BOSTON UNIVERSITY

College of Engineering

Thesis Proposal

3-Dimensional Model-Based Dynamic Feedback Control for Soft Robots

by

Nashr El Auliya
B.Sc., Boston University, 2024

Submitted in partial fulfillment of the requirements for the degree of

Master of Science

2025

This page intentionally left blank.

Contents

1 Abstract

The physical characteristics of soft robots inherently promise an ability to perform complex motions, as well as to safely and compliantly interact with sensitive environments. While trajectory tracking and environmental interaction control strategies for planar motion have been developed along with motion plans for it, it has yet to be robustly translated to three dimensions. This proposed thesis thus aims to develop a three-dimensional, model-based, closed loop dynamic controller for continuous soft robots. To develop a robust formulation of this controller, gravitational loads that could potentially violate model assumptions must be dynamically accounted for. Kinematic singularities inherent to the dynamic model used must also be analytically or numerically managed. A suitable dynamic model to underpin the control system must first either be formulated, augmented from an existing one, or selected. Then the model must be validated either analytically or simulatively, before the control system can subsequently be built around it. The controller may then finally be validated through hardware implementation.

2 Topic Background

2.1 Model-based Controls in Soft Robotics

Much of controls theory as a field has progressed in parallel to the advances in our understanding of dynamic modeling. In fact, much of the advancements in controls theory were driven by the need to supplement the gaps in our dynamic models. Starting from the frequency domain and linear controllers, to nonlinear controllers, and most recently machine learning.

The implementation of control strategies for standard rigid robots has followed this line of progression. Algorithms built around the dynamical model of the robot itself came first, then only recently did strategies employing machine learning began to be implemented–mostly to handle unpredictable scenarios beyond the models' assumptions. Control strategies of the former kind are known as *model*-based controls, where controllers are designed based on models that mathematically represent the robot's dynamics [1]. It should be apparent that this has been the case since the dynamics of a rigid-bodied robot can more readily be modelled.

On the other hand, the development of control strategies for *soft* robots have followed the opposite direction: much of the early works in soft robotics controls employed machine learning strategies due to the high level of complexity and dimensionality required in the dynamical models. Thus, *learning*-based controls were the model for soft robot control strategies through the formative years of the field. Only in recent years did this precedent began to be reassessed, particularly with the advent of *finite*-dimensional modelling (FDM) techniques compatible with the continuum dynamics of soft robots [2].

2.2 Model- vs. Learning-based Controls

Model- and learning-based controls differ in the fundamental methodology that underpin *how* each approach steers their dynamic systems. Consider the soft robot whose dynamics is represented as $\underline{\dot{q}} = \mathbf{f}(\underline{q}, \underline{u})$, where $\underline{q} = [q_1, q_2, ..., q_n]^T \in \mathbb{R}^n$ is the state parametrization used to describe the robot's configuration in space and $\underline{u}$ is some input signal. The application of a model- and learning-based controller, respectively, to the system may be represented according to Table 1.

Model-based	Learning-based
Given $\underline{\dot{q}} = \mathbf{f}(\underline{q}, \underline{u}) \xrightarrow[\text{for}]{\text{solve}} \underline{u}(\underline{q}, \underline{\bar{q}}, ...)$ $\mathbf{f}(\underline{q}, \underline{u}) \to \mathbf{f}(\underline{q}, \underline{u}(\underline{q}, \underline{\bar{q}}))$ Such that $\underline{\dot{q}} \approx \underline{\bar{\dot{q}}}$	Given some dataset $X = \{(\underline{q_0}, \underline{u_0}), \ldots, (\underline{q_i}, \underline{u_i})\}$ $X \xrightarrow[\text{from}]{\text{extract}} \underline{u}^{\text{learned}}$ Such that $\underline{\dot{q}} \approx \underline{\bar{\dot{q}}}$

Table 1: High-level generalization of model- vs. learning-based controllers, where $\underline{\bar{\dot{q}}}$, $\underline{\bar{q}}$ describes the desired/reference configuration.

For model-based controls, designing the controller based on the dynamics of the systems means solving for $\underline{u}$ such that it steers the system's output configuration towards the reference configuration. Using the terminology of the framework presented above, this inherently requires an expression for $\mathbf{f}$ to be known. The model-based approach as a whole revolves around tailoring $\mathbf{f}$ such that it lends itself for solving $\underline{u}$.

Formulating models that allow the input to be solved in a mathematically manageable way, while simultaneously remaining faithfully representative of the *actual* system behavior becomes one of model-based controls' biggest challenges. Learning-based controls circumvent this by using data-driven techniques and learning algorithms to arrive at $\mathbf{u}$. In some cases, $\mathbf{f}$ may even be left unknown as the full system is wholly formulated by the learning medium.

Analysis	Model-Based	Learning-Based
Advantages	Meaningful information regarding the dynamics and inputs of the system are preserved	Known to achieve higher maneuvering performance and better robustness [1]
Disadvantages	Less robust and adaptable at handling system configurations in fringe scenarios [1], [3]	More complex and typically lacks formal guarantees of safety [3]

Table 2: An assessment of Model- and Learning-Based controls.

A general assessment of the benefits and drawbacks between the two approaches is presented in Table 2. While both approaches bring different things to the table–and extensive research continue to be conducted on both, the author has decided to explore the model-based approach for this proposed thesis.

3 Prior Work

3.1 The Piece-wise Constant-Curvature Model

It was mentioned earlier that in the formulation of the dynamical model to develop a controller around, towing the balance between mathematical simplicity and physical accuracy is one of the major challenges in model-based controls. While this continues to be true, continuum dynamics-compatible FDM techniques has also indeed broken considerable ground regarding this issue. In their exact formulation, the dynamics of a soft robot is effectively an infinite-dimensional system. Such a system cannot be described without the use of partial differential equations (PDEs). However, by applying the appropriate assumptions, approximations, and/or discretization to the dynamical model of the soft robot, the system's description may be reliably "minimized" into a *finite*-dimensional system that can be described with ordinary differential equations (ODEs) instead.

One of the most widely-implemented family of FDM techniques is the Piece-wise Constant Strain (PCS) approximations (see Figure 1). It is a family of discretization methods applied to a type of approximation for soft robot dynamics known as "rod models". It is extremely common for soft robots to be "thin" and/or "elongated": to have one physical dimension dominate the other two. In that regard, such soft robots may be approximated as a rod with the continuum mechanics of one too–hence rod models. At the heart of this methodology is the assumption that volumetric deformations may be neglected, and modeling the dynamics around the dominant central axis is sufficient [2].

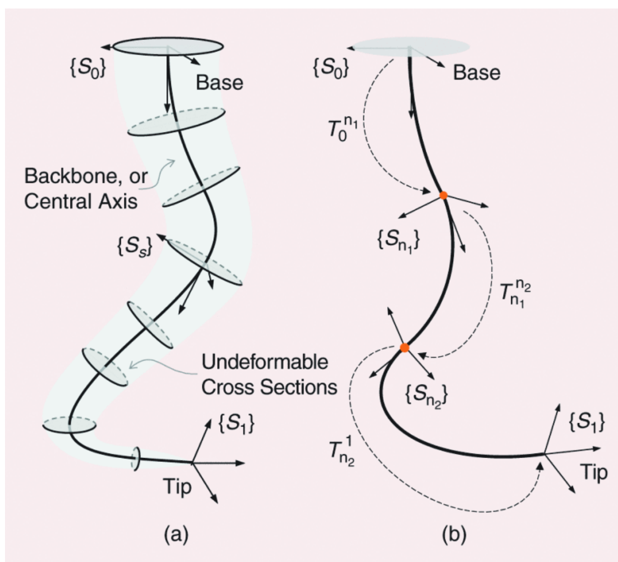


Figure 1: a) "Elongated" soft robots as described in rod models. b) PCS discretization applied to the rod model. Image taken from [2].

Among the many implementations of PCS, the planar Piece-wise Constant Curvature (PCC) model has been extensively used in soft robotics throughout the last decade. Its approximation of the robot as pieces of constant-curvature arcs linked together in series with mutually tangent connection points makes the kinematics and Jacobian formulation for the model *closed*-form [4]. A dynamic feedback controller using this model was developed in [5], with an implementation of a trajectory generator for the controller formulated in [6].

There are many viable models out there that can be used as the basis for the controller that this proposed thesis seeks to develop, a selection of them that seems promising with respect to the scope of this proposed thesis is outlined below.

3.2 PCC in an Augmented Rigid Body Model

The controller from [5] mentioned earlier uses the kinematics of PCC matched to the dynamics of an augmented rigid body, allowing control strategies typically implemented for rigid-body robots to be used in this scenario. This augmented formulation was initially proposed in [7] by Della Santina et al. as well.

Recall that the PCC Model essentially approximates a rod-like soft robot as a series of constant-curvature (CC) arc segments with mutually-tangent connections. Della Santina et al. proposed that for a given CC segment, the dynamics of the segment can be matched to a rigid-robotic structure comprised of Revolute-Prismatic-Prismatic-Revolute (RPPR) joints in series (see Figure 2). A point mass is located between the two prismatic joints and also matched to the CC segment's center of mass, ensuring the inertial properties of the rigid robot matches the CC segment [7].

Using this model, Della Santina et al. were able to develop two feedback controllers aimed at different implementations. The first controller is intended for trajectory tracking, while the second for Cartesian impedance control. While the model *and* controller does not cover three-dimensional motion, preliminary work into translating the model to three-dimensions is explored in [8]. Implementing this "augmented rigid body" model for this proposed thesis will involve expanding the preliminary work into the three-dimensional translation of the model, and then implementing the control design methodology already laid out in [5]. The body of literature available to build on regarding this model is one of the biggest appeals in using it for this proposed thesis. However, the kinematic singularity found in the three-dimensional straight configuration remains a challenge.

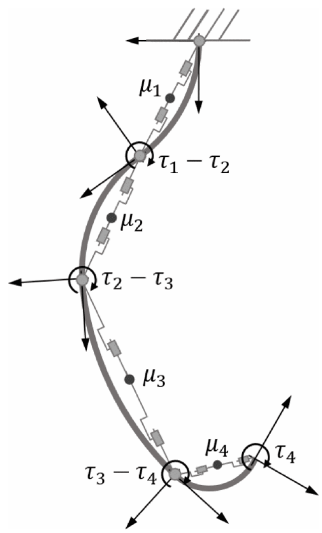


Figure 2: A series of CC Segments matched to a series of RPPR Robots. Image taken from [7].

3.3 PCC Using an Alternative State Parametrization

Another one of Cosimo Della Santina's works using the PCC model can be found in [9]. In this work, Della Santina et al. proposes an *alternative* state parametrization for PCC kinematics that describes the CC segments based on the arc lengths of four arcs that bound the surface of the segment.

In the preliminary work of [8], the PCC kinematics of a soft robot were parametrized according to Figure 3. In [9], Della Santina et al. refers to this state parametrization as the "α-parametrization". Kinematically speaking, the α-parametrization is defined as $\alpha_i = [\phi_i, \theta_i, \delta L_i]^T$ for the i-th CC segment, where $\alpha_i \in \mathbb{R}^3$. In this parametrization, ϕ_i and θ_i are the angles as indicated in Figure 3, while δL_i is the change in length of the soft robot's central axis.

The issue with this parametrization–which was mentioned in our discussion of the previous model–is that for a given physical configuration, it does not *uniquely* map to a vector α_i. In the case of the straight configuration for example, this means there are infinite choices of ϕ_i for $\theta_i = 0$. Giving rise to the kinematic singularity mentioned previously.

Now in [9], Della Santina et al. refers to the alternative state parametrization they propose as the "q-parametrization". Figure 4 illustrates how the i-th segment of a PCC soft robot is now represented instead.

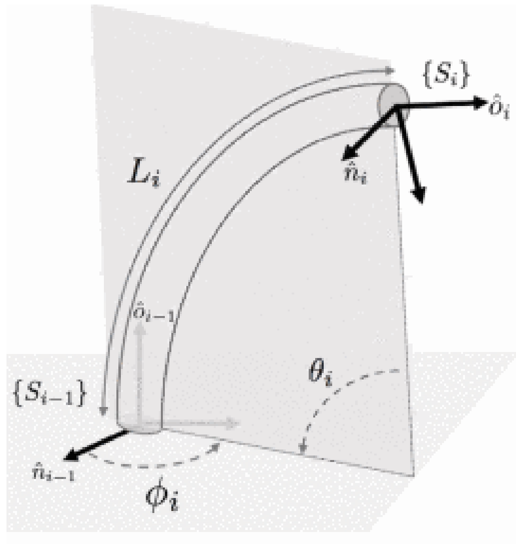


Figure 3: The standard or "old" state parametrization. Image taken from [8].

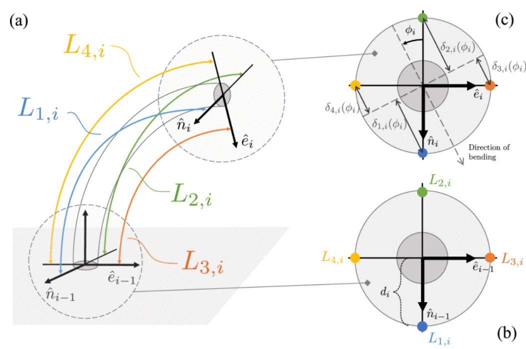


Figure 4: The alternative "Arc-Length Parametrization". Image taken from [9].

This new "Arc-Length Parametrization" defines a configuration q_i as $q_i = [\Delta_{x,i}, \Delta_{y,i}, \delta L_i]^T \in \mathbb{R}^3$. While explicit definitions of $\Delta_{x,i}$ and $\Delta_{y,i}$ will not be reported in this proposal, these two parameters essentially describe the difference in length between the two arcs whose ends are connected to the $\hat{n}$ (blue and green arcs) and $\hat{e}$ (yellow and orange arcs) axes, respectively.

With [9] proving that the q-parametrization is able to solve the limitations of the α-parametrization, its kinematic robustness does make this model a great option to move forward with for this proposed thesis.

4 Research Approach

The goal of this proposed thesis is to develop a three-dimensional, model-based, closed-loop dynamic controller for continuous soft robots. This proposal would now like to put forward a breakdown of how research in pursuit of the goal will be approached. The work shall be conducted in three phases: the Model Selection phase, Model Calibration phase, and finally the Controller Implementation phase, these phases are described below.

4.1 Model Selection

This proposal has discussed the importance and impacts of the constitutive dynamical model used when designing model-based controllers, and consequently how limitations born out of the model's assumptions can translate to implementation challenges in the controller. The Model Selection phase has been included in the research approach in light of this, specifically in the interest of making the development of the controller as practical as possible.

The majority of this phase has been and will continue to be dedicated to reviewing the body of literature available. We are particularly interested in identifying models that can be applied to rod-like continuous soft robots, but also models that possess at least one of the two following characteristics: 1) a model with some precedent of implementing a controller for it, or 2) a model whose underlying dynamic assumptions, kinematics, and/or parametrization has been shown to be robust. The models from [5] and [9] previously discussed in this proposal belong to each category, respectively. We preliminarily present a tabulated summary of our current findings in Table 3 below.

Through this literature review, the author hopes to gain a better view of what the prevailing dynamic models being used in the field are. The main goal of this phase is to identify *a* model whose feasibility regarding the design of a controller for it falls within the time scope of the proposed thesis. This proposal would like to stress that the main focus of the proposed thesis is *not* dynamical modelling of soft robots, but rather *controller design* for them. As such, we are more interested in designing around a "workable" model than identifying *the* perfect model.

Model	# of States	ODE?	Controller?
Augmented Rigid Body [5]	3	Yes, but only within certain boundaries	Yes
Arc-Length Parametrization [9]	3 (5*) *(parametric states)	Yes	Yes
Discrete Cosserat Approach [10]	6	No* *(solved via recursive algorithms)	No

Table 3: Characteristics and features of the dynamic models reviewed so far

4.2 Model Calibration

Since one of the goals of this proposed thesis is to validate the controller designed via hardware implementation, the chosen dynamic model will need to be tuned to the test bench platform we have at hand (see Figure 5). Practically speaking, this means evaluating what the various dynamical terms and matrices are for our soft robot.

From the construction and physical characteristics of the robot alone, some semblance of what the dynamical terms and matrices of the robot are can be extracted. This initial application of the model to our system can be considered the "un-tuned" model. It can then be tuned by motor-babbling (passing in a series of randomized inputs–"babbles") our soft robot, simulate motor-babbling the soft robot *using* the model, and then comparing the outputs between the two. Generally speaking, a "perfect" model would simulate an exact replica of the output that the robot itself physically produced. While the tuned model cannot do the same, it should simulate sufficiently accurate outputs when operating in configurations found within the boundaries established by the model's assumptions.

The main goal of this phase is to calibrate the dynamic model to such an extent that it can sufficiently represent how the system *responds* to an input signal. Since our soft robot is *pneumatically*-actuated, that means the input signal will be in terms of pressure change. Roughly speaking, we want to ensure that for a given pressure change, the model simulates a change in

Figure 5: The 3-D pneumatically-actuated "limb" serving as our test bench platform

configuration reflective of the change in configuration physically exhibited by the robot.

4.3 Controller Implementation

The goal of this phase is to arrive at a formulation for the controller, physically implement it on our test bench platform, and finally assess the controller's performance. The type of model determines how the controller ends up being formulated but assuming we end up moving forward with a PCC-based model, then the system can be considered as a Langragian system whose dynamics can be expressed in the standard form (see Equation 1).

$$\mathbf{B}(\underline{q})\underline{\ddot{q}} + \mathbf{C}(\underline{q}, \underline{\dot{q}})\underline{\dot{q}} + \mathbf{K}\underline{q} + \mathbf{D}\underline{\dot{q}} = \mathbf{A}(\underline{q})\underline{u} \tag{1}$$

Here, $\underline{q}$ is the configuration vector, $\mathbf{B}$ is the inertia matrix, $\mathbf{C}$ is the matrix collecting Coriolis and centrifugal forces, $\mathbf{K}$ is the elasticity matrix, $\mathbf{D}$ is the damping matrix, and $\mathbf{A}$ maps the input $\underline{u}$ into wrenches. In both [5] and [9] where this formulation was used, similar methodologies were used to subsequently arrive at the controller:

$$\begin{aligned}\underline{u} = \mathbf{A}^{-1}(&-\mathbf{B}\underline{\ddot{q}}_{ref} - \mathbf{C}\underline{\dot{q}}_{ref} - \mathbf{K}\underline{q}_{ref} \\ &- \mathbf{D}\underline{\dot{q}}_{ref} + \mathbf{K}_p(\underline{q}_{ref} - \underline{q}) - \mathbf{K}_D(\underline{\dot{q}}_{ref} - \underline{\dot{q}}))\end{aligned} \tag{2}$$

Our controller will be designed following the framework outlined in the works mentioned above, with the main functionality to be pursued being task-space regulation and then trajectory tracking. Then preliminary validation of the controller will be conducted via simulation before a more thorough experimentation is conducted on hardware.

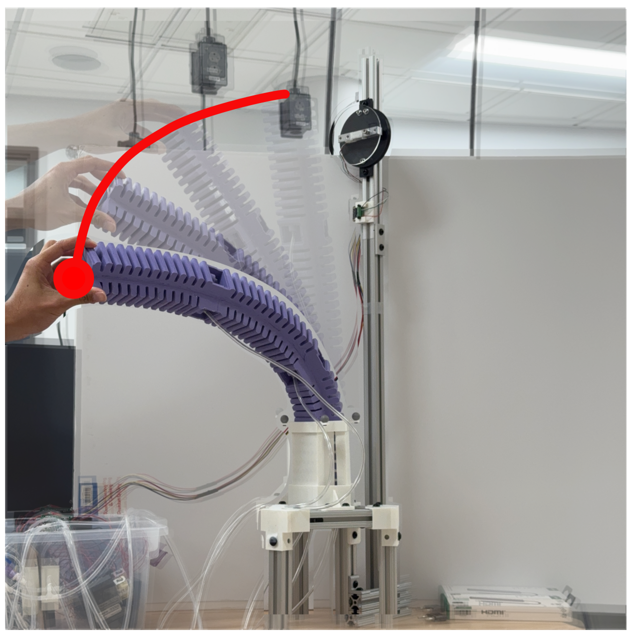
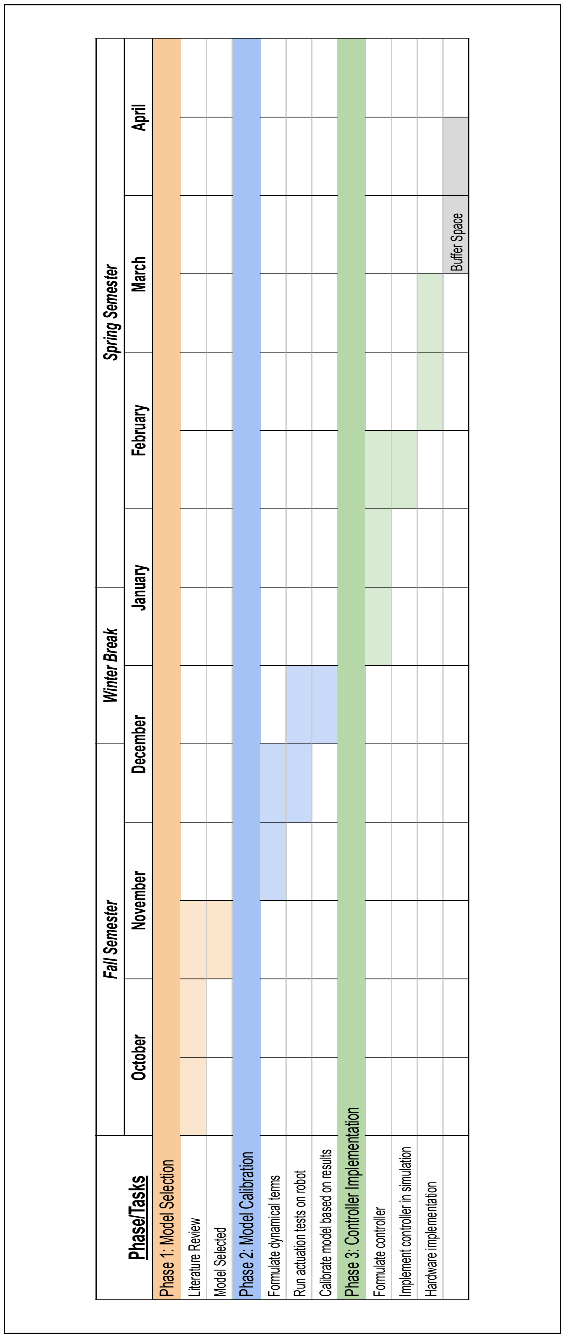


Figure 6: The soft robot stabilizing around a trajectory

The test bench platform to be experimented on consists of: our pneumatically actuated soft robot, an Arduino-based system that handles the pressure actuation, an integrated codebase that manages communication between the Python-based dynamics computation and the Arduino-based actuation system, and finally a three-dimensional motion capture system. Using this platform, we will be able to closely assess the controller's performance.

Our assessment methodology will revolve around the main functionalities being pursued. That is, we will assess the controller's performance based on its accuracy and stability. Practically speaking, this will involve building a dataset to serve as the reference configurations/trajectories, simulating what the system output looks like when using that dataset as the reference for our control input, running the control inputs from the dataset on our hardware, and finally comparing the results. Building the reference dataset will be done using the motion capture system: the soft robot's end effector can be pointed to various configurations or traced around different trajectories (like Figure 6), "manually". Similarly, the configurations and trajectories that end up being physically achieved by the robot via the control inputs will be recorded using the motion capture system. This way, we can compare the reference and actual configurations and trajectories on the same measurement/recording platform.

5 Proposed Timeline



6 References

[1] J. Rakhmatillaev, V. Bucinskas, and N. Kabulov, "An integrative review of control strategies in robotics", *Robotic Systems and Applications*, Jul. 10, 2025. DOI: 10.21595/rsa.2025.25014.

[2] 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. DOI: 10.1109/MCS.2023.3253419.

[3] L. Brunke, M. Greeff, A. W. Hall, *et al.*, "Safe learning in robotics: From learning-based control to safe reinforcement learning", *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 411–444, Volume 5, 2022 May 3, 2022. DOI: 10.1146/annurev-control-042920-020211.

[4] R. J. Webster III 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. 1, 2010. DOI: 10.1177/0278364910368147.

[5] 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. 1, 2020. DOI: 10.1177/0278364919897292.

[6] A. Dickson, J. C. P. Garcia, R. Jing, M. L. Anderson, and A. P. Sabelhaus, "Real-time trajectory generation for soft robot manipulators using differential flatness", in *2025 IEEE 8th International Conference on Soft Robotics (RoboSoft)*, Apr. 2025, pp. 1–7. DOI: 10.1109/RoboSoft63089.2025.11020810.

[7] 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. DOI: 10.1109/ROBOSOFT.2018.8404895.

[8] 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. DOI: 10.1109/ROBOSOFT.2019.8722799.

[9] 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, Apr. 2020. DOI: 10.1109/LRA.2020.2967269.

[10] 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. DOI: 10.1109/TRO.2018.2868815.

[11] A. K. Dickson, J. C. P. Garcia, M. L. Anderson, *et al.*, *Safe autonomous environmental contact for soft robots using control barrier functions*, Apr. 20, 2025. DOI: 10.48550/arXiv.2504.14755. arXiv: 2504.14755[cs].

[12] K. Wong, M. Stölzle, W. Xiao, C. D. Santina, D. Rus, and G. Zardini, *Contact-aware safety in soft robots using high-order control barrier and lyapunov functions*, May 5, 2025. DOI: 10.48550/arXiv.2505.03841. arXiv: 2505.03841[cs].

[13] C. D. Santina and D. Rus, "Control oriented modeling of soft robots: The polynomial curvature case", *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 290–298, Apr. 2020. DOI: 10.1109/LRA.2019.2955936.