2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Kinematic Modeling and Control of a Soft Robotic Arm with Non-constant Curvature Deformation

Zhanchi Wang, Gaotian Wang, Xiaoping Chen, and Nikolaos M. Freris

***Abstract*— The passive compliance of soft robotic arms renders the development of accurate kinematic models and model-based controllers challenging. The most widely used model in soft robotic kinematics assumes Piecewise Constant Curvature (PCC). However, PCC introduces errors when the robot is subject to external forces or even gravity. In this paper, we establish a three-dimensional (3D) kinematic representation of a soft robotic arm with pseudo universal and prismatic joints that are capable of capturing non-constant curvature deformations of the soft segments. We theoretically demonstrate that this constitutes a more general methodology than PCC. Simulations and experiments on the real robot attest to the superior modeling accuracy of our approach in 3D motions with unknown loads. The maximum position/rotation error of the proposed model is verified 6.7×/4.6× lower than the PCC model considering gravity and external forces. Furthermore, we devise an inverse kinematic controller that is capable of positioning the tip, tracking trajectories, as well as performing interactive tasks in the 3D space.**

I. INTRODUCTION

Due to the passive compliance and continuous deformation characteristics, soft manipulators have demonstrated advantages over their rigid-bodied counterparts in a variety of applications [1], [2], such as soft surgical robots [3], soft arms in interactive tasks [4], and soft artificial limbs [5]. The design of responsive, reliable, and stable controllers is a vital premise for the proliferation of soft robots in real-life applications. However, due to 'infinite' underactuated and compliant degrees of freedom (DoFs), controlling a soft arm is an exigent endeavor [6].

With the consideration of the universal approximation ability, machine learning, also known as model-free approach, is used to solve the control problems of soft arms with continuous deformation and nonlinear characteristics [7], [8], [9]. However, such methods require collecting large volumes of data for training and are difficult to generalize to different tasks and robot platforms. Analytical models thus play an important role in better predicting the behavior of soft robots and developing stability-guaranteed controllers. There has been work on developing models that treat soft robots as continuum mechanics with elasticity such as finite element method (FEM) [10], [11], Cosserat rod [12], [13], [14], and Euler–Bernoulli beam [15] based models. These approaches represent soft robots by stacking an infinite number of infinitesimal microsolids, thus allowing for high fidelity of the continuum mechanics of the soft body. However, these models require solving nonlinear partial differential equations, which makes them computationally demanding. Although discretization and dimensionality reduction techniques have been invoked to apply these models in control [16], [17], [18], their applicability is limited due to their complexity in sensing and incorporating external forces such as those generated by human interaction into the calculation [19].

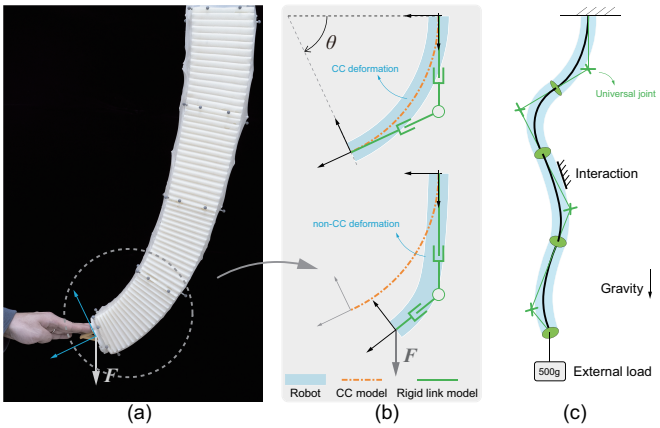

Fig. 1. A soft manipulator interacting with a human. The deformation of the tip segment is illustrated in (b): modeling a soft segment (blue) without and with an external force using a constant curvature model (orange) and a rigid link model (green). (c) Illustration of a rigid link model capturing the non-constant curvature deformation of a 3D multi-segment soft arm.

Piecewise Constant Curvature (PCC) model is the most widely adopted method for controlling soft manipulators which treats the arm as a concatenation of of multiple segments, each one being an arc of constant curvature (CC) [20]. A large number of models and controllers were developed based on PCC [6], [21], [22]. Della Santina et al. [23] developed a discrete joint robot analogue of a soft robot under PCC and established a controller leveraging methods for rigid robots. This method was extended to 3D in [24], [25], where a 10-joint rigid robot was used to imitate the motion of a single segment of constant curvature. Although the PCC assumption is natural, it fails to capture non-constant curvature deformation of the arm that is ubiquitous in the presence of external forces and gravity [6]. As shown in Fig. 1, when the soft arm interacts with the human, all segments of the arm no longer deform in a constant curvature manner. As the number of segments increases, the errors accumulate and propagate to the end effector thus deteriorating the fidelity of forward/inverse kinematics. Although researchers have developed variable curvature models [26],

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. Corresponding author: Nikolaos M. Freris.

which model a single segment as n segments of CC, most of them are designed to cope with non-constant curvature deformations caused by geometric structures rather than external forces or interactions.

From a modeling perspective, non-constant curvature deformation of soft manipulators under external force can be represented by a pseudo-rigid-body approximation [27], [28]. In [29], 20 serial-connected rigid links of fixed length are used to approximate the shape of a planar continuum manipulator with two segments. Increasing the number of joints and links means a finer discretization of the system and gives better model accuracy. However, from a control perspective, a redundant number of joints and links resort to over-parameterizing each segment of the soft robot. This results in an augmented state space which not only incurs an increased sensing burden for state estimation [29], [30] but also poses challenges in developing the mapping between the high-dimensional configuration space and the low-dimensional actuation space.

In this paper, we propose a methodology for the kinematic modeling of soft robots by casting a soft manipulator as a Piecewise Universal Joint (PUJ) robot, i.e., a set of segments where each one is modeled as two prismatic joints connected by a universal joint. The proposed PUJ model is based on assumptions that each actuated segment *bends on a plane* and *exhibits negligible torsional deformation*. These assumptions are widely adopted in literature [21], [31] and in particular valid from the design of the robot used in this paper exploiting the mechanical properties of the honeycomb structure [32]. In this model, the configuration is clearly connected to the physical quantities of the robot, i.e., the two bending and one elongation degrees of freedom, that is, the configuration can be intuitively mapped to the pneumatic actuation. Capitalizing on the derived model, we devise a closed-loop controller for motion control in the 3D space. The effectiveness of both the model and the controller is evaluated through simulations and experiments on a four-segment 3D soft robotic arm.

Contributions:

- We propose a new modeling methodology (PUJ) for 3D soft arms, where a single segment is described using four state variables, which capture the non-constant deformation of soft segments under the action of external force and gravity.
- We devise a model-based controller for motion control in the 3D space capable of positioning the tip, tracking trajectories, as well as performing interactive tasks.
- We experimentally validate the proposed model and controller both in simulations and on a 3D multi-segment soft manipulator. Our findings corroborate the advantages of PUJ vs. PCC under external forces and gravity in terms of accuracy.

II. Kinematics

This section presents a methodology for kinematic modeling of a multi-segment soft arm based on its rigid-link analogue, obtained by casting each segment as two prismatic joints connected by a universal joint. We further establish that the proposed *Piecewise Universal Joint* (PUJ) model is richer than PCC.

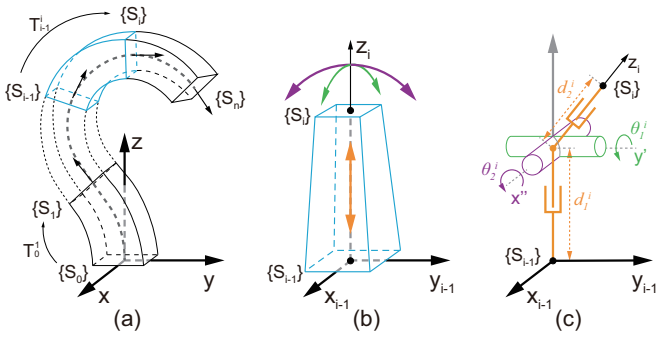

Fig. 2. An illustration of a soft arm alongside its equivalent pseudo rigid body representation. (a) The kinematic representation of a n-segment soft robotic arm. (b) An illustration of a single segment where two bending (green and purple) and one elongation (orange) motion primitives. (c) The kinematic representation of one segment: translation lengths d_1^i, d_2^i and rotation angles θ_1^i, θ_2^i are chosen as the state variables. The same color in (b) and (c) represents the same motion primitive.

A. Preliminaries

Consider a continuum segment that bends on a plane without twisting. A constant curvature deformation can be captured by two prismatic joints connected by a revolute joint. When subject to an external force, the segment will bend in a non-constant curvature fashion (see Fig. 1(b)), thus introducing errors in the position/orientation when adopting a CC model. In contrast, the rigid link model can still be used to accurately describe the position and orientation of the endpoint. This simple example serves as our inspiration for the methods presented in this paper. To generalize this observation to 3D multi-segment soft arms, we employ the following assumptions:

Assumption 1: Each segment bends in a single plane (in 3D space) at a time.

Assumption 2: Torsional deformation is negligible.

These assumptions have been verified to be reasonable on many soft arms [33], [26], [32], while for some other manipulators, for example, whose body is made of silica gel, torsion is not negligible [13], [34]. Here we focus on the former case. We also demonstrated through simulation that the PUJ model has better accuracy than PCC even when the assumptions are violated (see section V). Under these two assumptions, the motion characteristic of a soft segment closely resembles a rigid serial link mechanism, i.e., two prismatic joints connected by a universal joint (Fig. 2). In this paper, we capitalize on this principle to model the forward and inverse kinematics of a multi-segment soft arm.

B. PUJ Kinematics

We consider a soft robot composed of n segments and denote the reference coordinate systems at the base of the manipulator and the ends of each segment as $\{S_0\}, \ldots, \{S_n\}$, respectively: see Fig. 2(a) for an illustration. These coordinate systems serve as the kinematic representation of

the robot. We abstract the kinematics via a homogeneous transformation matrix $\boldsymbol{T}_{i-1}^{i} \in \mathbb{R}^{4\times4}$ [35, Chapter 2] between consecutive coordinate systems:

$$\boldsymbol{T}_{i-1}^{i} = \begin{bmatrix} \boldsymbol{R}_{i-1}^{i} & \boldsymbol{l}_{i-1}^{i} \\ (0,0,0) & 1 \end{bmatrix}, \boldsymbol{T}_{0}^{n} = \prod_{i=1}^{n} \boldsymbol{T}_{i-1}^{i}, \tag{1}$$

where $\boldsymbol{T}_{i-1}^{i}$ transforms vectors from the i-th coordinate frame to the $(i-1)$-th, $\boldsymbol{R}_{i-1}^{i} \in \mathbb{R}^{3\times3}$ is a rotation matrix, and $\boldsymbol{l}_{i-1}^{i} \in \mathbb{R}^{3}$ is a translation (column) vector. The forward kinematics for the end effector of an n-segment soft robotic manipulator are characterized by the product of the segment transformation matrices. , thus $\boldsymbol{T}_{0}^{n}$ serves to express the end effector in the coordinate frame of the base of the manipulator.

The n-segment soft robotic arm is represented by $2n$ prismatic joints connected by n serial universal joints. Taking the i-th segment for illustration (Fig. 2(b)), we showcase how to select the configuration variables for the universal joint (Fig. 2(c)), and proceed to provide the expression for the homogeneous transformation matrix of the i-th segment. The tuple $\boldsymbol{q}^{i} := (d_1^i, \theta_1^i, \theta_2^i, d_2^i)$ represents the configuration variables for the i-th segment, whence $\boldsymbol{q} := \{\boldsymbol{q}^1, \ldots, \boldsymbol{q}^n\}^\top$ is the kinematic state of the robot. 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 (Fig. 2(d)). The homogeneous transformation matrix for the i-th segment is then written as:

$$\begin{aligned} \boldsymbol{T}_{i-1}^{i} &= \mathbf{Trans}(\boldsymbol{z}, d_1^i)\mathbf{Rot}(\boldsymbol{y}, \theta_1^i)\mathbf{Rot}(\boldsymbol{x}, \theta_2^i)\mathbf{Trans}(\boldsymbol{z}, d_2^i) \\ &= \begin{bmatrix} c_{\theta_1^i} & s_{\theta_1^i} s_{\theta_2^i} & s_{\theta_1^i} c_{\theta_2^i} & d_2^i s_{\theta_1^i} c_{\theta_2^i} \\ 0 & c_{\theta_2^i} & -s_{\theta_2^i} & -d_2^i s_{\theta_2^i} \\ -s_{\theta_1^i} & c_{\theta_1^i} s_{\theta_2^i} & c_{\theta_1^i} c_{\theta_2^i} & d_1^i + d_2^i c_{\theta_1^i} c_{\theta_2^i} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \end{aligned} \tag{2}$$

where $\mathbf{Trans} \in \mathbb{R}^{4\times4}$ denotes the homogeneous transformation matrix of translation along an axis, $\mathbf{Rot} \in \mathbb{R}^{4\times4}$ denotes the homogeneous transformation matrix of rotation around an axis, and c_θ, s_θ respectively abbreviate $\cos\theta$, $\sin\theta$.

C. Inverse Kinematics

We consider the inverse kinematic problem, i.e., how to estimate the configuration vector $\hat{\boldsymbol{q}}$ from the posture of the reference coordinate systems. The postures $\{S_0\}, ..., \{S_n\}$ are measured from sensors such as cameras. The 3x3 upper left block in (2) is the measured rotation matrix $\boldsymbol{R}_{i-1}^{i}$ and the 3x1 upper right block is the measured translation vector $\boldsymbol{l}_{i-1}^{i}$. 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. The lengths d_1^i and d_2^i are derived from the measured translation vector $\mathbf{l}_{i-1}^{i} = (x_{i-1}^i, y_{i-1}^i, z_{i-1}^i)^\top$ as follows:

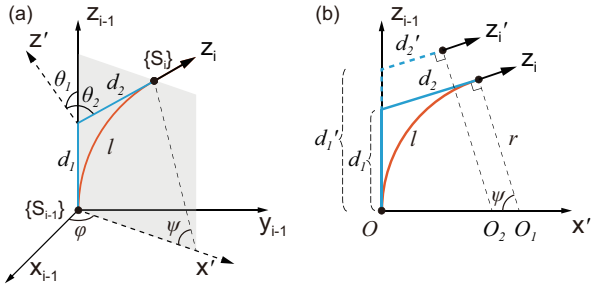

Fig. 3. An illustration of PCC and PUJ representations in 3D for a soft robot segment. (a) A PCC configuration can be represented by PUJ ($d_1 = d_2$). (b) The illustration on the x'-z_{i-1} plane for the two representations with variable d_1, d_2 for PUJ.

$$\begin{cases} d_2^i = -\frac{y_{i-1}^i}{s_{\theta_2^i}}, d_1^i = z_{i-1}^i - d_2^i c_{\theta_1^i} c_{\theta_2^i}, & \theta_2^i > \epsilon \;\&\; \theta_2^i > \theta_1^i \\ d_2^i = \frac{x_{i-1}^i}{s_{\theta_1^i} c_{\theta_2^i}}, d_1^i = z_{i-1}^i - d_2^i c_{\theta_1^i} c_{\theta_2^i}, & \theta_1^i > \epsilon \;\&\; \theta_1^i > \theta_2^i \\ d_1^i = d_2^i = \frac{z_{i-1}^i}{1 + c_{\theta_1^i} c_{\theta_2^i}} = \frac{z_{i-1}^i}{2}, & \theta_1^i < \epsilon \;\&\; \theta_2^i < \epsilon \end{cases}. \tag{3}$$

The parameter ϵ is used for numerical stability ($\epsilon = 0.05rad$ was chosen in our implementation). When the configuration of the segment is almost straight ($\theta_1^i < \epsilon$ & $\theta_2^i < \epsilon$), the two prismatic joints 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 $\boldsymbol{l}_{i-1}^{i}$ to obtain d_1^i and d_2^i with the constraint that $d_1^i = d_2^i$ in this scenario, whence the way to distinguish d_1 and d_2 is unique.

D. Kinematic Model Properties

The PUJ principle relaxes the PCC constraint that each segment of the manipulator must be deformed in a constant curvature manner and hence 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 including gravity. The superscript symbol is dropped in the following development to simplify the notation.

Lemma 2.1: For any position and orientation described by a tuple of PCC configuration variables (ϕ, ψ, l), there always exists a tuple of PUJ configuration variables $(d_1, \theta_1, \theta_2, d_2)$ to represent the same with $d_1 = d_2$.

Proof: In the PCC representation, the orientation of the z_i-axis of the tip coordinate system $\{S_i\}$ is derived by first rotating the base coordinate system $\{S_{i-1}\}$ by ϕ around the z_{i-1}-axis and then rotating by ψ around the rotated y_{i-1}-axis, as shown in Fig. 3(a). This is described as $(c_\phi s_\psi, s_\phi s_\psi, c_\psi)^\top$. As for the PUJ representation, the process is to first rotate θ_1 around the y_{i-1}-axis and then by θ_2 around the rotated x_{i-1}-axis, which gives $(s_{\theta_1} c_{\theta_2}, -s_{\theta_2}, c_{\theta_1} c_{\theta_2})^\top$. Equating the two vectors admits a

solution for (θ_1, θ_2) provided by:

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

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

For the position representation of $\{S_i\}$, we adopt a geometric perspective. We draw a tangent line of the arc at its tip on the plane where the deformation occurs, i.e., the x'-z_{i-1} plane as shown in Fig. 3(b). We set the position where the joints meet in the PUJ model as the point where the tangent line and the z'-axis intersect. When d_1, d_2, and l satisfy $d_1 = d_2 = \frac{l}{\psi}\tan(\frac{\psi}{2})$, the positions of $\{S_i\}$ given by both PCC and PUJ are the same. ■

Theorem 2.2: The state-space (position and orientation) described by the PCC model is a proper subset of the space described by the PUJ model.

Proof: The subset property follows from Lemma 2.1. We establish strict inclusion by showing 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 the case where PUJ and PCC are used to describe the soft robot deformations on the same x'-z_{i-1} plane (see Fig. 3(b)). In case $d_1' \neq d_2'$, it is not possible to connect the base and tip with an arc and still meet the orientation constraints of both ends. ■

III. Task Space Controller

The differential kinematics of the soft robot is given by

$$\dot{\mathbf{x}} = \mathbf{J}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{4}$$

where $\mathbf{J} \in \mathbb{R}^{6\times 4n}$ is the Jacobian matrix, and $\boldsymbol{q} \in \mathbb{R}^{4n}$ is the state vector as defined in Section II-C.

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 $\boldsymbol{\tau} = f_1(\bar{\mathbf{x}}, \dot{\bar{\mathbf{x}}}, \hat{\mathbf{x}})$, 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:

$$\begin{aligned} \Delta\boldsymbol{q} &= \mathbf{J}^\dagger \left(k_p \mathbf{e}_x + k_d \dot{\mathbf{e}}_x + k_i \int \mathbf{e}_x + \dot{\bar{\mathbf{x}}} \right), \\ \Delta\boldsymbol{\tau} &= \boldsymbol{K}\Delta\boldsymbol{q}, \end{aligned} \tag{5}$$

where $\mathbf{J}^\dagger$ denotes the Moore-Penrose pseudo-inverse of the Jacobian which is calculated numerically by virtually moving each joint by a small step and observing the change of the pose of the end effector. $\mathbf{e}_x := \bar{\mathbf{x}} - \hat{\mathbf{x}}$, and $\boldsymbol{K} \in \mathbb{R}^{4n\times 4n}$ is a diagonal stiffness matrix that represents the relationship between joint variables and torques. In this paper, the elements in $\boldsymbol{K}$ are obtained through measurements and set as fixed values to capture the first-order structural stiffness of the arm. k_p, k_d, k_i are the proportional–integral–derivative (PID) feedback gains (these are empirically tuned and set equal for all joints of the robot). The block diagram of the proposed task space controller is shown in Fig. 4.

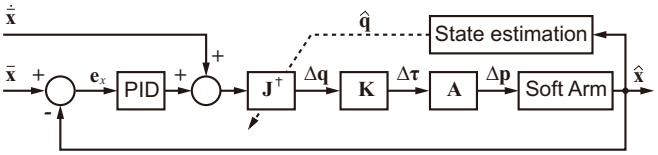

Fig. 4. Block diagram of the proposed kinematic controller in the task space: $\mathbf{J}^\dagger$ is the Moore-Penrose pseudo-inverse of the Jacobian and $\mathbf{A}$ is a conversion mapping between the torque in the PUJ model and the air pressure of the soft arm ($\mathbf{p}$). The pose of the soft arm in task space is obtained using a motion capture system.

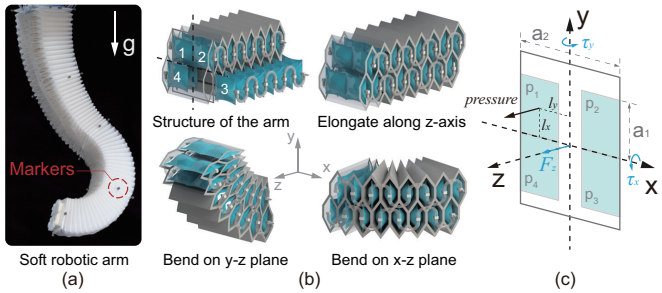

Fig. 5. Hardware overview. (a) The soft robotic manipulator for validating the proposed models. (b) Design and 3D deformation of a segment. (b) The cross-section of the manipulator and the schematic diagram of the airbag distribution.

IV. Soft Arm

A. Hardware Platform

We use a soft robotic manipulator [4] to validate the proposed model and controller. The design and fabrication of the soft arm are detailed in [32]. The manipulator consists of four segments and is actuated by 16 proportional valves. The manipulator is installed straight down and is capable of flexible motion in the 3D space. Reflective markers are placed 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 (Fig. 5(a)).

B. Torque To Pressure Mapping

The structure of the manipulator is shown in Fig. 5(b). Four groups of airbags are placed side by side in a honeycomb structure. The air pressures of the four airbags in the i-th segment are denoted by $p_1^i, p_2^i, p_3^i, p_4^i$. We adopt the assumption $p_1^i + p_4^i = p_2^i + p_3^i$ when operating the arm which corresponds to reducing the internal consumption pertaining to the driving forces.

The PUJ model represents the deformation of a single segment of the soft robot as two orthogonal bends as well as elongations. This property makes it possible to model the mapping from the actuation space to the configuration space through simple linear equations, as follows:

$$\begin{bmatrix} p_1^i & p_2^i \\ p_3^i & p_4^i \end{bmatrix} = \begin{bmatrix} p_y^i & 0 \\ p_y^i & 0 \end{bmatrix} + \begin{bmatrix} p_x^i & p_x^i \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} p_z^i & p_z^i \\ p_z^i & p_z^i \end{bmatrix}. \tag{6}$$

The vector $\mathbf{p} := (p_y^1, p_x^1, p_z^1 \ldots, p_y^n, p_x^n, p_z^n)^\top \in \mathbb{R}^{3n}$ concatenates the air pressure components that cause the bending around the y and x axes (negative values mean bending

in the opposite direction) as well as p_z^i that cause the elongation along the z axis. It is worth noting that the two prismatic joints in the PUJ model correspond to the same segment of the soft arm. Thus these two joints can not be controlled independently. The difference in their states is more a reflection of the external force on the arm. In the implementation, we sum the actuation of these two joints to control the elongation of the arm. We relate the air pressure components with the produced torque and force using $\tau = Fl$ and $F = ps$, in which s represents the area and l length. In the i-th segment of PUJ, the torque is then written as:

$$\boldsymbol{\tau}^i = [\tau_y^i, \tau_x^i, F_z^i]^\top = [2sl_y p_y^i, 2sl_x p_x^i, 4sp_z^i]^\top \quad (7)$$

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 Fig. 5(c).

C. Simulation

We implement a simulation environment based on Cosserat rod model to verify the proposed model. In particular, we use the Matlab toolbox (SoRoSim) developed for hybrid rigid-soft systems [36] to simulate a robot with four segments. Each segment is $15cm$ in length with a square cross-section of $10cm \times 10cm$. The properties of the material are set as density $1000Kg/m^3$, Youngs Modulus $1MPa$, and poissons ratio 0.5.

V. Experimental Results

This section verifies the PUJ-based kinematic model and the motion controller through a series of experiments.

A. Evaluation of the Model

1) Kinematic Modeling Accuracy in Simulation: We compare the modeling accuracy of the PUJ model and PCC model in different scenarios. For the first simulation, gravity and external load are set to 0, and a step excitation is given to the manipulator to bend all segments on a plane (Fig. 6(a)). The acceleration of the arm during this process is not zero which makes each segment have a non-constant curvature deformation. The second simulation considers the same actuation with gravity and a $500g$ load at the tip. The last simulation is performed to verify how the model behaves when the modeling assumptions are violated: each segment bends in different directions in the 3D space, putting the whole arm in a helical state. Under gravity and tip loading, the arm undergoes slight axial torsional deformation.

The configuration variables (ϕ, ψ, l) of the PCC model [21] and the configuration variables of the PUJ model are calculated from the homogeneous transformations of each segment. The position errors are calculated by comparing the calculated forward kinematics with the posture of the end effector in simulations. As shown in Fig. 6(a) and (b), PCC exhibits modeling errors when there is an external force. In contrast, PUJ can always correctly (error = 0) describe the kinematics of the soft manipulator in these cases since each segment of the arm is planar bent and not twisted. The oscillation in Fig. 6(a) and (b) is because giving a step excitation will make the arm oscillate near the equilibrium position. The error of the PCC changes with the change in acceleration. When the modeling assumptions that each segment deforms on a plane without torsional deformation are violated, both models incur errors, but the error of PUJ is significantly smaller than that of PCC (see Fig. 6(c)).

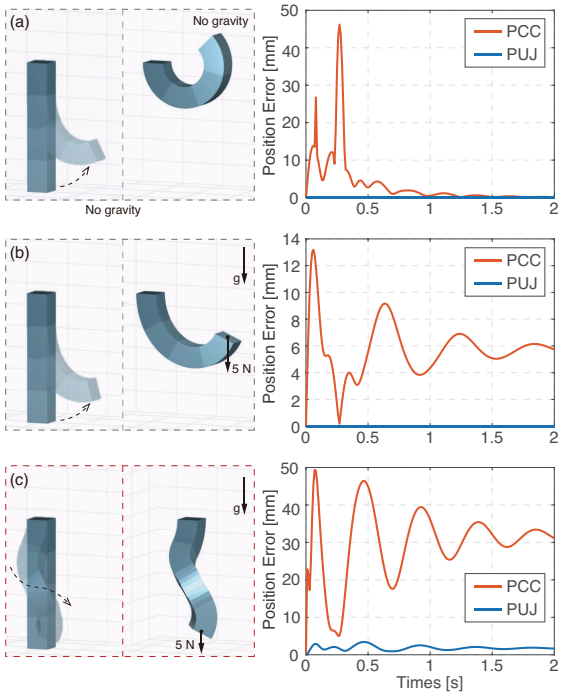

Fig. 6. Comparison of kinematic modeling accuracy between PCC and PUJ model in simulations. Three cases are shown: (a) bending without gravity, (b) bending with gravity and external force, and (c) complex non-planar bending, with position errors plotted on the right, respectively. In the first two cases, the errors of PUJ are 0 (RMSE are less than e^{-12}) since the modeling assumptions are perfectly met. The bending configuration of the last case involves non-planar torques, PUJ will produce errors, but much smaller than PCC.

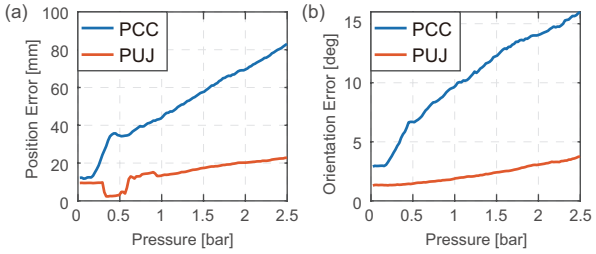

Fig. 7. Comparison of kinematic modeling accuracy between PCC and PUJ model on the real manipulator. The position (a) and orientation (b) errors for gradually increasing the actuation of the manipulator and bending on the plane $x = 0$ with $500g$ load at the tip. Orientation error is defined as the angle between the z-axis of the measurement and that calculated from the model.

2) Kinematic Modeling Accuracy on Real Robot: 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

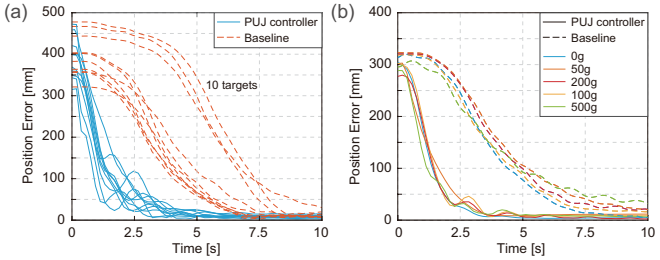

Fig. 8. Experimental results of the controller in regulating the end-effector to ten randomly selected targets (a) and one target with variable loads of $0g$, $100g$, $200g$, and $500g$ (b).

all sections of the manipulator on the plane $x = 0$. We use the MCS to obtain the posture of the markers on the manipulator. The measured postures are then converted to the configuration variables of the PCC model and the PUJ model as explained in the prequel. The results are illustrated in Fig. 7(a) and (b), where it becomes apparent that the PUJ model performs globally better than the PCC model. The maximum position/orientation error of the PCC model is $3.7/4.6$ times higher than the PUJ model. Since we use virtual rigid bodies in the MCS, the deformation of the robot causes an inevitable offset between the marked point and the actual point of interest (center of boths end of each segment), thereby introducing errors. The ratio of maximum error to arm length for the PUJ model is 3%. It is noteworthy that the error of the PUJ model does not scale with pressure: this is a positive attribute, especially when the soft manipulator is used in scenarios where external forces can not be ignored, such as daily interactive tasks.

B. Evaluation of Controllers

Ten targets in the workspace are chosen by randomly inflating. The manipulator is commanded to reach the targets from a fixed starting point (resting position). The simplified Jacobian model presented in our previous work [4] is used to establish a baseline method for comparisons. To test the ability to cope with external loads, we fixed the target and conducted four trials with variable loads of $0g$, $100g$, $200g$, and $500g$ which is unknown to the robot. The absolute position errors are shown in Fig. 8. The PUJ-based controller is able to reach different targets in the 3D space under different unknown loads (Mean steady-state error: 6.75mm for PUJ and 16.07mm for baseline). Compared with the baseline, the controller proposed in this paper reaches the target at the same accuracy using much less time (10% settling time: 3.34s for PUJ and 6.88s for baseline). For assessing the tracking ability of the controller, we use a predefined pressure sequence to inflate the manipulator to make its tip draw a star in the 3D space (Fig. 9). The pose of the tip is recorded at a frequency of $50Hz$, for a total of 500 poses representing 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 was set to $20Hz$ the velocity of the reference trajectory is given as $\dot{\bar{\boldsymbol{x}}}_t = \frac{\bar{\boldsymbol{x}}_t - \bar{\boldsymbol{x}}_{t-1}}{\delta t}$ (the robot completes the trajectory

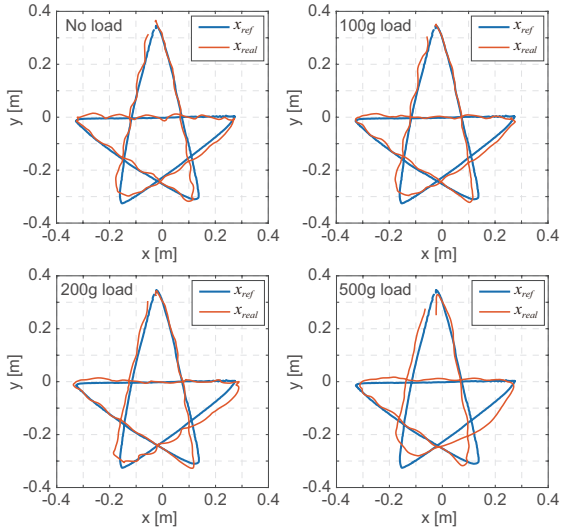

Fig. 9. Experiments of task space controller tracking trajectories under different external loads. From left to right are the experimental results with variable loads of $0g$, $100g$, $200g$, and $500g$. The control frequency is 20Hz.

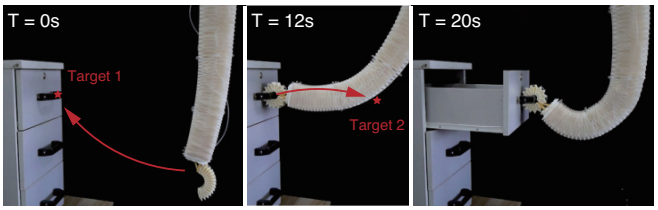

Fig. 10. Snapshots of the soft arm opening a drawer using the proposed task space controller.

within 25s). The experiment was conducted for varying loads of $100g$, $200g$, and $500g$. Fig. 9 shows the projection of the 3D trajectory on the x-y plane. The outer side of the reference trajectory is about $0.65m \times 0.9m$: this corroborates the ability of the controller for large-range motion control and the ability to handle unknown loads.

We further demonstrate the capability of the proposed controller in a practical interactive task (Fig. 10). We manually specify the position of a drawer handle (Target 1) so that the arm can reach and grab it. Then, we select another target point (Target 2) and command the arm to reach the target to output a force to open the drawer.

VI. Conclusion

We have proposed a new methodology for modeling soft robots in 3D tasks with unknown external forces. The proposed model exploits the visually perceived non-constant curvature deformation of the robot and fits the motion parameters of the robot to a pseudo-rigid body. Compared with the PCC, PUJ achieves a richer representation and yields more fidelitous and robust predictive models for describing non-constant curvature deformations. A closed-loop inverse kinematic controller based on the PUJ model is implemented on a four-segment soft arm, demonstrating fast and accurate motion control in the 3D space.

REFERENCES

[1] 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.

[2] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, pp. 467–475, 2015.

[3] M. Cianchetti, T. Ranzani, G. Gerboni, I. De Falco, C. Laschi, and A. Menciassi, "Stiff-flop surgical manipulator: Mechanical design and experimental characterization of the single module," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 3576–3581.

[4] 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.

[5] P. H. Nguyen, C. Sparks, S. G. Nuthi, N. M. Vale, and P. Polygerinos, "Soft poly-limbs: Toward a new paradigm of mobile manipulation for daily living tasks," *Soft Robotics*, vol. 6, no. 1, pp. 38–53, 2019.

[6] 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.

[7] T. George Thuruthel, E. Falotico, M. Manti, A. Pratesi, M. Cianchetti, and C. Laschi, "Learning closed loop kinematic controllers for continuum manipulators in unstructured environments," *Soft Robotics*, vol. 4, no. 3, pp. 285–296, 2017.

[8] X. You, Y. Zhang, X. Chen, X. Liu, Z. Wang, H. Jiang, and X. Chen, "Model-free control for soft manipulators based on reinforcement learning," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 2909–2915.

[9] Z. Tang, P. Wang, W. Xin, and C. Laschi, "Learning-based approach for a soft assistive robotic arm to achieve simultaneous position and force control," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8315–8322, 2022.

[10] F. Largilliere, V. Verona, E. Coevoet, M. Sanz-Lopez, J. Dequidt, and C. Duriez, "Real-time control of soft-robots using asynchronous finite element modeling," in *Proceedings of the IEEE/RSJ International Conference on Robotics and Automation*, 2015, pp. 2550–2555.

[11] 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 *Proceedings of the IEEE International Conference on Soft Robotics*, 2019, pp. 717–724.

[12] J. Spillmann and M. Teschner, "Corde: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects," in *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 2007, pp. 63–72.

[13] F. Renda, M. Giorelli, M. Calisti, M. Cianchetti, and C. Laschi, "Dynamic model of a multibending soft robot arm driven by cables," *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1109–1122, 2014.

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

[15] 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.

[16] O. Goury and C. Duriez, "Fast, generic, and reliable control and simulation of soft robots using model order reduction," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1565–1576, 2018.

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

[18] H. Li, L. Xun, G. Zheng, and F. Renda, "Discrete cosserat static model-based control of soft manipulator," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1739–1746, 2023.

[19] C. Armanini, F. Boyer, A. T. Mathew, C. Duriez, and F. Renda, "Soft robots modeling: A structured overview," *IEEE Transactions on Robotics*, 2023.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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 *Proceedings of the IEEE International Conference on Soft Robotics*, 2019, pp. 454–461.

[25] 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 *Proceedings of the American Control Conference*, 2021, pp. 4991–4996.

[26] 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.

[27] R. Kang, D. T. Branson, E. Guglielmino, and D. G. Caldwell, "Dynamic modeling and control of an octopus inspired multiple continuum arm robot," *Computers & Mathematics with Applications*, vol. 64, no. 5, pp. 1004–1016, 2012.

[28] M. Khoshnam and R. V. Patel, "A pseudo-rigid-body 3R model for a steerable ablation catheter," in *Proceedings of the IEEE/RSJ International Conference on Robotics and Automation*, 2013, pp. 4427–4432.

[29] 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.

[30] 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.

[31] P. Rao, Q. Peyron, S. Lilge, and J. Burgner-Kahrs, "How to model tendon-driven continuum robots and benchmark modelling performance," *Frontiers in Robotics and AI*, vol. 7, p. 630245, 2021.

[32] 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 *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, 2016, pp. 350–356.

[33] M. D. Grissom, V. Chitrakaran, D. Dienno, M. Csencits, M. Pritts, B. Jones, W. McMahan, D. Dawson, C. Rahn, and I. Walker, "Design and experimental testing of the octarm soft robot manipulator," in *Unmanned Systems Technology VIII*, vol. 6230. International Society for Optics and Photonics, 2006, p. 62301F.

[34] H. Wang, C. Wang, W. Chen, X. Liang, and Y. Liu, "Three-dimensional dynamics for cable-driven soft manipulator," *IEEE/ASME transactions on mechatronics*, vol. 22, no. 1, pp. 18–28, 2016.

[35] B. Siciliano and O. Khatib, *Handbook of Robotics*. Springer, 2016.

[36] A. T. Mathew, I. M. B. Hmida, C. Armanini, F. Boyer, and F. Renda, "Sorosim: A matlab toolbox for hybrid rigid-soft robots based on the geometric variable-strain approach," *IEEE Robotics & Automation Magazine*, 2022.