

Numerical Optimization-based Kinematics with Pose Tracking Control for Continuum Robots

Rui Peng, Ping Deng, Duo Tang, Peng Lu

Abstract—In this paper, we employ multiple IMUs into a triple-section continuum manipulator to precisely capture the attitude data of each section’s end disk. Leveraging the sensory and mechanical hardware system, we construct a sophisticated coordinate transformation scheme to accurately identify the detailed configuration states of the manipulator. Additionally, we introduce a numerical optimization strategy to develop a unified forward and inverse kinematic modeling framework, ensuring both iterative efficiency and accuracy. Through the IMUs’ real-time attitude feedback, we implement a closed-loop controller, enhancing the manipulator’s operational robustness and agility. In our experimental evaluations, we assess the convergence performance of both forward and inverse kinematics within a simulated environment and validate the precision of these kinematic models through real-time experiments on an actual continuum manipulator. Moreover, we evaluate the performance of the proposed controller by examining its accuracy during the manipulator’s continuous motions and analyzing its response characteristics. In contrast to previous research on continuum robots in the literature, we pioneer a fully integrated kinematic control architecture that is successfully implemented on a physical continuum robotic system.

Index Terms—Numerical optimization, kinematics, control, and continuum robots.

I. INTRODUCTION

Continuum robots [1], [2] have garnered substantial research attention for their inherent flexibility and safety, yet achieving high precision remains a significant challenge. Traditional forward kinematic approaches often struggle due to their reliance on actuation values that are difficult to measure accurately [3]. While advancements like permanent magnetic tracking [4], orientation sensors [5]–[8], and learning-based methods [9], [10] have enabled real-time shape reconstruction, their application is typically limited to robots with few degrees of freedom and a constrained workspace.

Inverse kinematics (IK), which is central to manipulation and planning, has been addressed by state-of-the-art approaches rooted in the FABRIK method and its variants [11]–[14], as well as convex optimization techniques [15]–[17]. However, a critical gap persists: the efficacy of these IK methods and existing motion controllers [18]–[23] remains unverified in real-world manipulation scenarios. Most hardware demonstrations [24], [25] have been limited to a small Cartesian workspace and did not involve physical interactions.

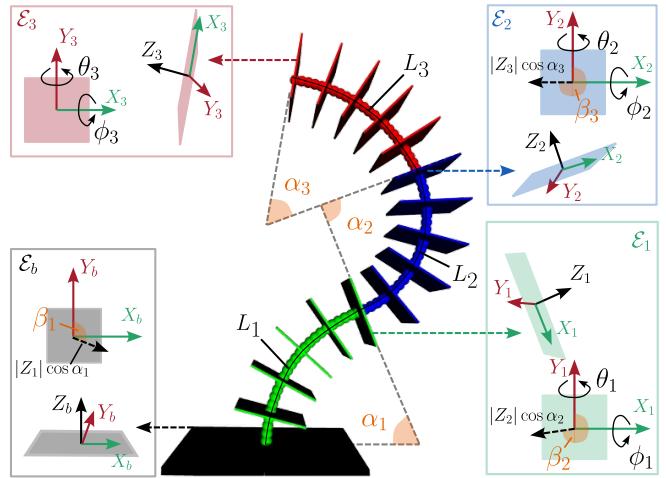


Fig. 1. Constant curvature continuum robot model (triple sections) with a detailed illustration of configuration angles and Euler angles. Key coordinates of all the end disks are depicted.

To address these limitations, we introduce a comprehensive kinematic control framework that integrates shape estimation, a desired configuration solver, and pose tracking [26]. The core contribution of this architecture is to enable actual interactive manipulation tasks that are beyond the capabilities of other continuum robots. Our framework uses multiple IMUs as the primary sensory input for high-dynamic attitude measurements. A numerical optimization strategy is employed for both forward kinematics (deriving the current shape) and inverse kinematics (determining the desired state for control). A closed-loop controller then regulates the configuration space to track the desired values. Validated on our custom simulator and hardware manipulator, the proposed architecture exhibits agile responsiveness to desired Cartesian poses and demonstrates simultaneous shape deformations under the designed controller.

II. PROPOSED METHODOLOGY

The derivation of the proposed continuum forward kinematics (FK) is based on the piecewise constant curvature assumption (PCC). The aim of the FK is to solve the real-time configuration space with respect to the given end disks’ attitude information.

A. Rotation Matrix Within One Section

For the i -th continuum section of our robotic system’s torso, the section’s configuration is described by $\zeta_i =$

The authors are with the Department of Mechanical Engineering, the University of Hong Kong, Hong Kong SAR, China.

$[\alpha_i, \beta_i]^\top$, where α_i denotes the bending angle, β_i denotes the directional angle, and the L_i denotes the arc length. Let the normal vector of the section's proximal end disk be $[x_i^p, y_i^p, z_i^p]^\top$, and the normal vector of the distal end disk be $[x_i^d, y_i^d, z_i^d]^\top$. To streamline the derivation of the rotation matrix in accordance with the section's configuration, we can assume that the initial position vector is given by $[x_i^p, y_i^p, z_i^p]^\top = [0, 0, 1]^\top$, and the desired position vector is defined as $[x_i^d, y_i^d, z_i^d]^\top = [s(\alpha_i) \cdot s(\beta_i), s(\alpha_i) \cdot c(\beta_i), c(\alpha_i)]^\top$.

Then, the rotation vector v_i^r between the proximal end disk and the distal end disk is denoted as $v_i^r = [x_i^r, y_i^r, z_i^r]^\top$. With the rotation vector v_i^r and the rotation radian α_i , we compute the rotation matrix R_{i-1}^i from the proximal end disk to the distal end disk within the i -the section by the Rodrigues formulation:

$$R_{i-1}^i = \begin{bmatrix} c\beta_i c\beta_i \gamma_i + c\alpha_i & -c\beta_i s\beta_i \gamma_i & s\alpha_i s\beta_i \\ -c\beta_i s\beta_i \gamma_i & s\beta_i s\beta_i \gamma_i + c\alpha_i & s\alpha_i c\beta_i \\ -s\alpha_i s\beta_i & -s\alpha_i c\beta_i & c\alpha_i \end{bmatrix} \quad (1)$$

where $\gamma_i = 1 - c\alpha_i$ and the section's angle configuration $[\alpha_i, \beta_i]^\top$. Therefore, we derive the rotation matrix within one continuum section based on the predefined frames in Fig. 1.

B. Attitude Of One End Disk

In our previous work [26], each segment's distal disk was equipped with a 6-axis IMU providing acceleration and angular velocity data. These measurements were processed through an EKF to estimate the disk orientation (quaternions), with details available in [1]. The quaternions were converted to Euler angles $\Phi_i = [\phi_i, \theta_i, \psi_i]^\top$ (Roll-Pitch-Yaw), though the yaw angle ψ_i remains indeterminate as we exclude magnetic field measurements due to environmental interference. Notably, our continuum robot design doesn't utilize active twisting motions.

C. Configuration Based On Gradient Optimization

As the Euler angles of three end disks represent the rotations relative to the base frame, the relation between sections' rotation matrices $\{R_b^1, R_b^2, R_b^3\}$ and the end disks' Euler angles $\{\Phi_1, \Phi_2, \Phi_3\}$ is given by:

$$\begin{bmatrix} R_b^1 \\ R_b^2 \\ R_b^3 \end{bmatrix} \iff \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \end{bmatrix} = \begin{bmatrix} R_b^1 \\ R_b^1 R_b^2 \\ R_b^1 R_b^2 R_b^3 \end{bmatrix} \iff \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \end{bmatrix} \quad (2)$$

where the operation \iff denotes the transformation between rotation matrices and corresponding Euler angles, as shown in Fig. 2. With known Euler angles $\{\Phi_1, \Phi_2, \Phi_3\}$ by IMUs, we obtain the individual rotation matrix:

$$\begin{aligned} R_b^i(\phi_i, \theta_i, \psi_i) &= R_b^i(\psi_i) R_b^i(\theta_i) R_b^i(\phi_i) = \\ &\begin{bmatrix} c\psi_i c\theta_i & c\psi_i s\theta_i s\phi_i - s\psi_i c\phi_i & c\psi_i s\theta_i c\phi_i + s\psi_i s\phi_i \\ s\psi_i c\theta_i & c\psi_i c\phi_i + s\psi_i s\theta_i s\phi_i & s\psi_i s\theta_i c\phi_i - c\psi_i s\phi_i \\ -s\theta_i & c\theta_i s\phi_i & c\theta_i c\phi_i \end{bmatrix} \end{aligned} \quad (3)$$

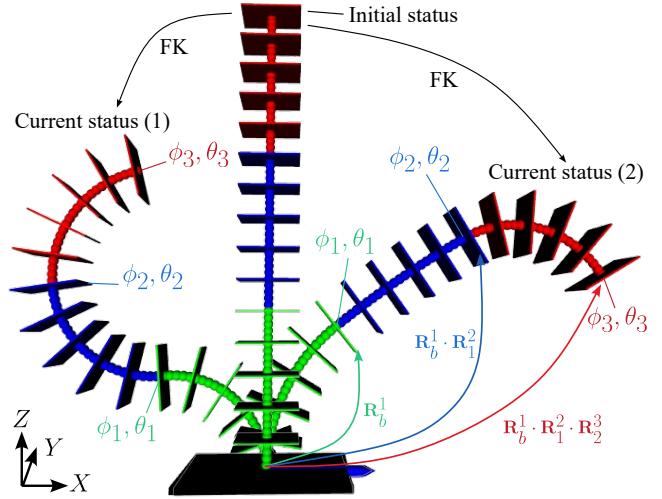


Fig. 2. Forward kinematics model in the simulation environment with given end disks' attitudes for the initial status to converge to current status.

where only ϕ_i, θ_i are reliable. Then, we intuitively extract the third row of both rotation matrices for the first section and build:

$$\begin{aligned} R_b^1(\phi_1, \theta_1, \psi_1) &= R_b^1(\alpha_1, \beta_1) \\ (R_b^1)^\top \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} &= \begin{bmatrix} -s\theta_1 \\ c\theta_1 \cdot s\phi_1 \\ c\theta_1 \cdot c\phi_1 \end{bmatrix} = \begin{bmatrix} -s\alpha_1 \cdot s\beta_1 \\ -s\alpha_1 \cdot c\beta_1 \\ c\alpha_1 \end{bmatrix} \end{aligned} \quad (4)$$

Similarly, for the second and the third sections, we have:

$$\begin{aligned} R_b^2(\phi_2, \theta_2, \psi_2) &= R_b^1(\alpha_1, \beta_1) \cdot R_b^2(\alpha_2, \beta_2) \\ R_b^3(\phi_3, \theta_3, \psi_3) &= R_b^1(\alpha_1, \beta_1) \cdot R_b^2(\alpha_2, \beta_2) \cdot R_b^3(\alpha_3, \beta_3) \end{aligned} \quad (5)$$

With available ϕ_2, θ_2, ϕ_3 , and θ_3 , we still extract the third row of both sides' rotation matrices and establish:

$$\begin{bmatrix} -s\theta_k \\ c\theta_k \cdot s\phi_k \\ c\theta_k \cdot c\phi_k \end{bmatrix} = (R_{k-1}^k)^\top \begin{bmatrix} (R_b^{k-1})(3, 1) \\ (R_b^{k-1})(3, 2) \\ (R_b^{k-1})(3, 3) \end{bmatrix}, k \in \{2, 3\} \quad (6)$$

where $(R)(i, j)$ denotes the value located at the i -th row and j -th column of the rotation matrix R .

While the connection between the entire continuum configurations and the Euler angles of all end disks has been clearly defined, the direct solution of these equations remains a formidable challenge. Consequently, we formulate three distinct optimization problems to determine the configuration angles through numerical iterations:

$$F_k = \underset{\alpha_k, \beta_k}{\operatorname{argmin}} \left\| \begin{bmatrix} -s\theta_k \\ c\theta_k \cdot s\phi_k \\ c\theta_k \cdot c\phi_k \end{bmatrix} - (R_{k-1}^k)^\top \begin{bmatrix} (R_b^{k-1})(3, 1) \\ (R_b^{k-1})(3, 2) \\ (R_b^{k-1})(3, 3) \end{bmatrix} \right\|_2 \quad (7)$$

where $k \in \{1, 2, 3\}$. When $k = 1$, $R_{k-1}^k = R_b^1$ and $R_b^{k-1} = I_{3 \times 3}$. The optimization process can be succinctly encapsulated in Algorithm 10. Here, J_i signifies the Jacobian matrix corresponding to R_{i-1}^i in relation to the parameters $\{\alpha_i, \beta_i\}$.

Algorithm 1: Numerical optimization for forward kinematics

```

1 [1]
2 Initialization:
3  $k \leftarrow 0, \epsilon \leftarrow 0.05$ 
4  $\zeta_{1,k} \leftarrow [\alpha_1^l, \beta_1^l], \zeta_{2,k} \leftarrow [\alpha_2^l, \beta_2^l], \zeta_{3,k} \leftarrow [\alpha_3^l, \beta_3^l]$ 
5  $k = 1$  to  $N$ 

$$\Delta\zeta_{1,k-1} = \epsilon \mathbf{J}_{1,k-1}^\top \begin{bmatrix} -s\theta_1 \\ c\theta_1 s\phi_1 \\ c\theta_1 c\phi_1 \end{bmatrix} - (\mathbf{R}_b^1)^\top \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

6  $\zeta_{1,k} = \zeta_{1,k-1} + \Delta\zeta_{1,k-1}$ 
7  $\Delta\zeta_{2,k-1} = \epsilon \mathbf{J}_{2,k-1}^\top \begin{bmatrix} -s\theta_2 \\ c\theta_2 s\phi_2 \\ c\theta_2 c\phi_2 \end{bmatrix} - (\mathbf{R}_b^2)^\top \begin{bmatrix} (\mathbf{R}_b^1)(3,1) \\ (\mathbf{R}_b^1)(3,2) \\ (\mathbf{R}_b^1)(3,3) \end{bmatrix}$ 
8  $\zeta_{2,k} = \zeta_{2,k-1} + \Delta\zeta_{2,k-1}$ 
9  $\Delta\zeta_{3,k-1} = \epsilon \mathbf{J}_{3,k-1}^\top \begin{bmatrix} -s\theta_3 \\ c\theta_3 s\phi_3 \\ c\theta_3 c\phi_3 \end{bmatrix} - (\mathbf{R}_b^3)^\top \begin{bmatrix} (\mathbf{R}_b^2)(3,1) \\ (\mathbf{R}_b^2)(3,2) \\ (\mathbf{R}_b^2)(3,3) \end{bmatrix}$ 
10  $\zeta_{3,k} = \zeta_{3,k-1} + \Delta\zeta_{3,k-1}$ 

```

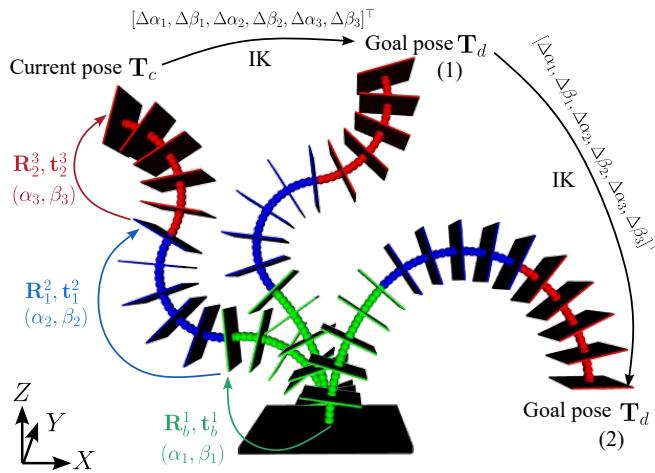


Fig. 3. Inverse kinematics model in the simulation environment with sections' transformation changing from the current pose to two goal poses.

The FK's cost function is established by the relation between two rotation matrices derived from the Euler angles (IMUs) and the continuum configuration space, as shown in Fig. 2. By numerical optimization, the deviation between two rotation matrices is expected to be minimized to zero. Then, the current configuration space is the best solution for constructing the continuum shape.

D. Inverse Kinematics

The inverse kinematics aims to solve complete configuration space to enable the end-effector's pose to reach the desired or given pose as accurately as possible. The 6-DOF pose is composed of the translation part and the rotation part. Within one section, the translation matrix from the

Algorithm 2: Numerical optimization for inverse kinematics

```

1 [1]
2 Initialization:
3  $k \leftarrow 0, \mu \leftarrow 0.05$ 
4  $\xi_k \leftarrow [\alpha_1, \beta_1, \alpha_2, \beta_2, \alpha_3, \beta_3]$ 
5  $k = 1$  to  $N$ 

$$\Delta\xi_{k-1} = \mu \mathbf{J}_{\xi, k-1}^\top \begin{bmatrix} \mathbf{R}_d & \mathbf{t}_d \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} \mathbf{R}_{ee} & \mathbf{t}_{ee} \\ 0 & 1 \end{bmatrix}$$

6  $\xi_k = \xi_{k-1} + \Delta\xi_{k-1}$ 

```

proximal end disk to the distal end disk can be obtained by the optimized configuration space $[\alpha_i, \beta_i]^\top$. According to Fig. 1, we have:

$$\mathbf{t}_{i-1}^i = \begin{bmatrix} L_i/\alpha_i \cdot (1 - c\alpha_i) \cdot s\beta_i \\ L_i/\alpha_i \cdot (1 - c\alpha_i) \cdot c\beta_i \\ L_i/\alpha_i \cdot s\alpha_i \end{bmatrix} \quad (8)$$

The end-effector's pose $\mathbf{T}_{ee} = \begin{bmatrix} \mathbf{R}_{ee} & \mathbf{t}_{ee} \\ 0 & 1 \end{bmatrix}$ can be given by:

$$\begin{aligned} \mathbf{R}_{ee} &= \mathbf{R}_b^1(\alpha_1, \beta_1) \cdot \mathbf{R}_b^2(\alpha_2, \beta_2) \cdot \mathbf{R}_b^3(\alpha_3, \beta_3) \\ \mathbf{t}_{ee} &= \mathbf{t}_b^1 + \mathbf{R}_b^1 \cdot \mathbf{t}_1^2 + \mathbf{R}_b^1 \cdot \mathbf{R}_b^2 \cdot \mathbf{t}_2^3 \end{aligned} \quad (9)$$

where $\mathbf{R}_{i-1}^i(\alpha_i, \beta_i)$ denotes the rotation matrix formed by $[\alpha_i, \beta_i]^\top$, as shown in Fig. 3. Given a desired end-effector's pose $\mathbf{T}_d = \begin{bmatrix} \mathbf{R}_d & \mathbf{t}_d \\ 0 & 1 \end{bmatrix}$, we build an optimization problem:

$$T = \underset{\xi}{\operatorname{argmin}} \left\| \begin{bmatrix} \mathbf{R}_d & \mathbf{t}_d \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} \mathbf{R}_{ee} & \mathbf{t}_{ee} \\ 0 & 1 \end{bmatrix} \right\|_2 \quad (10)$$

where the optimization process is presented in Algorithm 6. When T is minimized to zero, the current ξ represents the optimal solution in the desired configuration space.

E. Pose Control

To control our proposed continuum robotic system, we build a cascaded closed-loop controller that involves the feedback of forward kinematics and the input of inverse kinematics to implement the pose tracking control of the end-effector's pose in Cartesian space. Before the controller design, we present the statics and dynamics analysis of the continuum robot in Fig. 4. For each end disk in the plane, we build the force equilibrium equation:

$$\begin{aligned} I_i^m \cdot \dot{\Omega}_i &= (F_{i,t} - F_{i,e} - F_{i,f}) \cdot r \\ I_i^m &= m_i \cdot r^2 \\ F_{i,t} &= k_m \cdot v_m^2 \\ F_{i,e} &= \frac{E_{rod}}{L_i} \cdot \alpha_i \\ F_{i,f} &= f_\mu \cdot v_m \end{aligned} \quad (11)$$

where I_i^m is the i -th end disk's moment of inertia, $\dot{\Omega}_i$ is the angular acceleration, m_i is the end disk's mass, and r is the radius of the section. $F_{i,t}$, $F_{i,e}$, and $F_{i,f}$ denote the tendon actuation force, the elastic force, and the tendon friction

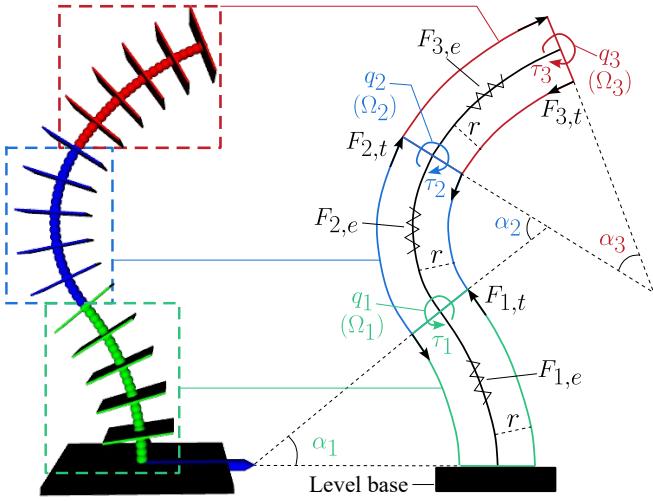


Fig. 4. Statics and dynamics diagram of the continuum arm. Key forces applied to the end disks result in configuration state changes.

force, respectively. Also, k_m denotes the motor acceleration gain, E_{rod} denotes the Youngâs modulus of the backbone, and f_μ denotes the frictional coefficient on the tendons. According to the equilibrium equation, we may manipulate the attitude of each end disk by inputting continuous motor velocity commands.

Given a target 6-DOF pose \mathbf{T}_d , we transform it into the desired configuration space $\boldsymbol{\xi}_d$ through the inverse kinematics. Simultaneously, leveraging real-time configuration space $\boldsymbol{\xi}_c$ obtained via numerical optimization-based forward kinematics, we construct the error matrix $\boldsymbol{\xi}_e$ for input into the configuration space controller:

$$\boldsymbol{\xi}_e = \boldsymbol{\xi}_d - \boldsymbol{\xi}_c \quad (12)$$

where the configuration space controller is designed to minimize the configuration error for controlling the end-effector's pose \mathbf{T}_c to converge to the desired pose:

$$\lim_{\boldsymbol{\xi}_e \rightarrow 0} (\mathbf{T}_d - \mathbf{T}_c) \rightarrow 0 \quad (13)$$

To reduce the $\boldsymbol{\xi}_e$, it is essential to transform the configuration space $\boldsymbol{\xi}$ into the length space \mathbf{L} and obtain the length error matrix \mathbf{L}_e :

$$\mathbf{L}_e = \mathbf{L}_d - \mathbf{L}_c \quad (14)$$

where \mathbf{L}_d and \mathbf{L}_c are the desired length space and the current length space. Then, a motor velocity controller is implemented to reduce the \mathbf{L}_e :

$$\begin{aligned} \dot{\mathbf{L}}_e &= \dot{\mathbf{L}}_d - \dot{\mathbf{L}}_c \\ \mathbf{v}_d &= \mathbf{K}_v \dot{\mathbf{L}}_e \end{aligned} \quad (15)$$

where \mathbf{v}_d denotes the desired velocity matrix for all the tendon motors with a proportional gain \mathbf{K}_v . However, the length matrix \mathbf{L}_c obtained from the motors' encoders can be modeled as:

$$\mathbf{L}_c = \hat{\mathbf{L}} + \boldsymbol{\iota}_l + \boldsymbol{\sigma}_l \quad (16)$$

where $\hat{\mathbf{L}}$ denotes the actual tendon length matrix, $\boldsymbol{\iota}_l$ denotes the loose length matrix, and $\boldsymbol{\sigma}_l$ denotes the elastic length matrix. Due to the difficulty of materials analysis, $\boldsymbol{\iota}_l$ and $\boldsymbol{\sigma}_l$ are not easy to obtain, which results in the inaccuracy of the \mathbf{L}_c . Therefore, relying on tendon lengths makes the configuration error matrix satisfy:

$$\underset{\mathbf{v}_d, \mathbf{L}_e}{\operatorname{argmin}} \|\boldsymbol{\xi}_e\|_2 > 0 \quad (17)$$

To avoid the inaccuracy of the length model, we establish an attitude space $[\mathbf{q}_1, \Omega_1, \mathbf{q}_2, \Omega_2, \mathbf{q}_3, \Omega_3]^\top$ for all the end disks of the tendon-driven continuum manipulator, where \mathbf{q}^* denotes the attitude and Ω^* denotes the angular velocity of the end disks. The attitude error matrix between the desired end disks' attitudes derived from the configuration space and the IMUs' attitudes is given as:

$$\boldsymbol{\mathbf{q}}_e^* = \mathbf{q}_d^* - \mathbf{q}_I^* \quad (18)$$

Therefore, we build an attitude-angular controller to minimize the attitude errors of three end disks:

$$\begin{aligned} \mathbf{v}_d &= \mathbf{K}_\omega \Omega_e^* + \mathbf{I}_\omega \int \Omega_e^* \\ \Omega_e^* &= \mathbf{K}_q \mathbf{q}_e^* - \Omega^* \end{aligned} \quad (19)$$

where \mathbf{K}_ω and \mathbf{I}_ω denote the proportional gain and the integral gain of the angular controller. \mathbf{K}_q denotes the proportional gain of the attitude controller and Ω_e^* denotes the angular error matrix. Consequently, the control input matrix \mathbf{u} is given by:

$$\begin{aligned} \mathbf{u} &= \mathbf{K}_v \mathbf{v}_e + \mathbf{I}_v \int \mathbf{v}_e + \mathbf{D}_v \dot{\mathbf{v}}_e \\ \mathbf{v}_e &= \mathbf{v}_d - \mathbf{v} \end{aligned} \quad (20)$$

where \mathbf{K}_ω , \mathbf{I}_ω , and \mathbf{D}_v denote the proportional gain, the integral gain, and derivative gain of the motor velocity controller, respectively. \mathbf{v}_e represents the motor velocity error matrix.

III. EXPERIMENTS

We present an experimental robotic system to evaluate the proposed kinematic control architecture. The setup features a tendon-driven continuum manipulator (Fig. 5(a)) with a compact, high-power actuation system, enabling versatile motion. The manipulator comprises three omnidirectional sections, maintaining constant curvature during bending. Key sensors, including embedded IMUs, are distributed along the torso and end disks of each section. A simulation environment provides real-time forward kinematics visualization. The system integrates an embedded controller (STM32F427IIH6) for sensor acquisition and motor control at 1000 Hz, alongside an onboard PC (Ubuntu 18.04, ROS Melodic) hosting a multi-threaded software platform (Fig. 5(b)).

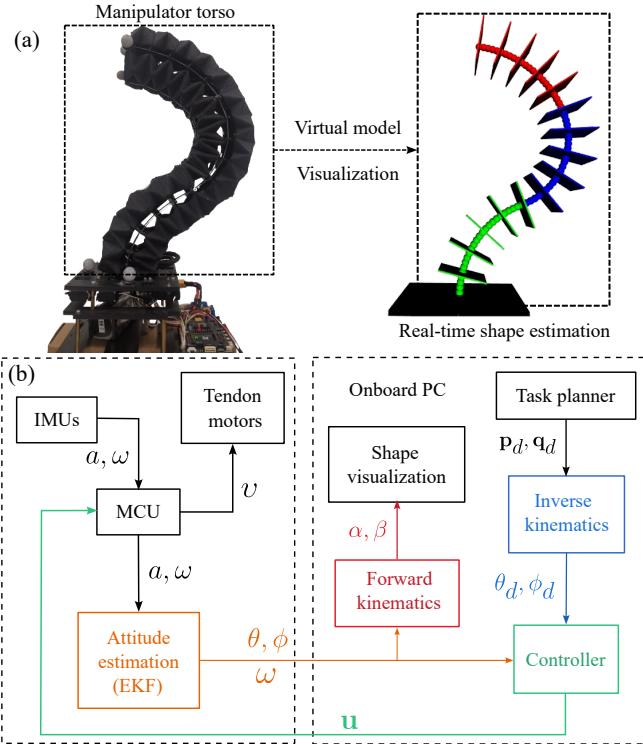


Fig. 5. Experimental setup. (a) The continuum robotic system with its real-time shape visualization environment. (b) The architecture of the systematic software.

A. Forward Kinematics In Simulation

The proposed forward kinematics (FK) aims to achieve the desired configuration space by minimizing the gap between initial and target end disk attitudes through numerical optimization. We initialize the configuration space and set desired Euler angles, then visualize the robot's model. The attitude errors are defined as $e_i = [||\phi_{i,d} - \phi_i||, ||\theta_{i,d} - \theta_i||]$, where “d” denotes desired angles.

Starting with initial configurations and desired attitudes, the FK reduces all Euler errors to less than 0.01 rad within 0.04 s, stabilizing the configuration angles. To test dynamic performance, we generate consecutive Euler angles in a vertical plane, achieving errors below 0.002 rad.

Next, we simulate more complex motions with Euler angles as $[\phi_1, \theta_1] = [0.698 \cos t, 0.698 \sin t]$, $[\phi_2, \theta_2] = [1.047 \cos t, 1.047 \sin t]$, $[\phi_3, \theta_3] = [1.396 \cos t, 1.396 \sin t]$. The FK maintains errors around 0.002 rad, except for a brief peak of 0.03 rad in the third end disk, confirming its feasibility and dynamic capability across varying scenarios.

B. Inverse Kinematics In Simulation

To assess the convergence of the proposed inverse kinematics for the continuum robot in solving six-DOF poses, we design two desired setups: 2D plane motion and 3D space motion. 2D motion restricts section-bending within one plane, while 3D motion involves large-range config changes. We split the pose into position p and orientation

q to showcase the IK iteration process. Initial errors p_{init} and q_{init} are the max errors before IK. We define position error rate $e_p^r = ||p - p_{init}||$ and orientation error rate $e_q^r = ||q - q_{init}||$.

In Fig. 7(a), the robot's current tip pose aims to reach the target within one plane. We test four IK step sizes: [0.0015, 0.0115, 0.0515, 0.1015] to find the optimal step size for quick error reduction. Error rates (step size: 0.0515 and 0.1015) drop to zero around the 100-th iteration in Fig. 7(a). For 3D motion, the iteration process with the same step sizes is in Fig. 7(b). Position and orientation error rates (step size: 0.1015) converge to 0.25 and 0.4 after 200 iterations due to large step size preventing local minima approach. Error rates (step size: 0.0515) drop to near zero around the 200-th iteration. Thus, we select 0.0515 as the optimal step size for subsequent experiments.

To evaluate IK efficiency, we design four desired poses for the continuum robot model. In Fig. 7(c), we set the vertical pose as the current pose with the optimal step size: 0.0515. Desired poses are in four different directions. All error rates drop to ≤ 0.15 at 0.25 s. Thus, the proposed IK efficiently handles different target poses within the workspace.

C. Trajectory Tracking And Control

Trajectory tracking is essential for continuum robots in various applications. We designed six trajectories, three in the horizontal and three in the vertical plane, to evaluate the tracking performance of our proposed kinematic control architecture on the continuum manipulator. Experimental results for the horizontal plane are shown in Fig. 8(a, b, c). Starting with fundamental triangle and rectangle trajectories, we then conducted a challenging “HKU” trajectory tracking motion. FK's tip position tracked the desired position well with less than 0.02 m errors.

To further evaluate tracking capability, we set three vertical plane trajectories: irregular-block, double-triangle, and circle. Tracking results, including desired and FK's paths, are shown in Fig. 8(d, e, f). Overall, the tip's position tracked the desired position, except for a period of deviation in Fig. 8(f). We have fully evaluated the proposed kinematic control architecture with our continuum robot. The robot can agilely respond to control commands and accurately provide real-time shape estimation for visualization. These results are fundamental for future robotic applications.

After verifying forward and inverse kinematics via simulation and experiments, we establish a closed-loop control architecture for versatile continuum robot motions. To assess the controller's performance, we design a 3D trajectory: $[0.27 \sin t, 0.27 \cos t, 3.3 + 0.02 \sin t]$ for the tip to track, as shown in Fig. 9. Focusing on response and accuracy, we record tip positions and compute errors. At 0.52 rad/s, errors are 0.025 m, mainly in Z-direction, with fluctuations between 0.015 m and 0.03 m. The real robot's shape aligns well with the simulation model. Increasing the velocity to 1.30 rad/s and 2.61 rad/s, errors rise to 0.276 m and 0.34 m, respectively. Thus, the velocity should be limited to 0.52 rad/s.

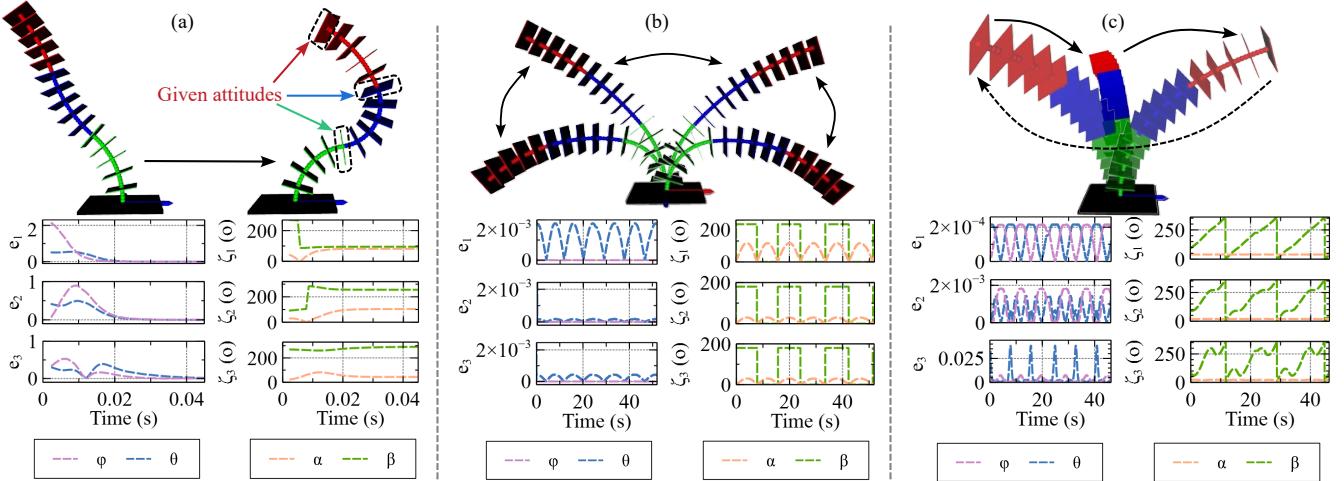


Fig. 6. Forward kinematics with three types of attitude setups is performed in simulation. The attitude errors of their cost function and variation of configuration space are depicted in corresponding plots. (a) A given arbitrary attitude setup. (b) Consecutive swing attitude setup within one vertical plane. (c) Consecutive circle attitude setup within one level plane. “o” denotes the unit: degree.

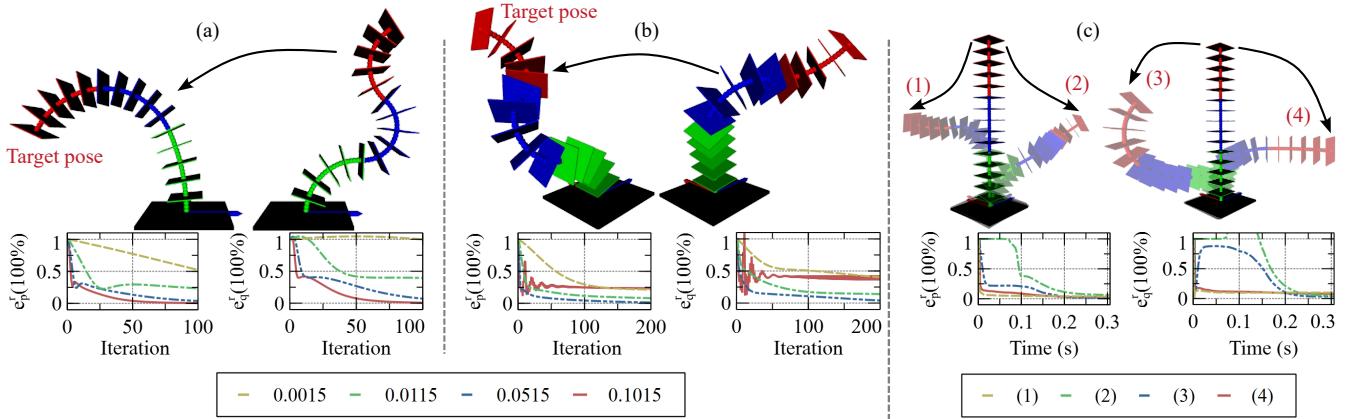


Fig. 7. Inverse kinematics with three types of desired pose setups is performed in simulation. The position and orientation errors of their cost function are depicted in corresponding plots. (a) 2D shape transition by IK. (b) 3D shape transition by IK. (c) Four shape transitions by IK from the initial vertical pose.

D. Interaction With The Continuum Robot

To implement the proposed inverse kinematics (IK) on the continuum robot, we developed a 6DOF interactor capable of providing the desired pose within the ROS-Rviz environment, complemented by a corresponding simulation model. With the confirmed convergence of the IK, the tip of the simulation model exhibits rapid and precise tracking of the interactor. Subsequently, the IK furnishes the desired configuration space to the closed-loop controller, which then generates the necessary control signals for the tendon motors. The aim is to ensure that the tip of the continuum robot agilely tracks the desired pose, while the actual shape of the robot’s body synchronizes with the simulation model through forward kinematics (FK). In Fig. 10, we present three distinct configurations of the robot and model, each with the 6DOF interactor being manipulated arbitrarily by

a human operator. Simultaneously, we capture and plot the desired position and orientation separately. The results from the FK are also displayed to validate whether the actual tip pose successfully tracks the desired pose. Consequently, the proposed IK demonstrates robust performance on the actual continuum robot, driven by the controller, with notably low tracking errors.

IV. CONCLUSION

This paper presents a kinematic control framework for tendon-driven continuum robots, enabling versatile motion through closed-loop control. By combining the piecewise constant curvature assumption with multi-IMU attitude measurements, we establish configuration-space orientation relationships via rotation matrices and resolve detailed angles through numerical optimization. The inverse kinematics de-

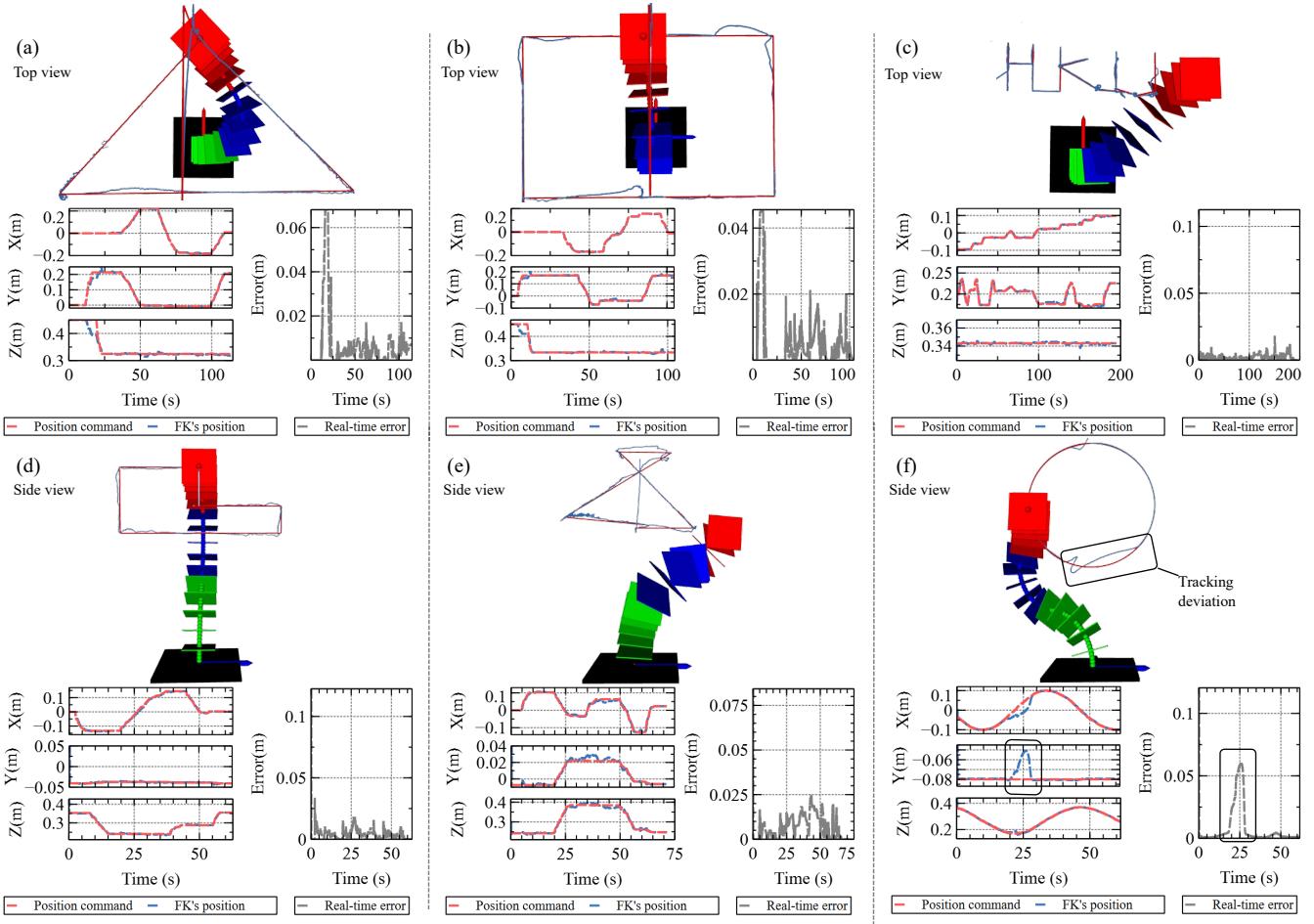


Fig. 8. Trajectory tracking motions with different trajectories. (a, b, c) Triangle trajectory, rectangle trajectory, and “HKU” trajectory tracking from the top view. (d, e, f) Irregular-block trajectory, double-triangle trajectory, and circle trajectory tracking from the side view. Each tracking motion has position plots.

termines optimal configurations by minimizing pose errors, while the integrated forward-inverse kinematic solution enables precise motion control. Experimental validation progresses from simulation-based convergence tests to physical implementation, demonstrating effective position/orientation tracking and trajectory execution that confirms the system’s agility and control precision.

REFERENCES

- [1] M. Russo, S. M. H. Sadati, X. Dong, A. Mohammad, I. D. Walker, C. Bergeles, K. Xu, and D. A. Axinte, “Continuum robots: An overview,” *Advanced Intelligent Systems*, vol. 5, no. 5, p. 2200367, 2023.
- [2] J. Zhang, Q. Fang, P. Xiang, D. Sun, Y. Xue, R. Jin, K. Qiu, R. Xiong, Y. Wang, and H. Lu, “A survey on design, actuation, modeling, and control of continuum robot,” *Cyborg and Bionic Systems*, vol. 2022, 2022.
- [3] J. Santoso and C. D. Onal, “An origami continuum robot capable of precise motion through torsionally stiff body and smooth inverse kinematics,” *Soft Robotics*, vol. 8, no. 4, pp. 371–386, 2021.
- [4] S. Song, H. Ge, J. Wang, and M. Q.-H. Meng, “Real-time multi-object magnetic tracking for multi-arm continuum robots,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–9, 2021.
- [5] H. Cheng, H. Xu, H. Shang, X. Wang, H. Liu, and B. Liang, “Orientation to pose: Continuum robots shape reconstruction based on the multi-attitude solving approach,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 3203–3209.
- [6] H. Bezwada, C. Woods, and V. Vikas, “Shape reconstruction of soft manipulators using vision and imu feedback,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9589–9596, 2022.
- [7] B. Deutschmann, M. Chalon, J. Reinecke, M. Maier, and C. Ott, “Six-dof pose estimation for a tendon-driven continuum mechanism without a deformation model,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3425–3432, 2019.
- [8] R. Peng, Z. Wang, and P. Lu, “Aecom: An aerial continuum manipulator with imu-based kinematic modeling and tendon-slacking prevention,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 8, pp. 4740–4752, 2023.
- [9] K. Tanaka, Y. Minami, Y. Tokudome, K. Inoue, Y. Kuniyoshi, and K. Nakajima, “Continuum-body-pose estimation from partial sensor information using recurrent neural networks,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11244–11251, 2022.
- [10] Y. Lu, W. Chen, Z. Chen, J. Zhou, and Y.-h. Liu, “Fbg-based variable-length estimation for shape sensing of extensible soft robotic manipulators,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 01–08.
- [11] F. Wang, W. Ye, X. Kang, H. Wang, J. Luo, L. Chen, and X. Tang, “Fabrikv: A fast, iterative inverse kinematics solver for surgical continuum robot with variable curvature model,” *10* 2023, pp. 8417–8424.

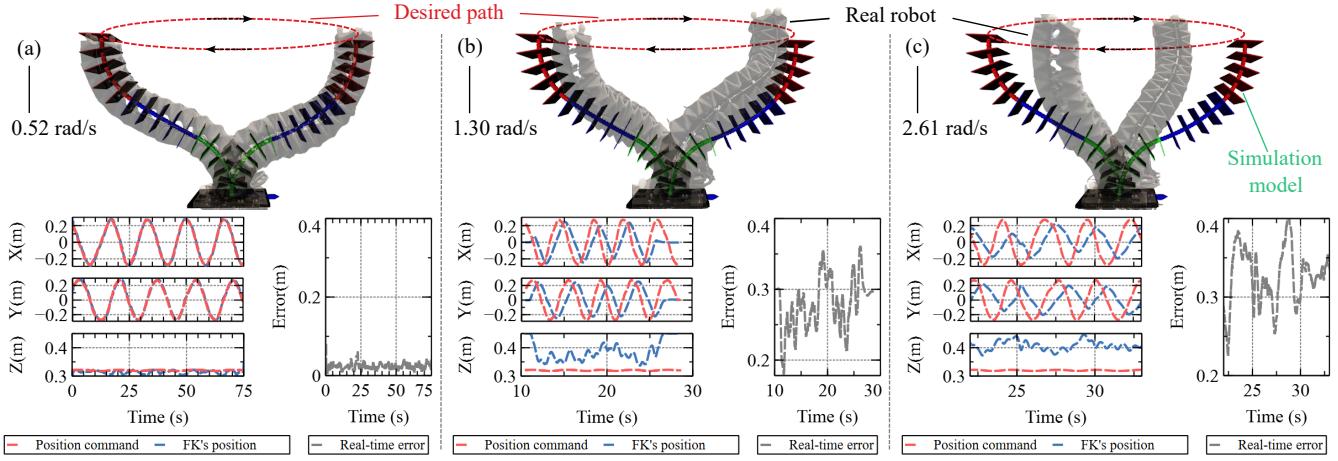


Fig. 9. Tip position's controller evaluation with different desired tip velocities in (a) 0.52 rad/s, (b) 1.30 rad/s, and (c) 2.61 rad/s. Three velocity setups share the same desired path. The synchronized real robot and simulation model are presented during the tracking motion.

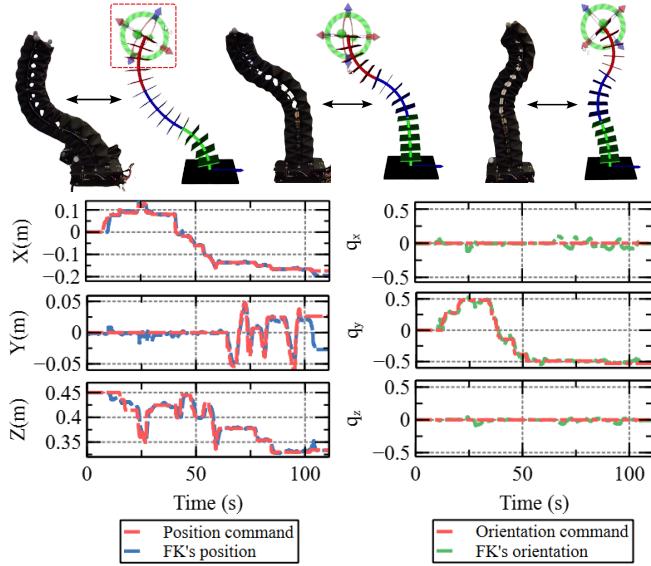


Fig. 10. IK performs on the actual continuum robot with four manipulation sets. The desired pose is directly given by the 6DOF interactor which is moved by a human operator in ROS-Rviz.

- [12] D. Kolpashchikov, O. Gerget, and V. Danilov, “Fabrikx: Tackling the inverse kinematics problem of continuum robots with variable curvature,” *Robotics*, vol. 11, p. 128, 11 2022.
- [13] W. Zhang, Z. Yang, T. Dong, and K. Xu, “Fabrikc: an efficient iterative inverse kinematics solver for continuum robots,” *2018*
- [15] H. J. Zhang, M. Giamou, F. Marić, J. Kelly, and J. Burgner-Kahrs, “Cidgikc: Distance-geometric inverse kinematics for continuum robots,” *IEEE Robotics and Automation Letters*, 2023.
- [16] K. Qiu, J. Zhang, D. Sun, R. Xiong, H. Lu, and Y. Wang, “An efficient multi-solution solver for the inverse kinematics of 3-section constant-curvature robots,” *arXiv preprint arXiv:2305.01458*, 2023.

IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), pp. 346–352, 2018. [Online]. Available: <https://api.semanticscholar.org/CorpusID:52165938>

- [14] H. Wu, J. Yu, J. Pan, L. Guoxin, and X. Pei, “Crik: A fast heuristic algorithm for the inverse kinematics of continuum robot,” *Journal of Intelligent Robotic Systems*, vol. 105, 07 2022.
- [17] Y. Li, D. Myszka, and A. Murray, “The kinematics of constant curvature continuum robots through three segments,” *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–8, 11 2023.
- [18] Q. Zhao, J. Lai, K. Huang, X. Hu, and H. K. Chu, “Shape estimation and control of a soft continuum robot under external payloads,” *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 5, pp. 2511–2522, 2021.
- [19] X. Huang, J. Zou, and G. Gu, “Kinematic modeling and control of variable curvature soft continuum robots,” *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 6, pp. 3175–3185, 2021.
- [20] D. Lin, N. Jiao, Z. Wang, and L. Liu, “A magnetic continuum robot with multi-mode control using opposite-magnetized magnets,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2485–2492, 2021.
- [21] 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.
- [22] S. Li, A. Kruszewski, T.-M. Guerra, and A.-T. Nguyen, “Equivalent-input-disturbance-based dynamic tracking control for soft robots via reduced-order finite-element models,” *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 5, pp. 4078–4089, 2022.
- [23] R. E. Goldman, A. Bajo, and N. Simaan, “Compliant motion control for multisegment continuum robots with actuation force sensing,” *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 890–902, 2014.
- [24] S.-S. Chiang, H. Yang, E. Skorina, and C. D. Onal, “Slinki: State lattice based inverse kinematics-a fast, accurate, and flexible ik solver for soft continuum robot manipulators,” in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2021, pp. 1871–1877.
- [25] G. Fang, Y. Tian, Z.-X. Yang, J. M. Geraedts, and C. C. Wang, “Efficient jacobian-based inverse kinematics with sim-to-real transfer of soft robots by learning,” *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 6, pp. 5296–5306, 2022.
- [26] R. Peng, Y. Wang, and P. Lu, “A tendon-driven continuum manipulator with robust shape estimation by multiple imus,” *IEEE Robotics and Automation Letters*, 2024.