



First-Order Dynamic Modeling and Control of Soft Robots

Thomas George Thuruthel^{1*}, Federico Renda² and Fumiya Iida¹

¹ Bio-Inspired Robotics Lab, Department of Engineering, University of Cambridge, Cambridge, United Kingdom, ² Khalifa University Center for Autonomous Robotic Systems, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates

Modeling of soft robots is typically performed at the static level or at a second-order fully dynamic level. Controllers developed upon these models have several advantages and disadvantages. Static controllers, based on the kinematic relations tend to be the easiest to develop, but by sacrificing accuracy, efficiency and the natural dynamics. Controllers developed using second-order dynamic models tend to be computationally expensive, but allow optimal control. Here we propose that the dynamic model of a soft robot can be reduced to first-order dynamical equation owing to their high damping and low inertial properties, as typically observed in nature, with minimal loss in accuracy. This paper investigates the validity of this assumption and the advantages it provides to the modeling and control of soft robots. Our results demonstrate that this model approximation is a powerful tool for developing closed-loop task-space dynamic controllers for soft robots by simplifying the planning and sensory feedback process with minimal effects on the controller accuracy.

Keywords: soft robotics, control, machine learning, dynamic modeling, first-order dynamics, model reduction

OPEN ACCESS

Edited by:

Concepción A. Monje,
Universidad Carlos III de
Madrid, Spain

Reviewed by:

Zheyuan Gong,
University of Toronto, Canada
Yisheng Guan,
Guangdong University of
Technology, China

*Correspondence:

Thomas George Thuruthel
tg444@cam.ac.uk

Specialty section:

This article was submitted to
Soft Robotics,
a section of the journal
Frontiers in Robotics and AI

Received: 25 March 2020

Accepted: 12 June 2020

Published: 21 July 2020

Citation:

George Thuruthel T, Renda F and
Iida F (2020) First-Order Dynamic
Modeling and Control of Soft Robots.
Front. Robot. AI 7:95.
doi: 10.3389/frobt.2020.00095

1. INTRODUCTION

Soft robotic technologies are becoming increasingly prevalent in the design and development of robots (Kim et al., 2013). Subsequently, there has been growing interests in the modeling and control of soft bodied systems, Unlike robots designed with rigid components, soft robotic systems present novel challenges and opportunities in developing their controllers (George Thuruthel et al., 2018).

The most common modeling and control strategy for soft robots are based on steady-state models, which, under the steady-state assumption, can be equated to the kinematic model (George Thuruthel et al., 2018). For cylindrically-shaped soft robots, this leads to the popular constant curvature model (Webster and Jones, 2010). For other shapes, geometrically exact models or Finite Element Method have been proposed (Trivedi et al., 2008; Renda et al., 2012; Duriez, 2013; Gong et al., 2018). Machine learning techniques can also be used to develop such mappings in a model-free manner (Giorelli et al., 2013; George Thuruthel et al., 2017; Jiang et al., 2017). Refer to Sadati et al. (2017) for a detailed comparison into multiple static modeling techniques. Due to their steady-state assumptions, such controllers will, however, be limited in their reachability, efficiency, and speed. Therefore, controllers developed from dynamic models are much more desirable.

A popular method for developing dynamic models for soft robots is based on the cosserat-rod mechanics. Such models have been extensively used for soft robotic manipulators driven by tendon actuation (Rucker and Webster, 2011; Renda et al., 2014, 2018). For fluidic actuation, other models have been adopted (Marchese et al., 2016; Della Santina et al., 2019). Hybrid models based on

lumped mass systems also looks promising for general soft robots (Sadati et al., 2019). However, all these models will be more computationally intensive than their static counterparts. Learning-based models are a promising alternative in such cases (Thuruthel et al., 2017; Gillespie et al., 2018). Nevertheless, deriving control strategies from dynamic models, in general, introduces additional complexities in motion planning. For an alternate approach that introduces a control-oriented modeling of soft robots, the readers are suggested to look into Della Santina and Rus (2019).

Unlike static controllers, developing fully-dynamic controllers would involve a planning stage. Typically, this has to be performed using some optimization techniques irrespective of the modeling strategy. A good example is the use of trajectory optimization for the control of a soft robotic manipulator using a model-based (Marchese et al., 2016) and model-free method (Thuruthel et al., 2017). This process is time consuming and hence debilitating for closed-loop dynamic control. For fully-actuated soft robots, closed-loop dynamic controllers can be developed in the configuration space (Della Santina et al., 2018). For task-space closed loop control, model-based reinforcement learning is the only viable solution till now, however, they tend to be highly task specific and time consuming (Thuruthel et al., 2018).

This article investigates the viability of a first-order dynamic model for soft robots. It must be noted that unlike state-space dimensionality reduction methods (Thieffry et al., 2018), we are reducing the temporal dimensionality of the dynamic model. Such a model reduction is based on the hypothesis that soft robots typically have high damping and low inertial properties. This makes it possible to approximate the second-order dynamic model to a first-order dynamic model by ignoring the inertial terms (Strogatz, 2001). Even in nature, the ubiquitous muscle dynamics can be modeled as a first-order dynamical system (Zajac, 1989). This model-order reduction provides two advantages. First, first-order dynamical systems are computationally cheaper than second-order dynamical systems. Second, it opens the possibility to develop novel closed-loop control strategies using the reduced-order state feedback. Here, we show the direct learning of the operational space dynamics of the first-order dynamic model. Due to simplifying step, controllers can be easily developed using machine learning and a simple path planning algorithm. Moreover, the sensory requirements for closed-loop dynamic control is reduced because of the simplification.

We investigate the viability of this simplifying assumption using extensive simulation studies. First, we present the theoretical reasoning behind the first-order assumption and its corresponding controller. Then we briefly introduce the fully dynamic simulation model that is used to verify the learned forward models and the dynamic controller. Finally, we present details on the learning architecture together with results of the model and the closed-loop task-space controller.

2. THEORY

Given a soft robot that can be kinematically modeled by the configuration-space $q \in \mathbb{R}^n$, the task-space variable can be obtained by the kinematic transformation:

$$x = F(q) \quad (1)$$

Where, $x \in \mathbb{R}^m$ and $m \leq n$. The task-space variable is typically the pose of the end-effector and is to be controlled. The configuration space is the set of independent variables that fully defines the state of the robot. The fully dynamic model of the soft robot can then be represented using the configuration-space variable as:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (2)$$

Here, $M(q)$ represents the inertial properties, $C(q, \dot{q})$ combines the coriolis, centrifugal and damping elements, $G(q)$ represents the gravitational and stiffness effects and τ is the generalized force applied internally by the robot.

Soft robots typically have high damping values with low inertial properties. This is because they are commonly fabricated with viscoelastic materials with low material density. After the initial transient motion of a soft robot from rest (when $\dot{q} = 0$), the first order term dominates the second order term [i.e., $C(q, \dot{q})\dot{q} \gg M(q)\ddot{q}$]. Hence, the second-order term can be ignored without sacrificing the accuracy of the model (Zajac, 1989). The dominant modeling error will occur during the initial transient motion (Strogatz, 2001). This transforms the second-order dynamical model (Equation 2) into:

$$C(q, \dot{q})\dot{q} + G(q) = \tau \quad (3)$$

After discretizing the equation, the dynamic model can now be represented through the mapping: $(q_i, \tau_i) \rightarrow q_{i+1}$. Where, q_i, q_{i+1} are the current and the next configuration of the soft robot, respectively. Correspondingly, this implies that a closed-loop dynamic controller would require only the zero-order state feedback (q) for control.

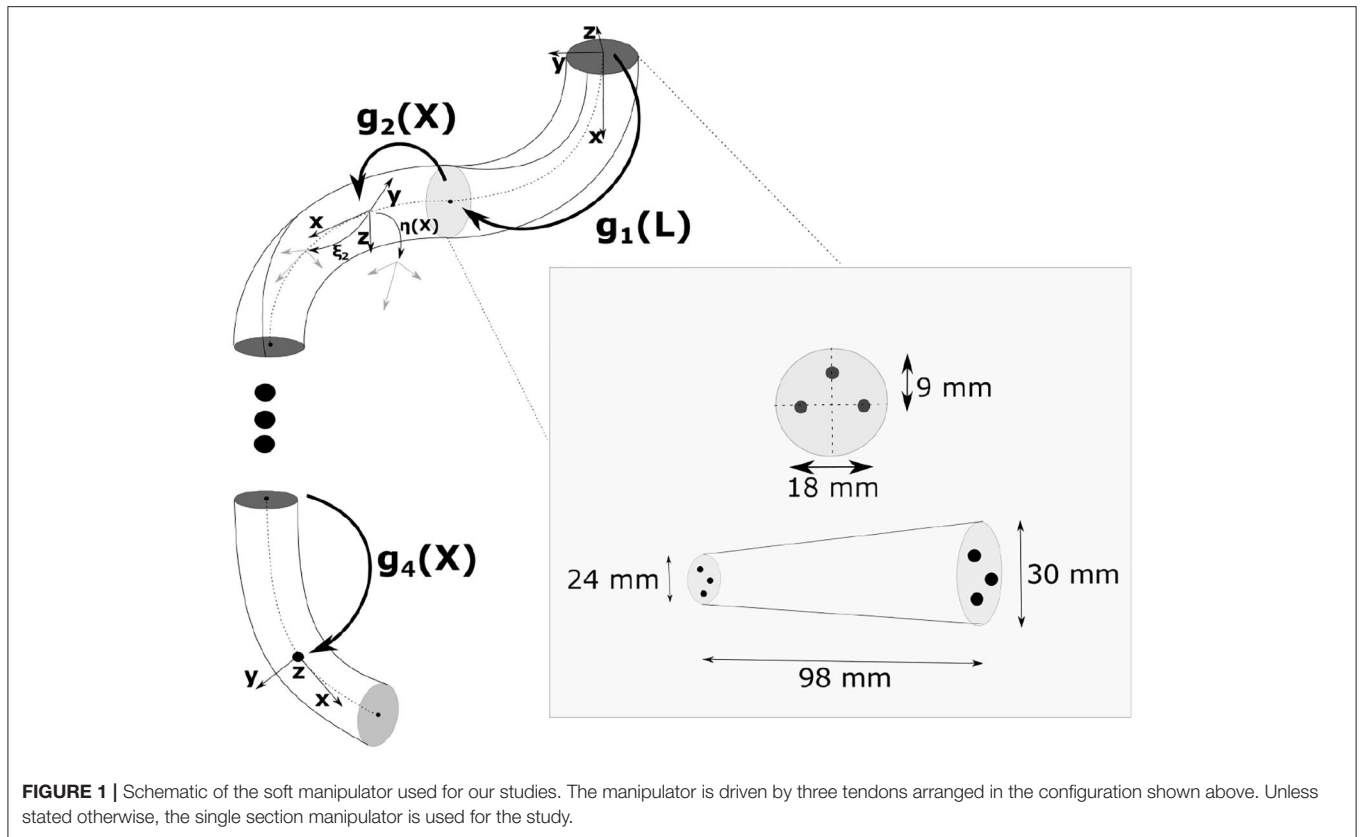
2.1. Controller Design

The obtained first-order dynamical Equation (3), the first-order configuration-space term can be replaced using the well known inverse kinematics mapping:

$$\begin{aligned} \dot{x} &= J(q)\dot{q} \\ \dot{q} &= J(q)^\dagger \dot{x} \end{aligned} \quad (4)$$

Where, $J(q)$ is Jacobian matrix and $J(q)^\dagger$ is any generalized inverse matrix. Note that we have ignored the null-space terms for brevity. Now the first-order dynamic equation can be reformulated as function of the task-space variables:

$$\tau = F(q, \dot{x}) \quad (5)$$



This functional mapping can now be directly learning by a machine learning architecture. We can now transform the mapping and introduce the target task-space variable x^d as:

$$\dot{x} = x^d - x^c \quad (6)$$

The operational space controller mapping now becomes:

$$\tau = F(q^c, x^d) \quad (7)$$

Here, (q^c, x^c) are the current configuration-space coordinates and the current task-space coordinates, respectively. For the special case when the cardinality of the configuration-space coordinate is the same as the cardinality of the task-space coordinate (mapping from $q \rightarrow x$ is bijective), the operational-space controller can be further simplified to a mapping: $(x^c, x^d) \rightarrow \tau$. A simple feedforward neural network can be used to learn this mapping (Figure 2). In this paper, we restrict our studies to this condition for simplicity. This allows us to test the learned controller by providing trajectories in the task-space without the need to solve the inverse kinematics problem. For redundant task-space controllers (non-bijective mappings), additional planning stages might be required to obtain the configuration-space trajectories. In other words, the control trajectory cannot be represented only in the task-space variables. In such a case, an augmented trajectory can be defined with the task-space variable along with a optimization routine to check for kinematic constraints as shown

before in Thuruthel et al. (2017). Note that the trajectories are only zero-order task space variables and the feedback required for closed-loop control is also zero-order. This greatly reduces the requirement on the sensors, the effect of noise and the complexity of the trajectory planner.

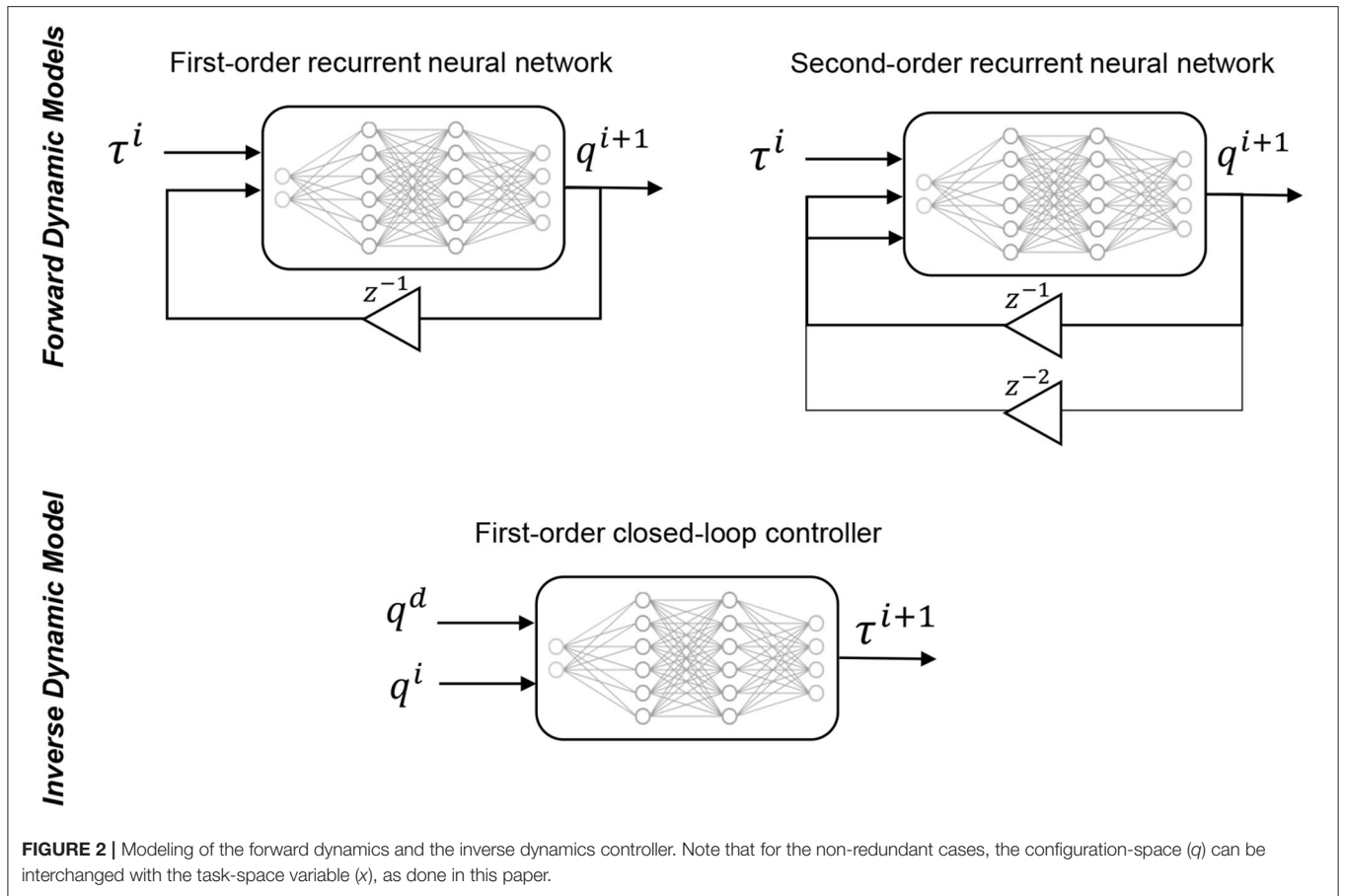
3. SIMULATION MODEL

The dynamic model is based on the Piece-wise Constant Strain (PCS) approach for soft-rigid multibody system of Renda and Seneviratne (2018) (see Figure 1). In the following, all the quantities are expressed in the local (body) coordinate frame if not specified otherwise. The superscript ' and $\dot{}$ represent partial differentiation with respect to the space variable and time variables, respectively. The accent \sim represents the usual isomorphism between a vector in \mathbb{R}^3 and its corresponding skew-symmetric matrix in $\mathfrak{so}(3)$.

3.1. Kinematics

The relative position and orientation of a soft body i with respect to its predecessor in the chain is defined as a curve $\mathbf{g}_i(\cdot): X \in [0, L_i] \mapsto \mathbf{g}_i(X) \in SE(3)$ with

$$\mathbf{g}_i(X) = \begin{pmatrix} \mathbf{R}_i & \mathbf{u}_i \\ \mathbf{0}^T & 1 \end{pmatrix}.$$

**TABLE 1** | Parameters of the learned forward dynamic model.

Parameter	Value
Type	NARX network
Hidden layer size	40
No. of samples	7000
Training algorithm	Levenberg-Marquardt backpropagation
Training:Testing:Validation ratio	70:15:15
Stopping criterion	Validation set error
Maximum no. of epochs	100

The continuous models of the position, velocity and acceleration of a soft body can be derived from the Cosserat rod theory, which gives (Boyer and Renda, 2016):

$$\mathbf{g}'_i = \mathbf{g}_i \hat{\xi}_i, \quad (8a)$$

$$\boldsymbol{\eta}'_i = \dot{\xi}_i - \text{ad}_{\xi_i} \boldsymbol{\eta}_i, \quad (8b)$$

$$\dot{\boldsymbol{\eta}}'_i = \ddot{\xi}_i - \text{ad}_{\xi_i} \boldsymbol{\eta}_i - \text{ad}_{\xi_i} \dot{\boldsymbol{\eta}}_i, \quad (8c)$$

where $\xi_i(X) \in \mathbb{R}^6$ defines the strain state, $\boldsymbol{\eta}_i(X) \in \mathbb{R}^6$ is the cross-section velocity twist and $\text{ad}_{(\cdot)} \in \mathbb{R}^{6 \times 6}$ is the adjoint operator

of the Lie algebra (see Nomenclature). Going further into detail, we have

$$\hat{\xi}_i(X) = \begin{pmatrix} \tilde{\mathbf{k}}_i & \mathbf{p}_i \\ \mathbf{0}^T & 0 \end{pmatrix} \in \mathfrak{se}(3), \quad \xi_i(X) = \begin{pmatrix} \mathbf{k}_i^T & \mathbf{p}_i^T \end{pmatrix}^T \in \mathbb{R}^6,$$

$$\hat{\boldsymbol{\eta}}_i(X) = \begin{pmatrix} \tilde{\mathbf{w}}_i & \mathbf{v}_i \\ \mathbf{0}^T & 0 \end{pmatrix} \in \mathfrak{se}(3), \quad \boldsymbol{\eta}_i(X) = \begin{pmatrix} \mathbf{w}_i^T & \mathbf{v}_i^T \end{pmatrix}^T \in \mathbb{R}^6,$$

with $\mathbf{k}_i(X) \in \mathbb{R}^3$ and $\mathbf{p}_i(X) \in \mathbb{R}^3$ the angular and linear strain; and $\mathbf{w}_i(X) \in \mathbb{R}^3$ and $\mathbf{v}_i(X) \in \mathbb{R}^3$ the angular and linear velocity, respectively.

To model constrained rod, such as the Kirchhoff-Love case with angular strain only, the strain field is specified as:

$$\xi_i = \mathbf{B}_i \mathbf{q}_i + \xi_i^*,$$

where $\mathbf{B}_i \in \mathbb{R}^{6 \times n_i}$ forms a basis for the allowed motion subspace, $\mathbf{q}_i \in \mathbb{R}^{n_i}$ contains the values of the allowed strains and, $\xi_i^* \in \mathbb{R}^6$ is the reference twist modeling the reference shape.

Assuming piece-wise constant strains (Renda et al., 2016), Equations (8) can be analytically integrated using the matrix exponential method, leading to:

$$g_i(X) = e^{X\hat{\xi}_i}, \tag{9a}$$

$$\eta_i(X) = \text{Ad}_{g_i}^{-1}\eta_h + \text{Ad}_{g_i}^{-1}\text{T}_{g_i}\mathbf{B}_i\dot{q}_i, \tag{9b}$$

$$\dot{\eta}_i(X) = \text{Ad}_{g_i}^{-1}\dot{\eta}_h + \text{Ad}_{g_i}^{-1}\int_0^X \text{Ad}_{g_i(s)}\text{ad}_{\eta_i(s)}ds\mathbf{B}_i\dot{q}_i + \text{Ad}_{g_i}^{-1}\text{T}_{g_i}\mathbf{B}_i\ddot{q}_i, \tag{9c}$$

where $\text{Ad}_{g_i}(X) \in \mathbb{R}^{6 \times 6}$ is the Adjoint operator of $SE(3)$, and $\text{T}_{g_i}(X) = \int_0^X e^{s\text{ad}_{\xi_i}}ds$ is the tangent operator of the exponential map, of which an analytic expression, derived from (Selig, 2007), is given in the Nomenclature.

Successive applications of the kinematics (Equation 9) for all the bodies of the system, yields to the definition of the geometric Jacobian $J_i(\mathbf{q}, X) \in \mathbb{R}^{6 \times n}$ and its derivative $\dot{J}_i(\mathbf{q}, \dot{\mathbf{q}}, X) \in \mathbb{R}^{6 \times n}$ (n

being the total number of DOFs), which relates the generalized coordinate vector $\mathbf{q} = [q_1^T \ q_1^T \ \dots \ q_N^T]^T \in \mathbb{R}^n$ (N being the total number of bodies) and the velocity twist $\eta_i(X)$, for each soft body i , as shown below.

$$\eta_i(X) = \sum_{h=0}^i \text{Ad}_{g_h \dots g_i}^{-1} \text{T}_{g_h} \mathbf{B}_h \dot{q}_h = \sum_{h=0}^i {}^i S_h \dot{q}_h = J_i(\mathbf{q}, X) \dot{\mathbf{q}}, \tag{10a}$$

$$\begin{aligned} \dot{\eta}_i(X) &= \sum_{h=0}^i {}^i \dot{S}_h \dot{q}_h + \text{Ad}_{g_h \dots g_i}^{-1} \int_0^X \text{Ad}_{g_h(s)} \text{ad}_{\eta_h(s)} ds \mathbf{B}_h \dot{q}_h \\ &= \sum_{h=0}^i {}^i \dot{S}_h \dot{q}_h + {}^i \dot{S}_h \dot{q}_h = J_i(\mathbf{q}, X) \ddot{\mathbf{q}} + \dot{J}_i(\mathbf{q}, \dot{\mathbf{q}}, X) \dot{\mathbf{q}}, \end{aligned} \tag{10b}$$

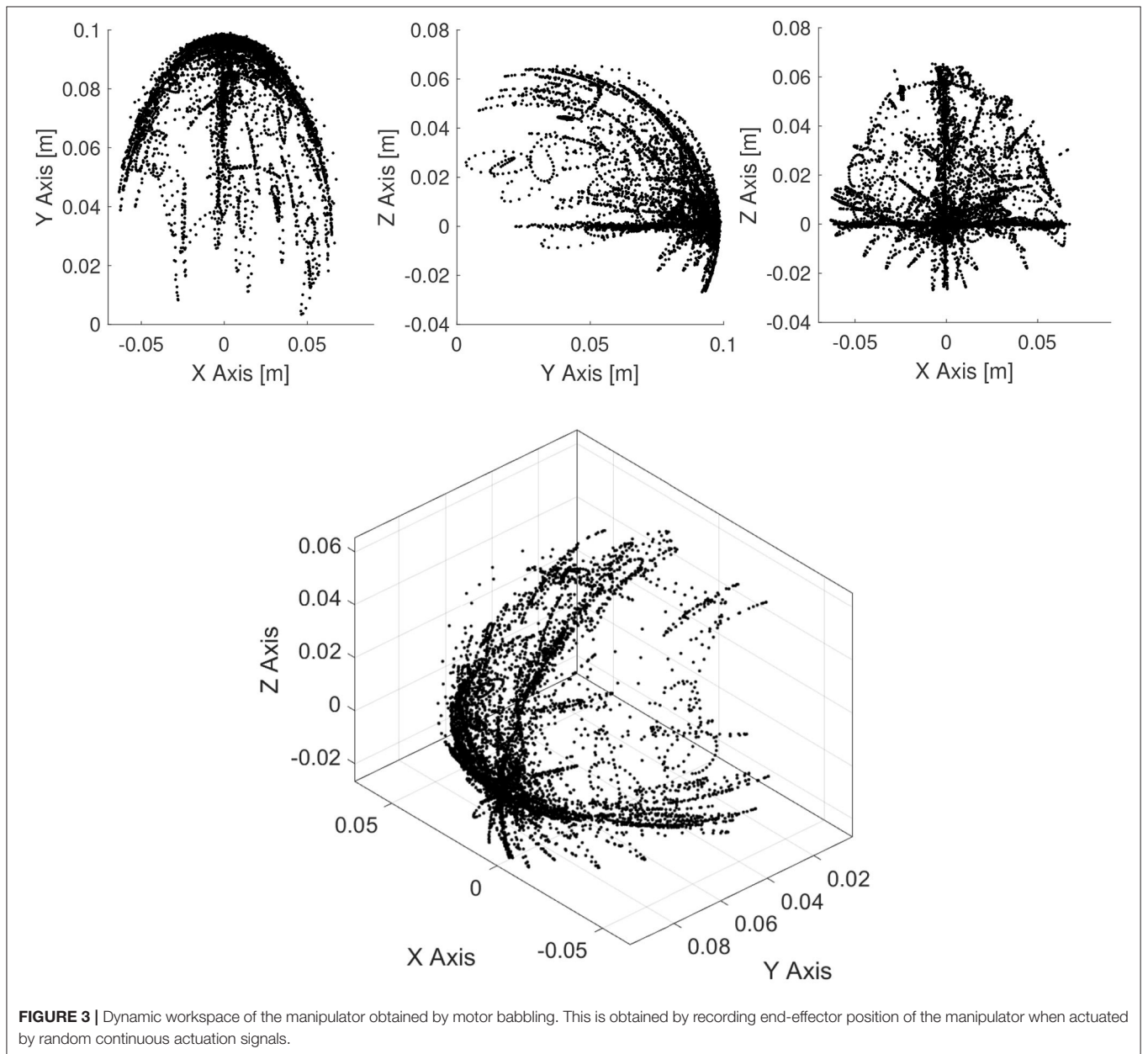


FIGURE 3 | Dynamic workspace of the manipulator obtained by motor babbling. This is obtained by recording end-effector position of the manipulator when actuated by random continuous actuation signals.

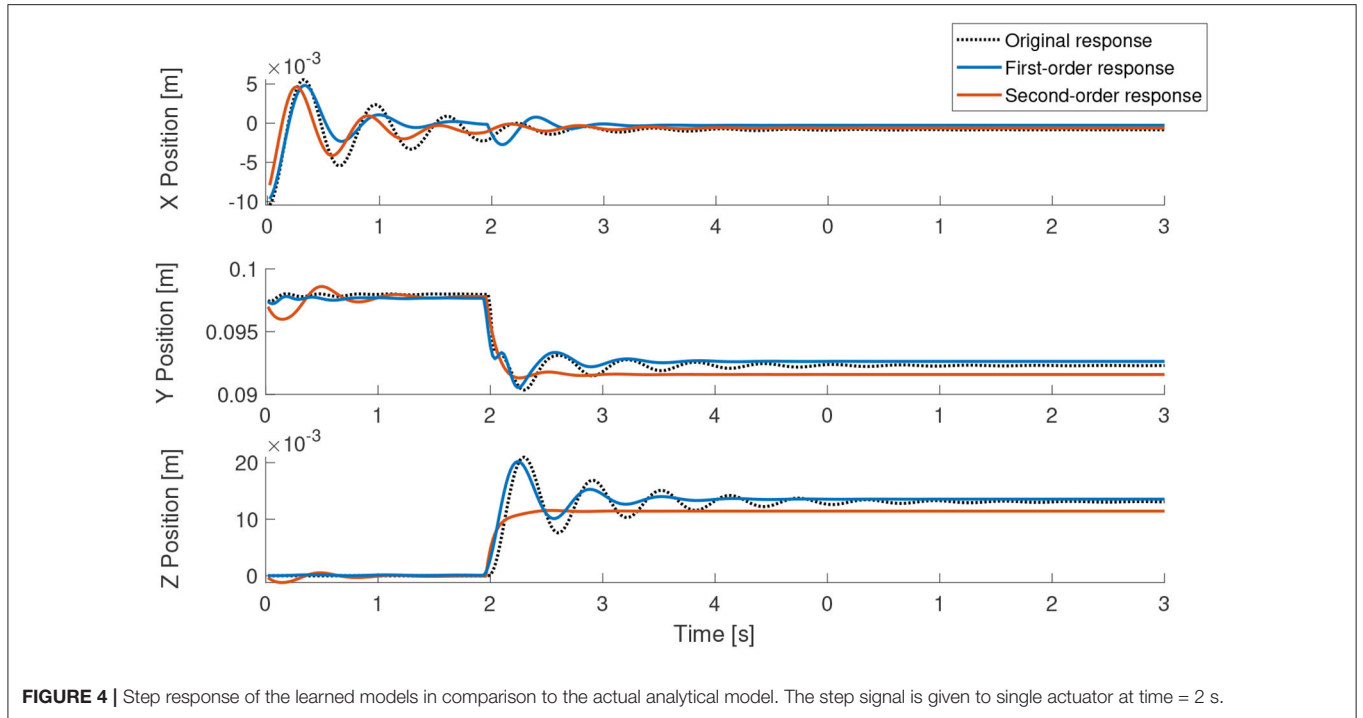


FIGURE 4 | Step response of the learned models in comparison to the actual analytical model. The step signal is given to single actuator at time = 2 s.

where the block elements of the i^{th} Jacobian ${}^i\mathcal{S}_{(\cdot)} \in \mathbb{R}^{6 \times n(\cdot)}$ and its derivative ${}^i\dot{\mathcal{S}}_{(\cdot)} \in \mathbb{R}^{6 \times n(\cdot)}$ have been defined. Note that the last three rows of Equation (10a) provide an analytical expression of the kinematics map required by Equation (4).

3.2. Dynamics

Once a Jacobian is found, the generalized dynamics of the system can be obtained by projecting the free dynamics of each soft body by virtue of the D'Alembert's principle. The free dynamic equation, with its boundary conditions, of a soft body is given by (Renda et al., 2018):

$$\begin{aligned} \mathcal{M}_i \dot{\eta}_i + \text{ad}_{\eta_i}^* \mathcal{M}_i \eta_i &= (\mathcal{F}_{i_i} - \mathcal{F}_{a_i})' + \text{ad}_{\xi_i}^* (\mathcal{F}_{i_i} - \mathcal{F}_{a_i}) + \bar{\mathcal{F}}_{e_i}, \\ (\mathcal{F}_{i_i} - \mathcal{F}_{a_i})(0) &= -\mathcal{F}_{J_i}, \quad (\mathcal{F}_{i_i} - \mathcal{F}_{a_i})(L_i) = -\text{Ad}_{g_{ij}}^* \mathcal{F}_{J_j}; \end{aligned} \quad (11)$$

where $\mathcal{M}_i(X) = \text{diag}(J_{x_i}, J_{y_i}, J_{z_i}, A_i, A_i, A_i) \rho_i \in \mathbb{R}^{6 \times 6}$ is the screw inertia matrix of the cross-section ($J_{\cdot}(X)$ being the second moment of area about the axis \cdot and $A_i(X)$ the area of the cross-section); $\bar{\mathcal{F}}_{e_i}(X) \in \mathbb{R}^6$ is the distributed external load; $\mathcal{F}_{a_i}(X) \in \mathbb{R}^6$ is the internal wrench due to the distributed actuation (Renda et al., 2017); $\mathcal{F}_{i_i}(X) \in \mathbb{R}^6$ is the internal wrench due to the elasticity of the soft body; $\mathcal{F}_{J(\cdot)} \in \mathbb{R}^6$ is the wrench transmitted across joint (\cdot) and $\text{ad}_{(\cdot)}^*$ (respectively $\text{Ad}_{(\cdot)}^*$) $\in \mathbb{R}^{6 \times 6}$ is the co-adjoint (respectively co-Adjoint) map of the Lie algebra (respectively Lie group) defined in Nomenclature. Regarding the internal elastic force, a linear viscoelastic constitutive model is

usually chosen:

$$\mathcal{F}_{i_i}(X) = \Sigma_i (\xi_i - \xi_i^*) + \Upsilon_i \dot{\xi}_i = \Sigma_i B_i q_i + \Upsilon_i B_i \dot{q}_i, \quad (12)$$

where

$$\Sigma_i(X) = \text{diag}(G_i J_{x_i}, E_i J_{y_i}, E_i J_{z_i}, E_i A_i, G_i A_i, G_i A_i) \in \mathbb{R}^{6 \times 6} \text{ and}$$

$$\Upsilon_i(X) = \text{diag}(J_{x_i}, 3J_{y_i}, 3J_{z_i}, 3A_i, A_i, A_i) v_i \in \mathbb{R}^{6 \times 6}$$

are the screw stiffness and viscosity matrix (E_i being the young modulus, G_i the shear modulus and v_i the shear viscosity).

By Jacobian projection of the free dynamics (Equation 11), we obtain the generalized dynamics in its classical configuration-space form:

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + (\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{D}) \dot{\mathbf{q}} + \mathbf{K} \mathbf{q} = \boldsymbol{\tau}(\mathbf{q}) + \mathbf{F}(\mathbf{q}), \quad (13)$$

where $\mathbf{M} \in \mathbb{R}^{n \times n}$ is the generalized mass matrix, $\mathbf{C} \in \mathbb{R}^{n \times n}$ is the generalized Coriolis matrix, $\mathbf{D} \in \mathbb{R}^{n \times n}$ is the block-diagonal generalized damping matrix, $\mathbf{K} \in \mathbb{R}^{n \times n}$ is the block-diagonal generalized stiffness matrix, $\mathbf{F} \in \mathbb{R}^n$ is the vector of generalized position-dependent external forces and $\boldsymbol{\tau} \in \mathbb{R}^n$ is the vector of applied actuation forces. Note that the dynamic Equation (13) can be written in the form required by Equation (2), with

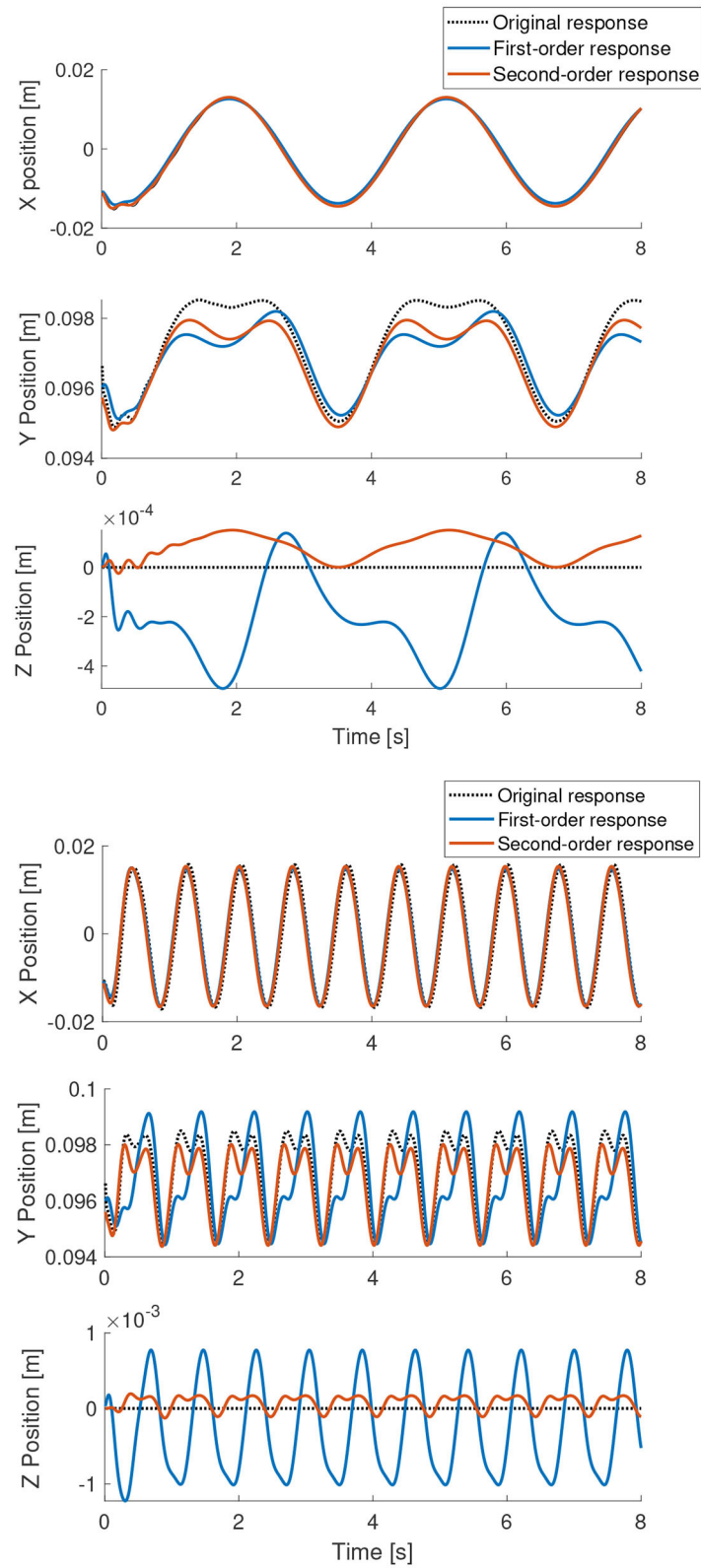


FIGURE 5 | Periodic response of the learned models in comparison to the actual analytical model.

$C(q, \dot{q}) = C(q, \dot{q}) + D$ and $G(q) = Kq - F(q)$. Going further into details, the coefficient matrices take the form:

$$M(q) = \sum_{i=1}^N \int_0^{L_i} J_i^T \mathcal{M}_i J_i dX, \quad (14a)$$

$$C(q, \dot{q}) = \sum_{i=1}^N \int_0^{L_i} J_i^T \left(\text{ad}_{J_i \dot{q}}^* \mathcal{M}_i J_i + \mathcal{M}_i \dot{J}_i \right) dX \quad (14b)$$

$$D = \text{diag} \left(B_1^T \int_0^{L_1} \Upsilon_1 dXB_1, B_2^T \int_0^{L_2} \Upsilon_2 dXB_2, \dots, B_N^T \int_0^{L_N} \Upsilon_N dXB_N \right), \quad (14c)$$

$$K = \text{diag} \left(B_1^T \int_0^{L_1} \Sigma_1 dXB_1, B_2^T \int_0^{L_2} \Sigma_2 dXB_2, \dots, B_N^T \int_0^{L_N} \Sigma_N dXB_N \right), \quad (14d)$$

$$\tau(q) = \left[\left(B_1^T \int_0^{L_1} \mathcal{F}_{a_1} dX \right)^T \left(B_2^T \int_0^{L_2} \mathcal{F}_{a_2} dX \right)^T \dots \left(B_N^T \int_0^{L_N} \mathcal{F}_{a_N} dX \right)^T \right]^T, \quad (14e)$$

$$F(q) = \sum_{i=1}^N \int_0^{L_i} J_i^T \bar{\mathcal{F}}_{e_i}. \quad (14f)$$

It is worth noting here the different structure of the components of the generalized dynamics Equation (13). Similarly to the minimal Lagrangian models of traditional rigid robots, inertial loads are characterized by full coefficient matrices, as can be seen from Equations (14a) and (14b), while damping and stiffness loads are characterized by block-diagonal coefficient matrices, as for Equations (14c) and (14d). This is in contrast with other modeling approaches that use absolute coordinates, such as Finite Elements, for which the opposite holds. Inertial coefficient matrices are block-diagonal while damping and stiffness coefficient matrices are full. Thus, neglecting inertial terms will be well suited for minimal Lagrangian models for soft robotic manipulator, such as the PCS approach.

4. METHODS AND RESULTS

This section investigates two studies. First, we validate the accuracy of the learned first-order model with respect to the learned second-order model. Second, we perform simulated experiments to validate the accuracy of the proposed controller. All the tests are performed on the fully dynamic model described in section 3.

4.1. Dynamic Modeling

The learned models are derived using a kind of recurrent neural network called a nonlinear autoregressive exogenous (NARX) model (Billings, 2013). The NARX network is particularly suited

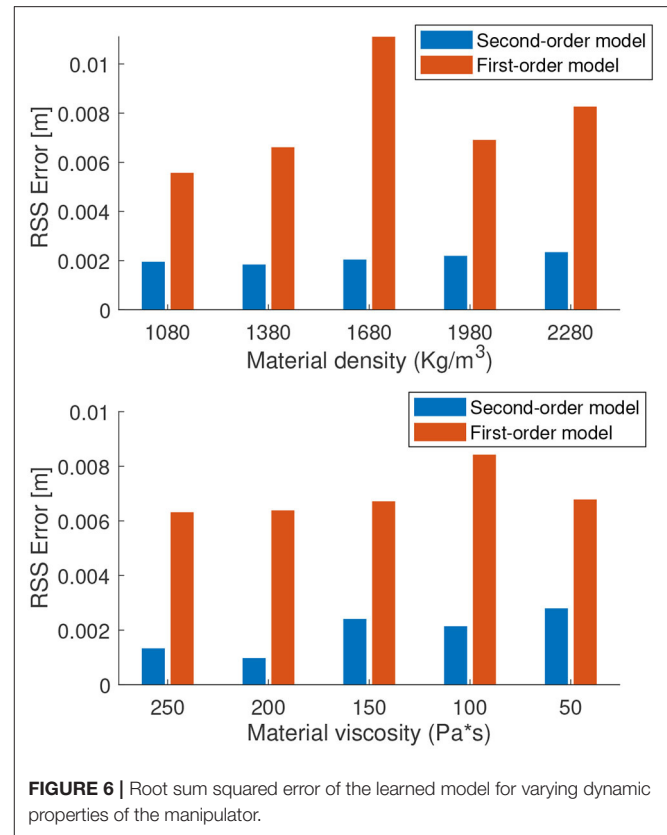


FIGURE 6 | Root sum squared error of the learned model for varying dynamic properties of the manipulator.

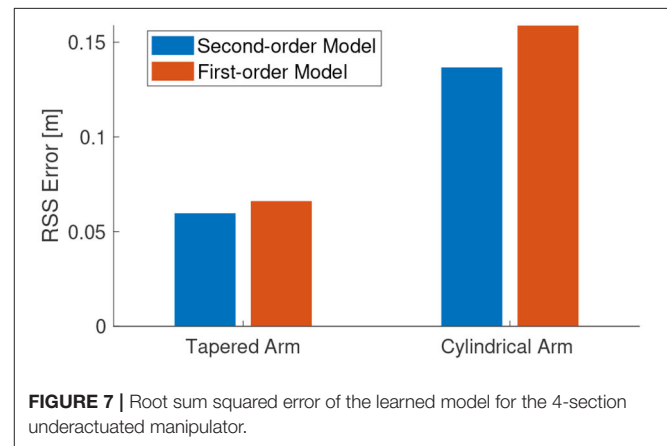
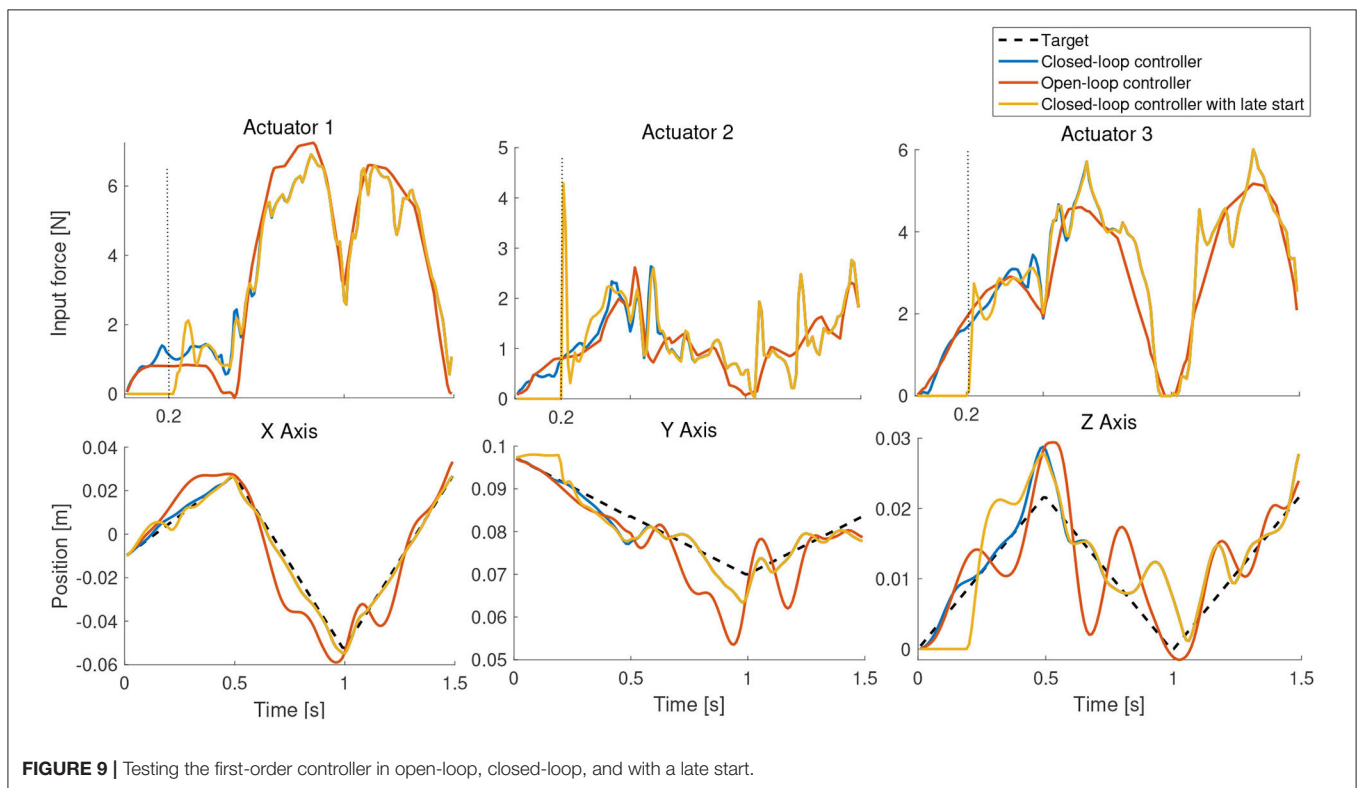
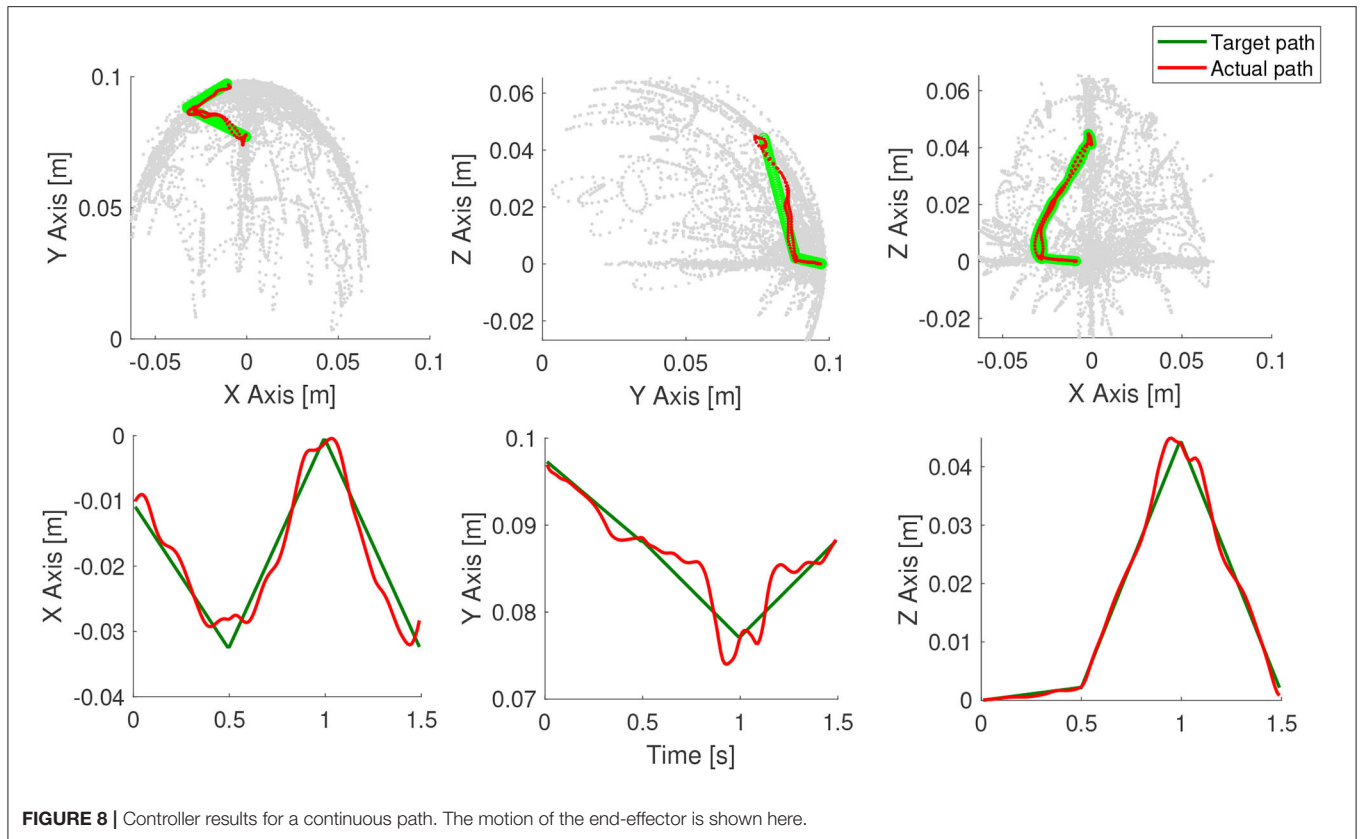
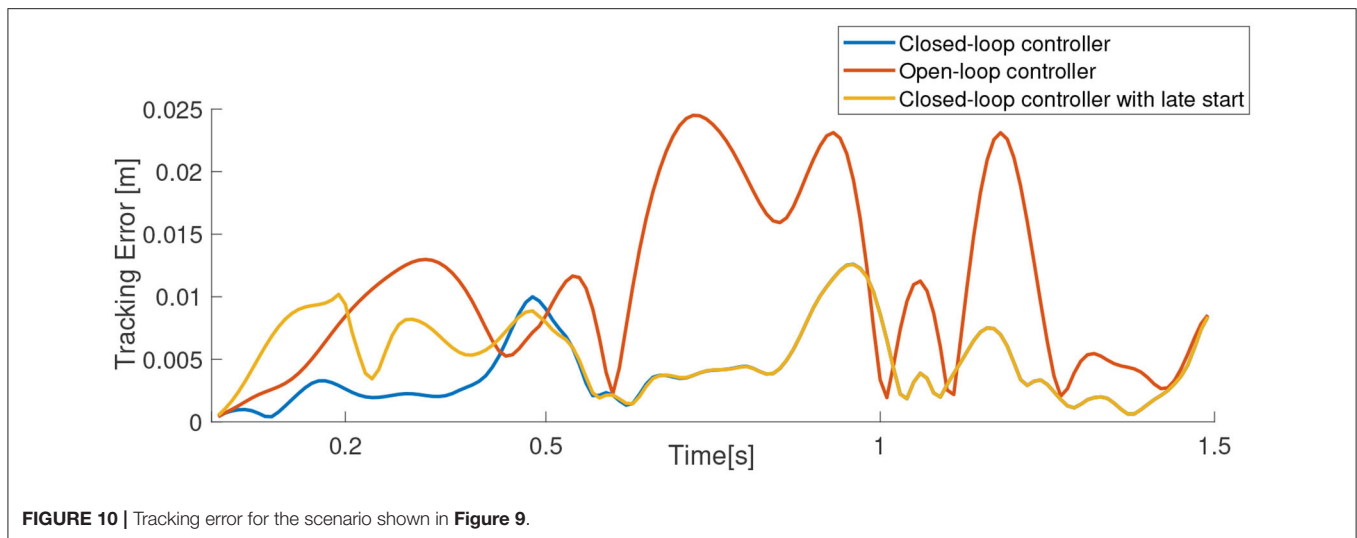


FIGURE 7 | Root sum squared error of the learned model for the 4-section underactuated manipulator.

TABLE 2 | Parameters of the first-order inverse dynamics controller.

Parameter	Value
Type	Feedforward neural network
Hidden layer size	30
No. of samples	7,000
Training algorithm	Levenberg-Marquardt backpropagation
Training:Testing:Validation ratio	70:15:15
Stopping criterion	Validation set error
Maximum no. of epochs	100





for our study as it allows us to define the feedback horizon of the recurrent connection explicitly (**Figure 2**). In other words, we can ensure that the neural network receives only zeroth-order feedback for the first-order model and the appended first-order feedback for the second-order model. For the single section soft manipulator, the configuration-space (q) is equivalent to the task-space variable (x), which is defined as the three-dimensional position of the end-effector. We use a recurrent network with one hidden layer with a size of 40 for both the first-order and second-order model, for a fair comparison. The training parameters of the NARX network is given in **Table 1**.

Random actuation of the tendons are performed (motor babbling) for 70 s to obtain the samples for learning the forward model (**Figure 3**). Specifically, random actuation inputs are used to drive the manipulator and the corresponding actuator inputs and end-effector position is recorded over time. The same data samples can also be used for developing the closed-loop controller. Note that the dynamic workspace is concentrated along the direction of the three actuators. This is because tendons in tension have a strong attractor behavior. Hence, it will be easier to move along this direction. When testing our controller, careful measure is taken to ensure that our trajectories pass across this workspace regions.

Figures 4, 5 show the performance of the learned model in comparison to the original cosserat model for a step and periodic response, respectively. Note the higher errors in the first-order model in the beginning of motion for the step response. Since the inertial effects are ignored, it is also visible that oscillations caused by overshoot is not found in the first-order model. However, the steady state error, with respect to the second-order model is relatively small. For the periodic excitation case, the difference between the first-order model and the second-order model is almost non-existent in the relevant coordinates. This is as expected since our approximations are more valid when the manipulator is in a non-stationary state.

In order to analyze the effects of the viscosity and the inertial effects on our modeling assumption, we further perform studies

on the accuracy of the first-order model for varying material viscosity and density. As material density increases and the material viscosity decreases, the inertial effects become more and more dominant. Hence, one would expect the accuracy of the first-order model to decrease and the second-order model to remain constant. However, this is not necessarily the case as the training of second-order recurrent neural networks is more prone to instabilities (Pascanu et al., 2013).

Figure 6 shows how the root sum squared (RSS) error of the first-order model is affected when the material properties of the soft arm is changed in a way that weakens our main assumption. The material density is increased up to a factor of 2 and the material viscosity is reduced by a factor of 6. The motor babbling inputs and the neural network parameters are kept the same for the tests. It is clear from the prediction errors that the second-order model always performs better than the first-order model. However, the change in accuracy of both the models are not affected significantly by the change in material properties. Note that the initial parameters of the simulated cosserat model soft arm were obtained from real experiments on an Octopus-inspired soft manipulator, which was manufactured with silicone and driven by tendons (Renda et al., 2014).

Increasing the length of the manipulator is another way to increase the inertial properties and weaken the first-order assumption. For this, we test the same methodology on a 4 section manipulator, however, actuated only on the first section. Each section has approximately the same length, with the total length of the manipulator adding to 418 mm. We test two designs, one with a tapered morphology and the other with a cylindrical morphology (higher inertial properties). The radius of the tapered morphology linearly reduces from 30 to 10 mm while the cylindrical morphology has a fixed diameter of 30 mm. For this test and the following controller results, the default parameters of the manipulator is used (i.e., with the material density of $1,080 \text{ kg/m}^3$ and viscosity of 300 Pas; Renda et al., 2014). The results of our forward dynamics prediction on both the systems are shown in **Figure 7**. Contrary to our

expectation, increasing the length of the manipulator did not affect the performance of the first-order model when compared to the second-order model. We believe this is because the inertial effects are still compensated by the medium (water) in which the manipulator is surrounded in. Our previous studies have shown that the dynamics of the manipulator becomes chaotic and hence unpredictable without a surrounding medium and when the length increases (Thuruthel et al., 2019). Based on our results, it can be deduced that this is more because of the first-order terms rather than the second-order terms. It could be because of the increased length, gravity, centripetal/centrifugal forces, higher DoFs etc. Note that this is a limitation of learning-based approaches as increased sensitivity to initial conditions decrease the stability of the training process and hence the performance of the model. The same applies to the parameter tuning process for analytical models. It could be concluded that it is a good practice to reduce the inertial effects of a soft-bodied system to improve the predictability of its dynamics, irrespective of the order of the model.

4.2. First-Order Dynamic Controller

The closed-loop task-space controller is derived by learning the operational-space dynamics mapping as described in section 2.1. As the mapping is not recursive, it can be learned using a simple feedforward neural network, as shown in **Figure 2**. For our non-redundant case, we can replace the configuration-space variable, q , with the task-space variable, x . The samples for learning the mapping is obtained through the same motor babbling process as described in the previous section. For training the controller the mapping is defined as: $(x^i, x^{i+1}) \rightarrow \tau$. When testing the controller, the next task-space coordinate, x^{i+1} , is replaced by the desired task-space variable x^d . The parameters of the neural network used for learning the first-order inverse dynamics mapping is given in **Table 2**.

Path planning is usually a complex problem in inverse dynamics based controllers, but as our inverse model is developed with only zeroth order state feedback, the development of the task-space trajectory is greatly simplified. Acceptable paths can easily be generated using the data points obtained from the workspace of the manipulator, which is obtained during the motor babbling phase. The desired paths can be generated by picking reachable points from the workspace and routing a path through them, ensuring that there is sufficient time for the manipulator to reach adjacent points. This can be easily done by fixing a cap on the maximum distance between adjacent task-space variables.

To test our controller, we generate randomized linear paths for the end-effector of the manipulator to follow. This is done by picking two random points from the robot workspace and linearly interpolating a trajectory between them and from the initial position of the end-effector. If the initial position of the end-effector is p^0 and the two random points are p^1 and p^2 , the generated path is from $p^0 \rightarrow p^1 \rightarrow p^2 \rightarrow p^1$. Note that the intermediate points are not necessarily reachable by this naive approach. Accurate trajectories can be generated by searching for adjacent points in the workspace or by projecting the trajectory onto the workspace surface.

The results of the trajectory tracking is shown in **Figure 8**. The performance of the controller is excellent considering the fact that it is myopic with no step-ahead planning and the task-space trajectory being non-optimal. Due to the low computational cost in running the inverse-dynamic controller, we are able to run closed-loop task-space controller at a very high control frequency of 100 Hz. This will also allow the controller to compensate for any modeling errors incurred by the approximation.

The *quality* of the task-space trajectory can be analyzed by running the controller in open-loop. This can be done by assuming the manipulator is able to reach each trajectory point perfectly and obtaining the best control action at each time step. The results of such a scenario is shown in **Figure 9**, along with the corresponding control inputs. As expected, the open-loop controller performs worse than the closed-loop controller even though there are no external disturbances in the simulation. The same scenario is repeated with the controller inactive for the first 0.2 s in **Figure 9**. As the desired targets are now farther from the current position of the manipulator, it is not necessary that the controller is able to follow the trajectory accurately. However, as evident from the results, the controller is able to recover and remarkably converge to the same solution as the original closed-loop controller. This also shows how important it is to close-the-loop, even at the cost of reduced model accuracy. The tracking error of the three tests described in this scenario is shown in **Figure 10**.

5. CONCLUSION

This paper presents and verifies a model simplifying assumption for soft robots. The core idea of the assumption is that soft robots, by definition, tend to have low inertial and high viscoelastic properties. This leads to dynamic behaviors which are well approximated by a first-order system, as typically observed in nature. We verify this assumption using a simulated fully dynamic model of an Octopus-like manipulator and a type of recurrent neural network called NARX network. Finally, we develop easy-to-develop closed-loop task-space dynamic controllers based on this assumption. Our results indicate that controllers developed on this assumption can compensate the errors in modeling accuracy with the increased control frequency. Our method makes path planning simpler for non-redundant cases. Moreover, the sensory requirements for closed-loop dynamic control is reduced because of the simplification. This is because the state feedback required for the controller is only the zeroth-order component. Our work also indicates that the additional modeling complexity that soft elements introduce can, to some extent, be reduced by designing low inertial highly visco-elastic soft robot designs.

Although we use machine learning tools to test our modeling assumption and develop our controller, the approach is equally suited for analytical approaches. In fact, ignoring the inertial elements would greatly simplify the modeling and parameter estimation process involved in model-based control of soft robots. Not only can we reduce the states of the dynamical system, but we also avoid the problem of estimating and inverting

the full mass matrix (see section 3.2). However, it must be kept in mind that first-order systems present numerical challenges in their implementation. Such problems are not found in a learning based approach and hence desirable in that respect. Typically, the first-order model leads to a stiff differential equation and requires specialized techniques for solving them. Interested readers are suggested to refer to Strogatz (2001) for further information. Future work involve extending the work to redundant systems.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author/s.

REFERENCES

- Billings, S. A. (2013). *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*. Chichester: John Wiley & Sons.
- Boyer, F., and Renda, F. (2016). Poincaré's equations for cosserat media: application to shells. *J. Nonlin. Sci.* 27, 1–44. doi: 10.1007/s00332-016-9324-7
- Della Santina, C., Katzschmann, R. K., Bicchi, A., and Rus, D. (2019). Model-based dynamic feedback control of a planar soft robot: trajectory tracking and interaction with the environment. *Int. J. Robot. Res.* 39, 490–513. doi: 10.1177/0278364919897292
- Della Santina, C., Katzschmann, R. K., Biechi, A., and Rus, D. (2018). "Dynamic control of soft robots interacting with the environment," in 2018 *IEEE International Conference on Soft Robotics* (Livorno: RoboSoft), 46–53.
- Della Santina, C., and Rus, D. (2019). Control oriented modeling of soft robots: the polynomial curvature case. *IEEE Robot. Autom. Lett.* 5, 290–298. doi: 10.1109/LRA.2019.2955936
- Duriez, C. (2013). "Control of elastic soft robots based on real-time finite element method," in 2013 *IEEE International Conference on Robotics and Automation* (Karlsruhe), 3982–3987.
- George Thuruthel, T., Ansari, Y., Falotico, E., and Laschi, C. (2018). Control strategies for soft robotic manipulators: a survey. *Soft Robot.* 5, 149–163. doi: 10.1089/soro.2017.0007
- George Thuruthel, T., Falotico, E., Manti, M., Pratesi, A., Cianchetti, M., and Laschi, C. (2017). Learning closed loop kinematic controllers for continuum manipulators in unstructured environments. *Soft Robot.* 4, 285–296. doi: 10.1089/soro.2016.0051
- Gillespie, M. T., Best, C. M., Townsend, E. C., Wingate, D., and Killpack, M. D. (2018). "Learning nonlinear dynamic models of soft robots for model predictive control with neural networks," in 2018 *IEEE International Conference on Soft Robotics* (Livorno: RoboSoft), 39–45.
- Giorelli, M., Renda, F., Ferri, G., and Laschi, C. (2013). "A feed-forward neural network learning the inverse kinetics of a soft cable-driven manipulator moving in three-dimensional space," in 2013 *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Tokyo), 5033–5039.
- Gong, Z., Cheng, J., Hu, K., Wang, T., and Wen, L. (2018). "An inverse kinematics method of a soft robotic arm with three-dimensional locomotion for underwater manipulation," in 2018 *IEEE International Conference on Soft Robotics* (Livorno: RoboSoft), 516–521.
- Jiang, H., Wang, Z., Liu, X., Chen, X., Jin, Y., You, X., et al. (2017). "A two-level approach for solving the inverse kinematics of an extensible soft arm considering viscoelastic behavior," in 2017 *IEEE International Conference on Robotics and Automation* (ICRA) (Singapore), 6127–6133.
- Kim, S., Laschi, C., and Trimmer, B. (2013). Soft robotics: a bioinspired evolution in robotics. *Trends Biotechnol.* 31, 287–294. doi: 10.1016/j.tibtech.2013.03.002
- Marchese, A. D., Tedrake, R., and Rus, D. (2016). Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator. *Int. J. Robot. Res.* 35, 1000–1019. doi: 10.1109/ICRA.2015.7139538

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

FUNDING

This publication was partially supported by Khalifa University of Science and Technology under Award Nos. FSU-2018-08, RC1-2018-KUCARS and Research Excellence (AARE) 2018-05 and by the SHERO project, a Future and Emerging Technologies (FET) programme of the European Commission (grant agreement ID 828818).

- Pascanu, R., Mikolov, T., and Bengio, Y. (2013). "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning* (Atlanta, GA), 1310–1318.
- Renda, F., Boyer, F., Dias, J., and Seneviratne, L. (2018). Discrete cosserat approach for multisection soft manipulator dynamics. *IEEE Trans. Robot.* 34, 1518–1533. doi: 10.1109/TRO.2018.2868815
- Renda, F., Cacucciolo, V., Dias, J., and Seneviratne, L. (2016). "Discrete cosserat approach for soft robot dynamics: a new piece-wise constant strain model with torsion and shears," in 2016 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Daejeon), 5495–5502.
- Renda, F., Cianchetti, M., Abidi, H., Dias, J., and Seneviratne, L. (2017). Screw-Based Modeling of Soft Manipulators With Tendon and Fluidic Actuation. *J. Mechanisms Robotics.* 9(4):041012. doi: 10.1115/1.4036579
- Renda, F., Cianchetti, M., Giorelli, M., Arienti, A., and Laschi, C. (2012). A 3D steady-state model of a tendon-driven continuum soft manipulator inspired by the octopus arm. *Bioinspir. Biomimet.* 7:025006. doi: 10.1088/1748-3182/7/2/025006
- Renda, F., Giorelli, M., Calisti, M., Cianchetti, M., and Laschi, C. (2014). Dynamic model of a multibending soft robot arm driven by cables. *IEEE Trans. Robot.* 30, 1109–1122.
- Renda, F., and Seneviratne, L. (2018). "A geometric and unified approach for modeling soft-rigid multi-body systems with lumped and distributed degrees of freedom," in 2018 *IEEE International Conference on Robotics and Automation* (ICRA) (Brisbane, QLD), 1567–1574. doi: 10.1109/ICRA.2018.8461118
- Rucker, D. C., and Webster, R. J. III. (2011). Statics and dynamics of continuum robots with general tendon routing and external loading. *IEEE Trans. Robot.* 27, 1033–1044.
- Sadati, S. H., Naghibi, S. E., Shiva, A., Michael, B., Renson, L., Howard, M., et al. (2019). TMTDyn: A matlab package for modeling and control of hybrid rigid-continuum robots based on discretized lumped systems and reduced-order models. *Int. J. Robot. Res.* doi: 10.1177/0278364919881685. [Epub ahead of print].
- Sadati, S. H., Naghibi, S. E., Shiva, A., Walker, I. D., Althoefer, K., and Nanayakkara, T. (2017). "Mechanics of continuum manipulators, a comparative study of five methods with experiments," in *Annual Conference Towards Autonomous Robotic Systems* (Guildford, UK: Springer), 686–702.
- Selig, J. (2007). *Geometric Fundamentals of Robotics*. Monographs in Computer Science. New York, NY: Springer.
- Strogatz, S. (2001). *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering (Studies in Nonlinearity)*. Boca Raton, FL: CRC press.
- Thieffry, M., Kruszewski, A., Duriez, C., and Guerra, T.-M. (2018). Control design for soft robots based on reduced-order model. *IEEE Robot. Autom. Lett.* 4, 25–32. doi: 10.1109/LRA.2018.2876734
- Thuruthel, T. G., Falotico, E., Renda, F., Flash, T., and Laschi, C. (2019). Emergence of behavior through morphology: a case study on an octopus inspired manipulator. *Bioinspir. Biomimet.* 14:034001. doi: 10.1088/1748-3190/ab1621

- Thuruthel, T. G., Falotico, E., Renda, F., and Laschi, C. (2017). Learning dynamic models for open loop predictive control of soft robotic manipulators. *Bioinspir. Biomimet.* 12:066003. doi: 10.1088/1748-3190/aa839f
- Thuruthel, T. G., Falotico, E., Renda, F., and Laschi, C. (2018). Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators. *IEEE Trans. Robot.* 35, 124–134. doi: 10.1109/TRO.2018.2878318
- Trivedi, D., Lotfi, A., and Rahn, C. D. (2008). Geometrically exact models for soft robotic manipulators. *IEEE Trans. Robot.* 24, 773–780. doi: 10.1109/TRO.2008.924923
- Webster, R. J. III, and Jones, B. A. (2010). Design and kinematic modeling of constant curvature continuum robots: a review. *Int. J. Robot. Res.* 29, 1661–1683. doi: 10.1177/0278364910368147
- Zajac, F. E. (1989). Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Crit. Rev. Biomed. Eng.* 17, 359–411.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 George Thuruthel, Renda and Iida. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

NOMENCLATURE

$[\text{ad}_{\xi,\eta}] (X) = \begin{pmatrix} \tilde{\mathbf{k}}, \tilde{\mathbf{w}} & \mathbf{0}_{3 \times 3} \\ \tilde{\mathbf{q}}, \tilde{\mathbf{v}} & \mathbf{k}, \tilde{\mathbf{w}} \end{pmatrix} \in \mathbb{R}^{6 \times 6}$. adjoint representation of the strain twist vector in $SE(3)$.

$[\text{ad}_{\xi,\eta}^*] (X) = \begin{pmatrix} \tilde{\mathbf{k}}, \tilde{\mathbf{w}} & \tilde{\mathbf{q}}, \tilde{\mathbf{v}} \\ \mathbf{0}_{3 \times 3} & \mathbf{k}, \tilde{\mathbf{w}} \end{pmatrix} \in \mathbb{R}^{6 \times 6}$. coadjoint representation of the strain twist vector in $SE(3)$.

$[\text{Ad}_g] (X) = \begin{pmatrix} \mathbf{R} & \mathbf{0}_{3 \times 3} \\ \tilde{\mathbf{u}}\mathbf{R} & \mathbf{R} \end{pmatrix} \in \mathbb{R}^{6 \times 6}$. Adjoint representation of $SE(3)$.

$[\text{Ad}_g^*] (X) = \begin{pmatrix} \mathbf{R} & \tilde{\mathbf{u}}\mathbf{R} \\ \mathbf{0}_{3 \times 3} & \mathbf{R} \end{pmatrix} \in \mathbb{R}^{6 \times 6}$. coAdjoint representation of $SE(3)$.

$[e^{X\hat{\xi}}] = \mathbf{I}_4 + X\hat{\xi} + \frac{1}{\theta^2} (1 - \cos(X\theta)) \hat{\xi}^2 + \frac{1}{\theta^3} (X\theta - \sin(X\theta)) \hat{\xi}^3$. Exponential map in $SE(3)$ with $\theta^2 = \mathbf{k}^T \mathbf{k}$.

$[\text{T}_g] (X) = X\mathbf{I}_6 + \frac{1}{2\theta^2} (4 - 4 \cos(X\theta) - X\theta \sin(X\theta)) \text{ad}_{\xi} + \frac{1}{2\theta^3} (4X\theta - 5 \sin(X\theta) + X\theta \cos(X\theta)) \text{ad}_{\xi}^2 + \frac{1}{2\theta^4} (2 - 2 \cos(X\theta) - X\theta \sin(X\theta)) \text{ad}_{\xi}^3 + \frac{1}{2\theta^5} (2X\theta - 3 \sin(X\theta) + X\theta \cos(X\theta)) \text{ad}_{\xi}^4$. Analytic expression of the tangent operator of the exponential map in $SE(3)$.