Low-dimensional Kinematic and Dynamic Model Identification for Planar Continuum Soft Robots from Pixels

by

Ricardo Valadas
(5386632)

18th October, 2024

Supervisor: Dr. C. Della Santina
Daily Supervisors: Maximilian Stölzle and Jingyue Liu
Course: RO57035 MSc Thesis
Faculty: Faculty of Mechanical Engineering (ME), Delft

TU Delft

Low-Dimensional Kinematic and Dynamic Model Identification for Planar Continuum Soft Robots from Image Pixels

Ricardo Valadas, 5386632
MSc Robotics, Delft University of Technology

***Abstract*—In recent years, soft robots have become a focal point of research due to their ability to mimic natural movements and adapt to unstructured settings. However, their inherent flexibility poses significant challenges, particularly in the areas of modelling and control. While data-driven methods can model soft robot behavior without explicit physical models, they require extensive data and lack interpretability. On the other hand, physics-based low-dimensional models have relied heavily on expert knowledge and intuition, sometimes leading to models that are either too simple and inaccurate, or excessively high-dimensional.**

This thesis introduces an end-to-end methodology for automatically identifying low-dimensional kinematic and dynamic models of planar continuum soft robots using image data. Based on the Piecewise Constant Strain (PCS) parametrization, the proposed approach determines an efficient segmentation for the soft robot to approximate its configuration. Afterward, a model identification strategy is employed to obtain a dynamic model that contains only the most essential strains. This model is formulated in the standard Euler-Lagrange framework, facilitating the integration with conventional model-based control schemes. The methodology is validated through simulations involving various planar soft manipulators and in the presence of noise, demonstrating its capability to generate accurate and computationally efficient models. This work provides a fast and practical tool to help the modelling and control of continuum soft robots, highlighting the potential for future applications in more complex actuation systems and real-world soft robots.

I. Introduction

The way humans conceptualize the role of robots has been changing in recent years. Traditional robots were built to execute simple and repetitive heavy-duty tasks with high precision. Lately, this idea has evolved to a more human- and service-centered approach, where robots can collaborate and perform activities that were thought to be exclusive to biological systems [1]. Handling fragile objects, interacting safely with humans, or adapting their shape to fit in confined environments are all features that could be achieved with the emergence of soft robots [2]. Among soft robots, continuum robots, which do not contain discrete joints, are particularly appealing due to their continuous and smooth deformation. The inherent flexibility presents, nonetheless, numerous challenges, particularly in modelling and control.

Accurate modelling is essential for optimizing both the design and control of soft robots, but the continuum nature introduces complex, nonlinear dynamics that are difficult to describe with traditional rigid-robot approaches [3]. Recently, data-driven methods have become popular as a way to overcome this. By learning directly from data, neural network architectures [4] [5] [6] can model the deformations of soft robots without requiring detailed physical derivations. This allows them to capture behaviours that are hard to express analytically [7] [8]. However, these methods have their limitations. Firstly, they often require large amounts of high-quality training data [9]. Secondly, these black-box techniques ignore any system's physical properties, which not only hinders interpretability but also does not guarantee good extrapolation performance [10].

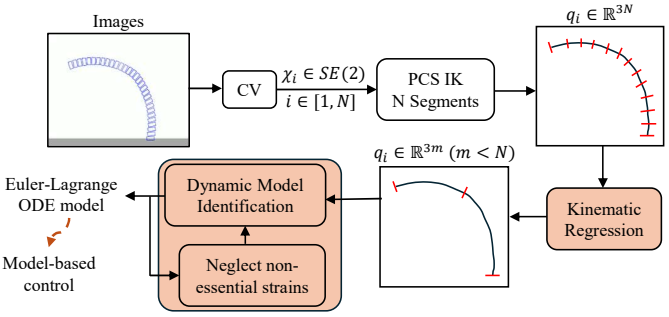


Fig. 1: Diagram overview of the end-to-end pipeline proposed. The main contributions of the work are highlighted in orange.

High-dimensional analytical models have been developed based on simulating the continuum mechanics of soft robots. They are solved using finite-element methods, both in 3D [11] [12] [13] and 1D formulations (i.e., discrete Cosserat rod) [14] [15] [16]. Although highly effective at simulating the nonlinear deformations, their high dimensionality and computational cost are often impractical for real-time control [17] [9].

Therefore, if the model is required for control applications, we need to find a good trade-off between computational efficiency and accuracy. Low-dimensional analytical models appear as a good solution to this. They rely on geometric simplifications that make them faster to compute and easily derivable for control law proofs [18]. A commonly used method is the Piecewise Constant Strain (PCS) [19], which divides the robot's structure into segments, each assumed to have constant deformation. Additional simplifications can be carried out if the soft robot's design allows for some of the strains to be neglected. For example, the popular Piecewise Constant Curvature (PCC) model [20] neglects all strains but bending. The constant strain assumption means the robot can be described through a finite set of configuration variables,

allowing for more efficient control. However, previous works have relied heavily on expert knowledge and designer intuition to derive these models, particularly in decisions involving the number of segments or which strains to consider [21]. This is not always an obvious choice and it could result in models that are higher-dimensional than necessary or that important strains are ignored based on a wrong intuition [22].

Instead, we propose in this work an end-to-end approach to automatically finding a PCS kinematic description and identifying the corresponding dynamic model, directly using as data source captured images of the robot in motion. Firstly, a kinematic regression procedure tries to minimize the number of segments of the PCS kinematic model, while enabling the user to balance model complexity and shape accuracy. Secondly, based on the determined kinematic model, a dynamic model identification is employed to estimate the inertial and impedance parameters through closed-form linear least-squares, while simultaneously neglecting less significant strains. One key factor lies in how the dynamic model is extracted in standard Euler-Lagrange form, allowing a quick deployment in classical model-based controllers [17] [23] [24], contrasting with purely data-driven models [17]. Our approach, by merging data-driven insights with a physics-based structure, can capture the system dynamics while remaining computationally efficient.

The method is validated in several simulated scenarios, including distinct underlying kinematics (PCS and Piecewise Affine Curvature (PAC)), varying numbers of segments, and the presence of measurement noise.

II. Preliminaries

A. PCS Kinematics

The continuous nature of soft robots, characterized by their ability to deform over a continuous space, implies that their motion is governed by a set of nonlinear partial differential equations (PDEs). As a result, an infinite number of degrees of freedom (DOF) are required to accurately characterize their dynamics. To make the modelling task tractable, several assumptions are commonly used. The first one takes advantage of the typical shape of these robots, which tend to have one physical dimension longer than the other two. The analysis of slender, elongated structures can be approximated to its central axis (backbone) [25]. The second assumption takes care of the infinite-dimensional problem by approximating the continuous deformation through space discretization along the backbone. PCS models are built upon both of these. The backbone is discretized into a few segments, and the six elemental local strains (bending, shear, axial and torsion) are considered constant in space, but variable in time, in each of the segments. Local strains are associated with either a pure translation or rotation along one of the axis of the reference frame attached to the end of a segment. Therefore, an analogy can be drawn between joint states in rigid robots and local strains in continuum robots: just like the pose of a rigid link is dependent on the previous joint variables, the pose at a certain point of the soft robot is dependent on the local strains of all the previous segments.

For the planar case, shown in Fig. 2, only three strains are present: bending, shear and axial. The configuration of segment i, with length $L_{0,i}$, is defined by $\mathbf{q}_i = [\kappa_{\text{be},i}, \sigma_{\text{sh},i}, \sigma_{\text{ax},i}]^\top \in \mathbb{R}^3$, where

- $\kappa_{\text{be},i}$ is the segment's bending strain, which corresponds to the segment's curvature;
- $\sigma_{\text{sh},i}$ is the segment's shear strain, measuring the relative shear displacement δx_i, $\sigma_{\text{sh},i} = \delta x_i / L_{0,i}$;
- $\sigma_{\text{ax},i}$ is the segment's axial strain, which measures the relative change in length δL_i, $\sigma_{\text{ax},i} = \delta L_i / L_{0,i}$.

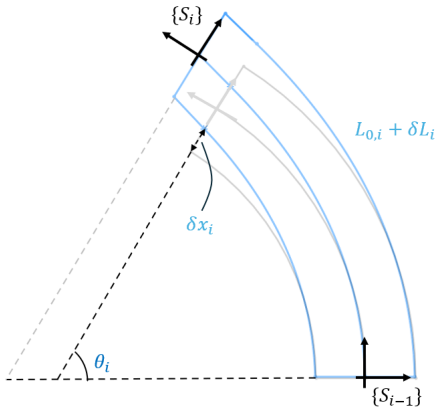


Fig. 2: Illustration of a planar PCS segment. In light gray the segment is represented only undergoing bending deformation, with amplitude θ_i. In blue, the same segment also exhibits shear and axial displacements. S_i and S_{i-1} denote the local frames of the current and preceding segments, respectively. δx_i and δL_i indicate the displacements caused by shear and axial strains.

This configuration vector fully specifies the shape of the robot. Given $s \in [0, L_{0,i}]$ the coordinate along the backbone, the orientation of the central axis along the segment i can be determined by integration of the curvature [20] via

$$\theta_i = \int_0^s \kappa_{\text{be},i}\, dv = \kappa_{\text{be},i} s\,. \tag{1}$$

The x- and $y-$ position of the backbone along the segment, $[p_{x,i}, p_{y,i}]^\top$, can be calculated by integrating the shear and axial strains (properly rotated)

$$\begin{bmatrix} p_{x,i} \\ p_{y,i} \end{bmatrix} = \int_0^s \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \begin{bmatrix} \sigma_{\text{sh},i} \\ 1+\sigma_{\text{ax},i} \end{bmatrix} dv. \tag{2}$$

Knowing the above relations, the forward and inverse kinematics relating the pose along the segment $\boldsymbol{\chi}_i = [p_{x,i}, p_{y,i}, \theta_i]^\top \in \mathbb{R}^3$ and the configuration $\mathbf{q}_i$ can be written in closed form [26] as

$$\boldsymbol{\chi}_i = \eta(\mathbf{q}_i, s) = \begin{bmatrix} \sigma_{\text{sh},i} \frac{\sin(\kappa_{\text{be},i} s)}{\kappa_{\text{be},i}} + (1+\sigma_{\text{ax},i}) \frac{\cos(\kappa_{\text{be},i} s)-1}{\kappa_{\text{be},i}} \\ \sigma_{\text{sh},i} \frac{1-\cos(\kappa_{\text{be},i} s)}{\kappa_{\text{be},i}} + (1+\sigma_{\text{ax},i}) \frac{\sin(\kappa_{\text{be},i} s)}{\kappa_{\text{be},i}} \\ \kappa_{\text{be},i} s \end{bmatrix}, \tag{3}$$

and

$$\mathbf{q}_i = \xi(\boldsymbol{\chi}_i, s) = \begin{bmatrix} \theta_i/s \\ \theta_i \left(p_{y,i} - \frac{p_{x,i}\sin(\theta_i)}{\cos(\theta_i)-1}\right)/2s \\ -1 + \theta_i \left(-p_{x,i} - \frac{p_{y,i}\sin(\theta_i)}{\cos(\theta_i)-1}\right)/2s \end{bmatrix}, \quad (4)$$

respectively. Notice that (3) and (4) have no singularity for $\kappa_{\text{be},i} = 0$ and $\theta_i = 0$. The limit in this situation where no curvature occurs in the segment is well defined and is equal to $\begin{bmatrix}\sigma_{sh}s & (1+\sigma_{ax})s & 0\end{bmatrix}^\top$ and $\begin{bmatrix}0 & p_{x,i}/s & -1+p_{y,i}/s\end{bmatrix}^\top$, respectively. However, some numerical instabilities might arise in practice [27].

B. Lagrangian Dynamics

The dynamics of physical systems are commonly expressed using the Lagrangian mechanics. According to this, an n_q-DOF system can be described by a set of generalized coordinates $\mathbf{q} \in \mathbb{R}^{n_q}$, their velocities $\dot{\mathbf{q}} \in \mathbb{R}^{n_q}$, and a scalar quantity known as the Lagrangian, expressed as

$$L(\mathbf{q}, \dot{\mathbf{q}}) = T(\mathbf{q}, \dot{\mathbf{q}}) - V(\mathbf{q}) \,. \quad (5)$$

The kinetic energy $T(\mathbf{q}, \dot{\mathbf{q}})$ is given by $T(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2}\dot{\mathbf{q}}^\top \mathbf{M}(\mathbf{q})\dot{\mathbf{q}}$, with $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n_q \times n_q}$ being the positive definite mass matrix. The potential energy $V(\mathbf{q})$ includes the gravitational and elastic potentials. Applying the principle of least action to the Lagrangian yields the Euler-Lagrange equations, describing the system's dynamics:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q}} = \boldsymbol{F}_{\text{ext}} \,, \quad (6)$$

where $\boldsymbol{F}_{\text{ext}} \in \mathbb{R}^{n_q}$ represents all non-conservative forces. We here consider external forces to be restricted to velocity-dependent dissipative forces $\boldsymbol{F}_d \in \mathbb{R}^{n_q}$ and actuation forces $\boldsymbol{F}_a \in \mathbb{R}^{n_q}$, which generally are given by

$$\boldsymbol{F}_{\text{ext}} = \boldsymbol{F}_d + \boldsymbol{F}_a = -\mathbf{D}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{A}(\mathbf{q})\boldsymbol{\tau} \,, \quad (7)$$

where $\mathbf{D}(\mathbf{q}) \in \mathbb{R}^{n_q \times n_q}$ is the damping matrix and $\mathbf{A}(\mathbf{q})\boldsymbol{\tau}$ represent the actuator forces applied on the robot. $\boldsymbol{\tau} \in \mathbb{R}^p$ is the control input and $\mathbf{A}(\mathbf{q}) \in \mathbb{R}^{n_q \times p}$ is the matrix that maps the point of application of the actuation to the configuration space. In this work, we will assume that $\mathbf{D}$ is diagonal and constant, and that actuators are directly collocated on the configurations, resulting in $\boldsymbol{\tau} \in \mathbb{R}^n_q$ and $\mathbf{A}(\mathbf{q})$ being the identity matrix.

By expanding (6) and after some manipulation, this set of equations can be rearranged into a more convenient matrix form,

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \mathbf{K}\mathbf{q} + \mathbf{D}\dot{\mathbf{q}} = \boldsymbol{\tau} \,. \quad (8)$$

Here, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \in \mathbb{R}^{n_q}$ represents the Coriolis and centrifugal force contribution, while $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^{n_q}$ and $\mathbf{K}\mathbf{q}$ denote the gravitational and elastic actions, where $\mathbf{K} \in \mathbb{R}^{n_q \times n_q}$ is the diagonal stiffness matrix.

This dynamic formulation can be derived for a PCS soft robot parametrization by combining its kinematic model with the inertial and impedance properties of its structure. The details for the derivations of each of these terms can be found in [27].

III. Methodology

In this work we propose an end-to-end strategy to find kinematic and dynamic models for planar continuum robots from image-recorded trajectories. Figure 1 shows an overview of the proposed methodology. We start with images of multiple trajectories, which are processed using computer vision (CV) to estimate the poses of N points along the soft robot. An initial N-segment PCS model is considered to obtain the robot's configuration through inverse kinematics (IK). We employ a kinematic regression procedure to reduce it to m segments while preserving accuracy. Finally, dynamic model identification is performed, estimating the dynamic parameters and eliminating negligible strains, resulting in a low-dimensional model suitable for control applications.

A. Shape Estimation from Images

The first step toward identifying a kinematic and dynamic model consists of collecting data of the system. A camera is placed parallel to the robot's plane of motion and records several trajectories of the robot. The goal is to extract the poses of N equally distant cross-sections along the robot. The value of N should be significantly larger than the expected number of segments required to model the robot's behaviour accurately.

Fig. 3 illustrates the steps involved in this process. For each frame, the image is binarized and, by marking the N points with visually salient features, the contours of the cross-sections are identified, and their center position (p_x, p_y) and orientation θ are extracted. For T image frames, this results in a time sequence of poses $\{\boldsymbol{\chi}_i(1), ..., \boldsymbol{\chi}_i(k), ..., \boldsymbol{\chi}_i(T)\}$, with $i \in [1, N]$.

The videos are analyzed through an algorithm in Python that uses functions from the OpenCV library to detect the cross-sections (`findContours`) and obtain their center positions and orientations (`minAreaRect`).

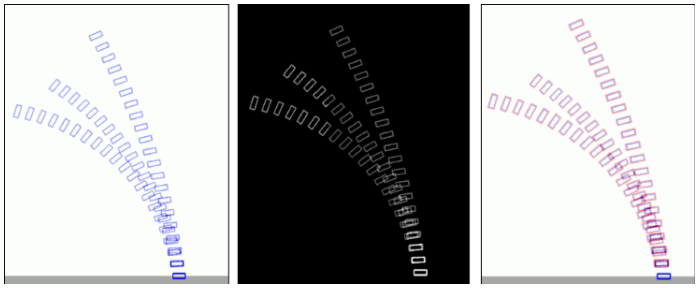


Fig. 3: Image analysis pipeline to obtain the Cartesian pose of N cross-sections, applied to a sequence of video frames. From left to write: 1) original image; 2) binarized image; 3) original image superimposed with the fitted rectangles.

B. Kinematic Regression

To develop a PCS kinematic model for a generic soft robot, it is necessary to determine the number of segments and the length of each segment. In this work, we achieve this by means of a strain-based algorithm.

Given that we now have access to the poses of N equally distant cross-sections along the robot, we initialize an N-segment PCS kinematic model, with nodes positioned at the center of the N cross-sections. Essentially, each of the N tracked cross-sections will consist of a segment's end section. We apply the closed-form inverse kinematics from (4) to map the Cartesian poses $\boldsymbol{\chi}_i$ into configuration variables $\mathbf{q}_i$. To avoid the numerical instabilities near the straight configuration ($\theta = 0$), a small ε is added to the bending angle. It is important to note that $[p_x, p_y, \theta]$ in (4) must be written with respect to the previous frame S_{i-1}. However, the computer vision step outputs all the poses $\boldsymbol{\chi}_i$ with respect to the fixed frame attached to the base of the soft robot. Therefore, a mapping must be applied to go from ${}^{0}\boldsymbol{\chi}_i$ into ${}^{i-1}\boldsymbol{\chi}_i$. This is achieved using the composition operation

$$ {}^{i-1}\boldsymbol{H}_i = {}^{i-1}\boldsymbol{H}_0 \, {}^{0}\boldsymbol{H}_i = \left({}^{0}\boldsymbol{H}_{i-1}\right)^{-1} {}^{0}\boldsymbol{H}_i \,, \tag{9} $$

where ${}^{0}\boldsymbol{H}_i \in SE(2)$ is the homogeneous transformation matrix that represents pose ${}^{0}\boldsymbol{\chi}_i$. For the planar case, this is given by

$$ {}^{0}\boldsymbol{H}_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & p_x^i \\ \sin\theta_i & \cos\theta_i & p_y^i \\ 0 & 0 & 1 \end{bmatrix} . \tag{10} $$

Once ${}^{i-1}\boldsymbol{H}_i$ has been computed, ${}^{i-1}\boldsymbol{\chi}_i$ can be extracted from the matrix entries.

After computing the sequence of configurations for the N initial segments, we employ a recursive algorithm to determine the final number of segments m, where $m < N$. The idea is that adjacent segments that have similar strain trajectories (i.e., deform similarly) can be merged together and considered a single segment. Note that each local strain have distinct units (bending strain has units m^{-1}, while shear and axial strains are unitless). In order to compute a metric that measures strain similarity over the entire strain-space, we apply a normalization by scaling each strain relative to their maximum values. In this work, the bending strain is not greater than $60\,\mathrm{m}^{-1}$, while shear and axial strains achieve maximum values of $30\,\%$.

The metric to measure strain similarity is the average strain-space (euclidean) distance between pairs of consecutive segments,

$$ \bar{d}_{i,i+1} = \frac{1}{T} \sum_{k=1}^{T} \left\| \bar{\mathbf{q}}_i(k) - \bar{\mathbf{q}}_{i+1}(k) \right\| , \tag{11} $$

where $\bar{\mathbf{q}}_i$ represent the i-th segment's scaled strains.

We compute this average strain distance for each pair of adjacent segments. For a kinematic model with n_S segments, this yields $n_S - 1$ pairs. If the distance $\bar{d}_{i,i+1}$ for the i-th pair is below or equal to a pre-defined threshold h, segments i and $i+1$ will be grouped into a single segment. Otherwise, the segments remain separate. This results in a new PCS kinematic model with fewer segments. Figure 4 illustrates this procedure. The configuration of each new merged segment is determined by performing a one-segment inverse kinematics on the distal ends of the merged segment.

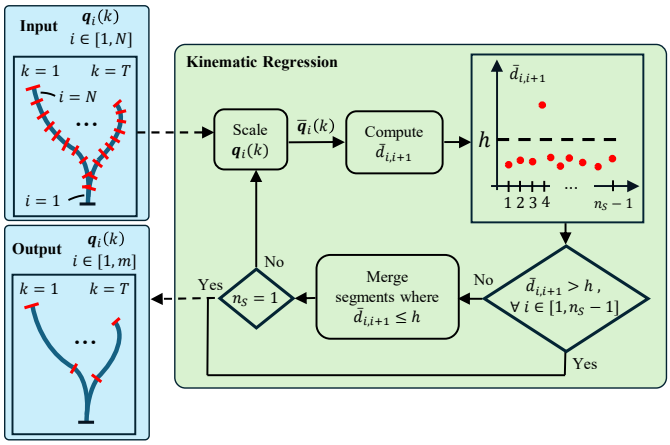


Fig. 4: Schematic of the kinematic regression algorithm. We start with a sequence of configurations (bending, shear, and axial strains) assuming an N-segment PCS model. The configurations are scaled and we analyze the similarity of the strain trajectories through the average strain distance between pairs of adjacent segments, $\bar{d}_{i,i+1}$. The segments where the distance falls below a threshold h are merged together. This process is repeated until no more merging is possible, resulting in a kinematic model with fewer segments ($m < N$).

This sequence is repeated until no more merging is possible, which can occur either if $\bar{d}_{i,i+1}$ is larger than h for every pair of segments or the model gets reduced to a single segment. Algorithm 1 illustrates an overview of all the steps.

Algorithm 1 Kinematic Regression

Input: Configurations $\mathbf{q}_i(k)$ $\{i = 1, ..., N\}$
Output: Configurations $\mathbf{q}_i(k)$ $\{i = 1, ..., m \ (m < N)\}$

repeat
  $\bar{\mathbf{q}}_i \leftarrow \mathbf{q}_i / \mathbf{q}_{\max}$ {Scale $\mathbf{q}_i$}
  merge_candidates $\leftarrow [\,]$
  current_merge $\leftarrow [1]$
  for $i = 1$ **to** $n_S - 1$ **do**
    $\bar{d}_{i,i+1} \leftarrow \frac{1}{T} \sum_{k=1}^{T} \|\bar{\mathbf{q}}_i(k) - \bar{\mathbf{q}}_{i+1}(k)\|$
    if $\bar{d}_{i,i+1} \le h$ **then**
      Append $i + 1$ to *current_merge*
    else
      Append *current_merge* to *merge_candidates*
      Reinitialize *current_merge* $\leftarrow [i + 1]$
    end if
  end for

  for each group in *merge_candidates* **do**
    $\mathbf{q}_{\text{merged}} = \text{IK}({}^{\text{init}}\boldsymbol{\chi}_{\text{end}})$
    {${}^{\text{init}}\boldsymbol{\chi}_{\text{end}}$ is the pose of the end section of the merged segment relative to the initial section}
  end for
until $n_S = 1$ **or** $\bar{d}_{i,i+1} > h \,, \forall\, i \in [1, n_S - 1]$

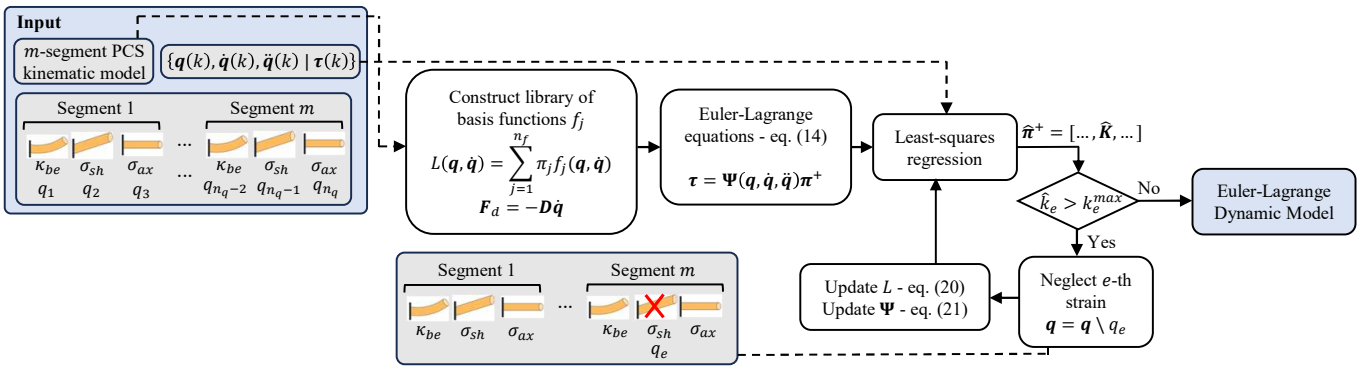


Fig. 5: Schematic of the dynamic model identification process. Starting with the m-segment PCS kinematic model, a library of basis functions is constructed to parameterize the system's Lagrangian, incorporating also dissipative forces. A regression framework is established using the Euler-Lagrange equations, and the dynamic parameters $\hat{\boldsymbol{\pi}}^+$ are estimated through linear least-squares. Strains with stiffness magnitudes exceeding a predefined threshold are neglected, prompting adjustments to the basis functions. The dynamic parameters are re-estimated iteratively until the stiffness threshold produces no more effects.

C. Dynamic Model Identification

After deriving the kinematic model, we proceed to identify the dynamic model using a data-driven approach. Figure 5 illustrates this process. The method begins by constructing a library of basis functions from the PCS dynamic model and employs a regression framework based on the Euler-Lagrange equations to estimate the system's dynamic parameters. The resulting dynamic model is then sparsified by removing non-essential strains based on the stiffness estimates. Sparsifying the strains is an important step because strain-based modelling approaches, such as the PCS model, though lower in dimension compared to FEM models, can still become computationally expensive, especially when multiple segments are considered [28]. Excluding rigid or minimally contributing dynamics leads to a more efficient representation without a large sacrifice on accuracy.

1) Construction of Library of Basis Functions based on the PCS Dynamic Model: In this work, we will perform a system identification to estimate the Lagrangian of the soft robot. Let us consider a soft manipulator with configuration variables $\mathbf{q} \in \mathbb{R}^{n_q}$. We leverage the knowledge of the previously found kinematic model to parametrize the Lagrangian as a linear combination of monomial basis functions,

$$L(\mathbf{q}, \dot{\mathbf{q}}) = \sum_{j=1}^{n_f} \pi_j f_j(\mathbf{q}, \dot{\mathbf{q}}) \,, \tag{12}$$

where f_j represent the basis functions within the Lagrangian of an m-segment planar PCS model, and π_j are the corresponding coefficients. The value m denotes the number of segments determined during the kinematic regression phase. The basis functions are obtained from the closed-form derivation of the kinetic and potential energies for the PCS model [27]. Note that we start with the Lagrangian basis functions of a PCS model that assumes all possible strains are present (three per CS segment).

One limitation of using the Lagrangian as the descriptor of a system's dynamics is that it does not inherently incorporate non-conservative forces, such as the intrinsic damping in the soft robot' structure. Therefore, to include this in the dynamic model, we introduce the dissipative forces as

$$\boldsymbol{F}_d = -\mathbf{D}\dot{\mathbf{q}} \,, \tag{13}$$

where $\mathbf{D} \in \mathbb{R}^{n_q \times n_q}$ is a diagonal matrix that holds the damping coefficients to be identified.

2) Regression of Dynamic Parameters: We resort to the Euler-Lagrange equations to set a regression framework that allows the estimation of the parameters. We define the Euler-Lagrange equations as

$$\begin{aligned} \boldsymbol{\tau} &= \frac{\partial^2 L}{\partial \dot{\mathbf{q}}^2}\ddot{\mathbf{q}} + \frac{\partial^2 L}{\partial \mathbf{q}\,\partial \dot{\mathbf{q}}}\dot{\mathbf{q}} - \frac{\partial L}{\partial \mathbf{q}} + \mathbf{D}\dot{\mathbf{q}} = \\ &= \sum_{j=1}^{n_f}\left[\pi_j\left(\frac{\partial^2 f_j}{\partial \dot{\mathbf{q}}^2}\ddot{\mathbf{q}} + \frac{\partial^2 f_j}{\partial \mathbf{q}\,\partial \dot{\mathbf{q}}}\dot{\mathbf{q}} - \frac{\partial f_j}{\partial \mathbf{q}}\right)\right] + D\dot{\mathbf{q}} = \\ &= \sum_{j=1}^{n_\psi} \pi_j^+ \boldsymbol{\psi}_j(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \in \mathbb{R}^{n_q} \,, \end{aligned} \tag{14}$$

where $\boldsymbol{\psi}_j$ denote the Euler-Lagrange basis functions and π_j^+ the associated coefficients, which comprise the Lagrangian basis function coefficients π_j along with the damping coefficients, $\mathrm{diag}(\mathbf{D})$. Therefore, $n_\psi = n_f + n_q$.

We can then formulate the Euler-Lagrange equations (14) in a matrix form as a linear combination

$$\boldsymbol{\tau} = \boldsymbol{\Psi}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\pi}^+ \,, \tag{15}$$

where $\boldsymbol{\Psi} \in \mathbb{R}^{n_q \times n_\psi}$ and $\boldsymbol{\pi}^+ \in \mathbb{R}^{n_\psi \times 1}$. In order to estimate the coefficients, we formulate a linear regression problem that accommodates the dataset,

$$\boldsymbol{\mathcal{T}} = \boldsymbol{X}\boldsymbol{\pi}^+ \tag{16}$$

where $\boldsymbol{\mathcal{T}} = [\boldsymbol{\tau}^\top(1), ..., \boldsymbol{\tau}^\top(T)]^\top \in \mathbb{R}^{Tn_q \times 1}$ and $\boldsymbol{X} = [\boldsymbol{\Psi}(\mathbf{q}(1), \dot{\mathbf{q}}(1), \ddot{\mathbf{q}}(1))^\top, ..., \boldsymbol{\Psi}(\mathbf{q}(T), \dot{\mathbf{q}}(T), \ddot{\mathbf{q}}(T))^\top]^\top \in \mathbb{R}^{Tn_q \times n_\psi}$. We set the optimization problem as a linear least squares regression,

$$\min \|\boldsymbol{\mathcal{T}} - \boldsymbol{X}\boldsymbol{\pi}^+\|_2^2 \,, \tag{17}$$

TABLE I: Kinematic and dynamic parameters of the manipulators used in the experiments. All segments across the manipulators have a constant circular cross-section with radius $r = 0.02\,\mathrm{m}$. The 1-segment PAC manipulator does not contain any dynamic parameters since it was implemented only as a kinematic simulator.

Case	Segment lengths (L) [m]	Mass density (ρ) [kg/m^3]	Young's modulus (E) [Pa]	Shear modulus (G) [Pa]	Damping matrix (D)
1-segment PCS	[0.1]	[1070]	[1e4]	[1e3]	diag([1e-5; 1.5e-1; 1.5e-1])
2-segment PCS	[0.07; 0.1]	[1070; 1049]	[1e4; 9e3]	[1e3; 3e3]	diag([3e-5; 5e-1; 5e-2; 3e-5; 2.5e-2; 5e-2])
3-segment PCS	[0.05; 0.1; 0.06]	[1240; 1049; 1070]	[1e4; 6e4; 8e4]	[1e3; 3e3; 5e3]	diag([5e-5; 4e-2; 5e-2; 5e-5; 5e-3; 5e-2; 5e-5; 5e-3; 5e-2])
1-segment PCS w/ high shear stiff.	[0.1]	[1070]	[1e4]	[1e6]	diag([1e-5; 5e0; 1.5e-1])
1-segment PAC	[0.15]	-	-	-	-

where we want to find the coefficient vector $\boldsymbol{\pi}^+$ that minimizes the mean squared error between the estimated actuation $\boldsymbol{X}\boldsymbol{\pi}^+$ and the actual one $\boldsymbol{\mathcal{T}}$ for all the samples. The closed-form solution for this problem is given by

$$\widehat{\boldsymbol{\pi}}^+ = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{\mathcal{T}}\,. \tag{18}$$

3) Sparsification of strains: This dynamic identification method offers the advantage of having interpretable results, as each estimated coefficient has some physical meaning within the PCS dynamic derivation. Specifically, among those we can extract the estimated stiffness matrix $\hat{\mathbf{K}}$, allowing us to assess the relevance of each strain by looking at its stiffness magnitude $\hat{k}_i$. A strain with high stiffness exhibits low displacement, approximating rigid behaviour. Therefore, such strain can be considered non-essential and neglected in the dynamics.

For this, we define a maximum stiffness, which is commonly modelled using the cross-section geometry and the material properties [29]. For a planar segment with constant cross-section, this is given by

$$\begin{aligned}\mathbf{K}^{\max} &= \operatorname{diag}\begin{bmatrix} k_{\mathrm{be}}^{\max} & k_{\mathrm{sh}}^{\max} & k_{\mathrm{ax}}^{\max}\end{bmatrix} \\ &= \operatorname{diag}\begin{bmatrix} I_c E^{\max} & A_c G^{\max} & A_c E^{\max}\end{bmatrix}\,,\end{aligned} \tag{19}$$

which hold the bending, shear and axial items. $E^{\max}$ and $G^{\max}$ are the Young's and shear modulus, respectively. A_c and I_c are the cross-section area and second moment of inertia, respectively. Given this, the e-th strain is neglected if its estimated stiffness is larger than the maximum, $\hat{k}_e > k_e^{\max}$.

Having reached this result, we must perform a new regression of the parameters to find the sparser dynamic model. For this, the configuration vector is updated by excluding the e-th strain q_e ($\mathbf{q} = \mathbf{q} \setminus q_e$), and the Lagrangian basis functions are adjusted such that the updated Lagrangian parameterization is given by

$$L(\mathbf{q}, \dot{\mathbf{q}}) = \sum_{j=1}^{n_f} \pi_j \lim_{(q_e, \dot{q}_e)\to 0} f_j(\mathbf{q}, \dot{\mathbf{q}})\,. \tag{20}$$

Similarly, the Euler-Lagrange basis functions $\boldsymbol{\Psi}$ can be updated by removing the e-th row and taking the following limit to the remaining entries,

$$\lim_{(q_e, \dot{q}_e, \ddot{q}_e)\to 0} \boldsymbol{\Psi}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\,. \tag{21}$$

Any columns that turn into all-*zeros* are also removed. Notice, therefore, that the library of basis functions is only derived once in the beginning, considering all possible strains. Later, in case of strain removal, only the above adjustments are performed to the set. After this, the linear least squares is used to find the new set of parameters. The sequence of steps described in III-C2 and III-C3 is repeated until the stiffness threshold produces no effects.

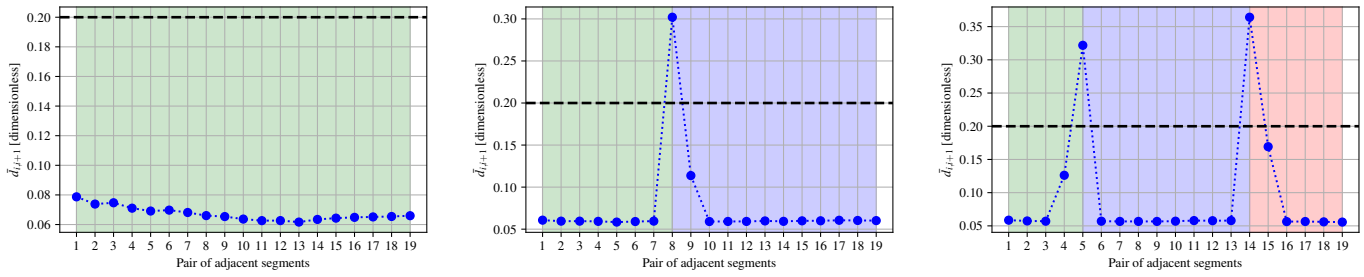
IV. Experiments

To validate the proposed method, the kinematic and dynamic regressions are tested in several simulation scenarios. Two simulators were used to generate the test cases: a planar PCS dynamic simulator and a planar Piecewise Affine Curvature (PAC) kinematic simulator. These simulators render videos of soft robot trajectories (see Fig. 3) to be used as input of the method's pipeline. In the end, the quality of the kinematic and dynamic models is assessed by comparing the estimated poses from both models with the ground-truth. The following subsections provide additional details and the results.

A. Simulation Setup

The planar PCS dynamic simulator used was introduced in [26]. This is an implementation of the mathematical formulations in II-A and II-B. Four different scenarios were created with this simulator. **Cases 1**, **2** and **3** are one-, two- and three-segment standard soft manipulators. **Case 4** is a one-segment manipulator where the shear strain stiffness was set to a large value to test the strain sparsification step of the dynamic regression.

To evaluate the robustness of the constant strain assumption in the models, a second planar simulator with distinct kinematics was used. It implements piecewise affine curvature (PAC)



(a) Case 1 (1-segment) (b) Case 2 (2-segment) (c) Case 3 (3-segment)

Fig. 6: Average strain distances between pairs of adjacent segments for the test cases generated using the PCS simulator. The poses of 20 cross-sections along the manipulators are tracked, resulting in 19 pairs of segments to be evaluated for strain similarity. The threshold is represented with a dashed line, and the background shading marks the resulting segments (separate segments are shaded with different colors).

[30] [31], in which the curvature of each segment $c_i(t)$ is no longer constant but rather affine in space,

$$c_i(t) = c_{0,i}(t) + c_{1,i}(t)s\,, \tag{22}$$

where $c_{0,i}$ and $c_{1,i}$ are the zero-order and first-order terms that define the curvature. The axial and shear strains are still considered constant in each segment. This simulator was used to generate **Case 5**, a one-segment PAC soft manipulator. Details on the manipulators used for the test cases are displayed in Table I.

Cases 1, 2 and 4 served to validate the method end-to-end, whereas Cases 3 and 5 were used to test only the kinematic regression part.

1) Data generation: For Cases 1 to 4, we generate eight trajectories with randomly sampled initial conditions and random actuation to excite the system every 10 milliseconds. Each trajectory produces a 0.5-second video captured at 1000 frames per second. For Case 5, since the PAC simulator only accounts for kinematics, we generate a single video featuring the robot in 500 randomly selected configurations $[c_0, c_1, \sigma_{sh}, \sigma_{ax}]$.

For the manipulators that are used to test the method end-to-end (Cases 1, 2 and 4), an additional 7-second trajectory is created to test the accuracy of the dynamic model. This trajectory is generated by applying a sinusoidal actuation sequence of the form $\boldsymbol{\tau} = \boldsymbol{f}(a_1 \sin(\omega_1 t) + a_2 \cos(\omega_2 t))$, where a_1 and a_2 are random amplitudes, and ω_1 and ω_2 are random frequencies.

For all the cases, we chose to track 20 equally distant cross-sections along the manipulators.

2) Evaluation metrics: To evaluate the models quantitatively, we introduce position and orientation task-space metrics. We use a **distributed** task-space mean absolute error (MAE), given by

$$e_p^{\text{dist}} = \frac{1}{NT}\sum_{k=1}^{T}\sum_{i=1}^{N} \|\widehat{\mathbf{p}}_i(k) - \mathbf{p}_i(k)\|_2 \tag{23}$$

and

$$e_\theta^{\text{dist}} = \frac{1}{NT}\sum_{k=1}^{T}\sum_{i=1}^{N} |\widehat{\theta}_i(k) - \theta_i(k)|\,, \tag{24}$$

where $\widehat{\mathbf{p}}_i(k)$ and $\widehat{\theta}_i(k)$ are the estimated position and orientation of point i along the structure, respectively, while $\mathbf{p}_i(k)$ and $\theta_i(k)$ are the ground-truth counterparts. These metrics give the average pose error across all T frames of a trajectory and all N cross-sections tracked along the robot, enabling a good evaluation of the kinematic model by capturing how well it represents the overall shape of the soft robot structure.

In addition, we also consider an **end-effector** task-space MAE given by

$$e_p^{\text{EE}} = \frac{1}{T}\sum_{k=1}^{T} \|\widehat{\mathbf{p}}_{EE}(k) - \mathbf{p}_{EE}(k)\|_2 \tag{25}$$

$$e_\theta^{\text{EE}} = \frac{1}{T}\sum_{k=1}^{T} |\widehat{\theta}_{EE}(k) - \theta_{EE}(k)|\,, \tag{26}$$

where $\widehat{\mathbf{p}}_{EE}(k)$ and $\widehat{\theta}_{EE}(k)$ are the estimated end-effector position and orientation, respectively, with $\mathbf{p}_{EE}(k)$ and $\theta_{EE}(k)$ being the ground-truth counterparts. This metric is particularly useful for assessing the obtained dynamic models, as the accuracy of the end-effector is crucial for an effective control performance and task execution.

B. Kinematic Regression Results

1) Cases 1, 2 and 3: The kinematic regression procedure was first tested for three instances of the planar PCS simulator: a one-, two- and three-segment manipulators (Cases 1, 2 and 3). Fig. 6 presents the average strain distances between the adjacent segment pairs for the three cases, following one iteration of the algorithm.

The plots for Cases 2 and 3 reveal one and two peaks, respectively, suggesting the locations for the segment breaks. In contrast, the smaller magnitude of the distance metric in Case 1, when compared to the other cases, indicates that all the segments could be merged into a single segment. The number of segments and respective lengths for the resulting models obtained with a threshold of $h = 0.2$ are shown in the second column of Table II. Note that these models are final because, for Cases 2 and 3, the updated strain trajectories for the newly defined segments are no longer sufficiently similar for further

merging. Also, for Case 1 no additional merging is possible since we already reached one single segment.

The distributed task-space errors between the resulting models and the ground truth is also evaluated. Knowing the final number of segments and their lengths, we determine the configurations $\mathbf{q}_i(k)$. Then, using the forward kinematics in (3), we obtain the estimated positions and orientations for the 20 tracked points along the robot. The results for these three cases are summarized in the first three rows of Table II.

TABLE II: Kinematic regression results for Cases 1, 2, 3 and 5. The second column contains the segment lengths for the obtained kinematic models. The distributed task-space errors for each model over the generated trajectories are stated in the third and fourth columns.

Case	Resulting model Segment lengths $(\widehat{L})$ [m]	e_p^{dist} [m]	e_θ^{dist} [rad]
1	[0.1]	8.21×10^{-5}	6.38×10^{-3}
2	[0.068; 0.102]	2.40×10^{-4}	1.15×10^{-2}
3	[0.0525; 0.0945; 0.063]	2.10×10^{-4}	9.67×10^{-3}
5	[0.0075; 0.105; 0.0375]	7.46×10^{-4}	8.92×10^{-2}

The kinematic regression was able to retrieve models that almost exactly match the manipulators used in the simulation. However, for Cases 2 and 3, the segments' lengths deviate slightly from the true ones (see second column of Table I). This is a consequence of the number of cross-sections tracked along the robots. Increasing this number would result in finer discretization and a closer approximation. Nonetheless, the position errors of the resulting models fall under 0.2% of the robots' lengths.

2) Case 5: We evaluate the robustness of the method with the PAC simulator, which no longer implements the constant bending assumption of the PCS model. The average strain distances for the initial 19 pairs of segments are presented in Fig. 7.

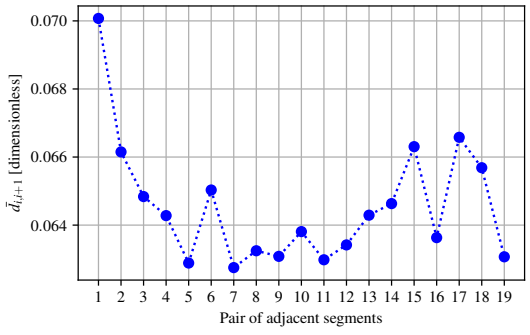


Fig. 7: Average strain distances for Case 5 after the first iteration of the kinematic regression algorithm.

Contrary to the results for the previous test cases, the plot does not make clear the exact locations where to merge segments. In this situation, we could think of the threshold as a hyperparameter and evaluate the quality of the models resultant from different values of h. Figure 8 shows the position and orientation errors as a function of the obtained number of segments, for several values of threshold. The choice for the final kinematic model depends on how the user wants to balance model complexity and shape reconstruction accuracy. For this case, we choose as the final model the one that maximizes the decrease in error over an increase in complexity (i.e., number of segments), which corresponds to the model with three segments. We can see that the 3-segment model exhibits a distributed position error of 0.5% of the robot's length, which verifies that this kinematic parametrization is suitable to approximate the robot's behaviour.

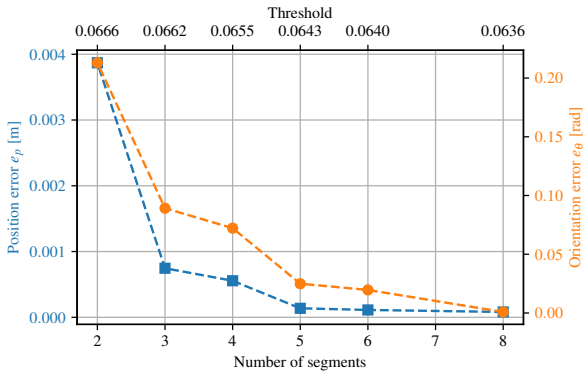


Fig. 8: Position (■) and orientation (●) distributed errors as a function of the number of segments considered for the kinematic model in Case 5.

C. Dynamic Model Identification Results

We now present and discuss the results of the dynamic model identification procedure. The procedure is the same across all the tested manipulators. From the kinematic regression step, we retrieve the kinematic model (with generic m-segments) and the respective configurations $\mathbf{q}(k) \in \mathbb{R}^{3m}$ for each of the 4000 time steps captured throughout the eight generated trajectories. The velocity $\dot{\mathbf{q}}(k)$ and acceleration $\ddot{\mathbf{q}}(k)$, also required for the regression, are numerically approximated using the Savitzky-Golay filter [32], with a window length of 25 and a third-order polynomial. We gather the basis functions of the m-segment PCS Lagrangian (considering all three strains per segment) and initialize the diagonal damping matrix $\mathbf{D} \in \mathbb{R}^{3m\times 3m}$ to be also identified. After running the least-squares regression and strain sparsification iteratively, we compute the Euler-Lagrange equations with the obtained model and retrieve the robot's equations of motion. We then simulate the model for the sinusoidal validation trajectory, using a Tsitouras 5(4) integrator [33] and a time-step of 0.1 ms.

1) Case 1: The Lagrangian of a one-segment PCS manipulator contains 51 basis functions. To those, we also add the 3 damping terms, yielding 54 coefficients to be identified.

After performing the least-squares regression, we inspect the estimated stiffness and compare it to the maximum defined stiffness. We choose $E^{\max} = 100\,\text{MPa}$ and $G^{\max} = 0.1\,\text{MPa}$, which are in line with the range of soft materials typically used [34]. For the cross-section area A_c and second moment of inertia I_c, we assumed a constant circular cross-section along

TABLE III: End-effector error metrics for the obtained dynamic models of Cases 1, 2, and 4, evaluated on a 7-second sinusoidal trajectory. For Cases 1 and 2, results are shown for models trained with and without noise. Case 4 presents end-effector errors for models where shear strain is either neglected or included.

Case	Noise	Strains Considered	e_p^{EE} [m]	e_θ^{EE} [rad]
1	Without noise	All	4.89×10^{-3}	1.13×10^{-1}
	With noise	All	1.37×10^{-2}	3.07×10^{-1}
2	Without noise	All	5.22×10^{-3}	1.38×10^{-1}
	With noise	All	1.68×10^{-2}	1.35×10^{-1}
4	Without noise	Without shear	4.57×10^{-3}	9.91×10^{-2}
	Without noise	All	5.14×10^{-3}	1.16×10^{-1}

the segment, with the same radius defined in the simulator. With the above parameters, the maximum stiffness is

$$\begin{aligned}\mathbf{K}^{\max} &= \operatorname{diag}\begin{bmatrix} k_{\mathrm{be}}^{\max} & k_{\mathrm{sh}}^{\max} & k_{\mathrm{ax}}^{\max} \end{bmatrix} \\ &= \operatorname{diag}\begin{bmatrix} 1.26 \times 10^{1} & 1.68 \times 10^{2} & 1.26 \times 10^{5} \end{bmatrix}. \end{aligned} \tag{27}$$

The estimated stiffness obtained from the regression is

$$\hat{\mathbf{K}} = \operatorname{diag}\begin{bmatrix} 1.20 \times 10^{-3} & 1.55 \times 10^{0} & 1.14 \times 10^{1} \end{bmatrix}. \tag{28}$$

Since all values are below the maximum, no strains will be neglected and the dynamic model is final.

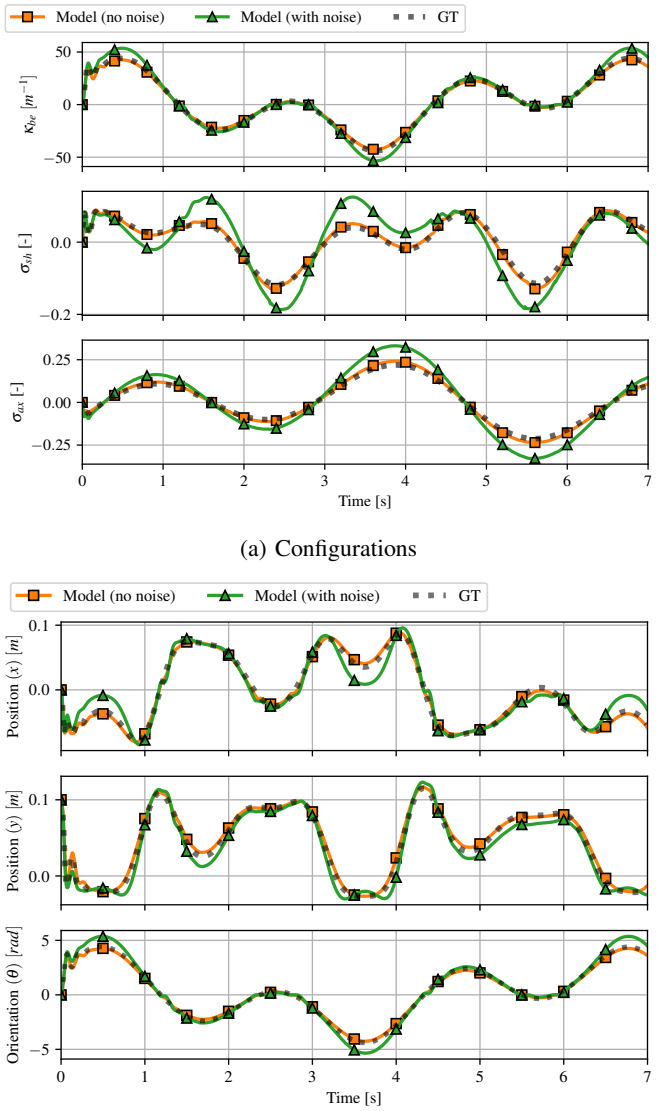
To evaluate the robustness of the method, we also corrupted the videos by adding zero-mean Gaussian noise to the position and orientation of each of the 20 cross-sections along the robot. Specifically, noise with a standard deviation of $5 \times 10^{-4}\,\mathrm{m}$ was added to both the x- and y-position measurements, while a standard deviation of $1\,\mathrm{deg}$ was applied to the orientation. Figure 9 shows the comparison between the models trained with and without noise.

The task-space error analysis, reported on Table III, show that the model trained without noise is able to give very accurate predictions, with the end-effector position error being under 5% of the robot's length. The model trained from noisy data can still provide reasonable predictions, even though the position error increases to $1.37\,\mathrm{cm}$ and the orientation error rises to $0.3\,\mathrm{rad}$.

An actual visualization of the trajectory also allows us to see that the end-effector error tends to increase after the robot crosses its straight configuration. Figure 10 shows a sequence of stills from the trajectory where this behaviour is noticeable. A possible reason for this is that, in the simulator used for generating the data, a small ε is added to the bending strain to avoid numerical instabilities due to divisions by zero. This introduces a minor nonlinearity that is not accounted for in the system's Lagrangian, and consequently, it is not captured by the learned model.

2) Case 2: The Lagrangian of a two-segment PCS manipulator has 438 basis functions. In total, that leads to 444 coefficients to be estimated (438 plus 6 damping terms). Using also $E^{\max} = 100\,\mathrm{MPa}$ and $G^{\max} = 0.1\,\mathrm{MPa}$, and assuming the same cross-section radius for both segments of $0.02\,\mathrm{m}$, the maximum stiffness is

$$\begin{aligned}\mathbf{K}^{\max} &= \operatorname{diag}\begin{bmatrix} k_{\mathrm{be},1}^{\max} & k_{\mathrm{sh},1}^{\max} & k_{\mathrm{ax},1}^{\max} & k_{\mathrm{be},2}^{\max} & k_{\mathrm{sh},2}^{\max} & k_{\mathrm{ax},2}^{\max} \end{bmatrix} \\ &= \operatorname{diag}\begin{bmatrix} 1.26 \times 10^{1} & 1.68 \times 10^{2} & 1.26 \times 10^{5} \\ & 1.26 \times 10^{1} & 1.68 \times 10^{2} & 1.26 \times 10^{5} \end{bmatrix}. \end{aligned} \tag{29}$$



(a) Configurations

(b) End-effector pose

Fig. 9: Verification of the model obtained for Case 1 on a sinusoidal trajectory. The dotted line denotes the ground-truth (GT) trajectory, while the green and orange lines correspond to the models trained with and without noise, respectively.

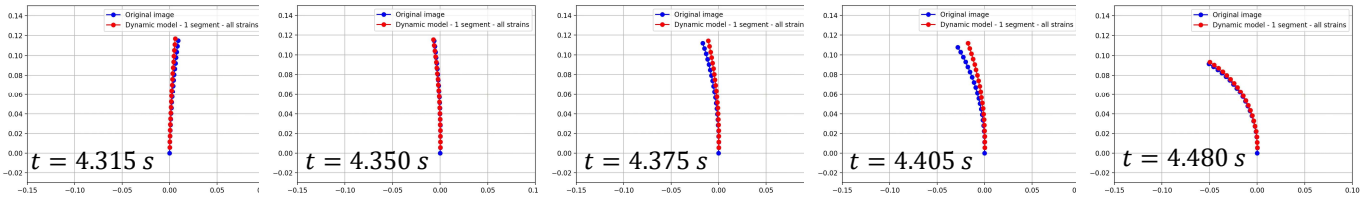


Fig. 10: Sequence of stills for the validation trajectory of Case 1, with the model trained without noisy data. The blue dots represent the actual position of the soft robot, while the red dots mark the position of the learned dynamic model.

After the least-squares regression, the estimated stiffness coefficients are

$$\widehat{\mathbf{K}} = \operatorname{diag}\begin{bmatrix} 1.07 \times 10^{-3} & 1.31 \times 10^{0} & 1.09 \times 10^{1} \\ & 1.11 \times 10^{-3} & 4.28 \times 10^{0} & 9.60 \times 10^{0} \end{bmatrix}, \tag{30}$$

which means that no strains are neglected since all the values are smaller than the corresponding maximum.

As with the one-segment case, we also corrupted the videos by adding measurement noise (through a zero-mean Gaussian distribution) to the position and orientation of the cross-sections split along the robot. However, in this case, the method could only tolerate noise with a standard deviation of $1 \times 10^{-4}\,\mathrm{m}$ for the position and $0.5\,\mathrm{deg}$ for the orientation. Training the models with data corrupted by higher noise levels caused the prediction results to diverge, as the estimated coefficients produced a mass matrix that was not positive definite, a condition necessary to ensure the stability of the dynamics. Figure 11 reports the comparison between the models trained both with and without noise for the sinusoidal trajectory used as validation.

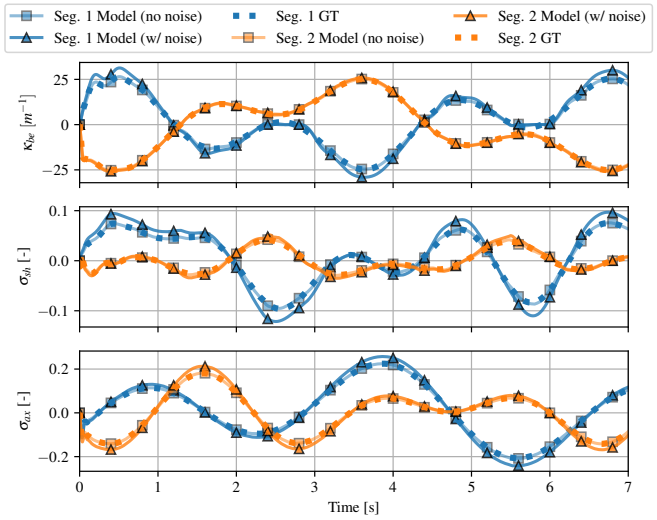
The model trained without noise reports great accuracy, with an end-effector position error of $5.22\,\mathrm{mm}$ (3% of the robot's length) and an orientation error of $0.14\,\mathrm{rad}$. An expected performance degradation is noticed for the model trained with noisy data, particularly for the end-effector position, which sees the error increase to $1.68\,\mathrm{cm}$. This raises some considerations for implementations in real-world scenarios, where measurement noise is inevitable. Thus, while the model still gives reasonable predictions to moderate levels of noise, applying noise filtering techniques during the kinematic regression process (where the measured task-space poses are converted into configurations) could help improve the robustness of the method to higher levels of noise.

3) Case 4: To evaluate the strain sparsification step in the dynamic regression, we used a one-segment PCS manipulator similar to Case 1, but with the shear strain stiffness set to a high value to simulate a scenario with minimal shear displacement.

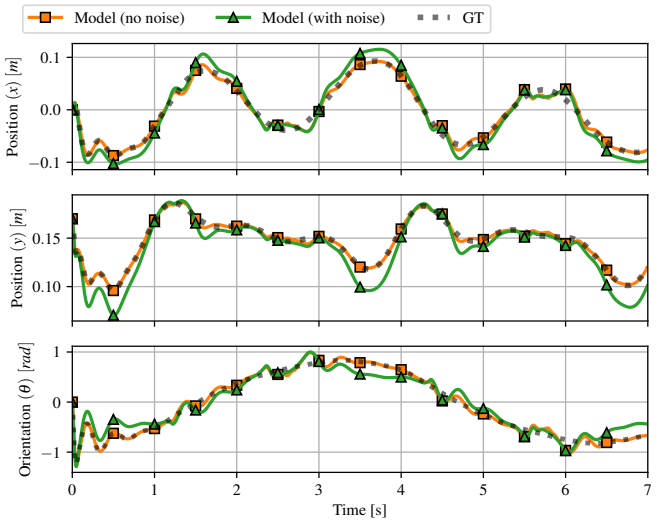
After running the regression, the obtained estimated stiffness was

$$\widehat{\mathbf{K}} = \operatorname{diag}\begin{bmatrix} 1.07 \times 10^{-3} & 1.20 \times 10^{3} & 1.16 \times 10^{1} \end{bmatrix}. \tag{31}$$

Comparing the values with the maximum stiffness in (27), the second entry, correspondent to the shear stiffness, is above the maximum value. Given this, the shear strain is neglected from the configuration vector and we update the Lagrangian basis functions through (20), which results in a reduction from 51 to



(a) Configurations



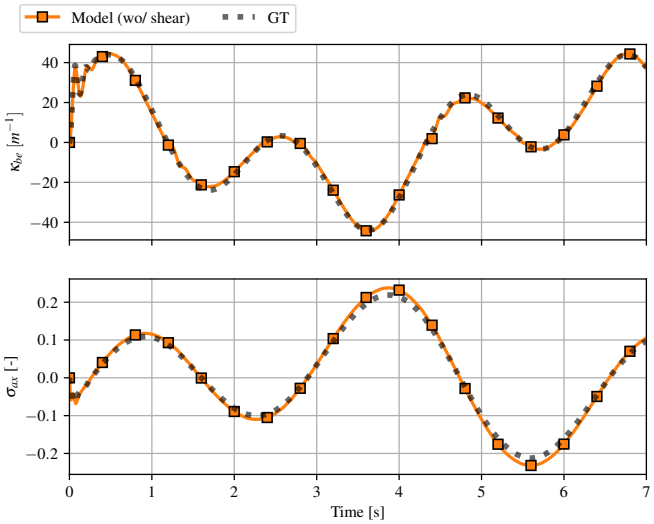
(b) End-effector pose

Fig. 11: Verification of the model obtained for Case 2 on a sinusoidal trajectory. The dotted lines denote the ground-truth (GT) trajectory. For the configuration plot, blue lines refer to the first segment while orange lines are associated with the second segment. For the end-effector pose plot, the orange and green lines mark the models trained with and without noise, respectively.

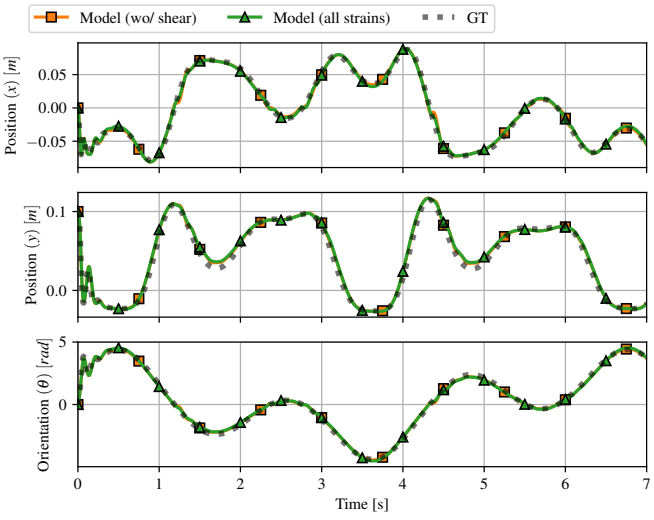
27 basis functions. The Euler-Lagrange basis functions are also updated through (21). The regression is again performed to estimate the 29 coefficients (27 plus the remaining 2 damping terms). In this case, the estimated stiffness is

$$\hat{\mathbf{K}} = \text{diag}\begin{bmatrix}\hat{k}_{\text{be}} & \hat{k}_{\text{ax}}\end{bmatrix} = \text{diag}\begin{bmatrix}1.21 \times 10^{-3} & 1.17 \times 10^{1}\end{bmatrix}, \tag{32}$$

which no longer requires additional sparsification. Figure 12 shows the results for rolling out this dynamic model over the sinusoidal trajectory. To observe the impact of neglecting the shear strain, we additionally plot the end-effector pose for the case where shear was still considered. The end-effector errors for this comparison are also reported in the last two rows of Table III.



(a) Configurations



(b) End-effector pose

Fig. 12: Verification of the model obtained for Case 4 on a sinusoidal trajectory. The dotted line denotes the ground-truth trajectory, while the green and orange lines correspond to the models trained with and without noise, respectively.

Figure 12a shows that, despite discarding the shear strain, the model can still accurately predict the behaviour of the remaining bending and axial strains. Interestingly, the model without shear strain performs similarly, even slightly better, in terms of end-effector accuracy, compared to the model that includes all strains. Although it could be expected that the higher order model would perform better, a possible explanation is that, with fewer basis functions parametrizing the Lagrangian, the regressed coefficients for the remaining strains are more accurately estimated.

V. Conclusion

This work presents a comprehensive approach to automatically identify the kinematics and dynamics of planar continuum soft robots, based on the Piecewise Constant Strain (PCS) model. Using only a few video-recorded trajectories, we successfully extracted kinematic models and used them as foundations to obtain dynamic models represented in standard Euler-Lagrange form. The proposed methodology offers a balance between model accuracy and computational efficiency, enabling real-time control applications.

The results demonstrated that our approach was able to provide accurate predictions of the robots' behavior in various simulated scenarios. The strain-based segmentation technique proved effective in achieving a good trade-off between model complexity and shape accuracy. Additionally, the dynamic identification method reliably estimated the system parameters while omitting negligible strains when necessary. However, while our method performed well under some levels of measurement noise, its robustness to higher noise levels leaves room for improvement.

Overall, this work provides a fast and practical modelling approach that integrates both physical insights and data-driven identification. Future work should focus on enhancing noise robustness (both process and measurement noise), adapting the method to more realistic actuation scenarios, validating it with real-world data, and extending the approach to three-dimensional soft robots.

Acknowledgment

The completion of this project would not be possible without the careful and always forgiving guidance of Maximilian and Jingyue. Thank you for pushing my boundaries and convincing me to always go a step further. A huge thanks to Cosimo for introducing me so passionately to the topic of soft robots and always advising me in the right direction.

I would also like to express my deepest gratitude to my parents, for always believing in me and for being my safe place through some difficult times in this journey. A heartfelt acknowledgment is reserved to my sister, for all the laughs, singing, and crazy ideas. Thank you for being an extension of myself.

Lastly, a special thanks to all my friends for their continuous support and for showing me that everyone trails their own beautiful path.

REFERENCES

[1] C. Della Santina, M. G. Catalano, and A. Bicchi, "Soft Robots," in *Encyclopedia of Robotics*. Springer, 2020, pp. 1–15.

[2] T. Ashuri, A. Armani, R. Jalilzadeh Hamidi, T. Reasnor, S. Ahmadi, and K. Iqbal, "Biomedical soft robots: current status and perspective," *Biomedical Engineering Letters*, vol. 10, no. 3, pp. 369–385, Aug. 2020.

[3] O. Yasa, Y. Toshimitsu, M. Y. Michelis, L. S. Jones, M. Filippi, T. Buchner, and R. K. Katzschmann, "An Overview of Soft Robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 6, pp. 1–29, May 2023.

[4] M. Giorelli, F. Renda, M. Calisti, A. Arienti, G. Ferri, and C. Laschi, "Learning the inverse kinetics of an octopus-like manipulator in three-dimensional space," *Bioinspiration & Biomimetics*, vol. 10, no. 3, p. 035006, May 2015.

[5] T. G. Thuruthel, E. Falotico, F. Renda, and C. Laschi, "Learning dynamic models for open loop predictive control of soft robotic manipulators," *Bioinspiration & Biomimetics*, vol. 12, no. 6, p. 066003, Oct. 2017.

[6] A. Tariverdi, V. K. Venkiteswaran, M. Richter, O. J. Elle, J. Tørresen, K. Mathiassen, S. Misra, and O. G. Martinsen, "A Recurrent Neural-Network-Based Real-Time Dynamic Model for Soft Continuum Manipulators," *Frontiers in Robotics and AI*, vol. 8, p. 631303, Mar. 2021.

[7] P. Hyatt, D. Wingate, and M. D. Killpack, "Model-Based Control of Soft Actuators Using Learned Non-linear Discrete-Time Models," *Frontiers in Robotics and AI*, vol. 6, Apr. 2019.

[8] G. Chen, X. Yang, Y. Xu, Y. Lu, and H. Hu, "Neural network-based motion modeling and control of water-actuated soft robotic fish," *Smart Materials and Structures*, vol. 32, no. 1, p. 015004, Dec. 2022.

[9] C. Armanini, F. Boyer, A. T. Mathew, C. Duriez, and F. Renda, "Soft Robots Modeling: A Structured Overview," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1728–1748, Jun. 2023.

[10] D. Kim, S.-H. Kim, T. Kim, B. B. Kang, M. Lee, W. Park, S. Ku, D. Kim, J. Kwon, H. Lee, J. Bae, Y.-L. Park, K.-J. Cho, and S. Jo, "Review of machine learning methods in soft robotics," *PLOS ONE*, vol. 16, no. 2, p. e0246102, Feb. 2021.

[11] L. Ding, L. Niu, Y. Su, H. Yang, G. Liu, H. Gao, and Z. Deng, "Dynamic Finite Element Modeling and Simulation of Soft Robots," *Chinese Journal of Mechanical Engineering*, vol. 35, no. 1, p. 24, Apr. 2022.

[12] C. Duriez, "Control of elastic soft robots based on real-time finite element method," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 3982–3987.

[13] P. Polygerinos, Z. Wang, J. T. B. Overvelde, K. C. Galloway, R. J. Wood, K. Bertoldi, and C. J. Walsh, "Modeling of Soft Fiber-Reinforced Bending Actuators," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 778–789, Jun. 2015.

[14] F. Boyer, V. Lebastard, F. Candelier, and F. Renda, "Dynamics of Continuum and Soft Robots: A Strain Parameterization Based Approach," *IEEE Transactions on Robotics*, vol. 37, no. 3, pp. 847–863, Jun. 2021.

[15] D. Trivedi, A. Lotfi, and C. D. Rahn, "Geometrically Exact Models for Soft Robotic Manipulators," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 773–780, Aug. 2008.

[16] F. Renda, M. Giorelli, M. Calisti, M. Cianchetti, and C. Laschi, "Dynamic Model of a Multibending Soft Robot Arm Driven by Cables," *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1109–1122, Oct. 2014.

[17] C. Della Santina, R. K. Katzschmann, A. Bicchi, and D. Rus, "Model-based dynamic feedback control of a planar soft robot: trajectory tracking and interaction with the environment," *The International Journal of Robotics Research*, vol. 39, no. 4, pp. 490–513, Mar. 2020.

[18] P. Schegg and C. Duriez, "Review on generic methods for mechanical modeling, simulation and control of soft robots," *PLoS ONE*, vol. 17, no. 1, p. e0251059, Jan. 2022.

[19] F. Renda, F. Boyer, J. Dias, and L. Seneviratne, "Discrete Cosserat Approach for Multisection Soft Manipulator Dynamics," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1518–1533, Dec. 2018.

[20] R. J. Webster and B. A. Jones, "Design and Kinematic Modeling of Constant Curvature Continuum Robots: A Review," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, Nov. 2010.

[21] Y. Toshimitsu, K. W. Wong, T. Buchner, and R. Katzschmann, "SoPrA: Fabrication & Dynamical Modeling of a Scalable Soft Continuum Robotic Arm with Integrated Proprioceptive Sensing," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2021, pp. 653–660.

[22] A. Garg, I. Good, D. Revier, K. Airis, and J. Lipton, "Kinematic Modeling of Handed Shearing Auxetics via Piecewise Constant Curvature," in *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)*, Apr. 2022, pp. 423–430.

[23] R. K. Katzschmann, C. D. Santina, Y. Toshimitsu, A. Bicchi, and D. Rus, "Dynamic Motion Control of Multi-Segment Soft Robots Using Piecewise Constant Curvature Matched with an Augmented Rigid Body Model," in *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, Apr. 2019, pp. 454–461.

[24] C. Della Santina, R. K. Katzschmann, A. Biechi, and D. Rus, "Dynamic control of soft robots interacting with the environment," in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*, Apr. 2018, pp. 46–53.

[25] M. Gazzola, L. H. Dudte, A. G. McCormick, and L. Mahadevan, "Forward and inverse problems in the mechanics of soft filaments," *Royal Society Open Science*, vol. 5, no. 6, p. 171628, Jun. 2018.

[26] M. Stölzle, D. Rus, and C. Della Santina, "An Experimental Study of Model-Based Control for Planar Handed Shearing Auxetics Robots," in *International Symposium on Experimental Robotics*. Springer Nature Switzerland, 2023, pp. 153–167.

[27] 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, Jun. 2023.

[28] A. Y. Alkayas, A. T. Mathew, D. Feliu-Talegon, P. Deng, T. G. Thuruthel, and F. Renda, "Soft Synergies: Model Order Reduction of Hybrid Soft-Rigid Robots via Optimal Strain Parameterization," 2024, arXiv:2405.12959.

[29] J. Shi, A. Shariati, S.-A. Abad, Y. Liu, J. S. Dai, and H. A. Wurdemann, "Stiffness modelling and analysis of soft fluidic-driven robots using Lie theory," *The International Journal of Robotics Research*, vol. 43, no. 3, pp. 354–384, Mar. 2024.

[30] F. Stella, Q. Guan, C. Della Santina, and J. Hughes, "Piecewise Affine Curvature model: a Reduced-Order Model for Soft Robot-Environment Interaction Beyond PCC," in *2023 IEEE International Conference on Soft Robotics (RoboSoft)*, Apr. 2023, pp. 1–7.

[31] F. Stella, N. Obayashi, C. D. Santina, and J. Hughes, "An Experimental Validation of the Polynomial Curvature Model: Identification and Optimal Control of a Soft Underwater Tentacle," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 410–11 417, Oct. 2022.

[32] A. Savitzky and M. J. E. Golay, "Smoothing and Differentiation of Data by Simplified Least Squares Procedures." *Analytical Chemistry*, vol. 36, no. 8, pp. 1627–1639, Jul. 1964.

[33] C. Tsitouras, "Runge–Kutta pairs of order 5(4) satisfying only the first column simplifying assumption," *Computers & Mathematics with Applications*, vol. 62, no. 2, pp. 770–775, Jul. 2011.

[34] N. Gariya and P. Kumar, "A review on soft materials utilized for the manufacturing of soft robots," *Materials Today: Proceedings*, vol. 46, pp. 11 177–11 181, Jan. 2021.