Task-Space Control of Continuum Robots using Underactuated Discrete Rod Models

Caleb Rucker [1,5], Eric J. Barth [2,4], Joshua Gaston[1,3], and James C. Gallentine[2,3]

***Abstract*— Underactuation is a core challenge associated with controlling soft and continuum robots, which possess theoretically infinite degrees of freedom, but few actuators. However, m actuators may still be used to control a dynamic soft robot in an m-dimensional output task space. In this paper we develop a task-space control approach for planar continuum robots that is robust to modeling error and requires very little sensor information. The controller is based on a highly underactuated discrete rod mechanics model in maximal coordinates and does not require conversion to a classical robot dynamics model form. This promotes straightforward control design, implementation and efficiency. We perform input-output feedback linearization on this model, apply sliding mode control to increase robustness, and formulate an observer to estimate the full state from sparse output measurements. Simulation results show exact task-space reference tracking behavior can be achieved even in the presence of significant modeling error, inaccurate initial conditions, and output-only sensing.**

I. Introduction

Continuously flexible robotic structures (continuum robots and soft robots) have great potential as highly dynamic manipulators and locomotors that use their inherent elastic energy storage to operate efficiently, perform stable and safe contact tasks, and adapt to uncertain environments. However, this great potential comes with the challenge of controlling an infinitely high degree-of-freedom system with a very limited number of actuators and sparse sensing. In this paper we add to the recent discussion around control of highly underactuated soft robotic structures by developing an approach for task-space dynamic control using discrete rod models.

A. Dynamic Modeling

Dynamic models for simulation and control of soft and continuum robots can be broadly categorized by (1) how the geometry of the robot is represented, and (2) how the equations of motion are formulated and solved [1], [2]. On the one hand, many models represent the continuum dynamics on a set of *minimal coordinates*, such as segment curvatures [3] [4], actuator lengths [5], strains [6], [7], parameterized curves [8] or virtual rigid-link joint angles [9]. The equations are then formulated on the basis of a principle of mechanics, such as Lagrangian dynamics, and ultimately represented in a classical rigid-link robot dynamics form such as

$$\mathbf{D}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})\ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) + \mathbf{G}(\boldsymbol{\theta}) = \boldsymbol{\tau} \tag{1}$$

where $\boldsymbol{\theta} \in \mathbb{R}^N$ is a *minimal* set of generalized coordinates that describe the robot configuration, and $\boldsymbol{\tau} \in \mathbb{R}^N$ is a set of actuation inputs (i.e. the model is *fully actuated*). To arrive at such a model for a continuum or soft robot, one must often apply a very coarse approximation scheme, e.g. by assuming constant curvature over an entire actuated segment. The advantages of such coarsely approximated models are 1) computational efficiency (for larger N, the equations in (1) are difficult to form analytically and expensive to solve) and 2) straightforward adaptation of conventional controller design approaches for fully actuated rigid-link robots onto continuum robots [9]. The main drawback of this strategy is that it ignores the underactuated dynamics that cause variations in curvature across actuated segments. These underactuated dynamics may be important to account for in modeling and control if one wishes to increase control bandwidth and more fully exploit the dynamic potential of soft/continuum robots.

In contrast, another category of continuum robot modeling first expresses the dynamics in a continuous form, described as a set of nonlinear partial differential equations in arc length and time [10], [1], based on the dynamic theories of Cosserat rods, Kirchhoff rods (unshearable, inextensible), or planar elastica. The solution to the PDEs is then approximated with various numerical methods. Models of this type are typically highly resolved in space (i.e. they are *underactuated* - there are many more model DOF than actuators), which allows more accurate dynamic simulation, and they typically describe the time evolution of a redundant set of variables (larger than the degrees of freedom) satisfying certain constraints of compatibility, i.e. they employ a *maximal coordinates* approach. (Certain choices for numerical discretization of the PDEs are equivalent to pseudo-rigid-body models and so-called discrete elastic rod models [11], [12] as we show later). A prevalent viewpoint in continuum and soft robotics research is that such highly resolved, maximal coordinate models are useful for simulation, but difficult to exploit for control (e.g. [7]) because 1) they are *underactuated* (which complicates control) and 2) they not expressed in the classic *minimal* form (1). In

[1]The University of Tennessee, Knoxville, Department of Mechanical, Aerospace, and Biomedical Engineering

[2]Vanderbilt University, Nashville, Department of Mechanical Engineering

[3]IEEE Student Member

[4]IEEE Member

[5]IEEE Senior Member

Parts of this work are based upon work supported by the National Science Foundation under EFMA-1935278 and under NSF CAREER Award IIS-1652588. Any opinion, findings, and conclusions or recommendations expressed herein are those of the authors and do not necessarily reflect the views of the National Science Foundation.

this paper, we would like to promote the contrary view: underactuated maximal coordinate models obtained from discrete mechanics models or discretization of a continuous PDE *are* suitable for control design and efficient control implementation, particularly in task space.

B. Control

The majority of work on dynamic control of continuum manipulators assumes a fully actuated model (e.g [13], [14], [9]) in minimal coordinates such as piecewise constant curvatures. Some recent work has considered control with underactuated models using parametric curves [8], or reduced order models generated from general 3D finite element simulations [15], [3], [16], while early continuum robot control work acknowledged and modeled the underactuated dynamics [17], formulating linear boundary control laws that sought to stabilize the system and reduce vibration. Perhaps the most closely related work to this paper is [18] which recently showed that m actuators are sufficient to control an underactuated soft robot in an m degree of freedom task space. This was done by leveraging the operational space dynamics formulation and projecting the required actuation torques into a lower dimensional space using muscle synergies.

We believe the input-output feedback linearization part of our controller might be numerically equivalent to the operational-space control in [18]. However, the formulation herein may facilitate a more straightforward implementation and efficient computation, since our constrained Lagrangian model (or equivalently the discretized PDE model) never needs to be explicitly expressed in either minimal coordinates or operational space, nor does it need to consider muscle synergies. In addition, robustness and state estimation were left to future work in [18]. Herein we contribute toward those goals by proposing a sliding mode outer-loop controller and a passive observer design. We additionally include simulations with parametric modeling error and limited sensing.

C. Contributions

In the following sections we present a control design for soft robots based on constrained Lagrangian model for discrete elastic rods in maximal of coordinates. This approach allows the model to be easily constructed and implemented programmatically for large numbers of links, and we show that it is equivalent to a first-principles PDE-based approach. The approach also exhibits a high degree of efficiency for fast, computationally-lean solving. We also present an input-output feedback linearization control design methodology that is augmented by a sliding mode outer loop that provides robustness to parametric uncertainty. To provide the full state estimate needed for the controller we propose an observer to reconstruct the unmeasured states of the model from very few measurements by exploiting the stable zero dynamics (or internal dynamics) of the system. Finally, we verify the entire approach in simulation using the parameters of a 2-DOF soft robot prototype. The approach is demonstrated to be robust by achieving exact task space tracking even in the presence of non-negligible modeling error and output-only sensing.

II. Dynamic Model

A. PDE Perspective

The dynamics of a classical Kirchhoff rod in 2D are described by the following set of nonlinear partial differential equations [19]:

$$\begin{aligned} \mathbf{p}' &= \begin{bmatrix}\cos\theta \\ \sin\theta\end{bmatrix} \\ \theta' &= u \\ \mathbf{n}' + \mathbf{f} &= \rho A \ddot{\mathbf{p}} \\ \zeta' + \mathbf{p}' \times \mathbf{n} + \ell &= \rho I \ddot{\theta} \end{aligned} \tag{2}$$

where $'$ denotes a derivative with respect to arc length, s, $\dot{}$ denotes a derivative with respect to time, t, $\mathbf{p}(s,t) \in \mathbb{R}^2$ is the position along the rod, $\theta(s,t) \in \mathbb{R}$ is the tangent angle measured counterclockwise from the global x axis, $u(s,t) \in \mathbb{R}$ is the curvature, $\mathbf{n}(s,t) \in \mathbb{R}^2$ is the internal force, $\zeta \in \mathbb{R}$ is the internal moment about z (out of the plane, counterclockwise positive), ρ is the material density, A is the cross sectional area, I is the second area moment of the cross section, $\mathbf{f}(s,t) \in \mathbb{R}^2$ is an external distributed force, and $\ell(s,t) \in \mathbb{R}$ is an external distributed moment.

The equations in (2) are incomplete until a constitutive law is formulated to relate u and ζ.

The following viscoelastic model is commonly assumed:

$$\zeta = EIu + BI\dot{u} \tag{3}$$

where E is Young's modulus, I is the second area moment of the cross section, and B is a damping modulus. Substituting this into (2), we get a complete PDE set in terms of state variables $\mathbf{p}$, θ, and $\mathbf{n}$:

$$\begin{aligned} \mathbf{p}' &= \begin{bmatrix}\cos\theta \\ \sin\theta\end{bmatrix} \\ \mathbf{n}' + \mathbf{f} &= \rho A \ddot{\mathbf{p}} \\ EI\theta'' + BI\dot{\theta}'' + \begin{bmatrix}-\sin\theta \\ \cos\theta\end{bmatrix}^\top \mathbf{n} + \ell &= \rho I \ddot{\theta} \end{aligned} \tag{4}$$

An intuitive and symmetric discretization strategy for these PDE's is to assume a set of $N+1$ ordered equally-spaced nodes in arc length, $\{s_0\ s_1\ s_2\ \dots\ s_N\}$ (over the total length L) where positions $\mathbf{p}_i$ and moments ζ_i naturally live at each node, while forces $\mathbf{n}_i$ and angles θ_i naturally live on the N edges (or links) halfway between nodes $i-1$ and i as shown in Figure 1 (left). Assuming a constant step size $h = s_i - s_{i-1}$, centered approximations for the first and second derivatives in arc length can then be applied, resulting in

$$\begin{aligned} (\mathbf{p}_i - \mathbf{p}_{i-1})/h &= \begin{bmatrix}\cos\theta_i \\ \sin\theta_i\end{bmatrix} \\ (\mathbf{n}_{i+1} - \mathbf{n}_i)/h + \mathbf{f}(s_i) &= \rho A \ddot{\mathbf{p}}_i \\ EI(\theta_{i-1} - 2\theta_i + \theta_{i+1})/h^2 + & \\ BI(\dot{\theta}_{i-1} - 2\dot{\theta}_i + \dot{\theta}_{i+1})/h^2 + & \\ \begin{bmatrix}-\sin\theta_i \\ \cos\theta_i\end{bmatrix}^\top \mathbf{n}_i + \ell(s_i) &= \rho I \ddot{\theta}_i \end{aligned} \tag{5}$$

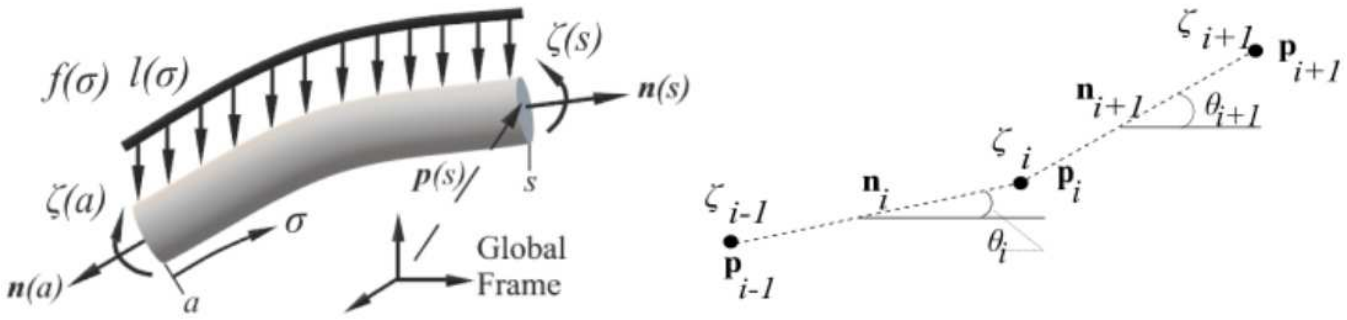

Fig. 1. An arbitrary section of a rod from a to s along σ is shown on the left. The distributed forces and moments are represented as f and l respectively. The internal forces and moments are represented as $\mathbf{n}$ and ζ respectively. On the right, the internal forces $\mathbf{n}$ and moments ζ are indicated on the links and nodes on ether side of the i^{th} node of a set of discrete links. The dots represent the nodes at position $\mathbf{p}$ and the dashed lines represent the "links" of the system. The tangent angle of each link is indicated by θ.

The system in (5) is a set of differential algebraic equations with index 3 (the number of differentiations required to obtain differential equations for all state variables). There are no derivatives for the $\mathbf{n}_i$ variables, and the first equation in the list is an algebraic constraint. One can reduce the index by writing the derivatives of the constraint equations:

$$\begin{aligned}
\mathbf{p}_i - \mathbf{p}_{i-1} &= h \begin{bmatrix} \cos\theta_i \\ \sin\theta_i \end{bmatrix} \\
\dot{\mathbf{p}}_i - \dot{\mathbf{p}}_{i-1} &= h \begin{bmatrix} -\sin\theta_i \\ \cos\theta_i \end{bmatrix} \dot{\theta}_i \\
\ddot{\mathbf{p}}_i - \ddot{\mathbf{p}}_{i-1} &= h \begin{bmatrix} -\cos\theta_i \\ -\sin\theta_i \end{bmatrix} \dot{\theta}_i^2 + h \begin{bmatrix} -\sin\theta_i \\ \cos\theta_i \end{bmatrix} \ddot{\theta}_i
\end{aligned} \tag{6}$$

Then enforcing the second derivative of the constraints along with the original differential equations in the DAE set, one obtains an index-1 DAE system which can be written as:

$$\begin{aligned}
\ddot{\mathbf{p}}_{i-1} - \ddot{\mathbf{p}}_i + h \begin{bmatrix} -\sin\theta_i \\ \cos\theta_i \end{bmatrix} \ddot{\theta}_i &= h \begin{bmatrix} \cos\theta_i \\ \sin\theta_i \end{bmatrix} \dot{\theta}_i^2 \\
\mathbf{n}_{i+1} - \mathbf{n}_i + \mathbf{f}_i &= m\ddot{\mathbf{p}}_i \\
k(\theta_{i-1} - 2\theta_i + \theta_{i+1}) \; + & \\
b(\dot{\theta}_{i-1} - 2\dot{\theta}_i + \dot{\theta}_{i+1}) \; + & \\
h \begin{bmatrix} -\sin\theta_i \\ \cos\theta_i \end{bmatrix}^\top \mathbf{n}_i + \ell_i &= J\ddot{\theta}_i
\end{aligned} \tag{7}$$

where $m = h\rho A$ is the mass associated with each node, $J = h\rho I$ is the density weighted second area moment integrated over the discrete length h, $k = EI/h$ and $b = BI/h$ are equivalent rotational spring and damping constants associated with the joints, $\mathbf{f}_i = h\mathbf{f}(s_i)$ is the effective external force applied at each node, and $\ell_i = h\ell(s_i)$ is the effective external moment applied at each link.

The equations in (7) are applicable for $i = 1...N-1$. If the robot has a fixed base at $s = 0$ we may take $\mathbf{p}_0$, $\dot{\mathbf{p}}_0$, $\ddot{\mathbf{p}}_0$, θ_0, and $\dot{\theta}_0$ to be prescribed values. If the robot has a free distal end, we may prescribe

$$\begin{aligned}
-\mathbf{n}_N + \mathbf{f}_N &= \frac{m}{2}\ddot{\mathbf{p}}_N \\
k(\theta_{N-1} - \theta_N) \; + b(\dot{\theta}_{N-1} - \dot{\theta}_N) \; + & \\
h \begin{bmatrix} -\sin\theta_N \\ \cos\theta_N \end{bmatrix}^\top \mathbf{n}_N + \ell_N &= J\ddot{\theta}_N,
\end{aligned} \tag{8}$$

since there is no material beyond s_n to provide mass or exert internal force or moment on the proximal material.

The discretized and index-reduced PDE is essentially equivalent to an ODE in the positions and angles because the system can be directly solved for their second derivatives by eliminating the internal forces. Since the reduced index system only enforces the second derivative of the constraints, direct integration of all the coordinates in time can result in constraint drift during numerical simulation. There are numerous strategies available to reduce such drift (such as Baumgarte stabilization), but these are unnecessary for this model. The $\mathbf{p}_i$ and $\dot{\mathbf{p}}_i$ which satisfy the constraints can be determined as a function of the θ_i and $\dot{\theta}_i$ recursively from (6). Thus, for simulation purposes, we may use $\theta_1 \; \ldots \; \theta_N$ and $\dot{\theta}_1 \; \ldots \; \dot{\theta}_N$ as a minimal set of state variables to be integrated in time, and simply extract $\ddot{\theta}_1 \; \ldots \; \ddot{\theta}_N$ from the full solution of the maximal coordinate model (7). This strategy combines the advantages of maximal coordinates (ease of expression, programming, and computational efficiency) with the advantages of minimal coordinates (impossibility of constraint drift). We aim to show in this paper that such maximal coordinate models are also convenient for formulating dynamic controllers directly in task space.

B. Constrained Lagrangian Perspective

The above dynamic model formulated from the original Kirchhoff rod PDEs is equivalent to a multibody dynamics model that can be generated using either a Newton-Euler approach or a constrained Lagrangian approach. Writing down the equivalent constrained Lagrangian model will help us express the equations in a compact well-known form that will facilitate straightforward programming, efficient numerical solution, and controller design.

Consider a dynamic system in $\mathbf{q}$ subject to constraints $\boldsymbol{\phi}(\boldsymbol{q}) = 0$. Its associated Lagrangian is

$$L(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\lambda}) = T - V + \boldsymbol{\phi}(\mathbf{q})^\top \boldsymbol{\lambda}$$

where T is kinetic energy, and V is potential energy, and $\boldsymbol{\lambda}$ are the Lagrange multipliers. The equations of motion are then

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial L}{\partial \mathbf{q}} + \mathbf{A}^\top \boldsymbol{\lambda} = \mathbf{Q}$$

and

$$\boldsymbol{\phi}(\boldsymbol{q}) = 0$$

where $\mathbf{A} = \frac{\partial \boldsymbol{\phi}(\mathbf{q})}{\partial \mathbf{q}}$, and $\mathbf{Q}$ collects any non-conservative generalized forces that do not arise from the constraints. This forms an index-3 DAE of a form equivalent to (5). Taking the first and second derivatives of the constraint equations, we get

$$\mathbf{A}\dot{\mathbf{q}} = 0, \qquad \mathbf{A}\ddot{\mathbf{q}} = \boldsymbol{\gamma}$$

where $\boldsymbol{\gamma} = -\dot{\mathbf{A}}\dot{\mathbf{q}}$. Then the equivalent index-1 dynamic system can be written as

$$\begin{bmatrix} \mathbf{M} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \boldsymbol{\gamma} \end{bmatrix} \tag{9}$$

where $\mathbf{F}$ contains all the unconstrained generalized forces. This helps realize and organize the structure of the large system formed by (7). By inspection, we can see that the equations in (7) have exactly this form. The generalized coordinates are $\mathbf{q} = \begin{bmatrix} \mathbf{p}_1^\top & \theta_1 & ... & \mathbf{p}_N^\top & \theta_N \end{bmatrix}^\top$. The Lagrangian corresponding to our spatially discretized PDE system is

$$\begin{aligned} L = & \frac{1}{2}\sum_{i=1}^{N-1} m\dot{\mathbf{p}}_i^\top \dot{\mathbf{p}}_i + \frac{m}{4}\dot{\mathbf{p}}_N^\top \dot{\mathbf{p}}_N + \frac{1}{2}\sum_{i=1}^{N} J\dot{\theta}_i^2 \\ & -\frac{1}{2}\sum_{i=1}^{N} k\left(\theta_i - \theta_{i-1}\right)^2 + \boldsymbol{\phi}(\mathbf{q})^\top \boldsymbol{\lambda}. \end{aligned} \tag{10}$$

The Lagrange multiplier vector $\boldsymbol{\lambda} = \begin{bmatrix} \mathbf{n}_1^\top & \dots & \mathbf{n}_N^\top \end{bmatrix}^\top$ is the set of internal link forces, and the constraint equations $\boldsymbol{\phi}(\mathbf{q}) = \begin{bmatrix} \boldsymbol{\phi}_1^\top & \dots & \boldsymbol{\phi}_N^\top \end{bmatrix}^\top = \mathbf{0}$ are given by

$$\boldsymbol{\phi}_i = h\begin{bmatrix} \cos\theta_i \\ \sin\theta_i \end{bmatrix} - \mathbf{p}_i + \mathbf{p}_{i-1}$$

for $i = 1...n$. The generalized forces $\mathbf{Q}$ consist of damping moments and externally applied forces and moments. These terms manifest in $\mathbf{F}$ as detailed below.

The $n \times n$ matrix $\mathbf{M}$ is diagonal and given by the Euler-Lagrange equation as

$$\mathbf{M} = \text{diag}(m, m, J, m, m, J, ..., m/2, m/2, J)$$

The matrix $\mathbf{A}$ is sparse and defined by the constraint equations. It can be built row by row by specifying only the nonzero entries. The submatrix of $\mathbf{A}$ consisting of rows $2i-1$ to $2i$ and columns $3i-5$ to $3i$ is given as:

$$\begin{bmatrix} 1 & 0 & 0 & -1 & 0 & -h\sin\theta_i \\ 0 & 1 & 0 & 0 & -1 & h\cos\theta_i \end{bmatrix}$$

for $i = 2...n$, and the corresponding rows $2i-1$ to $2i$ of the vector $\boldsymbol{\gamma}$ are

$$h\dot{\theta}_i^2 \begin{bmatrix} \cos\theta_i \\ \sin\theta_i \end{bmatrix}.$$

If the rod is cantilevered with a prescribed $\mathbf{p}_0(t)$, this implies a prescribed $\ddot{\mathbf{p}}_0(t)$, and the upper left 2×3 of $\mathbf{A}$ is

$$\begin{bmatrix} -1 & 0 & -h\sin\theta_1 \\ 0 & -1 & h\cos\theta_1 \end{bmatrix}$$

while the first 2 rows of $\boldsymbol{\gamma}$ are

$$h\dot{\theta}_1^2 \begin{bmatrix} \cos\theta_1 \\ \sin\theta_1 \end{bmatrix} - \ddot{\mathbf{p}}_0$$

Finally, rows $3i-2$ to $3i$ of the vector $\mathbf{F}$ are specified as

$$\begin{bmatrix} \mathbf{f}_i \\ k(\theta_{i-1} - 2\theta_i + \theta_{i+1}) + b(\dot{\theta}_{i-1} - 2\dot{\theta}_i + \dot{\theta}_{i+1}) + \ell_i \end{bmatrix}$$

for $i = 1...n-1$ (letting θ_0 have a prescribed value), while the last two rows of $\mathbf{F}$ corresponding to the distal end are

$$\begin{bmatrix} \mathbf{f}_N \\ k(\theta_{N-1} - \theta_N) + b(\dot{\theta}_{N-1} - \dot{\theta}_N) + \ell_i \end{bmatrix}.$$

This completes all the blocks defining the constrained Lagrangian form of the model in (9), consistent with the discretized Kirchhoff rod PDE in (5). The linear system (9) is sparse with a structure that can be efficiently solved in $O(n)$ time using standard linear solvers, such as MATLAB's mldivide( ). Without optimization the link accelerations are computed on the order of 1 millisecond for $N = 40$ in MATLAB. As discussed before, for simulation purposes, the link angle accelerations are extracted from this solution, integrated in time, and used to recursively compute the positions and velocities. This avoids the constraint drift problem typically associated with maximal coordinate formulations.

C. Robot Actuation

It is straightforward to include typical soft robot actuation methods into the model above. In a tendon actuated robot, the k^{th} tendon exerts tension T_k at the point were it is distally attached (suppose at node i), offset from the elastic center of stiffness by some distance r_k. This creates a torque $\tau_k = T_k r_k$ applied as an external moment ℓ_i applied at node i, and added into the appropriate row of $\mathbf{F}$. As discussed in [10], the tendon also creates an applied force in the tangential direction at the attachment point, as well as a distributed load along the length, always orthogonal to the tangent and proportional to the curvature. However, *in the planar case*, the combined effect of these point and distributed tendon forces only influences the local axial component of internal force. They do not affect the local transverse component of internal force or the internal moment along the length, and thus do not affect curvature or dynamic shape. Thus, it is common (e.g in [13]) to treat the actuation as consisting only of a point moment at the attachment point for simplicity, while understanding that the true internal force is simply the model predicted internal force minus the collective tendon tension in the axial direction. We note that in 3D, this simplification no longer holds [10], but that fact does not affect the applicability of the approach in this paper.

In a planar fluid-powered soft robot, a pressurized chamber exerts the same set of forces and moments as a tendon, but in the opposite direction. (the two models are slightly different in 3D, however, as detailed in [1].) Thus, for either a planar fluid-powered robot or a planar tendon-actuated robot, we can model the actuation of a pressurized chamber or a tensioned tendon as a point moment τ_k applied at its distal termination point along the robot, as is commonly done.

D. Full Model for Control

Considering then an arbitrary set of m actuation torques $\mathbf{u} \in \mathbb{R}^m$, we find that the full model for control takes the form

$$\begin{bmatrix} \mathbf{M} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \boldsymbol{\gamma} \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \mathbf{u} \tag{11}$$

where $\mathbf{B} \in \mathbb{R}^{3N \times m}$ is a sparse matrix with a one placed in element $(3i, k)$ if actuator k applies a torque at node i, and $\mathbf{u} \in \mathbb{R}^m$ is the vector of actuator torque inputs. We note that it is possible to eliminate $\boldsymbol{\lambda}$ from (11) and arrange it in a form similar to (1), with a different right hand side due to the underactuation. However, the resulting analytical expressions are highly inconvenient for large N, and task-space controller design is easily facilitated in the form (11) as we show in the next section.

III. Controller Design

In this section we formulate an output control design based on the model of (11). Consider that we want to control an output $\mathbf{y} \in \mathbb{R}^m$ which is some linear function of $\mathbf{q}$, such that $\mathbf{y} = \mathbf{C}\mathbf{q}$ for some matrix $\mathbf{C} \in \mathbb{R}^{m \times 3N}$. Task space outputs are easily described in this way because the pose at the end-effector (or any other point) is included in $\mathbf{q}$.

A. Input-Output Feedback Linearization

Using the Lagrangian dynamics form, the second derivative of the output can then be expressed as a linear function of the actuation torques as follows:

$$\ddot{\mathbf{y}} = [\mathbf{C} \;\; \mathbf{0}] \begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix} = \ddot{\mathbf{y}}_0 + \mathbf{J}\mathbf{u}$$

where

$$\ddot{\mathbf{y}}_0 = [\mathbf{C} \;\; \mathbf{0}] \begin{bmatrix} \mathbf{M} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{F} \\ \boldsymbol{\gamma} \end{bmatrix}$$

is the output acceleration that would occur in the absence of any actuation input, and

$$\mathbf{J} = [\mathbf{C} \;\; \mathbf{0}] \begin{bmatrix} \mathbf{M} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}$$

is a $m \times m$ dynamic Jacobian matrix relating actuator torque to output acceleration. Note that computation of $\dot{\mathbf{y}}_0$ and $\mathbf{J}$ does not require inversion of the large system matrix, but can be done with a much less expensive linear solve. The term

$$[\mathbf{C} \;\; \mathbf{0}] \begin{bmatrix} \mathbf{M} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix}^{-1}$$

can be efficiently computed with MATLAB's right matrix divide, or similar function, and is merely solving m sparse linear systems. Feedback linearization of the output is then accomplished by the inner loop control law:

$$\mathbf{u} = \mathbf{J}^{-1}(\mathbf{a} - \ddot{\mathbf{y}}_0) \tag{12}$$

where $\mathbf{a}$ is symbolic of the desired output acceleration and serves as the new control input that will be specified by an outer loop control policy. Thus the feedback linearization transforms the nonlinear input-output dynamics into a set of decoupled double integrators.

B. Inertial Coupling and Singularities

Note that such feedback linearization can be done only if there is *strong inertial coupling* [20] between the actuation and the output. This corresponds to $\mathbf{J}$ being nominally nonsingular. Strong inertial coupling in this model arises from the inextensibility constraints of the Kirchhoff rod PDE, although $\mathbf{J}$ may become singular in certain configurations due to the robot's kinematic singularities. If $\mathbf{J}$ does loose rank in a particular configuration, we can replace the inverse in (13) with the Moore-Penrose pseudo-inverse. More generally, we can use a damped pseudo-inverse that will prevent large commanded actuation torques and guard against ill-conditioning around any singularities:

$$\mathbf{u} = \left(\mathbf{J}^\top \mathbf{J} + \beta \mathbf{I}\right)^{-1} \mathbf{J}^\top (\mathbf{a} - \ddot{\mathbf{y}}_0) \tag{13}$$

where β is a small damping parameter and can be chosen to balance tracking accuracy against mitigation of singularities.

C. Linear outer-loop control

If $\mathbf{r}(t)$ is the reference we desire the output to track, then a outer-loop linear control law with feedforward acceleration:

$$\mathbf{a} = \ddot{\mathbf{r}} - k_d \dot{\mathbf{e}} - k_p \mathbf{e} \tag{14}$$

will drive the output error $\mathbf{e} = \mathbf{h} - \mathbf{r}$ asymptotically to zero, with second order behavior according to our choice of PD gains k_p and k_d. However, classical feedback linearization, when combined with such a linear outer control loop, is known to suffer from poor robustness to model parameter uncertainty [21]. Our preliminary simulations showed that with this approach, even very small changes to the modelled Young's modulus caused the controlled continuum robot system to become unstable.

D. Sliding Mode Control

A sliding mode control approach can greatly improve robustness while preserving the basic structure and advantages of the input-output feedback linearization above. We define a sliding variable $\mathbf{s}$ in terms of the output error as follows:

$$\mathbf{s} = \dot{\mathbf{e}} + \alpha \mathbf{e}$$

for $\alpha > 0$. Thus, driving $\mathbf{s}$ to zero eventually enforces stable first-order output error dynamics with time constant $1/\alpha$. To derive a control which robustly drives $\mathbf{s}$ to zero, consider the Lyapunov function

$$V = \frac{1}{2} \mathbf{s}^\top \mathbf{s}$$

Its derivative

$$V = \mathbf{s}^\top \dot{\mathbf{s}}$$

is forced to be negative definite (even under some finite distrubance or parametric error in the model) if we can choose the control input to achieve $\dot{\mathbf{s}} = -k \operatorname{sgn} \mathbf{s}$, where k is a gain related to the expected amount of disturbance, and sgn takes the element-wise sign of a vector. So sliding mode control should enforce

$$\dot{\mathbf{s}} = \ddot{\mathbf{y}} - \ddot{\mathbf{r}} + \alpha \dot{\mathbf{e}} = -k \operatorname{sgn} (\dot{\mathbf{e}} + \alpha \mathbf{e})$$

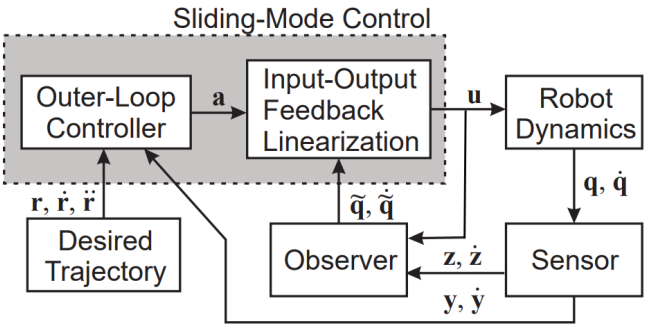

Fig. 2. The overall structure of the control system.

Since we can dictate $\ddot{\mathbf{y}} = \mathbf{a}$ using the input-output feedback linearization above, the sliding mode controller is then equivalent to replacing the linear outer loop control (14) with the nonlinear control law

$$\mathbf{a} = \ddot{\mathbf{r}} - \alpha\dot{\mathbf{e}} - k\,\mathrm{sgn}\,(\dot{\mathbf{e}} + \alpha\mathbf{e}) \tag{15}$$

The well-studied "chattering" phenomenon sometimes caused by the discontinuity in the $\mathrm{sgn}\,(\dot{\mathbf{e}} + \alpha\mathbf{e})$ function can be eliminated by various means (conventionally deadband, boundary layer, etc.). In our implementation we simply replace sgn with a continuous approximation, such as the Gauss error function $\mathrm{erf}\,(c(\dot{\mathbf{e}} + \alpha\mathbf{e}))$ where c encodes the steepness of the transition from -1 to 1.

E. Stable Zero Dynamics

Since the open loop system is passive, consisting of only of inertial, capacitive, and dissipative elements, the system's zero dynamics are stable. The transient oscillations of the zero dynamics will decay at their natural rate, and any steady-state oscillations will simply consist of the passive system's response to the frequency content of the reference trajectory, and the controller additionally stabilizes the output error. A full analysis of closed loop stability is outside the scope of this paper, but Theorem 1 in [20] (relevant to input-output feedback linearization with a linear outer loop controller) gives us hope that conditions for stability may be derived for our modified sliding mode approach in future work.

F. Force-Based Observer

The inner-loop feedback linearization described above requires knowledge of the full robot state $\mathbf{q}$ and $\dot{\mathbf{q}}$. While high-resolution shape sensors can provide this information, typical low-cost continuum robots have much more limited sparse sensing. Suppose that we can only sense the position $\mathbf{p}$ and velocity $\dot{\mathbf{p}}$ at a subset of the model nodes (say k of them), and designate the set of sensed positions as $\mathbf{z} = [\ldots \mathbf{p}_i^\top \ldots]^\top \in \mathbb{R}^{2k}$ for all nodes i where measurements are available. A conceptually intuitive observer design simply augments a forward dynamics simulation with external forces that are controlled to drive the estimated positions to their measured values. Thus the observer takes the form

$$\begin{bmatrix} \mathbf{M} & \mathbf{A}(\tilde{\mathbf{q}})^\top \\ \mathbf{A}(\tilde{\mathbf{q}}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\tilde{\mathbf{q}}} \\ \tilde{\boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_0(\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}) + \mathbf{B}\mathbf{u} + \mathbf{g} \\ \boldsymbol{\gamma}(\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}) \end{bmatrix} \tag{16}$$

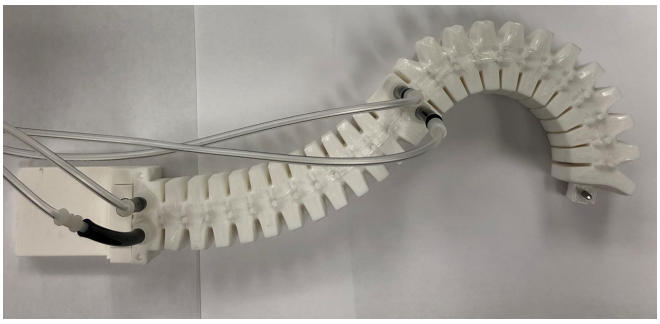

Fig. 3. Prototype pneumatic 2-DOF soft robot. Configuration shown with negative bending at the distal end and positive bending at the proximal end.

TABLE I
SIMULATED MODEL PARAMETERS

EI (Nm^2)	BI (Nm^{2}s)	ρA (kg/m)	ρI (kg m)	L (m)	N
0.14	0.003	0.8	0	0.275	20

where $\mathbf{g} \in \mathbb{R}^{3n}$ is a sparse vector containing a force $\mathbf{g}_i \in \mathbb{R}^2$ in elements $(3i{-}2 : 3i{-}1)$ whenever $\mathbf{p}_i$ and $\dot{\mathbf{p}}_i$ are measured by a sensor:

$$\mathbf{g}_i = L_d(\dot{\mathbf{p}}_i - \dot{\tilde{\mathbf{p}}}_i) + L_p\,(\mathbf{p}_i - \tilde{\mathbf{p}}_i)$$

with positive observer gains L_p, and L_d. This simple law cannot destabilize the observer system because it is passive, implementing a virtual spring and damper tied between the measured and estimated positions. Since the zero dynamics are stable, the estimated state will converge to the true state over time in the absence of model error. In the presence of modeling error, the stable zero dynamics guarantee a stable observer.

Figure 2 illustrates the resulting overall structure of the control approach. The sliding mode control consists of the nonlinear outer loop controller (15) with the input-output feedback linearization (13). The feedback linearization relies on the full state estimate from the observer (16) based on all of the sensed data $\mathbf{z}$ while the outer loop controller relies only on sensed output $\mathbf{y}$

IV. SIMULATION OF SOFT ROBOT CONTROL

We tested our nonlinear controller/observer combination numerically in simulation using a model for a pneumatic soft robot prototype shown in Figure 3. The prototype is 3D printed with 85A shore hardness Ninjaflex TPU (Ninjatek) filament using a Flexion Extruder (Diabase Engineering) mounted on a Lulzbot TAZ6. It has embedded pneumatic chambers designed to give the robot two actuated degrees of freedom in the plane. In our model, the actuation manifests as a torque τ_1 applied at the distal tip, and another τ_2 applied at the midpoint. In preliminary testing with the

TABLE II
CONTROLLER PARAMETERS

β	α	k	c	L_p	L_d
0.1	40	80	40	0.5	0.1

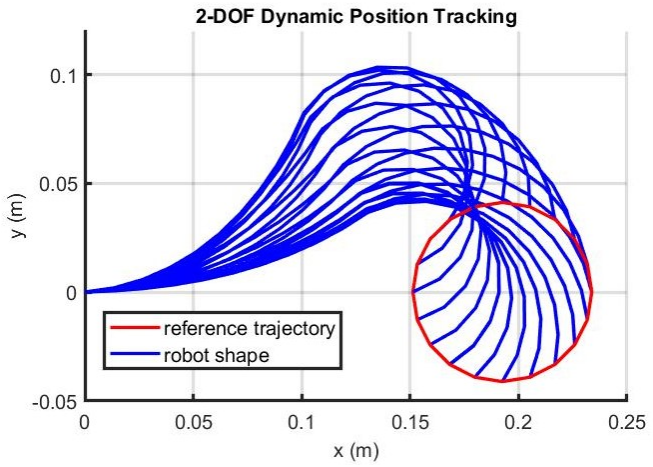

Fig. 4. Snapshots of the dynamic robot behavior during the circle task after the tracking error has converged to zero (after 0.2 seconds).

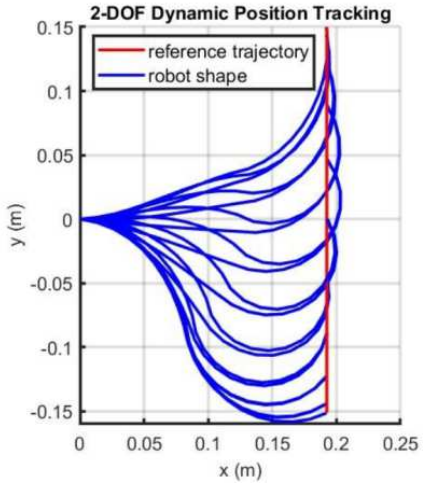

Fig. 5. Snapshots of the dynamic robot behavior during the line following task after the tracking error has converged to zero (after 0.2 seconds).

prototype, we identified appropriate physical constants and model parameters. These are listed in Table I. Note that while ρI is not really zero, it is small for most slender robots, and the model can be calibrated accurately enough assuming $\rho I = 0$. (Note that this causes no problem in the model. Even though $\mathbf{M}$ is singular, the larger system matrix in (11) is still full rank).

We implemented our controller/observer approach in Simulink, and the parameters used in the outer-loop controller and observer are listed in Table II. While the feedback linearization block and the observer assumed the model parameters in Table I, the simulated robot plant model used different parameters to simulate parameteric modeling error and assess controller robustness. The simulated robot had a 10% greater density and 10% lower stiffness and damping constants than the controller and observer assumed. The only output measurement used in the controller and observer was the tip position and velocity (i.e. $\mathbf{z} = \mathbf{y} = \mathbf{p}_N$), representing a very minimal sensing scenario, and the observer initial conditions were set differently than the robot model. Thus these numerical experiments simulate a practical scenario with limited sensor information, unknown initial conditions, and non-negligible model inaccuracy.

We performed two reference tracking experiments where the robot is commanded to follow a tip trajectory in space. The reference trajectories were designed to be dynamically significant by containing frequency content in the range of the first two modal frequencies of the underactuated robot dynamics (approx. 3.5 Hz and 19 Hz). The first trajectory of tracking a point moving around the red circle shown in Figure 4 at a frequency of 5 Hz. The second task is to follow a point moving sinusoidally back and forth along the red line (again at 5 Hz) as shown in Figure 5.

A. Results

Figures 4 and 5 show the dynamic robot behavior executing each task once the tracking error has converged to zero (i.e. after about 0.2 seconds). Figures 6 and 7 detail the error

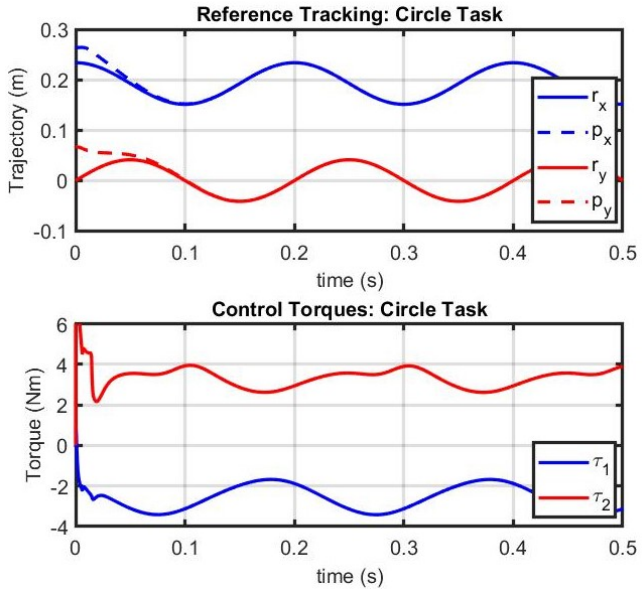

Fig. 6. Top: The desired reference trajectory is compared to the robot output for the circle tracking task. Bottom: The control torques indicate that significant inertial dynamics are being overcome.

convergence behavior and the actuator torques commanded by the sliding mode controller. The state estimation error does not converge to zero because of the model differences between the observer and the robot, but the reference tracking error goes to zero in spite of this due to the robustness of the sliding mode controller.

The snapshots in Figure 5 are timed to show the same task-space points on subsequent passes moving left and right. The fact that there are several pairs of snapshots with the same end-effector location but different body deformations indicates the significance of the inertial dynamics in this task. The presence of non-sinusoidal torque signals in Figures 6 and 7 also indicate that significant inertial dynamics and couplings between the actuated DOFs are being overcome.

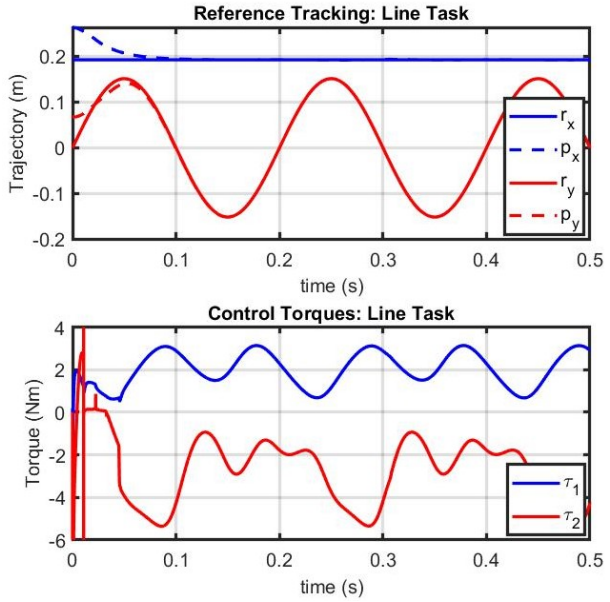

Fig. 7. Top: The desired reference trajectory is compared to the robot output for the cyclical line tracking task. Bottom: The control torques indicate that significant inertial dynamics and couplings between the actuated DOFs are being overcome.

V. Conclusions

We have presented a constrained Lagrangian modeling approach that was shown to be equivalent to a first-principles PDE classical Kirchhoff rod model. The construction of the constrained model was shown with $3N$ maximal coordinates, $2N$ constraint equations, and $2N$ unknown constraint forces. Based on the maximal coordinate form of this model, we derived an input-output feedback linearization control approach that maps the small number of actuation torques of the highly underactuated system to the second time-derivative of the output coordinates of interest, leveraging the strong inertial coupling present in the model. Singularities were addressed with a damped pseudo-inverse approach to this mapping. The feedback linearization was augmented with a sliding mode approach to drive the error to zero in the face of parametric uncertainties. This resulted in a robust control law. Finally an observer was formulated from the constrained model formulation by augmenting it with corrective virtual external forces based on the errors in the measured and expected states.

Simulations of a 2-DOF soft robot prototype demonstrated the salient features of the modeling, model-based control design, and observer design in performing two dynamic position tracking tasks: the tip position of the robot tracing a circle and a line at a frequency that induces significant dynamic loads. These results showed robustness to parameter differences between the plant and controller as well as excellent tracking performance.

References

[1] J. Till, V. Aloi, and C. Rucker, "Real-time dynamics of soft and continuum robots based on cosserat rod models," *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 723–746, 2019.

[2] S. H. Sadati, S. E. Naghibi, A. Shiva, B. Michael, L. Renson, M. Howard, C. D. Rucker, K. Althoefer, T. Nanayakkara, S. Zschaler, C. Bergeles, H. Hauser, and I. D. Walker, "Tmtdyn: A matlab package for modeling and control of hybrid rigid–continuum robots based on discretized lumped systems and reduced-order models," *The International Journal of Robotics Research*, vol. 40, no. 1, pp. 296–347, 2021.

[3] 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)*, 2019, pp. 454–461.

[4] H. Wang, C. Wang, W. Chen, X. Liang, and Y. Liu, "Three-dimensional dynamics for cable-driven soft manipulator," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 18–28, 2017.

[5] V. Falkenhahn, T. Mahl, A. Hildebrandt, R. Neumann, and O. Sawodny, "Dynamic modeling of bellows-actuated continuum robots using the euler–lagrange formalism," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1483–1496, 2015.

[6] 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, 2021.

[7] F. Renda, C. Armanini, V. Lebastard, F. Candelier, and F. Boyer, "A geometric variable-strain approach for static modeling of soft manipulators with tendon and fluidic actuation," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4006–4013, 2020.

[8] S. Mbakop, G. Tagne, M.-H. Frouin, A. Melingui, and R. Merzouki, "Inverse dynamics model-based shape control of soft continuum finger robot using parametric curve," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8053–8060, 2021.

[9] C. Wang, C. G. Frazelle, J. R. Wagner, and I. D. Walker, "Dynamic control of multisection three-dimensional continuum manipulators based on virtual discrete-jointed robot models," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 2, pp. 777–788, 2021.

[10] D. C. Rucker and R. J. Webster III, "Statics and Dynamics of Continuum Robots With General Tendon Routing and External Loading," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1033–1044, dec 2011.

[11] W. Huang, X. Huang, C. Majidi, and M. K. Jawed, "Dynamic simulation of articulated soft robots," *Nature Communications*, vol. 11, 2020.

[12] N. N. Goldberg, X. Huang, C. Majidi, A. Novelia, O. M. O'Reilly, D. A. Paley, and W. L. Scott, "On planar discrete elastic rod models for the locomotion of soft robots," *Soft Robotics*, vol. 6, no. 5, pp. 595–610, 2019, pMID: 31112073.

[13] C. D. 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, 2020.

[14] V. Falkenhahn, A. Hildebrandt, R. Neumann, and O. Sawodny, "Dynamic control of the bionic handling assistant," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 6–17, 2017.

[15] M. Thieffry, A. Kruszewski, C. Duriez, and T.-M. Guerra, "Control design for soft robots based on reduced-order model," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 25–32, 2019.

[16] S. Li, A. Kruszewski, T.-M. Guerra, and A.-T. Nguyen, "Equivalent-input-disturbance-based dynamic tracking control for soft robots via reduced-order finite-element models," *IEEE/ASME Transactions on Mechatronics*, pp. 1–12, 2022.

[17] I. A. Gravagne, C. D. Rahn, and I. D. Walker, "Large-Deflection Dynamics and Control for Planar Continuum Robots," *IEEE/ASME Transactions on Mechatronics*, vol. 8, no. 2, pp. 299–307, jun 2003.

[18] C. D. Santina, L. Pallottino, D. Rus, and A. Bicchi, "Exact task execution in highly under-actuated soft limbs: An operational space based approach," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2508–2515, 2019.

[19] S. S. Antman, *Nonlinear Problems of Elasticity*, 2nd ed., S. S. Antman, J. E. Marsden, and L. Sirovich, Eds. Springer Science, 2005.

[20] M. Spong, "Partial feedback linearization of underactuated mechanical systems," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, vol. 1, 1994, pp. 314–321 vol.1.

[21] H. Guillard and H. Bourles, "Robust feedback linearization," in *Proc. 14 th International Symposium on Mathematical Theory of Networks and Systems*, 2000.