

Research Article

Flexible head-following motion planning for scalable and bendable continuum robots

Te Li, Guoqing Zhang, Xinyuan Li, Xu Li^{*}, Haibo Liu, Yongqing Wang

State Key Laboratory of High-performance Precision Manufacturing, School of Mechanical Engineering, Dalian University of Technology, Dalian 116023, China

ARTICLE INFO

Article history:

Received 24 February 2024

Revised 21 April 2024

Accepted 30 April 2024

Available online 4 May 2024

Keywords:

Continuum robot

Motion planning

Scalable and bendable

Head-following

ABSTRACT

Continuum robots, which are characterized by high length-to-diameter ratios and flexible structures, show great potential for various applications in confined and irregular environments. Due to the combination of motion modes, the existence of multiple solutions, and the presence of complex obstacle constraints, motion planning for these robots is highly challenging. To tackle the challenges of online and flexible operation for continuum robots, we propose a flexible head-following motion planning method that is suitable for scalable and bendable continuum robots. Firstly, we establish a piecewise constant curvature (PCC) kinematic model for scalable and bendable continuum robots. The article proposes an adaptive auxiliary points model and a method for updating key nodes in head-following motion to enhance the precise tracking capability for paths with different curvatures. Additionally, the article integrates the strategy for adjusting the posture of local joints of the robot into the head-following motion planning method, which is beneficial for achieving safe obstacle avoidance in local areas. The article concludes by presenting the results of multiple sets of motion simulation experiments and prototype experiments. The study demonstrates that the algorithm presented in this paper effectively navigates and adjusts posture to avoid obstacles, meeting the real-time demands of online operations. The average time for a single-step solution is 4.41×10^{-5} s, and the average tracking accuracy for circular paths is 7.8928 mm.

© 2024 The Author(s). Published by Elsevier B.V. on behalf of Shandong University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Continuum robots are inspired by mollusks or tissues such as elephant trunks and octopus tentacles [1]. Due to their flexible material, slender body, and ability to deform flexibly, continuum robots have great potential for use in narrow and complex tasks, such as interventional surgery [2], disaster rescue [3], and human-machine collaboration [4]. The motion planning problem for continuum robots is more challenging than for traditional rigid body robots due to their flexible bodies, multiple solutions, and complex obstacle constraints.

Motion planning methods for continuum robots can be classified into four categories: Jacobi pseudo-inverse [5], intelligent optimization algorithms [6], learning class algorithms [7], and heuristic algorithms based on empirical constructions [8–10].

The Jacobi pseudo-inverse method is an iterative solution method for a system of linear equations. Xu et al. introduced a length variable to the traditional Jacobi pseudo-inverse, making the scalable and bendable continuum robot solvable with an inverse solution [11]. Due to the complexity of the continuum

robot model, its iterative solution efficiency is generally low, making it more suitable for offline motion planning.

The intelligent optimization algorithm is a heuristic search algorithm that converts the problem of inverse motion planning solutions into an optimization problem, with the minimum path deviation as the objective. Gao et al. proposed an end-following strategy based on a differential evolutionary algorithm for a four-segmented continuum robot [12]. Junius et al. proposed a gradient descent-based inverse kinematics solution for a three-segmented origami robot to achieve path-tracking [13]. The heuristic search mechanism of this intelligent optimization algorithm effectively solves the tracking accuracy problem. However, its efficiency may not meet the requirements for online operation.

Learning class methods based on artificial neural networks can realize fast planning solutions with the support of large amounts of training data [14]. Li et al. developed a nonlinear inverse kinematics model for a variable-section extensible-bendable continuum robot based on BP neural networks and achieved motion planning [15]. However, if the continuum robot has redundant degrees of freedom, the planning error may be significant.

The heuristic algorithm for planning the remaining joints of the continuum robot is based on empirical construction. It follows a sequential approach starting from the head of the robot and

^{*} Corresponding author.

E-mail address: imlixu@dlut.edu.cn (X. Li).

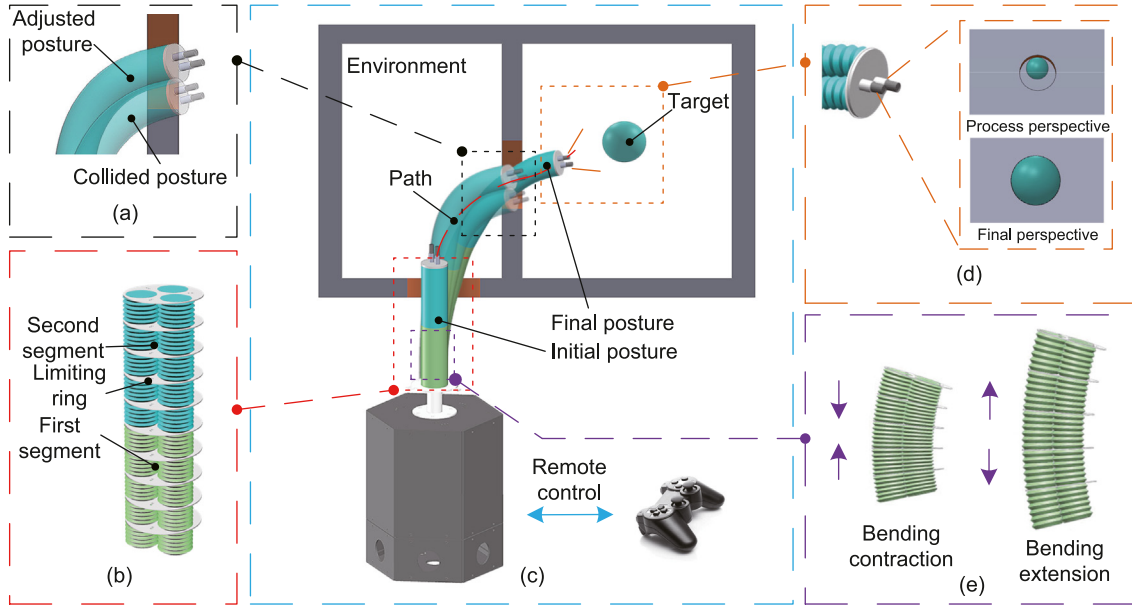


Fig. 1. The scalable and bendable continuum robotic system. The middle figure (c) shows the continuum robot in an obstacle environment starting with an initial posture to detect the target, following the path using the head-following algorithm, and adjusting the posture based on the local posture adjustment strategy to avoid obstacles. The left figure (a) shows the posture of the robot before and after the deployment of the posture adjustment strategy. The left figure (b) shows the initial posture of the continuum robot. The right figure (d) shows the change of perspective of the vision sensor during motion. The right figure (e) shows the change of scale and bending of a single segment.

using the path points as the goal to ensure smooth motion along the path. Gilbert et al. proposed the Follow-The-Leader strategy, which involves the section-by-section growth of concentric tube continuum robots to enable the robot to follow the end [16]. Liu et al. proposed a two-layer geometric iterative method based on FABRIK to solve the inverse kinematics (IK) and path planning problems of a cable-driven segmented robot [17]. They also proposed a FABRIK-based two-layer geometric iterative method to solve the IK and path planning problems of a non-scalable continuum robot. Wu et al. proposed a CRRIK algorithm and an obstacle avoidance algorithm for a non-scalable continuum robot [18,19]. Heuristic algorithms based on empirical constructions generally have better computational efficiency and tracking accuracy. However, they can be affected by the joint points updating strategy, particularly when dealing with multiple solutions for composite motion robots.

In summary, the difficulty of motion planning for continuum robots is determined by the complexity of the application environment and their model nonlinearity. Research on motion planning for continuum robots with composite motions has not been thoroughly investigated. Developing online, flexible operation motion planning technology that considers efficiency, accuracy, and operation safety remains a challenge. To this end, this paper proposes a flexible head-following motion planning method for scalable and bendable continuum robots. The main contributions are:

- (1) This paper proposes a neural-network-based path-adaptive head-following (PAHF) algorithm for continuum robots. The algorithm introduces a path-adaptive auxiliary points model in the process of updating the robot's joint points. This allows the robot to efficiently follow a path while maintaining good tracking accuracy, thereby improving its motion performance in complex environments.
- (2) This paper proposes a strategy for local posture adjustment in continuum robots, allowing them to adjust their posture while following a path. This enhances the robot's safety when avoiding obstacles.

- (3) The proposed algorithm's computational efficiency, tracking accuracy, obstacle avoidance capability, and algorithm universality are demonstrated through a series of simulations and prototype experiments. These results provide a reliable technical solution for online safe operation motion planning of scalable and bendable continuum robots in complex environments.

2. Methods

2.1. Kinematics modeling

The forward kinematics modeling of a scalable and bendable continuum robot is performed based on the segmental constant curvature (PCC) assumption. As shown in Fig. 1, the continuum robot consists of multiple segments, each with three equivalent degrees of freedom for pitch, yaw, and telescoping.

As shown in Fig. 2, take the i th ($i = 1, 2, \dots, n$) segment model as an example, establish the coordinate systems and at both ends of the segment. By translating the base coordinate system $\{x_{i-1}, y_{i-1}, z_{i-1}\}$ along $\vec{P}_{i-1}P_i$, and then rotated counterclockwise by φ_i along the z_{i-1} axis, counterclockwise by θ_i along the y_{i-1} axis, and clockwise by φ_i along the z_{i-1} axis, which is coincident with the end coordinate system $\{x_i, y_i, z_i\}$, the homogeneous transformation matrix of a single segment of the continuum robot can be expressed as follows:

$${}^i_{i-1}T = \text{Trans}(P_{i-1}P_i)\text{Rot}_z(\varphi_i)\text{Rot}_y(\theta_i)\text{Rot}_z(-\varphi_i)$$

$$= \begin{bmatrix} c^2\varphi_i c\theta_i + s^2\varphi_i & c\varphi_i s\varphi_i c\theta_i - c\varphi_i s\varphi_i & c\varphi_i s\theta_i & \frac{L_i}{\theta_i} c\varphi_i(1 - c\theta_i) \\ c\varphi_i s\varphi_i c\theta_i - c\varphi_i s\varphi_i & c^2\varphi_i c\theta_i + c^2\varphi_i & s\varphi_i s\theta_i & \frac{L_i}{\theta_i} s\varphi_i(1 - c\theta_i) \\ -c\varphi_i s\theta_i & -s\varphi_i s\theta_i & c\theta_i & \frac{L_i}{\theta_i} s\theta_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where the parameters are shown in Table 1. So the posture of the continuum robot is expressed as $C = [L_1, L_2, \dots, L_n, \varphi_1, \varphi_2, \dots, \varphi_n, \theta_1, \theta_2, \dots, \theta_n]$.

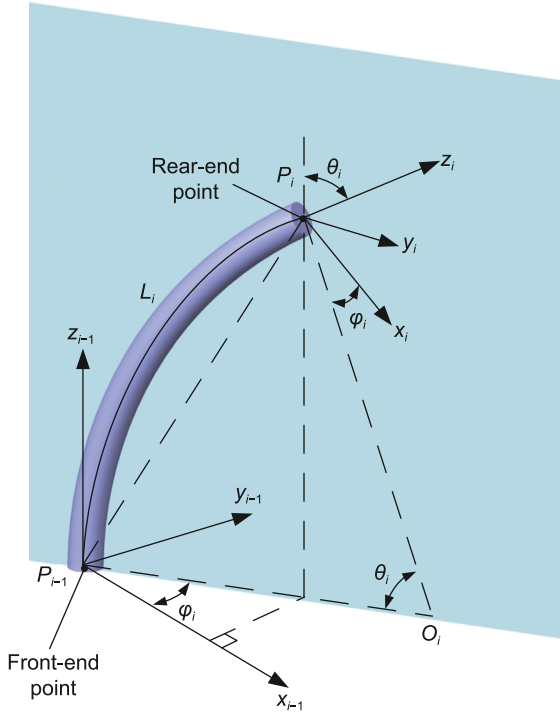


Fig. 2. Kinematics model.

Table 1

Nomenclature used in kinematic modeling and algorithm.

Symbol	Representation
n	Number of the segments
m	Number of the control points
p	Number of the target path points
q	Number of the model parameters traversals
r	Number of the robot's discrete points
i	Index of the segments, $i = 1, 2, \dots, n$
j	Index of the single-section control points, $j = 1, 2, \dots, m$
k	Index of the path points, $k = 1, 2, \dots, p$
l	Index of the model parameters traversals, $l = 1, 2, \dots, q$
s	Index of the robot's discrete points, $s = 1, 2, \dots, r$
L_i	Length of the i th segment
φ_i	Rotation angle of the i th segment
θ_i	Bending angle of the i th segment in the bending plane; indicates that the i th segment is straight.
$\theta_i = 0$	
L_{path_1}, L_{path_2}	Length of the target path
$\varphi_{path_1}, \varphi_{path_2}$	Rotation angle of the target path
$\theta_{path_1}, \theta_{path_2}$	Bending angle of the target path

The continuum robot is composed of segments connected in series, and applying the chain rule, the homogeneous transformation matrix between the robot coordinate system $\{x_{i-1}, y_{i-1}, z_{i-1}\}$ of the robot's rear-end and the base coordinate system $\{x_0, y_0, z_0\}$ is

$${}^n_0T = {}^1_0T^2 \cdots {}^n_{n-1}T = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix} \quad (2)$$

where R is a 3×3 posture matrix of the robot's rear-end and P is a 3×1 position matrix of the robot's rear-end.

2.2. Path-adaptive head-following (PAHF) motion planning

2.2.1. Auxiliary-points-based joint points update model

The process of head-following motion planning for continuum robots involves setting the rear-end point at the target path points and then sequentially updating the subsequent joint points to move along the path. The scalable and bendable continuum

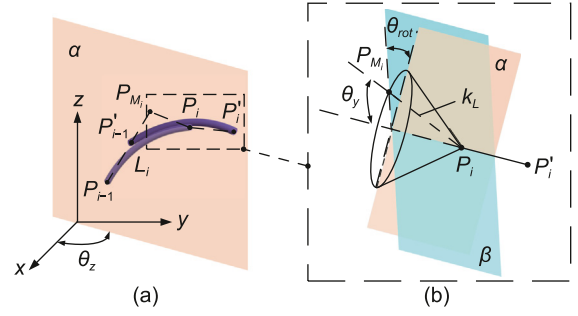


Fig. 3. Single-segment joint point update.

robot has redundant solutions, and its joint points update strategy affects tracking accuracy and computational efficiency. To balance computational efficiency and tracking accuracy, we propose an auxiliary-points-based joint points update model.

As shown in Fig. 3(a), taking segment i as an example, joint point P_i is updated to the next target point P'_i , and joint point P_{i-1} follows the motion to P'_{i-1} . P_{M_i} is an auxiliary point set to determine the direction of $\overrightarrow{P_{i-1}P'_i}$.

The auxiliary point P_{M_i} is located on the conic plane defined by the vertex P_i and the axis $P_iP'_i$, and it satisfies the given relation:

$$P_{M_i} = P_i + k_L F(\theta_{rot}) G(\theta_y) \overrightarrow{P_iP'_i} \quad (3)$$

$$F(\theta_{rot}) = I \cos(\theta_{rot}) + (1 - \cos(\theta_{rot})) \begin{pmatrix} \overrightarrow{n_i n_i^T} \\ \overrightarrow{n_i n_i^T} \end{pmatrix} \begin{bmatrix} 0 & -\overrightarrow{n_{iz}} & \overrightarrow{n_{iy}} \\ \overrightarrow{n_{iz}} & 0 & \overrightarrow{n_{ix}} \\ -\overrightarrow{n_{iy}} & \overrightarrow{n_{ix}} & 0 \end{bmatrix} \quad (4)$$

$$G(\theta_y) = \text{Rot}_z(\theta_z) \text{Rot}_y(\theta_y) \text{Rot}_z(-\theta_z) \quad (5)$$

where $\overrightarrow{n_i}(\overrightarrow{n_{ix}}, \overrightarrow{n_{iy}}, \overrightarrow{n_{iz}})$ is the unit direction vector of $\overrightarrow{P_iP'_i}$, k_L is the length of the generatrix, θ_y is the oblique angle of this conic, θ_{rot} is the angle between the plane α and the plane β , and θ_z is the angle between the plane α and the plane xoz .

The point P'_{i-1} is located on the line where $P_{i-1}P_{M_i}$ is located and its updated formula is

$$P'_{i-1} = P_{i-1} + k_{\text{direction}_i} \frac{\overrightarrow{P_{i-1}P_{M_i}}}{\|\overrightarrow{P_{i-1}P'_{i-1}}\|} \|\overrightarrow{P_{i-1}P'_{i-1}}\| \quad (6)$$

where $k_{\text{direction}_i}$ is the direction parameter during the updating of the front-end point P_{i-1} of the i th segment.

As shown in Fig. 4, when $\theta_{\text{direction}_i} < \pi/2$, the update vector $\overrightarrow{P_{i-1}P'_{i-1}}$ is updated in the same direction as $\overrightarrow{P_{i-1}P_{M_i}}$. Conversely, it is updated backward. $k_{\text{direction}_i}$ satisfies the following equation:

$$k_{\text{direction}_i} = \begin{cases} 1 & \theta_{\text{direction}_i} < \pi/2 \\ -1 & \theta_{\text{direction}_i} \geq \pi/2 \end{cases} \quad (7)$$

where $\theta_{\text{direction}_i}$ is the angle between $\overrightarrow{P_{i-1}P_{M_i}}$ and $\overrightarrow{P_{i-1}P'_{i-1}}$.

In determining the length of $\overrightarrow{P_{i-1}P'_{i-1}}$ for each segment, the principle of ensuring the uniformity of the change in length for each segment of the scalable and bendable continuum robot, to the greatest extent possible, is satisfied by

$$\|\overrightarrow{P_{i-1}P'_{i-1}}\| = k_{i-1} \|\overrightarrow{P_iP'_i}\| \quad (8)$$

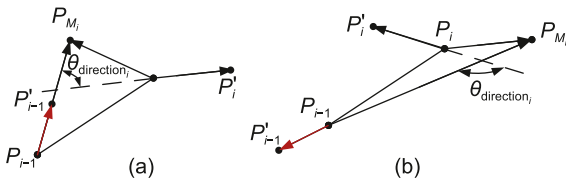


Fig. 4. Single-segment update direction.

$$k_{i-1} = \frac{L_1 + L_2 + \dots + L_{i-1}}{L_1 + L_2 + \dots + L_i} \quad (9)$$

where L_i is the length of segment i and k_{i-1} is the scale factor of $\vec{P_{i-1}P'_i}$.

2.2.2. Auxiliary points setting method considering tracking accuracy

Eqs. (3)–(5) show that the position of the auxiliary points is related to the model parameters $IV = [k_L, \theta_{rot}, \theta_y]$. The selection of the auxiliary points affects the updated position of the joint target, which in turn affects the tracking accuracy of different target paths. Therefore, a method for selecting the auxiliary points based on a BP neural network is proposed.

Fig. 5 shows the initial robot configuration as $C_{init} = [L_{1min}, L_{2min}, L_{3min}, 0, 0, 0, 0, 0, 0]$. The robot's discrete points are represented by $Points_{Robot} = \{R_1, R_2, \dots, R_s, \dots, R_r\}$. The expression of the target path is the same as that of the robot configuration and the target path parameters are defined as $C_{Path} = [L_{Path_1}, L_{Path_2}, \varphi_{Path_1}, \varphi_{Path_2}, \theta_{Path_1}, \theta_{Path_2}]$. The target path points are described as $Points_{Path} = \{Q_1, Q_2, \dots, Q_k, \dots, Q_p\}$.

Using IV_i as the model parameters and C_{init} as the initial configuration, the continuum robot follows the target path $Path$. d_{kmax} is the path deviation when the rear-end point of the robot moves to the path point Q_k and can be expressed as:

$$d_{kmax} = \max\{\|R_1Q_1\|, \dots, \|R_sQ_k\|, \dots, \|R_rQ_p\|\} \quad (10)$$

where $s = 1, 2, \dots, r$ and $k = 1, 2, \dots, p$.

Then the robot's tracking accuracy $Error_{Path}$ can be expressed as the maximum value of the path deviation of the whole process:

$$Error_{Path} = \max\{d_{1max}, \dots, d_{kmax}, \dots, d_{pmax}\} \quad (11)$$

Under a specific target path, the corresponding $Error_{Path_i}$ is obtained by traversing the model parameters IV_i , and the optimal value of which is used as the optimal tracking accuracy $Error_{best}$, and the parameter corresponding to the best tracking accuracy is taken as the optimal parameter IV_{best} :

$$Error_{best} = \min\{Error_{Path_1}, Error_{Path_2}, \dots, Error_{Path_i}, \dots, Error_{Path_q}\} \quad (12)$$

To achieve high-precision path-tracking of various target paths, we construct a path-adaptive network. The network takes the target path parameters C_{Path} as input and the adaptive parameters IV_{best} as output. The target path parameters, as input to the network, are limited by the structure of the continuum robot. So we generate random paths within a certain parameter range according to the above requirements and structural restrictions, obtain training sample data, and train the network. The network can predict the path-adaptive parameters IV_{best} in the robot path-tracking process, enabling the determination of the optimal direction of the auxiliary points during joint points updating.

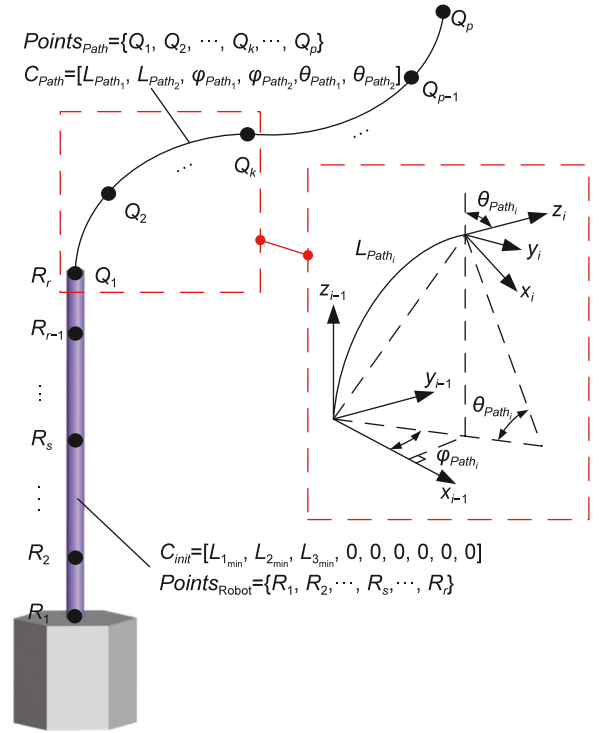


Fig. 5. The robot tracks the target path with an initial posture.

2.3. Local posture adjustment strategy

Due to the complexity of the robot's environment and the constraints of motion accuracy, the continuum robot may experience local joint interference when performing head-following motion. Therefore, a strategy for adjusting local joints is proposed to enhance the robot's flexible manipulation capability.

2.3.1. Control points setting

As shown in Fig. 6, some segments of robot may collide with the environment during robot motion planning. For this segment, uniformly distribute m (m is an even number) control points, denoted as $P_{i,j}$. The number of control points can be adjusted based on the geometry of robot, computational real-time requirements, and operational flexibility.

2.3.2. Joint points correction vector

Fig. 6 shows the correction vectors for the j th control point of the i th segment, which is used to correct the robot's posture. As the continuum robot's motion is actively driven by segments, it is not possible to control any point within a segment. Therefore, the correction vector of the control points is converted to the neighboring joint points.

As shown in Fig. 6(a), the control points for non-base and non-end segments are divided into two parts. The correction vector for the first $m/2$ control points is assigned to the front-end point of the segment, while the correction vector for the second $m/2$ control points is assigned to the rear-end point of the segment and converted using the proportionality relation. The specific correction vectors are calculated as shown below:

$$\vec{N}_i = \frac{2}{m} \sum_{j=m/2+1}^{m-1} \frac{m-1}{j-1} \vec{N}_{i,j} + \frac{2}{m} \sum_{j=1}^{m/2} \frac{m-1}{m-j} \vec{N}_{i+1,j}, \quad 2 \leq i \leq n-2 \quad (13)$$

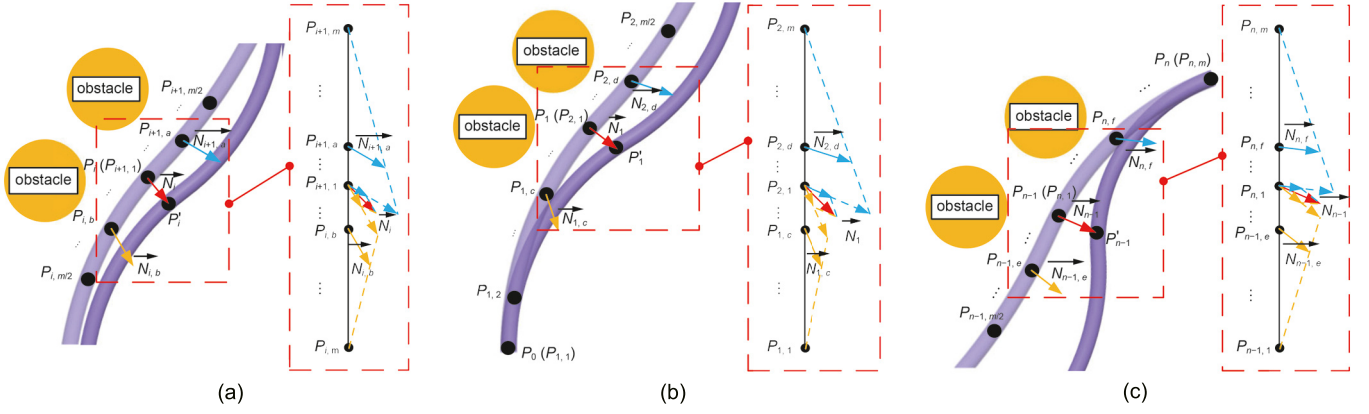


Fig. 6. Local posture adjustment of a three-segment continuum robot. (a) The non-base and non-end segments. (b) The base segment. (c) The end segment.

As shown in Fig. 6(b) and (c), for the base segment and the end segment, the base point and the rear-end point cannot move during the posture adjustment process. Therefore the correction vectors of the base point and the first segment rear-end point are

$$\vec{N}_0 = 0 \quad (14)$$

$$\vec{N}_1 = \frac{1}{m} \sum_{j=1}^{m-1} \frac{m-1}{j-1} \vec{N}_{1,j} + \frac{2}{m} \sum_{j=1}^{m/2} \frac{m-1}{m-j} \vec{N}_{2,j} \quad (15)$$

The rear-end point and n th segment front-end point correction vectors are

$$\vec{N}_n = 0 \quad (16)$$

$$\vec{N}_{n-1} = \frac{2}{m} \sum_{j=m/2+1}^{m-1} \frac{m-1}{j-1} \vec{N}_{n-1,j} + \frac{1}{m} \sum_{j=1}^{m-1} \frac{m-1}{m-j} \vec{N}_{n,j} \quad (17)$$

The next position of the joint point is

$$P'_i = P_i + \vec{N}_i \quad (18)$$

3. Simulation analysis

To verify the effectiveness of the algorithms, we conducted simulation experiments on path-tracking and local posture adjustment using a three-segment scalable and bendable continuum robot as an example. The initial length of each segment is $L_1 = 50$ mm, $L_2 = 60$ mm, and $L_3 = 60$ mm. We compared the performance of the PSO, FABRIK, and CRRIK algorithms on a computer equipped with an Intel i5-12400F (2.50 GHz) and NVIDIA GeForce GTX 1650 SUPER (4 GB).

3.1. Validation of the effectiveness of path-adaptive auxiliary points model

To verify the effect of the adaptive auxiliary points adjustment strategy proposed in this paper on tracking accuracy, comparative simulation experiments are conducted. We randomly selected three groups of fixed model parameters $IV_l = [k_{l_i}, \theta_{rot_l}, \theta_{y_l}]$ ($l = 1, 2, 3$) and path-adaptive parameters $IV_{pre} = [k_{l_{pre}}, \theta_{rot_{pre}}, \theta_{y_{pre}}]$ predicted by a neural network for path-tracking experiments on three different types of paths. The model and path parameters for each group are listed below:

$$IV_1 = [0, 0, 0] \quad (19)$$

$$IV_2 = [5, \frac{\pi}{3}, 0] \quad (20)$$

Table 2

Tracking accuracy for paths with different parameters.

Paths	Tracking accuracy (mm)			
	IV_1	IV_2	IV_3	IV_{pre}
$Path_1$	12.2522	8.7057	13.8371	3.9552
$Path_2$	16.7528	12.6707	15.4372	8.9694
$Path_3$	16.1166	12.9734	17.3307	10.7538

$$IV_3 = [7, \frac{\pi}{2}, \frac{\pi}{4}] \quad (21)$$

$$C_{Path_1} = [60, 0, 0, 0, \frac{2\pi}{5}, 0] \quad (22)$$

$$C_{Path_2} = [50, 50, 0, \pi, \frac{\pi}{4}, \frac{\pi}{3}] \quad (23)$$

$$C_{Path_3} = [50, 50, 0, \frac{\pi}{2}, \frac{\pi}{4}, \frac{\pi}{3}] \quad (24)$$

where $Path_1$ is a single-segment circular path, $Path_2$ is a planar two-segment circular path, and $Path_3$ is a spatial two-segment circular path.

Fig. 7 shows that for the three target paths with varying degrees of complexity, the area of the blue deviation region under the parameters IV_{pre} are the smallest, indicating the highest tracking accuracy of the robot. The resulting blue deviation areas for the three randomly selected parameters, IV_1 , IV_2 , and IV_3 , are large to varying degrees, indicating significant path deviations during tracking. The top view on the lower side of Fig. 7 also shows the highest robot tracking accuracy with IV_{pre} .

Table 2 shows the specific tracking deviation values. The tracking deviation for $Path_1$ and $Path_2$ is 2 ~3 times lower under the IV_{pre} than under the random parameters. For $Path_3$, the tracking deviation under the IV_{pre} is more than 2 mm smaller.

In summary, the results indicate that the auxiliary points adaptive adjustment strategy is effective in improving the robot's tracking accuracy under different target paths.

3.2. Simulation and comparative experiments on path-tracking

To verify the effectiveness of the Path-adaptive head-following (PAHF) algorithm, we compared it with the PSO, FABRIK, and CRRIK algorithms for tracking three different types of paths, each in 2 mm steps.

The results of path-tracking are presented in Fig. 8. PAHF and PSO exhibit excellent tracking accuracy, followed by FABRIK, while CRRIK has the worst tracking accuracy.

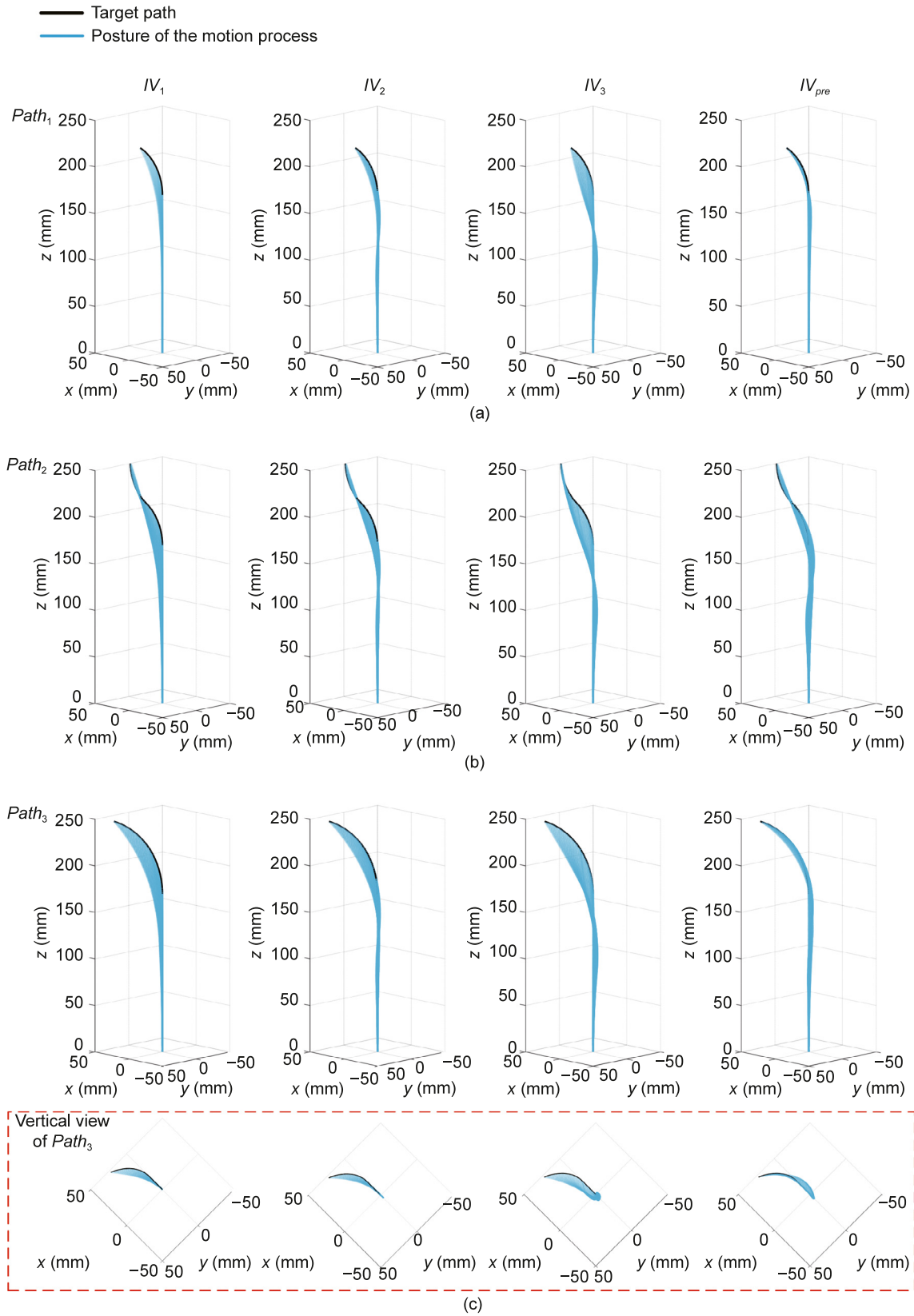


Fig. 7. Effect of model parameters on path following motion.

Table 3 shows the tracking accuracy of the path-tracking motion. The PAHF algorithm exhibits a significant advantage in terms of tracking accuracy, with an average tracking accuracy of 7.8928 mm for different paths. This value is 1.88 and 3.78 times

higher than that of FABRIK and CRRIK. However, it is slightly lower than the non-real-time PSO algorithm, which has a tracking accuracy 0.77 times that of PSO. For a circular path of type $Path_1$ with a single segment, the tracking accuracy is 3.10 and 5.92

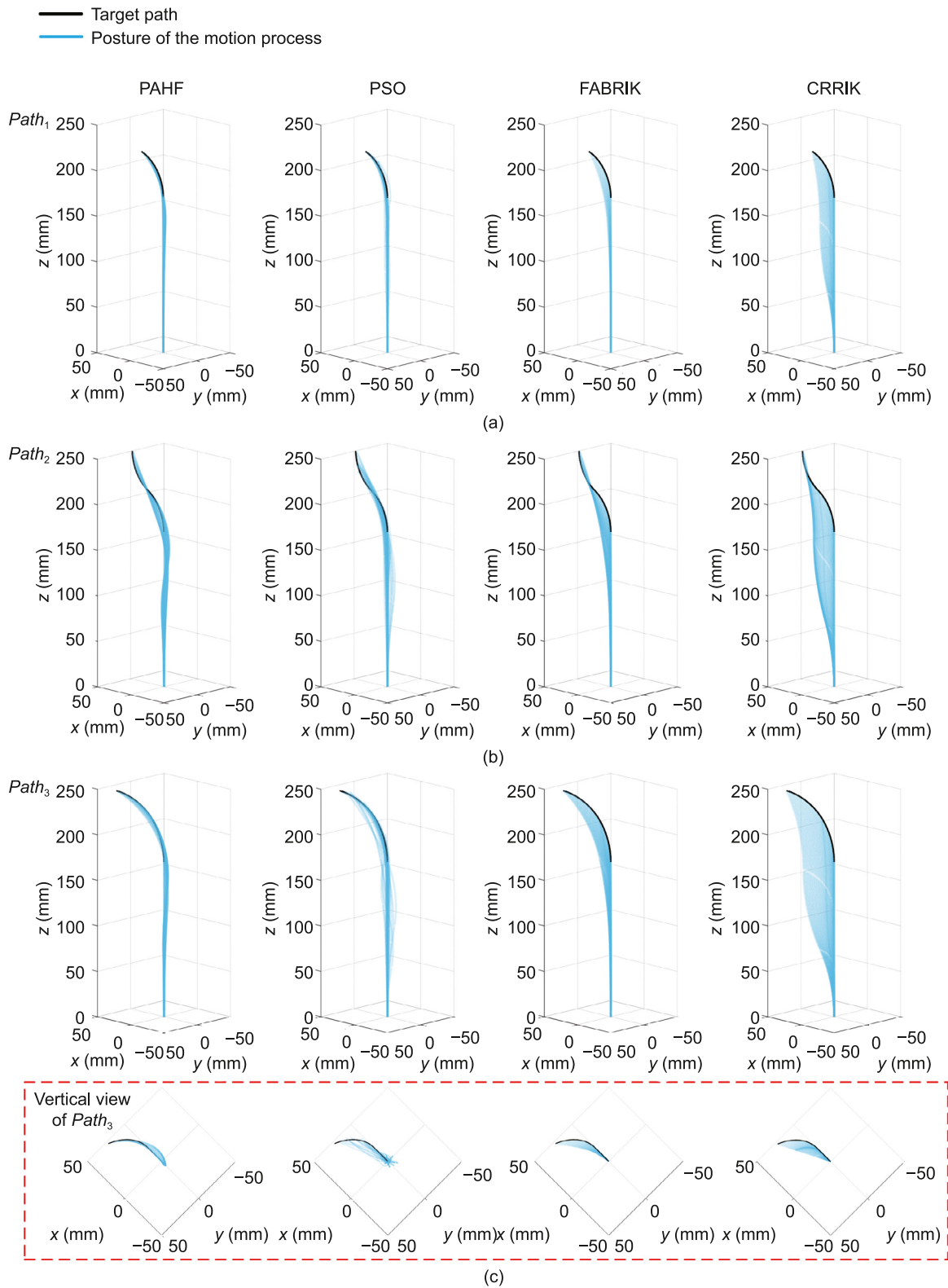


Fig. 8. Path-tracking simulation results. (a) Tracking results of algorithms for $Path_1$. (b) Tracking results of algorithms for $Path_2$. (c) Tracking results of algorithms for $Path_3$.

times higher than that of FABRIK and CRRIK. In the case of planar two-segment circular paths of type $Path_2$, the tracking accuracies are 1.59 and 3.46 times higher, while for spatial two-segment circular paths of type $Path_3$, the tracking accuracies are 1.46 and 3.26 times higher.

Table 4 shows the average single-step solution time (100 times) in path-tracking motion. The average solution time for different paths using PAHF is 4.41×10^{-5} s, which is 75295.26 times longer than the PSO algorithm. PAHF uses neural network

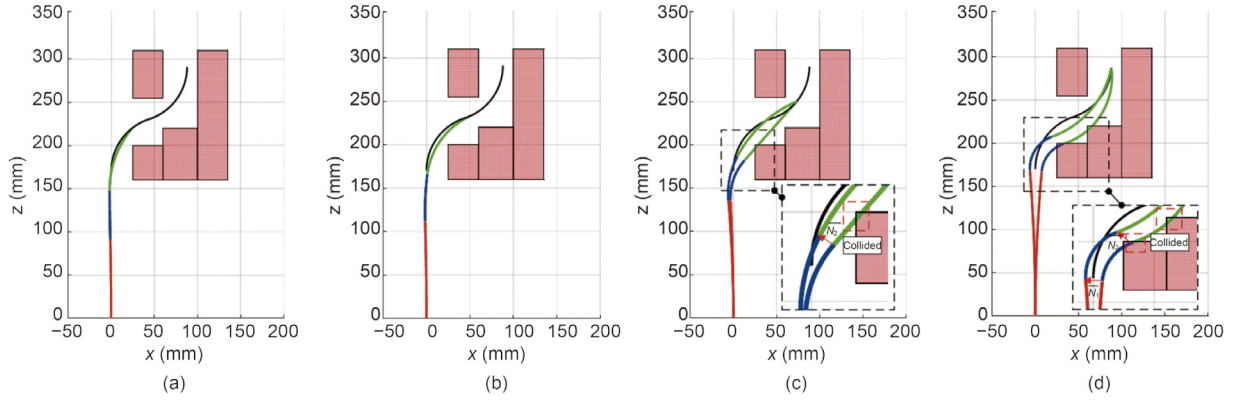


Fig. 9. Posture adjustment simulation result. (a) and (b) show the collision-free path-tracking process of the robot. (c) and (d) show the local posture adjustment in case of collision with obstacles in the subsequent motion of the robot.

Table 3

Tracking accuracy for paths with different algorithms.

Paths	Tracking accuracy (mm)			
	PAHF	PSO	FABRIK	CRRIK
$Path_1$	3.9552	2.9988	12.2642	23.4048
$Path_2$	8.9694	12.6707	15.4372	30.9937
$Path_3$	10.7538	12.9734	17.3307	35.0392

Table 4

The average single-step solution time (100 times) for paths with different algorithms.

Paths	Time (s)			
	PAHF	PSO	FABRIK	CRRIK
$Path_1$	5.61×10^{-5}	3.1892	0.94×10^{-5}	0.91×10^{-5}
$Path_2$	3.85×10^{-5}	3.3453	0.96×10^{-5}	0.95×10^{-5}
$Path_3$	3.76×10^{-5}	3.4223	0.95×10^{-5}	0.94×10^{-5}

prediction, which is slightly less efficient than heuristic algorithms like FABRIK and CRRIK. However, the difference is only at the microsecond level, which meets real-time requirements. The time gap will decrease with an increase in the number of feed steps. The efficiency of the PSO algorithm depends on tracking accuracy. Under the guarantee of optimal tracking accuracy, the PSO algorithm requires 56848.48, 86890.91, and 91018.62 times more time than the PAHF algorithm for $Path_1$, $Path_2$, and $Path_3$.

The results indicate that the PAHF algorithm performs better in terms of tracking accuracy, particularly when dealing with single-segment circular paths. This is achieved while ensuring real-time robot motion.

3.3. Simulation of local posture adjustment

To verify the effectiveness of the local posture adjustment strategy, the robot performs obstacle avoidance and path-tracking motions in a confined space. During the obstacle environment, the robot tracks $Path_4$, with the C_{init} . The parameters of the target path are listed below:

$$C_{Path_4} = \left[80, 80, 0, \pi, \frac{2\pi}{5}, \frac{2\pi}{5} \right] \quad (25)$$

Fig. 9 shows the variations in motion of the robot's head-following motion when deploying the posture adjustment strategy in a confined space. The robot's end moves along the target path throughout the motion, and Fig. 9(a) and (b) demonstrate the robot's excellent tracking accuracy during the head-following motion. As shown in Fig. 9(c) and (d), the robot's local joints

come into contact with obstacles during its motion. To ensure the robot's safety during motion, a local joint adjustment strategy is employed to maintain a safe distance from the obstacles.

4. Experiments and results

4.1. Experimental platform

To further verify the algorithm's effectiveness, an experimental platform for a scalable and bendable continuum robot is designed, as shown in Fig. 10. According to the actual parameters of the robot, the path-adaptive auxiliary point model is trained, and the path-tracking and local adjustment experiments are carried out on the experimental platform.

The robot has two segments that can be scalable and bendable. These segments have a minimum length of $L_{min} = [105 \text{ mm}, 170 \text{ mm}]$ and a maximum length of $L_{max} = [170 \text{ mm}, 245 \text{ mm}]$. The robot is designed to maintain its initial length and not exceed its maximum elongation limit.

4.2. Path-tracking experiment

To further verify the path-tracking capability of the continuum robot, prototype experiments were conducted on the robotics experimental platform.

Fig. 11 shows the results of the path-tracking experiments conducted on the continuum robot. The robot was tested on two single-segment circular paths, P_1P_2 and P_3P_4 , with different curvatures. The experiments were conducted with an initial posture. And the target path P_1P_2 and P_3P_4 are shown:

$$C_{P_1P_2} = \left[100, 0, 0, 0, \frac{5\pi}{9}, 0 \right] \quad (26)$$

$$C_{P_3P_4} = \left[120, 0, 0, 0, \frac{7\pi}{18}, 0 \right] \quad (27)$$

The experimental results indicate that the robot's path-adaptive head-following algorithm exhibits exceptional tracking accuracy.

4.3. Local posture adjustment experiment

To further verify the local posture adjustment capability of the continuum robot, prototype experiments were conducted on the robotics experimental platform.

Fig. 12 shows the robot's capability to adjust its posture during motion. The continuum robot tracks the end path P_5P_6 with an initial posture. And the target path P_5P_6 are shown:

$$Pointset_{P_5P_6} = [(0, 0, 265), (0, 150, 265)] \quad (28)$$

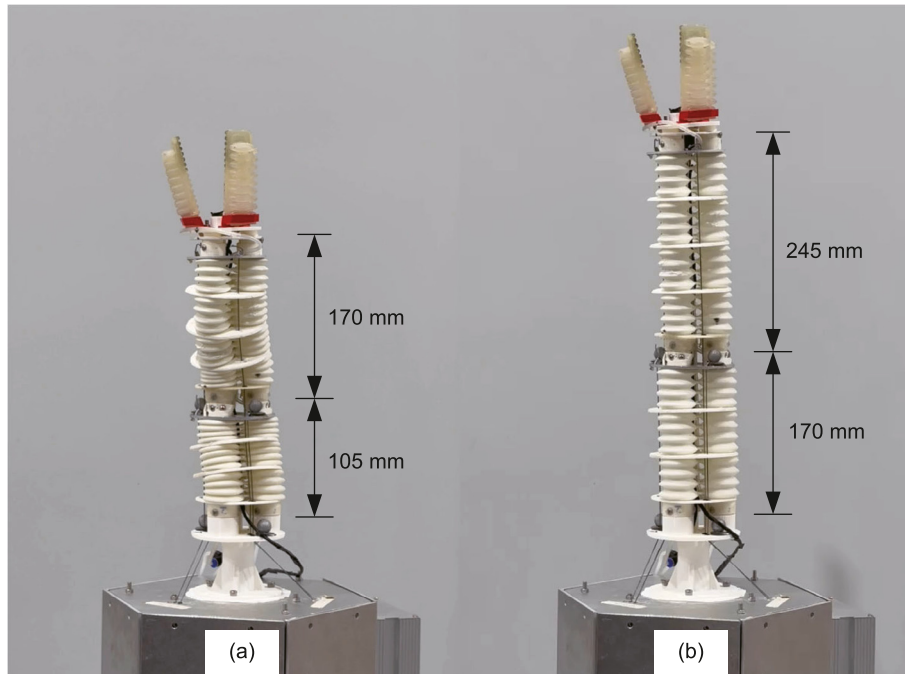


Fig. 10. Scalable and bendable continuum robot experimental platform.

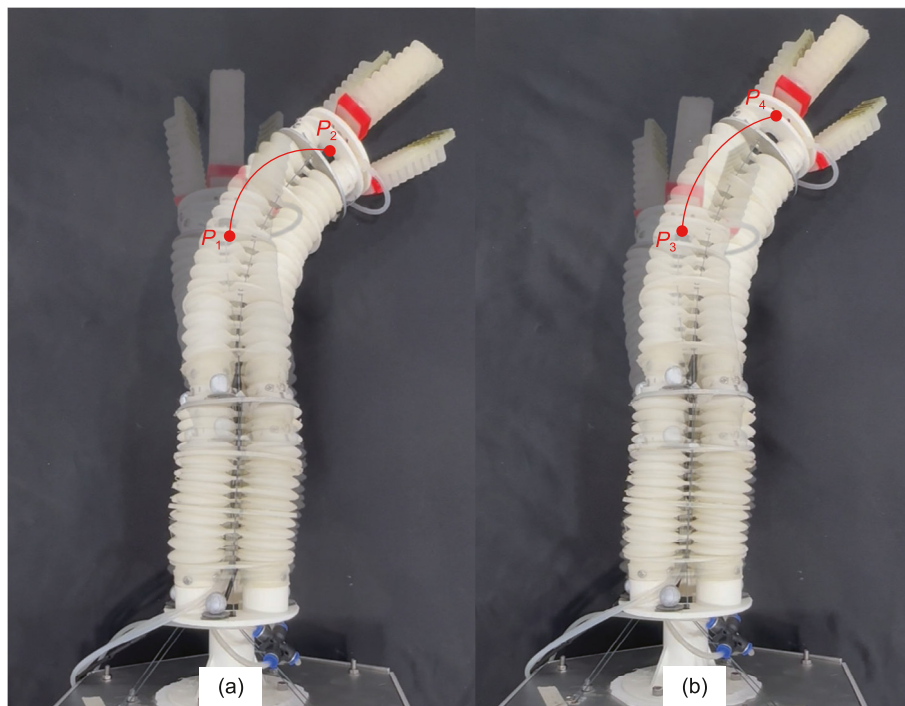


Fig. 11. Path-tracking experiment.

When the robot is about to collide during motion, a local posture adjustment strategy is deployed to ensure collision-free path-tracking while keeping the rear-end point's position unchanged.

Fig. 13 shows the robot's capability to adjust its posture during head-following path-tracking motion. The continuum robot tracks the end path P_7P_8 with an initial posture. And the target path P_7P_8 are shown:

$$C_{P_7P_8} = \left[150, 0, 0, 0, \frac{\pi}{4}, 0 \right] \quad (29)$$

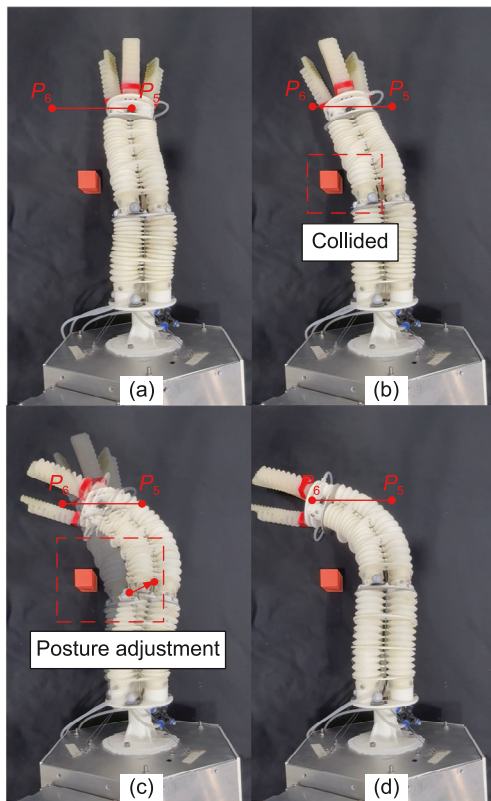


Fig. 12. Posture adjustment experiment. (a) The robot tracks the path P_5P_6 with an initial posture. (b) The robot is about to collide during the motion. (c) The robot deploys the local posture adjustment strategy, where the rear-end point remains unchanged to adjust the local posture. (d) The robot completes path-tracking with a collision-free posture.

When the robot is about to collide during motion, a local posture adjustment strategy is deployed to ensure collision-free path-tracking while keeping the rear-end point's position unchanged.

The experimental results indicate that the robot's local posture adjustment strategy ensures the safety of its motion process.

5. Conclusion

This paper proposes a head-following motion planning method for scalable and bendable continuum robots to address the issue of real-time flexible operation in narrow environments. The method is based on the PCC model and introduces an adaptive auxiliary points model and a head-following motion joint points update method. Additionally, a local posture adjustment strategy is integrated into the head-following motion planning. The method demonstrates effective head-following and local obstacle avoidance posture adjustment motion capabilities through multiple sets of simulation experiments and prototype experiments. The algorithm meets the real-time requirement for online operation while maintaining an average tracking accuracy of 7.8928 mm, providing a reliable solution for the flexible operation of continuum robots.

In future work, we will investigate error compensation measurements of the PCC model to improve the algorithm's reliability in practical applications. Additionally, we will develop the robot's perception system, combined with the SLAM algorithm, to enable the robot's motion planning capability in unknown environments.

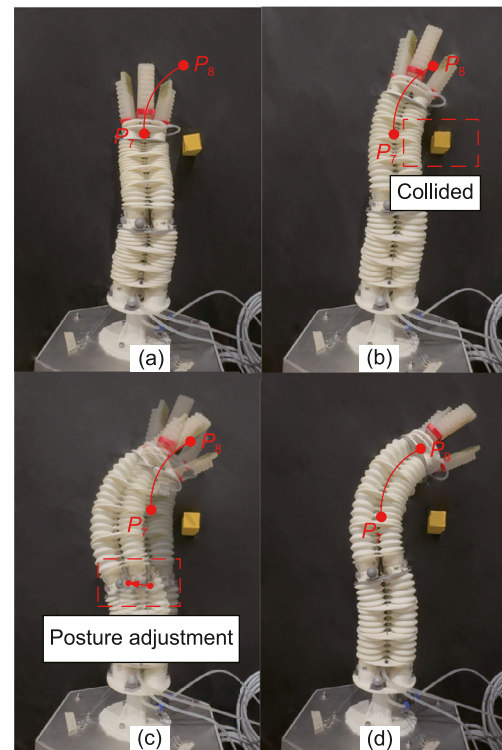


Fig. 13. Head-following path-tracking experiment with posture adjustment. (a) The robot tracks the path P_7P_8 with an initial posture. (b) The robot is about to collide during the motion. (c) The robot deploys the local posture adjustment strategy, where the rear-end point remains unchanged to adjust the local posture. (d) The robot completes path-tracking with a collision-free posture.

CRedit authorship contribution statement

Te Li: Funding acquisition, Writing – review & editing. **Guoqing Zhang:** Data curation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. **Xinyuan Li:** Supervision, Writing – review & editing. **Xu Li:** Writing – review & editing. **Haibo Liu:** Writing – review & editing. **Yongqing Wang:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported in part by the Fundamental Research Funds for the Central Universities, China (DUT22GF301).

References

- [1] I.D. Walker, Robot strings: Long, thin continuum robots, in: *Aerospace Conference*, 2013.
- [2] J. Burgner-Kahrs, D.C. Rucker, H. Choset, Continuum robots for medical applications: A survey, *IEEE Trans. Robot.* 31 (6) (2015) 1261–1280.
- [3] T. TolleyMichael, F. ShepherdRobert, C. GallowayKevin, J. WoodRobert, M. WhitesidesGeorge, et al., A resilient, untethered soft robot, *Soft Robot.* (2014).
- [4] Y. Goergen, G. Rizzello, S. Seelecke, P. Motzki, Modular design of an SMA driven continuum robot, in: *Smart Materials, Adaptive Structures and Intelligent Systems*, vol. 84027, American Society of Mechanical Engineers, 2020, V001T04A007.

- [5] B.A. Jones, I.D. Walker, A new approach to Jacobian formulation for a class of multi-section continuum robots, in: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, IEEE, 2005, pp. 3268–3273.
- [6] Y. Gao, K. Takagi, T. Kato, N. Shono, N. Hata, Continuum robot with follow-the-leader motion for endoscopic third ventriculostomy and tumor biopsy, *IEEE Trans. Biomed. Eng.* 67 (2) (2019) 379–390.
- [7] M. Giorelli, F. Renda, M. Calisti, A. Arienti, G. Ferri, C. Laschi, Neural network and Jacobian method for solving the inverse statics of a cable-driven soft arm with nonconstant curvature, *IEEE Trans. Robot.* 31 (4) (2015) 823–834.
- [8] A. Martin, A. Barrientos, J. Del Cerro, The natural-CCD algorithm, a novel method to solve the inverse kinematics of hyper-redundant and soft robots, *Soft Robot.* 5 (3) (2018) 242–257.
- [9] A. Aristidou, J. Lasenby, FABRIK: A fast, iterative solver for the inverse kinematics problem, *Graph. Models* 73 (2011) (2011) 243–260.
- [10] W. Zhang, Z. Yang, T. Dong, K. Xu, FABRIKc: An efficient iterative inverse kinematics solver for continuum robots, in: 2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM, IEEE, 2018, pp. 346–352.
- [11] K. Xu, J. Zhao, M. Fu, Development of the SJTU unfoldable robotic system (SURS) for single port laparoscopy, *IEEE/ASME Trans. Mechatronics* 20 (5) (2014) 2133–2145.
- [12] G. Gao, P. Wang, H. Wang, Follow-the-leader motion strategy for multi-section continuum robots based on differential evolution algorithm, *Ind. Robot: Int. J. Robot. Res. Appl.* 48 (4) (2021) 589–601.
- [13] J. Santoso, C.D. Onal, An Origami continuum robot capable of precise motion through torsionally stiff body and smooth inverse kinematics, *Soft Robot.* 8 (4) (2021) 371–386.
- [14] T. George Thuruthel, E. Falotico, M. Manti, A. Pratesi, M. Cianchetti, C. Laschi, Learning closed loop kinematic controllers for continuum manipulators in unstructured environments, *Soft Robot.* 4 (3) (2017) 285–296.
- [15] X. Li, T. Zheng, D. Sui, N. Lin, Q. Zhang, J. Zhao, Y. Zhu, A 3D printed variable cross-section pneumatic soft manipulator with high-precision positioning capability: Design and control implementation, *Sensors Actuators A* 342 (2022) 113644.
- [16] H.B. Gilbert, J. Neimat, R.J. Webster, Concentric tube robots as steerable needles: Achieving follow-the-leader deployment, *IEEE Trans. Robot.* 31 (2) (2015) 246–258.
- [17] T. Liu, T. Yang, W. Xu, G. Mylonas, B. Liang, Efficient inverse kinematics and planning of a hybrid active and passive cable-driven segmented manipulator, *IEEE Trans. Syst. Man Cybern.: Syst.* 52 (7) (2021) 4233–4246.
- [18] H. Wu, J. Yu, J. Pan, G. Li, X. Pei, CRRIK: A fast heuristic algorithm for the inverse kinematics of continuum robot, *J. Intell. Robot. Syst.* 105 (3) (2022) 55.
- [19] H. Wu, J. Yu, J. Pan, X. Pei, A novel obstacle avoidance heuristic algorithm of continuum robot based on FABRIK, *Sci. Chin. Technol. Sci.* 65 (12) (2022) 2952–2966.