

Dynamical Modeling and Control of Soft Robots with Non-constant Curvature Deformation

Zhanchi Wang, Gaotian Wang, Xiaoping Chen, and Nikolaos M. Freris, *Senior Member, IEEE*,

Abstract—The Piecewise Constant Curvature (PCC) model is the most widely used soft robotic modeling and control. However, the PCC fails to accurately describe the deformation of the soft robots when executing dynamic tasks or interacting with the environment. This paper presents a simple three-dimensional (3D) modeling method for a multi-segment soft robotic manipulator with non-constant curvature deformation. We devise kinematic and dynamical models for soft manipulators by modeling each segment of the manipulator as two stretchable links connected by a universal joint. Based on that, we present two controllers for dynamic trajectory tracking in configuration space and pose control in task space, respectively. Model accuracy is demonstrated with simulations and experimental data. The controllers are implemented on a four-segment soft robotic manipulator and validated in dynamic motions and pose control with unknown loads. The experimental results show that the dynamic controller enables a stable reference trajectory tracking at speeds up to $7m/s$.

Index Terms—Soft Robotics, modeling, dynamical control.

I. INTRODUCTION

BIOLICAL systems can generate dynamic motions efficiently as well as interact with the environment compliantly with their body's softness and elasticity [1], [2]. These systems have not only been a source of inspiration for soft robotics research [3]–[5] but also a near-perfect demonstration given by nature to answer the question of “why soft?” [6]. However, due to the infinite passive degrees of freedom, developing models and controllers for soft robots to execute dynamic tasks or interact with the environment is one of the great challenges in the field [7], [8].

Unlike traditional robots consisting of rigid links and discrete joints, soft robots are generally made of soft materials, which can produce continuous deformation and exhibit passive compliance. Simulating and predicting these properties is the primary goal of soft robot modeling. Investigating into the mechanics of soft materials, some models that can characterize continuous deformation have been developed, such as Cosserot rod model [9]–[11], Euler Bernoulli beam model [12], and so on. These models are widely used for the simulation of soft robots. However, solving these models involves the computation of nonlinear partial differential equations, which are computationally expensive. Applying these models for control often requires the development of additional methods to reduce model dimensionality [13], [14].

Z. Wang, X. Chen, and N. Freris are with the School of Computer Science and G. Wang is with the School of Physics; University of Science and Technology of China; (Emails: {zkdwzc, vectorwang}@mail.ustc.edu.cn, {xpchen, nfr}@ustc.edu.cn.)

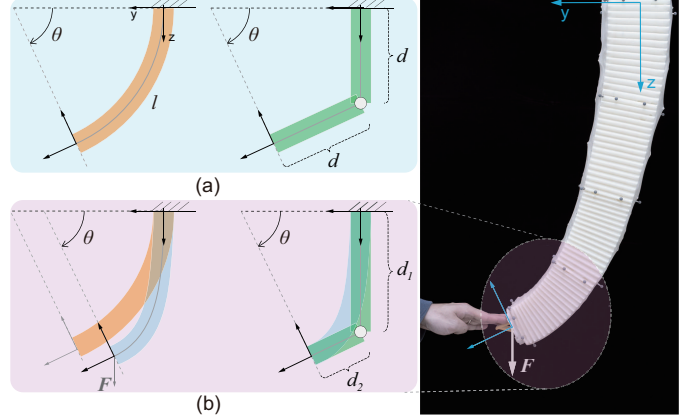


Fig. 1. Representation of a soft robot without (a) and with (b) external force with a constant curvature model and a rigid-link model.

A lower-dimensional, accurate, and yet simple model is important for analyzing and control of soft robots and will contribute to the long-term success of the field. The most commonly used simplified model of soft robotics is the Piecewise Constant Curvature (PCC) model [15]. This type of model treats each segment of the soft robot as an arc of constant curvature [16]. The PCC model is widely used for motion control [17], [18] and motion planning [19] of soft robots. Although the PCC model is simple and intuitive, it has obvious limitations: not all soft robots deform according to segmented constant curvature, as shown in Figure 1. The constant curvature modeling method has three important assumptions: 1) axial torsion is not considered, 2) each single-segment deforms in a plane, and 3) the single-segment deforms in an arc manner. The first two assumptions are reasonable for most soft robot designs. But the last assumption can be easily violated when the soft robots interact with the environment [20], perform dynamic motions [21], or even the hardware have a non-uniform design [22]. In these cases, there will be inevitable errors in describing soft robots with the PCC model, which affects the reliability of the PCC-based controllers. More importantly, as the number of segments increases, the errors accumulate and propagate to the end effector, thus deteriorating the fidelity of forward kinematics, which is the basis for formulating dynamics and developing controllers.

In addition to the above modeling methods that can continuously characterize the behavior of soft robots, the deformation of soft robots can also be fitted with discrete rigid links and joints [23], [24], which inspires a large class of discrete

modeling methods [25]. These discrete methods use a small number of joint/link states to represent soft robots with infinite degrees of freedom based on some simplifying assumptions. An augmented rigid robot model for dynamical modeling of a 2D soft robot was presented in [26]. This model was extended to three dimensions (3D) by modeling a single segment of the soft manipulator as a rigid robot with ten joints [27], [28]. Another method of drawing equivalence of a 3D soft arm to a rigid robot and realizing dynamical control was demonstrated in [29]. However, these rigid-link robot models are still established under the PCC paradigm; that is, these rigid robot models can only accurately describe a robot that ideally conforms to PCC deformation, so there are still inevitable errors where the PCC model fails.

To address the non-constant curvature deformation of a soft manipulator, a variable constant curvature approximation was developed and used to realize the visual servo control in 3D space [30]. The same approach was used in [22] to address the variable curvature deformation caused by the non-uniform geometric design. Another model that approximates the continuous shape of a continuum manipulator by a serial chain of rigid links was presented in [31], which permits a description of manipulator shape under different loading conditions. However, these methods characterize complex deformations by finer discretization of single-segment soft robots, which implies a larger state space and increased computational cost. More importantly, how to sense the state of each unit in closed-loop control is a challenge, either requiring strong sensing methods [31], or assuming the motion of each unit conform to a specific pattern [32].

In this paper, we propose a modeling method that can describe the non-constant curvature deformation of soft manipulators in the presence of interactions and dynamic motions. Our method capitalizes on a Piecewise Universal Joint (PUJ) model, which regards each segment of the soft manipulator as two stretchable links connected by a universal joint. Compared with the PCC model, the PUJ model relaxes the constraint that one segment of the manipulator must be deformed in a constant curvature manner and achieves a more general representation where we prove PCC is a special case. The PUJ model formulates a soft robot as a classic rigid-link serial manipulator so that the analysis and control methods of rigid robots can be applied to soft robots. We present two closed-loop controllers for soft robotic arms in both configuration and task space. The effectiveness of both models and controllers are evaluated theoretically within the modeling hypotheses and through experiments on a four-segment soft robotic arm in 3D space.

In summary, this work provides a new theoretical framework and tools for modeling and control soft robots with non-constant curvature deformation. The main **contributions** of this paper enlist:

- We introduce a simple modeling methodology for soft robots called Piecewise Universal Joint (PUJ), which formulates soft robots into low-dimensional rigid manipulators.
- We present a kinematic model and a task space kinematic controller for a soft manipulator based on PUJ. The model

can describe the non-constant curvature deformation, and the controller can handle unknown loads.

- We present a dynamical model and a configuration space dynamical controller for a soft manipulator based on the PUJ. The model is capable of accurately predicting dynamical behavior. The dynamical controller can track a reference trajectory at speeds up to $7m/s$.
- We experimentally validate the proposed methodology, models, and controllers in simulations and on a 3D soft manipulator.

II. MODELING

This section presents a method for modeling soft robot kinematics and dynamics. By equating the soft robot to a rigid-link robot, the modeling methods and control strategies designed for rigid-body robots are transferred to the soft robot with consideration of joint viscoelasticity. We further show that the model presented here is a general form of the PCC model that can still be effective under external forces or interactions.

A. Modeling Intuition and Assumption

PCC is a widely used model, which discretizes the soft robot into circular arcs with fixed curvatures connected end to end. Although the PCC is a geometric model that can describe continuous deformation, the forward kinematics is generally obtained by the employment of the method developed for rigid link robots, for example, D-H parameters [16]. It is an intuitive idea to model soft robots with rigid links. Consider the deformation of a single planar segment in the ideal case; if we only focus on the position and orientation of the endpoint, we can describe it with both the PCC model and a revolute joint connecting two rigid links, as shown in Figure 1(a). When there is an external force, the single-segment soft robot no longer bends with a constant curvature (Figure 1). The PCC description will cause a deviation in position/orientation (Figure 1(b)). In this case, a revolute joint connecting two rigid links can still be used to accurately describe the position and orientation of the endpoint, as shown in Figure 1(b). This simple example serves as our inspiration for the methods presented in this paper.

A soft robotic manipulator that operates in the 3D space is usually composed of multiple independently actuated segments that can typically bend on a plane without torsional deformation. This characteristic serves as the chief modeling assumption in this paper. Under this assumption, the motion characteristics of a soft robotic manipulator are very similar to a rigid serial link mechanism, i.e., a series of universal joints. A universal joint (also called Cardan joint or Hooke's joint) is a compound joint whose two axes are orthogonal [33]. Universal joints are often used to design redundant robots and continuum robots to achieve semi-soft flexible motion [16], [34], [35]. In this paper, we capitalize on this assumption, termed Piecewise Universal Joint (PUJ), to model the kinematics and dynamics of a soft robotic manipulator in 3D as a series of rigid links connected by universal joints.

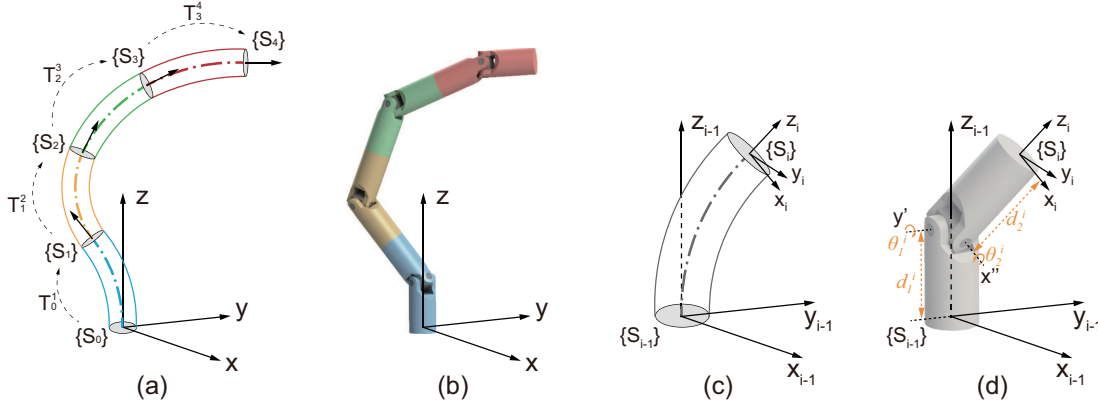


Fig. 2. An illustration of a soft robotic manipulator alongside its equivalent rigid robotic representation in 3D space. (a) The kinematic representation of a four-segment soft robotic manipulator, where reference frame S_0 is attached at the robot's base, and S_i is attached at the end of the i -th segment. The homogeneous transformation T_{i-1}^i maps S_{i-1} to S_i . (b) An equivalent representation of the soft manipulator based on PUJ. (c) An illustration of a single segment of the soft manipulator: S_{i-1} and S_i are the two coordinate frames attached at both ends of the segment. (d) A kinematic representation based on a universal joint: translation lengths d_1^i , d_2^i and rotation angles θ_1^i , θ_2^i are chosen as the configuration variables.

B. Kinematics

This section introduces a kinematic model for soft robotic manipulators, which maps the soft robot to a discrete joint robot.

1) *Kinematics Based on Piecewise Universal Joint:* Consider a soft manipulator composed of n segments. We denote the reference coordinate systems at the base of the manipulator and the ends of each segment as $\{S_0\}, \dots, \{S_n\}$, as illustrated in Figure 2(a). These coordinate systems are used to represent the state of the soft manipulator. We abstract the kinematics via a homogeneous transformation matrix $T_{i-1}^i \in \mathbb{R}^{4 \times 4}$ [33, Chapter 2] between consecutive coordinate systems:

$$T_{i-1}^i = \begin{bmatrix} R_{i-1}^i & \mathbf{p}_{i-1}^i \\ (0, 0, 0) & 1 \end{bmatrix}, \quad (1)$$

where $R_{i-1}^i \in \mathbb{R}^{3 \times 3}$ is a rotation matrix and $\mathbf{p}_{i-1}^i \in \mathbb{R}^3$ is a translation (column) vector. It follows that the forward kinematics of an n -segment soft robotic manipulator is given by the product of the segment transformation matrices as:

$$T_0^n = \prod_{i=1}^n T_{i-1}^i, \quad (2)$$

where T_{i-1}^i transforms vectors from the i -th coordinate frame to the $(i-1)$ -th, thus T_0^n serves to express the end effector in the manipulator base coordinate frame.

The n -segment soft robotic manipulator is equivalently represented by n serial universal joints (Figure 2(b)). Taking the i -th segment for illustration, we showcase how to select the configuration variables for the universal joint, and proceed to provide the expression for the homogeneous transformation matrix of the i -th segment (Figure 2(c)). The tuple $(d_1^i, \theta_1^i, \theta_2^i, d_2^i)$ represents the configuration variables, denote as \mathbf{q} , for the i -th segment. The coordinate system $\{S_i\}$ fixed at the tip of the i -th segment is obtained from $\{S_{i-1}\}$ via a sequence of four operations: 1) translation along the z -axis by d_1^i , 2) rotation around the translated y -axis by θ_1^i , 3) rotation around the rotated x -axis by θ_2^i , and 4) translation along the

twice rotated z -axis by d_2^i (see Figure 2(d)). The homogeneous transformation matrix for the i -th segment is written as:

$$T_{i-1}^i = \text{Trans}(z, d_1^i) \text{Rot}(y, \theta_1^i) \text{Rot}(x, \theta_2^i) \text{Trans}(z, d_2^i) \\ = \begin{bmatrix} c_{\theta_1^i} & s_{\theta_1^i} s_{\theta_2^i} & s_{\theta_1^i} c_{\theta_2^i} & d_2^i s_{\theta_1^i} c_{\theta_2^i} \\ 0 & c_{\theta_2^i} & -s_{\theta_2^i} & -d_2^i s_{\theta_2^i} \\ -s_{\theta_1^i} & c_{\theta_1^i} s_{\theta_2^i} & c_{\theta_1^i} c_{\theta_2^i} & d_1^i + d_2^i c_{\theta_1^i} c_{\theta_2^i} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3)$$

where $\text{Trans} \in \mathbb{R}^{4 \times 4}$ denotes the homogeneous transformation matrix of translation along an axis, $\text{Rot} \in \mathbb{R}^{4 \times 4}$ denotes the homogeneous transformation matrix of rotation around an axis, and c_θ , s_θ is respectively the abbreviation of $\cos \theta$, $\sin \theta$.

2) *State Estimation:* In this part, we consider the inverse problem, i.e., how to estimate the configuration variables $\hat{\mathbf{q}}$ and its derivative $\dot{\hat{\mathbf{q}}}$ from the posture of the reference coordinate systems. The postures $\{S_0\}, \dots, \{S_n\}$ are usually measured from sensors such as cameras. The 3×3 upper left block in (3) is the measured rotation matrix R_{i-1}^i and the 3×1 upper right block is the measured translation vector \mathbf{p}_{i-1}^i . Angles θ_1^i and θ_2^i are derived by converting the measured rotation matrix to the form of Z - Y - X Euler angles $(\alpha, \beta, \gamma)^\top$. For a universal joint, $\alpha = 0$, $\beta = \theta_1^i$, and $\gamma = \theta_2^i$. This procedure is rather standard using software packages (e.g., *rotm2eul* in Matlab) and is not detailed here, for length considerations.

The translation lengths d_1^i and d_2^i are derived from the measured translation vector $\mathbf{p}_{i-1}^i = (x_{i-1}^i, y_{i-1}^i, z_{i-1}^i)^\top$ as follows:

$$\begin{cases} d_2^i = -\frac{y_{i-1}^i}{s_{\theta_2^i}}, d_1^i = z_{i-1}^i - d_2^i c_{\theta_1^i} c_{\theta_2^i}, & \theta_2^i > \epsilon \ \& \ \theta_2^i > \theta_1^i \\ d_2^i = \frac{x_{i-1}^i}{s_{\theta_1^i} c_{\theta_2^i}}, d_1^i = z_{i-1}^i - d_2^i c_{\theta_1^i} c_{\theta_2^i}, & \theta_1^i > \epsilon \ \& \ \theta_1^i > \theta_2^i \\ d_1^i = d_2^i = \frac{z_{i-1}^i}{1 + c_{\theta_1^i} c_{\theta_2^i}}, & \theta_1^i < \epsilon \ \& \ \theta_2^i < \epsilon \end{cases} \quad (4)$$

We set $\epsilon = 0.05 \text{ rad}$ in implementation. When the configuration of the segment is almost straight ($\theta_1^i < \epsilon$ & $\theta_2^i < \epsilon$), the two translation axes coincide, which causes the contribution of these two degrees of freedom to the end effector motion to be indistinguishable. In order to achieve a robust method for

state estimation, we use the last term of \mathbf{p}_{i-1}^i to obtain d_1^i and d_2^i with the constraint that $d_1^i = d_2^i$ in this scenario.

The velocity vector $\dot{\mathbf{q}}$ is simply calculated via the discrete derivative $\dot{\mathbf{q}} = \frac{\mathbf{q}_t - \mathbf{q}_{t-1}}{\delta t}$ where $\delta t = 0.005\text{s}$ was chosen for simulations and $\delta t = 0.05\text{s}$ for experiments on the real manipulator. The time granularity of discretization in the real experiments was chosen according to the actual response frequency of the pneumatic soft manipulator. We use the position of the end effector expressed in the base coordinate system $\{S_0\}$ as the estimation of $\hat{\mathbf{x}}$ for the task space controller proposed in Section III-B.

3) *Exponential Coordinate Representation*: Exponential coordinate representation is the most intuitive and convenient way to describe rigid body motions; we adopt this to describe the kinematics and dynamics of the serial universal joints. For brevity, we provide the overview here with a detailed description accessible at [36, Chapter 3].

The forward kinematics of manipulator composed of n serial universal joints is represented by the product of exponentials (PoE) as presented in [37]:

$$T(\mathbf{q}) = e^{[S_1]q_1} \dots e^{[S_n]q_n} H, \quad (5)$$

where $\mathbf{q} := (d_1^1, \theta_1^1, d_2^1, \theta_2^1, \dots, d_1^n, \theta_1^n, d_2^n, \theta_2^n)^\top \in \mathbb{R}^{4n}$ is the collection of all configuration variables; $T(\mathbf{q}) \in \mathbb{R}^{4 \times 4}$ is the pose of the end effector; $H \in \mathbb{R}^{4 \times 4}$ is the pose of the end-effector when all configuration variables of the manipulator are zero (called zero position); S_i is the normalized screw axis of joint i as expressed in the fixed base frame, with S_i given by:

$$[S_i] = \begin{bmatrix} [\omega_i] & \mathbf{v}_i \\ (0, 0, 0) & 0 \end{bmatrix}, [\omega_i] = \begin{bmatrix} 0 & -\omega_3^i & \omega_2^i \\ \omega_3^i & 0 & -\omega_1^i \\ -\omega_2^i & \omega_1^i & 0 \end{bmatrix}. \quad (6)$$

If joint i is revolute, $\omega_i := (\omega_1^i, \omega_2^i, \omega_3^i)^\top$ is a unit vector along the positive direction of the joint axis and $\mathbf{v}_i \in \mathbb{R}^3$ is the velocity of the point that coincides with the base coordinate system when the joint rotates in an angular velocity of $\|\omega_i\|$. If joint i is prismatic, $\omega_i = (0, 0, 0)^\top$ and $\mathbf{v}_i := (v_1^i, v_2^i, v_3^i)^\top$ is a unit vector along the positive direction of the joint axis. Table I lists the screw axes for a single universal joint as an example.

TABLE I
SCREW AXES S_i FOR ONE UNIVERSAL JOINT

i	ω_i	\mathbf{v}_i
1	$(0, 0, 0)^\top$	$(0, 0, 1)^\top$
2	$(0, 1, 0)^\top$	$(-d_0, 0, 0)^\top$
3	$(1, 0, 0)^\top$	$(0, d_0, 0)^\top$
4	$(0, 0, 0)^\top$	$(0, 0, 1)^\top$

d_0 is the initial length of the first link of a universal joint when at its zero position.

Unlike the commonly used Denavit–Hartenberg representation [38], the product of exponential representation has the advantage that no link reference frames need to be explicitly defined, resulting in simple and intuitive equations for kinematics and dynamics [39].

4) *Kinematic Model Properties*: The PUJ principle relaxes the constraint that each segment of the manipulator must be deformed in a constant curvature manner in PCC assumption and achieves a more general representation. In this section, we will take a single segment as an example and prove that the proposed model has some important properties that are the basis for developing controllers that can handle external forces in 3D gravitational space. The superscript symbol is dropped in the following development to simplify the notations.

Lemma 1. *For any position and orientation described by a tuple of PCC configuration variables (ϕ, ψ, l) , we can always find a tuple of PUJ configuration variables $(d_1, \theta_1, \theta_2, d_2)$ to represent the same where we have $d_1 = d_2$.*

Proof. In the most commonly used PCC representation, the orientation of z_i -axis of the tip coordinate system $\{S_i\}$ is derived by first rotates base coordinate system $\{S_{i-1}\}$ by ϕ around z_{i-1} -axis and then rotates ψ around the rotated y_{i-1} -axis, as shown in Figure 3(a), which is described as $(c_\phi s_\psi, s_\phi s_\psi, c_\psi)^\top$. As for PUJ representation, the process is first rotating θ_1 around y_{i-1} -axis and then θ_2 around the rotated x_{i-1} -axis, which is described as $(s\theta_1 c\theta_2, -s\theta_2, c\theta_1 c\theta_2)^\top$. Combining terms of these two we can get:

$$\begin{cases} \theta_1 = \tan^{-1}(\cos \phi \tan \psi) \\ \theta_2 = -\sin^{-1}(\sin \phi \sin \psi) \end{cases} \quad -\pi < \phi < \pi, 0 < \psi < \pi,$$

which means PUJ variables can equivalently describe any orientation described by PCC variables.

For the position representation of $\{S_i\}$, we can prove it from a geometric perspective. We draw a tangent line of the arc at its tip on the plane where the deformation occurs, as is the $x'-z_{i-1}$ plane shown in Figure 3(a). And set the position of the joints in the PUJ model as the point where the tangent line and z' -axis intersect. We can see from Figure 3(b), when d_1, d_2 , and l satisfy the relationship of $d_1 = d_2 = \frac{l}{\psi} \tan(\frac{\psi}{2})$, the positions of $\{S_i\}$ given by both PCC and PUJ presentations are the same. Thus Lemma 1 is proved.

Theorem 1. *The state space (position and orientation) described by the PCC model is a proper subset of the space described by the PUJ model.*

Proof. Here we use a counterexample to show that there exists a state denoted as coordinate system $\{S_i\}$ described by the PUJ model that can not be described by the PCC model. We consider a simple situation where PUJ and PCC are used to describe the soft robot deformation on the same $x'-z_{i-1}$ plane, as shown in Figure 3(b). When $d'_1 \neq d'_2$, it's obvious that there does not exist an arc to connect the base and tip and meet the orientation constraints of both ends on the $x'-z_{i-1}$ plane. For this state, the PUJ model can accurately describe the position and orientation of the soft robot, while the PCC model is bound to be inaccurate.

Lemma 2. *The state space of the relative complement of the PCC model in the PUJ model can only be reached when the soft robot is subjected to forces.*

Proof. Assume that at a certain initial moment, the position and orientation of the soft robot are accurately described by both PUJ and PCC models. From Lemma 1, we can infer that $d_1 = d_2$ in the PUJ model at this time. Then we

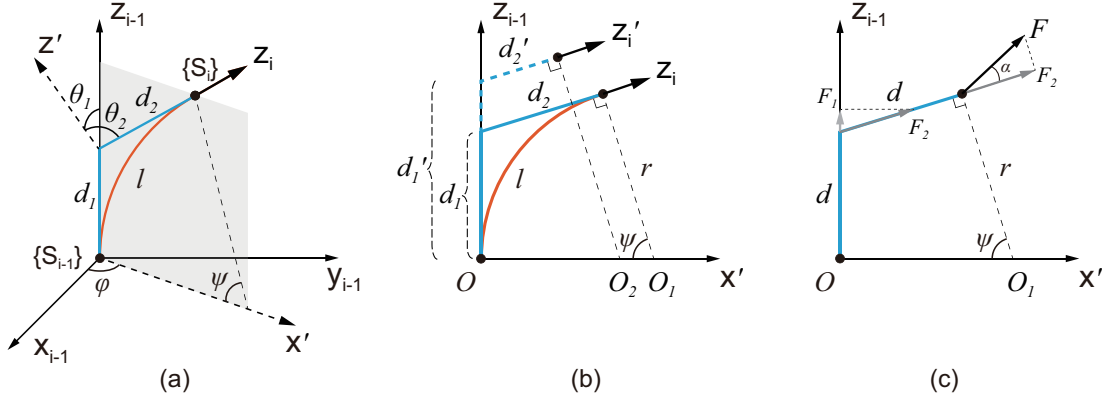


Fig. 3. An illustration of PCC representation of a soft robotic manipulator alongside its PUJ representation. (a) The illustration in 3D space for the equivalence of the two representations. (b) The illustration on $x'-z_{i-1}$ plane for the equivalence and difference of the two representations when d_1, d_2 vary. (c) The stress-strain analysis of the PUJ model under an external force on $x'-z_{i-1}$ plane.

consider the effect of force. As shown in Figure 3(c), an arbitrary external force acts on the tip of the segment. By decompose of the force, we can derive that the force on the second link $F_2 = F \cos \alpha$, and the force on the first link $F_1 = F_2 \cos \psi = F \cos \alpha \cos \psi$. As long as $\psi \neq 0$, $F_1 \neq F_2$. Assuming that the properties of the two links are the same, then $\ddot{d}_1 \neq \ddot{d}_2$ and $\dot{d}_1 \neq \dot{d}_2$. So d_1 will no longer equal to d_2 at the next moment. From the proof process of Theorem 1, we know that the PCC model can not accurately describe the position and orientation of the end-effector under forces.

Remark 1. Even if there is no external force, if the acceleration of the movement and the mass of the manipulator are not negligible, the force between the segments can not be ignored, which will lead to inaccuracy when applying the PCC model.

Corollary 1. Four coordinates are required to describe the position and orientation of the end effector for a soft manipulator that is arbitrarily deformed in a plane.

Proof. The minimum number of coordinates required to describe the position and pose of a body in Euclidean space is six. If we only care about the orientation rather than the pose of the end effector, we can describe it with five coordinates. Three of which describe the location, and two describe the orientation. Within the modeling assumptions that the soft robotic manipulator is deformed in a plane, the orientation and position of the end effector are not fully independent. Under this constraint, four coordinates are required to describe the position and orientation of the end effector for a soft manipulator. The PUJ model is such a method with four coordinates.

C. Dynamics

The dynamical characterization is of great importance for soft robots' design, analysis, and control. This section introduces a dynamical model based on the PUJ model presented in the prequel. The dynamical model is derived from rigid body dynamics. We consider the viscoelasticity of the soft manipulator by adding an elastic and a damping term into the dynamical equations. This methodology bears the advantage that the well-established dynamical system modeling and

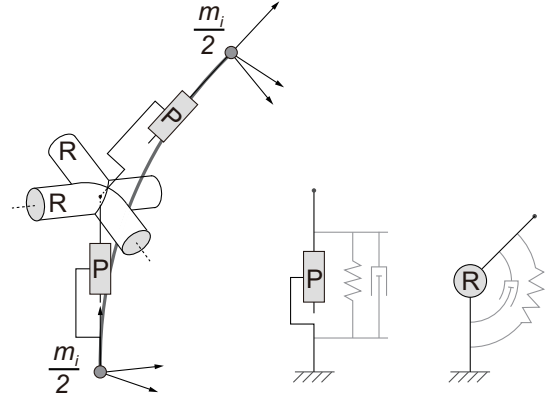


Fig. 4. An illustration of PUJ representation in 3D space. Two prismatic joints P and two revolute joints R are used to imitate the movement of the segment. The mass of a single segment m_i is evenly distributed to both ends of the segment, while the mass of other elements is regarded as zero.

analysis for rigid body robots now carries to soft robotic manipulators.

The general form of the dynamics for a n -joint rigid robot are captured by [33]:

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

where $\mathbf{q} \in \mathbb{R}^n$ is the state of the robot, $M(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the inertia matrix, $C(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times n}$ is the matrix capturing the Coriolis and centrifugal terms, $G(\mathbf{q}) \in \mathbb{R}^n$ is the gravity term, and $\boldsymbol{\tau} \in \mathbb{R}^n$ concatenates the torques applied to each joint.

There are several methods to obtain the rigid body dynamical model, such as Lagrange dynamics [18] and Newton-Euler method [40]. In our work, we invoke the Newton-Euler recursive method, which is computationally efficient and suitable for deriving the equivalent model of a soft robotic manipulator with many degrees of freedom. We derive the dynamics based on the screw theory. For the detailed derivation process, we refer the reader to the book by [36, Chapter 8].

An n -segment soft robotic manipulator is represented by n serial universal joints, each one comprising of two revolute joints and two prismatic joints, for a total of $4n$ joints. We

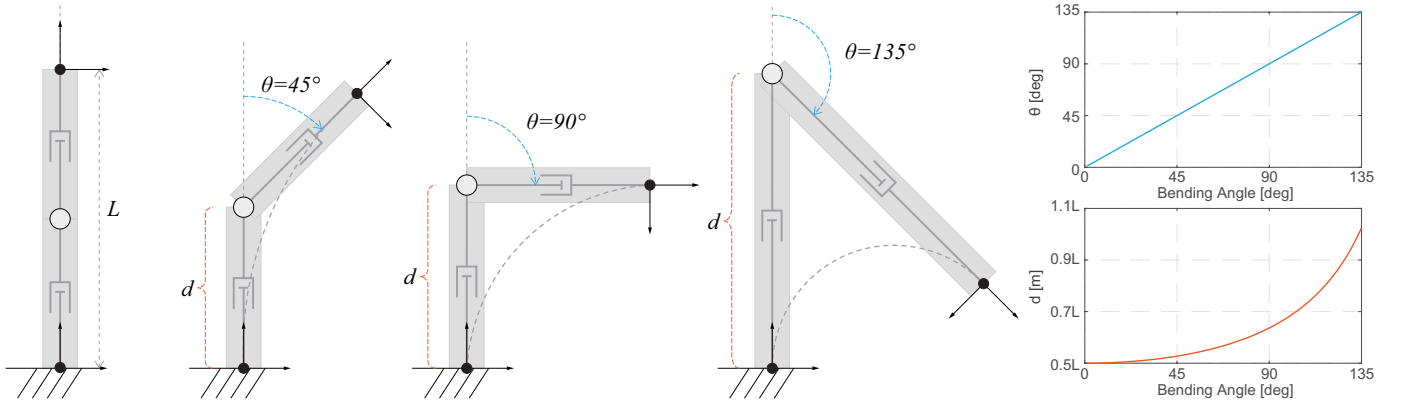


Fig. 5. An illustration of bending segment of the constant length alongside its equivalent PUI representation in the planar case.

further make a simplifying approximation assuming that the mass of the segment is evenly distributed at both ends as two equal point masses, as shown in Figure 4. This mass distribution ensures that the mass of the manipulator is always distributed on the central axis of the soft manipulator.

Soft robotic manipulators are usually made of materials that exhibit unignorable viscoelasticity. Consequently, the rigid body model fails to fully describe the motion characteristics of the soft manipulator. For this reason, we augment the model by adding an elastic term and a damping term to describe the soft manipulator as a rigid-link robot with joint viscoelasticity. We only consider first-order viscoelasticity in this paper; that is, elastic and damping is captured by two linear terms $K\mathbf{q}$ and $D\dot{\mathbf{q}}$, respectively. The complete dynamical equation is:

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + G(\mathbf{q}) + K\mathbf{q} + D\dot{\mathbf{q}} = \boldsymbol{\tau}, \quad (7)$$

where $\mathbf{q} := (d_1^1, \theta_1^1, \theta_2^1, d_2^1, \dots, d_1^n, \theta_1^n, \theta_2^n, d_2^n)^\top \in \mathbb{R}^{4n}$ is the state of the PUI model. $K \in \mathbb{R}^{4n \times 4n}$ is a diagonal stiffness matrix, $D \in \mathbb{R}^{4n \times 4n}$ is a diagonal damping matrix. The procedure to estimate these matrices is detailed in the next section.

The control schemes introduced later in this paper will depend on some properties of the dynamic model. These properties could be obtained from the basic characteristics of traditional rigid robots [33]. Some of these characteristics are essential for proving the stability of the proposed controllers. We will briefly introduce them here and will not repeat to derive them from scratch for completeness.

- **Property 1.** The inertia matrix $M(\mathbf{q})$ is a bounded symmetric positive-definite matrix, which is expressed as

$$\lambda_m I_{4n} \leq M(\mathbf{q}) \leq \lambda_M I_{4n},$$

where λ_m and λ_M are positive constants.

- **Property 2.** It is always possible to find a particular choice of $C(\mathbf{q}, \dot{\mathbf{q}})$ to make $\dot{M}(\mathbf{q}) - 2C(\mathbf{q}, \dot{\mathbf{q}})$ skew symmetric, which is expressed as

$$\mathbf{z}^\top \left(\dot{M}(\mathbf{q}) - 2C(\mathbf{q}, \dot{\mathbf{q}}) \right) \mathbf{z} = 0,$$

where \mathbf{z} is an arbitrary vector in \mathbb{R}^{4n} .

- **Property 3.** The Coriolis and centrifugal matrix $C(\mathbf{q}, \dot{\mathbf{q}})$ is bounded, which is expressed as

$$\|C(\mathbf{q}, \dot{\mathbf{q}})\| \leq c_0 \|\dot{\mathbf{q}}\|,$$

where c_0 is a bounded constant.

- **Property 4.** The gravity force/torque vector $G(\mathbf{q})$ is bounded, which is expressed as

$$\|G(\mathbf{q})\| \leq g_0,$$

where g_0 is a bounded constant.

D. State-space Reduction

In the PUI model, the two links of the same universal joint are physically the same segment of the soft manipulator, and we can not actively control the independent motion of the two prismatic joints. The difference in joint length is the result of external forces, as proved in **Lemma 2**. At the same time, the prismatic joint does not directly represent the elongation of the soft robot. For example, Figure 5 shows a bending segment of the constant length alongside its equivalent PUI representation in the planar case. We can see that as the bending angle increases, the length of the PUI prismatic joints increases. When approaching 180° , it becomes infinite. The length of this segment has not changed during the whole process. Therefore, we propose a method to reduce the dimensions of $4n$ joints of n -segment soft robot into $2n$ joints and $2n$ parameters in the PUI model. The revolute joint variables are reserved, and the prismatic joint variables are only used as parameters. We show that by adopting this state-space reduction method, the model accuracy is guaranteed while the computational cost is reduced. The reduced dynamical equation modified from (7) is:

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

where $\mathbf{q} := (\theta_1^1, \theta_2^1, \dots, \theta_1^n, \theta_2^n)^\top \in \mathbb{R}^{2n}$ is the reduced state vector concatenates all revolute joint variables of the PUI model. $\mathbf{d} := (d_1^1, d_2^1, \dots, d_1^n, d_2^n)^\top \in \mathbb{R}^{2n}$ is a parameter vector concatenates all prismatic joint variables. The procedure to estimate these vectors is detailed in the Section II-B2. To simplify the notations, the parameter vector \mathbf{d} is dropped in the following development.

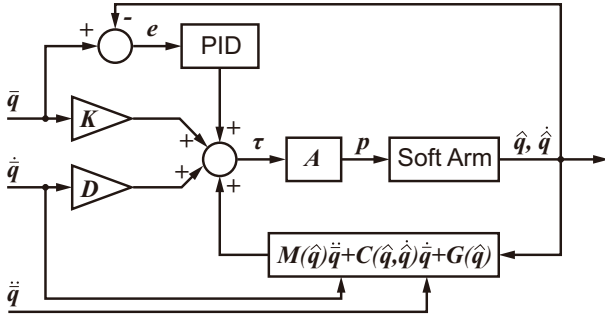


Fig. 6. Block diagram of the proposed trajectory tracking controller in the configuration space. K and D are feed-forward terms. M , C , and G contain model information. A is a conversion mapping between the torque of the PUA model and the air pressure of the soft arm. The state of the soft arm is estimated using a motion capture system.

III. CONTROL

In this section, we introduce two closed-loop controllers. The first targets trajectory tracking in the configuration space, while the second serves to control the motion in the task space.

A. Configuration Space Controller

The vector $\mathbf{q} = (\theta_1^1, \theta_2^1, \dots, \theta_1^n, \theta_2^n)^\top \in \mathbb{R}^{2n}$ is defined as the state in the configuration space; vectors $\dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^{2n}$ respectively correspond to angular velocity and angular acceleration. Given a reference trajectory $\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}, \ddot{\bar{\mathbf{q}}} \in \mathbb{R}^{2n}$, the closed-loop dynamic tracking controller in the configuration space takes the form $\mathbf{u} = f_1(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}, \ddot{\bar{\mathbf{q}}}, \hat{\mathbf{q}}, \dot{\hat{\mathbf{q}}})$, in which $\hat{\mathbf{q}}, \dot{\hat{\mathbf{q}}}$ are estimates of the state and its derivative obtained by a state estimation module.

The control scheme for trajectory tracking in the configuration space are given by:

$$\begin{aligned} \tau = & K_p \mathbf{e} + K_d \dot{\mathbf{e}} + K_i \int \mathbf{e} \\ & + M(\hat{\mathbf{q}})\ddot{\bar{\mathbf{q}}} + C(\hat{\mathbf{q}}, \dot{\hat{\mathbf{q}}})\dot{\bar{\mathbf{q}}} + G(\hat{\mathbf{q}}) + K\bar{\mathbf{q}} + D\dot{\bar{\mathbf{q}}}, \end{aligned} \quad (9)$$

where $\mathbf{e} := \bar{\mathbf{q}} - \hat{\mathbf{q}}$ is the tracking error, and K_p, K_d, K_i are feedback proportional–integral–derivative (PID) gains. The gains are tuned through experiments and set equal for all eight joints of the robot. The proposed model-based closed-loop controller contains a mixture of feedback and feed-forward terms: the terms $K\bar{\mathbf{q}} + D\dot{\bar{\mathbf{q}}}$ are feed-forward, $G(\hat{\mathbf{q}})$ is a feedback term for compensating gravity, while the other terms contain both feed-forward and feedback components. The integral term is used to eliminate the influence of modeling errors. The block diagram of the configuration space controller is shown in Figure 6.

To prove stability, we first consider a relatively ideal situation: error in the dynamics model, uncertainty in the state estimation from the sensor information, and external force are neglected. At this time, we omit the integral term and use a PD controller

$$\begin{aligned} \tau = & K_p \mathbf{e} + K_d \dot{\mathbf{e}} \\ & + M(\mathbf{q})\ddot{\bar{\mathbf{q}}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\bar{\mathbf{q}}} + G(\mathbf{q}) + K\bar{\mathbf{q}} + D\dot{\bar{\mathbf{q}}}. \end{aligned}$$

By substituting (8), the resulting form of the closed-loop dynamic equation is

$$M(\mathbf{q})\ddot{\mathbf{e}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{e}} + \tilde{D}\dot{\mathbf{e}} + \tilde{K}\mathbf{e} = 0, \quad (10)$$

where $\tilde{D} = D + K_d$, and $\tilde{K} = K + K_p$. Consider the positive-definite Lyapunov candidate extended from the energy function:

$$V = \frac{1}{2}\dot{\mathbf{e}}^\top M(\mathbf{q})\dot{\mathbf{e}} + \frac{1}{2}\mathbf{e}^\top \tilde{K}\mathbf{e}. \quad (11)$$

Then, the derivative of V is

$$\dot{V} = \frac{1}{2}\dot{\mathbf{e}}^\top \dot{M}(\mathbf{q})\dot{\mathbf{e}} + \dot{\mathbf{e}}^\top M(\mathbf{q})\ddot{\mathbf{e}} + \dot{\mathbf{e}}^\top \tilde{K}\mathbf{e}.$$

By substituting (10), we obtain

$$\dot{V} = \frac{1}{2}\dot{\mathbf{e}}^\top \left(\dot{M}(\mathbf{q}) - 2C(\mathbf{q}, \dot{\mathbf{q}}) \right) \dot{\mathbf{e}} - \dot{\mathbf{e}}^\top \tilde{D}\dot{\mathbf{e}}. \quad (12)$$

Considering Property 2, we have

$$\dot{\mathbf{e}}^\top \left(\dot{M}(\mathbf{q}) - 2C(\mathbf{q}, \dot{\mathbf{q}}) \right) \dot{\mathbf{e}} = 0,$$

thus we have

$$\dot{V} = -\dot{\mathbf{e}}^\top \tilde{D}\dot{\mathbf{e}} \leq 0, \quad (13)$$

under the consideration of hypotheses that $\tilde{D} \succ 0$.

However, since the closed-loop system is a time-varying system, (13) does not say anything about the convergence of \mathbf{e} to zero as $t \rightarrow \infty$. To invoke Barbalat's lemma to the proof of the stability for time-varying systems which requires high-order derivatives of Lyapunov candidate remain bounded [41], we further look into \ddot{V} , which is

$$\ddot{V} = 2\dot{\mathbf{e}}^\top \tilde{D}\dot{M}(\mathbf{q})^{-1} \left(C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{e}} + \tilde{D}\dot{\mathbf{e}} + \tilde{K}\mathbf{e} \right), \quad (14)$$

where $\|\ddot{V}\|$ is bounded because $\dot{\mathbf{e}}, \tilde{D}, M, C$, and \tilde{K} are bounded (Property 1,3,4). Thus the closed-loop system is globally asymptotically stable.

We further prove that the proposed PD controller can handle the identification errors of stiffness and damping matrix. Considering model uncertainty in K and D , the control law is expressed as

$$\begin{aligned} \tau = & K_p \mathbf{e} + K_d \dot{\mathbf{e}} \\ & + M(\mathbf{q})\ddot{\bar{\mathbf{q}}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\bar{\mathbf{q}}} + G(\mathbf{q}) + \hat{K}\bar{\mathbf{q}} + \hat{D}\dot{\bar{\mathbf{q}}}, \end{aligned}$$

where identified stiffness matrix is $\hat{K} = K + \delta K$, identified damping matrix is $\hat{D} = D + \delta D$. $\delta K, \delta D$ represent the uncertainty of parameters. In this case, the closed-loop system dynamics becomes

$$M(\mathbf{q})\ddot{\mathbf{e}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{e}} + \tilde{D}\dot{\mathbf{e}} + \tilde{K}\mathbf{e} + \delta K\bar{\mathbf{q}} + \delta D\dot{\bar{\mathbf{q}}} = 0.$$

The uncertainty of parameters only increases the feed-forward error. In this case, we can prove the stability with contraction theory [42] as presented by [26].

When there is a fixed external disturbance f_{dis} . The dynamic model of the system is expressed as

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + G(\mathbf{q}) + K\mathbf{q} + D\dot{\mathbf{q}} = \tau + J^\top f_{dis}. \quad (15)$$

In this case, the closed-loop system dynamics by substituting (9) and (15) is

$$M(\mathbf{q})\ddot{\mathbf{e}} + (C(\mathbf{q}, \dot{\mathbf{q}}) + \tilde{D})\dot{\mathbf{e}} + \tilde{K}\mathbf{e} + K_i \int \mathbf{e} + J^\top f_{dis} = 0. \quad (16)$$

This nonlinear system is linearized at the equilibrium point \mathbf{q}_{eq} and result in the following higher-order error dynamics

$$\ddot{\mathbf{e}} + M(\mathbf{q}_{eq})^{-1} \left((C(\mathbf{q}_{eq}, \dot{\mathbf{q}}_{eq}) + \tilde{D})\ddot{\mathbf{e}} + \tilde{K}\dot{\mathbf{e}} + K_I\mathbf{e} \right) = 0, \quad (17)$$

$$K_I\mathbf{e}_{eq} = 0. \quad (18)$$

According to linear system theory, the error dynamics are exponentially stable by a suitable choice of the gain matrices of the PID controller. If the disturbance is not fixed but bounded, then the error will also be bounded, according to the characteristics of the BIBO system.

Calculating the matrix mappings M , C , and G in (9) involves solving the inverse dynamics via the Newton-Euler recursive method (Algorithm 1) [36], which in our case also incorporates the elastic and damping terms (step 8). The method consists of two iterative processes, namely forward and backward iterations. The forward iteration calculates the position (step 2), velocity (step 3), and acceleration (step 4) of each link from 1 to n , and the backward iteration calculates the force/torque on each link (step 7), and the force/torque for each joint (step 8), from joint n to 1. In Algorithm 1, \mathcal{A}_i is the screw axis for joint i expressed in the i -th link frame, $H_{i,i-1}$ is the home pose of the i -th link expressed in the $(i-1)$ -th link frame. Since we regard the prismatic joint variables in the PUJ model as parameters, $H_{i,i-1}$ is a function of the prismatic joint length d_i . \mathcal{G}_i is the spatial inertia matrix of link i . The twist and wrench are represented by \mathcal{V}_i , \mathcal{F}_i , respectively, while Ad denotes the adjoint representation of a homogeneous transformation matrix, and ad denotes the Lie bracket of a twist. We refer the reader to [36, Chapter 8] for more details. After computing $\boldsymbol{\tau}$ using Algorithm 1, we proceed to compute the matrices M , C , and G ; we illustrate for M , the other cases being analogous. The matrix is built column by column by calling Algorithm 1 $2n$ times. In the i -th call, $\dot{\mathbf{q}}$, K , D , and the contribution of gravity are set to 0, while $\ddot{\mathbf{q}}$ is a vector with all entries equal to zero except for the i -th entry equal to 1 [36]. The mapping relationship from $\boldsymbol{\tau}$ to soft arm actuation is manipulator dependent, and we will introduce the mapping along with the hardware in the experimental setup section.

Algorithm 1 Newton-Euler Inverse Dynamics

Input: $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{d}$

Output: $\boldsymbol{\tau}$

```

1: for  $i = 1$  to  $n$  do
2:    $T_{i,i-1} = e^{-[A_i]q_i} H_{i,i-1}(d_i)$ 
3:    $\mathcal{V}_i = \text{Ad}_{T_{i,i-1}}(\mathcal{V}_{i-1}) + \mathcal{A}_i \dot{q}_i$ 
4:    $\dot{\mathcal{V}}_i = \text{Ad}_{T_{i,i-1}}(\dot{\mathcal{V}}_{i-1}) + \text{ad}_{\mathcal{V}_i}(\mathcal{A}_i) \dot{q}_i + \mathcal{A}_i \ddot{q}_i$ 
5: end for
6: for  $i = n$  down to 1 do
7:    $\mathcal{F}_i = \text{Ad}_{T_{i+1,i}}^\top(\mathcal{F}_{i+1}) + \mathcal{G}_i \dot{\mathcal{V}}_i - \text{ad}_{\mathcal{V}_i}^\top(\mathcal{G}_i \mathcal{V}_i)$ 
8:    $\tau_i = \mathcal{F}_i^\top \mathcal{A}_i + K_i q_i + D_i \dot{q}_i$ 
9: end for
10: return  $\boldsymbol{\tau}$ 
  
```

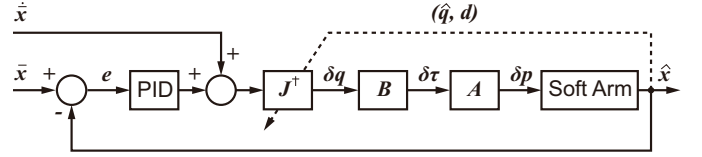


Fig. 7. Block diagram of the proposed kinematic controller in the task space. J^\dagger is the Moore-Penrose pseudo-inverse of Jacobian. A is a conversion mapping between the torque of the PUJ model and the air pressure of the soft arm. The pose of the soft arm in task space is obtained using a motion capture system.

B. Task Space Controller

Motion control in the task space is indispensable for soft manipulators to effectively perform tasks. Given a reference trajectory $\bar{\mathbf{x}}, \dot{\bar{\mathbf{x}}} \in \mathbb{R}^6$, the closed-loop controller in the task space takes the form $\mathbf{u} = f_2(\bar{\mathbf{x}}, \dot{\bar{\mathbf{x}}}, \hat{\mathbf{x}})$, where $\hat{\mathbf{x}} \in \mathbb{R}^6$ is the pose of the end effector as measured from sensors. We propose a motion controller in the task space, as follows:

$$\dot{\mathbf{q}} = J^\dagger \left(K_P \mathbf{e}^* + K_D \dot{\mathbf{e}}^* + K_I \int \mathbf{e}^* + \dot{\bar{\mathbf{x}}} \right), \quad (19)$$

$$\delta \boldsymbol{\tau} = B \dot{\mathbf{q}} \delta t,$$

where J^\dagger denotes the Moore-Penrose pseudo-inverse of Jacobian and $\mathbf{e}^* := \bar{\mathbf{x}} - \hat{\mathbf{x}}$. B is a mapping matrix measured from experiments. The block diagram of the proposed task space controller is shown in Figure 7. The differential kinematics of the soft arm is given by

$$\dot{\hat{\mathbf{x}}} = J(\hat{\mathbf{q}}, \mathbf{d}) \dot{\mathbf{q}}, \quad (20)$$

where \mathbf{d} is model parameters. Substituting (19) and (20), we have the following closed-loop equation:

$$K_P \mathbf{e}^* + (K_D + I) \dot{\mathbf{e}}^* + K_I \int \mathbf{e}^* = 0.$$

By differentiation, we can get the second-order error dynamics

$$(K_D + I) \ddot{\mathbf{e}}^* + K_P \dot{\mathbf{e}}^* + K_I \mathbf{e}^* = 0. \quad (21)$$

According to linear system theory, the error dynamic is exponentially stable by a suitable choice of the gain matrices K_P , K_D , and K_I .

IV. EXPERIMENTS

This section verifies the model and controller proposed in this paper. We first describe the soft manipulator and control system and analyze the mapping between the pneumatic pressure and torque. We introduce a mechanism for system identification, i.e., for learning the parameters in (8). After that, we verified the PUJ-based kinematics model, dynamical model, joint space controller, and task space controller proposed in the paper through a series of experiments.

A. Experimental setup

1) *Hardware platform:* We use a soft robotic manipulator previously presented by [20] to validate the proposed models and controllers (see Figure 8). The design, fabrication, and simulation optimization methods of the soft Honeycomb

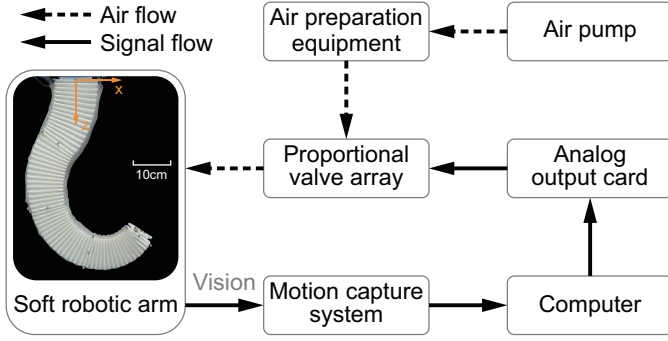


Fig. 8. The soft robotic manipulator and accompanying hardware system for validating the proposed models.

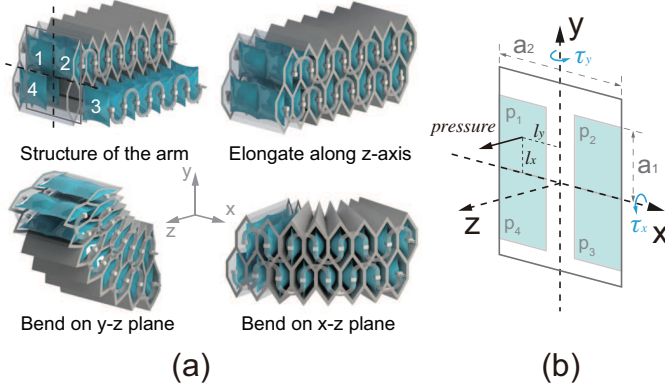


Fig. 9. Structure of the pneumatically actuated soft robotic manipulator. (a) Design and 3D deformation of a segment of the manipulator. (b) The cross-section of the manipulator and the schematic diagram of the airbag distribution.

Pneumatic Network Arm are detailed in [43]. The manipulator consists of four segments and is actuated by 16 proportional valves. We place reflective markers on both ends of each segment of the manipulator and use MCS (Motion Capture System, Prime 13, OptiTrack) to obtain the pose of the markers. The manipulator is 66cm long and is capable of flexible motion in 3D space.

2) *Torque To Pressure Mapping*: To apply (7) to the pneumatic soft robotic manipulator, it is imperative to devise a conversion mapping between the air pressure of the manipulator and the torque of the equivalent model. The structure of the manipulator is shown in Figure 9(a). Four groups of airbags are placed side by side in the honeycomb structure. The air pressures of the four airbags are denoted by p_1, p_2, p_3, p_4 . We adopt the assumption $p_1 + p_4 = p_2 + p_3$, which corresponds to reducing the internal consumption pertaining to the driving forces. We apply the following decomposition:

$$\begin{bmatrix} p_1 & p_2 \\ p_3 & p_4 \end{bmatrix} = \begin{bmatrix} p_x & p_x \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} p_y & 0 \\ p_y & 0 \end{bmatrix}, \quad (22)$$

where p_x, p_y respectively correspond to the air pressure components that cause the bending around the x and y axes. We relate the air pressure components with the produced force and torque using the formulas $F = ps$, and $\tau = Fl$, in which s represents area and l length:

$$\tau_x = 2p_x s l_x, \quad \tau_y = 2p_y s l_y, \quad (23)$$

where s is the average contact area between an airbag and the honeycomb structure and $l_x = \frac{a_1}{2}$, $l_y = \frac{a_2 - a_1}{2}$ are the average distances that generate torque around the x and y axes, as shown in Figure 9(b). Table II lists the design parameters of the four-segment soft robotic manipulator. Note that the

TABLE II
DESIGN PARAMETERS OF THE SOFT ROBOTIC MANIPULATOR

Segment	Length [cm]	Mass [kg]	Structure a_2 [cm]	Airbag a_1 [cm]
1 (base)	18.5	1.2	11	4.5
2	17.5	0.8	10	4.0
3	15.5	0.6	9	3.5
4 (tip)	14.5	0.4	8	3.0

contact area between the airbag and the honeycomb structure may vary with pressure. To simplify, we assume that there is a linear mapping between air pressure components and torque

$$\tau_x = A p_{x,y}, \quad (24)$$

where A is a conversion mapping between the torque of the PUJ model and the air pressure of the soft manipulator, which is obtained through an identification process explained in the next section.

3) *System Identification*: We describe how to estimate parameters K, D in (8) and A in (24). The other matrices M, C , and G are related to the mass and geometric length of the equivalent rigid links, calculated by measuring the mass and the initial length of the manipulator (as listed in Table II). For the detailed recursive calculation process, we refer the reader to [36, Section 8.3]. The stiffness coefficient K and the damping coefficient D are obtained from experimentally collected data using linear regression. More-Penrose pseudo-inversion is most often used to solve least square problems [44]. By substituting (8) and (24) we can write the system dynamics as $K\mathbf{q} + D\dot{\mathbf{q}} - A\mathbf{p} = -M(\mathbf{q})\ddot{\mathbf{q}} - C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - G(\mathbf{q})$, where the left side contains the terms to be identified and the right side are computed from data, along with the previously identified matrices M, C, G .

To improve the accuracy of identification, we implement two data collection processes. In the first process, we raise the tip of the soft manipulator, then let the manipulator swing freely under gravity ($\tau = 0$). The data is only used for the identification of the stiffness matrix K and the damping matrix D , which is

$$K\mathbf{q} + D\dot{\mathbf{q}} = -M(\mathbf{q})\ddot{\mathbf{q}} - C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - G(\mathbf{q}). \quad (25)$$

The data were collected at a frequency of 30Hz. The values of $\mathbf{q}, \dot{\mathbf{q}}$, and $\ddot{\mathbf{q}}$ need to be measured for identification. We use the state estimation method introduced in Section II-B2 to obtain $\mathbf{q} = (\theta_1^1, \theta_2^1, \dots, \theta_1^n, \theta_2^n)^T$ from the manipulator postures. Based on the natural response in (8), we fit measurements \mathbf{q} to the function $ae^{-bt} \sin(\omega t - c)$, which allows for both denoising and robust computation of the derivatives $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$.

The second process is to slowly increase the air pressure from 0 to 2.5bar in 40s to maintain the quasi-static motion of the manipulator ($\dot{\mathbf{q}}, \ddot{\mathbf{q}} = 0$). The data together with the

identified stiffness matrix K are used for the identification the actuation matrix A , that is

$$Ap = Kq + G(q). \quad (26)$$

Given the structural property of the soft manipulator used in this paper, we identify the stiffness ($Nmrad^{-1}$), damping ($Nmsrad^{-1}$), and actuation ($Nmbar^{-1}$) across the x and y axes separately. The identification result gives:

$$K = \text{diag}(5.78, 7.36, 4.71, 5.98, 3.81, 4.21, 2.86, 2.99)$$

$$D = \text{diag}(0.38, 0.38, 0.26, 0.31, 0.22, 0.25, 0.08, 0.05)$$

$$A = \text{diag}(2.52, 4.30, 1.53, 2.62, 1.33, 1.75, 0.76, 1.11).$$

4) *Simulation*: In addition to the above soft manipulator platform, we also built a simulation environment to verify the proposed model. We use the model developed for large deformation flexible rods in [45] to simulate the robot. A manipulator with four segments are simulated. Each segment is divided into three small segments of equal length. The kinematics and dynamics of each small segment are calculated using two equal-length links connected by two mutually perpendicular revolute joints. Similar methods are widely used in soft robot simulations [26], [27], [46]. The mass, length, stiffness, and damping properties of the manipulator in the simulation are the same as the real manipulator properties measured above. The simulation environment is implemented in *MuJoCo* [47]. The control signal is transmitted, and the motion data of the simulated manipulator is collected through the interface between *MuJoCo* and *Matlab*.

B. Evaluation of the model

The objective of the following experiments is to validate the modeling accuracy of the proposed PUJ kinematic and dynamical models. We conducted the following simulations and experiments.

1) *Kinematic modeling accuracy*: We implement simulations to show the modeling accuracy of the PUJ model and the PCC model in scenarios that involve dynamical movements and interaction with the environment. The first simulation demonstrates the effect of dynamic motions of the manipulator on the model accuracy. Gravity and external load are set to 0, and a step excitation is given to the manipulator to generate dynamic motion. Then we perform the same step excitation with gravity and 500g load at the tip as a second simulation. The last simulation is designed to study the effect of interaction with the environment on the model accuracy. A fixed obstacle is placed in the workspace of the manipulator. We gradually increase the actuation of each segment to deform the manipulator and interact with the obstacle. During simulations, the homogeneous transformations are obtained as the posture of each segment of the manipulator. The postures are then converted to the configuration variables (ϕ, ψ, l) of the PCC model [15] and the configuration variables/model parameters ($\theta_1, \theta_2 / (d_1, d_2)$) of the PUJ model. After calculating the configuration variables, the forward kinematics of the PUJ model are calculated using (5). The position errors are calculated by comparing the calculated forward kinematics with the posture of the end effector in simulations. As shown

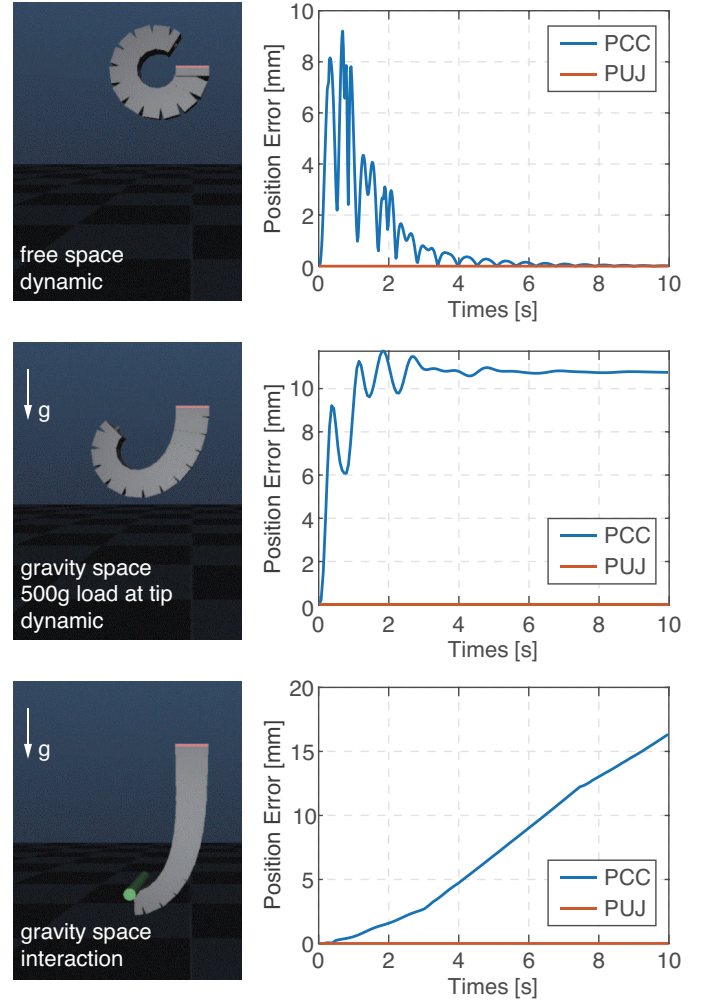


Fig. 10. Comparison of kinematic modeling accuracy between piecewise constant curvature (PCC) model and piecewise universal joint (PUJ) model in simulations. The position errors that (a) give a step excitation to the manipulator in free space without gravity, (b) give a step excitation to the manipulator in gravity space with 500g load at the tip, and (c) gradually increase the actuation of the manipulator in gravity space and interact with a fixed obstacle.

in Figure 10 (a), (b), and (c), no matter there is a dynamic motion or interaction with the environment, the PCC will exhibit modeling errors. In contrast, the PUJ model can always correctly describe the kinematics of the soft manipulator. The oscillation in Figure 11 (a) and (b) is because giving a step excitation will make the arm oscillate near the equilibrium position. The acceleration of the movement will change, and the error of the PCC also changes with time. We do not show the orientation error in simulations because the bending angle variables in the PCC model are calculated using the perceived orientation information [27], which is theoretically equal to zero as the PUJ model.

For the experiments on the real platform, a 500g weight was added to the tip of the manipulator. We gradually increased the air pressure from 0 (zero position) to 2.5bar to bend all sections of the manipulator on the $x = 0$ plane. We use the MCS to obtain the posture of the markers on the manipulator. Similar to simulations, the measured postures

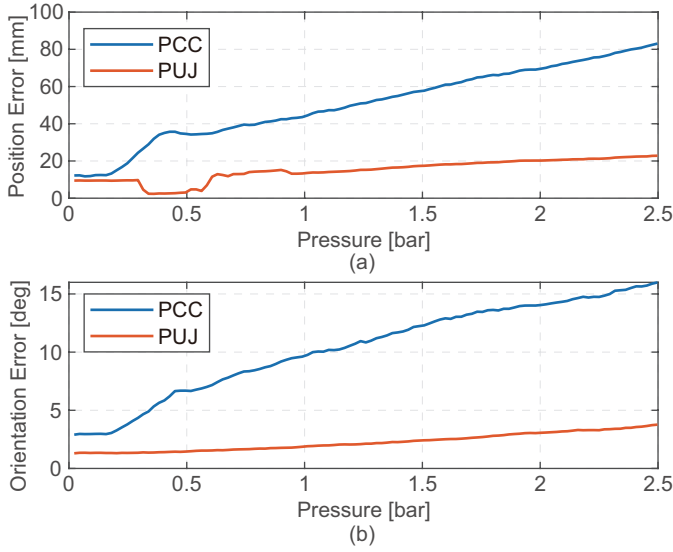


Fig. 11. Comparison of kinematic modeling accuracy between piecewise constant curvature (PCC) model and piecewise universal joint (PUJ) model on the real manipulator. The position (c) and orientation (d) errors of the experiment.

are then converted to the configuration variables of the PCC model and the PUJ model. The position and orientation errors are calculated by comparing the calculated forward kinematics with the posture of the end effector obtained by the MCS. The results are illustrated in Figure 11 (c) and (d). The PUJ model performs globally better than the PCC model. As the deformation of the manipulator increases, the gravitational moment of the load continues to increase, whence the deviation of the PCC model also increases. The maximum position/orientation error of the PCC model is 3.7/4.6 times higher than the PUJ model. It is noteworthy that the error of our proposed PUJ model does not scale with pressure: this is a positive attribute, especially when the soft manipulator is used in scenes where external forces can not be ignored, such as daily interactive tasks. The experiment is carried out on the $x = 0$ plane and does not involve the axial torsion of the manipulator. The error of PUJ mainly comes from the pasting of the markers on the manipulator. When the acceleration of the motion of the manipulator is not negligible, there will be interaction forces between adjacent segments, which will also affect the accuracy of the PCC model.

2) *Dynamic response*: Experiments are carried out to verify the dynamical modeling accuracy of both active and passive responses of the soft manipulator. The numerical simulation of forward dynamics based on (8) is implemented in Matlab based on the package provided by [36].

The first dynamic experiment is used to verify the validity of the proposed model and the accuracy of system identification. The actuation pressure in this experiment is equal to zero, and our PUJ model is used to predict the passive response of the manipulator. In the experiment, we raised the manipulator in the $x = 0$ plane and recorded the configuration variables at $t = 0s$ as the initial state of the PUJ model for the numerical evaluation. We then let the manipulator swing freely

under gravity, record its motion, and identify the configuration variables explained in the previous section. We carried out two experiments: a) without load and b) under $500g$ load at the end effector. The experimental results in Figure 12 show that the proposed PUJ model exhibits high accuracy. The average joint angle errors are 0.006 rad and 0.015 rad for the experiments without and with load, respectively. Note that the proposed PUJ model can accurately capture the characteristics of the sinusoidal decay motion of each segment of the manipulator (Figure 12(a)) as well as predict the motion characteristics of the manipulator well when the tip is loaded (Figure 12(b)).

The second dynamic experiment is used to verify the validity of the proposed model for excitation with a step response. We conducted two experiments (as shown in Figure 12): on-plane motion (all segments bend in the $x = 0$ plane) and out-of-plane motion (the first two segments bend on the $y = 0$ plane and the last two on the $x = 0$ plane) without load with a $500g$ load on the tip. Initially, the manipulator is at the zero position. At $t = 0s$, certain segments of the manipulator are provided with $2.5bar$ pressure step inputs as excitation to generate on-plane/out-of-plane motion, and the pressure is maintained throughout the experiments. The responses of the real soft manipulator and the model-based numerical simulated responses are recorded for all three experiments.

For on-plane motion, the average joint error is $0.012rad$, and the maximum joint steady-state error is $0.027rad$. For out-of-plane motion with load, the average joint error is $0.017rad$, and the maximum joint steady-state error is $0.062rad$. In addition, for these test scenarios, our model predicts a more oscillatory behavior than the manipulator demonstrates, which may be explained in view of the fact that the elasticity and damping of the manipulator are identified without actuation. The air pressure can not change instantaneously but slowly increases when actuating the manipulator. This will result in greater damping observed in the manipulator.

C. Evaluation of the controller

The objective of the following experiments is to validate the effectiveness of the proposed configuration space controller and task space controller based on the PUJ model. We perform two groups of experiments pertaining to trajectory tracking in the configuration space and motion control in the task space.

1) *Configuration space controller*: For configuration space controller, we start by testing the effectiveness of the proposed configuration space controller for tracking sinusoidal reference trajectories of different frequencies and amplitudes:

$$\bar{q}_i(t) = \begin{cases} 0.3 \sin\left(\frac{2\pi}{15}t\right), & i \in \{1, 3, 5, 7\} \\ 0.1, & i \in \{2, 4, 6, 8\} \end{cases},$$

$$\bar{q}_i(t) = \begin{cases} 0.1, & i \in \{2, 4, 6, 8\} \\ 0.3 \sin\left(\frac{2\pi}{15}t\right), & i \in \{1, 3, 5, 7\} \end{cases},$$

$$\bar{q}_i(t) = \begin{cases} 0.3 \cos\left(\frac{\pi}{8}t\right), & i \in \{1, 3, 5, 7\} \\ 0.3 \sin\left(\frac{\pi}{8}t\right), & i \in \{2, 4, 6, 8\} \end{cases},$$

$$\bar{q}_i(t) = \begin{cases} 0.3 \sin\left(\frac{\pi}{8}t\right), & i \in \{1, 3, 5, 7\} \\ 0.2 \sin\left(\frac{\pi}{4}t\right), & i \in \{2, 4, 6, 8\} \end{cases}.$$

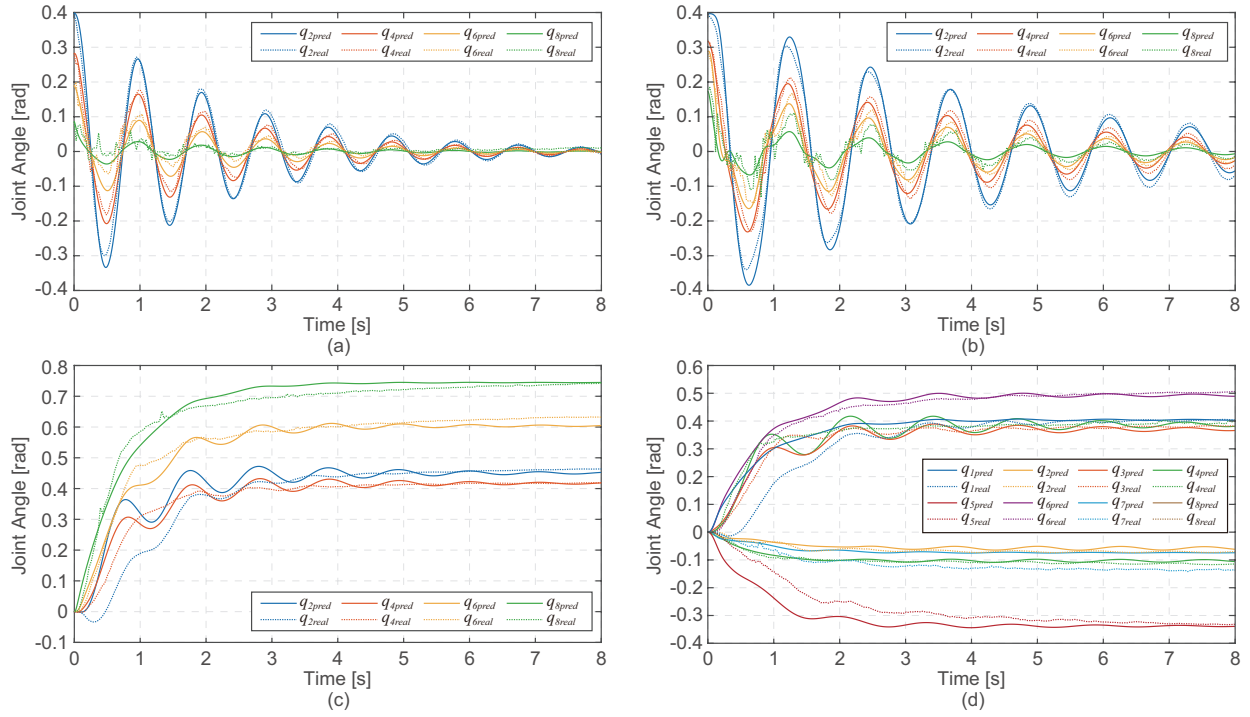


Fig. 12. Experimental results of dynamical model verification. The dynamic response of the manipulator swinging freely under gravity (a) without load on the tip and (b) with a 500g load on the tip. The dynamic step response of (c) in-plane motion and (d) out-of-plane motion with 500g load on the tip. Solid lines illustrate model predictions and dashed lines real.

Where $i \in \{1, 3, 5, 7\}$ represent the first revolute joint of each segment of a four-segment PUJ model and $i \in \{2, 4, 6, 8\}$ represent the second. For the first two trajectories, instead of commanding the manipulator to oscillate in the $x = 0$ or $y = 0$ plane, we set a fixed angle of 0.1rad for odd/even-numbered joints to force the manipulator to generate an out-of-plane motion to test the independent control ability of the controller to the two joints in each segment. The third trajectory will lead the manipulator to draw a circle in the task space. The last trajectory makes the tip of the manipulator move in an '8'-shape in the task space by setting different amplitudes and frequencies. The PID gains of the controller are adjusted based on experience and set to the same value in all experiments for all joints. The velocity and acceleration of the reference trajectory are calculated by the first and second-order difference of the position. The control frequency of all tracking experiments is 50Hz .

The controller can closely track different reference trajectories with a high-precision, as shown in Figure 13. The controller can maintain independent control of the two joints in each manipulator; that is, some joints oscillate greatly, and the other joints maintain a fixed angle, as shown in Figure 13(a) and (b). When at the initial position, each joint of the model is 0. It can quickly converge to the reference trajectory under the action of the controller and keep the tracking of trajectory, Figure 13(c). The closer the joint to the tip of the manipulator (the joint number is larger), the less affected by unmodeled factors, the better the trajectory tracking accuracy. Note that it is possible to further improve the control accuracy by adjusting the PID of each joint separately.

To show the upper limit effectiveness of the configuration space controller, we gradually increase the frequency of the reference trajectory until it can not be tracked stably. The resulting trajectory is regarded as the upper limit of the capacity of the controller, as shown in Figure 14. The reference trajectory is:

$$\bar{q}_i(t) = \begin{cases} 0, & i \in \{1, 3, 5, 7\} \\ 0.3 \sin\left(\frac{12\pi}{5}t\right), & i \in \{2, 4, 6, 8\} \end{cases}.$$

The dynamical controller is able to utilize the natural softness (stiffness and damping) of the soft manipulator to achieve fast dynamical motion. Two stages are observed in this experiment: The first stage occurs in the first cycle, and the tip segment responds quickly and produces the largest amplitude. We call it the swing-up stage. In the second stage, the maximum swing angle of the tip segment decreases, and the maximum swing angle of the root segment increases, which is the main contribution to the fast motion of the manipulator. We call it the stable swing stage. In the stable swing stage, the fast motion generated by the manipulator can swing for half a period in 0.4s , and the maximum speed of the tip is observed to reach 7m/s . In the experiment, a stable delay between the actual motion of the manipulator and the reference trajectory is observed, which can be explained as our dynamic model does not consider the delay of the pneumatic actuation.

2) *Task space controller*: We evaluate the effectiveness of the task space motion controllers by a set of point-to-point experiments in 3D space. We randomly chose three targets in the workspace by inflating the manipulator and commanding the controller to drive the manipulator to reach the targets

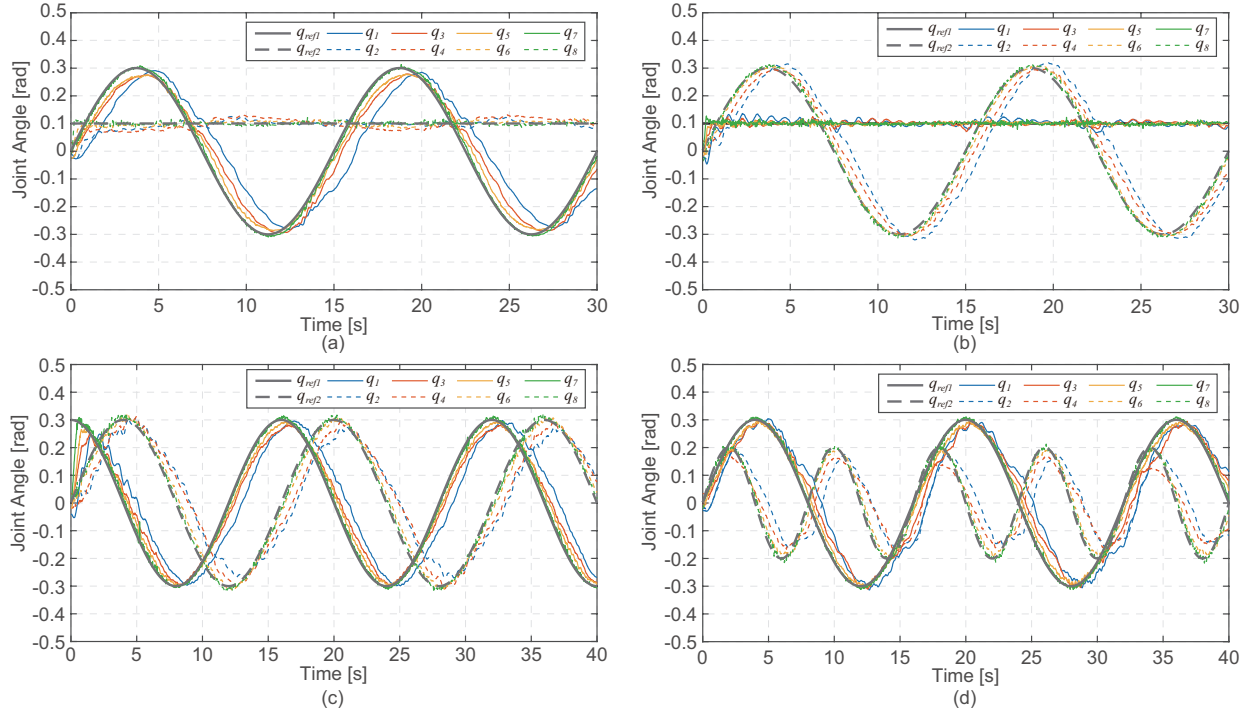


Fig. 13. Experimental results of configuration space controller tracking trajectories of different frequencies and amplitudes. (a) Odd-numbered joints track the sin signal, and even-numbered joints maintain a fixed angle of 0.1rad . (b) Even-numbered joints track the sin signal, and odd-numbered joints maintain a fixed angle of 0.1rad . (c) Odd-numbered joints and even-numbered joints track cos and sin signals, respectively, making the tip of the manipulator draw a circle in the task space. (d) The odd-numbered joints and even-numbered joints track the sin signal at the same time, and the frequency of the latter is doubled, making the tip of the manipulator draw '8'-shape in the task space.

from a fixed starting point (resting position). To demonstrate the effectiveness of the controllers under unknown interactions with the environment. Each target was repeatedly reached four times with a load of $0g$, $100g$, $200g$, and $500g$, respectively. The position and orientation deviations between the target and the tip of the manipulator are recorded during the experiments. Figure 15 presents the position and orientation error of the task space controller in regulating the end-effector. The controller is able to reach different targets in the 3D space under different unknown loads. Although external forces will change the process of regulating, the system can reach a similar accuracy after an oscillatory transient under the action of the controller.

We further demonstrate the capability of the proposed task space controller in trajectory tracking tasks. First, we artificially inflate the manipulator and let the tip of the manipulator draw a star in the task space, as shown in Figure 16. In this process, the pose of the tip is recorded at equal time intervals, and a total of 500 poses are recorded as the discretized reference trajectory. The manipulator was commanded to track the trajectory from an initial state (the top vertex of the star). The control frequency is 20Hz and the target pose is updated every control cycle and the velocity of the reference trajectory is given as $\dot{\mathbf{x}}_t = \frac{\mathbf{x}_t - \mathbf{x}_{t-1}}{\delta t}$. As with the above experiment, to demonstrate the controller's ability to handle unknown interactions, loads of $100g$, $200g$, and $500g$ were added to the tip, and the experiment was repeated four times. In the experiment, the trajectory of the end of the manipulator was recorded. Figure 16 shows the projection of the trajectory on

the x-y plane. The controller can achieve trajectory tracking in the task space and is robust to unknown loads. The outer side of the reference trajectory is about $0.6m$, which is close to the total length of the manipulator, which further verifies the large-range motion control ability of the controller in 3D space. As the load increases, the tracking effect gradually deteriorates, especially at the vertex of the star. This can be attributed to the fact that trajectory tracking is not quasi-static, and the speed direction of the reference trajectory has a sudden change at the vertex, which impacts the movement of the manipulator, making it impossible to reach the vertex completely.

V. DISCUSSION AND LIMITATION

In this paper, the presented method does not consider the active axial elongation deformation of the soft robot but can capture the passive elongation under external forces. At the same time, this modeling method is general and can be extended to the modeling of extensible soft robots. As introduced in the modeling part, the length of the links in the PUJ model is related to the elongation of soft segments and coupled with bending. Thus, developing a decoupling method for extensible soft robots is an interesting topic for future study.

The mass of the manipulator is distributed at both ends of the two links in the PUJ model, which guarantees to the greatest extent that the mass is always distributed on the central axis of the manipulator during the deformation. [26] Assumes the center of mass is in the middle of the chord. In their case, as the deformation of the manipulator increases,

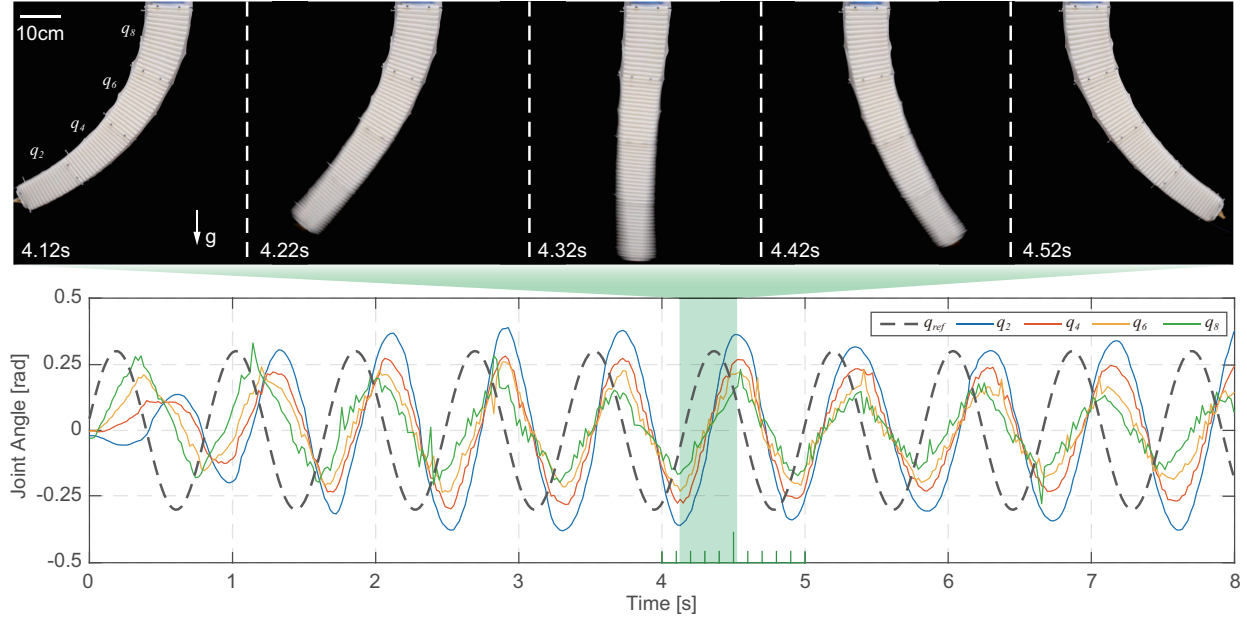


Fig. 14. Experimental results of fast dynamic motion in configuration space. The manipulator swings quickly in the $x = 0$ plane by giving a high-frequency reference signal. Snapshots of the manipulator movement from 4.12s to 4.52s are shown above. The manipulator can complete a half-cycle swing within 0.4s, resulting in a speed of 7m/s for the tip of the manipulator in the task space.

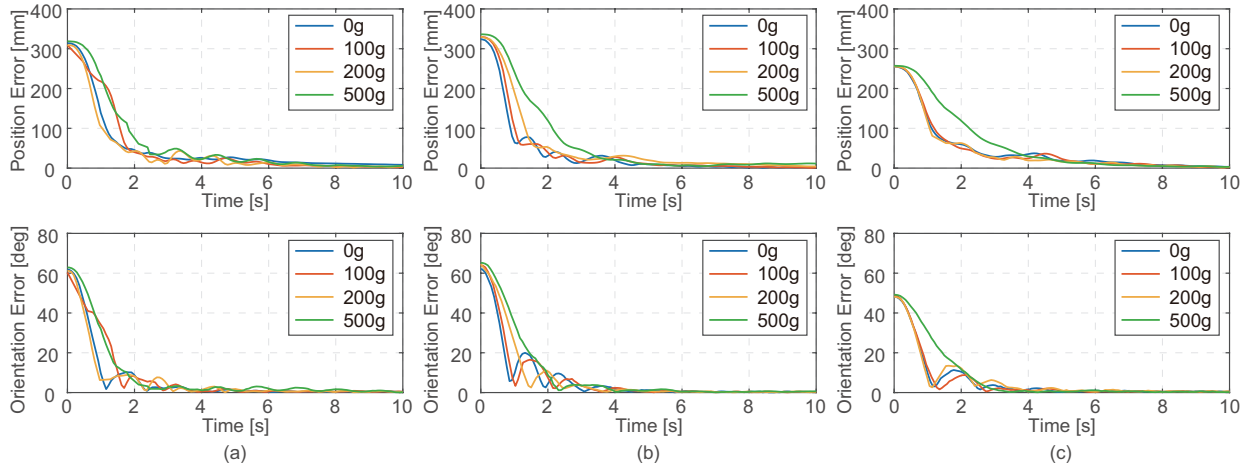


Fig. 15. Experimental results of task space controller in regulating the end-effector position and orientation to three randomly selected targets (a), (b), and (c). Each target was reached four times with a load of 0g, 100g, 200g, and 500g. The controller has a uniform performance in the task space and is robust to unknown external loads. The control frequency is 20Hz.

the deviation between the position of the center of mass in the model and the actual manipulator increases, which makes the dynamic response of the model deviate from reality. The mass distribution of the PUJ model also has a defect in the root segment. Since the root is fixed, the mass distribution is equivalent to neglecting the motion of the first half segment of the robot.

The descriptive ability of the model is related to the degree of freedom, which is defined as the number of independent variables needed to describe the motion. We can theoretically use six variables to completely describe the relative state of the root and tip of a soft robot in space. But in fact, we should be traded off the number of variables and descriptive ability of the model according to the design and application

scenarios of the soft robot. As proved in **Corollary 1**, the PUJ is a four-variable method that can describe the position and orientation of the end effector of an actuator that is deformed arbitrarily in a plane. The PCC has only three variables, and it is impossible to describe this bending from a theoretical point of view. Based on this corollary, we can also develop other four-variable methods to describe the state soft robotic manipulators to provide a theoretical foundation for them to interact with the environment.

Compared to rigid robots, soft robots do not have "joints" in common sense, and there are few sensors for joint-level sensing. Therefore, perception is generally done in task space, which is a double-edged sword. On the one hand, the cost of perception is greater, and feedback frequency is lower. On the

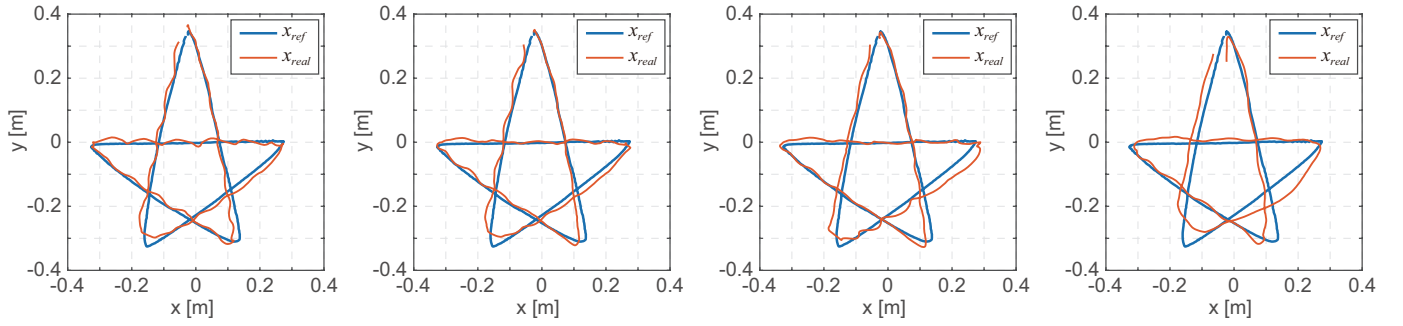


Fig. 16. Experimental results of task space controller tracking trajectories under different external loads. From left to right are the experimental results with a load of 0g, 100g, 200g, and 500g. The control frequency is 20Hz.

other hand, the perception of task space also provides more possibilities for modeling. For example, the deformation of a soft robot is a combination of actuation force and external force. By sensing the deformation in the task space, we can take account of the effect of external force to a certain extent without using force sensors. There is a coupling effect between position and orientation in the PCC model, which can be seen as a waste of perception information for soft robots that use task space sensors. The PUJ model proposed in this paper decouples position and orientation and describes more complex deformations.

It is intuitive to model and control the soft arm as a rigid robot. In this case, we have to consider what properties distinguish soft robots from rigid robots? In modeling, we need to take special considerations to such properties. On the one hand, this paper takes elasticity and viscosity into consideration. On the other hand, passive compliance is also considered by a set of passive model parameters. We can not accurately predict the complex deformations caused by the interaction between the passively compliant soft robot and the environment. Therefore, observing and updating the model online may be a good choice. In this paper, we divide the four parameters of PUJ into two active control degrees of freedom and two passive observation degrees of freedom based on this idea. We proved theoretically and experimentally that the PUJ could accurately describe the soft arm's kinematics under external force.

This paper uses the control methods developed for rigid robots to control the soft arm. Theoretically, a soft arm requires an infinite state space to be described, and a discrete, finite model is always an approximation. An open question remains unanswered: What kind of approximate model can meet the needs of control? This paper shows that by considering the soft robot's stiffness and damping, the soft robot's fastest dynamic control is achieved (the velocity of the end effector can reach 7m/s). In future work, we will further study the dynamic control of the soft arm in the task space to make full use of the dynamic interaction between the soft robot and the environment.

VI. CONCLUSION

In this paper, we propose a simple soft robot modeling method that can accurately describe the kinematics of the soft

robot when performing dynamic motions or interacting with the environment. We realized the dynamical modeling and control of a multi-segment soft manipulator in 3D space based on this method. This method equates the soft robot to a rigid-link robot, namely piecewise universal joint. Thus the well-established knowledge on control and analysis of rigid robots are transferred to soft robots. Compared with the constant curvature model commonly used in the field, this method is accurate yet simple, which relaxes the constraint that one segment of the manipulator must be deformed in a constant curvature manner and achieves a more general representation. We proposed two controllers for 3D multi-segment soft robotic manipulators based on the PUJ model: one for dynamical trajectory tracking in the configuration space and another for motion control in the task space. Our experiments support the viability of the modeling method in the context of dynamical control of a soft manipulator. To the best of our knowledge, the dynamical controller can achieve stable control of the highest motion speed (up to 7m/s) of the soft manipulator.

ACKNOWLEDGMENTS

The authors would like to thank Hao Jiang, Yusong Jin, and Yinghao Gan for fruitful discussions.

REFERENCES

- [1] M. H. Dickinson, C. T. Farley, R. J. Full, M. Koehl, R. Kram, and S. Lehman, "How animals move: an integrative view," *science*, vol. 288, no. 5463, pp. 100–106, 2000.
- [2] R. Pfeifer and J. Bongard, *How the body shapes the way we think: a new view of intelligence*. MIT press, 2006.
- [3] D. Trivedi, C. D. Rahn, W. M. Kier, and I. D. Walker, "Soft robotics: Biological inspiration, state of the art, and future research," *Applied bionics and biomechanics*, vol. 5, no. 3, pp. 99–117, 2008.
- [4] S. Kim, C. Laschi, and B. Trimmer, "Soft robotics: a bioinspired evolution in robotics," *Trends in biotechnology*, vol. 31, no. 5, pp. 287–294, 2013.
- [5] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, pp. 467–475, 2015.
- [6] E. W. Hawkes, C. Majidi, and M. T. Tolley, "Hard questions for soft robotics," *Science robotics*, vol. 6, no. 53, p. eabg6049, 2021.
- [7] T. George Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, "Control strategies for soft robotic manipulators: A survey," *Soft Robotics*, vol. 5, no. 2, pp. 149–163, 2018.
- [8] C. Armanini, C. Messer, A. T. Mathew, F. Boyer, C. Duriez, and F. Renda, "Soft robots modeling: a literature unwinding," *arXiv preprint arXiv:2112.03645*, 2021.
- [9] J. Spillmann and M. Teschner, "Corde: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects," in *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2007, pp. 63–72.
- [10] J. Till, C. E. Bryson, S. Chung, A. Orekhov, and D. C. Rucker, "Efficient computation of multiple coupled cosserat rod models for real-time simulation and control of parallel continuum manipulators," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5067–5074.
- [11] J. Till, V. Aloï, 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.
- [12] S. H. Sadati *et al.*, "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.
- [13] 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, 2018.
- [14] R. K. Katzschmann, M. Thieffry, O. Goury, A. Kruszewski, T.-M. Guerra, C. Duriez, and D. Rus, "Dynamically closed-loop controlled soft robotic arm using a reduced order finite element model with state observer," in *IEEE International Conference on Soft Robotics*, 2019, pp. 717–724.
- [15] R. J. Webster III and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, 2010.
- [16] M. W. Hannan and I. D. Walker, "Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots," *Journal of Robotic Systems*, vol. 20, no. 2, pp. 45–63, 2003.
- [17] A. D. Marchese and D. Rus, "Design, kinematics, and control of a soft spatial fluidic elastomer manipulator," *The International Journal of Robotics Research*, vol. 35, no. 7, pp. 840–869, 2016.
- [18] I. S. Godage, G. A. Medrano-Cerda, D. T. Branson, E. Guglielmino, and D. G. Caldwell, "Dynamics for variable length multisection continuum arms," *The International Journal of Robotics Research*, vol. 35, no. 6, pp. 695–722, 2016.
- [19] A. D. Marchese, R. K. Katzschmann, and D. Rus, "Whole arm planning for a soft and highly compliant 2d robotic manipulator," 2014.
- [20] H. Jiang, Z. Wang, Y. Jin, X. Chen, P. Li, Y. Gan, S. Lin, and X. Chen, "Hierarchical control of soft manipulators towards unstructured interactions," *The International Journal of Robotics Research*, vol. 40, no. 1, pp. 411–434, 2021.
- [21] A. D. Marchese, R. Tedrake, and D. Rus, "Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator," *The International Journal of Robotics Research*, vol. 35, no. 8, pp. 1000–1019, 2016.
- [22] T. Mahl, A. Hildebrandt, and O. Sawodny, "A variable curvature continuum kinematics for kinematic control of the bionic handling assistant," *IEEE transactions on robotics*, vol. 30, no. 4, pp. 935–949, 2014.
- [23] H.-J. Su, "A pseudorigid-body 3r model for determining large deflection of cantilever beams subject to tip loads," 2009.
- [24] V. K. Venkiteswaran, J. Sikorski, and S. Misra, "Shape and contact force estimation of continuum manipulators using pseudo rigid body models," *Mechanism and machine theory*, vol. 139, pp. 34–45, 2019.
- [25] R. S. Penning and M. R. Zinn, "A combined modal-joint space control approach for continuum manipulators," *Advanced Robotics*, vol. 28, no. 16, pp. 1091–1108, 2014.
- [26] C. Della Santina, R. K. Katzschmann, A. Bicchi, and D. Rus, "Model-based dynamic feedback control of a planar soft robot: trajectory tracking and interaction with the environment," *The International Journal of Robotics Research*, vol. 39, no. 4, pp. 490–513, 2020.
- [27] R. K. Katzschmann, C. Della 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 *IEEE International Conference on Soft Robotics*, 2019, pp. 454–461.
- [28] M. Trumić, C. Della Santina, K. Jovanović, and A. Fagiolini, "Adaptive control of soft robots based on an enhanced 3d augmented rigid robot matching," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 4991–4996.
- [29] 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, 2020.
- [30] H. Wang, W. Chen, X. Yu, T. Deng, X. Wang, and R. Pfeifer, "Visual servo control of cable-driven soft robotic manipulator," in *IROS*, 2013, pp. 57–62.
- [31] R. J. Roesthuis and S. Misra, "Steering of multisegment continuum manipulators using rigid-link modeling and fbg-based shape sensing," *IEEE transactions on robotics*, vol. 32, no. 2, pp. 372–382, 2016.
- [32] M. Khoshnam and R. V. Patel, "A pseudo-rigid-body 3r model for a steerable ablation catheter," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 4427–4432.
- [33] B. Siciliano and O. Khatib, *Handbook of Robotics*. Springer, 2016.
- [34] A. Yeshmukhametov, K. Koganezawa, and Y. Yamamoto, "Design and kinematics of cable-driven continuum robot arm with universal joint backbone," in *IEEE International Conference on Robotics and Biomimetics*, 2018, pp. 2444–2449.
- [35] W. Xu, T. Liu, and Y. Li, "Kinematics, dynamics, and control of a cable-driven hyper-redundant manipulator," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 4, pp. 1693–1704, 2018.
- [36] K. M. Lynch and F. C. Park, *Modern robotics*. Cambridge University Press, 2017.
- [37] R. W. Brockett, "Robotic manipulators and the product of exponentials formula," in *Mathematical Theory of Networks and Systems*, 1984, pp. 120–129.
- [38] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *ASME Journal of Applied Mechanics*, vol. 22, no. 2, pp. 215–221, 1955.
- [39] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.
- [40] J. Jung, R. S. Penning, N. J. Ferrier, and M. R. Zinn, "A modeling approach for continuum robotic manipulators: Effects of nonlinear internal device friction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 5139–5146.
- [41] J.-J. E. Slotine, W. Li *et al.*, *Applied nonlinear control*. Prentice hall Englewood Cliffs, NJ, 1991, vol. 199.
- [42] W. Lohmiller and J.-J. E. Slotine, "On contraction analysis for non-linear systems," *Automatica*, vol. 34, no. 6, pp. 683–696, 1998.
- [43] H. Jiang, X. Liu, X. Chen, Z. Wang, Y. Jin, and X. Chen, "Design and simulation analysis of a soft manipulator based on honeycomb pneumatic networks," in *IEEE International Conference on Robotics and Biomimetics*, 2016, pp. 350–356.
- [44] R. MacAusland, "The moore-penrose inverse and least squares," *Math 420: Advanced Topics in Linear Algebra*, pp. 1–10, 2014.
- [45] G. Chen, Z. Zhang, and H. Wang, "A general approach to the large deflection problems of spatial flexible rods using principal axes decomposition of compliance matrices," *Journal of Mechanisms and Robotics*, vol. 10, no. 3, p. 031012, 2018.
- [46] B. L. Conrad, J. Jung, R. S. Penning, and M. R. Zinn, "Interleaved continuum-rigid manipulation: An augmented approach for robotic minimally-invasive flexible catheter-based procedures," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 718–724.
- [47] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.