrt{\sum_{i=1}^{n_b} (\delta_{L_i})^2}, \quad (50)$$

ensuring a smooth and differentiable function to obtain the controller formulation as defined in (16). Besides constraining the solution space, the incorporation of a secondary objective enables the evaluation of adaptive controller's performance in utilizing kinematic redundancy - a desirable capability especially in soft robotic systems.

A. Kinematic simulation

For the kinematic simulation, we consider the nominal case $n_{b_t} = n_{b_v}$, i.e., the *true system* encapsulates a discretized representation that comprises the same number of bodies as the *virtual model*. The control architecture is shown in fig. 6 and is designed such that the control inputs for the *real system* are generated using the state of the *virtual model*, while sensor measurements are mimicked using the forward kinematic map $h(\cdot)$ on the true state; allowing us to evaluate the decoupled state feedback algorithm (2) and the adaptive controller simultaneously.

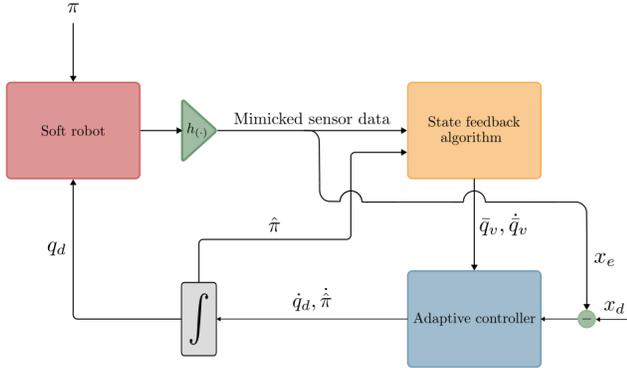


Fig. 6: The control architecture used to evaluate the convergence and performance of the adaptive framework under nominal conditions.

The *true system* is initialized with $n_{b_t} = 3$ and the kinematic parameters $\pi = [0.105 \ 0.255 \ 0.24] m$, where the sensors are 'attached' at the distal ends each body. Hence, defining the set of curvilinear material abscissae (22) as $\mathcal{S}_s = \{0.105, 0.36, 0.60\}$ in the strain-free state. Furthermore, the initial estimates for the adaptive controller are set to $\hat{\pi} = [0.12 \ 0.28 \ 0.27] m$ with the linear- and angular error thresholds $1e^{-4} m$ and $5e^{-3} rad$ for the decoupled state feedback algorithm respectively, considering a maximum number of $N = 7$ iterations.

The adaptive controller is subjected to a trajectory tracking task for a run-time of $T = 20 s$ with the trajectory $r(t)$ defined as:

$$r(t) = r_0 + A \cdot \sin(\omega t), \quad (51)$$

where

$$r_0 = [0 \ 0 \ -0.5]^T \text{ and } A = [0.5 \ 0.3 \ 0.1]^T$$

for an angular frequency of $\omega = 0.2\pi$. The inverse kinematic controller gain is set to $K = I_3$ with $\Gamma = \text{diag}(90, 42.5, 35)$ for the adaptive law.

The obtained simulation results are provided in fig. 7 as a four-way panel. Here, sub-figure 7a emulates a visual of the time-wise evolution of the controlled system for the first three seconds of simulation, i.e., $T \in [0, 3]$, with the leftmost plot showing the *true system*, while the center plot presents the *virtual model* used to control it.

From a visual perspective, the adaptive discretization and hence decoupled nature become evident. In the initial stages, $\pi \neq \hat{\pi}$, therefore sensor measurements on the *true system*, as indicated by the red markers in the center plot, do not align with the section tips of the *virtual model*. However, over time, the adaptive controller adjusts the discretization such that the *virtual model* attains a perfect reconstruction of the controlled structure.

This observation is substantiated in the remaining panels 7b and 7c, which display the end-effector error and the total length of both the *true system* and *virtual model*, respectively. In addition, 7d introduces the shape reconstruction error to quantify the shape representation of the *virtual model* with respect to the *true system*:

$$\mathcal{E}_{shape} = \frac{1}{N} \sum_{i=1}^N \|p_i^{true} - p_i^{approx}\|, \quad (52)$$

for $N = 500$ points $p_i^{true}, p_i^{approx} \in \mathbb{R}^3$ uniformly sampled along the backbones.

Based on panel 7b and 7d, it becomes evident that over time, the end-effector error(s) as well as the shape reconstruction error converge to zero during the trajectory tracking task, confirming our visual observation. This is intuitively understood by taking a closer look at the total length of the *virtual model*. As shown in fig. 7c, the length of our *virtual model* converges to the length of the *true system*, while the kinematic parameters also show full convergence in fig. 8; highlighting the relationship of the body-wise elongation inferred through the decoupled state algorithm and associated (estimated) rest length through $L_i = L_{0_i} + \delta L_i$.

Whilst no formal guarantee can be given for the convergence of estimated the kinematic parameters to the true values, it can thus be shown that for appropriately chosen gains the kinematic parameters can converge to the true state; i.e. $\|\hat{\pi}\| \rightarrow 0$ as $t \rightarrow \infty$.

In addition to the adaptive controller, the decoupled state feedback algorithm also shows increased performance as the shape of the *virtual model* gravitates closer to the *true system*. Fig. 9 highlights how both the linear- and angular sensor error norms, as returned by the decoupled state feedback algorithm, tend to full convergence over time. An important notion, whose significance shall become more pronounced in section V-B.

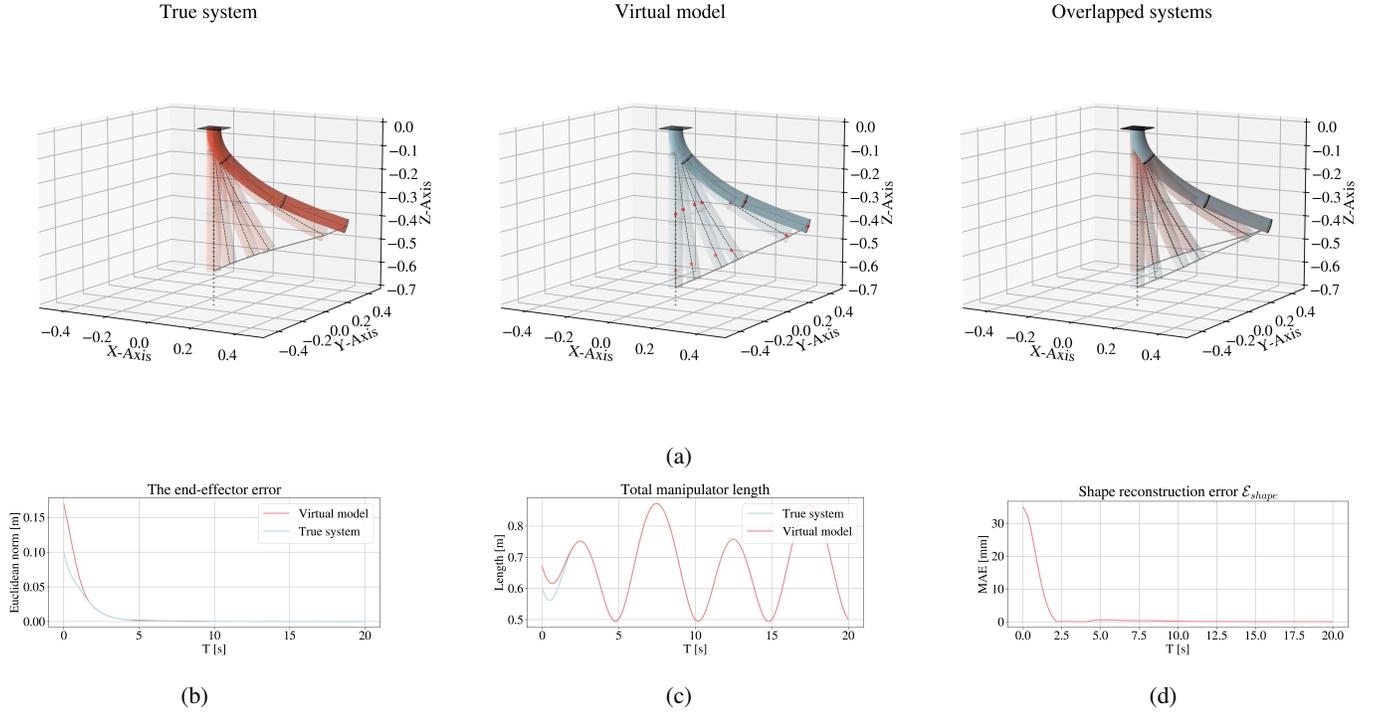


Fig. 7: A four-way panel highlighting the adaptive controller performance. (a) Emulates a visual of the time-wise evolution of the controlled system for the first three seconds of simulation, i.e $T \in [0, 3]$. The leftmost figure is the *true system* controlled by the adaptive controller based on the *virtual model* depicted in the center plot. The right-side image shows an overlap of the *true system* and *virtual model* to visually showcase the convergence of $\|\tilde{\pi}\| \rightarrow 0$. The red dots in the center image represent the sensor locations and thus section tips of the *true system*, indicating a clear initial decoupled state. (b) The evolution of the end-effector error of the *true system* over time. (c) A comparison of the total manipulator length for both the *true system* and the *virtual model*. (d) The shaped reconstruction error (52) over time.

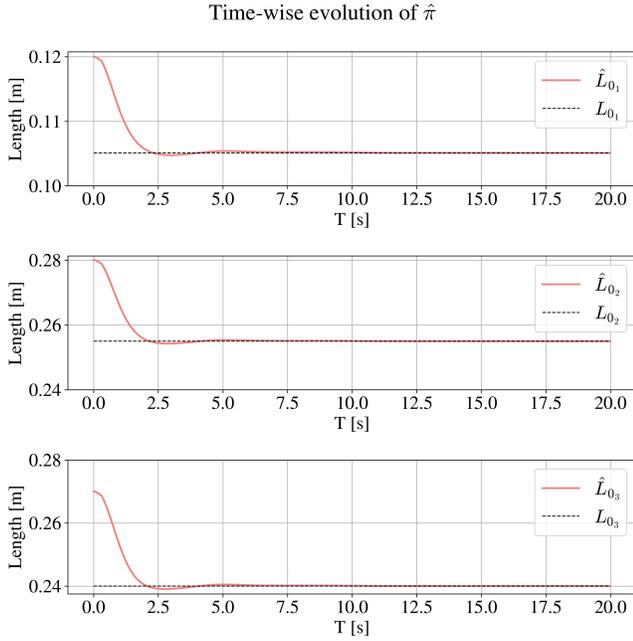


Fig. 8: The time-wise evolution of the estimated kinematic parameters $\hat{\pi}$ of the *virtual model* converging to the true values $\pi = [0.105 \ 0.255 \ 0.24]$ m in the nominal case.

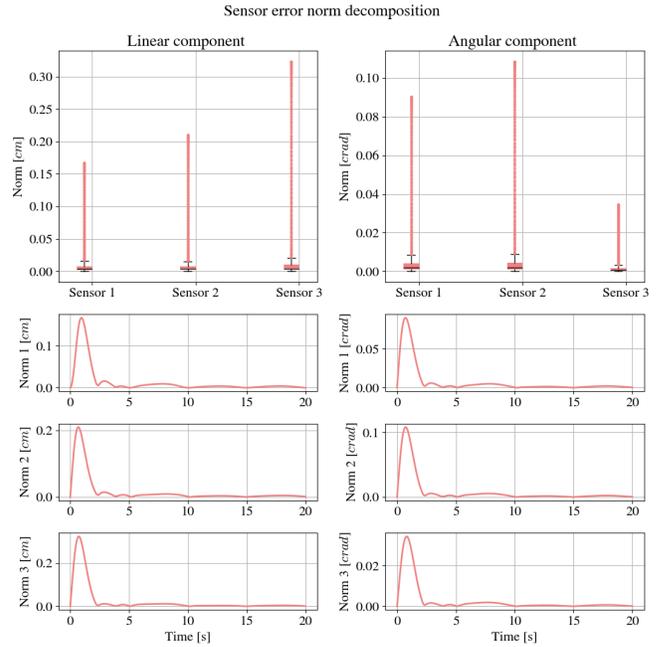


Fig. 9: The sensor error norms of the decoupled state feedback algorithm decomposed into angular and linear components for the adaptive controller.

Simulation run I - Robot I + Position feedback only:

The first simulation considers the decoupled state feedback algorithm to run on *positional* sensor measurements only, with the inverse kinematic controller gain set to $K = 1.8I_3$ for both the standard- and adaptive controller. The latter using $\Gamma = 40.5I_3$ for the adaptive law defined in (14).

As shown in fig. 11 the performance degradation of the adaptive controller is significantly better in comparison to the standard controller with increasing payloads across the workspace. The shape reconstruction error \mathcal{E}_{shape} , as defined in (52), is reduced on average by 28% decreasing from 1.82 mm to 1.31 mm. Moreover, both the minimum and maximum mismatch are decreased by 7% (from 0.19 mm to 0.17 mm) and 21.7% (from 3.32 mm to 2.60 mm) as indicated by the shaded area.

Under the adaptive control strategy, the shape of the *virtual model* thus aligns significantly more closely with the controlled *true system*.

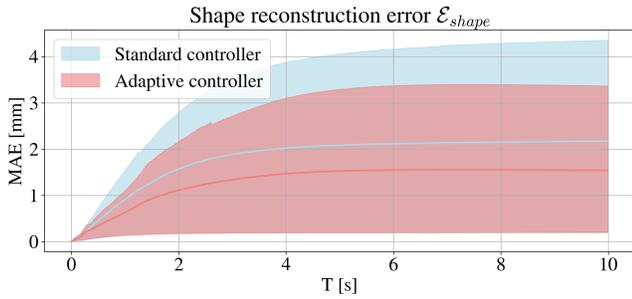


Fig. 11: A comparison of the shape reconstruction error \mathcal{E}_{shape} across all simulations for both the adaptive- and standard control strategy.

In addition, based on the data presented in fig. 12 the adaptive controller also demonstrates a substantial performance increase on the secondary task-objective defined in (50); lowering the cost-function on average by 57.9% from 4.61 cm^2 to 1.94 cm^2 . Again, with a more bounded performance profile as the lower-bound is reduced by 91.2% (from 2.71 mm to 0.24 mm) and the upper-bound by 27.6% from 8.45 mm to 6.11 mm.

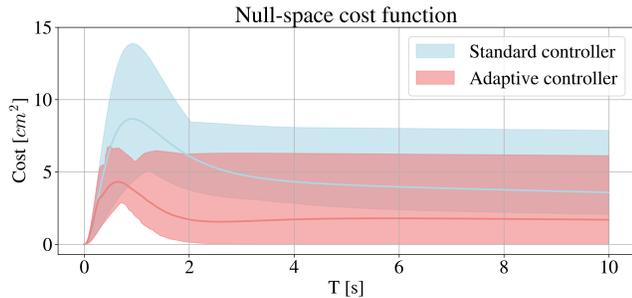


Fig. 12: A comparison of the null-space cost function between the adaptive- and standard controller.

As one might notice, unlike with the kinematic simulation $\mathcal{E}_{shape} \not\rightarrow 0$; meaning that perfect shape reconstruction is not attained between the *true system* and *virtual model*. However, this behavior can be expected as there is a theoretical limit on how well a lower resolution model can represent the shape of a higher resolution system.

Furthermore, the evolution of $\hat{\pi}_v$ tells a similar story as we look at fig. 13. In the initial stages of simulation, the updates are rather stagnant before enveloping into fairly aggressive updates and smoothing out once the end-effector position of the true system is getting close to x_d . This is intuitively understood by revisiting (15), which tells us that as $e \rightarrow 0$ and/or \dot{q} gets smaller, the adaptive law barely updates.

In other words, there is no incentive to drive adaptive law once the task error is nearing convergence.

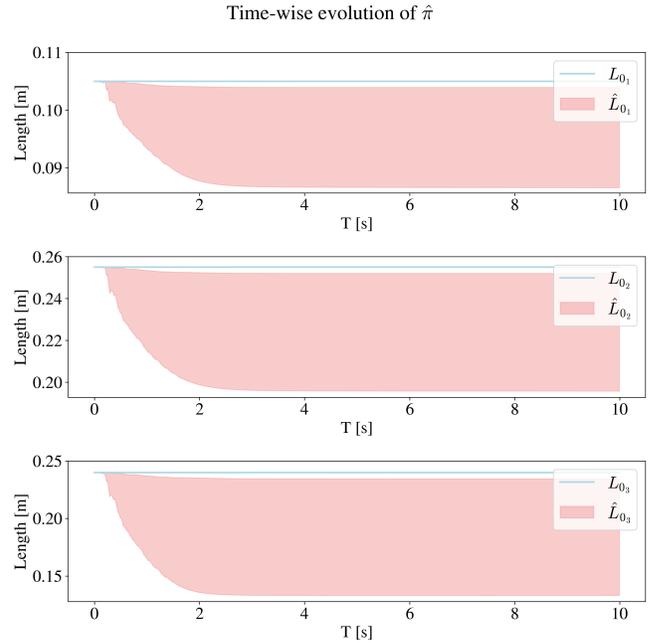


Fig. 13: A visual overview of the evolution of the adaptive kinematic parameters $\hat{\pi}_v$ and fixed parameters π_v

Simulation run II - Robot I + Full pose feedback: The second simulation considers the (decoupled) state feedback algorithm to utilize the sensor *pose* setup, alleviating the inferred virtual model state from orientational ambiguity. The inverse kinematic controller gain remains identical at $K = 1.8I_3$ for both the standard- and adaptive controller, with the latter using $\Gamma = 24.5I_3$ for the adaptive law defined in (14).

Before proceeding to the average performance, we first briefly introduce a single run from the broader set of simulations to provide a more intuitive understanding of the implications. The run considers the setpoint $x_d = [0.325, 0.15, -0.275]$ with a payload of $m = 0.5kg$, to highlight the performance degradation for increasing deformation complexity under both control strategies.

Figure 14 provides a side-by-side, dual-angle, stroboscopic plot of the controlled system and the associated *virtual model*.

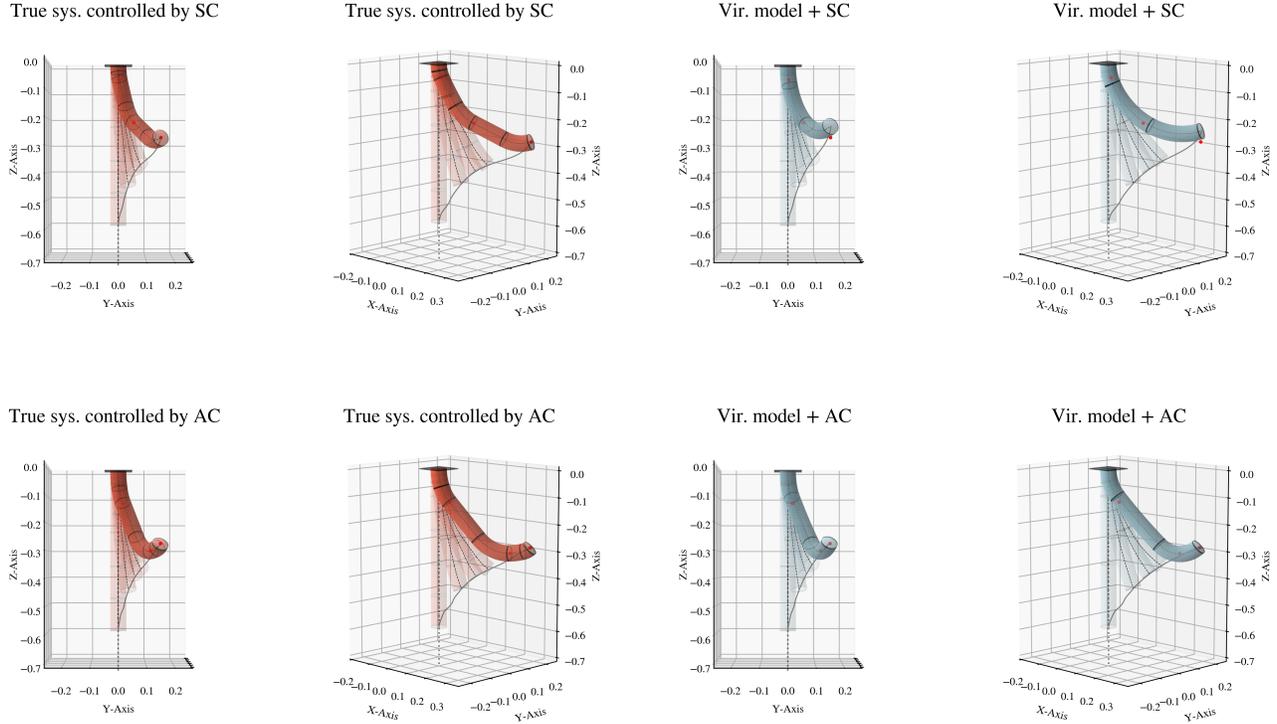


Fig. 14: A side-by-side, dual angle, stroboscopic plot as a visual comparison on the evolution of the *true system* carrying a $0.50kg$ payload to the setpoint $x_d = [0.325, 0.15, -0.275]$; controlled with the standard and adaptive controller on the first and second row, respectively.

Here, one notices a clear distinction in performance; the mismatch between the *true system* and corresponding *virtual model* is very different between the two control strategies. In the adaptive case, the *virtual model* neatly aligns with the *true system* while in the non-adaptive case it does not. To further quantify this, \mathcal{E}_{shape} is lowered by 45.8% from $12.33mm$ to $6.68mm$ during task execution when adopting the adaptive strategy (see fig. 15). Moreover, in the adaptive case, the

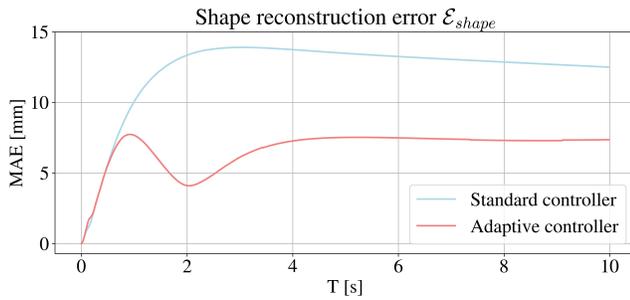


Fig. 15: A comparison of the shape reconstruction error \mathcal{E}_{shape} between the adaptive- and standard controller on the specified run.

sensor- and actuator locations on the *true system* (denoted with the red markers) coincide with the backbone of the *virtual model*, unlike for the standard case.

Indeed, for the fixed-parameter (or standard) model, the sensors and actuator locations should coincide with the section tips of the *virtual model* as was defined in the initial state. However, in contrast to simulation I, the decoupled state feedback algorithm now considers *pose* measurements, which paints an interesting picture if we reconsider fig. 9. It was shown that as $\hat{\pi} \rightarrow \pi$, and the *virtual model* converged to the *true state*, the decoupled state feedback algorithm tended to converge as well.

In other words, the inferred state of the *virtual model* became more accurate over time as the kinematic parameters converged to the true values. However, when a higher resolution system, i.e., the *true system*, is controlled with the standard controller, fig. 16 shows opposite behavior.

As the deformation of the structure becomes increasingly complex over time, the decoupled feedback algorithm provides less accurate inferred states of the *virtual model*. Eluding to the fact that while orientational ambiguity is relieved, the deformation becomes too complex to be accurately captured with the standard (fixed-parameter) PCC model when one

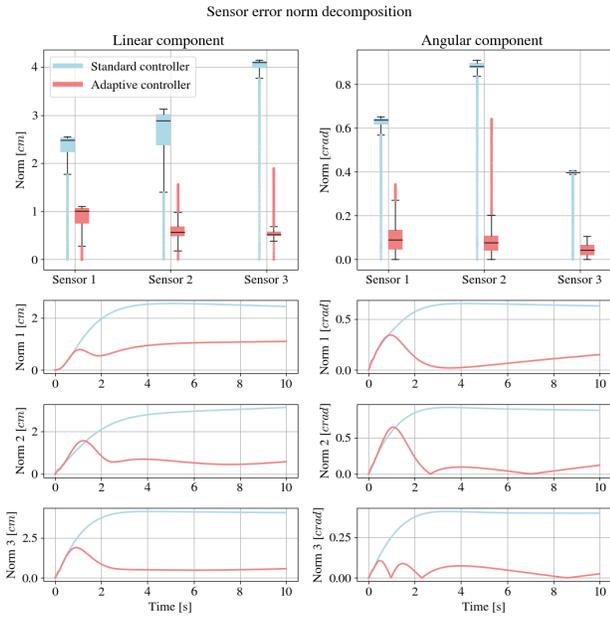


Fig. 16: The sensor error norms returned by the decoupled state feedback algorithm, decomposed into angular and linear components under both control strategies considering a payload of $m = 0.50 \text{ kg}$.

accounts for both orientation and position of the sensors.

In this scenario, the adaptive controller proves particularly effective (see fig. 16): Reducing the positional component of the sensor error between 2.5 and 4.5 cm , an improvement of 62 to 86% depending on the sensor. Similarly, the angular component is reduced around 0.35 to 0.85 crad , achieving an improvement between 84 to 90%.

To put this into perspective, the end-effector position of the inferred *virtual model* using the standard controller is more than 4 cm removed from sensor measurement recorded at the end-effector of the corresponding *true system*; while in the adaptive case the offset is less than 0.5 cm (see fig. 14).

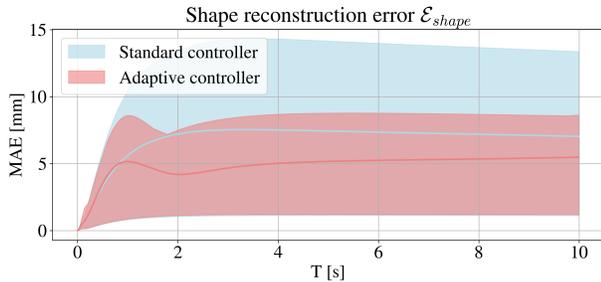


Fig. 17: The shape reconstruction error \mathcal{E}_{shape} aggregated over all simulation runs.

These findings remain consistent when we aggregate the performance of all simulation runs. As shown in fig. 17, \mathcal{E}_{shape} is lowered 28.6% on average (from 6.82 mm to 4.87 mm)

using the adaptive controller. Even though there is a slight performance decrease on the lower-bound of 6.32% (from 1.03 mm to 1.10 mm), the overall performance degradation for the adaptive controller is significantly better as the upper bound is 36.7% lower, coming down from 12.9 mm to 8.17 mm . Moreover, fig. 18 shows that the adaptive controller lowers the average cost on the secondary objective by 51.9% with a tighter overall spread as the upper- and lower-bound are reduced by 38.1% and 92.1% respectively.

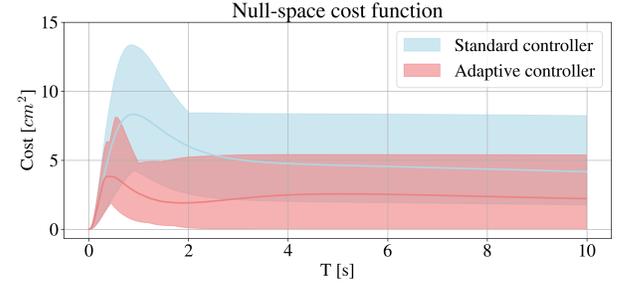


Fig. 18: A comparison of the null-space cost function between the adaptive- and standard controller averaged across all runs.

To briefly conclude, for higher resolution systems, the decoupled state feedback algorithm does not fully converge as the deformations are too complex for lower resolution models. However, the adaptive framework provides significantly more accurate inferred model states as the decoupled state algorithm converges closer to the measured values (fig. 19).

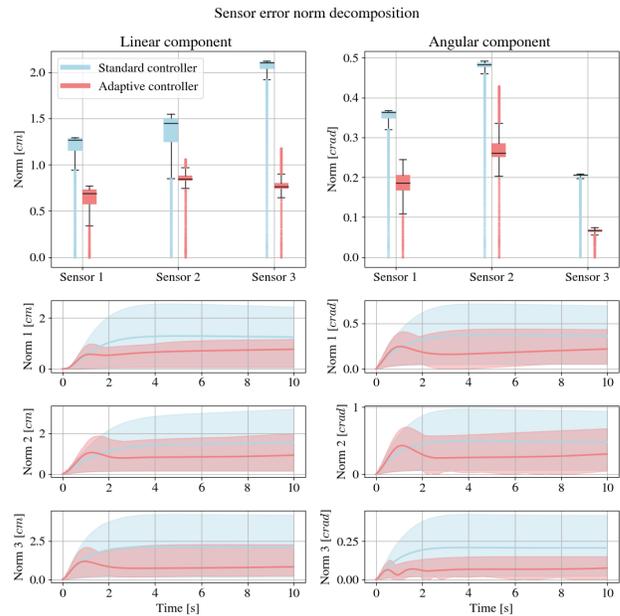


Fig. 19: The sensor error norms returned by the decoupled state feedback algorithm, decomposed into angular and linear components under both control strategies averaged across all runs.

VI. EXPERIMENTAL VALIDATION

The platform is a tendon-driven ‘trunk-scale’ soft robotic arm composed of five trimmed-helicoid modules, 3D printed with 72D soft TPU [24]. The tendons are controlled using a Raspberry Pi 5 and nine Dynamixel XM430-W210-R motors, acting on tendon differentials with respect to the calibrated zero-state. In addition, a dedicated motor is used to control the custom gripper. An overview of the tendon routing is provided in figure 20. Note that the platform utilizes the *independent manipulator* setup.

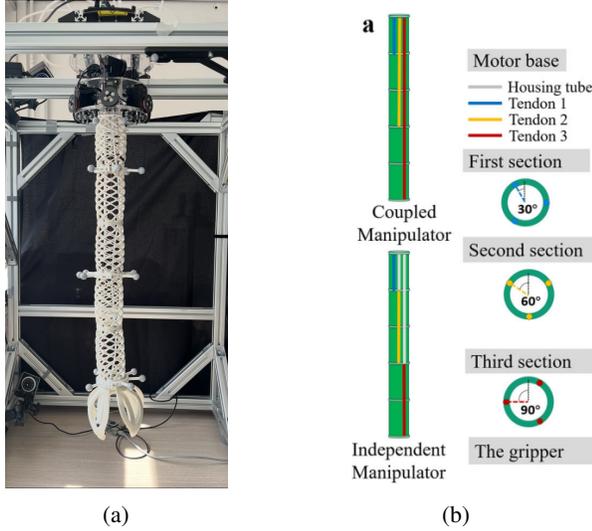


Fig. 20: (a) The experimental platform (b) A schematic overview of the tendon routing, configured as the independent manipulator setup as opposed to the coupled manipulator layout used in simulation. Image taken from [24]

To run the (decoupled) state feedback algorithm, eight Prime W13 Optitrack cameras are used, as shown in figure 21.

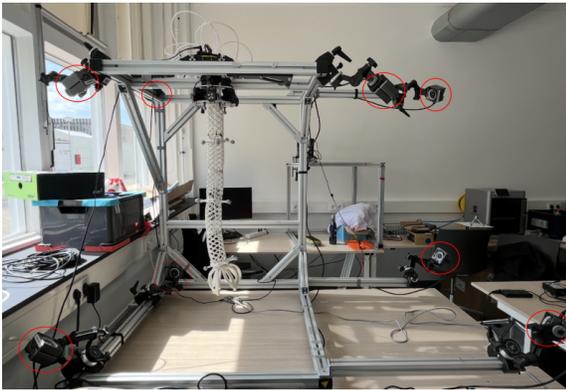


Fig. 21: The experimental setup: a tendon-driven five-module TH structure mounted on an aluminum extrusion profile frame. The red circles indicate the positions of the eight Optitrack cameras.

To deal with the deformable structure of the manipulator, which violates the ‘Rigid body’ constraints in Motive (the Op-

titrack software), custom 3D printed detachable PLA marker rings were manufactured. In addition, as the operational space of the manipulator was rather constrained, markers were developed with an extruded base to increase the detectability of the markers in close proximity to the cameras.

Each ring exhibits unique geometrical properties to remove body-wise ambiguity within the software; designed to maintain the same geometrical center, independent of the marker distribution, which is essential for accurate tracking of the structural backbone. As shown in fig. 20a, the distal ends of the first, third, and fifth TH module were used to attach the markers, aligning with the tendon mounting points.

Within Motive each set of markers is assigned a ‘Rigid Body’ ID whose coordinates are transformed into the assigned base-frame: four equally spaced markers around the perimeter of the manipulator base, with a translated pivot point of 20.5 mm in the positive z -direction; aligned with the backbone of the manipulator at the base (see fig. 22).

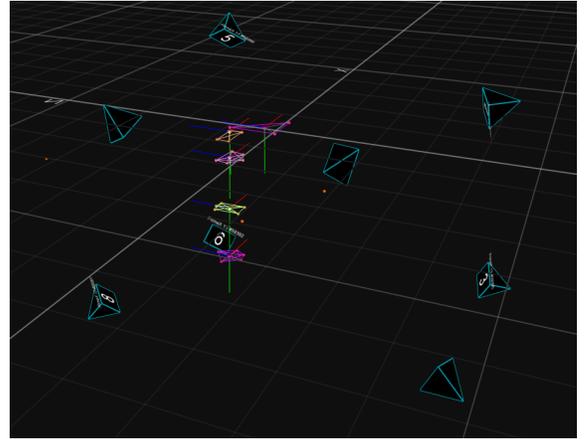


Fig. 22: The calibrated zero-state of the manipulator within motive: the defined rigid-bodies represent the ground plane, baseframe, and attached sensors from top to bottom, respectively. Note, the tracked rigid bodies are shown as right-handed Y -down coordinate frames, but streamed as Z -up.

A. Experiment I - Evaluating the decoupled state feedback algorithm

To evaluate the state feedback algorithm, in conjunction with the used sensing modality, we uniformly sample tendon differentials over the course of eight unique runs. Providing arbitrary shapes and deformations within a safe operating range for the platform:

$$\mathcal{X}_i = \begin{cases} \mathcal{U}(-0.035, 0) & \forall i \in \{1, 2, 3\} \\ \mathcal{U}(-0.075, 0) & \forall i \in \{4, 5, 6, 7, 8, 9\}, \end{cases}$$

with i the tendon index. In addition, random estimates of the kinematic parameters $\hat{\pi}$ are sampled according to:

$$\hat{\pi} = \pi + \epsilon, \text{ with } \epsilon \sim \mathcal{U}(-0.3, 0.3)$$

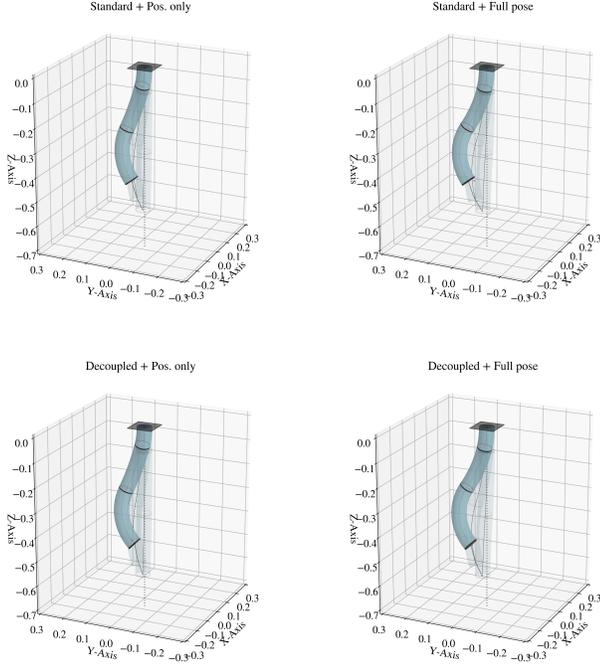


Fig. 23: A side-by-side, four-panel stroboscopic plot to compare the inferred virtual model states under decoupled and non-decoupled conditions for sample run 4.

In doing so, we can evaluate the algorithm’s performance under both decoupled and non-decoupled conditions, the sampling results are included in table I. Each run is initiated by acquiring both the curvilinear material set \mathcal{S}_s (22) and the kinematic parameters π for the non-decoupled case, defined in the initial calibrated zero-state using the Optitrack setup. Note, indeed, that π strictly corresponds to the non-decoupled case; as $\hat{\pi}$ is randomly sampled.

To alleviate jerky motion, the sampled tendon differentials are linearly interpolated from the calibrated zero-state to acquire n -unique control inputs for each run; published to the motor controllers, while the (non)-decoupled state feedback algorithm infers the state of the virtual model through real-time sensor measurements, considering both the *position* and *pose* setting simultaneously.

In figure 23, a four-way panel is provided to visually assess the performance in more detail; corresponding to run 4 shown in boldface in table I (see fig. 25a for the true system). It becomes apparent that the algorithm works as expected in both the non-decoupled (top-row) and decoupled (bottom-row) cases, considering either *position* or *pose* state feedback. The algorithm provides a smooth evolution of the inferred model state during task execution using real-time Cartesian sensor measurements.

While the discussed orientational ambiguity is already slightly visible in fig. 23 if the *position* measurements are used,

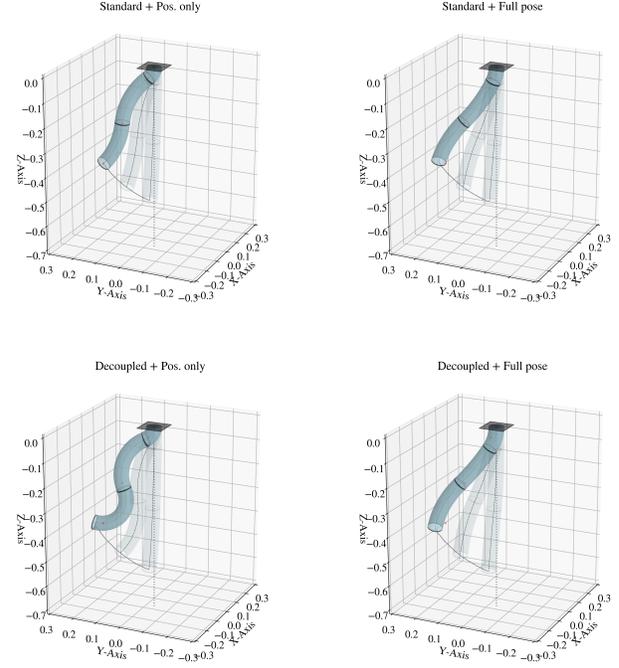


Fig. 24: A side-by-side, four-panel stroboscopic plot to compare the inferred virtual model states under decoupled and non-decoupled conditions for sample run 8.

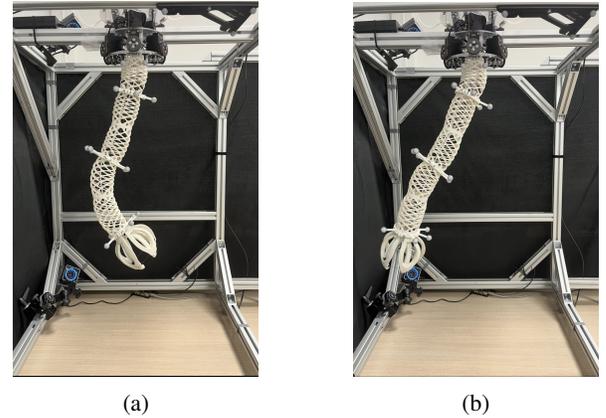


Fig. 25: A side-by-side comparison of the final manipulator state for sample runs four and eight, respectively.

different runs show a more pronounced implication. Hence, confirming earlier results: relieving orientational ambiguity introduces uncertainty in the inferred model state as the deformations become too complex to be fully resolved with a three-body system. Upon closer inspection of run 8, for example (see table I and fig. 25b), one can conclude that the inferred model state (see fig. 24) is markedly different from the true state of the system even though the algorithm converges if *position* measurements are used.

TABLE I: Overview of experimental parameters

Overview of experimental parameters															
Run	Sampled tendon differentials (\mathcal{X})									π			$\hat{\pi}$		
1	0.012	0.006	0.006	0.040	0.030	0.024	0.070	0.070	0.050	0.114	0.246	0.245	0.127	0.245	0.252
2	0.004	0.027	0.014	0.034	0.009	0.068	0.004	0.004	0.049	0.114	0.246	0.245	0.094	0.253	0.226
3	0.029	0.018	0.013	0.066	0.046	0.052	0.005	0.004	0.070	0.105	0.245	0.245	0.102	0.255	0.259
4	0.026	0.013	0.002	0.047	0.046	0.008	0.028	0.023	0.074	0.106	0.244	0.245	0.100	0.218	0.231
5	0.027	0.018	0.022	0.030	0.035	0.025	0.032	0.052	0.043	0.105	0.245	0.245	0.103	0.271	0.259
6	0.003	0.023	0.013	0.011	0.001	0.074	0.012	0.003	0.017	0.105	0.245	0.245	0.125	0.255	0.267
7	0.021	0.021	0.011	0.015	0.015	0.028	0.007	0.037	0.014	0.065	0.244	0.244	0.079	0.257	0.270
8	0.031	0.003	0.012	0.048	0.031	0.024	0.037	0.055	0.060	0.066	0.245	0.245	0.084	0.254	0.260

In this case, introducing the orientational term in the state feedback algorithm might not yield full convergence, but it provides a virtual model state that more accurately represents the true state. Interestingly enough, under these specific decoupled conditions, the effect is even more pronounced, highlighting the underlying trade-off.

To quantify the performance of the decoupled state feedback algorithm across the sampled workspace, figure 26 provides the average sensor error norms aggregated across all runs for the *pose* configuration.

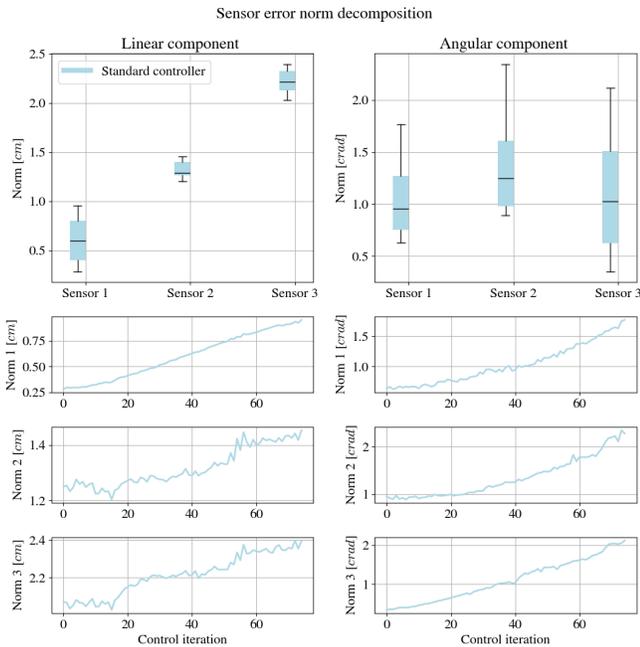


Fig. 26: Error norm decomposition for the decoupled state feedback algorithm averaged over all sampled runs for the standard controller

It is worth emphasizing that the presented data is *not* an evaluation of the adaptive controller as the kinematic parameters are randomly sampled, and not driven by the adaptive law. Rather, it provides an evaluation of the algorithm when dealing with decoupled states. Thereto, these measurements have not been included in fig. 26 as it would only skew the performance with stochasticity rather than objectivity.

B. Experiment II

The second experiment aligns with the dynamic simulations conducted in subsection V-B; i.e., benchmarking the adaptive framework (A-PCC) with the fixed-parameter PCC model via setpoint reaching tasks with varying end-effector payloads.

The sensors and actuators remain mounted at the distal ends of the first, third, and fifth TH-Module. Hence, we initialize the curvilinear material sets $\mathcal{S}_{s/a}$ (22) and the (estimated) kinematic parameters $\pi = \hat{\pi}$ in the calibrated zero-state via the motion capture, which runs at $50Hz$.

As earlier introduced, the model of actuation presented in sec. V-B and the tendon routing of the experimental platform do not correspond (recall the *coupled* and *independent* setup presented in fig. 20). Consequently, the reachable workspace is particularly constrained: the desired configuration q_d , provided by the standard and/or adaptive controller, is mapped to tendon differentials that do not accurately reflect the required values to reach q_d from a kinematic perspective.

Hence, to quantify the severity of this issue, we introduce an additional metric:

$$\mathcal{E}_{mapping} = \|x_d(q) - x_m\|, \quad x_d = h(q_d), \quad (53)$$

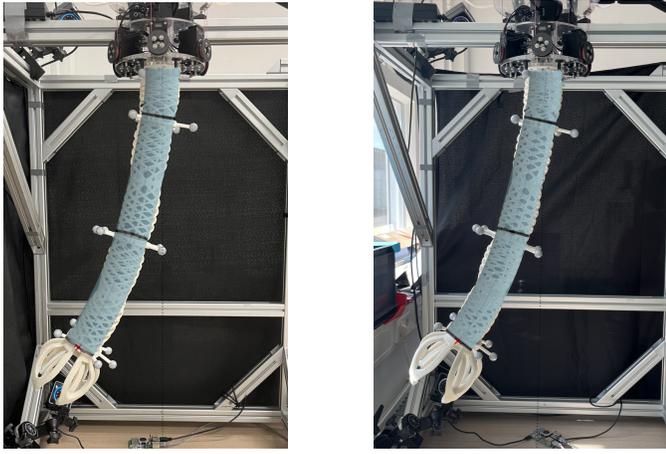
where x_d is the 'expected' end-effector position acquired via the forward kinematic map $h(\cdot)$ and x_m being the measured end-effector position after providing the control input $\hat{y} = \hat{B}(\hat{\pi})^T q_d$ (or $y = B^T q_d$ in the non-adaptive case) as defined in (49).

Run I - Gripper + Position feedback only: The first experimental run considers the desired setpoint

$$x_d = \begin{bmatrix} 0, 0.15, -0.5 \end{bmatrix}, \quad (54)$$

with the (decoupled) state feedback algorithm using *positional* sensor measurements. Besides the custom gripper, no additional end-effector payload is attached. The controller gain is set to $K = 1.5I_3$ for both the standard and adaptive controller, with the latter using $\Gamma = 40I_3$ for the adaptive law defined in (14). The control loop is terminated once the end-effector position converges within $1mm$ of the desired setpoint.

The absence of a direct measurement on the backbone of the robot manipulator precludes the use of \mathcal{E}_{shape} as defined in (52), complicating a quantitative performance comparison in that regard.



(a) Standard controller (b) Adaptive controller

Fig. 27: A dual-panel visual overlay of the *true system* and associated *virtual model(s)* in the YZ -plane. The visualized state corresponds to the moment of task completion, considering *position* measurements only.

To address this, a visual overlay of the *true system* and associated *virtual model* is provided in the YZ -plane (see fig. 27). While the visual differences, in this case, might be difficult to observe, and task execution times remain approximately the same for both control strategies (see fig. 28), the way they execute the task is inherently different.

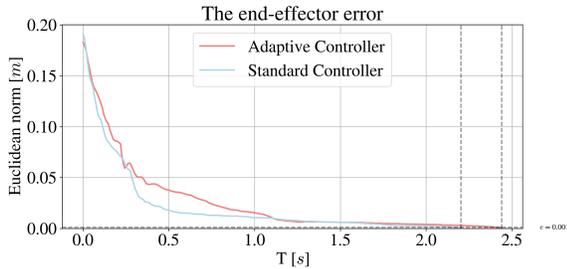


Fig. 28: The measured end-effector error norm with no external payload running on *position* feedback only; control-loop termination is indicated with the vertical dashed line.

As shown in figure 29, the adaptive controller effectively reduces the secondary objective cost-function by an average of 69.4% (from 2.02 cm^2 to 0.62 cm^2). More importantly, however, the control precision is adversely affected as introduced in (53) - even for setpoints defined close to the initial state. Yet, as illustrated in fig. 30, the adaptive controller mitigates this issue as it reduces the average mismatch by 58.6% (from 6.15 cm to 2.55 cm).

A feat that could also be physically observed during the experiments due to reduced tendon slack near the motor housing and a less frequent need to recalibrate the system.

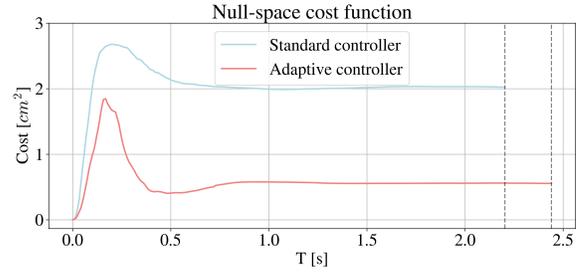


Fig. 29: The secondary cost function objective with no external payload running on *position* feedback only; control-loop termination is indicated with the vertical dashed line.

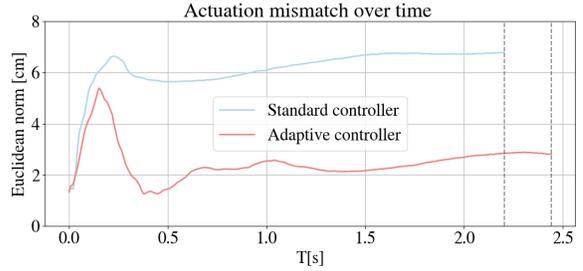


Fig. 30: The actuation mismatch with no external payload running on *position* feedback only; control-loop termination is indicated with the vertical dashed line.

Run II - Gripper + Full pose feedback: The second run considers identical conditions to run I, but in this case utilizes the *pose* measurements. The controller gains remain unchanged with $K = 1.5I_3$ for both the standard- and adaptive controller, with the latter using and $\Gamma = 40I_3$ for the adaptive law defined in (14).

Again, a visual overlay of the *true system* and associated *virtual model(s)* is provided in the YZ -plane(see fig. 31). In this case, however, the visual discrepancies are more pronounced. As shown in fig. 32, the adaptive controller also exhibits significantly faster convergence to the desired setpoint: converging in 2.76 s compared to 6.01 s for the standard controller.

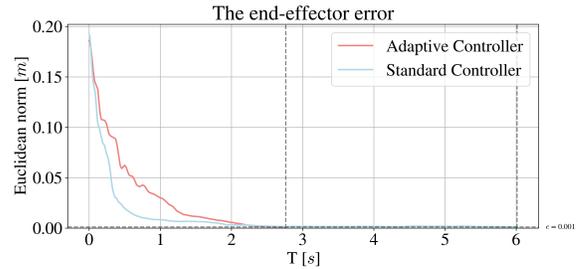


Fig. 32: The measured end-effector error norm with no external payload running on *pose* feedback; control-loop termination is indicated with the vertical dashed line.

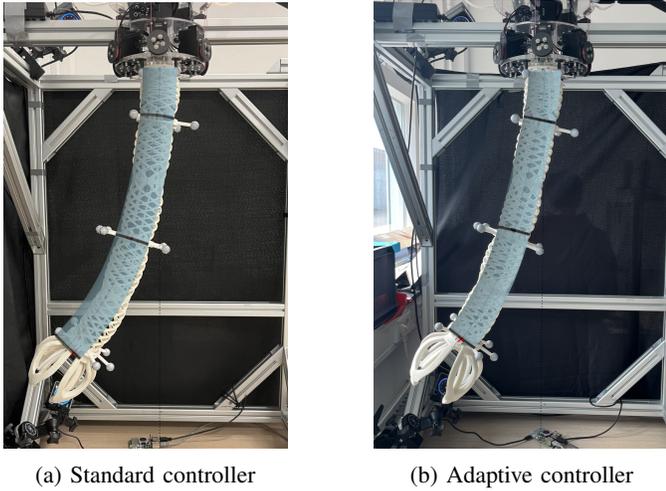


Fig. 31: A dual-panel visual overlay of the *true system* and associated *virtual model(s)* in the YZ -plane. The visualized state corresponds to the moment of task completion, considering *pose* measurements.

While stochastic variations may partially influence this improvement due to the actuation mapping issue, particularly for subtle movements close to the setpoint, the additional performance metrics provide further insight. As illustrated in figure 33, the adaptive controller reduces the secondary objective cost function by an average of 66.4%, decreasing from 1.97 cm^2 under the standard control strategy to 0.66 cm^2 .

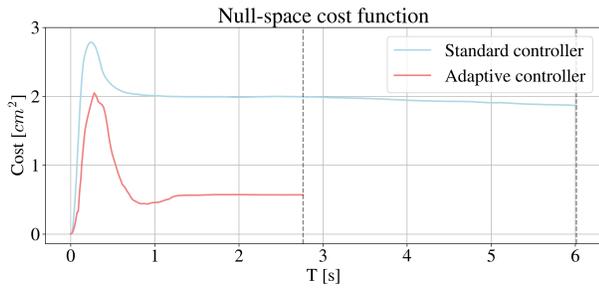


Fig. 33: The secondary cost function objective with no external payload running on *pose* feedback; control-loop termination is indicated with the vertical dashed line.

But more than that, the actuation mismatch in fig. 34 reveals that at the time of control-loop termination, the adaptive controller stabilizes at 3.2 cm , whereas the standard controller continues to grow to 8.5 cm - providing an overall improvement of 57.42%.

A result that ties directly into the fact that the inferred model state thus becomes more accurate when adaptivity is introduced: reducing the positional norm by an average of 0.25 cm to 0.5 cm while maintaining comparable angular norms on average (see fig. 35).

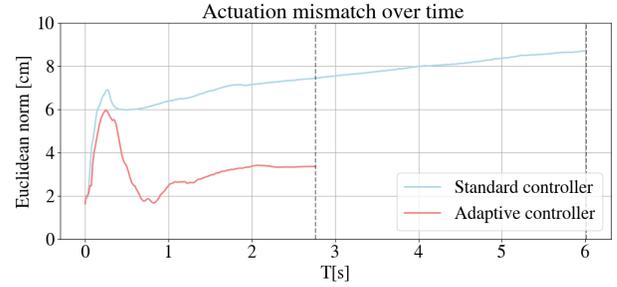


Fig. 34: The actuation mismatch with no external payload running on *pose* feedback; control-loop termination is indicated with the vertical dashed line.

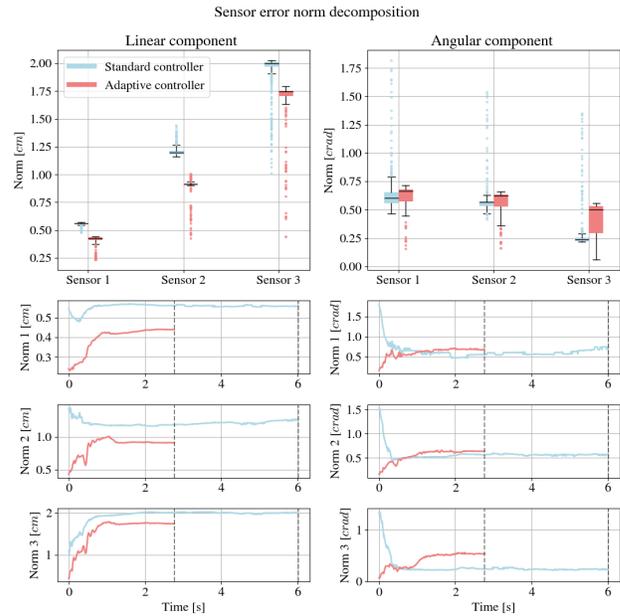


Fig. 35: The decoupled state feedback performance with no external payload running on *pose* feedback; control-loop termination is indicated with the vertical dashed line.

Run III - Gripper with 0.210 kg + Full pose feedback: The third experimental run highlights the ability to utilize lower control gains by introducing adaptivity, even when carrying a payload. Hence, the robot manipulator carries an apple with a mass of $m = 0.21 \text{ kg}$, utilizing full *pose* sensor feedback but with distinct controller levels.

The standard controller gain is set to $K_S = 2I_3$, whereas the adaptive controller operates with a reduced gain level of $K_A = 1.5I_3$. The latter uses $\Gamma = 30I_3$ for the adaptive law defined in (14).

As observed in figure 36, the adaptive controller exhibits a slightly slower convergence time to the desired setpoint; approximately 1s in comparison to the standard controller. This outcome does, however, align with the expectations of using a lower gain configuration. To illustrate, an evaluation of the standard controller operating at identical gain levels

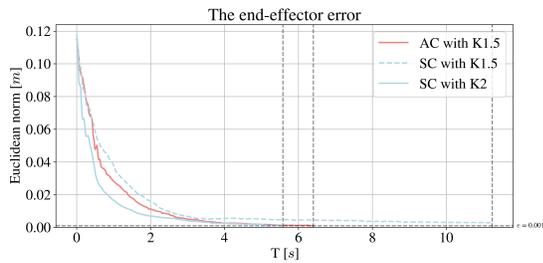


Fig. 36: A comparison of the measured end-effector error with a 0.21kg payload attached, considering *pose* measurements and three distinct controller gain levels.

($K_S = 1.5I_3$) is also presented, demonstrating that while the adaptive controller successfully reaches the setpoint under lower gain conditions, the standard controller does not fully converge and is manually terminated after $T = 11.24s$.

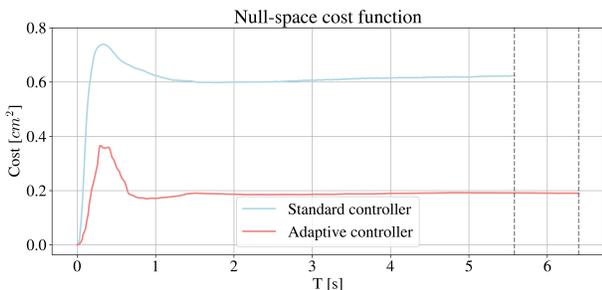


Fig. 37: A comparison of secondary cost function objective considering full pose measurements and 0.21kg payload

Interestingly enough, even when operating at the lower gain level, the adaptive controller yields further advantages in comparison to the higher gain standard controller. It reduces the secondary objective cost function by 68.3% (see fig. 37) and decreases the actuation mismatch (53) by 27.3% as illustrated in fig. 38.

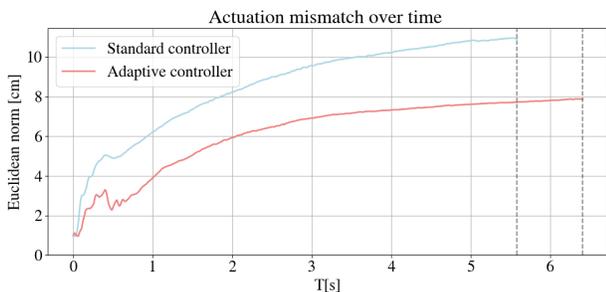


Fig. 38: A comparison of actuation mismatch considering full pose measurements and 0.21kg payload

Moreover, figure 39 demonstrates that the inferred model state improves: the angular norms are reduced by 0.5 to 1 *crad*, while the positional terms are between 0.3 *cm* and 0.5 *cm*

lower on average for the first and second sensor respectively, while conceding 0.15 *cm* on the end-effector position.

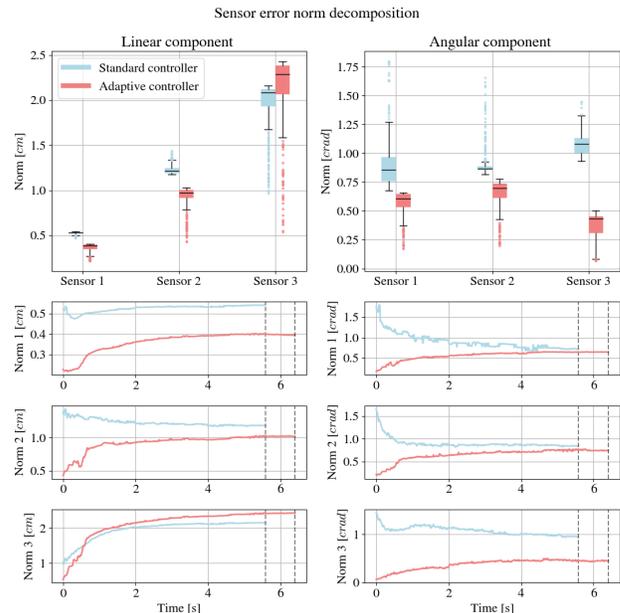


Fig. 39: A comparison of state feedback performance considering full pose measurements and 0.21kg payload.

VII. CONCLUSION & DISCUSSION

In this work, we proposed an adaptive kinematic modeling framework to address the challenge of underactuation from a different perspective - not by increasing model resolution or complexity - but by making the model itself adaptive.

To introduce adaptivity into the framework, we made the kinematic model parametric with respect to the parameters L_{0_i} and dynamically adapted the set using a novel inverse kinematic adaptive controller. The controller presents a unique and desirable trait in that it inherently enhances the shape representation as a by-product of the primary control objective. In addition, we also provided a proof that the control and parameter update law result in global asymptotic convergence of the position error.

The controller was presented in conjunction with a decoupling algorithm to separate both the sensing and actuation inputs on the real system from the virtual model, as part of a novel decoupled state feedback algorithm. This algorithm allowed us to infer the virtual model state from an arbitrary number of sensors and model sections, independent of the underlying discretization as estimated by the adaptive controller.

While the framework was developed alongside the widely used PCC-model, and introduces unique challenges on its own, in principle, it can be transferred to any higher-fidelity kinematic model that allows for a linearly parametrized structure to compensate for the underactuation.

The proposed solution was evaluated considering a kinematic controller on the fully actuated case. Here, it was shown that the estimated parameters can converge to the true values

during a kinematic trajectory tracking task under nominal conditions. Furthermore, dynamic simulations and experiments demonstrated increased performance in shape representation, model state inference, and the use of redundant degrees of freedom through a secondary task objective.

Therefore, we can conclude for the hypothesis of the thesis, the adaptive framework mitigates the issue of underactuation better - in comparison to fixed-parameter models presented in literature - while retaining simplicity by adapting to real-world deformation online.

Further work should be dedicated to a more extensive experimental validation, unconstrained by actuation mapping discrepancy and void of frequent recalibration. In addition, the behavior of the Savitzky-Golay type filter is yet to be evaluated, as it introduces an additional degree of freedom to the adaptive law. The configuration velocity \dot{q} is estimated on the N -previous inferred states q , meaning that lower values of N allow for more aggressive, noisy updates on the kinematic parameters, while higher values provide smoother but less responsive updates.

To achieve additional performance results, controller tuning can be optimized. The work considered a general performance across the workspace for various payloads, while more task-specific conditions allow for the adaptive controller to be tuned accordingly, potentially utilizing asymmetrical gain levels.

Moreover, as the validation considers setpoint reaching tasks, instead of trajectory-tracking, the controller is sensitive to the gains when optimizing the shape representation. This is intuitively understood by the fact that the adaptive law is driven by the end-effector error, and therefore no incentive exists to optimize the shape fidelity.

On a more general note, the (decoupled) state feedback algorithm also introduces an interesting aspect. To relieve the virtual model (either adaptive or not) of orientational ambiguity, one concedes on positional alignment of the model with the real world sensor measurements. Therefore, it would be interesting to explore a hierarchy-based numerical inverse kinematics scheme. Providing control over the degree of alignment from an orientational point of view, while maintaining full positional priority.

As a closing remark, it would be interesting to transfer the framework to a more complex model to evaluate the potential or even a composite adaptive controller, utilizing the adaptive model while addressing uncertainty in the dynamics simultaneously.

ACKNOWLEDGMENT

This letter marks the conclusion of my MSc Thesis in the Cognitive Robotics department at Delft University of Technology. Starting as a novice within the domain of soft robotics, back when I started my literature review, I would have never imagined coming this far. The last year has been challenging and rewarding on many different levels, and the experience, knowledge, and confidence I have gathered along the way are invaluable, both on a personal and academic level.

This work would not have been possible without my supervisors, Dr. Cosimo Della Santina and Pietro Pustina. Thank you, Cosimo, for providing a place within your research group and, the continued guidance and support over the course of this work. Thank you, Pietro, for all the time and energy you invested; you have been a brilliant mentor whom I could always count on. Thank you both for reassuring me that I should be more confident in my work. It has been a privilege to work with both of you.

I would also like to thank Francesco Stella and Seb Tiburzio for their assistance in setting up and familiarizing me with the experimental platform. This gratitude extends to the other PhD candidates working in the CoR lab who have assisted me.

Last but not least, I would like to thank my family and girlfriend, both of whom have tirelessly supported me over the last couple of years to achieve this milestone. Thank you for encouraging me to get the best out of myself.

REFERENCES

- [1] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.
- [2] C. Armanini, F. Boyer, A. Mathew, C. Duriez, and F. Renda, "Soft robots modeling: A structured overview," *IEEE Transactions on Robotics*, vol. PP, pp. 1–21, 06 2023.
- [3] C. Della Santina, C. Duriez, and D. Rus, "Model-based control of soft robots: A survey of the state of the art and open challenges," *IEEE Control Systems Magazine*, vol. 43, no. 3, pp. 30–65, 2023.
- [4] H. Abidi, G. Gerboni, M. Brancadoro, J. Fras, A. Diodato, M. Cianchetti, H. Wurdemann, K. Althoefer, and A. Menciassi, "Highly dexterous 2-module soft robot for intra-organ navigation in minimally invasive surgery," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 14, no. 1, p. e1875, 2018.
- [5] X. Wang, H. Kang, H. Zhou, W. Au, M. Y. Wang, and C. Chen, "Development and evaluation of a robust soft robotic gripper for apple harvesting," *Computers and Electronics in Agriculture*, vol. 204, p. 107552, 2023.
- [6] P. A. der Maur, B. Djambazi, Y. Haberthür, P. Hörmann, A. Kübler, M. Lustenberger, S. Sigrist, O. Vigen, J. Förster, F. Achermann *et al.*, "Roboa: Construction and evaluation of a steerable vine robot for search and rescue applications," in *2021 IEEE 4th International Conference on Soft Robotics (RoboSoft)*. IEEE, 2021, pp. 15–20.
- [7] D. Trivedi, C. D. Rahn, W. M. Kier, and I. D. Walker, "Soft robotics: Biological inspiration, state of the art, and future research," *Applied Bionics and Biomechanics*, vol. 5, pp. 99–117, 2008.
- [8] C. Laschi, T. G. Thuruthel, F. Lida, R. Merzouki, and E. Falotico, "Learning-based control strategies for soft robots: Theory, achievements, and future challenges," *IEEE Control Systems Magazine*, vol. 43, no. 3, pp. 100–113, 2023.
- [9] D. B. Camarillo, C. R. Carlson, and J. K. Salisbury, "Configuration tracking for continuum manipulators with coupled tendon drive," *IEEE transactions on robotics*, vol. 25, no. 4, pp. 798–808, 2009.
- [10] F. Renda, F. Boyer, J. Dias, and L. Seneviratne, "Discrete cosserrat approach for multisection soft manipulator dynamics," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1518–1533, 2018.
- [11] F. Renda, C. Armanini, A. Mathew, and F. Boyer, "Geometrically-exact inverse kinematic control of soft manipulators with general threadlike actuators' routing," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7311–7318, 2022.
- [12] C. Duriez, "Control of elastic soft robots based on real-time finite element method," pp. 3982–3987, 2013.
- [13] O. Gouy and C. Duriez, "Fast, generic, and reliable control and simulation of soft robots using model order reduction," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1565–1576, 2018.
- [14] C. Della Santina, M. Bianchi, G. Grioli, F. Angelini, M. Catalano, M. Garabini, and A. Bicchi, "Controlling soft robots: balancing feedback and feedforward elements," *IEEE Robotics & Automation Magazine*, vol. 24, no. 3, pp. 75–83, 2017.

- [15] M. Trumić, C. Della Santina, K. Jovanović, and A. Fagiolini, "Adaptive control of soft robots based on an enhanced 3d augmented rigid robot matching," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 4991–4996.
- [16] A. Kazemipour, O. Fischer, Y. Toshimitsu, K. W. Wong, and R. K. Katzschmann, "Adaptive dynamic sliding mode control of soft continuum manipulators," pp. 3259–3265, 2022.
- [17] M. Azizkhani, A. L. Gunderman, I. S. Godage, and Y. Chen, "Dynamic control of soft robotic arm: An experimental study," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 1897–1904, 2023.
- [18] T. George Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, "Control strategies for soft robotic manipulators: A survey," *Soft robotics*, vol. 5, no. 2, pp. 149–163, 2018.
- [19] M. Grube, J. C. Wieck, and R. Seifried, "Comparison of modern control methods for soft robots," *Sensors*, vol. 22, no. 23, p. 9464, 2022.
- [20] C. Della Santina, A. Bicchi, and D. 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*, vol. 5, no. 2, pp. 1001–1008, 2020.
- [21] T. Marcucci, C. Della Santina, M. Gabiccini, and A. Bicchi, "Towards minimum-information adaptive controllers for robot manipulators," in *2017 American Control Conference (ACC)*, 2017, pp. 4209–4214.
- [22] C.-C. Cheah, C. Liu, and J.-J. Slotine, "Adaptive tracking control for robots with unknown kinematic and dynamic properties," *I. J. Robotic Res.*, vol. 25, pp. 283–296, 01 2006.
- [23] K. Lynch and F. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017. [Online]. Available: <https://books.google.nl/books?id=86G0nQAACAAJ>
- [24] Q. Guan, F. Stella, C. Della Santina, J. Leng, and J. Hughes, "Trimmed helicoids: an architected soft structure yielding soft robots with high precision, large workspace, and compliant interactions," *npj Robotics*, vol. 1, no. 1, p. 4, 2023.

APPENDIX A - ADDITIONAL SIMULATION RESULTS

The aim of this appendix is to highlight the generality and robustness of the proposed solution on a secondary continuum manipulator. As opposed to the non-uniformly discretized *true system* in section V-B, this simulation run considers a uniformly discretized system with a shorter overall length of 0.45m. In coherence, the *true system* is initialized with the kinematic parameters:

$$\pi_t = \begin{bmatrix} 0.075 & 0.075 & 0.075 & 0.075 & 0.075 & 0.075 \end{bmatrix},$$

where both the sensors and actuators remain 'attached' at the distal ends of the second, fourth and sixth section. Hence, defining the new sets of curvilinear material abscissae as $\mathcal{S}_s = \mathcal{S}_a = \{0.15, 0.30, 0.45\}$ in the strain-free state.

The *virtual model* in both the adaptive- and non-adaptive cases is initialized as:

$$\pi_v = \hat{\pi}_v = \begin{bmatrix} 0.15 & 0.15 & 0.15 \end{bmatrix},$$

i.e. the 'attached' sensors in the *true system* correspond to the section tips of the virtual model and are thus presumed to be reasonably measurable. Note, indeed, that in the adaptive case these parameters are continuously updated thus this is only valid in the initial state; as t varies the model will decouple. Furthermore, the structure maintains the same model of actuation as shown in section V-B.

In similar fashion, we consider ten different setpoints dispersed across the workspace, with each setpoint evaluated for three different end-effector payloads: $m = 0kg$, $m = 0.25kg$, and $m = 0.50kg$, considering a simulation runtime of $T = 10s$. In addition, the performance is assessed using

only the *pose-feedback* setup for the decoupled state feedback algorithm. The PID-controller gains remain unchanged and are thus set to $K_P = 750I_9$, $K_I = I_9$ and $K_D = 5I_9$. The inverse kinematic controller gains are set to $K = 1.5I_3$ for both the standard and adaptive controller, with the latter using $\Gamma = 18.5I_3$ for the adaptive law defined in (14).

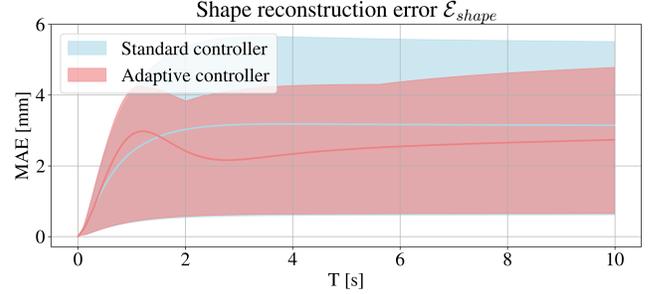


Fig. 40: A comparison of the shape reconstruction error \mathcal{E}_{shape} between the adaptive- and standard controller

Figure 41 provides the familiar stroboscopic visual comparison of a selected run from the subset. As shown in fig. 40, the shape fidelity is improved as \mathcal{E}_{shape} is lowered on average 28.6% (from 6.82mm to 4.87mm) during task execution using the adaptive controller, accompanied by a lower-bound concession of 5.82% (from 0.55mm to 0.58mm) in favor of the standard controller but with significantly less performance degradation for the adaptive controller - the upper bound is reduced by 19.4% from 5.16mm to 4.15mm.

Moreover, fig. 42 shows that the adaptive controller lowers the average cost on the secondary objective by 25.8%, while the upper- and lower-bound are reduced by 17.8% and 36.1% respectively.

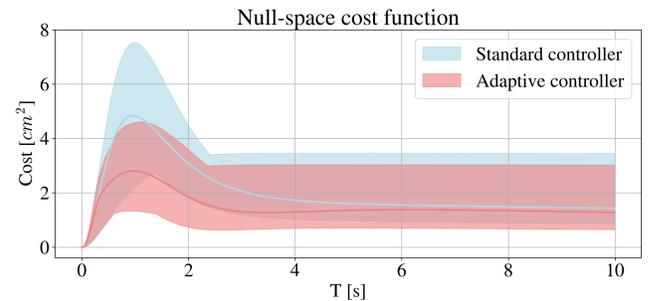


Fig. 42: A comparison of the null-space cost function between the adaptive- and standard controller averaged across all runs.

Finally, the adaptive controller provides significantly more accurate inferred model states as the decoupled state algorithm converges closer to the measured values. This result is substantiated in fig. 43

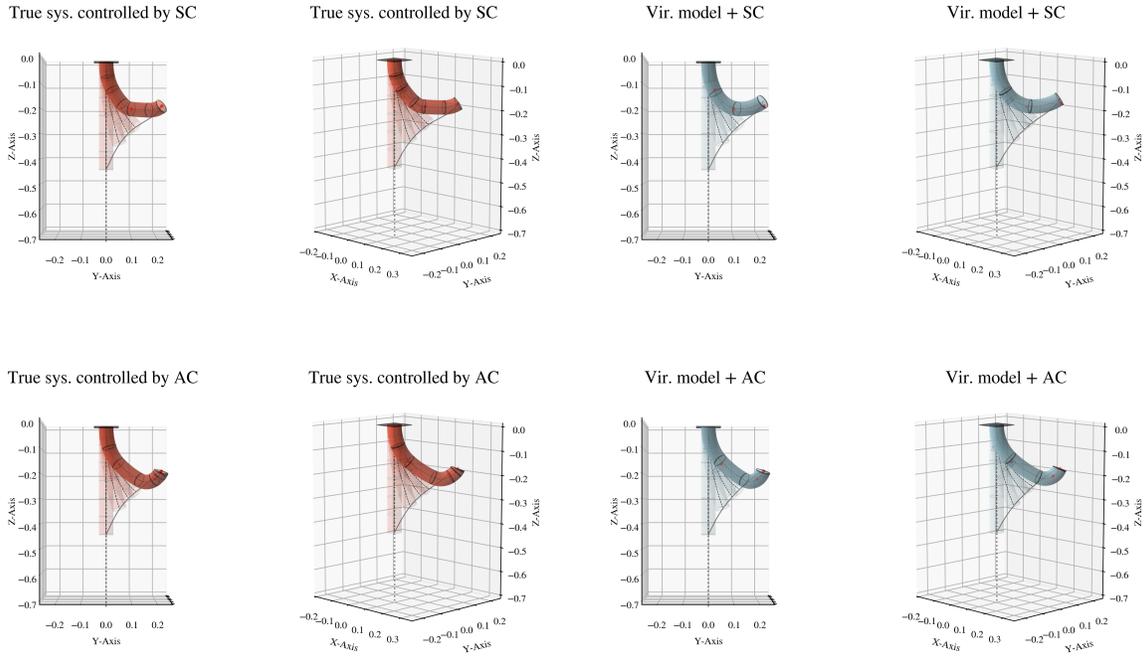


Fig. 41: A side-by-side, dual angle, stroboscopic plot as a visual comparison on the evolution of the *True system* carrying a 0.50kg payload to the setpoint $x_d = [0.15, 0.225, -0.1875]$; controlled with the standard and adaptive controller on the first and second row respectively. The red markers indicate the sensor- and actuator locations on the true system and the (decoupled) location in the virtual model space.

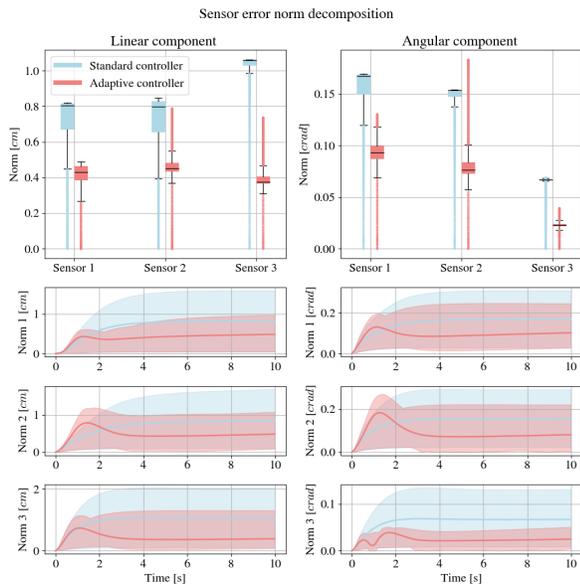


Fig. 43: The decomposed sensor error norms of the decoupled state feedback algorithm