



## Article

# Impedance Iterative Learning Backstepping Control for Output-Constrained Multisection Continuum Arms Based on PMA

Yuexuan Xu <sup>1,2</sup>, Xin Guo <sup>1,2,\*</sup>, Jian Li <sup>3,4,\*</sup>, Xingyu Huo <sup>5,\*</sup>, Hao Sun <sup>1,2,\*</sup> , Gaowei Zhang <sup>6</sup> , Qianqian Xing <sup>1</sup>, Minghe Liu <sup>1,2</sup>, Tianyi Ma <sup>1,2</sup> and Qingsong Ding <sup>1,2</sup>

<sup>1</sup> School of Artificial Intelligence and Data Science, Hebei University of Technology, Tianjin 300400, China

<sup>2</sup> Engineering Research Center of Intelligent Rehabilitation Device and Detection Technology, Ministry of Education, School of Artificial Intelligence and Data Science, Hebei University of Technology, Tianjin 300400, China

<sup>3</sup> School of Automation, Beijing University of Posts and Telecommunications, Beijing 100876, China

<sup>4</sup> Beijing Key Laboratory of Rehabilitation Technical Aids for Old-Age Disability and Key Laboratory of Neuro-Functional Information and Rehabilitation Engineering of the Ministry of Civil Affairs, National Research Center for Rehabilitation Technical Aids, Beijing 100176, China

<sup>5</sup> Department of Cardiology (Ward 1), Tianjin Chest Hospital, Tianjin 300202, China

<sup>6</sup> School of Electronic Information and Automation, Civil Aviation University of China, Tianjin 300300, China

\* Correspondence: guoxin@hebut.edu.cn (X.G.); jianli\_628@126.com (J.L.); hxystar1984@163.com (X.H.); sunhao@hebut.edu.cn (H.S.)



**Citation:** Xu, Y.; Guo, X.; Li, J.; Huo, X.; Sun, H.; Zhang, G.; Xing, Q.; Liu, M.; Ma, T.; Ding, Q. Impedance Iterative Learning Backstepping Control for Output-Constrained Multisection Continuum Arms Based on PMA. *Micromachines* **2022**, *13*, 1532. <https://doi.org/10.3390/mi13091532>

Academic Editors: Leiyu Zhang, Peng Su and Dongsoong Han

Received: 13 August 2022

Accepted: 13 September 2022

Published: 16 September 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Abstract:** Background: Pneumatic muscle actuator (PMA) actuated multisection continuum arms are widely applied in various fields with high flexibility and bionic properties. Nonetheless, their kinematic modeling and control strategy proves to be extremely challenging tasks. Methods: The relationship expression between the deformation parameters and the length of PMA with the geometric method is obtained under the assumption of piecewise constant curvature. Then, the kinematic model is established based on the improved D-H method. Considering the limitation of PMA telescopic length, an impedance iterative learning backstepping control strategy is investigated. For one thing, the impedance control is utilized to ensure that the ideal static balance force is maintained constant in the Cartesian space. For another, the iterative learning backstepping control is applied to guarantee that the desired trajectory of each PMA can be accurately tracked with the output-constrained requirement. Moreover, iterative learning control (ILC) is implemented to dynamically estimate the unknown model parameters and the precondition of zero initial error in ILC is released by the trajectory reconstruction. To further ensure the constraint requirement of the PMA tracking error, a log-type barrier Lyapunov function is employed in the backstepping control, whose convergence is demonstrated by the composite energy function. Results: The tracking error of PMA converges to 0.004 m and does not exceed the time-varying constraint function through cosimulation. Conclusion: From the cosimulation results, the superiority and validity of the proposed theory are verified.

**Keywords:** multisection continuum arms; constant curvature model; adaptive ILC with initial error; barrier Lyapunov function; ANSYS/ADAMS/MATLAB

## 1. Introduction

Bionics is inspiring researchers in a growing number of fields, and bionic knowledge is applied by continuum robotics more clearly and intuitively than in other areas of research. Due to the limbs of various mollusks exhibiting extraordinary locomotion, manipulation, and dexterity in complex and restricted environments, more and more flexible materials are utilized to mimic the locomotion mechanisms of mollusk organs, such as an elephant's trunk and octopus' arms in nature, to create new types of bionic robots, also known as

continuum robots, for various applications [1]. A continuum robot can be defined as a robot consisting of a rubber structure that is capable of continuous bending with infinite degrees of freedom. By contrast to conventional robotic arms, continuum robots do not possess discrete joints [2], such that they have the competence to change their original shape and size by altering the elongation and bending joints to adapt to unstructured environments. Continuum robots can be applied in agriculture to pick fruits and vegetables [3]; in industry to carry objects [4]; and in medicine as neuroendoscopic [5] and colonoscopic robots [6], which have promising application prospects. Through the study of the drive approach of continuum robots, it is found that the pneumatic muscle is a lightweight actuator. Compared with other actuators, its significant advantages are mainly reflected in the excellent bionic performance, flexibility, simple structure, high compressibility ratio, and low production price of pneumatic muscle [7]. From the perspective of future development, PMA will become an extremely pivotal part of flexible actuators, which is especially suitable for multisection continuum robots.

Before constructing a reasonable control algorithm, a kinematic model of a continuum robot needs to be established, which is necessarily more complicated and challenging than a conventional robot with a small number of rigid joints. Some valuable methods have been proposed by researchers in kinematic modeling of continuum robots, such as the Cosserat beam theory [8–10], which can provide a more accurate description of the deformation of a continuum robot by taking material properties, load, and gravity into account. The curvilinear equation of motion for an octopus tentacle was designed by Ivanescu et al. when coiling around an object by analyzing the coiling action of an octopus [11]. An accurate geometric model was developed by Rucker et al. to simulate the motion of a concentric tube continuum of a robot arm [12]. Anderson equated the continuum robot with a multisection curve in order to approximate the deformation of each section of the continuum robot [13]. The solution of higher-order partial differential equations and the complicated knowledge of material mechanics were involved in most of the above models. For multisection continuum robots, the real-time control and calculation amount will be dramatically affected. Single hidden-layer linear regression networks were designed by Reinhart and Jochen to learn the kinematics of continuum robots [14]. Nonetheless, this class of kinematic models obtained through autonomous learning cannot describe the specific deformations of continuum robots. Due to the structural characteristics of the continuum robot, the deformation of the continuum constitutive section is close to the constant curvature deformation in the case of no load and small load. The assumption based on constant curvature deformation has become the mainstream approach for kinematic modeling of continuum robots, which not only greatly simplifies the difficulty of describing the deformation of continuum constitutive joints, but also facilitates the calculation of robot poses in real time [15–18]. Among them, substantial fundamental research has been conducted by Walker's team to lay down the theoretical framework of the constant curvature assumption [19]. Novel modal kinematics are presented in the literature [20], stating that the numerical singularity is bypassed in the constant curvature assumption, and the model can be extended to the multisection continuum robots with recursive methods.

Aiming at enhancing the dynamic motion of robots, a growing number of continuum robot dynamics models have been proposed by researchers, which give the general spatial dynamics form of multisection continuum robots in the form of Euler–Lagrange equations. For instance, an approach for modeling multisection continuum robot dynamics is presented by Webster et al. based on the center-of-gravity method [21]. A common drawback of this method is that it does not sufficiently consider the dynamic changes induced by the environment on the robot [22,23]. Furthermore, the computational complexity of the dynamics model grows exponentially with the increase in the number of driven joints. Therefore, it is quite difficult to construct accurate and general dynamic models for different structures of continuum robots.

For tracking control problems under unknown model parameters, iterative learning control (ILC) has proven to be an extremely practical and efficient model-free control

strategy, especially for periodic and repetitive tasks [24]. The basic principle of iterative learning is to utilize the input-output information of a previous system to revise the system parameters, which in turn generates new control inputs. A proposed iterative learning backstepping controller is constructed by [25] combining it with a second-order sliding mode control theory, which is capable of tracking the desired trajectory with certain constraints for a class of nonlinear MIMO systems. Nevertheless, during the design of iterative learning controllers, specific conditions usually need to be satisfied.

- (1) The initial tracking error of the system required is zero.
- (2) Expected output is a priori.
- (3) The experiment must be completed within a limited timeframe.

Condition (1) is considered as an assumption by most of the papers. However, the initial tracking error inevitably exists in applications limited by the reset accuracy, such that it is essential to study ILC with an initial state error. An ILC based on the initial signal correction method is employed by [26] to relax the initial value condition of the ILC. Unfortunately, a large tracking error is generated during the signal correction. Since the expansion and contraction of the pneumatic muscle are constrained, the tracking error must be limited to a given range to guarantee that the pneumatic muscle moves under the working conditions [27]. Therefore, it is indispensable to discuss ILC under output-constrained conditions for a PMA-type continuum robot. An output-constrained control for nonlinear systems is presented in [28] by designing an asymmetric time-varying barrier Lyapunov function (BLF) to ensure the constraints. The power-added integral control is combined with a novel segmental BLF [29], which is applied to a high-order Hessenberg-type nonlinear system containing output constraints in order to effectively solve the asymptotic quiescence problem of the aforementioned system.

Another essential problem in addressing the tracking error convergence in a nonlinear system is the permanent unmodeled uncertainties and nonparametric disturbances. As an accepted constraint solution, an integral-type BLF is deployed to fulfill the output constraint in [30]. As to enhance robustness, unknown nonlinear functions and lumped disturbances are handled by a radial basis function neural network (RBFNN) control and robust control. The derivative of the robust control gain is estimated by a finite-time differentiator. [31] investigated a BFASM controller under the condition of actuator saturation and uncertainties. That is to say, compared to a conventional sliding mode control, the upper bound information of disturbance is not required in the aforementioned method. However, this approach does not deal well with the chatter phenomenon near the equilibrium point. In the context of a multisection continuum robot, based on PMA, a few research questions have been addressed with the BLF method from the authors' research findings. Nevertheless, as to ensure the safe operation of PMA in the convergence process, especially in the ILC strategy with insufficient iterations, BLFs have the competence to be implemented to guarantee the tracking errors are within an acceptable boundary.

Inspired by the above work, a complete set of kinematic modeling is proposed for a class of PMA-based multisection continuum robots. Meanwhile, an impedance iterative learning backstepping control strategy with unknown parameters of the dynamics model is implemented. The ILC is adopted to estimate the unknown parameters of the multisection continuum arms, while an adaptive algorithm-based approach is responsible for handling external disturbances and interaction torques. What is more is that a BLF is implemented to ensure that the tracking error is bounded throughout the iterative process, considering the time-varying constraint on the tracking error. The aforementioned theories are especially adopted for nonrepetitive tasks in which the operative range of the actuator is required to be within a certain limit. The proposed theories can be extended to various application scenarios, such as rehabilitation training, industrial robots responsible for handling and grinding, exploration robots, and agricultural picking and irrigation robots. Hence, the main contributions of this paper can be summarized as follows:

- (1) A multisection continuum arm based on four pneumatic muscles driven in parallel is self-developed, whose comprehensive kinematic model based on the segmental constant curvature assumption is proposed.
- (2) A dynamic controller is designed for the condition of unknown parameters of the dynamic model. A strictly double-closed-loop force-position hybrid control strategy is presented for the lumped disturbances of the model, in which adaptive ILC is applied to the inner loop of the controller without a priori knowledge of the dynamic parameters, and the condition of zero initial tracking error is released by employing the trajectory reconstruction method. Moreover, the adaptive approach is implemented to enhance the system's robustness.
- (3) The log-type BLF is proposed to always satisfy the time-varying constraint of position-tracking error in the time domain and iterative domain, so as to ensure the safe operation of the pneumatic muscle in the given operating range.
- (4) The designed system is simulated jointly in an ANSYS/ADAMS/MATLAB environment. Compared with PID control and iterative learning sliding mode control (ILSMC), the effectiveness of the proposed algorithm is demonstrated.

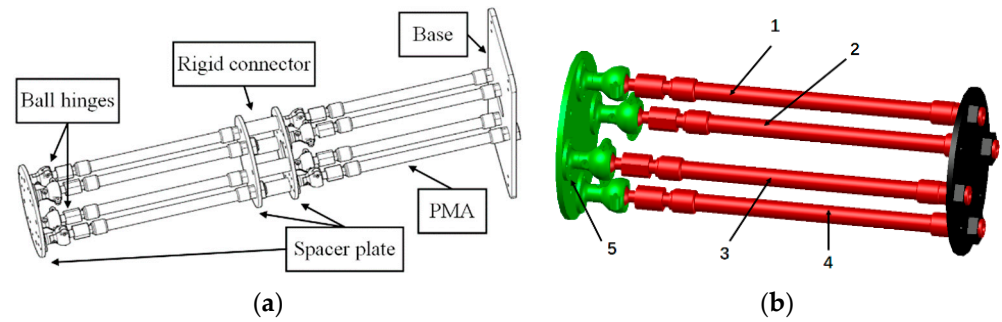
The remainder of this paper is organized as follows: The PMA-based kinematic model of multisection continuum arms is shown in Section 2. The design process of the impedance ILC controller with log-type BLF is given in Section 3. The convergence property of the designed algorithm is demonstrated by means of the composite energy function (CEF) method in Section 4. In Section 5, cosimulation results on ANSYS/ADAMS/MATLAB are depicted to verify the flexible interaction and tracking performance of the proposed algorithm. Finally, conclusions are drawn in Section 6.

## 2. System Modeling and Problem Formulation

A self-developed PMA-based multisection continuum arm is designed as the research object in this paper, whose structure is shown in Figure 1a. The arms are composed of two pneumatic continuum nodes and a rigid connector where each node consists of four pneumatic muscles in parallel (uniformly distributed at 90°), as depicted in Figure 1b. Each pneumatic muscle is connected to the spacer plate by a ball hinge to guarantee that each pneumatic node has four degrees of freedom. The degrees of freedom of a single pneumatic node are calculated as Equation (1). By controlling the magnitude of air pressure in the pneumatic muscles, the individual PMA can be deformed by elongation or bending. What is more, is that the continuum arms are allowed to achieve complicated deformation movements by combining two pneumatic nodes.

$$DOF = 6M - \sum_{i=1}^N P_i \quad (1)$$

where  $DOF$  denotes the DOF of a single continuum node,  $M$  is the number of moving parts of the continuum node,  $N$  indicates the number of kinematic pairs,  $P_i$  is the number of DOF restricted by the  $i$ th kinematic pair. As shown in Figure 1b,  $M = 5$  and the kinematic pair 1–4 restricts five DOF respectively, and the kinematic pair five restricts six DOF, so that a single continuum node has four DOF.



**Figure 1.** (a) Multisection continuum arm structure. (b) Single continuum node structure.

2.1. Kinematic Modeling Based on Constant Curvature Assumption

For the continuum arms to be precisely controlled, an accurate, stable, and efficient kinematic model is essential. A kinematic model is capable of avoiding nonlinear morphological mapping in joint space. For continuum arms, the relationship between PMA length and robot-end poses is described by the kinematic model. The kinematic modeling process for a multisection continuum arm is presented as follows.

$$X = f_{D-H}(f_b(f_a(l_i))) \tag{2}$$

where  $f_a$  denotes the mapping from PMA length to arc parameters;  $f_b$  is the transformation of arc parameters into the function of discrete joints;  $f_{D-H}$  denotes the curve parameter homogeneous transformation matrix (CPHTM) established using the modified D-H method.

2.1.1. Forward Kinematics

The curvature of each section uniform along the centerline of a single pneumatic node requires the following two assumptions: (1) The continuum robot consists of a series of consecutive sections; (2) the potential energy inside each node is uniformly distributed; and the combined force inside each node is uniformly distributed along the centerline of the node. The continuum arms designed in this paper meet the above premises.

The deformation model of a single continuum node is depicted in Figure 2, which can be described by the three arc parameters under the assumption of a constant curvature, bending angle  $\theta_i$ , centerline curvature  $k_i$  (or radius of curvature  $r_i$ ) of the continuum node, and the angle relative to the bending plane concerning the  $+X_i$  axis  $\phi_i$ . The multisection continuum arms designed in this paper contain two continuum nodes, so  $i = 2$ . Given the length of the pneumatic muscle ( $l_{i1}, l_{i2}, l_{i3}, l_{i4}$ ) in a single continuum node, the distance from the center of the pneumatic muscle to the center of the spacer plate is  $d$ . The bottom of the continuum node is denoted as  $\{O_i\}$ , and the top of the continuum node is denoted as  $\{O_{i+1}\}$ . The continuum node can be considered as an infinite number of sheets spliced together, and a single sheet is denoted as  $\{O_i^\kappa\}$ ,  $\kappa \in [0, 1]$ . When  $\kappa_i = 0$ , the coordinate  $(X_i^\kappa, Y_i^\kappa, Z_i^\kappa)$  of sheet  $\{O_i^\kappa\}$  coincides with the base coordinate  $(X_1, Y_1, Z_1)$ ; when  $\kappa_i = 1$ , the coordinate  $(X_i^\kappa, Y_i^\kappa, Z_i^\kappa)$  of sheet  $\{O_i^\kappa\}$  coincides with the top coordinate  $(X_2, Y_2, Z_2)$ . The arc parameter  $(\theta_i, k_i, \phi_i)$  of the deformation of the continuum node can be calculated with the geometric method.



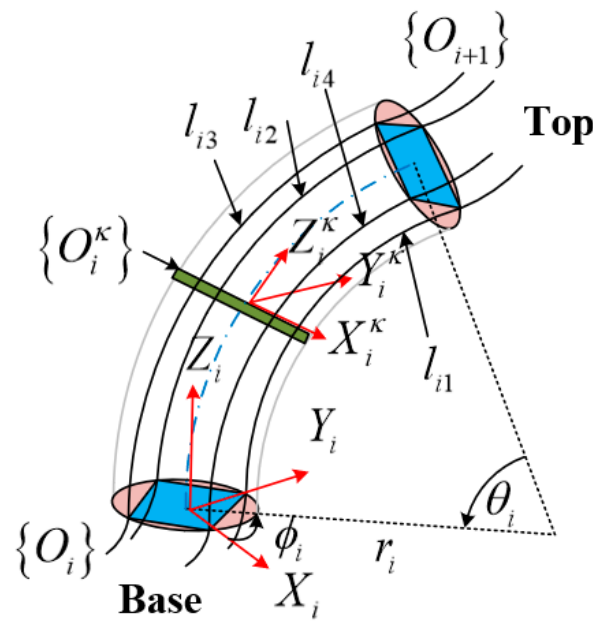


Figure 2. Deformation mode of a single continuum node.

A schematic diagram of a single continuum node bending at deflection angle  $\phi_i$  is provided in Figure 3a, while the top view of the continuum node is shown in Figure 3b. Separate reference coordinate systems are established for the bottom and top. From the geometry of the figure, the radius of curvature measured from the center of the continuum node is related to the radius of curvature of each PMA. In the case of PMA1, the geometric relationship between  $r_i$  and  $r_1$  is presented as:

$$r_1 = r_i - d \cos \alpha_1 \tag{3}$$

where  $\alpha_1$  is the angle between the bending direction of a continuum node and PMA1.

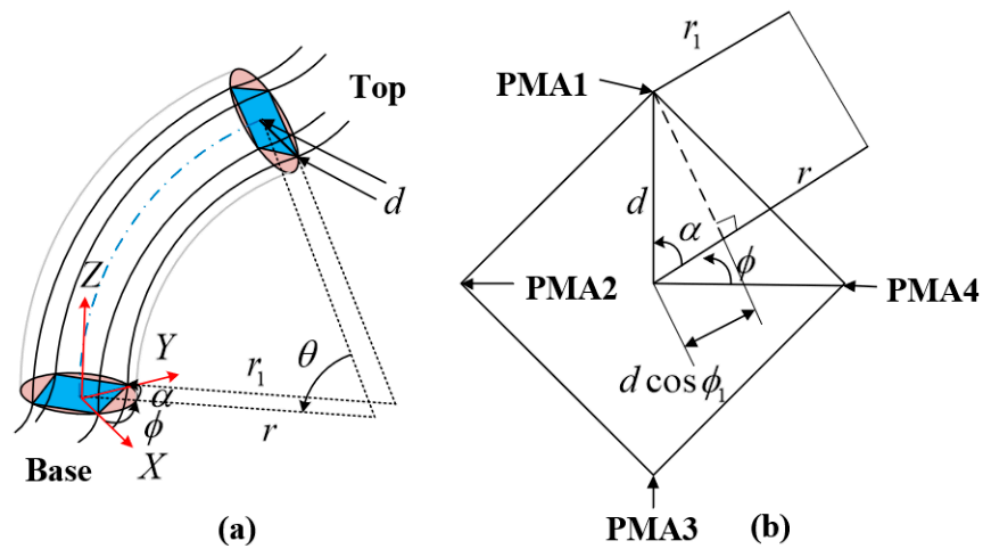


Figure 3. Schematic diagram of the single continuum node: (a) the angle of deflection  $\phi_i$ ; (b) top view.

The arc length of the centerline of the continuum node is defined as  $l_i$ . From the arc length formula,  $l_i = \theta_i r_i$ ,  $l_1 = \theta_i r_1$  is obtained. Then, the arc length formula is substituted

into Equation (3) to obtain the relationship between the arc length of the centerline  $l_i$  and the arc length of the PMA  $l_1$  (applicable to all PMAs).

$$l_i = l_1 + \theta_i d \cos \alpha_1 \tag{4}$$

In Figure 3b, it is easy to see that the following relationship exists between  $\alpha$  and  $\phi_i$  for each PMA.

$$\begin{cases} \cos \alpha_1 = \cos(90^\circ - \phi_i) = \sin \phi_i \\ \cos \alpha_2 = \cos(180^\circ - \phi_i) = -\cos \phi_i \\ \cos \alpha_3 = \cos(270^\circ - \phi_i) = -\sin \phi_i \\ \cos \alpha_4 = \cos(360^\circ - \phi_i) = \cos \phi_i \end{cases} \tag{5}$$

Substitute Equation (5) into Equation (4) to get:

$$l_i(\vec{q}) = \frac{l_1 + l_2 + l_3 + l_4}{4} \tag{6}$$

where  $\vec{q} = [l_1, l_2, l_3, l_4], \sum_j^4 \cos \alpha_j = 0$ .

Generalizing Equation (4) for each PMA and taking the difference of nonadjacent PMAs, Equation (4) can be described as:

$$\begin{cases} \theta_i d = \frac{l_3 - l_1}{\cos \alpha_1 - \cos \alpha_3} = \frac{l_3 - l_1}{2 \sin \phi_i} \\ \theta_i d = \frac{l_4 - l_2}{\cos \alpha_2 - \cos \alpha_4} = \frac{l_4 - l_2}{-2 \cos \phi_i} \end{cases} \tag{7}$$

Divide the above formula to get the deflection angle  $\phi_i$ .

$$\phi_i(\vec{q}) = \arctg\left(\frac{l_3 - l_1}{l_2 - l_4}\right) \tag{8}$$

Substitute the curvature formula into Equation (3) to obtain centerline curvature  $k_i$ .

$$k_i = \frac{l_i - l_{ij}}{l_i d \cos \alpha_{ij}}, j = 1, 2, 3, 4 \tag{9}$$

Let  $j = 1$ , substitute Equations (5) and (8) into (9) to get the centerline curvature  $k_i$  and radius of curvature  $r_i$ .

$$k_i(\vec{q}) = \frac{(-3l_1 + l_2 + l_3 + l_4) \sqrt{(l_4 - l_2)^2 + (l_3 - l_1)^2}}{d(l_1 + l_2 + l_3 + l_4)(l_3 - l_1)} \tag{10}$$

$$r_i(\vec{q}) = \frac{d(l_1 + l_2 + l_3 + l_4)(l_3 - l_1)}{(-3l_1 + l_2 + l_3 + l_4) \sqrt{(l_4 - l_2)^2 + (l_3 - l_1)^2}} \tag{11}$$

**Remark 1.**  $\sin(\arctg(\frac{l_3 - l_1}{l_2 - l_4}))$  is involved in the simplification process of the above equation, and the auxiliary triangle can be designed to obtain the following:

$$\sin(\arctg(\frac{l_3 - l_1}{l_2 - l_4})) = \frac{\sqrt{(l_4 - l_2)^2 + (l_3 - l_1)^2}}{(l_3 - l_1)} \tag{12}$$

The bending angle  $\theta_i$  of a continuum node is obtained from  $l_i = \theta_i r_i$

$$\theta_i(\vec{q}) = \frac{(-3l_1 + l_2 + l_3 + l_4) \sqrt{(l_4 - l_2)^2 + (l_3 - l_1)^2}}{4d(l_3 - l_1)} \tag{13}$$

So far, the functional relationship between all arc parameters of a single continuum node and each PMA has been obtained.

Based on the assumption of equal circular arcs and the structural characteristics of continuum arms, the arc parameters  $(\theta_i, k_i, \phi_i)$  are utilized to replace the translation parameters  $D$  and rotation parameters  $\theta$  of the traditional robot arm. A single continuum node can be discretized into five joints, as shown in Figure 4. Thus, the arc-parameter homogeneous transition matrix (CPHTM) of the first continuum node is obtained as follows:

$$T_i(\vec{q}) = T_{rz}(\phi_i)T_{px}(r_i)T_{ry}(\theta_i)T_{px}(-r_i)T_{rz}(-\phi_i) = \begin{bmatrix} R_i(\vec{q}) & P_i(\vec{q}) \\ 0_{1 \times 3} & 1 \end{bmatrix} \tag{14}$$

where  $T_{rz} \in R^{4 \times 4}$  and  $T_{ry} \in R^{4 \times 4}$  are the rotational homogeneous transformation matrix (RHTM) about the  $z$ -axis and the  $y$ -axis, respectively.  $T_{px} \in R^{4 \times 4}$  is the translation homogeneous transformation matrix (THTM) along the  $x$ -axis, which is expressed as follows:

$$T_{rz}(\phi_i) = \begin{bmatrix} \cos \phi_i & -\sin \phi_i & 0 & 0 \\ \sin \phi_i & \cos \phi_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

$$T_{ry}(\theta_i) = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{16}$$

$$T_{px}(r_i) = \begin{bmatrix} 1 & 0 & 0 & r_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17}$$

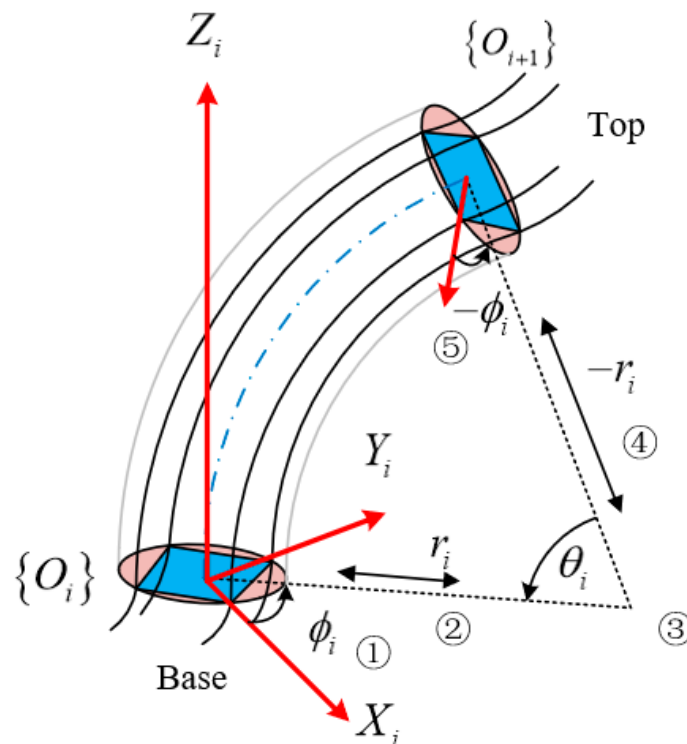


Figure 4. Diagram of single continuum node CPHTM based on enhanced D-H method.



$R_i(\vec{q}) \in SO(3)$  denotes the rotation matrix under the radian parameter, and  $P_i(\vec{q}) \in R^{3 \times 1}$  represents the displacement vector under the radian parameter.

**Remark 2.** Note that the arc parameter expression  $(\theta_i, k_i, \phi_i)$  calculated above has singular values, that is, the continuum arms have no numerical solution to Equations (8), (10), and (13) in the vertical state ( $l_{4i} = l_{4i-1} = l_{4i-2} = l_{4i-3}$ ). In order to avoid the singular value of the arms in the forward kinematics solution process, the arms are divided into a bending state and a vertical state. When the continuum arms merely stretch vertically in the z-axis,  $R_i(\vec{q})$  changes into the unit matrix,  $P_i(\vec{q}) = [0, 0, l_i]^T$ .

**Remark 3.** The multisection continuum arms designed in this paper include a rigid connector where the kinematic model also needs to be considered. As shown in Figure 5, the CPHTM of the first continuum node is denoted as  $T(\vec{q}_1, \kappa_1)$ , the CPHTM of the second node of the intermediate rigid connector is denoted as  $T_m$ , and the CPHTM of the third continuum node is represented as  $T(\vec{q}_2, \kappa_2)$ . The intermediate rigid connector is essentially a rigid parallel platform. It is assumed that the rigid connector does not deform during the movement, that is, the RHTM does not change, and only the THTM is calculated.

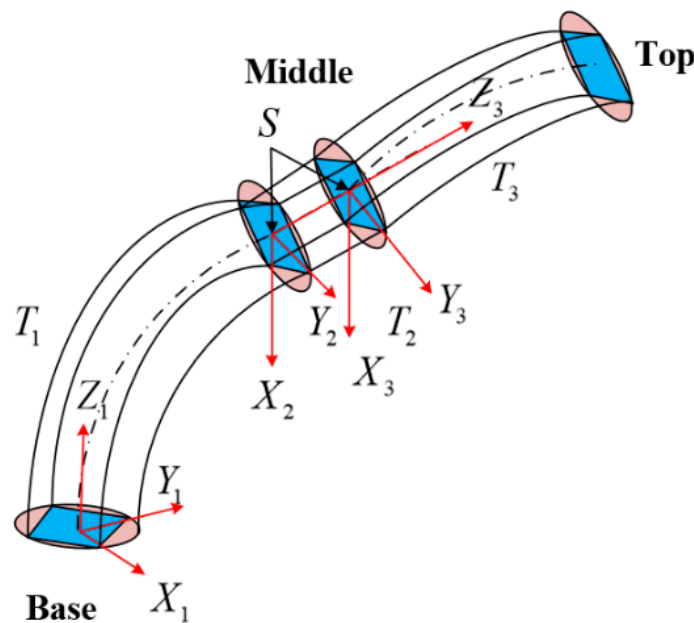


Figure 5. MHTM of multisection continuum arms.

The modal homogeneous transformation matrix (MHTM) of the multisection continuum arm is denoted as:

$$\begin{aligned}
 {}^3_0T &= T(\vec{q}_1, \kappa_1) T_{rz}(0) T_m T_{rz}(0) T(\vec{q}_2, \kappa_2) \\
 &= \begin{bmatrix} \bar{R}_{3 \times 3}(\vec{q}, \kappa^2) & \bar{P}_{3 \times 1}(\vec{q}, \kappa^2) \\ 0_{1 \times 3} & 1 \end{bmatrix} \tag{18}
 \end{aligned}$$

From Equation (18),  $\bar{R}_{3 \times 3}(\vec{q}, \kappa^2)$  is the modal rotation transformation matrix (MRTM),  $\bar{R}_{3 \times 3} = R_1 R_m R_2 \in SO(3), R_m = 1; \bar{P}_{3 \times 1} = P_1 + P_m + R_1(P_1 + P_m) \in R^{3 \times 1}$  denotes the modal translation transformation matrix (MTTM);  $\vec{q} = [\vec{q}_1, \vec{q}_2]^T \in R^{2 \times 4}$  represents the

vector set of joint space,  $\vec{q}^i = [q_{4i-3}, q_{4i-2}, q_{4i-1}, q_{4i}]^T$ ;  $\kappa^2$  indicates the vector set of slice scalar coefficients,  $\kappa^2 = [\kappa_1, \kappa_2]^T \in R^2$ ; The installation position has  $0^\circ$  deviation around the z-axis. Please refer to Supplementary Materials for specific values.

### 2.1.2. Inverse Kinematics

In the previous subsection, the MHTM of a multisection continuum arm is calculated given the known length of the PMA. The lengths of eight PMAs are calculated in this section given the end poses of the arms. In order to facilitate the calculation and implementation, the inverse kinematics model of continuum arms is studied with the method of geometry.

$$l_i = f_a^*(f_b^*(X)) \tag{19}$$

In the above formula,  $f_b^*$  represents the functional relationship of transforming the arms end pose into arc parameters;  $f_a^*$  denotes the functional relationship from arc parameters to PMA length.

Taking a single continuum node as an example for analysis, the pose matrix of the top coordinate system  $(X_2, Y_2, Z_2)$  relative to the base coordinate system  $(X_1, Y_1, Z_1)$  is:

$${}^1_0T = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} & p_x \\ r_{yx} & r_{yy} & r_{yz} & p_y \\ r_{zx} & r_{zy} & r_{zz} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{20}$$

The value of  $p_x, p_y, p_z$  in Equation (20) needs to be given in advance. In the first step, the functional relationship between the pose matrix and the arc parameters is to be established, as depicted in Figure 6. Drawing upon the knowledge of geometry, the functional relationship can be expressed by the following equation:

$$\phi(\vec{q}) = \arctg \frac{y_2}{x_2} \tag{21}$$

$$\theta(\vec{q}) = 2 * \arctg \left( \frac{x_2}{z_2 * \cos \phi} \right) \tag{22}$$

$$r(\vec{q}) = \frac{z_2}{\sin \theta} \tag{23}$$

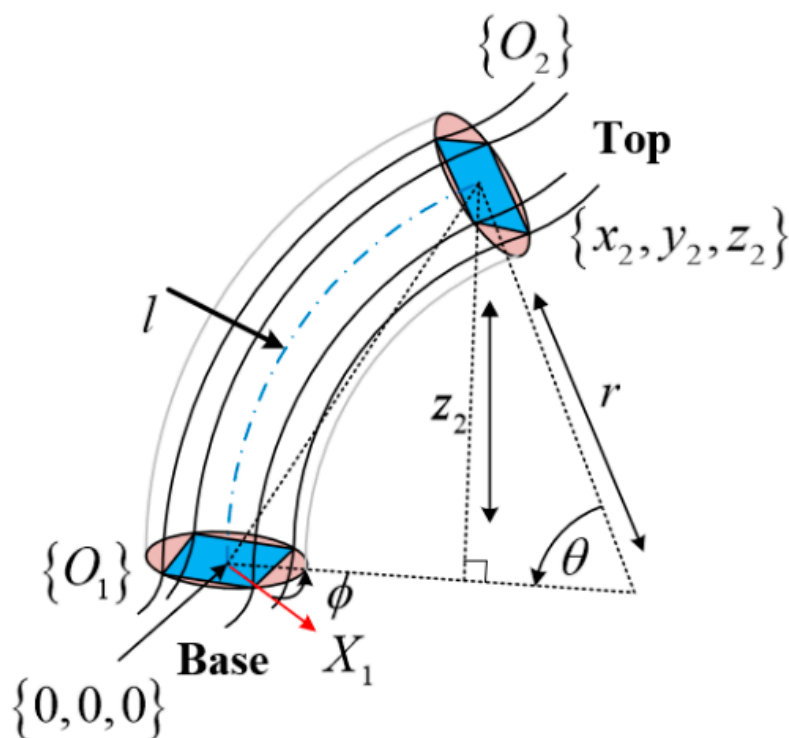


Figure 6. Geometric diagram of the inverse kinematics model of the continuum node.

In the second step, the functional relationship between the arc parameters and the length of each PMA is to be determined. In combination with Figure 6, an auxiliary plane N is designed, which passes through the lowest point M at the top of the node  $\{O_{i+1}\}$  and is perpendicular to each PMA, see Figure 7a. A straight line is drawn through the point  $A_1, A_2, A_3, A_4, O$ , which is perpendicular to the plane N, and the height of  $A_1, A_2, A_3, A_4, O$  from the plane N is denoted as  $h_{1\phi}, h_{2\phi}, h_{3\phi}, h_{4\phi}, h_{O\phi}$ , as shown in Figure 7b.

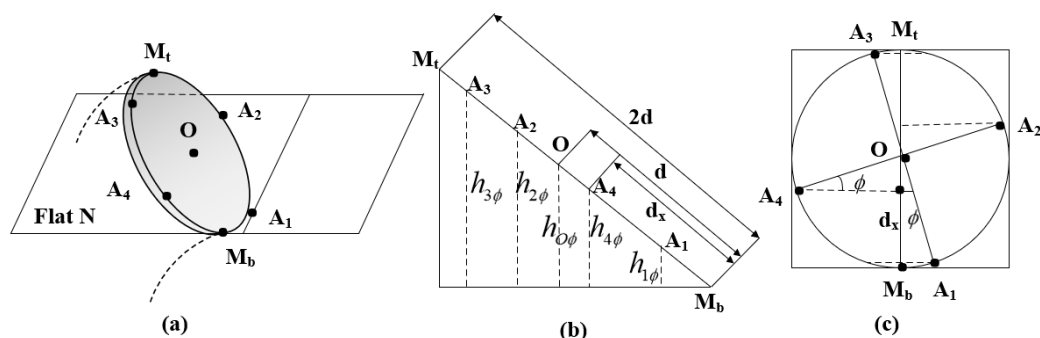


Figure 7. (a) Auxiliary plane N. (b) Projection view of  $h_{1\phi}, h_{2\phi}, h_{3\phi}, h_{4\phi}, h_{O\phi}$ . (c) Projection view of  $A_1, A_2, A_3, A_4$ .

Combined with Figure 6, the height from the center of the circle to the plane N is obtained as follows:

$$h_{O\phi} = d \sin\left(\frac{\theta}{2}\right) \tag{24}$$

It can be seen from Figure 7b that  $h_{1\phi}, h_{2\phi}, h_{3\phi}, h_{4\phi}, h_{O\phi}$  and  $A_1, A_2, A_3, A_4, O$  are proportional to the distance  $d_x$  from point  $M_b$  in the projected view.

$$\frac{h_{x\phi}}{h_{O\phi}} = \frac{d_x}{d} \tag{25}$$

From Figure 7c, the distances from point  $A_1, A_2, A_3, A_4$  to the straight line  $M_t M_b$  in the projected view can be expressed as the following equation:

$$\begin{cases} d_1 = d(1 - \cos \phi) \\ d_2 = d(1 + \sin \phi) \\ d_3 = d(1 + \cos \phi) \\ d_4 = d(1 - \sin \phi) \end{cases} \quad (26)$$

Hence,

$$\begin{cases} h_{1\phi} = d(1 - \cos \phi) \sin\left(\frac{\theta}{2}\right) \\ h_{2\phi} = d(1 + \sin \phi) \sin\left(\frac{\theta}{2}\right) \\ h_{3\phi} = d(1 + \cos \phi) \sin\left(\frac{\theta}{2}\right) \\ h_{4\phi} = d(1 - \sin \phi) \sin\left(\frac{\theta}{2}\right) \end{cases} \quad (27)$$

The chord length from the base  $\{O_1\}$  to the plane N in the bending plane is defined as  $\chi_\phi$ ,  $\chi_\phi$  and can be denoted as:

$$\chi_\phi = 2 \sin\left(\frac{\theta}{2}\right)(r - d) \quad (28)$$

From the definition of plane N, the relationship between the chord lengths  $h_{1\phi}, h_{2\phi}, h_{3\phi}, h_{4\phi}$ , and  $\ell_1, \ell_2, \ell_3, \ell_4$  of PMA can be obtained as:

$$\begin{cases} \ell_1 = \chi_\phi + 2h_{1\phi} = 2 \sin\left(\frac{\theta}{2}\right)(r - d \cos \phi) \\ \ell_2 = \chi_\phi + 2h_{2\phi} = 2 \sin\left(\frac{\theta}{2}\right)(r - d \sin \phi) \\ \ell_3 = \chi_\phi + 2h_{3\phi} = 2 \sin\left(\frac{\theta}{2}\right)(r + d \cos \phi) \\ \ell_4 = \chi_\phi + 2h_{4\phi} = 2 \sin\left(\frac{\theta}{2}\right)(r + d \sin \phi) \end{cases} \quad (29)$$

The relationship between the chord length of PMA and the length of each PMA is:

$$l_j = \frac{\ell_j \theta}{2 \sin \frac{\theta}{2}} \quad (30)$$

Eventually, the length of each PMA is obtained as:

$$\begin{cases} l_1 = \theta r - \theta d \cos \phi \\ l_2 = \theta r - \theta d \sin \phi \\ l_3 = \theta r + \theta d \cos \phi \\ l_4 = \theta r + \theta d \sin \phi \end{cases} \quad (31)$$

The inverse solution process of the multisection continuum arms is basically the same as the single continuum node, and will not be repeated in this paper.

**Remark 4.** Note that Equation (31) is consistent with Equations (4) and (5). Nonetheless, two geometric methods to model the forward and inverse kinematics of the multisection continuum arms are proposed in this paper.

The kinematic modeling errors are presented in Table 1. A total of 50 groups of experiments were carried out in this paper, and some of the results were selected for display. Compared with the ideal length of PMA, the actual length of PMA is obtained through the inverse kinematics solution. It can be intuitively concluded that the kinematic modeling accuracy described in this paper can reach 99.987% where the validity of the kinematic model is verified by experiments.

**Table 1.** Kinematic modeling error.

The Actual Length of PMA/m	The Ideal Length of PMA/m	Error/m
0.31254066	0.3125	0.00004066
0.32196066	0.3225	−0.00053934
0.33228767	0.3325	−0.00021233
0.34226136	0.3425	−0.00023864
0.35197250	0.3525	−0.0005275
0.36253373	0.3625	0.00003373
0.31254066	0.3125	0.00004066
0.32196066	0.3225	−0.00053934
0.33228767	0.3325	−0.00021233

2.2. Dynamic Model with Unknown Parameters

It is indispensable for the continuum arms to interact with the external environment in practical applications. In more detail, this interaction is not only reflected at the cognitive level (information transfer), but also at the physical level (interaction torque). The arms' movement and control stability will be affected by the interaction torque generated by contact. Therefore, the main disturbance types are analyzed in the whole continuum arms, and on this basis, the dynamic model of multisection continuum arms with unknown model parameters is established.

The uncertainties in the continuum arms mainly include external disturbances, interaction torques, model parameter errors and dynamic changes.  $d_{ex}(t)$  is defined as the external disturbance and model error of the system.  $d_{em}(t)$  is represented as the interaction torque generated in the robot space which can be expressed as:

$$d_{em}(t) = J^T F_{int}(t) \tag{32}$$

where  $J^T$  is the Jacobian matrix, and  $F_{int}(t)$  is the force generated by the external environment on the end of the robot, that is, the interaction force.

**Remark 5.** Considering the periodic motion of the ends of the continuum arms, the model parameter error and the interaction torque also exhibit periodic characteristics, which can be regarded as a part of the ideal model parameters obtained by iterative learning. Therefore, the specific value of the Jacobian matrix is not solved in this paper. According to the above analysis, the lumped disturbance is represented by  $d_{lump}(t)$  of the continuum arms. Ultimately, the dynamic equation of the multisection continuum arms under the condition of unknown model parameters can be expressed as:

$$M_x(q)\ddot{q}(t) + C_x(q, \dot{q})\dot{q}(t) + G_x(q) + d_{lump}(t) = u(t) \tag{33}$$

where  $q(t), \dot{q}(t),$  and  $\ddot{q}(t)$  denote the state vectors of position, velocity, and acceleration of each PMA.  $u(t) \in R^{8 \times 1}$  indicates the output torque of each PMA.  $M_x(q) \in R^{8 \times 8}$  denotes the inertial matrix.  $C_x(q, \dot{q}) \in R^{8 \times 8}$  represents the Coriolis/centrifugal matrix.  $G_x(q) \in R^{8 \times 1}$  indicates the gravity vector.

**Remark 6.** The position vector set of the PMA in the previous section is represented by  $\vec{q}_i(t) = [l_1(t), l_2(t), l_3(t), l_4(t)]^T, i = 1, 2$ . For the convenience of expression in the following text,  $q(t) \in R^{8 \times 1}$  is uniformly applied to represent the position matrix of the PMA.

The proposed dynamic equation has the following properties:

**Property 1.** The inertia matrix  $M_x(q) \in R^{8 \times 8}$  is a symmetric positive definite matrix which exists in the upper and lower bounds.

$$0 < \delta_{\min}(M_x) \leq \|M_x(q)\| \leq \delta_{\max}(M_x) \tag{34}$$

where  $\delta_{\min}(M_x)$  is the smallest eigenvalue of the positive definite matrix  $M_x$ ;  $\delta_{\max}(M_x)$  denotes the largest eigenvalue of the positive definite matrix  $M_x$ .  $\|\cdot\|$  stands for the Euclidean norm operation.

**Property 2.**  $\dot{M}_x(q) - 2C_x(q, \dot{q})$  is an obliquely symmetric matrix.

$$\sigma^T [\dot{M}_x(q) - 2C_x(q, \dot{q})] \sigma = 0 \tag{35}$$

**Property 3.** The gravity matrix has an upper bound.

$$G_x(q) \leq o_{\max} \tag{36}$$

For the convenience of subsequent expressions, Formula (33) is rewritten as:

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = -M_x^{-1}(x_1) (C_x(x)x_2(t) + G_x(x_1) + d_{lump}(t)) + M_x^{-1}(x_1)u(t) \end{cases} \tag{37}$$

where the state variable is  $x_1(t) = q(t), x_2(t) = \dot{q}(t)$ .

The expected position trajectories of the eight PMAs at the  $k$ th iteration are denoted as  $x_{ck}(t) \in R^{8 \times 1}$ . It should be noted that, to obtain the desired motion trajectory  $x_{ck}(t)$  in the joint space, the position-based impedance control is implemented to correct the trajectory of the robot end. Then, the inverse kinematics solution is utilized to obtain  $x_{ck}(t)$ . Then, the position and velocity tracking errors of each iteration can be expressed as:

$$s_{1k}(t) = x_{ck}(t) - x_{1k}(t), \dot{s}_{1k}(t) = \dot{x}_{ck}(t) - \dot{x}_{2k}(t) \tag{38}$$

**Remark 7.** For the convenience of the following description,  $M_x(q), C_x(q, \dot{q}), G_x(q), d_{lump}(t)$  is replaced by  $M_{xk}, C_{xk}, G_{xk}, d_{lump,k}(t)$ , respectively.

### 2.3. Problem Formulation

The position-tracking control problem studied can be expressed as follows: The control law  $u(t)$  is designed to guarantee that the position-tracking error of each PMA can stably converge to 0 and not exceed the given time-varying constraint function  $\psi(t)$  under the influence of unknown model parameters and external lumped disturbances. Hence, the control requirement can be described as:

$$\lim_{t \rightarrow \infty} \|s_{1k}(t)\| = 0, \forall t \in [0, +\infty) \ \& \ \|s_{1k}(t)\| \leq \psi(t), \|s_{1k}(t)\| \in R^{8 \times 1} \tag{39}$$

Moreover, as for ensuring that the ideal static balance force (ISBF)  $F_{id}$  remains constant when the arms interact with the external environment, a position-based impedance control strategy is investigated, and the force control requirements are achieved by correcting the expected trajectory in the Cartesian space.

$$F_{id} = f_{con} \tag{40}$$

where  $f_{con}$  is a constant.

### 3. Controller Design

The dynamic equation of the multisection continuum arms has been presented in the previous section. The proposed arms are characterized by high nonlinearity, strong time-varying coupling, creep, and uncertainty, which cause the controller design to be



extremely difficult under the existing conditions. In this paper, a double closed-loop control scheme is proposed for a multisection continuum arm driven by eight PMAs. For one thing, the impedance control acts as the outer loop of the controller to ensure that the ISBF stays constant. For another, adaptive ILC is combined with backstepping control as the inner loop of the proposed controller. Specifically, the ILC is employed to estimate the unknown time-varying parameters of the dynamic equation, and the adaptive algorithm is implemented to suppress the upper bound of the lumped disturbance and feed it back to the controller for real-time compensation to enhance the robustness of the continuum arms. Meanwhile, as to guarantee that the position-tracking errors of the arms converge within the given time-varying constraint function, a backstepping control strategy based on the log-type BLF is designed. The control flow chart is depicted in Figure 8.

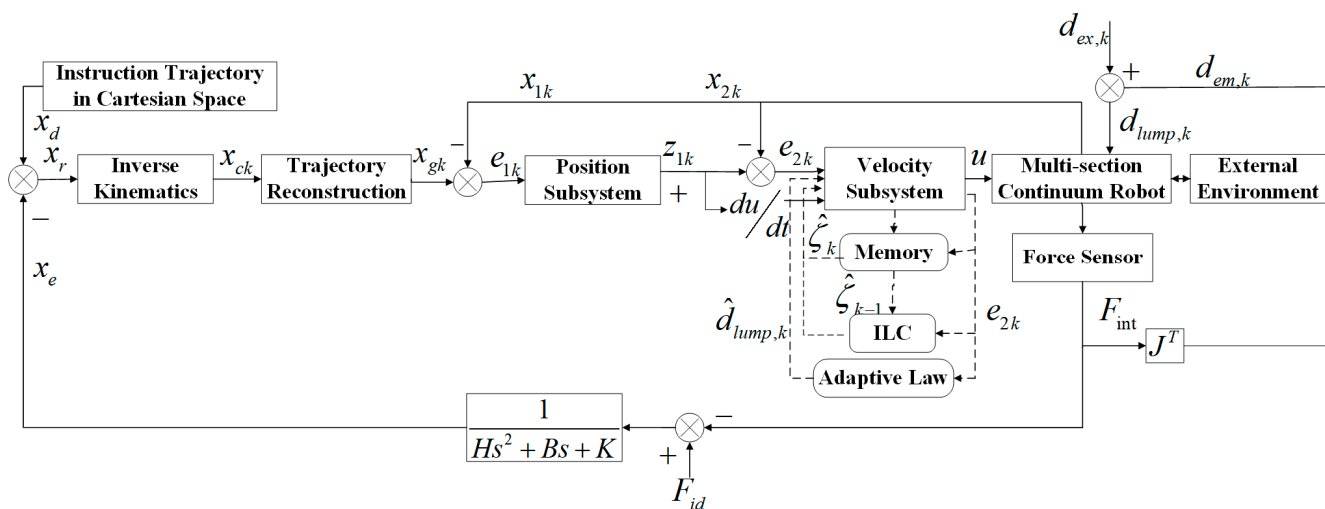


Figure 8. The control flow diagram of the proposed algorithm.

In order to fulfill the desired control requirement, the following assumptions and lemmas are given first, which facility the controller design in subsequent discussion.

**Assumption 1.** The desired trajectory  $x_{ck}(t)$  of the joint space is continuously bounded and second-order differentiable, i.e.,  $\ddot{x}_{ck}(t)$  exists.  $\zeta(t) = [M_{xk}, C_{xk}, G_{xk}]^T$  is defined as the unknown parameter matrix. For a given desired trajectory  $x_{ck}(t)$ , there exists an ideal parameter  $\zeta_d = [M_{xd}, C_{xd}, G_{xd}]^T$ , such that the actual trajectory of each PMA can track the desired trajectory.

**Assumption 2.** The lumped disturbance  $d_{lump,k}(t)$  is a slow time-varying disturbance and is uniformly bounded, i.e.,  $\dot{d}_{lump,k}(t) \approx 0$  and  $\|d_{lump,k}(t)\| \leq D_{max}$ .

**Lemma 1.** The following equations are valid for arbitrary matrices  $U, V, W \in R^{m \times n}$ .

$$tr[(U - V)^T(U - V) - (U - W)^T(U - W)] = tr[(W - V)^T(2(U - V) + (V - W))] \tag{41}$$

where the trace of the matrix is represented by  $tr(\cdot)$ .

**Lemma 2.** For arbitrary functions  $\psi_1(t)$  and  $\psi_2(t)$ , let  $N := R^l \times Z_1 \subset R^{l+1}$  and  $Z_1 := \{z_1 \in R : -\psi_1(t) < z_1 < \psi_2(t)\} \subset R$  be open intervals for the system.

$$\dot{\eta} = h(t, \eta) \tag{42}$$

where  $\eta = [w, z_1]^T \in N$ , and  $C : R^+ \times N \rightarrow R^{l+1}$  are piecewise continuous functions and satisfy the global Lipschitz condition with respect to  $z_1$  in the  $R^+ \times N$  domain. If there exist

continuously differentiable positive definite functions  $Q : R^l \rightarrow R^+$  and  $V_1 : Z_1 \rightarrow R^+$  satisfy the following:

$$z_1 \rightarrow -\psi_1(t) \cup z_1 \rightarrow \psi_2(t), V_1(z_1) \rightarrow \infty \tag{43}$$

$$\underline{\alpha}(\|w\|) \leq Q(w) \leq \bar{\alpha}(\|w\|) \tag{44}$$

where  $\underline{\alpha}$  and  $\bar{\alpha}$  are  $k_\infty$  functions. Define  $V(\eta) = V_1(z_1) + Q(w)$ ,  $z_1(0)$  is in the open set  $(-\psi_1(t), \psi_2(t))$ , if

$$\dot{V} = \frac{\partial V}{\partial \eta} h \leq 0 \tag{45}$$

Then,  $z_1$  will always remain in the open set  $z_1 \in (-\psi_1(t), \psi_2(t))$ .

**Lemma 3.** For any  $\|s_{1k}(t)\| \leq \psi(t)$ , the following inequality holds:

$$\log \frac{\psi^2(t)}{\psi^2(t) - s_{1k}^T(t)s_{1k}(t)} \leq \frac{s_{1k}^T(t)s_{1k}(t)}{\psi^2(t) - s_{1k}^T(t)s_{1k}(t)} \tag{46}$$

### 3.1. Impedance Control Strategy

In this paper, the position-based impedance control strategy is applied to obtain the position deviation under the change of the interaction force so that the desired trajectory is corrected. The equation expression is shown as follows:

$$F_{id} - F_{int}(t) = H(\ddot{x}_d(t) - \ddot{x}_r(t)) + B(\dot{x}_d(t) - \dot{x}_r(t)) + K(x_d(t) - x_r(t)) \tag{47}$$

where  $F_{id}$  denotes the ISBF,  $H, B, K$ , respectively indicates the inertia coefficient, damping coefficient and stiffness coefficient.  $x_d(t), x_r(t)$  represents the command trajectory and desired trajectory in the Cartesian space.

Equation (47) is converted to the transfer function form.

$$x_e(t) = \frac{F_{id} - F_{int}(t)}{Hs^2 + Bs + K} \tag{48}$$

where  $x_e(t) = x_d(t) - x_r(t)$  is the adjustment amount of the command trajectory.

According to the adjustment amount of the command trajectory; the desired trajectory of the continuum arms is obtained as:

$$x_r(t) = x_d(t) - x_e(t) = x_d(t) - \frac{F_{id} - F_{int}(t)}{Hs^2 + Bs + K} \tag{49}$$

**Remark 8.** It can be concluded from Equation (49) that when the interaction force is greater than the ISBF, i.e.,  $F_{int}(t) > F_{id}$ , on account of  $H, B, K > 0$ , the adjustment amount of the command trajectory is obtained as  $x_e(t) < 0$ , i.e.,  $x_d(t) < x_r(t)$ . For instance, the end position of the desired trajectory becomes larger than the end position of the command trajectory while the multisection continuum arms accelerate to stretch or decelerate to contract, which causes the interaction force to decrease, and vice versa.

Eventually, the inverse kinematics solution (2.1.2) is implemented to obtain the desired motion trajectory  $x_{ck}(t)$  of each PMA in the joint space, which is appointed to the tracking target of the position controller.

### 3.2. Adaptive Iterative Learning Backstepping Control Strategy with Initial Error

#### 3.2.1. Reconstruction of the Desired Trajectory

The initial value of the system state is strictly required by traditional ILC, which ensures that the initial state of the system at each iteration is strictly consistent with the initial state of the desired trajectory, i.e.,  $x_{ck}(0) = x_{1k}(0) = x_{10}(0)$ . Nevertheless, in practical applications, the initial state of the continuum arms and the initial state of the desired trajectory are hardly guaranteed to be equal in the iterative domain. Hence, as for the aforementioned phenomenon, the initial condition of ILC is released by the trajectory reconstruction approach when a fixed deviation permanently exists between the initial state of the system and the initial state of the desired trajectory.

The main idea of the trajectory reconstruction approach is that the initial value of the reconstructed trajectory is equal to the initial value of the state in the trajectory operating interval  $[0, T]$ . Meanwhile, the reconstructed trajectory at time  $[\rho, T]$  is the same as the original desired trajectory. The reconstructed trajectory form is shown as follows:

$$x_{gk}(t) = x_{ck}(t) - \Gamma_1(t)(x_{ck}(0) - x_{1k}(0)) - \Gamma_2(t)(\dot{x}_{ck}(0) - \dot{x}_{2k}(0)) \tag{50}$$

where  $\Gamma_1(t), \Gamma_2(t)$  is the time-varying function which is not unique. However, the following conditions are required to be satisfied when the time-varying function is designed.

$$\begin{cases} x_{gk}(0) = x_{1k}(0) \\ x_{gk}(\rho) = x_{ck}(\rho) \\ x_{gk}^{(m)}(\rho) = x_{ck}^{(m)}(\rho) \end{cases} \tag{51}$$

$(\cdot)^{(m)}$  represents the m-order derivative of the function, and  $m \geq 2$  in this paper.

The time-varying function  $\Gamma(t)$  can be designed as:

$$\Gamma_1(t) = \begin{cases} (1 - \frac{t}{\rho})^3, & t \in [0, \rho) \\ 0, & t \in [\rho, T] \end{cases} \tag{52}$$

$$\Gamma_2(t) = \begin{cases} t * (1 - \frac{t}{\rho})^3, & t \in [0, \rho) \\ 0, & t \in [\rho, T] \end{cases} \tag{53}$$

#### 3.2.2. Adaptive Iterative Learning Backstepping Controller

**Step 1:** To implement  $\|s_{1k}(t)\| \leq \psi(t), \forall t > 0$ , the following log-type BLF is defined as:

$$V_{1k}(t) = \frac{1}{2} \log \frac{\psi^2(t)}{\psi^2(t) - s_{1k}^T(t)s_{1k}(t)} \tag{54}$$

Then, its derivative with respect to time can be expressed as:

$$\dot{V}_{1k}(t) \leq \frac{s_{1k}^T(t)\dot{s}_{1k}(t) - Q_1 s_{1k}^T(t)s_{1k}(t)}{\psi^2(t) - s_{1k}^T(t)s_{1k}(t)} \tag{55}$$

where  $Q_1 = \sup_t \sqrt{\left(\frac{\dot{\psi}(t)}{\psi(t)}\right)^2} + \epsilon_k, \epsilon_k > 0$  is the auxiliary variable.

Then, introduce the stable function  $z_{1k}(t)$  to be designed, while the position-tracking error and the tracking error of the stable function is defined as:

$$s_{1k}(t) = x_{gk}(t) - x_{1k}(t), s_{2k}(t) = z_{1k}(t) - x_{2k}(t) \tag{56}$$

Equation (55) can be expressed as:

$$\dot{V}_{1k}(t) \leq \frac{s_{1k}^T(t) (\dot{x}_{gk}(t) + s_{2k}(t) - z_{1k}(t)) - Q_1 s_{1k}^T(t) s_{1k}(t)}{\psi^2(t) - s_{1k}^T(t) s_{1k}(t)} \quad (57)$$

As to ensure the stability of the system, the stability function  $z_{1k}(t)$  is designed as:

$$z_{1k}(t) = \dot{x}_{gk}(t) + k_1 s_{1k}(t) (\psi^2(t) - s_{1k}^T(t) s_{1k}(t)) - Q_1 s_{1k}(t), k_1 > 0 \quad (58)$$

Substitute the above formula into  $\dot{V}_{1k}(t)$

$$\dot{V}_{1k}(t) \leq -k_1 s_{1k}^T(t) s_{1k}(t) + \frac{s_{1k}^T(t) s_{2k}(t)}{\psi^2(t) - s_{1k}^T(t) s_{1k}(t)} \quad (59)$$

If  $s_{2k}(t) = 0$ ,  $\dot{V}_{1k}(t) \leq -k_1 s_{1k}^2(t)$ . Hence, the controller needs to be designed in step 2.

**Step 2:** Since the state variables in Equation (37) do not need to be constrained, the following Lyapunov function is defined as:

$$V_{2k}(t) = V_{1k}(t) + \frac{1}{2} s_{2k}^T(t) M_{xk} s_{2k}(t) + \frac{1}{2\gamma} \tilde{d}_{lump,k}^T(t) \tilde{d}_{lump,k}(t) \quad (60)$$

where  $\gamma > 0$  is the adaptive step size.

The adaptive estimation error of the lumped disturbance is:

$$\tilde{d}_{lump,k}(t) = d_{lump,k}(t) - \hat{d}_{lump,k}(t) \quad (61)$$

$\hat{d}_{lump,k}(t)$  denotes the adaptive estimation value of the lumped disturbance. From Assumption 2, it can be known that the lumped disturbance is a slow time-varying disturbance, hence,  $\dot{\tilde{d}}_{lump,k}(t) = -\dot{\hat{d}}_{lump,k}(t)$ . Then, derive Equation (60) with respect to time  $t$ .

$$\begin{aligned} \dot{V}_{2k}(t) &\leq -k_1 s_{1k}^T(t) s_{1k}(t) + \frac{s_{1k}^T(t) s_{2k}(t)}{\psi^2(t) - s_{1k}^T(t) s_{1k}(t)} + s_{2k}^T(t) M_{xk} \dot{s}_{2k}(t) \\ &\quad + \frac{1}{2} s_{2k}^T(t) \dot{M}_{xk} s_{2k}(t) + \frac{1}{\gamma} \tilde{d}_{lump,k}^T(t) \dot{\tilde{d}}_{lump,k}(t) \\ &\leq -k_1 s_{1k}^T(t) s_{1k}(t) + s_{2k}^T(t) [M_{xk} \dot{z}_{1k}(t) + C_{xk} x_{2k}(t) + G_{xk} + d_{lump,k}(t) - u(t)] \\ &\quad + \frac{s_{1k}^T(t) s_{2k}(t)}{\psi^2(t) - s_{1k}^T(t) s_{1k}(t)} + \frac{1}{2} s_{2k}^T(t) \dot{M}_{xk} s_{2k}(t) - \frac{1}{\gamma} \tilde{d}_{lump,k}^T(t) \dot{\hat{d}}_{lump,k}(t) \end{aligned} \quad (62)$$

Combined with Property 2 of dynamic Equation (33), it can be concluded that  $s_{2k}^T(t) [M_{xk} - 2C_{xk}] s_{2k}(t) = 0$ . Therefore, the above formula can be simplified to:

$$\begin{aligned} \dot{V}_{2k}(t) &\leq -k_1 s_{1k}^T(t) s_{1k}(t) + \frac{s_{1k}^T(t) s_{2k}(t)}{\psi^2(t) - s_{1k}^T(t) s_{1k}(t)} - \frac{1}{\gamma} \tilde{d}_{lump,k}^T(t) \dot{\hat{d}}_{lump,k}(t) \\ &\quad + s_{2k}^T(t) [\zeta_d^T(t) \Phi_k(t) + d_{lump,k}(t) - u(t)] \end{aligned} \quad (63)$$

where  $\zeta_d(t) = [M_{xk}, C_{xk}, G_{xk}]^T$  represents the ideal unknown parameter matrix.  $\Phi_k(t) = [z_{1k}^T(t), z_{1k}^T(t), 1]^T$  denotes the state matrix of the  $k$ th iteration.

Ultimately, the control law is designed as:

$$u(t) = \hat{\zeta}_k^T(t) \Phi_k(t) + \hat{d}_{lump,k}(t) + \frac{s_{1k}(t)}{\psi^2(t) - s_{1k}^T(t) s_{1k}(t)} + k_2 s_{2k}(t) + k_3 \text{sgn}(s_{2k}(t)) \quad (64)$$

where  $\text{sgn}(s_{2k}(t))$  indicates a sign function. The self-adaptive estimated value of the ideal unknown parameter matrix is defined as  $\hat{\zeta}_k(t)$ . With the increase in the number of iterations, the ideal unknown parameter matrix  $\zeta_d(t)$  is gradually approximated to  $\hat{\zeta}_k(t)$ , which has the recursive characteristic.  $\hat{\zeta}_k(t)$  can be expressed by the iterative learning law.

$$\hat{\zeta}_k(t) = \hat{\zeta}_{k-1}(t) + \zeta \Phi_k(t) s_{2k}^T(t) \tag{65}$$

where  $\zeta$  denotes the iterative learning step size.

In Equation (64),  $\hat{d}_{lump,k}(t)$  represents the self-adaptive estimated value of the lumped disturbance  $d_{lump,k}(t)$ . The self-adaptive law is designed as:

$$\dot{\hat{d}}_{lump,k}(t) = \gamma s_{2k}(t) \tag{66}$$

where  $\gamma$  denotes the adaptive step size.

Substitute Equation (64) into (62).

$$\dot{V}_{2k}(t) \leq -k_1 \|s_{1k}(t)\|^2 - k_2 \|s_{2k}(t)\|^2 - k_3 \text{sgn}(s_{2k}(t)) + s_{2k}^T(t) \tilde{\zeta}_k^T(t) \Phi_k(t) \tag{67}$$

The adaptive estimation error of the ideal unknown parameter matrix is proposed as:

$$\tilde{\zeta}_k(t) = \zeta_d(t) - \hat{\zeta}_k(t) \tag{68}$$

If the parameter estimation matrix  $\hat{\zeta}_k(t)$  is capable of stably converging to  $\zeta_d(t)$  in the iterative domain, it can be summarized in Equation (67) that the continuum arms can accurately track the reconstructed desired trajectory  $x_{gk}(t)$ .

#### 4. Convergence Analysis

**Theorem 1.** *In this paper, the control law (64), iterative learning law (65), and adaptive law (66) are designed to solve the trajectory tracking control problem of the multisection continuum arms (33) under the condition of unknown model parameters. As the number of iterations  $k$  increases, the ideal parameter matrix  $\zeta_d(t)$  is approximated to the parameter estimation matrix  $\hat{\zeta}_k(t)$  with arbitrary precision. Moreover, the tracking errors  $s_{1k}(t)$  and  $s_{2k}(t)$  can stably converge to 0 in the time period  $[\rho, T]$ , and the tracking error is limited by the time-varying constraint function  $\psi(t)$  within  $\forall t \geq 0$ .*

**Proof.** To evaluate the convergence of the proposed control scheme in the iterative and time domains, comprehensive energy function (CEF) is constructed as:

$$\Omega_k(t) = \sum_{i=1}^4 V_k^i(t), i = 1, 2, 3, 4 \tag{69}$$

where

$$V_k^1(t) = \frac{1}{2} \log \frac{\psi^2(t)}{\psi^2(t) - s_{1k}^T(t) s_{1k}(t)} \tag{70}$$

$$V_k^2(t) = \frac{1}{2} s_{2k}^T(t) M_{xk} s_{2k}(t) \tag{71}$$

$$V_k^3(t) = \frac{1}{2\gamma} \tilde{d}_{lump,k}^T(t) \tilde{d}_{lump,k}(t) \tag{72}$$

$$V_k^4(t) = \frac{1}{2\zeta} \int_0^t \text{tr}(\tilde{\zeta}_k^T(t) \tilde{\zeta}_k(t)) \square \tag{73}$$

**Remark 9.** *In practical application, the preconditions of Equations (59) and (63) need to be satisfied when the CEF is designed. Moreover, a new type of Lyapunov convergence proof is implemented drawing upon Lemma 1 without a unique approach.*

**Step 1:** Prove that the CEF is bounded.

The difference between the CEF of the  $k$ th and the  $(k - 1)$ th iteration is obtained as:

$$\Delta\Omega_k(t) = \Delta V_k^1(t) + \Delta V_k^2(t) + \Delta V_k^3(t) + \Delta V_k^4(t) \tag{74}$$

The BLF  $\Delta V_k^1(t)$  of the above formula can be expressed as:

$$\begin{aligned} \Delta V_k^1(t) &= \frac{1}{2} \log \frac{\psi^2(t)}{\psi^2(t) - s_{1k}^T(t)s_{1k}(t)} - \frac{1}{2} \log \frac{\psi^2(t)}{\psi^2(t) - s_{1k-1}^T(t)s_{1k-1}(t)} \\ &\leq \frac{1}{2} \log \frac{\psi^2(0)}{\psi^2(0) - s_{1k}^T(0)s_{1k}(0)} - \frac{1}{2} \log \frac{\psi^2(t)}{\psi^2(t) - s_{1k-1}^T(t)s_{1k-1}(t)} \\ &\quad + \int_0^t \left[ -k_1 s_{1k}^T(\tau)s_{1k}(\tau) + \frac{s_{1k}^T(\tau)s_{2k}(\tau)}{\psi^2(\tau) - s_{1k}^T(\tau)s_{1k}(\tau)} \right] d\tau \end{aligned} \tag{75}$$

$s_{1k}(0) = x_{gk}(0) - x_{1k}(0) = 0$  is known from Equation (50). Hence, the above formula can be simplified as follows:

$$\Delta V_k^1(t) \leq -\frac{1}{2} \log \frac{\psi^2(t)}{\psi^2(t) - s_{1k-1}^T(t)s_{1k-1}(t)} + \int_0^t \left[ -k_1 s_{1k}^T(\tau)s_{1k}(\tau) + \frac{s_{1k}^T(\tau)s_{2k}(\tau)}{\psi^2(\tau) - s_{1k}^T(\tau)s_{1k}(\tau)} \right] d\tau \tag{76}$$

The second term  $\Delta V_k^2(t)$  in equation (74) can be written as:

$$\begin{aligned} \Delta V_k^2(t) &= \frac{1}{2} s_{2k}^T(t)M_{xk}s_{2k}(t) - \frac{1}{2} s_{2k-1}^T(t)M_{xk-1}s_{2k-1}(t) \\ &= \frac{1}{2} s_{2k}^T(0)M_{xk}s_{2k}(0) - \frac{1}{2} s_{2k-1}^T(t)M_{xk-1}s_{2k-1}(t) \\ &\quad + \int_0^t s_{2k}^T(\tau) \left[ \zeta_d^T(\tau)\Phi_k(\tau) + d_{lump,k}(\tau) - u(\tau) \right] d\tau \end{aligned} \tag{77}$$

From formula (58), we have:

$$s_{2k}(0) = z_{1k}(0) - x_{2k}(0) = \dot{x}_{gk}(0) - x_{2k}(0) = 0 \tag{78}$$

Substituting Equation (78) and the control law into Equation (77), we get:

$$\begin{aligned} \Delta V_k^2(t) &= \int_0^t s_{2k}^T(\tau)\tilde{\zeta}_k^T(\tau)\Phi_k(\tau)d\tau + \int_0^t s_{2k}^T(\tau)\tilde{d}_{lump,k}(\tau)d\tau - \int_0^t \frac{s_{2k}^T(\tau)s_{1k}(\tau)}{\psi^2(\tau) - s_{1k}^T(\tau)s_{1k}(\tau)}d\tau \\ &\quad - \frac{1}{2} s_{2k-1}^T(t)M_{xk-1}s_{2k-1}(t) - \int_0^t k_2 \|s_{2k}(\tau)\|^2 d\tau - \int_0^t k_3 \|s_{2k}(\tau)\| d\tau \end{aligned} \tag{79}$$

For the third term  $\Delta V_k^3(t)$  in Equation (74), we have:

$$\begin{aligned} \Delta V_k^3(t) &= \frac{1}{2\gamma} \tilde{d}_{lump,k}^T(t)\tilde{d}_{lump,k}(t) - \frac{1}{2\gamma} \tilde{d}_{lump,k-1}^T(t)\tilde{d}_{lump,k-1}(t) \\ &= -\frac{1}{\gamma} \int_0^t \tilde{d}_{lump,k}^T(\tau)\dot{\tilde{d}}_{lump,k}(\tau)d\tau - \frac{1}{2\gamma} \tilde{d}_{lump,k-1}^T(t)\tilde{d}_{lump,k-1}(t) \end{aligned} \tag{80}$$

Equation (80) is substituted into the adaptive law (66), we get:

$$\Delta V_k^3(t) = -\int_0^t \tilde{d}_{lump,k}^T(\tau)s_{2k}(\tau)d\tau - \frac{1}{2\gamma} \tilde{d}_{lump,k-1}^T(t)\tilde{d}_{lump,k-1}(t) \tag{81}$$

For the fourth term  $\Delta V_k^4(t)$  in Equation (74)

$$\Delta V_k^4(t) = \frac{1}{2\zeta} \int_0^t \left[ tr\left(\tilde{\zeta}_k^T(\tau)\tilde{\zeta}_k(\tau)\right) - tr\left(\tilde{\zeta}_{k-1}^T(\tau)\tilde{\zeta}_{k-1}(\tau)\right) \right] d\tau \tag{82}$$



From Lemma 1, it can be known that:

$$\begin{aligned} \text{tr}\left(\tilde{\zeta}_k^T(t)\tilde{\zeta}_k(t)\right) - \text{tr}\left(\tilde{\zeta}_{k-1}^T(t)\tilde{\zeta}_{k-1}(t)\right) &= 2\text{tr}\left[\left(\hat{\zeta}_{k-1}(t) - \hat{\zeta}_k(t)\right)^T\left(\zeta_d(t) - \hat{\zeta}_k(t)\right)\right] \\ &\quad - \text{tr}\left[\left(\hat{\zeta}_k(t) - \hat{\zeta}_{k-1}(t)\right)^T\left(\hat{\zeta}_k(t) - \hat{\zeta}_{k-1}(t)\right)\right] \end{aligned} \tag{83}$$

Using Equation (83) and the iterative learning law (65), Equation (82) can be simplified as:

$$\Delta V_k^4 = -\int_0^t \text{tr}\left(s_{2k}(t)\Phi_k^T(t)\tilde{\zeta}_k(t)\right)d\tau - \frac{\zeta}{2}\int_0^t \left(\|\Phi_k(t)\|\|s_{2k}(t)\|\right)^2 d\tau \tag{84}$$

Combining Equations (76), (79), (81), and (84), we have:

$$\begin{aligned} \Delta\Omega_k(t) &\leq -k_1\int_0^t \|s_{1k}(t)\|^2 d\tau - \int_0^t k_2\|s_{2k}(t)\|^2 d\tau - \int_0^t k_3\|s_{2k}(t)\| d\tau \\ &\quad - \frac{1}{2}\log\frac{\psi^2(t)}{\psi^2(t)-s_{1k-1}^T(t)s_{1k-1}(t)} - \frac{1}{2\gamma}\tilde{d}_{lump,k-1}^T(t)\tilde{d}_{lump,k-1}(t) \\ &\quad - \frac{\zeta}{2}\int_0^t \left(\|\Phi_k(t)\|\|s_{2k}(t)\|\right)^2 d\tau - \frac{1}{2}s_{2k-1}^T(t)M_{xk-1}s_{2k-1}(t) \end{aligned} \tag{85}$$

**Remark 10.** The equivalent transformation  $s_{2k}^T(t)\tilde{\zeta}_k^T(t)\Phi_k(t) = \text{tr}\left(s_{2k}(t)\Phi_k^T(t)\tilde{\zeta}_k(t)\right)$  is implemented when the above formula is simplified.

With Equation (85) we can conclude that  $\Omega_k(t)$  is monotonically nonincreasing in the iterative domain. Therefore, when  $\Omega_0(t)$  is bounded, the CEF  $\Omega_k(t)$  is bounded at any number of iterations. Since the CEF conforms to the general properties of the sequence in the iterative domain, it can be obtained from the definition of the sequence.

$$\Omega_k(t) = \Omega_0(t) + \sum_{i=1}^k \Delta\Omega_k(t) \tag{86}$$

Taking the derivative of  $\Omega_0(t)$  with respect to time T. Then, Equation (67) is combined by Equation (73). Finally, the following result can be easily obtained as:

$$\begin{aligned} \dot{\Omega}_0(t) &= \dot{V}_0^1(t) + \dot{V}_0^2(t) + \dot{V}_0^3(t) + \dot{V}_0^4(t) \\ &= -k_1\|s_{10}(t)\|^2 - k_2\|s_{20}(t)\|^2 - k_3\|s_{20}(t)\| \\ &\quad + \text{tr}\left(s_{20}(t)\Phi_0^T(t)\tilde{\zeta}_0(t)\right) + \frac{1}{2\zeta}\text{tr}\left(\tilde{\zeta}_0^T(t)\tilde{\zeta}_0(t)\right) \end{aligned} \tag{87}$$

The iterative learning law  $\hat{\zeta}_0(t) = \zeta\Phi_0(t)s_{20}^T(t)$  is substituted into Equation (87), we get:

$$\dot{\Omega}_0(t) = -k_1\|s_{10}(t)\|^2 - k_2\|s_{20}(t)\|^2 - k_3\|s_{20}(t)\| + \frac{1}{\zeta}\text{tr}\left(\zeta_d^T(t)\tilde{\zeta}_0(t)\right) - \frac{1}{2\zeta}\text{tr}\left(\tilde{\zeta}_0^T(t)\tilde{\zeta}_0(t)\right) \tag{88}$$

In the light of dynamic Equation (33) and the iterative learning law (65), the ideal parameter  $\zeta_d(t)$  is bounded, and the parameter error matrix  $\tilde{\zeta}_0(t)$  of the 0th iteration is continuously bounded within  $[0, T]$ . Hence,  $\dot{\Omega}_0(t)$  exists as an upper bound. The following conclusion can be drawn as:

$$\Omega_0(t) = \Omega_0(0) + \int_0^t \dot{\Omega}_0(t)d\tau \tag{89}$$

Ultimately, it can be concluded that  $\Omega_k(t)$  is strictly bounded in the iterative domain.

**Step 2:** Demonstrate that the system state  $s_{1k}(t), s_{2k}(t)$  converges. From Equation (86), the following conclusion can be easily reached.

$$\Omega_k(t) - \Omega_0(t) = \sum_{i=1}^k \Delta\Omega_k(t) \leq 0 \tag{90}$$

Since each term of the CEF  $V_k^i(k), i = 1, 2, 3, 4$  is a non-negative definite matrix, we get:

$$\begin{aligned} \sum_{i=1}^k \Delta\Omega_k(t) &\leq -k_1 \int_0^t \|s_{1k}(t)\|^2 d\tau - \int_0^t k_2 \|s_{2k}(t)\|^2 d\tau - \int_0^t k_3 \|s_{2k}(t)\| d\tau \\ &\quad - \frac{1}{2} \log \frac{\psi^2(t)}{\psi^2(t) - s_{1k-1}^T(t) s_{1k-1}(t)} - \frac{1}{2\gamma} \tilde{d}_{lump,k-1}^T(t) \tilde{d}_{lump,k-1}(t) \\ &\quad - \frac{c}{2} \int_0^t (\|\Phi_k(t)\| \|s_{2k}(t)\|)^2 d\tau - \frac{1}{2} s_{2k-1}^T(t) M_{xk-1} s_{2k-1}(t) \leq 0 \end{aligned} \tag{91}$$

In conclusion, both  $\|s_{1k}(t)\|$  and  $\|s_{2k}(t)\|$  will eventually be forced to 0 when  $k \rightarrow \infty$ . It can be seen from Equation (59) that when  $\lim_{k \rightarrow \infty} \|s_{1k}(t)\| = \lim_{k \rightarrow \infty} \|s_{2k}(t)\| = 0$ , the BLF satisfies the condition  $\dot{V}_{1k}(t) \leq 0$  where  $s_{1k}(t)$  is constrained by the time-varying constraint function  $\psi(t)$  during the convergence process. Thus far, the convergence proof process of the closed-loop system has been fully demonstrated.

### 5. Simulation Results

To evaluate the validity and superiority of the proposed scheme, the performance of the model and controller was verified via cosimulation. Above all, the cosimulation environment of a virtual prototype was configured by utilizing the ANSYS/ADAMS/MATLAB software. Cosimulation experiments of the virtual prototype were run on a workstation with GTX1050Ti GPU, 16G memory, and an Intel Core i5 CPU for implementation. Then, the expected sinusoidal trajectory with the initial error was employed as the control input of each PMA, and comparative studies with other algorithms were carried out.

In this paper, the finite element analysis of PMA was first carried out by ANSYS software. Subsequently, the generated flexible body was imported into ADAMS software to connect with the rigid body structure. Secondly, the parameters of the multisection continuum arms and the physical simulation environment can be flexibly configured through ADAMS. Finally, the numerical model of the controlled object was generated through the control plants and imported into MATLAB/Simulink for cosimulation.

**Remark 11.** To further increase the uncertainty and external disturbance of the control system, the bottom of the continuum arms is connected to the SCARA type 2-DOF manipulator. The rigid-flexible coupling system is partially different from the structure shown in Figure 1. Nonetheless, the final conclusion is not impacted by model simplification. The model of the cosimulation is executed, as shown in Figure 9.

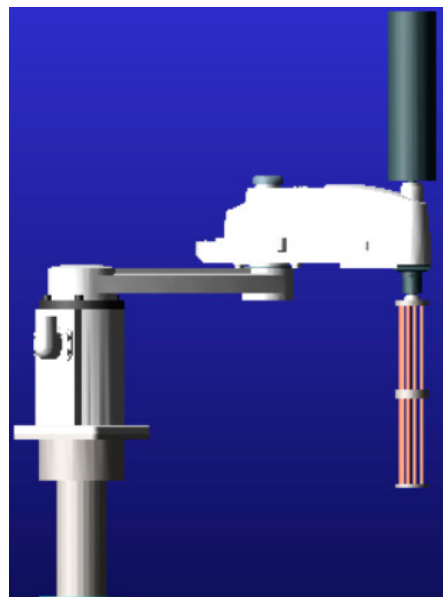


Figure 9. Cosimulation model of continuum robot system.

**Remark 12.** To simulate the flexible deformation process of the continuum arms, a single PMA is transferred into ANSYS for preprocessing. First, the material properties of PMA are composed of super-elastic rubber material, which guarantees the flexibility of the pneumatic muscle for one thing, and restrains the radial deformation of the pneumatic muscle during inflation for another. After the material parameters of PMA are obtained, mesh division is performed. Afterwards, the upper and lower ends of the PMA are coupled with mass21 for rigid regions [32,33]. Eventually, the preprocessing model (.mnf) is generated and imported into ADAMS (as shown in Figure 10).

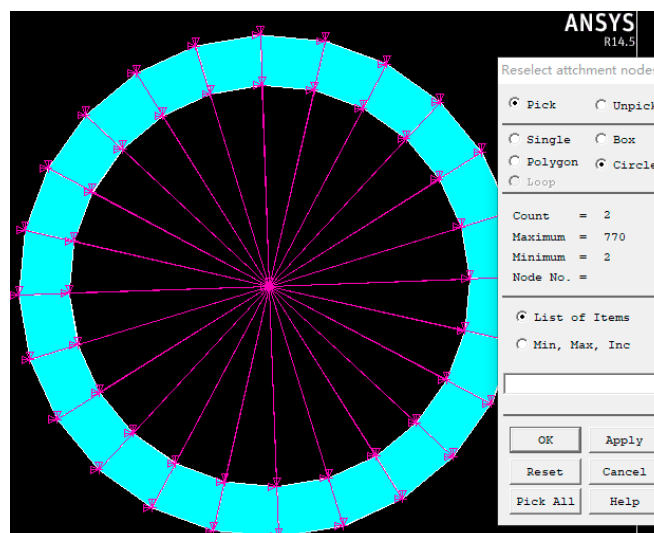


Figure 10. The preprocessing model (.mnf) obtained from ANSYS.

For the multisection continuum arms system, the gravity direction is specified as the negative direction of the Y-axis. The material property of the rigid part is defined as structural steel. The command trajectory  $x_d(t)$  of the end of the multisection continuum arms is set as follows:

$$x_d(t) = a_0 + a_1 * \cos(t * w) + b_1 * \sin(t * w) \tag{92}$$

where  $a_0 = -0.03537$ ,  $a_1 = 0.04735$ ,  $b_1 = 0.04688$ , and  $w = 5.7702$ . The IBSF is set as  $F_{id} = 3$ , and the man-machine interaction force is selected as  $F_{int} = 5 \sin(1.5\pi ft) + 4$ . Moreover, the impedance coefficient is selected as  $K = 100$ ,  $B = 15$ , and  $H = 0.22$ . In the light of Equation (48), the adjustment amount  $x_e(t)$  of the command trajectory is obtained as follows:

$$x_e(t) = a * \exp(b * t) + c * \exp(d * t) \tag{93}$$

where  $a = -0.00668$ ,  $b = 0.05365$ ,  $c = 0.006695$ , and  $d = -7.35$ . The desired trajectory of the end of the continuum arms is obtained by Equation (49).

$$x_r(t) = a_0 + a_1 * \cos(t * w) + b_1 * \sin(t * w) \tag{94}$$

where  $a_0 = -0.02467$ ,  $a_1 = 0.02465$ ,  $b_1 = 0.07891$ , and  $w = 0.1532$ . The desired trajectory  $x_{ck}(t) \in R^{2 \times 1}$  of the SCARA manipulator is set as:

$$x_{ck1,2}(t) = \sin\left(\frac{\pi t}{50}\right) \tag{95}$$

The desired motion trajectory  $x_{ck}(t)$  of each PMA in the joint space is obtained from the kinematic inverse model proposed in Section 2.1.2, and its mathematical expression is:

$$\begin{cases} x_{ck3,4,7,8}(t) = 0.2150 - \sin\left(\frac{\pi t}{25}\right) * 0.05 \\ x_{ck5,6,9,10}(t) = 0.2150 - \sin\left(\frac{\pi t}{50}\right) * 0.05 \end{cases} \tag{96}$$

To test the effectiveness of the trajectory reconstruction method in the presence of initial errors, the initial joint position of the multisection continuum arms and the SCARA manipulator in the  $k$ th iteration is set to  $x_{1k}(0) = x_{2k}(0) = [0, \dots, 0]^T \in R^{10 \times 1}$ . It can be seen that the initial value of the desired trajectory is not equal to the system state. Afterwards, the reconstructed desired trajectory  $x_{gk}(t) \in R^{10 \times 1}$  is obtained by Equation (50).

The corresponding control parameters are listed in Table 2.

**Table 2.** Control parameters.

Controllers	Parameters
End time of trajectory reconstruction	$\rho = 0.5$
Simulation time	$T = 5$
Iterative learning step	$\zeta = 0.5$
Adaptive step	$\gamma = 0.2$
Control gains	$k_1 = 10, k_2 = 20, k_3 = 0.5$
Number of iterative learning	$k = 10$
Time-varying constraint function	$\psi(t) = e^{-2t} + 0.01$
Lumped disturbance	$d_{lump}(t) = 0.1 * (t + \sin(0.1 * t)) + 2$
Auxiliary parameter	$Q_1 = 2$

**Remark 13.** The selection of control parameters is supposed to be abided by the following approaches. It should be emphasized that the control variate method is adopted for the parameter adjustment. Firstly, the adaptive step  $\gamma$  can be appropriately increased, which can effectively compensate for lumped disturbance and smooth the input torque in the iterative domain. Then, increasing the iterative learning step  $\zeta$  can speed up the convergence of the iterative domain, generally not exceeding 1. Moreover, increasing  $k_1$  slowly is capable of enhancing the system convergence accuracy and efficiency but has a negative impact on system stability. Correspondingly, increasing  $k_2$  can stabilize the system and ease chatter phenomena in the time domain. Particularly,  $k_2$  is greater than  $k_1$  and  $k_3$  should not exceed 1. Last but not least, the formulation of the time-varying constraint function counts on the desired maximum error in practical systems.

To validate the superiority of the proposed control scheme, comparative experiments are conducted by designing an impedance PD controller and an impedance iterative learn-

ing sliding mode controller (IILSMC) in this paper. The impedance PD control parameters are specified as follows:

$$\begin{bmatrix} P \\ D \end{bmatrix} = \begin{bmatrix} 10^{-4} & 10^{-4} & 10^{-3} & 10^{-3} & 10^{-3} & 10^{-3} & 10^{-3} & 10^{-3} & 10^{-3} & 10^{-3} \\ 10^{-2} & 10^{-2} & 10^{-1} & 10^{-1} & 10^{-1} & 10^{-1} & 10^{-1} & 10^{-1} & 10^{-1} & 10^{-1} \end{bmatrix} \quad (97)$$

The IILSMC control law and control parameters are set as follows

$$\begin{cases} u_k(t) = \hat{\xi}_k^T(t)I_k(t) + \hat{d}_{lump,k}(t) + k_1s_k(t) + k_2\text{sgn}(s) \\ \hat{d}_{lump,k}(t) = s_k(t) \\ \hat{\xi}_k(t) = \hat{\xi}_{k-1}(t) + qI_k(t)s_k^T(t), \hat{\xi}_{-1}(t) = 0 \end{cases} \quad (98)$$

where  $s_k(t) = c_1e_{1k}(t) + e_{2k}(t)$ ,  $x_{ak}(t) = c_1e_{1k}(t) + \dot{x}_{gk}(t)$ , and  $I_k(t) = [\dot{x}_{ak}^T(t), x_{ak}^T(t), 1]^T$  to ensure the persuasion, except that the control gain parameter is specified to  $c_1 = [2, 2, 3, 3, 3, 3, 3, 3, 3, 3]^T$ ,  $q = 1$ ,  $\gamma = [10, 2, 1, 1, 1, 1, 1, 1, 1, 1]^T \times 100$ ,  $k_1 = [10, 10, 8, 8, 8, 8, 8, 8, 8, 8]^T$ , and  $k_2 = 2$ , other simulation conditions are the same. The simulation results are depicted in Figures 11–18.

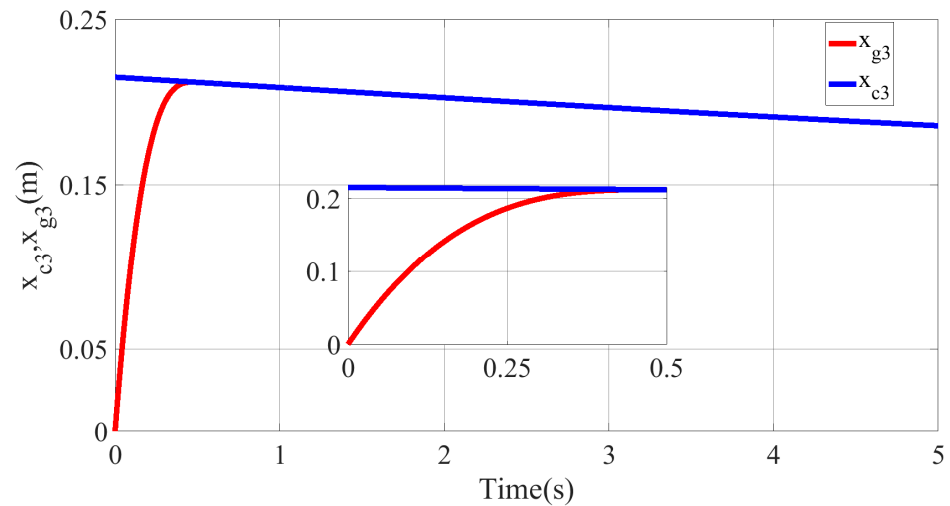
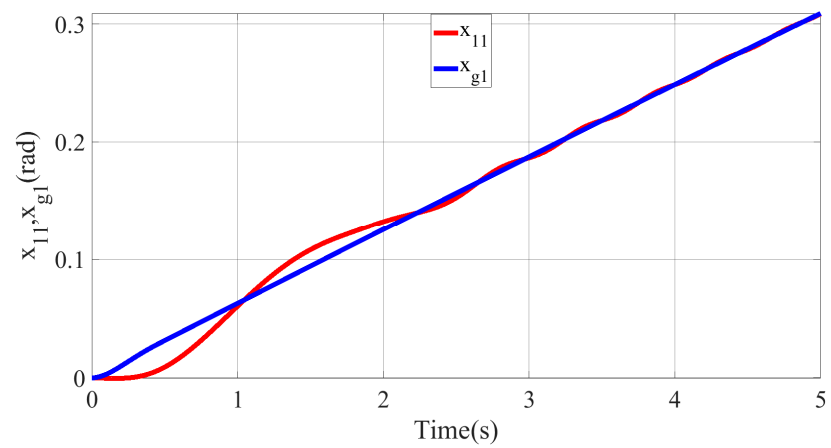


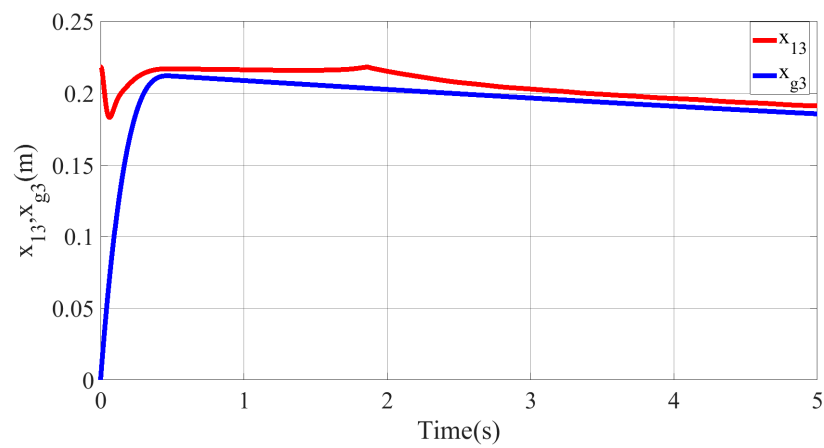
Figure 11. Motion curves before and after trajectory reconstruction.

The desired motion curve of PMA1 before and after trajectory reconstruction is, respectively, shown in the above figure. The desired motion curve before trajectory reconstruction  $x_{ck}(t)$  is the blue curve in the figure, while the desired motion curve after trajectory reconstruction  $x_{gk}(t)$  is the red curve in the figure. The initial value of the system after the trajectory reconstruction is equal to the initial value of the desired motion trajectory. Meanwhile, the desired trajectory after the trajectory reconstruction at the closed interval  $[\rho, T]$  is the same as the original desired trajectory.

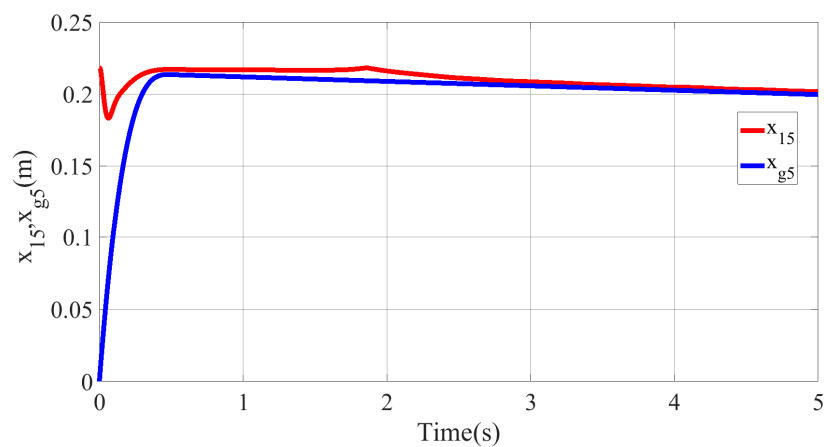
The tracking situation of the actual motion trajectories and the desired trajectories of each joint after 10 iterations is depicted in the above figure. It can be gathered that the motion trajectory can accurately track the reconstructed desired trajectory with arbitrary precision at  $k = 10$ . From Figure 12b,c, it can be found that the actual motion trajectory of PMA is forced to change the motion trajectory and quickly track the desired motion trajectory at approximately 1.8 s on account of the designed BLF. Consequently, the position-tracking error is limited by the given time-varying constraint function  $\psi(t)$ . Regarding time interval  $t \in [0, \rho)$ , there exists an obvious tracking error in the joint position because the selected lumped disturbance and time-varying constraint function are based on the reconstructed desired trajectory. Nevertheless, the reconstructed desired trajectory  $x_{gk}(t)$  can be accurately tracked by the actual motion trajectory  $x_{1k}(t)$  of each joint.



(a)



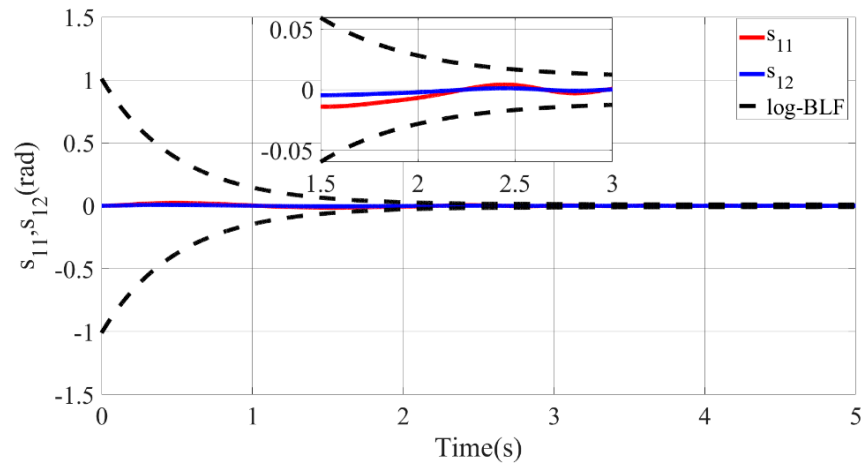
(b)



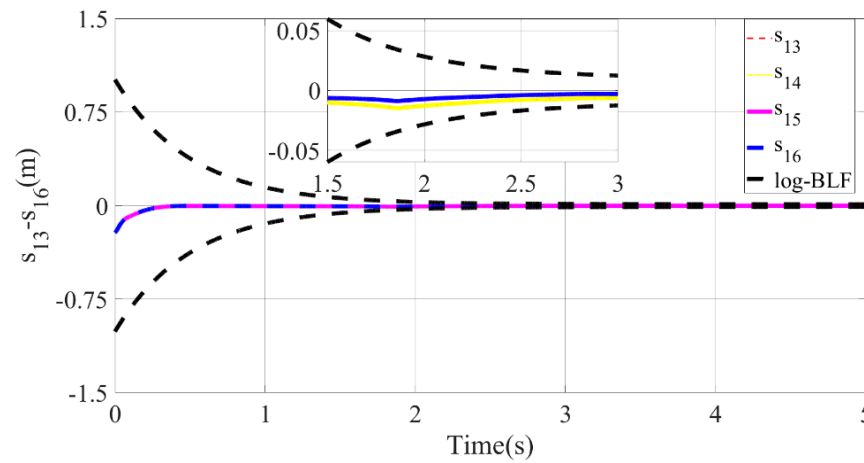
(c)

**Figure 12.** Iterative convergence curve of each joint-position tracking. (a) The joint-angle tracking curve of the SCARA manipulator. (b) Position-tracking curve of PMA1. (c) Position-tracking curve of PMA3.

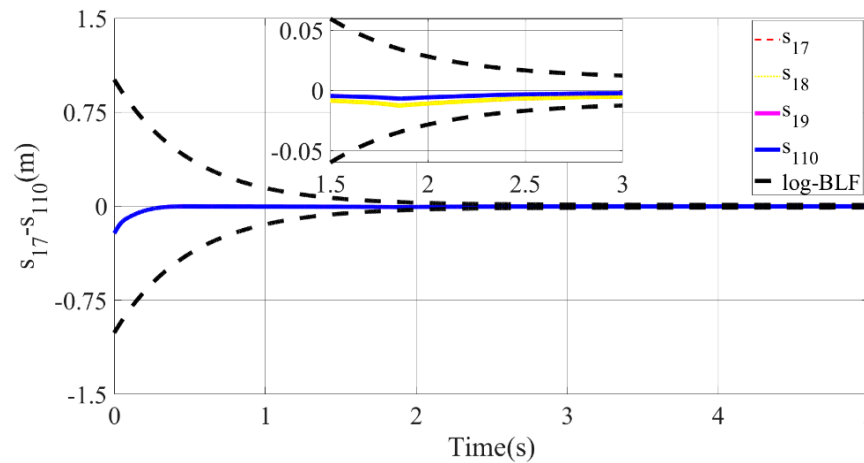




(a)



(b)



(c)

**Figure 13.** Iterative convergence curve of each joint-position tracking. (a) Angle-tracking error of SCARA manipulator in the joint space. (b) Position-tracking error of PMA1-4. (c) Position-tracking error of PMA5-8.

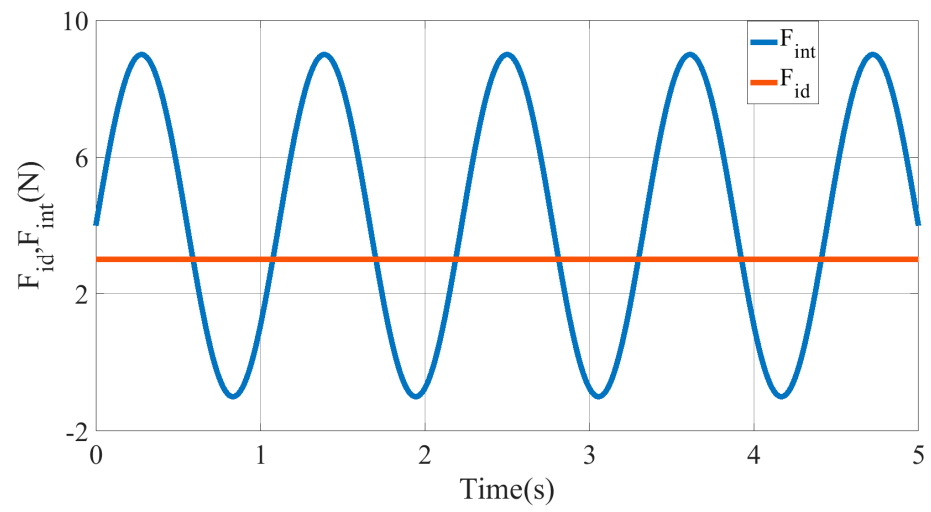


Figure 14. The curve of ISBF  $F_{id}$  and human-machine interaction force  $F_{int}$ .

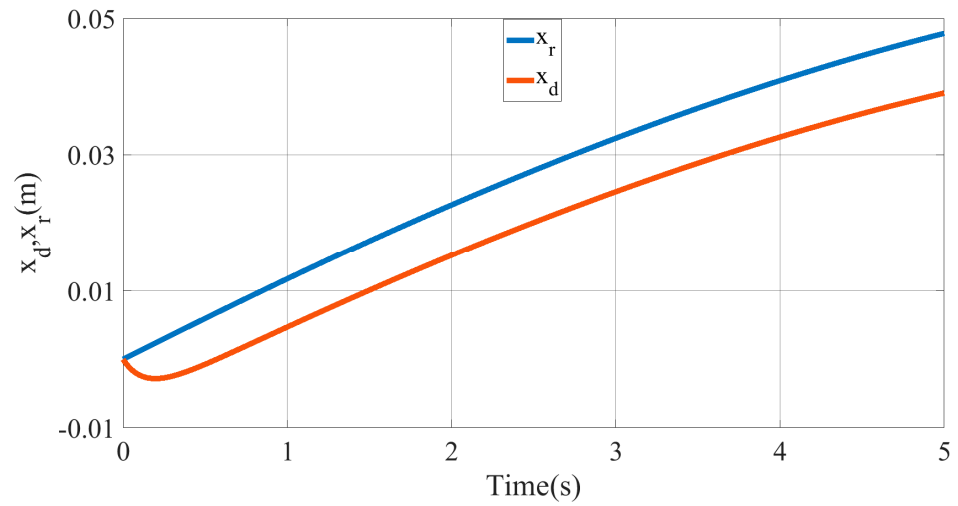
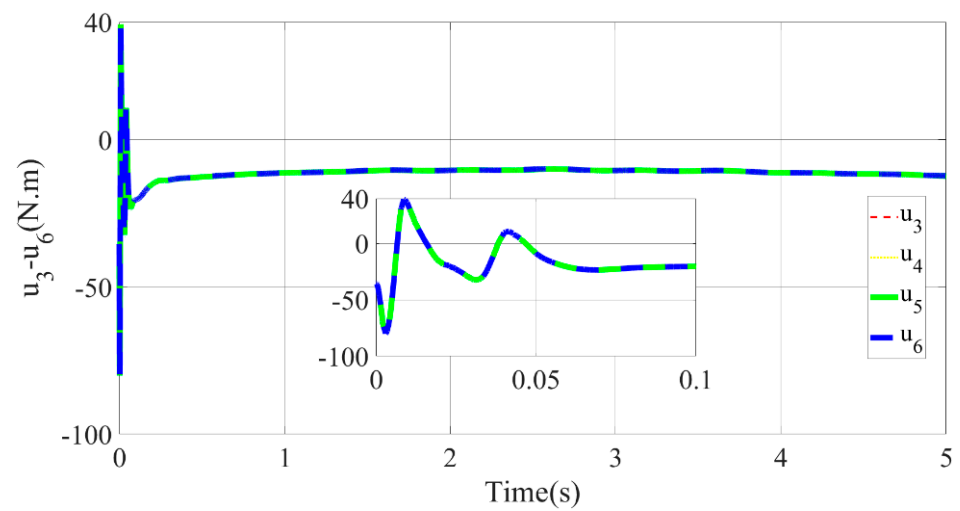


Figure 15. The commanded trajectory  $x_d$  and the desired trajectory  $x_r$  curve of the multisection continuum arms in the Cartesian space.



(a)

Figure 16. Cont.

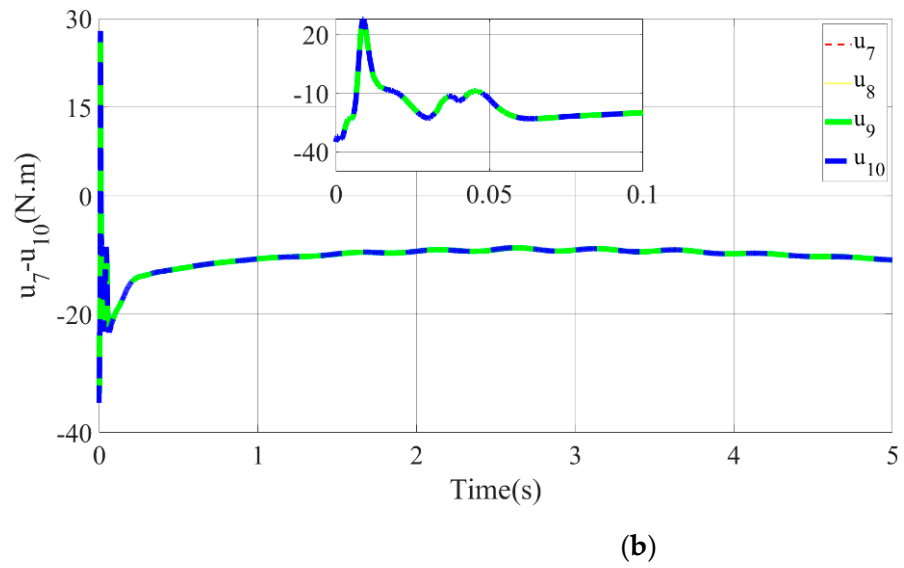


Figure 16. Input torque curve of each PMA. (a) Input torque of PMA1-4. (b) Input torque of PMA5-8.

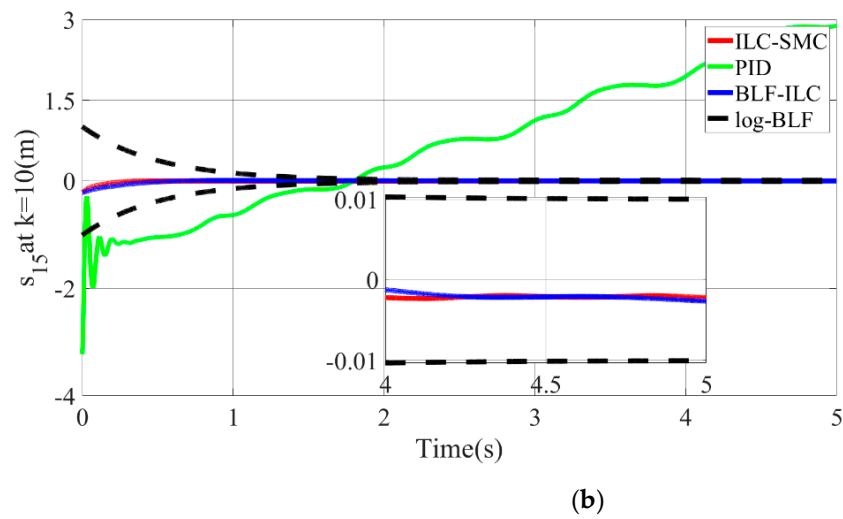
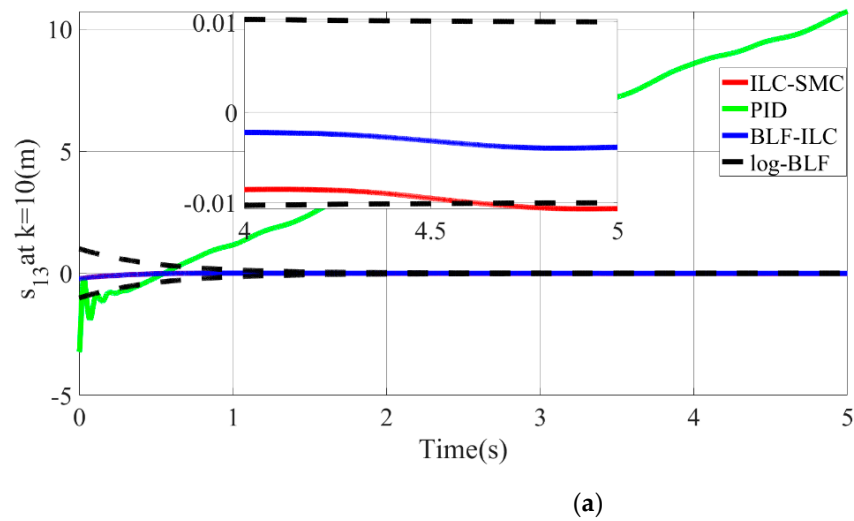
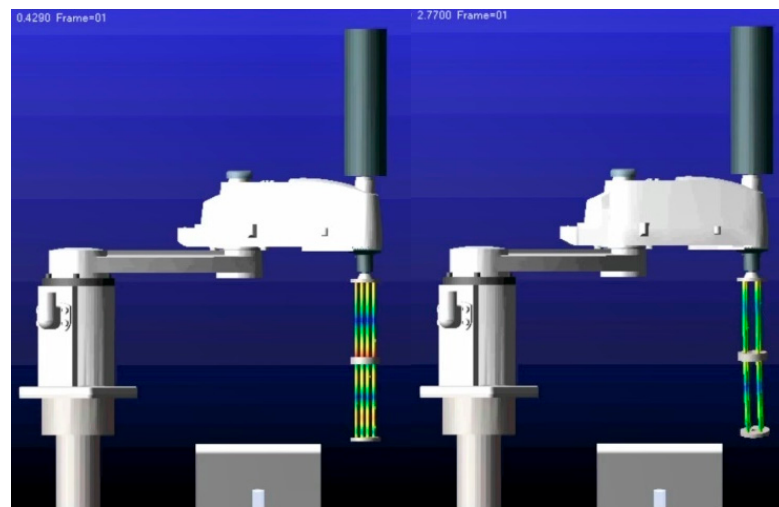
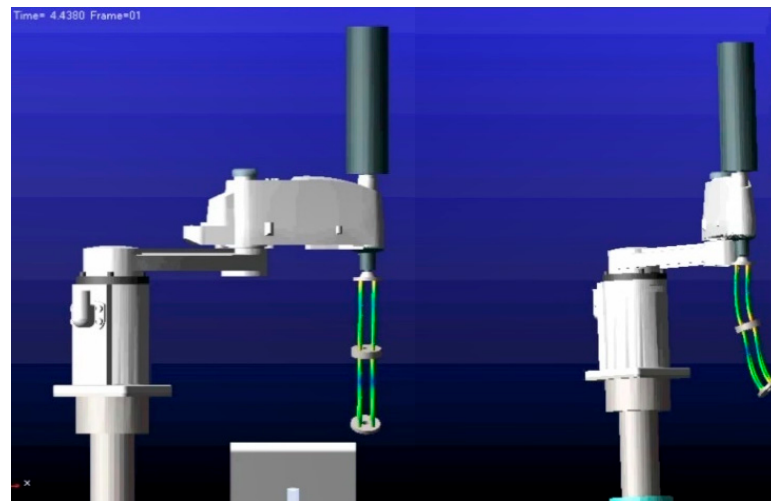


Figure 17. Comparison of simulation results with other control algorithms. (a) Position-tracking error curves of different control algorithms for PMA1. (b) Position-tracking error curves of different control algorithms for PMA3.



(a)



(b)

**Figure 18.** Cosimulation demonstration. (a) Cosimulation animation in  $T = 0.42s$  and  $T = 2.76s$ ; (b) cosimulation animation in  $T = 4.44s$  and  $T = 5s$ .

To demonstrate more clearly and intuitively that the tracking error of each joint position is uniformly bounded under the given time-varying constraint function, the error convergence curve of each joint-position tracking is depicted in Figure 13. When the tracking error approaches the constraint functions  $\psi(t)$  and  $-\psi(t)$  (black dotted lines in the figure) at a certain moment, it will be forcibly limited by the controller within the constraint function, and will never exceed the given time-varying constraint function during the convergence process. In conclusion, the position-tracking error of PMA can be guaranteed to be within 0.002 m and 0.004 m.

The desired trajectory and force balance curve for the end of the multisection continuum arms are shown in Figures 14–16. Owing to the introduction of impedance control to correct the command trajectory, the desired trajectory obtained is not a standard sinusoid. Hence, the impedance control can ensure that the ISBF is maintained at a given constant value. By adjusting the impedance coefficient, it was found that reducing the value of the impedance coefficient  $H, B, K$  resulted in a smoother input torque, but led to an increase in the adjustment amount of the desired trajectory  $x_e$ .

When the number of iterations  $k = 10$ , the position-tracking error curves of different control algorithms for PMA 1 and 3 are shown in Figure 17. Compared with IILSMC and PD impedance control, the proposed control scheme has the lowest tracking errors, which do not exceed 0.004 m. Furthermore, the actual motion trajectory has the competence to approximate the desired motion trajectory with arbitrary accuracy as the number of iterations increases. Contrasted by the other two algorithms, the position-tracking error obtained by the PD impedance control is divergent in the time-domain, which can be concluded that the traditional PD control cannot effectively handle the lumped disturbance. In other words, the robustness of the system based on PD control is poor. Quite the reverse, the control requirements are always satisfied by IILSMC and the proposed scheme.

Compared with IILSMC, the proposed algorithm possesses higher control accuracy. Specifically, the more superior the control accuracy of the proposed algorithm is, the more the motion speed of the pneumatic muscle increases (e.g., PMA1). For the proposed control scheme, when the tracking error converges to a given time-varying constraint function, the tracking error is forced to stay within the constraint function by controlling the input torque. The above phenomenon can be visualized in Figure 17a. Hence, in contrast with PD control and IILSMC, the proposed control scheme has a better tracking performance for a class of multisection continuum arms with lumped disturbance.

The cosimulation animation of the continuum arms is depicted in Figure 18. Through the cosimulation of ANSYS/ADAMS/MATLAB software, the deformation cloud and force situation of each PMA can be obtained. Apart from that, the motion attitude of the continuum arms can be simulated by the cosimulation animation in real-time. From the overall motion attitude, it can be drawn that the continuum arm model and its control algorithm designed in this paper are generally reasonable.

## 6. Conclusions

In this paper, a kinematic model of multisection continuum arms based on pneumatic muscle drive is established, where a four-parallel structure is adopted for the single pneumatic node of the continuum arms. To establish an accurate kinematic model, the mapping relationship between the deformation parameters of the continuum node and the length of each PMA is obtained based on the assumption of constant curvature utilizing the geometric method. The final kinematic model obtained has an error of up to 0.013%. For the existence of time-varying lumped disturbance of the multisection continuum arms, a dual closed-loop force-position hybrid control strategy is implemented in this paper. To begin with, the ISBF is maintained at 3 N by virtue of position-based impedance control. Meanwhile, to ensure that the tracking error of the multisection continuum arms is bounded, an adaptive iterative learning backstepping control strategy based on the log-type BLF is employed to achieve each PMA to track the desired trajectory with arbitrary accuracy where the tracking error converges stably to 0.004 m and does not exceed a given time-varying constraint function. Compared with IILSMC, the control accuracy of the proposed algorithm is enhanced by 57.14%. Furthermore, the lumped disturbance of the continuum arms can be effectively suppressed by the adaptive algorithm, where the input torque of each PMA is smooth without a chattering phenomenon. Specifically, the condition that the initial error of iterative learning is zero is released by the trajectory reconstruction approach. For PMA-driven multisection continuum robots, the modeling method proposed in this paper is generalized. Last but not least, for systems with complicated models, high degrees of freedom, and periodic motions, the control scheme proposed in this paper is general, where a mathematical model of the controlled object is not required.

Even so, in practical applications, there are some shortcomings in the proposed scheme which are supposed to be further investigated. For extremely complicated models, the computational efficiency tends to decrease as the amount of memory information surges. In the subsequent study, a specific experimental prototype will be executed to further verify the validity of the proposed theory.

**Supplementary Materials:** The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/mi13091532/s1>; S1: Kinematic Modeling; S2: Control Algorithms and Co-Simulation; S3: Co-simulation animation demonstration.

**Author Contributions:** All authors contributed to the study conception and design. Y.X., Formula Derivation, Algorithm Validation, Software, Writing—original draft; X.G., Conceptualization; J.L., Essay polish; X.H., Collecting and organizing data. H.S., Program debugging; G.Z., Methodology; Q.X., Data curation; T.M., Literature collection; M.L., Supervision; Q.D., Mechanical design. All authors have read and agreed to the published version of the manuscript.

**Funding:** The research work was supported by the National Key R&D Program of China (2020YFC2007700) and the Science and Technology Research Project for Higher Education Institutions in Hebei Province (QN2020252). Authors, Yuexuan Xu, Jian Li, etc., have received research support from Shun Kangda Medical Technology Co., Ltd. and the National Rehabilitation Aids Research Center.

**Data Availability Statement:** All data generated or analyzed during the study are included in the article, and the data that support the findings of this study are openly available.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Dong, X.; Wang, M.; Mohammad, A.; Ba, W.; Russo, M.; Norton, A.; Kell, J.; Axinte, A. Continuum Robots Collaborate for Safe Manipulation of High-Temperature Flame to Enable Repairs in Challenging Environments. *IEEE/ASME Trans. Mechatron.* **2022**, 1–4. [[CrossRef](#)]
- Liu, Q.; Liu, A.; Meng, W.; Ai, Q.; Xie, S.Q. Hierarchical Compliance Control of a Soft Ankle Rehabilitation Robot Actuated by Pneumatic Muscles. *Front. Neurobotics* **2017**, *11*, 64. [[CrossRef](#)] [[PubMed](#)]
- Simon, H.; Smitherman, H.; Atchley, A.; Davis, J.; Tenhundfeld, N. Nuts and Bolts About You: Finding the Right Match in Gendered Robots. In Proceedings of the 2020 Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, USA, 24 April 2020; pp. 1–5.
- Li, L.; Jin, T.; Tian, Y.; Yang, F.; Xi, F. Design and Analysis of a Square-Shaped Continuum Robot With Better Grasping Ability. *IEEE Access* **2019**, *7*, 57151–57162. [[CrossRef](#)]
- Rox, M.F.; Ropella, D.S.; Hendrick, R.J.; Blum, E.; Naftel, R.P.; Bow, H.C.; Herrel, D.S.; Weaver, K.D.; Chambless, L.B.; Webster, R.J., III. Mechatronic Design of a Two-Arm Concentric Tube Robot System for Rigid Neuroendoscopy. *IEEE/ASME Trans. Mechatron.* **2020**, *25*, 1432–1443. [[CrossRef](#)]
- Ryu, H.T.; Kang, L.; Yi, B.J. Application of Cosserat Rod Theory to Configuration Estimation of Coionoscope. In Proceedings of the 2018 15th International Conference on Ubiquitous Robots (UR), Honolulu, HI, USA, 26–30 June 2018; pp. 11–13.
- Nuchkrua, T.; Leephakpreeda, T.; Mekarporn, T. Development of robot hand with Pneumatic Artificial Muscle for rehabilitation application. In Proceedings of the 7th IEEE International Conference on Nano/Molecular Medicine and Engineering, Phuket, Thailand, 10–13 November 2013; pp. 55–58.
- Rucker, D.C.; Jones, B.A.; Webster, R.J., III. A Geometrically Exact Model for Externally Loaded Concentric-Tube Continuum Robots. *IEEE Trans. Robot.* **2010**, *26*, 769–780. [[CrossRef](#)]
- Doroudchi, A.; Berman, S. Configuration Tracking for Soft Continuum Robotic Arms Using Inverse Dynamic Control of a Cosserat Rod Model. In Proceedings of the 2021 IEEE 4th International Conference on Soft Robotics (RoboSoft), New Haven, CT, USA, 12–16 April 2021; pp. 207–214.
- Janabi-Sharifi, F.; Jalali, A.; Walker, I.D. Cosserat Rod-Based Dynamic Modeling of Tendon-Driven Continuum Robots: A Tutorial. *IEEE Access* **2021**, *9*, 68703–68719. [[CrossRef](#)]
- Ivanescu, M.; Florescu, M.C.; Popescu, N.; Popescu, D. Coil function control problem for a hyperredundant robot. In Proceedings of the 2007 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Zürich, Switzerland, 4–7 September 2007; pp. 1–6.
- Blumenschein, L.H.; Koehler, M.; Usevitch, N.S.; Hawkes, E.W.; Rucker, D.C.; Okamura, A.M. Geometric Solutions for General Actuator Routing on Inflated-Beam Soft Growing Robots. *IEEE Trans. Robot.* **2022**, *38*, 1820–1840. [[CrossRef](#)]
- Andersson, S.B. Discretization of a Continuous Curve. *IEEE Trans. Robot.* **2008**, *24*, 456–461. [[CrossRef](#)]
- Melingui, A.; Merzouki, R.; Mbede, J.B.; Escande, C.; Daachi, B.; Benoudjit, N. Qualitative approach for inverse kinematic modeling of a Compact Bionic Handling Assistant trunk. In Proceedings of the 2014 International Joint Conference on Neural Networks (IJCNN), Beijing, China, 6–11 July 2014; pp. 754–761.
- Cheng, H.; Liu, H.; Wang, X.; Liang, B. Approximate Piecewise Constant Curvature Equivalent Model and Their Application to Continuum Robot Configuration Estimation. In Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, Canada, 11–14 October 2020; pp. 1929–1936.
- Della Santina, C.; Bicchi, A.; Rus, D. On an Improved State Parametrization for Soft Robots With Piecewise Constant Curvature and Its Use in Model Based Control. *IEEE Robot. Autom. Lett.* **2020**, *5*, 1001–1008. [[CrossRef](#)]

17. Zeng, W.; Yan, J.; Yan, K.; Huang, X.; Wang, X.; Cheng, S.S. Modeling a Symmetrically-Notched Continuum Neurosurgical Robot With Non-Constant Curvature and Superelastic Property. *IEEE Robot. Autom. Lett.* **2021**, *6*, 6489–6496. [[CrossRef](#)]
18. Mehl, M.; Bartholdt, M.; Schappler, M. Dynamic Modeling of Soft-Material Actuators Combining Constant Curvature Kinematics and Floating-Base Approach. In Proceedings of the 2022 IEEE 5th International Conference on Soft Robotics (RoboSoft), Edinburgh, UK, 4–8 April 2022; pp. 1–8.
19. Gonthina, P.S.; Wooten, M.B.; Godage, I.S.; Walker, I.D. Mechanics for Tendon Actuated Multisection Continuum Arms. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 3896–3902.
20. Meng, F.; Lv, Y.; Ma, G.; Zhu, Y. A Model-Based Sliding-Mode Tracking Controller for Pneumatic Muscle Actuated Continuum Arms. In Proceedings of the 2018 Eighth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC), Harbin, China, 19–21 July 2018; pp. 636–640.
21. Godage, I.S.; Webster, R.J.; Walker, I.D. Center-of-Gravity-Based Approach for Modeling Dynamics of Multisection Continuum Arms. *IEEE Trans. Robot.* **2019**, *35*, 1097–1108. [[CrossRef](#)]
22. Huo, Y.; Li, P.; Chen, D.; Liu, Y.H.; Li, X. Model-Free Adaptive Impedance Control for Autonomous Robotic Sanding. *IEEE Trans. Autom. Sci. Eng.* **2021**, 1–11. [[CrossRef](#)]
23. Safaei, A. Cooperative Adaptive Model-Free Control With Model-Free Estimation and Online Gain Tuning. *IEEE Trans. Cybern.* **2021**, 1–13. [[CrossRef](#)]
24. Zhang, G.; Wang, J.; Yang, P.; Guo, S. Iterative learning sliding mode control for output-constrained upper-limb exoskeleton with non-repetitive tasks. *Appl. Math. Model.* **2021**, *97*, 366–380. [[CrossRef](#)]
25. Wang, J.; Li, R.; Zhang, G.; Wang, P.; Guo, S. Continuous sliding mode iterative learning control for output constrained MIMO nonlinear systems. *Inf. Sci.* **2021**, *556*, 288–304. [[CrossRef](#)]
26. Zhang, X.; Longman, R.W. Stability of Sliding Mode ILC Design for a Class of Nonlinear Systems With Unknown Control Input Delay. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–15. [[CrossRef](#)]
27. Shen, D.; Wu, J.; Wang, X.; Tian, M. Design and Analysis of a Novel Flat Pneumatic Artificial Muscle. In Proceedings of the 2021 IEEE 8th International Conference on Industrial Engineering and Applications (ICIEA), Chengdu, China, 23–26 April 2021; pp. 110–114.
28. Tee, K.P.; Ge, S.S.; Tay, E.H. Barrier Lyapunov Functions for the control of output-constrained nonlinear systems. *Automatica* **2009**, *45*, 918–927. [[CrossRef](#)]
29. Tee, K.P.; Ren, B.; Ge, S.S. Control of nonlinear systems with time-varying output constraints. *Automatica* **2011**, *47*, 2511–2516. [[CrossRef](#)]
30. Ni, J.; Shi, P. Global Predefined Time and Accuracy Adaptive Neural Network Control for Uncertain Strict-Feedback Systems with Output Constraint and Dead Zone. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 7903–7918. [[CrossRef](#)]
31. Shao, K.; Zheng, J.; Tang, R.; Li, X.; Man, Z.; Liang, B. Barrier Function Based Adaptive Sliding Mode Control for Uncertain Systems with Input Saturation. *IEEE/ASME Trans. Mechatron.* **2022**, 1–11. [[CrossRef](#)]
32. Li, N.; Yang, Z.; Huang, H.; Zhang, G. The Dynamic Simulation of Robotic Tool Changer Based on ADAMS and ANSYS. In Proceedings of the 2016 International Conference on Cybernetics, Robotics and Control (CRC), Hong Kong, China, 19–21 August 2016; pp. 13–17.
33. Liu, M.; Li, J.; Sun, H.; Guo, X.; Xuan, B.; Ma, L.; Xu, Y.; Ma, T.; Ding, Q.; An, B. Study on the Modeling and Compensation Method of Pose Error Analysis for the Fracture Reduction Robot. *Micromachines* **2022**, *13*, 1186. [[CrossRef](#)]