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

# A Constrained Path Following Method for Snake-like Manipulators via Controlled Winding Uncoiling Strategy

Mingrui Luo[1,2], Yunong Tian[1,2], Yinghua Cao[1,2], Minghao Chen[1,2], Yanfeng Zhang[1,2], En Li[1,2], and Min Tan[2]

***Abstract*— Benefiting from its hyper-redundant structure, the biomimetic snake-like manipulator retains its remarkable flexibility even within confined spaces. However, its motion planning and control pose significant challenges. This paper imitates the winding uncoiling behavior of snakes to achieve controllable constrained path following. Firstly, based on control points, a recursive computational model and an equivalent planning angle model are established, enabling efficient and analytical determination of joint positions, collision regions, and motion parameters during the path following. Subsequently, the sliding control point algorithm and motion smoothing restriction algorithm are designed. The former ensures that the remaining segments during following strictly remain within the collision-free regions defined by the base and path controls, while the latter smooths the control parameters based on velocity and acceleration limitations. Finally, simulation and practical experiments demonstrate the feasibility of the proposed methods. The prototype that applied our method can reach targets and accomplish tasks, further validating the applicability of the snake-like manipulator.**

## I. INTRODUCTION

Over the years, manipulation within unstructured and confined spaces has posed significant challenges, as manipulators need to maintain the required flexibility to accomplish tasks even while avoiding obstacles, joint singularities, and joint limits. Achieving these objectives simultaneously proves difficult for conventional industrial manipulators [1], [2]. Snakes exhibit a wide range of motion capabilities and demonstrate remarkable flexibility and agility in complex environments [3], [4]. Inspired by these characteristics, researchers have proposed various types of snake-inspired robots [5]. Among them, snake-like manipulators imitate the long backbone composed of numerous vertebrae in snakes, utilizing multiple segments of short rigid links [6]. These manipulators exhibit excellent adaptability to the environment and strong obstacle avoidance capabilities [7], [8].

The considerable redundancy of snake-like manipulators enhances their motion capabilities but also introduces challenges in motion planning and control [7]. In existing research, snake-like manipulators are primarily characterized by two motion modes: point-to-point motion, which centers on inverse kinematics solutions [9], [10], [11], and follow-the-leader (FTL) motion [12], [13], [14], which emulates the serpentine locomotion of snakes [15]. In FTL motion, only the guiding path for the head needs to be planned, and the remaining segments follow the head's trajectory sequentially, avoiding high-dimensional planning spaces [16]. However, challenges in FTL motion include efficiently computing the configurations of the remaining segments corresponding to the head's position, ensuring obstacle avoidance safety during the following [17], and addressing constrained motion conditions to ensure smooth following [18].

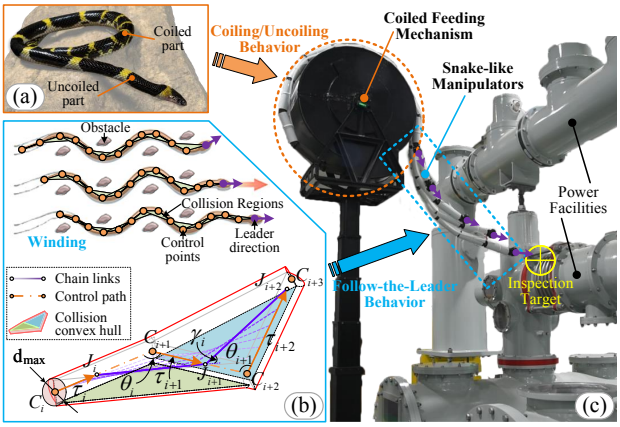


Fig. 1. Imitate the winding and coiling/uncoiling behaviors of snakes to achieve the spatial motion of snake-like manipulators. (a) Coiling is the basic behavior of snakes. (b) Follow-the-leader behavior based on control points. (c) The winding uncoiling motion simplifies task manipulation.

Existing methods for computing the configurations of the remaining segments include search-based [19], [20] and geometry-based [21], [22] approaches. Search-based methods face challenges in balancing computational cost and accuracy [23], especially when dealing with numerous constrained conditions [24]. On the other hand, geometry-based methods are more amenable to real-time implementation [25] and often use geometric model interpolation to obtain configurations. However, establishing a direct correlation between collision detection and constrained conditions within a geometric model presents challenges [26].

In light of the issues present in the existing following methods, this paper imitates the winding uncoiling behavior of snakes and proposes a constrained path following method for snake-like manipulators. The proposed method exhibits the following characteristics: (1) Establishment of a recursive

This work was supported in part by the National Natural Science Foundation under Grant 62273344 and 62206275. (Corresponding author: En Li.)

[1]The authors are with the Engineering Laboratory of Industrial Vision and Intelligent Equipment Technology, Institute of Automation, Chinese Academy of Sciences, 95 Zhongguancun East Road, Beijing 100190, China. (e-mail: luomingrui2020@ia.ac.cn; yunong.tian@ia.ac.cn; caoyinghua2022@ia.ac.cn; chenminghao2019@ia.ac.cn; zhangyanfeng2020@ia.ac.cn; en.li@ia.ac.cn)

[2]The authors are with the School of Artificial Intelligence, University of Chinese Academy of Sciences, No.19(A) Yuquan Road, Beijing, 100049, China. (e-mail: min.tan@ia.ac.cn)

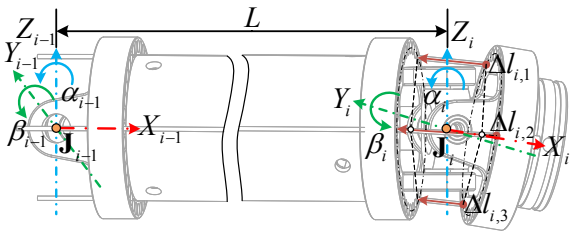


Fig. 2. Kinematics parameter definition and coordinate system.

computational model and an equivalent planning angle model based on control points, enabling efficient and analytical determination of joint positions, collision regions, and motion parameters during the following; (2) Introduction of a sliding control point algorithm, ensuring that the remaining segments strictly remain in the collision-free region defined by the base and path control; (3) Introduction of a motion smoothing restriction algorithm that smooths the control parameters based on velocity and acceleration limitations.

The remainder of this paper is organized as follows: Section II describes the motivation and definition; Section III establishes the control-point-based following model according to the definition; Section IV introduces the constrained path following algorithms based on the following model; Section V presents experimental results. The last section summarizes the whole paper and gives the conclusions.

## II. PROBLEM FORMULATION

### A. Bioinspired Motivation

Snakes employ different modes of locomotion based on diverse terrains and survival requirements. Coiling and winding are two fundamental and commonly observed forms of snake locomotion. Coiling provides enhanced concealment and defensive capabilities while winding facilitates exploration of unknown environments or the pursuit of prey. Inspired by these behaviors, a coiled feeding mechanism is designed to enable coiling and uncoiling motions in snake-like manipulators [11]. During coiling, the mechanism exhibits a compact structure for efficient storage, while during uncoiling, it offers agile movement for effective manipulation tasks. Furthermore, when snakes are winding, their bodies follow a series of control points sequentially, which reduces the risk of other body parts encountering obstacles as these points are located in areas already explored by the head. With this motivation, we integrate the controlled winding uncoiling strategy above with the FTL strategy and develop a control-point-based constrained path following method, thereby achieving efficient and reliable task manipulation for snake-inspired manipulators, as shown in Fig. 1.

### B. Kinematics Definition

The snake-like manipulator consists of multiple identical segments connected by orthogonal joints and actuated by three cables. Based on the coordinate system established in Fig. 2, the forward kinematics can be expressed as follows:

$$\mathbf{T}_{i+1} = \mathbf{T}_i \ \mathrm{Rot}(Z_i,\alpha_i)\mathrm{Rot}(Y_i,\beta_i)\mathrm{Trans}(X_i,L) \tag{1}$$

where $\mathbf{T}_i$ represents the homogeneous transformation matrix, $\alpha_i$ and $\beta_i$ ($|\alpha_i|,|\beta_i| \le \frac{\pi}{6}$) denote the joint angles around the Z-axis and Y-axis, respectively, and $L$ represents the distance between the centers of adjacent joints, Trans() and Rot() denote the translation and rotation functions, respectively.

Based on this forward kinematics, the conversion between the center position point $\mathbf{J}_i$ of the orthogonal joint and the joint angles can be expressed as follows:

$$\begin{pmatrix}\mathbf{J}_{i+1}^T & 1\end{pmatrix}^T = \mathbf{T}_{i-1}\begin{pmatrix}p_x & p_y & p_z & 1\end{pmatrix}^T \tag{2}$$

$$\alpha_i = \arctan(p_y/p_x)\ ,\ \beta_i = \arctan\left[(-p_z\arcsin\alpha_i)/(p_y)\right] \tag{3}$$

where $p_x, p_y, p_z$ are position components of the homogeneous transformation matrix between adjacent links.

According to the controlled winding uncoiling strategy, the guiding path is determined by the control points $\mathbf{C}_i$, and the position points $\mathbf{J}_i$ are obtained through interpolation. Specifically, as shown in Fig. 1(b), for adjacent segments, the position points $\mathbf{J}_i$ and $\mathbf{J}_{i+1}$ are interpolated from four equidistant control points $\mathbf{C}_i$, $\mathbf{C}_{i+1}$, $\mathbf{C}_{i+2}$, $\mathbf{C}_{i+3}$ using interpolation coefficients $\tau_i$ and $\tau_{i+1}$, as follows:

$$\mathbf{J}_i = \mathbf{C}_i + (\mathbf{C}_{i+1} - \mathbf{C}_i)\tau_i,\ \mathbf{J}_{i+1} = \mathbf{C}_{i+1} + (\mathbf{C}_{i+2} - \mathbf{C}_{i+1})\tau_{i+1}. \tag{4}$$

Since the intermediate motion is obtained through control point interpolation, the resulting interpolated curve satisfies the convex hull property, meaning that the curve will always stay within the polygon defined by the control points. Considering the diameter of the segment as $d_{\max}$, it forms a planned convex hull region, as shown in Fig. 1(b). This convex hull region consists of two parts: the central triangular block formed by stretching the control lines symmetrically along their normal direction by $d_{\max}$, and the edge portion formed by cylindrical sections with a diameter of $d_{\max}$. As long as the control convex hull region does not intersect with obstacles, it ensures that the snake-like manipulator does not collide with obstacles during any intermediate motion, which is crucial for accurate and collision-free path following.

## III. CONTROL-POINT-BASED MODELING

The control-point-based following model enables the representation of the intermediate motion of winding in an analytical manner, which means that obstacle avoidance and motion constraints can be pre-planned based on the control points. The calculation of interpolation coefficients and equivalent planning angles is crucial in this model.

### A. Coefficient Recursive Equation

In order to satisfy the geometric condition of equal segment lengths, there exists a relationship between $\tau_i$ and $\tau_{i+1}$. To simplify the expression of this relationship, we introduce a control angle $\theta_i$ ($\frac{\pi}{2} \le \theta_i \le \frac{3\pi}{2}$), such that $\tau_{i+1}$ can be recursively generated from $\tau_i$ and the angle $\theta_i$.

As shown in Fig. 1(b), there are two planes composed of $C_i, C_{i+1}, C_{i+2}$ and $C_{i+1}, C_{i+2}, C_{i+3}$ respectively. Therefore, control angle $\theta_i$ is defined as follows:

$$\theta_i = \begin{cases} \arccos \dfrac{\overrightarrow{C_{i+1}C_i}\cdot\overrightarrow{C_{i+1}C_{i+2}}}{\left\|\overrightarrow{C_{i+1}C_i}\right\|\left\|\overrightarrow{C_{i+1}C_{i+2}}\right\|} & \overrightarrow{n_i}\cdot\overrightarrow{n_{i+1}} < 0 \\ 2\pi - \arccos \dfrac{\overrightarrow{C_{i+1}C_i}\cdot\overrightarrow{C_{i+1}C_{i+2}}}{\left\|\overrightarrow{C_{i+1}C_i}\right\|\left\|\overrightarrow{C_{i+1}C_{i+2}}\right\|} & otherwise, \end{cases} \tag{5}$$

where $\overrightarrow{n_i}$ is the normal vector of the plane formed by $C_i, C_{i+1}, C_{i+2}$, and $\overrightarrow{n_{i+1}}$ is the normal vector of the plane formed by $C_{i+1}, C_{i+2}, C_{i+3}$.

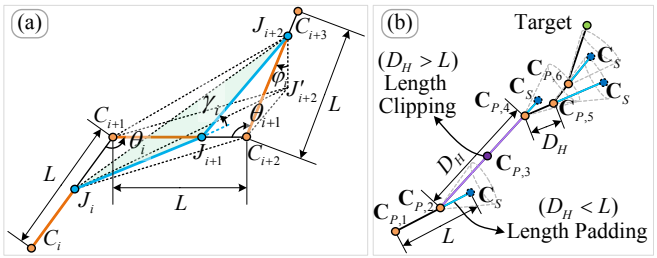


Fig. 3. (a) Geometric relationships in the calculation of joint planning angles. (b) Handling non-uniform control points: length padding and clipping.

Based on the cosine rule, geometric relationships of segment lengths in triangle $J_iC_{i+1}J_{i+1}$ as

$$L^2 = [(1-\tau_i)L]^2 + (\tau_{i+1}L)^2 - 2(1-\tau_i)\tau_{i+1}L^2\cos\theta_i \\ 0 \le \tau_i \le 1,\ 0 \le \tau_{i+1} \le 1,\ \pi/2 \le \theta_i \le 3\pi/2. \tag{6}$$

Then, the coefficient recursive equation between $\tau_i$ and $\tau_{i+1}$ can be introduced as the solution to equation (6):

$$\tau_{i+1} = (1-\tau_i)\cos\theta_i + \sqrt{2\tau_i - \tau_i^2 + [(1-\tau_i)\cos\theta_i]^2}. \tag{7}$$

Compute the partial derivative of $\tau_{i+1}$ with respect to $\tau_i$:

$$\frac{\partial\tau_{i+1}}{\partial\tau_i} = -\cos\theta_i - \frac{(1-\tau_i)(\cos^2\theta_i - 1)}{\sqrt{2\tau_i - \tau_i^2 + [(1-\tau_i)\cos\theta_i]^2}}. \tag{8}$$

For the interval $\pi/2 \le \theta_i \le 3\pi/2$, there exists only one valid zero point $\tau_{i_0}$ for $\frac{\partial\tau_{i+1}}{\partial\tau_i}$, and it holds that $\tau_{i_0} \ge 1$ as

$$\tau_{i_0} = 1 - \cos\theta_i/|\sin\theta_i|. \tag{9}$$

Therefore, it can be established that within the constraint range, all subsequent points can be recursively determined from the initial point, satisfying interpolation requirements.

### B. Equivalent Joint Planning Angle

Orthogonal joint angles must adhere to structural constraints in following, but mapping them from position points is complex, particularly with increased joint count. To explicitly manage angle constraints using only control parameters, the joint planning angle $\gamma_i$ is introduced as

$$\gamma_i = \arccos\left[(\overrightarrow{J_iJ_{i+1}}\cdot\overrightarrow{J_{i+1}J_{i+2}})/(\left\|\overrightarrow{J_iJ_{i+1}}\right\|\left\|\overrightarrow{J_{i+1}J_{i+2}}\right\|)\right]. \tag{10}$$

The relationship between the joint planning angle and the orthogonal joint angle can be expressed as

$$\cos\alpha_i\cos\beta_i = \cos\gamma_i. \tag{11}$$

Since $0 \le \cos\beta_i \le 1$, it follows that $\cos\alpha_i > \frac{\cos\gamma_i}{\cos\beta_i}$, which implies $|\alpha_i| \le |\gamma_i|$. Similarly, for $\beta_i$, we have $\max(|\alpha_i|,|\beta_i|) \le |\gamma_i|$. Therefore, the equivalent joint planning angle can represent the boundary of the orthogonal joint angle.

To simplify the expression of joint planning angles, as shown in Fig. 3(a), control angles $\theta_i(\frac{\pi}{2} \le \theta_i \le \frac{3\pi}{2})$, $\theta_{i+1}(\frac{\pi}{2} \le \theta_i \le \frac{3\pi}{2})$, and auxiliary angle $\varphi_i(0 \le \theta_i \le \pi)$ are introduced to generate $\gamma_i$ solely based on $\tau_i, \theta_i, \theta_{i+1}$, and $\varphi_i$:

$$\gamma_i = f(\tau_i)_{|\theta_i,\theta_{i+1},\varphi_i}. \tag{12}$$

The auxiliary angle $\varphi_i$ is the angle between the normal vector $\overrightarrow{J'_{i+2}J_{i+2}}$ of the plane formed by control points $C_i, C_{i+1}, C_{i+2}$ and the control line $C_{i+2}C_{i+3}$. Its relationship with the control points is as follows:

$$\varphi_i = \arccos\frac{\overrightarrow{C_{i+2}C_{i+3}}\cdot(\overrightarrow{C_iC_{i+1}}\times\overrightarrow{C_{i+1}C_{i+2}})}{\left\|\overrightarrow{C_{i+2}C_{i+3}}\right\|\left\|\overrightarrow{C_iC_{i+1}}\times\overrightarrow{C_{i+1}C_{i+2}}\right\|}. \tag{13}$$

Based on the geometric relationship formed by the spatial triangle $J_iJ_{i+1}J_{i+2}$, we can obtain:

$$\gamma_i = \pi - \arccos\left[(2L^2 - \overline{J_iJ_{i+2}}^2)/(2L^2)\right]. \tag{14}$$

Therefore, the joint planning angle is directly influenced by the distance $\overline{J_iJ_{i+2}}$, which can be determined based on the geometric relationship formed by the triangle $J_iJ'_{i+2}J_{i+2}$ as

$$\overline{J_iJ_{i+2}}^2 = \overline{J_iJ'_{i+2}}^2 + (\tau_{i+2}L\cos\varphi_i)^2 \tag{15}$$

where $\tau_{i+2}$ can be recursively obtained from $\tau_i$ using (7).

The length of $J_iJ'_{i+2}$ can be determined based on the geometric relationship formed by the triangle $J_iC_{i+2}J'_{i+2}$:

$$\overline{J_iJ'_{i+2}}^2 = \overline{J_iC_{i+2}}^2 + (\tau_{i+2}L\sin\varphi_i)^2 - 2\overline{J_iC_{i+2}}\tau_{i+2}L\sin\varphi_i\cos\psi \\ \psi = \begin{cases} \psi_1 + \psi_2 & \theta_{i+2} < \pi \\ \psi_1 - \psi_2 & otherwise, \end{cases} \tag{16}$$

where the auxiliary projection angle $\psi$ and the relationship between $\theta_{i+2}$ and $\pi$ determine the orientation of the projection line $C_{i+2}J'_{i+2}$ with respect to the control line $C_{i+1}C_{i+2}$, resulting in two different composite components $\psi_1$ and $\psi_2$.

The distance $\overline{J_iC_{i+2}}$ and the angular component $\psi_2$ can be determined based on the geometric relationship formed by the spatial triangle $J_iC_{i+1}C_{i+2}$:

$$\overline{J_iC_{i+2}}^2 = (1-\tau_i)^2L^2 + L^2 - 2(1-\tau_i)L^2\cos\theta_i \tag{17}$$

$$\psi_2 = \arccos\frac{L^2 + \overline{J_iC_{i+2}}^2 - (1-\tau_i)^2L^2}{2L\overline{J_iC_{i+2}}}. \tag{18}$$

The angular component $\psi_1$ can be determined based on the geometric relationship formed by the spatial triangles $C_{i+1}C_{i+2}J'_{i+2}$ and $C_{i+1}J'_{i+2}J_{i+2}$:

$$\psi_1 = \arccos\frac{L^2 + (\tau_{i+2}L)^2 - \overline{C_{i+1}J_{i+2}}^2}{2\tau_{i+2}L^2\sin\varphi_i} \tag{19}$$

where the distance $\overline{C_{i+1}J_{i+2}}$ can be obtained based on the geometric relationship formed by the triangle $C_{i+1}C_{i+2}J_{i+2}$:

$$\overline{C_{i+1}J_{i+2}}^2 = L^2 + (\tau_{i+2}L)^2 - 2\tau_{i+2}L^2\cos\theta_{i+2}. \tag{20}$$

The joint planning angle $\gamma_i$ corresponding to the specified interpolation coefficient $\tau_i$ under the constraint of control angles can be solved by simultaneously solving equations (14) to (20). If $\tau_i$ is a function of time $t$, the analytical expressions for the joint planning angle described in (12) can be used to further obtain the analytical expressions for velocity $\dot\gamma_i$ and acceleration $\ddot\gamma_i$ as

$$\dot\gamma_i = \frac{\partial f(\tau_i)_{|\theta_i,\theta_{i+1},\varphi_i}}{\partial\tau_i}\frac{\partial\tau_i}{\partial t},\ \ddot\gamma_i = \frac{\partial^2 f(\tau_i)_{|\theta_i,\theta_{i+1},\varphi_i}}{\partial\tau_i^2}\frac{\partial^2\tau_i}{\partial t^2}. \tag{21}$$

The planning angle model reduces redundant variables via geometric constraints, replacing the 12 coordinate parameters of 4 control points with 3 control parameters, simplifying the analytical expression and improving planning efficiency.

## IV. CONSTRAINED PATH FOLLOWING

The winding uncoiling motion of the snake-like manipulator, following the planned path of its head, is subject to constraints imposed by control points and motion capabilities (velocity and acceleration). To achieve the mentioned constrained following, the sliding control points (SCP) algorithm for handling control point constraints and the motion smoothing restriction algorithm for addressing motion capability

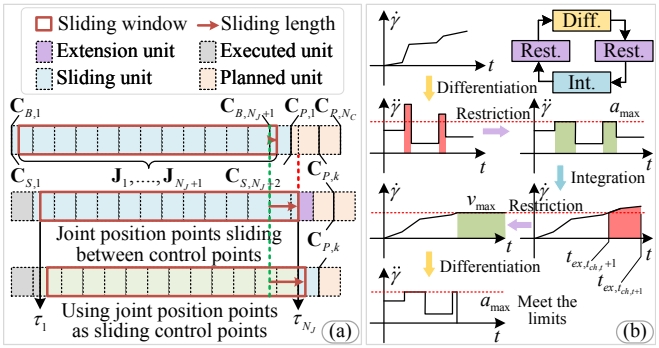


Fig. 4. Constrained path planning algorithms include sliding control points (a) and motion smoothing restriction (b).

**Algorithm 1:** R-SCP Based Path Following

**Input:** Head sliding sequence $\{L_{H,t}\}$, Limitations $\{\mathscr{C}_d\} = \{v_{\max}, a_{\max}, \{\mathbf{C}_i\}\}$, Control period $T$
**Output:** Constrained joint position sequence $\{\mathbf{J}'_{i,t}\}$

$\{\gamma_t\} \leftarrow \text{CalcPlanAngle}(\{\text{L}_{\text{H,t}}\})$ (Eq.(12))
**for** $k = 1 \rightarrow k_{\max}$ **do**
  **for** $d = 1 \rightarrow 2$ **do**
    $\{\gamma_t\} \leftarrow \text{DiffPlanAngle}(\{\gamma_t\}, \{L_{H,t}\}, T)$ (Eq.(21))
    $\{\gamma_t\} \leftarrow \text{MotionRestriction}(\{\gamma_t\}, \mathscr{C}_d, T)$
  **end**
  $flag \leftarrow true$
  **for** $d = 1 \rightarrow 2$ **do**
    **if** $\text{CheckLimits}(\{\gamma_t\}) = false$ **then**
      $flag \leftarrow false$
      $\{\gamma_t\} \leftarrow \text{MotionRestriction}(\{\gamma_t\}, \mathscr{C}_d, T)$
    **end**
    $\{\gamma_t\} \leftarrow \text{IntPlanAngle}(\{\gamma_t\}, \{L_{H,t}\}, T)$ (Eq.(21))
  **end**
  **if** $flag = true$ **then**
    **Break**
  **end**
**end**
$\{L'_{H,t}\} \leftarrow \text{RevCalcPlanAngle}(\{\gamma_t\})$ (Eq.(12))
$\{\mathbf{J}'_{i,t}\} \leftarrow \text{SlidingControlPoints}(\{\mathbf{C}_{\text{B,i}}\}, \{\mathbf{C}_{\text{P,i}}\}, \{\text{L}'_{\text{H,t}}\})$
**Return** $\{\mathbf{J}'_{i,t}\}$

limitations are integrated into the Restricted SCP (R-SCP) path following, converting the head sliding sequence into a joint position sequence adhering to trajectory execution constraints, as depicted in Alg. 1.

### A. Sliding Control Points Algorithm

The SCP algorithm shown in Alg. 2 generates the joint center position points corresponding to the head sliding length based on the base and path control points. Referring to [27], in the FTL motion, segments are constrained partly to the feeding's motion path by the base control points and partly to the planned path by the path control points. The SCP gradually transitions the segments from the base control point to the path control points, promoting their sliding on the planned path, known as winding uncoiling.

The $N_{J+1}$ position points corresponding to $N_J$ joints are obtained through pairwise interpolation of $N_{J+2}$ control points (Fig. 4(a)). The control points involved in interpolation form a sliding window, consisting of $N_{J+1}$ sliding units with the same length $L$ as adjacent joints. The adjacent path control points form planning units. The distance $D_H$ between the control points in the window head is calculated, and the cumulative sum $x_H$ gives the sliding length of the window. Sliding units must be of equal length, so if $D_H < L$, an extension unit is added to pad length, if $D_H > L$, a control

**Algorithm 2:** Sliding Control Points

**Input:** Base control points $\{\mathbf{C}_{B,1}, ..., \mathbf{C}_{B,N_J+1}\}$, Path control points $\{\mathbf{C}_{P,1}, ..., \mathbf{C}_{P,N_C}\}$, Head sliding length $L_H$
**Output:** Joint position points $\{\mathbf{J}_1, ..., \mathbf{J}_{N_J+1}\}$

Initialize $x_H \leftarrow 0$, $\{\mathbf{C}_{S,i}\} \leftarrow \{\mathbf{C}_{B,1}, ..., \mathbf{C}_{B,N_J+1}, \mathbf{C}_{P,1}\}$
**for** $k = 2 \rightarrow N_C$ **do**
  $\tau_{N_J} \leftarrow ||\mathbf{C}_{S,N_J+2} - \mathbf{C}_{S,N_J+1}||/L$, $x_H \leftarrow x_H + \tau_{N_J} L$
  $\mathbf{C}_{S,N_J+2} \leftarrow \mathbf{C}_{S,N_J+1} + \frac{1}{\tau_{N_J}}(\mathbf{C}_{S,N_J+2} - \mathbf{C}_{S,N_J+1})$
  $\{\theta_i\} \leftarrow \text{CalcCtrlAngle}(\{\mathbf{C}_{S,i}\})$ (Eq. (5))
  **if** $x_H \geq L_H$ **then**
    $\tau_{N_J} \leftarrow \tau_{N_J} - (x_H - L_H)/L$
  **end**
  **for** $j = N_J + 1 \rightarrow 1$ **do**
    $\mathbf{J}_j \leftarrow \mathbf{C}_{S,j} + \tau_{N_J}(\mathbf{C}_{S,j+1} - \mathbf{C}_{S,j})$ (Eq. (4))
    $\tau_{N_J} \leftarrow 1 - \text{CalcTau}(1 - \tau_{\text{N}_\text{J}}, \{\theta_{\text{i}}\})$ (Eq. (7))
  **end**
  **if** $x_H \geq L_H$ **then**
    **Return** $\{\mathbf{J}_1, ..., \mathbf{J}_{N_J+1}\}$
  **end**
  $\{\mathbf{C}_{S,i}\} \leftarrow \{\mathbf{J}_1, ..., \mathbf{J}_{N_J+1}, \mathbf{C}_{P,k}\}$
**end**

**Algorithm 3:** Motion Smoothing Restriction

**Input:** Planning angle sequence $\{\gamma_t\}$, Limitation $\mathscr{C}_d$, Control period $T$
**Output:** Constrained Planning angle sequence $\{\gamma'_t\}$

$\{t_{ex,i}\} \leftarrow \text{find}(|\{\gamma_t\}| > \mathscr{C}_{\text{d}})$
$\{t_{ch,i}\} \leftarrow \{0, \text{find}(\text{diff}(\{t_{ex,i}\}) \neq 1), \text{length}(\{t_{ex,i}\})\}$
$\{\gamma'_t\} \leftarrow \{\}$
**for** $t = 1 \rightarrow \text{length}(\{\text{t}_{\text{ch,i}}\}) - 1$ **do**
  $\{t_{r,i}\} \leftarrow \{t_{ex,i}\}[t_{ch,t} + 1 : t_{ch,t+1}]$, $\mathscr{C}'_d \leftarrow \mathscr{C}_d$
  **for** $k = 1 \rightarrow k_{\max}$ **do**
    $S_t \leftarrow \text{trapz}(\gamma[\{t_{r,i}\}])$, $N_t \leftarrow \text{round}(S_t/\mathscr{C}'_d/T)$
    $\mathscr{C}''_d \leftarrow \min(S_t/N_t/T, \mathscr{C}_d)$
    **if** $|\mathscr{C}''_d - \mathscr{C}'_d| < 0.01$ **then**
      **Break**
    **end**
    $\{t_{r,i}\} \leftarrow \text{find}(|\{\gamma_t\}| > \mathscr{C}''_d))$, $\mathscr{C}'_d \leftarrow \mathscr{C}''_d$
  **end**
  $\{\gamma'_t\} \leftarrow \{\{\gamma'_t\}, \{\gamma_t\}[t_{ex,t_{ch,t-1}} : t_{ex,t_{ch,t}+1}],$
    $\mathscr{C}''_d * \text{ones}(1, N_t)\}$
**end**
**Return** $\{\gamma'_t\}$

point is added to clip length (Fig. 3(b)). The head sliding factor $\tau_{N_J}$ is the maximum sliding ratio of the head sliding unit in the window. According to the recursive equation in Section III-A, the coefficients $\tau_j$ are calculated for each sliding unit from the head to the tail, and new position points are interpolated. The new $N_{J+1}$ position points, together with the next path control point, form a new sliding window, and this process continues until $x_H \geq L_H$.

### B. Motion Smoothing Restriction Algorithm

The time-scaling approach is a simple, effective, and widely used technique [28]. The motion smoothing restriction is modified based on this [29], including the forward and backward smoothing processes of the joint planning angles in Section III-B. As shown in Fig. 4(b), during the forward process, the planning angles are first differentiated and then subjected to smoothing constraints. On the other hand, during the backward process, the planning angles undergo limit checking and smoothing constraints before integration. If all planning angles remain within the limits, the motion capability constraint is satisfied.

As shown in Alg. 3, each smoothing process includes searching for over-limit intervals, distributing excess values,

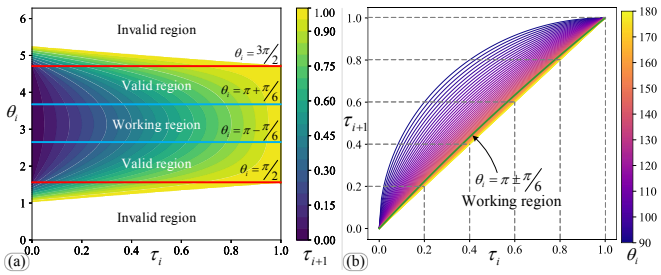


Fig. 5. Recursive calculation model characteristics: (a) Trajectory completeness; (b) Monotonicity and approximate linearity of position points.

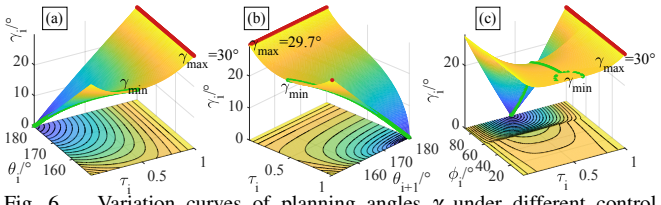


Fig. 6. Variation curves of planning angles $\gamma$ under different control parameters $\theta_i, \theta_{i+1}, \varphi_i$ and $\tau_i$: (a) $\tau_i - \theta_i - \gamma_i$ $(\theta_{i+1} = \frac{5\pi}{6}, \varphi_i = \frac{\pi}{4})$; (b) $\tau_i - \theta_{i+1} - \gamma_i$ $(\theta_i = \frac{5\pi}{6}, \varphi_i = \frac{\pi}{4})$; (c) $\tau_i - \varphi_i - \gamma_i$ $(\theta_i = \frac{5\pi}{6}, \theta_{i+1} = \frac{5\pi}{6})$.

and concatenating interval sequences. We first find the index $t_{ex}$ of the control intervals that exceed the limits, and then use the non-consecutive points $t_{ch}$ of the interval indices as boundaries for the over-limit intervals. Each over-limit interval is integrated, and the portion exceeding the limit value is truncated to create multiple control intervals while maintaining the limit value. Finally, the resulting interval sequence is formed by concatenating intervals that do not exceed the limit value with intervals that maintain the limit value, in sequential order.

## V. EXPERIMENTS AND RESULTS

### A. Analysis of Control Point-Based Following Model

The coefficient $\tau_{i+1}$ corresponding to the $\tau_i$ at different control angles $\theta_i$ are computed to reflect the characteristics of the model. According to the definition, the valid region for the control angle is $\frac{\pi}{2} \le \theta_i \le \frac{3\pi}{2}$, while due to mechanical limitations, the working region is reduced to $\pi \pm \frac{\pi}{6}$. As shown in Fig. 5(a), within the valid region, any $t_i$ corresponds to a unique $t_{i+1}$ in the valid interval $[0,1]$. This indicates that the next position point, determined recursively from the initial point, must lie on the line formed by adjacent control points. Therefore, the adjacent control points can determine all the position points during following, demonstrating trajectory completeness. As shown in Fig. 5(b), the relationship between $\tau_i$ and $\tau_{i+1}$ is monotonic but nonlinear. The nonlinear nature is significantly influenced by the $\theta_i$. However, within the working region, the nonlinear error is less than 0.035, indicating that the rate of movement of the next point is nearly consistent with that of the initial point. Fig. 6 shows the non-monotonic variation of planning angle $\gamma_i$ with coefficient $\tau_i$, influenced by different control parameters. The effect becomes more pronounced with increasing $|\theta_i - \pi|, |\varphi|$, leading to extreme points in the planning angle during following. Ensuring the planning angle at these extreme points stays within the limit value ($\gamma_{\max} \le \pi/6$) guarantees no position exceeds physical limits.

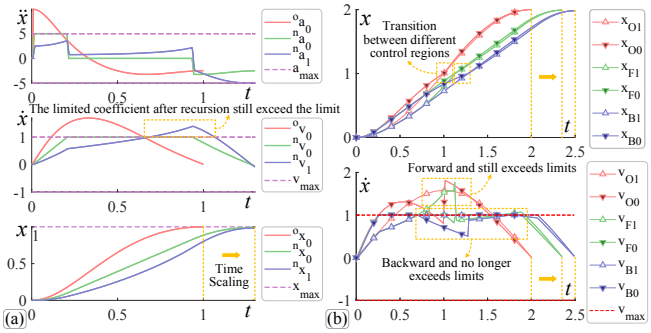


Fig. 7. Motion smoothness validation: (a) Forward velocity and acceleration restrictions applied to initial coefficient $\tau_0$; (b) Forward and backward velocity restrictions applied to $\tau_0$ and its recursive coefficient $\tau_1$.

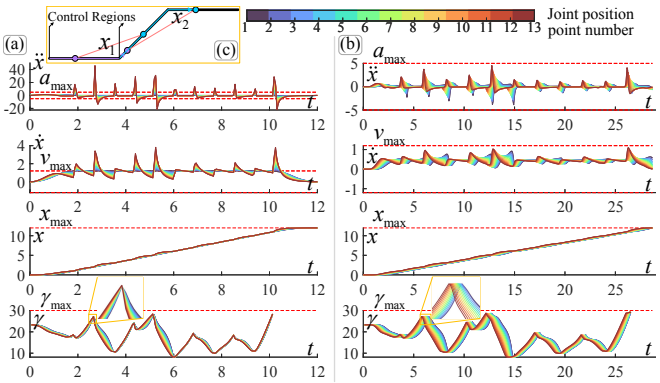


Fig. 8. Restriction effects on displacement $x$, velocity $\dot{x}$, acceleration $\ddot{x}$, and planning angles $\gamma$ for multiple control points: (a) Before; (b) After. Lines of different colors represent different joint position point numbers.

### B. Analysis and Verification of Motion Restrictions

In this subsection, the cumulative coefficient of the sliding point within control regions formed by adjacent control points is defined as displacement $x$ (Fig. 8(c)). One time unit represents sliding through one control region. Initially, the following velocity and acceleration of a initial coefficient $\tau_0$ are constrained (Fig. 7(a)). The results show that motion smoothness restriction scales time while keeping displacement constant. However, with only forward process from $\tau_0$ to $\tau_1$, although the ${}^n v_0$ and ${}^n a_0$ of $\tau_0$ do not exceed the limit, those of its recursive coefficient $\tau_1$ do exceed the limit. Due to coupling between coefficients, the backward process is required from $\tau_1$ to $\tau_0$, as shown in Fig. 7(b). Sliding points generate motion shocks when crossing different control regions. The forward process still results in the velocity $V_{F1}$ of $\tau_1$ exceeding the limit, while after the backward process, both $V_{B0}$ and $V_{B1}$ meet the requirements, making the transition between different control regions smoother.

In further experiments, the motion of 13 sliding points of the 12-segment manipulator is constrained as shown in Fig. 8. Despite the initial point being smoothed by a double S-curve, it still has $a_{\max}$ of 46.2 and $v_{\max}$ of 3.8, while with the proposed method, $a_{\max}$ reduces to 4.6, and $v_{\max}$ to 1.1, showing improvement. Testing 1000 randomly configured control points, $a_{\max}$ is reduced by an average of 92.8%, and $v_{\max}$ by 71.6%, both staying within limits. The planning angles also do not exceed the physical limit $\gamma_{\max}$, and the peak phenomenon is significantly improved.

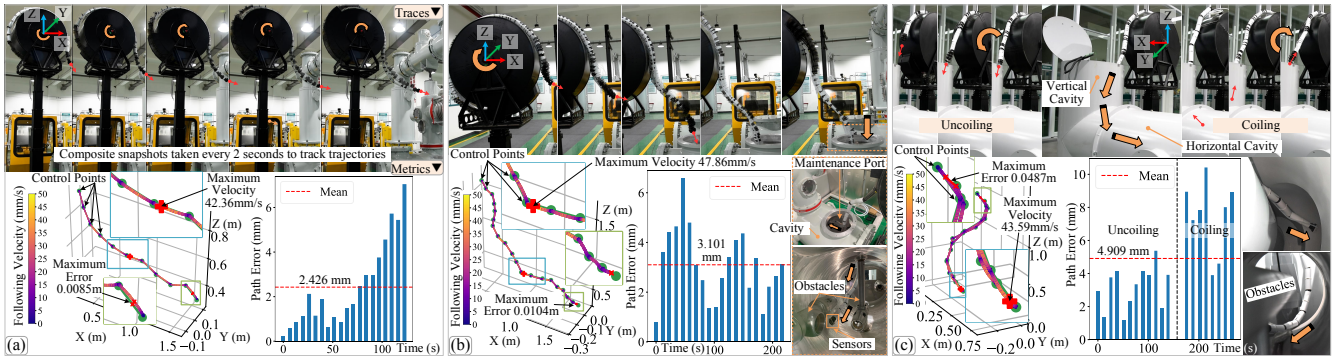


Fig. 9. Tested method's effects in diverse application scenes: (a) External maintenance; (b) Single cavity; (c) Combined cavity. Each set includes traces and metrics. Traces composited from timed motion snapshots. Actual position and velocity deduced from cable length; error from planned control path.

TABLE I

COMPARISON WITH TYPICAL FOLLOWING METHODS

| Method | | Time (ms)↓ | Path Error (mm)↓ | Tip Error (mm)↓ | Velocity (mm/u)[1]↓ | Angle (°)↓ | Collision (%)↓ |
|---|---|---|---|---|---|---|---|
| GS | Avg. | 14.5 | **7.3** | 14 | 26.8 | 32.9 | 42.3 |
| [23], [20] | Max. | 26.7 | 83 | 80.2 | 82.7 | 71.4 | 95.2 |
| SEG | Avg. | 0.567 | 7.98 | **0.99** | 26.2 | 31.3 | **0.0** |
| [21], [22] | Max. | 1.23 | 80.5 | **1.0** | 179 | 68.2 | **0.0** |
| SCP | Avg. | **0.095** | 7.39 | **0.99** | 26.2 | 25.5 | **0.0** |
| | Max. | **1.0** | **58.1** | **1.0** | 135 | 56.1 | **0.0** |
| R-SCP | Avg. | 0.232 | 7.51 | 1.0 | **1.32** | **25.5** | **0.0** |
| | Max. | 1.03 | 60.3 | **1.0** | **25.3** | **55.7** | **0.0** |

[1] mm/u represents the millimeters of movement per control cycle.

### C. Performance Comparison of Different Methods

The introduced R-SCP method is engineered [1] and compared with algorithmic reproductions of representative methods introduced in the introduction: Gradient Search (GS) and Geometric Segmentation (SEG). Quantitative metrics comprise computation time, path error (PE), tip error (TE), velocity, planning angle, and out-of-control-region obstacle collision rate. Referring to [24], $PE = \frac{\sum_{c=1}^{n} |\mathbf{p}_C(c) - \mathbf{p}_J(c)|}{n}$, $TE = |\mathbf{p}_C(L_H) - \mathbf{p}_J(L_H)|$, with $\mathbf{p}_C(c)$ as discrete control path element and $\mathbf{p}_J(c)$ as linkage element.

Based on Table I, R-SCP exhibits notable speed improvement with marginal PE variation, particularly SCP without motion smoothing, allowing finer follower position computation within the same control cycle. Compared to search methods such as GS, geometric methods with strict control line alignment such as SEG and R-SCP, exhibit low TE and avoid obstacles outside the control region. Because of motion smoothing, R-SCP adheres to max velocity (50mm/u) and angle limits (60°), preventing unrealistic follower outcomes.

### D. Field Experiment and Application Verification

As shown in Fig. 9, the proposed method R-SCP is deployed on the prototype for varied inspection tasks, validating its smooth path-following ability. The head collision-free path control points are generated by the SFTL algorithm in [27] based on the sensor-perceived obstacle distribution. Then the R-SCP computes joint positions based on control points at each time, and solved by inverse kinematics for angles and cable lengths. At the same time, the actual cable lengths can be used to obtain the head position and velocity through forward kinematics. The PE based on Section V-C is group averaged to avoid dense bars.

In the experiment, the prototype is raised to an appropriate height, then follows a stepwise planned control path, enters the cavity through a maintenance port, navigates around obstacles, and finally coils back along the path. From interval-captured trajectory snapshots, each moment's segment strictly resides within the safety region determined by control points, demonstrating R-SCP's collision-free control, even with only head-mounted sensors. In terms of smoothness, the maximum velocity across scenes stays within the 50mm/s limit. Notably, the transitions between different control points may lead to impacts. Consequently, our method mitigates velocity during transitions, then restores and approximates the upper limit (47.86mm/s) to minimize motion time. Regarding errors, minimal average (2.426mm) in the unobstructed scene (Fig. 9(a)), and a higher average (4.909mm) in the complex obstacle scene (Fig. 9(c)). Additionally, errors increase in later segments of long paths (Fig. 9(a)) and along abrupt trajectories (Fig. 9(b)). Due to potential cable slack during motion, uncoiling exhibits smaller errors than coiling (Fig. 9(c)). Overall, the proposed method maintains motion smoothness, actual error depends on planned paths, yet retains an average error below 10mm.

[1]Project Code: https://github.com/HILMR/C-CDHRM

## VI. CONCLUSIONS AND FUTURE WORK

This paper proposes efficient constrained path following for snake-like manipulators. Snakes' winding simplifies via sequential path control points. Motivated by this, established a control-point-based following model, including a coefficient recursive equation and equivalent planning angles. Using the model, sliding control point and motion smoothing restriction algorithms are designed. The former enforces collision-free following between base and path control for remaining segments. The latter adjusts control parameters based on constraints for smoothness. The experiments initially validate the model's characteristics and prove motion smoothing can achieve 71.6% velocity reduction and 92.8% acceleration reduction. Later, the proposed method, with the same settings and metrics, takes 41% of SEG time, 1.6% of GS time, and 14% of SEG velocity, 30% of GS velocity, avoiding angle and collision overruns. In diverse real-world scenes, the prototype stays within velocity limits, averaging 3.48mm follow error. The current method is limited by uniform segment lengths, so in future work, we will develop more versatile configuration-adaptive following algorithms, intending to open-source them within our project.

## REFERENCES


[1] Z. Mu, L. Zhang, L. Yan, Z. Li, R. Dong, C. Wang, and N. Ding, "Hyper-redundant manipulators for operations in confined space: Typical applications, key technologies, and grand challenges," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 6, pp. 4928–4937, Dec. 2022.

[2] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE Trans. Robot.*, vol. 31, no. 6, pp. 1261–1280.

[3] J. K. Hopkins, B. W. Spranklin, and S. K. Gupta, "A survey of snake-inspired robot designs," *Bioinspiration Biomimetics*, vol. 4, no. 2, p. 021001, Jun. 2009.

[4] P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, *Snake robots: modelling, mechatronics, and control*. Springer, 2013.

[5] J. Liu, Y. Tong, and J. Liu, "Review of snake robots in constrained environments," *Robot. Auton. Syst.*, vol. 141, p. 103785.

[6] R. J. Webster and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *Int. J. Rob. Res.*, vol. 29, no. 13, pp. 1661–1683, Nov. 2010.

[7] W. Xu, T. Liu, and Y. Li, "Kinematics, dynamics, and control of a cable-driven hyper-redundant manipulator," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 4, pp. 1693–1704, 2018.

[8] D. Xu, E. Li, Z. Liang, and Z. Gao, "Design and tension modeling of a novel cable-driven rigid snake-like manipulator," *J Intell Robot Syst*, vol. 99, no. 2, pp. 211–228, 2020.

[9] Y. Wang, Q. Yan, T. Li, G. Tuo, X. Li, and H. Liu, "Efficient inverse kinematics optimization solution method of smooth configuration for hyper-redundant robot," in *2021 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2021, pp. 625–630.

[10] T. Liu, T. Yang, W. Xu, G. Mylonas, and B. Liang, "Efficient inverse kinematics and planning of a hybrid active and passive cable-driven segmented manipulator," *IEEE Trans. Syst., Man, Cybern., Syst.*, 2021.

[11] M. Luo, E. Li, A. Zhang, M. Tan, and Z. Liang, "A bioinspired coiled cable-driven manipulator: Mechatronic design and kinematics planning with multiconstraints," *IEEE/ASME Trans. Mechatronics*, vol. 28, no. 6, pp. 3155–3166, Dec. 2023.

[12] A. Mohammad, M. Russo, Y. Fang, X. Dong *et al.*, "An efficient follow-the-leader strategy for continuum robot navigation and coiling," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7493–7500, 2021.

[13] E. S. Conkur, "Path following algorithm for highly redundant manipulators," *Rob. Auton. Syst.*, vol. 45, no. 1, pp. 1–22, Oct. 2003.

[14] E. Amanov, J. Granna, and J. Burgner-Kahrs, "Toward improving path following motion: Hybrid continuum robot design," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4666–4672.

[15] L. Tang, L.-M. Zhu, X. Zhu, and G. Gu, "A serpentine curve based motion planning method for cable-driven snake robots," in *2018 25th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*. IEEE, 2018, pp. 1–6.

[16] H. Choset and W. Henning, "A follow-the-leader approach to serpentine robot motion planning," *J. Aerosp. Eng.*, vol. 12, no. 2, pp. 65–73, Apr. 1999.

[17] C. Girerd, A. V. Kudryavtsev, P. Rougeot, P. Renaud, K. Rabenorosoa, and B. Tamadazte, "Slam-based follow-the-leader deployment of concentric tube robots," vol. 5, no. 2, pp. 548–555.

[18] B. H. Meng, I. S. Godage, and I. Kanj, "Smooth path planning for continuum arms," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7809–7814.

[19] H. Ji, H. Xie, and H. Yang, "A spatial path following method for hyper-redundant manipulators by step-by-step search and calculating," in *2022 7th International Conference on Robotics and Automation Engineering (ICRAE)*, Nov. 2022, pp. 292–298.

[20] L. Tang, L. Zhu *et al.*, "Confined spaces path following for cable-driven snake robots with prediction lookup and interpolation algorithms," *Sci. China-Technol. Sci.*, vol. 63, no. 2, pp. 255–264, 2020.

[21] H. Xie, C. Wang, S. Li, L. Hu, and H. Yang, "A geometric approach for follow-the-leader motion of serpentine manipulator," *Int. J. Adv. Rob. Syst.*, vol. 16, no. 5, p. 1729881419874638, 2019.

[22] M. Luo, Y. Tian, E. Li, M. Chen, C. Kang, G. Yang, and M. Tan, "A novel coiled cable-conduit-driven hyper-redundant manipulator for remote operating in narrow spaces," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 2426–2432.

[23] D. Palmer, S. Cobos-Guzman, and D. Axinte, "Real-time method for tip following navigation of continuum snake arm robots," *Rob. Auton. Syst.*, vol. 62, no. 10, pp. 1478–1485, Oct. 2014.

[24] M. Neumann and J. Burgner-Kahrs, "Considerations for follow-the-leader motion of extensible tendon-driven continuum robots," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 917–923.

[25] Y. Bulut and E. S. Conkur, "A real-time path-planning algorithm with extremely tight maneuvering capabilities for hyper-redundant manipulators," *Engineering Science and Technology, an International Journal*, vol. 24, no. 1, pp. 247–258, Feb. 2021.

[26] C. Wang, H. Xie, and H. Yang, "An iterative path-following method for hyper-redundant snake-like manipulator with joint limits," *Industrial Robot: the international journal of robotics research and application*, vol. 50, no. 3, pp. 505–519, Jan. 2023.

[27] M. Luo, Y. Tian, E. Li, M. Chen, and M. Tan, "A local obstacle avoidance and global planning method for the follow-the-leader motion of coiled hyper-redundant manipulators," *IEEE Trans. Ind. Informat.*, pp. 1–12, 2024.

[28] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *IEEE Trans. Autom. Control*, vol. 54, no. 10, pp. 2318–2327, 2009.

[29] H. Pham and Q.-C. Pham, "A new approach to time-optimal path parameterization based on reachability analysis," *IEEE Trans. Robot.*, vol. 34, no. 3, pp. 645–659, 2018.