Robotica (2026), **44**, pp. 1551–1572
doi:10.1017/S0263574726103701

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Trajectory planning for cable-driven continuum robots using model predictive control and basis spline optimization

Qi Chen, Shishuai Zhao and Yanan Qin

University of Shanghai for Science and Technology, China
Corresponding author: Qi Chen; E-mail: chenqi8@usst.edu.cn

Received: 17 March 2026 UTC; **Revised:** 27 April 2026 UTC; **Accepted:** 1 June 2026 UTC;
First published online: 8 July 2026 UTC

Keywords: model predictive control; artificial potential field; cable-driven continuum robot; whale optimization algorithm; two-layer B-spline optimization framework

Abstract

To guarantee the safety and stability of cable-driven continuum robots (CDCRs) in multi-obstacle scenarios, this paper proposes a novel trajectory planning and tracking framework that integrates an artificial potential field (APF) based model predictive control (MPC) with a two-layer B-spline trajectory optimization framework. APF is incorporated as a penalty term in the objective function to enhance trajectory safety. Additionally, an improved whale optimization algorithm (IWOA) is introduced to circumvent the local optimal trap of MPC. Particularly, to improve tracking accuracy and stability, a two-layer B-spline optimization framework is proposed to enhance the smoothness of the planned trajectory. Simulation and experimental results show that the designed trajectory planning and tracking algorithm can generate collision-free trajectories and achieve high tracking accuracy in multi-obstacle environments.

1. Introduction

The cable-driven continuum robot (CDCR) with multi-degree-of-freedom is a flexible manipulator inspired by biological structures, such as elephant trunks and octopus tentacles [1–3]. CDCRs can be adjusted by changing the length of driving cables, making it particularly suitable for complex tasks. Recently, with the rapid development of robotic technology, CDCRs have been increasingly utilized in various fields such as laparoscopic surgery [4], space station maintenance [5], and rehabilitation [6]. However, the application of CDCRs in these complex environments requires high trajectory planning and tracking capabilities to ensure safety and accuracy.

Trajectory planning is crucial for achieving safe and stable obstacle avoidance [7]. It is necessary to achieve real-time recognition and avoidance of obstacles in complex 3D environments and reach designated locations [8]. Trajectory planning algorithms mainly include the following types: the Dijkstra algorithm, A$*$ algorithm, probabilistic roadmaps (PRMs) [9], artificial potential field (APF) [10, 11], and rapidly-exploring random trees (RRTs) [12, 13], as well as numerical optimization methods such as model predictive control (MPC) [14, 15]. In the past few years, MPC has emerged as an efficient method for addressing multi-constraint issues, rendering it well-suited for trajectory planning [16, 17]. The trajectory planning algorithm based on MPC is extensively employed in various applications. The APF algorithm, which generates repulsive forces around obstacles to prevent collisions and attractive forces toward target points, is widely used in trajectory planning. Therefore, the combination of APF and MPC can ensure the generation of reliable and safe obstacle avoidance trajectories.

Numerous scholars have proposed diverse algorithms based on MPC for trajectory planning with obstacle avoidance. Lee et al. [18] proposed a nonlinear MPC (NMPC) algorithm that considers obstacles as a combination of multi-line segments and incorporates these segments as obstacle avoidance

© The Author(s), 2026. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574726103701 Published online by Cambridge University Press

constraints for NMPC, thereby significantly reducing computational burden. However, it should be noted that this algorithm relies solely on hard constraints, which may result in infeasible solutions and cause collisions with obstacles. Li et al. [19] designed an event-triggered MPC (EMPC) to reduce the computational burden associated with the optimization process. However, similar to [18], the utilization of hard constraints to avoid obstacles poses a risk of collision. Since hard constraints must be strictly observed, once these constraints are violated, it will lead to infeasible solutions. Therefore, Zhu et al. [20] proposed the APF-MPC algorithm, which adopts APF as the penalty term of the objective function to avoid infeasible solutions. When the soft constraint is violated, APF will punish the unexpected behavior. However, trajectory planning controllers based on APF-MPC may encounter local optima in multi-obstacle environments. APF-MPC-based trajectory planning controllers often encounter the issue of local optima in multi-obstacle environments. This is because MPC is operated within a limited range of predictions and cannot obtain a global optimal solution. The whale optimization algorithm (WOA) is an optimization technique that draws inspiration from the hunting pattern exhibited by natural pods of whales [21]. WOA can be introduced into MPC to alleviate the problem of local optima, but it is inefficient in obtaining the global optimal solution. To improve the convergence speed and global search capability of WOA, this paper designs an improved whale optimization algorithm (IWOA) by enhancing the search mechanism and parameter adjustment of the WOA. The combination of IWOA and APF-MPC (APF-IMPC) can generate safe and feasible trajectories as well as achieve high-precision trajectory tracking control of CDCRs in multi-obstacle environments.

Another issue is that most existing trajectory planning methods often ignore trajectory smoothness, resulting in low tracking efficiency and high collision risk with obstacles [22]. To address the issue of non-smooth trajectories, various algorithms have been employed to optimize trajectory smoothness, including Bézier curves [23], B-spline curves [24, 25], and cubic polynomials [26]. B-spline curves can be accurately depicted by formulas, making it easy to be modified by control points. The local modification characteristic of B-splines has garnered the favor of numerous researchers for its adaptability. However, the convex hull property of B-spline curves makes it confined within the convex hull of the control polyline, which may result in potential collisions with obstacles during multi-line trajectory optimization. To address the problem of collision with obstacles after B-spline optimization, we propose a two-layer B-spline optimization framework. The first layer adopts B-spline to elevate the smoothness of the collision-free trajectory generated by APF-IMPC. If the trajectory still has a risk of collision with obstacles, the second-level adjustment framework will be activated.

The main achievements of this paper can be summarized as follows:

1) This paper presents a comprehensive framework that integrates an APF-based IMPC with a two-layer B-spline optimization strategy. This innovative framework enhances trajectory planning and tracking performance of CDCRs in multi-obstacle environments.
2) APF is used as the penalty term of the objective function to ensure the generation of a safe and feasible trajectory. Moreover, IWOA is introduced to overcome the local optima of traditional MPC. By incorporating IWOA, APF-IMPC guarantees the generation of global optimal solutions in the trajectory planning process.
3) To improve trajectory smoothness, a novel two-layer B-spline optimization framework is introduced. The first layer can make the trajectory smoother in a relatively simple environment. The second layer will adjust the control points to address any remaining collision issues, ensuring that safe and smooth trajectories are generated in environments with multi-obstacle and narrow passages.

The remaining sections of this article are structured as follows: the kinematic model of the CDCR is discussed in Section 2. Trajectory planning is presented in Section 3. Trajectory tracking is covered in Section 4. The validation of the proposed algorithm and the experimental results are detailed in Sections 5 and 6, respectively. In Section 7, the conclusive statement is ultimately presented.

https://doi.org/10.1017/S0263574726103701 Published online by Cambridge University Press

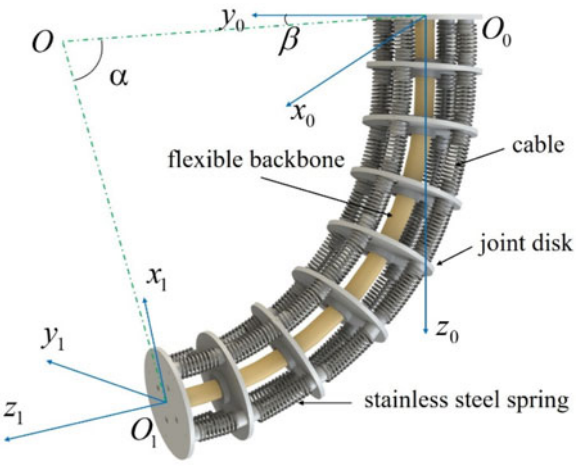


Figure 1. *Structure of the CDCR.*

2. Kinematic modeling

The CDCR consists of a series of flexible joints and four cables that run through each disk. The pose of the CDCR is controlled by pulling these cables, as shown in Figure 1.

The CDCR is depicted in Figure 1. Where α is the bending angle and β is the rotation angle. The robot is composed of a joint disk and a flexible backbone made of silicone rubber. Four high-strength polyethylene fiber cables are evenly distributed at an interval of 90° along the length of the backbone.

It should be noted that the CDCR is nonlinear. Assuming a consistent curvature–a widely adopted and validated approximation for continuum robots operating under no-load or light-payload conditions [27]–the radius of curvature for the four driving ropes can be represented as:

$$\zeta_i = \zeta - r \cdot \cos \beta_i \tag{1}$$

where ζ and ζ*i* ($i = 1, 2. . .4$) are the bending curvature radius of the flexible support and the ith cable, respectively. r is the radius of the indexing circle where the cable hole of the joint disk is located. β_i is the rotational angle of the ith cable generated by the driving cable. $\beta = \beta_i + (i - 1)\ \pi/2$. According to the parallel characteristics of the driving cable, the length variation ΔL_i of the four driving cables can be deduced as follows:

$$\Delta L_i = L - L_i = (\zeta - \zeta_i)\alpha = r\alpha \cdot \cos \beta_i \tag{2}$$

where L and L_i are the single joint length of the robot and the cable length of the *i*th driving cable, respectively.

According to (2), we can get

$$\beta = \arctan\left(\Delta L_{i+1} / \Delta L_i\right) \tag{3}$$

$$\alpha = \Delta L_i / (r \cdot \cos \beta) \tag{4}$$

$\mathbf{C}_t = [x, y, z]^T$ is the position vector of the end effector. The kinematic model of the CDCR is denoted by:

$$\mathbf{C}_t = \left[\left(\frac{L}{\alpha}\right)\cos\beta\,(1-\cos\alpha)\left(\frac{L}{\alpha}\right)\sin\beta\,(1-\cos\alpha)\left(\frac{L}{\alpha}\right)\sin\alpha\right]^T \tag{5}$$

3. Trajectory planning with obstacle avoidance

This section presents an innovative nonlinear APF-IMPC controller to generate obstacle avoidance trajectories in multi-obstacle environments. APF is adopted as the penalty term of the objective function

https://doi.org/10.1017/S0263574726103701 Published online by Cambridge University Press

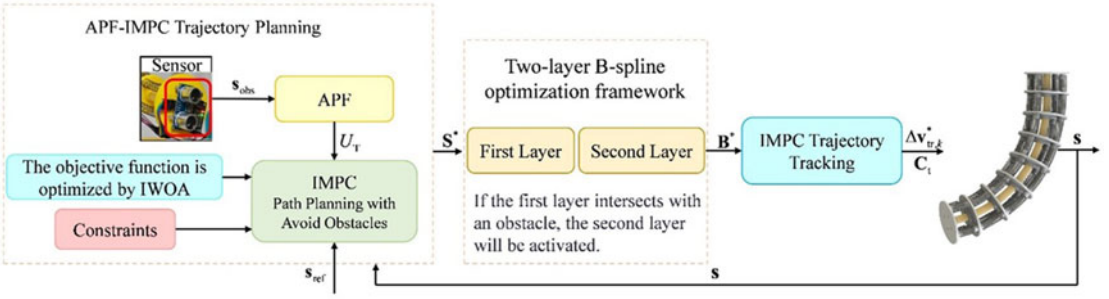


Figure 2. *Diagram of the obstacle avoidance strategy.*

to avoid infeasible solutions caused by only using hard constraints. The utilization of IWOA as the optimization process of MPC solves the problem of local optima. The combination of APF and IMPC ensures the generation of collision-free trajectories for the CDCR. Furthermore, the designed two-layer B-spline optimization framework improves trajectory smoothness and safety. This significantly enhances tracking performance. Figure 2 illustrates the flow chart of the collision-free trajectory planning strategy.

3.1. APF formation

To create an APF potential field, the penalty cost is defined as the repulsive force between the CDCR and obstacles to ensure that the planned trajectory closely follows the desired trajectory while maintaining a safe distance from obstacles.

Let $l(\mathbf{s}, \mathbf{s}_{goal})$ represents the Euclidean distance between the CDCR and the desired position. The current position is denoted as $\mathbf{s}$ and the goal position is denoted as $\mathbf{s}_{goal}$. The attractive field U_{at} is as follows:

$$U_{at} = \begin{cases} \frac{1}{2} M_{at} l^2 \left(\mathbf{s}, \mathbf{s}_{goal}\right) & , l\left(\mathbf{s}, \mathbf{s}_{goal}\right) < l_g \ \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \\ \varepsilon_{apf} M_{at} l(\mathbf{s}, \mathbf{s}_{goal}) & , l(\mathbf{s}, \mathbf{s}_{goal}) \geq l_g \end{cases} \tag{6}$$

where M_{at} is the attractive coefficient, and l_g represents the maximum value of the attract force range. ε_{apf} is a constant coefficient. The gravitational force F_{at} generated by the attractive field U_{at} can be calculated by the following:

$$F_{at} = -\nabla U_{at} = \begin{cases} -M_{at} l(\mathbf{s}, \mathbf{s}_{goal}), & l(\mathbf{s}, \mathbf{s}_{goal}) < l_g \\ -\varepsilon_{apf} M_{at} l(\mathbf{s}, \mathbf{s}_{goal}), & l(\mathbf{s}, \mathbf{s}_{goal}) \geq l_g \end{cases} \tag{7}$$

The repulsive potential field is generated near the obstacle. The diagram of the repulsive potential field is shown in Figure 3. As the CDCR gradually approaches the obstacle, it will enter the repulsive field, which subsequently exerts a force that pushes the CDCR away from the obstacle. This process facilitates stable navigation around obstacles by mitigating negative impacts caused by inertia. The repulsive potential field $U_{j,ru}$ of the obstacle is expressed as:

$$U_{j,ru} = \begin{cases} \frac{1}{2} M_{ru} \left(\frac{1}{l(\mathbf{s}, \mathbf{s}_{j,obs})} - \frac{1}{l_{min}} \right)^2, & l\left(\mathbf{s}, \mathbf{s}_{j,obs}\right) < l_{min} \\ 0, & l(\mathbf{s}, \mathbf{s}_{j,obs}) < l_{min} \end{cases} \tag{8}$$

where M_{ru} is the repulsive coefficient, $l(\mathbf{s}, \mathbf{s}_{j,obs})$ represents the distance between the end of CDCR and the jth obstacle, and l_{min} represents the safe distance. The corresponding repulsive force can be derived

https://doi.org/10.1017/S0263574726103701 Published online by Cambridge University Press

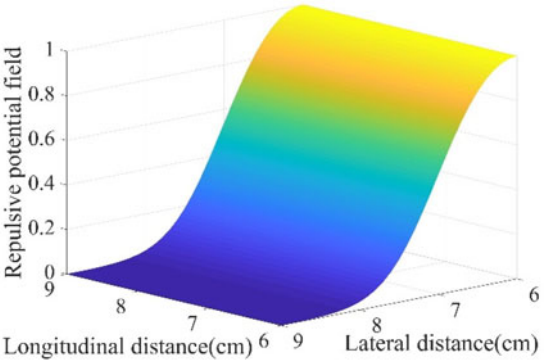


Figure 3. *Schematic diagram of APF repulsive potential field (the obstacle is located at a lateral distance of 6 cm).*

as follows:

$$F_{j,\mathrm{ru}} = -\nabla U_{j,\mathrm{ru}} = \begin{cases} M_{\mathrm{ru}} \left(\dfrac{1}{l(\mathbf{s}, \mathbf{s}_{j,\mathrm{obs}})} - \dfrac{1}{l_{\min}} \right) \dfrac{1}{l(\mathbf{s}, \mathbf{s}_{j,\mathrm{obs}})^2} & l\left(\mathbf{s}, \mathbf{s}_{j,\mathrm{obs}}\right) \le l_{\min} \\ 0 & l(\mathbf{s}, \mathbf{s}_{j,\mathrm{obs}}) > l_{\min} \end{cases} \tag{9}$$

The total potential field U_{T} is calculated using the following formula:

$$U_{\mathrm{T}} = U_{\mathrm{at}} + \sum_{j=1}^{N_j} U_{j,\mathrm{ru}} \tag{10}$$

where N_j is the number of obstacles.

The total force F_{T} is calculated using the following formula:

$$F_{\mathrm{T}} = F_{\mathrm{at}} + \sum_{j=1}^{N_j} F_{j,\mathrm{ru}} \tag{11}$$

By using this method, the CDCR will continue to move towards its target location and effectively avoid obstacles.

3.2. Constraint condition

The imposition of restrictions is essential to maintain the efficiency and stability of the CDCR. The bending angle α of the CDCR is constrained as follows:

$$\alpha_{\min} \le \alpha \le \alpha_{\max} \tag{12}$$

where $\alpha_{\min}$ and $\alpha_{\max}$ are the maximum and minimum angular, respectively. The limited control increment of the CDCR is

$$\Delta \mathbf{v}_{\min} \le \Delta \mathbf{v}(t) \le \Delta \mathbf{v}_{\max} \tag{13}$$

where $\Delta \mathbf{v}$, $\Delta \mathbf{v}_{\min}$ and $\Delta \mathbf{v}_{\max}$ represent control increment, minimum control increment, and maximum control increment, respectively. The control increment $\Delta \mathbf{v}$ limits abrupt changes in the control action, ensuring smoother transitions during trajectory adjustments. Similarly, the rotation angle of the CDCR is limited by:

$$\beta_{\min} \le \beta \le \beta_{\max} \tag{14}$$

where $\beta_{\min}$ and $\beta_{\max}$ represent the minimum and maximum rotation angle, respectively.

https://doi.org/10.1017/S0263574726103701 Published online by Cambridge University Press

Furthermore, incorporate constraints on motion space as follows:

$$\mathbf{s} \in \mathbf{A} \tag{15}$$

where $\mathbf{A}$ is the motion range of the CDCR.

The optimization problem of trajectory planning can be formulated as:

$$\begin{gathered}
\Delta \mathbf{v}^* = \underset{\Delta \mathbf{v}}{\operatorname{argmin}} W(\Delta \mathbf{v}) \\
W(\Delta \mathbf{v}) = \sum_{i=1}^{N_\mathrm{p}} \| \mathbf{s}(k+i|t) - \mathbf{s}_{\mathrm{ref}}(k+i|t) \|_{\mathbf{Q}}^2 + \sum_{i=0}^{N_\mathrm{c}-1} \| \Delta \mathbf{v}(k+i|t) \|_{\mathbf{R}}^2 + \sum_{i=1}^{N_\mathrm{p}} \varepsilon^2 U_\mathrm{T}(k+i|t) \\
\text{s.t: } \alpha_{\min} \le \alpha \le \alpha_{\max} \\
\Delta \mathbf{v}_{\min} \le \Delta \mathbf{v}(t) \le \Delta \mathbf{v}_{\max} \\
\beta_{\min} \le \beta \le \beta_{\max} \\
F_{\min} \le F_\mathrm{T} \le F_{\max} \\
\mathbf{s} \in \mathbf{A}
\end{gathered} \tag{16}$$

where $\mathbf{s}_{\mathrm{ref}}(k+i|t)$ represents the global reference trajectory, N_c expresses the control domain, and N_p expresses the predictive domain. $\mathbf{Q}$ and $\mathbf{R}$ are the state weighting matrix and control increment weighting matrix, respectively. The addition of relaxation factor ε can prevent infeasible solutions in the execution process.

The mathematical representation of the objective function is as follows:

$$\min_{\Delta \mathbf{V}(t)} \frac{1}{2} \mathbf{X}^\mathrm{T} \mathbf{K} \mathbf{X} + \mathbf{f}^\mathrm{T} \mathbf{X} \tag{17}$$

where $\mathbf{X} = \begin{bmatrix} \Delta \mathbf{V}(t) \\ \varepsilon \end{bmatrix}$, $\mathbf{K} = \begin{bmatrix} 2\mathbf{H}_t^\mathrm{T} \mathbf{Q} \mathbf{H}_t + \mathbf{R} & 0 \\ 0 & 2U_\mathrm{T} \end{bmatrix}$, $\mathbf{f}^\mathrm{T} = \begin{bmatrix} -2\mathbf{e}_\mathrm{s}^\mathrm{T} \mathbf{Q} \mathbf{H}_t & 0 \end{bmatrix}$ and $\Delta \mathbf{V}(t) = \begin{bmatrix} \mathbf{v}(t \mid t) \\ \Delta \mathbf{v}(t+1 \mid t) \\ \cdots \\ \Delta \mathbf{v}(t+N_c \mid t) \end{bmatrix}$. $\mathbf{H}_t$ is the coefficient matrix of control increment $\Delta \mathbf{v}$. $\mathbf{e}_\mathrm{s}$ is the error between the planned trajectory and reference trajectory.

To solve the local optimal problem of MPC, the IWOA algorithm is used for the rolling optimization of MPC:

$$\begin{gathered}
\min_{\Delta \mathbf{V}(t)} \frac{1}{2} \mathbf{X}^\mathrm{T} \mathbf{K} \mathbf{X} + \mathbf{f}^\mathrm{T} \mathbf{X} \\
\text{s.t: } \mathbf{v}_{\min} \le \mathbf{v}(t) \le \mathbf{v}_{\max} \\
\Delta \mathbf{v}_{\min} \le \Delta \mathbf{v}(t) \le \Delta \mathbf{v}_{\max}
\end{gathered} \tag{18}$$

The conventional parameter adjustment method of WOA is relatively simplistic, leading to slow convergence rates when dealing with complex optimization problems. Moreover, it is difficult to obtain the global optimal solution. To address these issues of WOA, we propose an improved version of WOA that incorporates an elite strategy. By employing the elite strategy at each iteration, high-quality individuals are preserved while inferior ones are eliminated to mitigate the impact of random search on convergence outcomes. Therefore, this improvement not only obtains the global optimal solution but also enhances search efficiency.

The elite can be characterized with global control parameters (GC) and individual state parameters (IS).

$$\begin{gathered}
GC = \left(\frac{t_{\max} - t + 1}{t_{\max}} \right)^2 \\
IS = \frac{f(x_i, t) - f_{\min}}{f_{\max} - f_{\min}}
\end{gathered} \tag{19}$$

https://doi.org/10.1017/S0263574726103701 Published online by Cambridge University Press

where t is the iteration times, $t_{\max}$ represents the maximum iteration, $f(x_i, t)$ represents the fitness values of the present individual at time t, and $f_{\min}$ and $f_{\max}$ represent the minimum and maximum fitness values, respectively. The global control parameter GC and individual state parameter IS ensure a balance between exploration and exploitation by dynamically adjusting the search range. The search space is controlled by employing GC, which gradually decreases as the number of iterations increases. Individual fitness values are calculated using IS.

The definition of global control parameters for the elite population is as follows:

$$EP = floor[(GC - 1) \times t_{\max} + TP] \tag{20}$$

where the *floor*() function selects the value to the nearest integer, and *TP* is the total population size. A harmonious equilibrium between exploration and extraction can be attained among elite individuals by selecting the optimal *EP* based on individual fitness values.

The conventional WOA primarily relies on simple random selection and linear parameter adjustment in the random search phase and employs two fixed strategies for local search during the bubble-net attack phase. In contrast, IWOA integrates these phases into a unified approach to improve the global search capability and local search efficiency through the elite retention mechanism, thereby improving overall optimization performance. The expressions for updating an individual location are as follows [21]:

$$\begin{aligned} Z(t+1) &= Z_{\text{best}}(t) - \chi_u \cdot o \cdot \mu_a \\ \mu_a &= |r \cdot Z_b(t) - Z(t)| \\ o &= 2\omega\sigma_1 - \omega \\ \omega &= 2 - \tfrac{2t}{t_{\max}} \end{aligned} \tag{21}$$

where χ_u is the search direction factor indicating the population, $Z(t)$ is the position vector of the current individual, and $Z_{\text{best}}(t)$ is the optimal position vector of the current individual. $r = 2\sigma_2$, σ_1 and σ_2 are random numbers between [0, 1]. ω is a convergence factor that traverses from 2 to 0. If the present fitness is less than the past fitness value, then the χ_u value is -1. If the present fitness value is greater than the past fitness value, then the χ_u value is 1. Each iteration generates two new individuals to improve the quality of the elite individual:

$$\begin{aligned} Z_{\text{new1}} &= Z_{\min} + \sigma(Z_{\max} - Z_{\min}) \\ Z_{\text{new2}} &= Z_{\max} + \sigma(Z_{\max} - Z_{\min}) \end{aligned} \tag{22}$$

where $Z_{\min}$ and $Z_{\max}$ are the individuals with the lowest and highest fitness values in the population, respectively, σ is the random number between [0, 1].

To improve search accuracy, the original logarithmic spiral is replaced with an Archimedean spiral. The equation for the bubble-net attack phase is provided as:

$$\begin{aligned} Z(t+1) &= \mu_b \cdot \varepsilon\tau \cdot \cos(2\pi\tau) + Z_b(t) \\ \mu_b &= |r \cdot Z_{\text{best}}(t) - Z(t)| \end{aligned} \tag{23}$$

where ε is the defining constant of the Archimedean helix, τ is a random number in $[-1, 1]$.

The comprehensive model can be formulated as:

$$Z(t+1) = \begin{cases} Z(t+1) = Z_{\text{best}}(t) - \chi_u \cdot o \cdot \mu_a & \sigma < \upsilon \\ Z(t+1) = \mu_b \cdot \varepsilon\tau \cdot \cos(2\pi\tau) + Z_b(t) & \sigma > \upsilon \end{cases} \tag{24}$$

where υ is employed to concurrently achieve the exploration phase (prey encirclement) and the exploitation phase (bubble-netting), and υ changes between 1 and 0 as the number of iterations increases.

After the rolling optimization process of IWOA, certain columns indicating the control increment are obtained as:

$$\Delta\mathbf{V} = \left[\, \Delta\boldsymbol{v}^*_{z,t} \;\; \Delta\boldsymbol{v}^*_{z,t+1} \;\cdots\; \Delta\boldsymbol{v}_{z,t+N_c-1}{}^* \,\right]^{\mathrm{T}} \tag{25}$$

https://doi.org/10.1017/S0263574726103701 Published online by Cambridge University Press

where $\Delta \boldsymbol{v}^*_{z,t}, \Delta \boldsymbol{v}^*_{z,t+1}, \ldots, \Delta \boldsymbol{v}^*_{z,t+N_c-1}$ represent the control increment obtained through rolling optimization using IWOA.

The control output can be calculated by

$$\boldsymbol{v}_z(t) = \boldsymbol{v}_z(t-1) + \Delta \boldsymbol{v}_{zt}{}^* \tag{26}$$

The completion of the MPC rolling optimization will result in the generation of a series of motion trajectories, which are defined as:

$$\mathbf{S}^* = \left[\mathbf{s}^*(t)\ \mathbf{s}^*(t+1) \cdots \mathbf{s}^*(t+N_c-1) \right]^{\mathrm{T}} \tag{27}$$

3.3. Optimize trajectories using the two-layer B-spline optimization framework

The combination of APF and IMPC generates a collision-free trajectory for the CDCR in the environment with obstacles. However, the planned trajectory generated by APF-IMPC may lack smoothness, which reduces tracking accuracy and stability of CDCRs and increases the risk of collision with obstacles. Therefore, in order to improve the smoothness of the planned trajectories, a two-layer B-spline optimization framework is employed.

3.3.1. B-spline curves preliminaries

B-splines can generate continuous and differentiable piecewise polynomial curves. The knot values of the B-spline curve serve as the generated points of the trajectory. The formula expressing the B-spline curve is presented as [27]:

$$\mathbf{b}(u) = \sum_{i=0}^{m} N_{i,p}(u)\, \mathbf{p}_i, \quad u_0 \le u \le u_e \tag{28}$$

where $N_{i,p}(u)$ represents the basis function of the B-spline curve at knot value u, p stands for the degree of the B-spline curve, and $\mathbf{p}_i$ denotes control points of the B-spline curve.

According to the Cox-DeBoor recursion, the basic functions of the B-spline curve are as follows:

$$\begin{aligned} N_{i,0}(u) &= \begin{cases} 1 & \text{if } u_i < u < u_{i+1} \\ 0 & \text{other} \end{cases} \\ N_{i,p}(u) &= \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \end{aligned} \tag{29}$$

The kth derivative of $N_{i,p}(u)$ can be calculated as:

$$N^{(k)}_{i,p}(u) = p \left(\frac{N^{(k-1)}_{i,p-1}}{u_{i+p} - u_i} - \frac{N^{(k-1)}_{i+1,p-1}}{u_{i+p+1} - u_{i+1}} \right) \tag{30}$$

where $k = 1, 2, \ldots, p$. So, the kth derivation of the B-spline curve can be computed as:

$$\mathbf{b}^{(k)}(u) = \sum_{i=0}^{n} N^{(k)}_{i,p}(u) \mathbf{p}_i \tag{31}$$

Cubic B-splines are employed to ensure the continuity of velocity and acceleration of the CDCR, while effectively balancing computational efficiency and smoothness constraints [28].

By employing quasi-uniform B-splines for approximate processing, the knot vector of B-splines can be expressed as:

$$\psi = \left\{ \underbrace{u_0, \ldots, u_0}_{p+1}, u_{p+1}, \ldots, u_n, \underbrace{u_e, \ldots, u_e}_{p+1} \right\} \tag{32}$$

where $u_0 = 0$, the node values at both ends of the node vector have a multiplicity of $p + 1$.

https://doi.org/10.1017/S0263574726103701 Published online by Cambridge University Press

B-spline needs to sample $m+1$ points from the trajectory $\mathbf{S}^*$ generated by IMPC. The newly generated $m+1$ points are $\mathbf{b}_0, \mathbf{b}_1, \ldots, \mathbf{b}_m$. The node values are calculated using spline parameterization and the speed constraint [29]:

$$u_{i+4} - u_{i+3} = \max \frac{|\mathbf{s}^*(t+i+1) - \mathbf{s}^*(t+i)|}{v_{\max}}, i = 0, 1, \ldots, m-1. \tag{33}$$

B-spline is utilized to improve the smoothness and continuity of the trajectory. There are two methods of B-spline in trajectory processing: interpolation and approximation. When employing the interpolation method, it is imperative for the trajectory to precisely traverse through the designated feature points. It may result in a substantial alteration of curvature at the control points and subsequently induce trajectory instability. In contrast, the approximation rule enables the trajectory to generally approximate the feature points, thereby reducing the likelihood of abrupt changes and oscillations. Although the approximation method may make the trajectory intersect with obstacles, it yields smoother trajectories compared to interpolation.

The smoothness of the B-spline curve is calculated by the following formula:

$$\min O = \eta \sum_{k=0}^{m} \lambda_k \left| \mathbf{b}\left(\overline{u}_k\right)^2 - \mathbf{s}_k \right|^2 + (1-\eta) \int_{\overline{u}_0}^{\overline{u}_m} \left| \frac{\mathrm{d}^2 \mathbf{b}(u)}{\mathrm{d}u^2} \right|^2 \mathrm{d}u \tag{34}$$

where $\eta \epsilon\ [0, 1]$ is the smoothing parameter, and λ_k is the error measurement weight for the kth segment of the B-spline. The first term is relevant to the distance between the B-spline curve and waypoints. Another term reflects the smoothness penalty. As η increases, the weight of the first term in (34) also increases, which makes the B-spline curve approach the locus point. When η decreases, the second term in (34) will have a greater weight, resulting in a smoother trajectory.

By substituting (28)–(30) in (34), we can get a new objection function:

$$\min O(\mathbf{P}) = \eta \mathrm{tr}[(\mathbf{S}^* - \mathbf{N}\mathbf{P})^{\mathrm{T}} \chi (\mathbf{S}^* - \mathbf{N}\mathbf{P})] + (1-\eta)\mathrm{tr}\left[\mathbf{P}^{\mathrm{T}} \Theta_N \mathbf{P}\right] \tag{35}$$

where $\mathbf{P} = [\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_m]^T, \mathbf{S}^* = [\,\mathbf{s}^*(t)\ \mathbf{s}^*(t+1) \cdots \mathbf{s}^*(t+N_c-1)\,]^{\mathrm{T}}, \chi = \mathrm{diag}(\lambda_0, \ldots, \lambda_m), \{\mathbf{N}\}_{ij} = N_{j,p}(\overline{u}_i)$ and $\{\Theta_N\}_{jk} = \int N_j^{''}(t) N_k^{''}(t) \mathrm{d}t$. By defining the derivative of $O(\mathbf{P}) = 0$, the control point $\mathbf{P}$ can be derived as:

$$\mathbf{P} = \left(\tfrac{1-\eta}{\eta}\Theta_N + \mathbf{N}^{\mathrm{T}} \chi \mathbf{N}\right)^{-1} \mathbf{N}^{\mathrm{T}} \chi \mathbf{S}^* \tag{36}$$

3.3.2. The two-layer B-spline optimization framework

B-spline enhances the smoothness of the optimized APF-IMPC collision-free trajectory. However, due to the uncontrollable change of curvature between control points in the process of obstacle avoidance, the trajectory will collide with obstacles. Therefore, to improve the smoothness and safety of the APF-IMPC planned trajectory, a two-layer B-spline optimization framework is designed. By implementing this second layer, the proposed framework explicitly overcomes the traditional flaw of B-splines, where the curve breaches minimum clearance limits due to convex hull constraints. The diagram in Figure 4 illustrates the flowchart of the two-layer B-spline.

The first layer improves the smoothness of the trajectory generated by APF-IMPC. In the environment with fewer obstacles, it can achieve a safe and smooth trajectory.

It should be noted that it is necessary to determine whether it collides with obstacles after optimization. The formula for detecting the collision between the B-spline optimized trajectory and obstacles is as follows:

$$||\mathbf{b}(u) - \mathbf{s}_{j,\mathrm{obs}}||^2 \geq l_{\min} \tag{37}$$

where $\mathbf{s}_{j,\mathrm{obs}}$ is the position of the closest obstacle. If the first layer's optimized trajectory interacts with the obstacle, the second layer is activated.

The second layer is suitable for use in the case of narrow channels and multi-obstacles. The second layer consists of two processes. The first adjustment process modifies the control parameters of the

https://doi.org/10.1017/S0263574726103701 Published online by Cambridge University Press

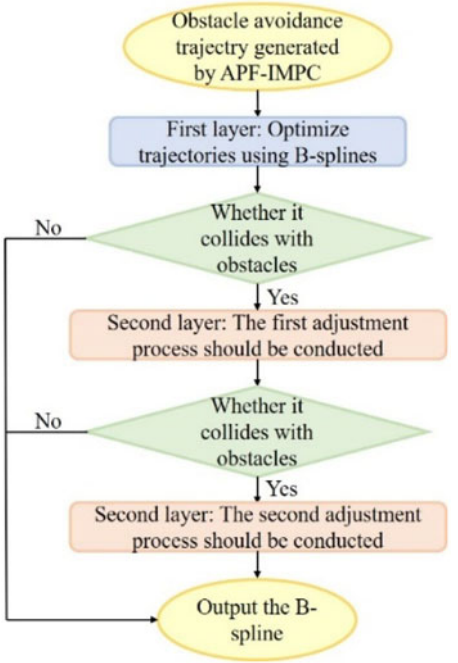


Figure 4. *Flow chart of the two-layer B-spline framework.*

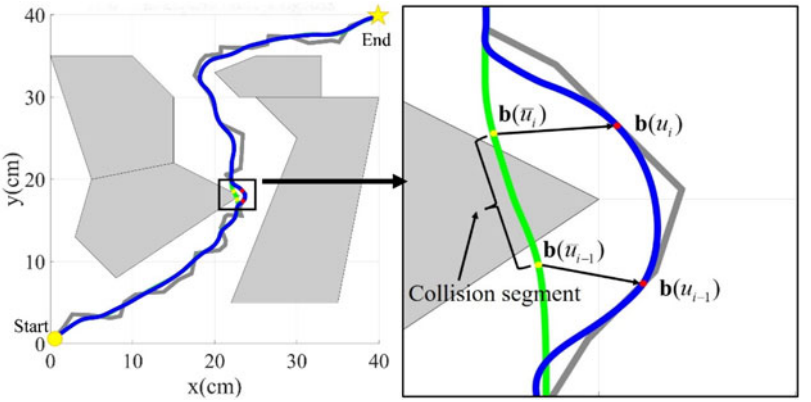


Figure 5. *The first process of the second layer for the B-spline strategy.*

B-spline for the collision segment, making the optimized trajectory closely align with the trajectory generated by APF-IMPC. If the kth segment of the optimized trajectory collides with the obstacle, a significantly large positive integer is assigned to the weight λ_k for measuring segment error, ensuring that the kth B-spline segment moves towards the APF-IMPC trajectory.

The first adjustment process is depicted in Figure 5, where the trajectory produced by APF-IMPC is denoted as a gray line, the B-spline optimization trajectory is depicted by the green line, and the adjusted trajectory is shown in blue. The collision position that occurs between segment $\mathbf{b}(\overline{u}_{i-1})\mathbf{b}(\overline{u}_i)$ and the obstacle is marked by the black grid in Figure 5. Here, $\mathbf{b}(\overline{u}_{i-1})$ and $\mathbf{b}(\overline{u}_i)$ are the track points before adjustment. $\mathbf{b}(u_{i-1})$ and $\mathbf{b}(u_i)$ are the trajectory points generated by the first adjustment process. The trajectory points $\mathbf{b}(\overline{u}_{i-1})$ and $\mathbf{b}(\overline{u}_i)$ move towards $\mathbf{b}(u_{i-1})$ and $\mathbf{b}(u_i)$of the APF-IMPC trajectory by assigning weight parameter $\lambda_{i-1} = \lambda_i = 10^7$ in the colliding segment.

https://doi.org/10.1017/S0263574726103701 Published online by Cambridge University Press

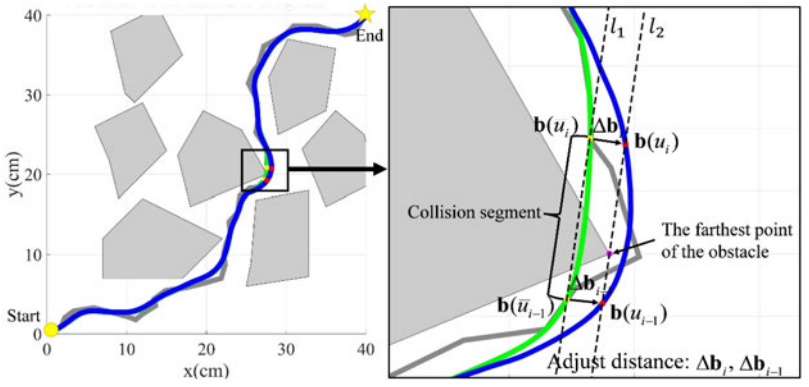


Figure 6. *The second process of the second layer for the B-spline strategy.*

The calculation burden of the first adjustment process is small, allowing it to effectively solve the problem of collision with obstacles in most cases. However, when the CDCR comes to narrow passages, the first adjustment process is not sufficient to ensure safety. Therefore, when the collision with the obstacle still occurs after the first adjustment process, the second adjustment process is activated to further avoid collision with the obstacle.

As shown in Figure 6, the gray trajectory illustrates the trajectory generated by APF-IMPC, while the green trajectory represents the trajectory after applying the first layer of optimization. The blue trajectory depicts the trajectory modified by the second adjustment process of the second layer. If the trajectory optimized by B-spline collides with the obstacle at $\mathbf{b}(\overline{u}_{i-1})\mathbf{b}(\overline{u}_i)$, then $\mathbf{b}(\overline{u}_{i-1})$ and $\mathbf{b}(\overline{u}_i)$ need to move towards the outside of the obstacle. Since the obstacle avoidance trajectory after B-spline optimization is always curved, we only need to ensure that the two endpoints $\mathbf{b}(\overline{u}_{i-1})$ and $\mathbf{b}(\overline{u}_i)$ of the collision section are in the same straight line with the farthest point of the obstacle so as to ensure that the trajectory is collision-free (as shown in the second image of Figure 6).

Therefore, according to the positions of points $\mathbf{b}(\overline{u}_{i-1})$ and $\mathbf{b}(\overline{u}_i)$, a straight line l_1 passing through the two points can be obtained. Move the straight line l_1 in the direction without obstacles until it reaches the farthest point of obstacles. This line is called l_2. Draw a perpendicular line from $\mathbf{b}(\overline{u}_i)$ to l_1, and the intersection point between the perpendicular line and l_2 is $\mathbf{b}(u_i)$. The adjusted distance between $\mathbf{b}(\overline{u}_i)$ and $\mathbf{b}(u_i)$ is denoted as $\Delta\mathbf{b}_i$. Similarly, we can obtain $\mathbf{b}(u_{i-1})$ and $\Delta\mathbf{b}_{i-1}$. This linear solution can not only ensure the safety of the trajectory but also reduce the computational burden. It should be noted that in practical applications, the vertical distance between line l_2 and the farthest point of the obstacle should be greater than the safe distance.

Next, the optimized collision-free trajectory can be obtained only by the required moving distance of the control point. According to (28), we can obtain

$$\left[N_{i,3}\left(\overline{u}_i\right) N_{i+1,3}\left(\overline{u}_i\right) N_{i+2,3}\left(\overline{u}_i\right)\right]\begin{bmatrix}\Delta\mathbf{p}_i^{\mathrm{T}}\\ \Delta\mathbf{p}_{i+1}^{\mathrm{T}}\\ \Delta\mathbf{p}_{i+2}^{\mathrm{T}}\end{bmatrix}=\Delta\mathbf{b}_i^{\mathrm{T}} \tag{38}$$

where $\Delta\mathbf{p}_i$ represents the movement distance of the control point. As can be seen from Figure 6, points $\mathbf{b}(\overline{u}_{i-1})$ and $\mathbf{b}(\overline{u}_i)$ need to be adjusted at the same time. Therefore, it can be obtained by extending (39):

$$\begin{bmatrix}N_{i-1,3}\left(\overline{u}_{i-1}\right) & N_{i,3}\left(\overline{u}_{i-1}\right) & N_{i-1,3}\left(\overline{u}_{i-1}\right) & 0\\ 0 & N_{i,3}\left(\overline{u}_i\right) & N_{i+1,3}\left(\overline{u}_i\right) & N_{i+2,3}\left(\overline{u}_i\right)\end{bmatrix}\begin{bmatrix}\Delta\mathbf{p}_{i-1}^{\mathrm{T}}\\ \Delta\mathbf{p}_i^{\mathrm{T}}\\ \Delta\mathbf{p}_{i+1}^{\mathrm{T}}\\ \Delta\mathbf{p}_{i+2}^{\mathrm{T}}\end{bmatrix}=\begin{bmatrix}\Delta\mathbf{b}_{i-1}{}^{\mathrm{T}}\\ \Delta\mathbf{b}_i{}^{\mathrm{T}}\end{bmatrix} \tag{39}$$

https://doi.org/10.1017/S0263574726103701 Published online by Cambridge University Press

By applying the minimum-length solution method [30], we can get

$$\Delta \mathbf{P}_c^{\mathrm{T}} = \mathbf{N}_{cp}^{\mathrm{T}} \left(\mathbf{N}_{cp} \mathbf{N}_{cp}^{\mathrm{T}} \right)^{-1} \Delta \mathbf{B}_c^{\mathrm{T}} \tag{40}$$

where $\Delta \mathbf{P}_c^{\mathrm{T}} = [\ \Delta \mathbf{p}_{i-1}^{\mathrm{T}} \quad \Delta \mathbf{p}_i^{\mathrm{T}} \quad \Delta \mathbf{p}_{i+1}^{\mathrm{T}} \quad \Delta \mathbf{p}_{i+2}^{\mathrm{T}}\ ]^{\mathrm{T}}$, $\mathbf{N}_{cp} = \begin{bmatrix} N_{i-1,3}(\overline{u}_{i-1}) & N_{i,3}(\overline{u}_{i-1}) & N_{i-1,3}(\overline{u}_{i-1}) & 0 \\ 0 & N_{i,3}(\overline{u}_i) & N_{i+1,3}(\overline{u}_i) & N_{i+2,3}(\overline{u}_i) \end{bmatrix}$, $\Delta \mathbf{B}_c^{\mathrm{T}} = \begin{bmatrix} \Delta \mathbf{b}_{i-1}{}^{\mathrm{T}} \\ \Delta \mathbf{b}_i{}^{\mathrm{T}} \end{bmatrix}$. After the adjustment process, a smoother collision-free trajectory can be obtained.

4. Trajectory tracking

IMPC is also further employed to guarantee precise tracking of the planned trajectory.

It is worth noting that while the geometric adjustment in the B-spline framework ensures spatial safety, it may occasionally generate reference points that challenge the joint rotation limits. The incorporation of rigorous kinematic constraints (12), (13), (14), and (15) within this tracking IMPC serves as a critical safeguard, ensuring that the physical execution strictly conforms to the flexible joints' rotational and bending capacities.

The IMPC algorithm adjusts acceleration iteratively to achieve optimal values, thereby preventing velocity jumps during tracking. In this study, APF is not employed in trajectory tracking for two primary reasons: (1) during the trajectory planning stage, a safe collision-free trajectory has already been generated by APF-IMPC. (2) APF utilized as the penalty term of the objective function will increase computational burden. Therefore, hard constraints are introduced in trajectory tracking to ensure that the distance between the CDCR and obstacles remains greater than the safe distance.

The MPC-based trajectory tracking controller is constructed as follows:

$$\begin{gathered} \Delta \mathbf{v}_{tr}^{*} = \underset{\Delta v}{\operatorname{argmin}} W\left(\Delta \mathbf{v}_{\mathrm{tr}}\right) \\ W\left(\Delta \mathbf{v}_{\mathrm{tr}}\right) = \sum_{j=1}^{N_{\mathrm{lp}}} \| \mathbf{s}\left(k+j|t\right) - \mathbf{b}^{*}(k+j|t) \|_{\mathbf{Q}_{\mathrm{l}}}^{2} + \sum_{j=0}^{N_{\mathrm{lc}}-1} \| \Delta \mathbf{v}_{\mathrm{tr}}\left(k+j|t\right) \|_{\mathbf{R}_{\mathrm{l}}}^{2} \\ \text{s.t: } l(s, s_{obs}) > l_{min} \\ \alpha_{\min} \le \alpha \le \alpha_{\max} \\ \Delta \mathbf{v}_{\min} \le \Delta \mathbf{v}_{\mathrm{tr}}(t) \le \Delta \mathbf{v}_{\max} \\ \beta_{\min} \le \beta \le \beta_{\max} \\ \mathbf{s} \in \mathbf{A} \end{gathered} \tag{41}$$

where $\Delta \mathbf{v}_{\mathrm{tr}}$ represents the control increment of the CDCR. N_{lp} and N_{lc} are prediction domains and control domains, respectively. $\mathbf{Q}_{\mathrm{l}}$ is the state weighting matrix, and $\mathbf{R}_{\mathrm{l}}$ is the control increment weighting matrix. Impose stability constraints:

$$||\mathbf{s}(k+N_{\mathrm{lp}}|t) - \mathbf{b}^{*}(k+N_{\mathrm{lp}}|t)||_{\mathbf{Q}_{\mathrm{l}}}^{2} = 0 \tag{42}$$

Convert (42) to the objective function form with a constraint:

$$\begin{gathered} \min_{\Delta V(t)} \frac{1}{2} \mathbf{X}_{\mathrm{tr}}{}^{\mathrm{T}} \mathbf{K}_{\mathrm{tr}} \mathbf{X}_{\mathrm{tr}} + \mathbf{f}_{\mathrm{tr}}{}^{\mathrm{T}} \mathbf{X}_{\mathrm{tr}} + \epsilon^{2} \rho_{\mathrm{tr}} \\ \text{s.t: } \mathbf{v}_{\min} \le \mathbf{v}(t) \le \mathbf{v}_{\max} \\ \Delta \mathbf{v}_{\min} \le \Delta \mathbf{v}_{\mathrm{tr}}(t) \le \Delta \mathbf{v}_{\max} \end{gathered} \tag{43}$$

where $\mathbf{X}_{\mathrm{tr}} = \begin{bmatrix} \Delta \mathbf{V}_{\mathrm{tr}}(t) \\ \varepsilon \end{bmatrix}$, $\mathbf{K}_{\mathrm{tr}} = \begin{bmatrix} 2\mathbf{H}_{\mathrm{tr}}^{\mathrm{T}} \mathbf{Q}_{\mathrm{l}} \mathbf{H}_{\mathrm{tr}} + \mathbf{R}_{\mathrm{l}} & 0 \\ 0 & \rho_{\mathrm{tr}} \end{bmatrix}$, $\mathbf{f}_{\mathrm{tr}}{}^{\mathrm{T}} = \begin{bmatrix} -2\mathbf{e}_{\mathrm{str}}^{\mathrm{T}} \mathbf{Q}_{\mathrm{l}} \mathbf{H}_{\mathrm{tr}} & 0 \end{bmatrix}$ and $\Delta \mathbf{V}_{\mathrm{tr}}\left(t\right) = \begin{bmatrix} \Delta \mathbf{v}_{\mathrm{tr}}(k \mid t) \\ \Delta \mathbf{v}_{\mathrm{tr}}(k+1 \mid t) \\ \cdots \\ \Delta \mathbf{v}_{\mathrm{tr}}(k+N_{\mathrm{c}} \mid t) \end{bmatrix}$. $\mathbf{v}(t)$, $\mathbf{v}_{\min}$ and $\mathbf{v}_{\max}$ are the moving speed, minimum moving speed, and maximum

https://doi.org/10.1017/S0263574726103701 Published online by Cambridge University Press

moving speed, respectively. $\mathbf{H}_{\mathrm{tr}}$ is the coefficient matrix of $\Delta\mathbf{v}_{\mathrm{tr}}$. $\mathbf{e}_{\mathrm{str}}$ is the error between the tracking trajectory and the planned trajectory.

In the process of tracking trajectory, IWOA is used to optimize $\Delta\mathbf{V}_{\mathrm{tr}}(t)$ to reduce the oscillation phenomenon and tracking error. The IWOA optimization process still uses (19)–(24), which is not detailed here.

After dynamic rolling optimization of IWOA, it can get a series of speed increments.

$$\Delta\mathbf{V}_{\mathrm{tr}}{}^{*} = \left[\Delta\mathbf{v}_{\mathrm{tr},k}{}^{*}\ \Delta\mathbf{v}_{\mathrm{tr},k+1}{}^{*}\ \cdots\ \Delta\mathbf{v}_{\mathrm{tr},k+N_{\mathrm{lc}}-1}{}^{*}\right]^{\mathrm{T}} \tag{44}$$

where $\Delta\mathbf{v}_{\mathrm{tr},k}{}^{*}$, $\Delta\mathbf{v}_{\mathrm{tr},k+1}{}^{*}$, …, $\Delta\mathbf{v}_{\mathrm{tr},k+N_{\mathrm{lc}}-1}{}^{*}$ represents the control increment obtained through the rolling optimization using IWOA.

5. Simulation and discussion

5.1. Simulation platform construction

In this part, the effectiveness of the proposed APF-IMPC and two-layer B-spline optimization framework is verified through a series of simulations using MATLAB/Simulink 2022b.

Parameter setting: To mitigate the occurrence of sudden speed change, the permissible range of joint velocities is set between −0.3 rad/s and 0.3 rad/s. Additionally, the constraints on the bending angle α and rotation angle β are (0, π/2) and (−π, π), respectively. $N_{\mathrm{p}} = 6, N_{\mathrm{c}} = 1$. The time interval between samples is 0.1 s. The safe distance between the CDCR and obstacles is $l_{\mathrm{min}} = 4$ cm.

5.2. Simulation comparison

Simulations are conducted to evaluate the performance of APF-IMPC and the two-layer B-spline optimization.

1) *Obstacle avoidance effectiveness of APF-IMPC:* Figure 7 shows the trajectories of four different algorithms in a multi-obstacle environment. Table I shows the total length of these four trajectories. The effectiveness of the proposed APF-IMPC is comprehensively validated in comparison with NMPC [18], EMPC [19], and APF-MPC [20]. In a multiple obstacle scene depicted in Figure 7, the start point is (1, 1, 0) and the endpoint is (40, 40, 25). As illustrated in Figure 7, both NMPC and EMPC encounter collisions with obstacles. This is because EMPC and NMPC only use hard constraints to avoid obstacles, making them easy to collide with obstacles due to infeasible solutions during the planning process. In contrast, both APF-IMPC and APF-IMPC adopt APF as the penalty term of the objective function to avoid the occurrence of infeasible solutions. Moreover, it can be seen from Table I that the planned trajectory length of APF-MPC is larger than that of APF-IMPC. This is because APF-MPC falls into local optima, while APF-IMPC adopts the IWOA algorithm to derive the global optimal solution for MPC. Therefore, APF-IMPC generates a shorter trajectory. To evaluate the computational burden, the average execution time per control step for each algorithm was recorded, as shown in Table II. Although the proposed APF-IMPC exhibits a slightly higher execution time compared to standard NMPC due to the integration of the APF penalty, its average computational time remains well below the 0.4s sampling period used in our physical setup. This indicates that the proposed algorithm efficiently balances trajectory safety, smoothness, and real-time computational feasibility without causing processor overload.
2) Two-layer B-spline optimization effectiveness: The second set of simulation experiments verifies the effectiveness of the two-layer B-spline optimization. We validate the first and second layers of the B-spline optimization separately. The first layer uses B-spline to improve the smoothness of the planned trajectory. Figure 8 clearly demonstrates that the first layer of B-spline optimization significantly enhances trajectory smoothness.

https://doi.org/10.1017/S0263574726103701 Published online by Cambridge University Press

Table I. *Length of the planned trajectory for these four methods.*

Algorithms	Length of the planned trajectory
APF-IMPC	64.29 cm
NMPC	63.75 cm
EMPC	68.52 cm
APF-MPC	71.28 cm

Table II. *Average execution time per control step for different algorithms*

Algorithms	Length of the planned trajectory
APF-IMPC	85 ms
NMPC	45 ms
EMPC	38 ms
APF-MPC	65 ms

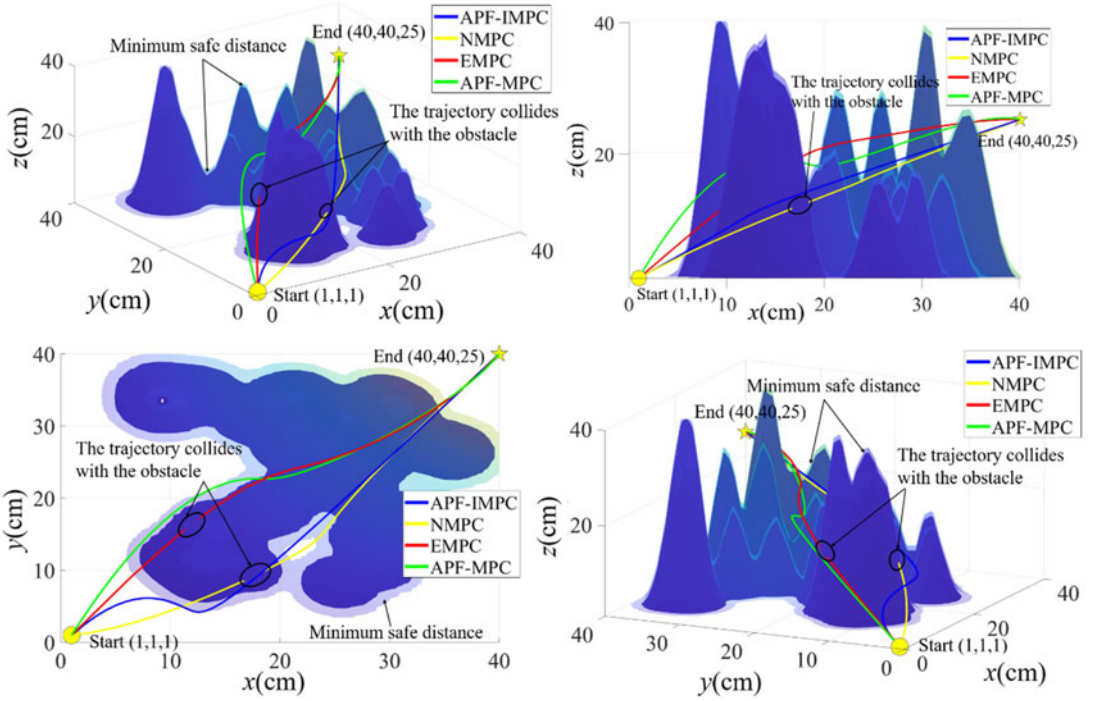


Figure 7. *Simulation results of obstacle avoidance in multi-obstacle environments.*

As depicted in Figure 9, the use of the first layer may result in collisions. By implementing the first adjustment progress in the second layer of the two-layer B-spline optimization framework, the colliding segments are moved to the original trajectory generated by APF-IMPC. It can be seen from Figure 9 that the collision problem is successfully solved by the first adjustment progress.

In the scene of multi-obstacle and narrow channels, the first adjustment process in the second layer may also cause a collision with obstacles. As shown in Figure 10, if the first progress of the second

https://doi.org/10.1017/S0263574726103701 Published online by Cambridge University Press

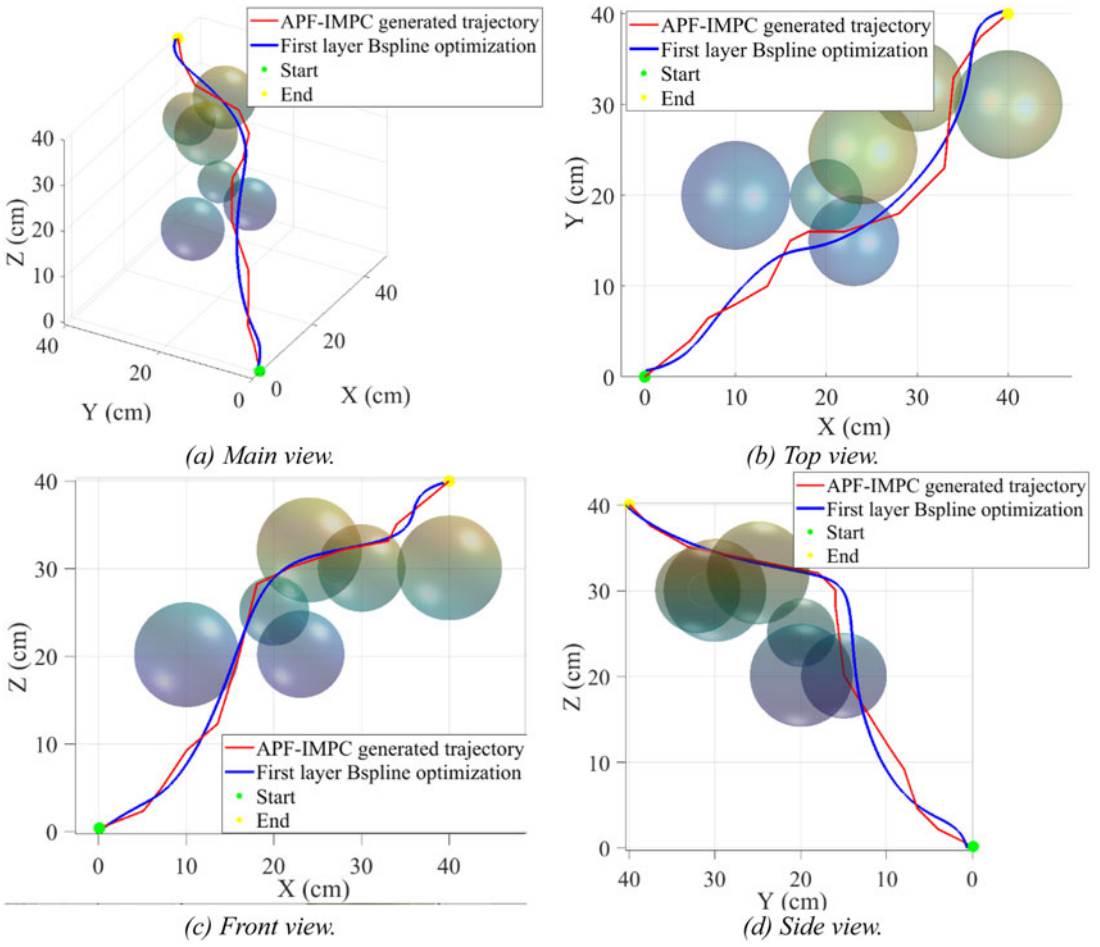


Figure 8. *Optimization result of the first layer B-spline.*

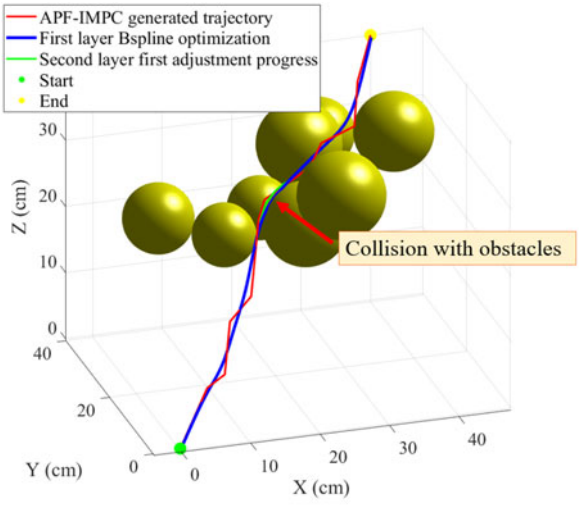
layer for the two-layer B-spline optimization framework is still unable to separate the trajectory from the obstacles, then the second adjustment progress is implemented to move the trajectory point of the collision section to the side without obstacles, ensuring the safety of the optimized trajectory.

Through the above simulations, it can be verified that the two-layer B-spline optimization framework improves trajectory smoothness while ensuring safe obstacle avoidance.

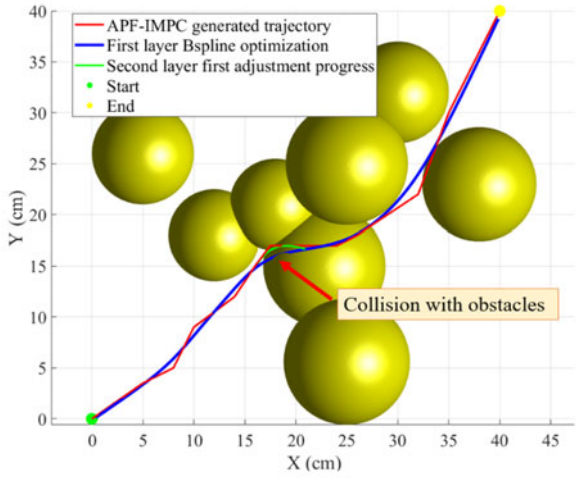
6. Experiment results

The obstacle avoidance performance of the APF-IMPC nested B-spline optimization framework, NMPC [18], EMPC [19], and APF-MPC [20] in a multi-obstacle environment is compared through experiments. As shown in Figure 11, the posture sensor GY-95T and the ultrasonic sensor are positioned at the end of the CDCR to measure its position and distance from obstacles, respectively. Four TBS-K30 servos are arranged at the top of the CDCR. The sampling period is 0.4 s. The experimental sampling period was set to 0.4 s to accommodate the mechanical response time of the cable-driven system. This lag primarily arises from the necessity to tension the driving cables and the complex synchronization required among multiple actuators before a posture change is executed.

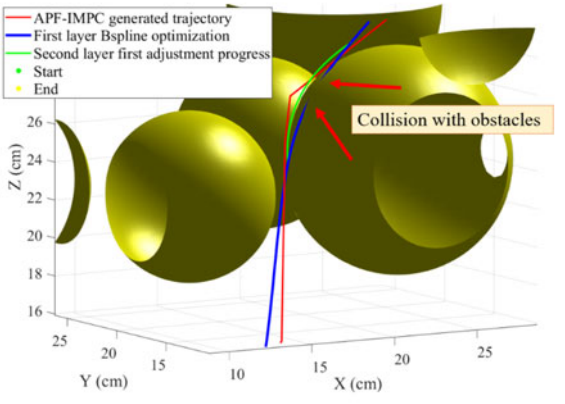
https://doi.org/10.1017/S0263574726103701 Published online by Cambridge University Press



(a) Main view.



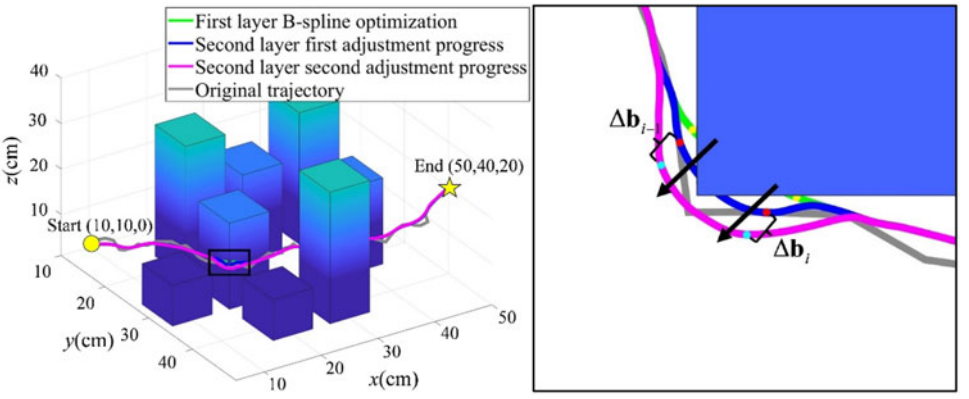
(b) Top view.



(c) Partial top view.

Figure 9. *Simulation of the first adjustment progress in the second layer of the two-layer B-spline optimization framework.*

https://doi.org/10.1017/S0263574726103701 Published online by Cambridge University Press



(a) The overall picture. *(b) Partial top view.*

Figure 10. *Simulation of the second adjustment progress in the second layer of the two-layer B-spline optimization framework.*

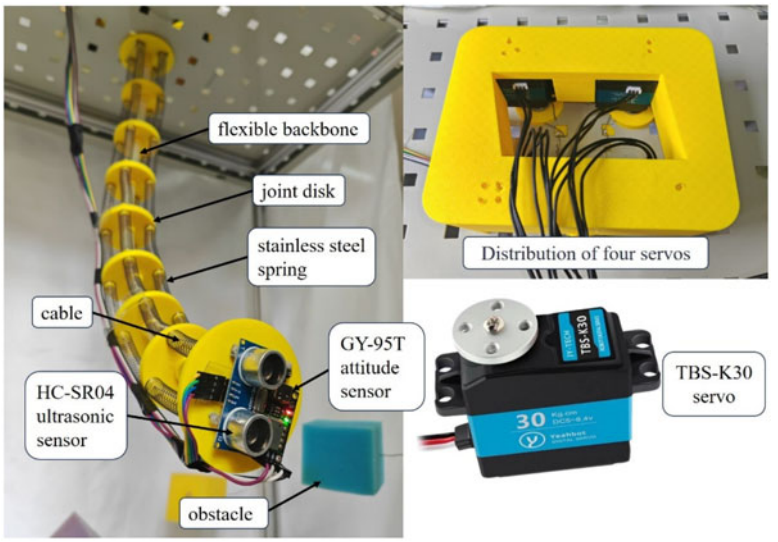


Figure 11. *Experimental setup of the CDCR.*

6.1. Obstacle avoidance effectiveness

Figure 12 shows the obstacle avoidance process of APF-IMPC, NMPC, EMPC, and APF-MPC. In this comparative experiment, to ensure consistency, all algorithms adopt the two-layer B-spline optimization framework. As depicted in Figure 12 (a) and Figure 12 (d), APF-IMPC and APF-MPC demonstrate a safe obstacle avoidance capability. On the contrary, it can be seen from Figure 12 (b) that the trajectory of the NMPC collides with a pink obstacle at the marked position of the blue circle. Similarly, in Figure 12 (c), the trajectory controlled by EMPC collides with a yellow obstacle at the marked position of the blue circle. The reason is that NMPC and EMPC only depend on hard constraints, which may lead to infeasible solutions. In contrast, both APF-IMPC and APF-IMPC adopt APF as the penalty term of the objective function to solve these problems.

However, APF-MPC suffers from the jitter problem, as indicated by the two positions marked with purple circles in Figure 12 (d). On the contrary, there is no jitter in the tracking trajectory of APF-IMPC in Figure 12 (a). This is because the APF-MPC algorithm is easy to get stuck in local optima. In contrast,

https://doi.org/10.1017/S0263574726103701 Published online by Cambridge University Press

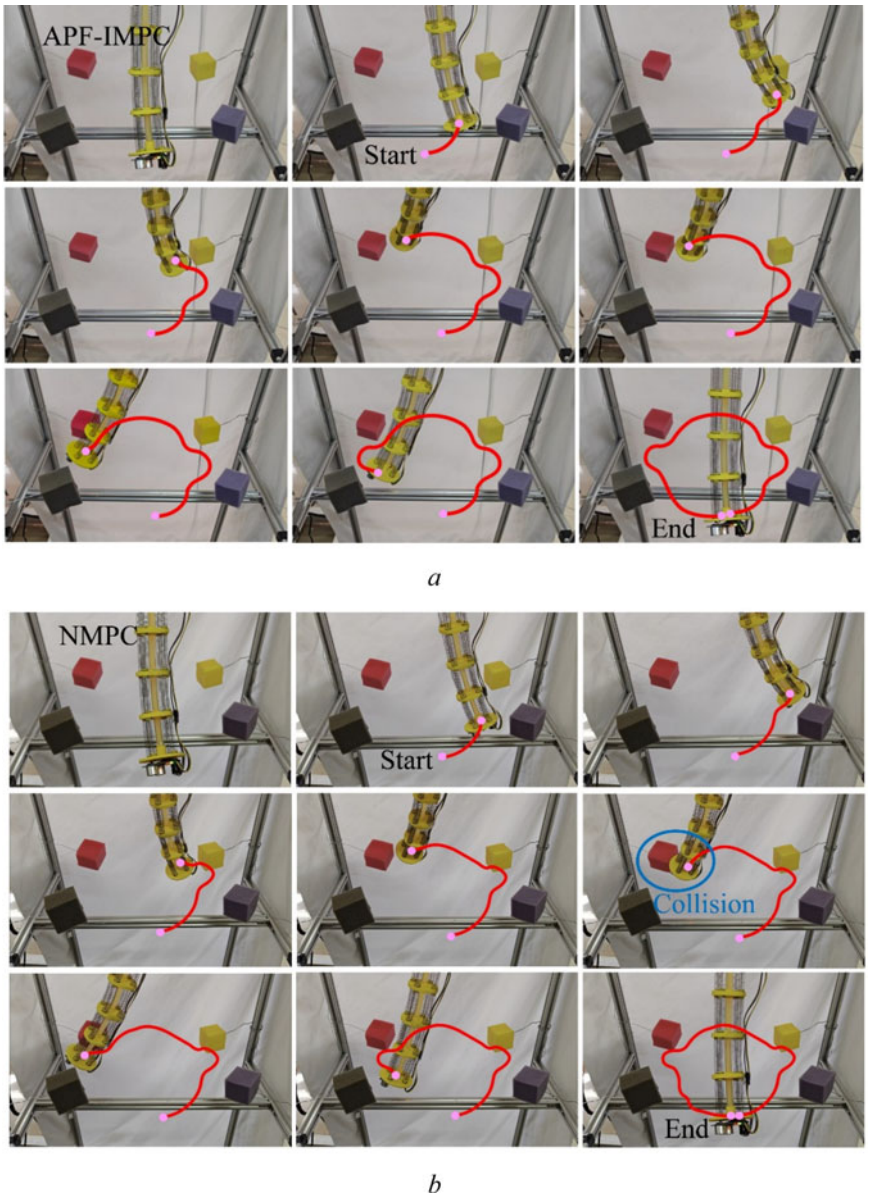
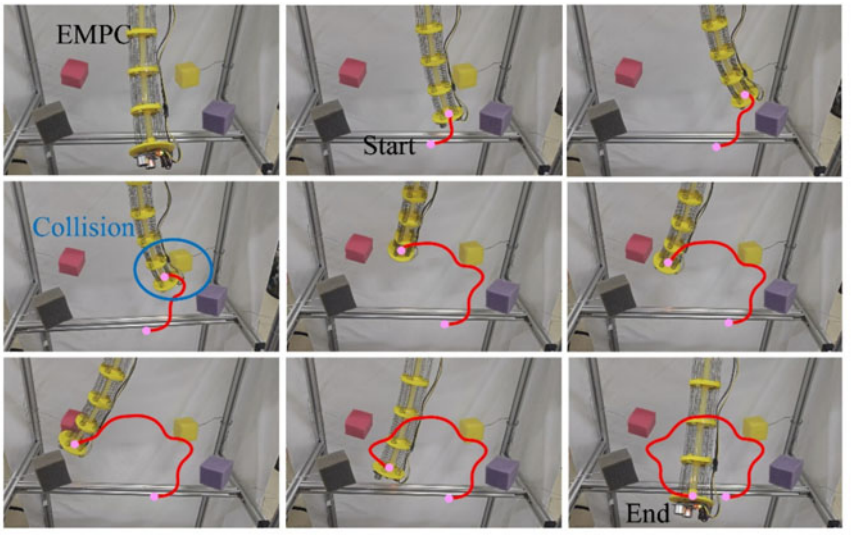


Figure 12. *O-shaped collision-free trajectory in a multi-obstacle environment. (a) APF-IMPC. (b) NMPC. (c) EMPC. (d) APF-MPC.*

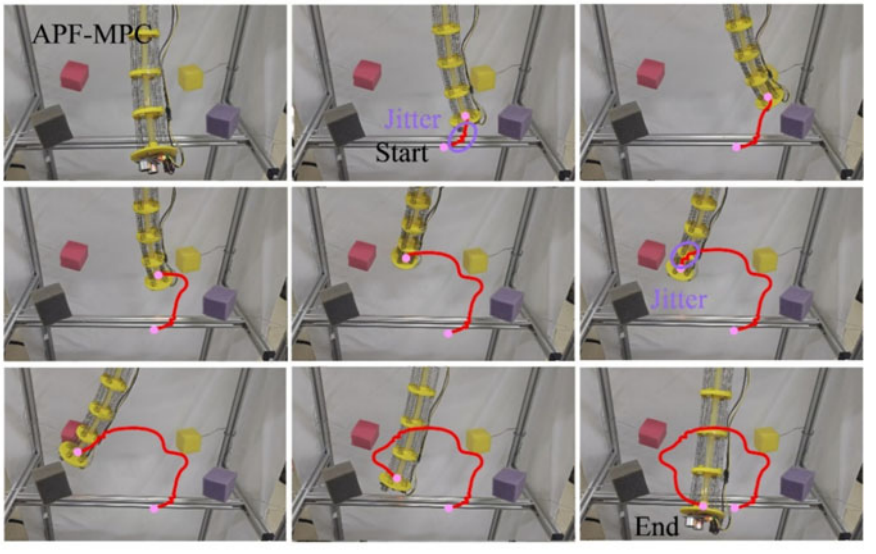
the designed IWOA algorithm adopted in APF-IMPC can help the control system avoid local optima and obtain the global optimal solution.

To quantitatively evaluate the stability of the trajectory tracking, the positional errors (e_x, e_y, and e_z) between the end effector and the desired trajectory are recorded. As depicted in Figure 13, the tracking errors along the x, y, and z axes are consistently constrained within a small bounded range. Notably, the error curves exhibit smooth transitions without high-frequency oscillations or abrupt spikes, even during aggressive obstacle avoidance maneuvers. These evaluation indicators firmly demonstrate the superior stability and anti-jitter capability of the proposed APF-IMPC tracking controller.

https://doi.org/10.1017/S0263574726103701 Published online by Cambridge University Press



c



d

Figure 12. *Continued.*

Therefore, it can be seen from the experiment that APF-IMPC can effectively generate a safe and feasible trajectory compared to NMPC and EMPC, as well as ensuring high-precision trajectory tracking compared to APF-MPC.

6.2. The effectiveness of B-spline optimization framework

In this experiment, the number of obstacles is increased to test the performance of the two-layer B-spline optimization framework. In Figure 14, the green line represents the planned trajectory generated by the traditional B-spline optimization algorithm. The red line represents the planned trajectory using the two-layer B-spline optimization framework, and the blue dotted line represents the original trajectory without optimization.

https://doi.org/10.1017/S0263574726103701 Published online by Cambridge University Press

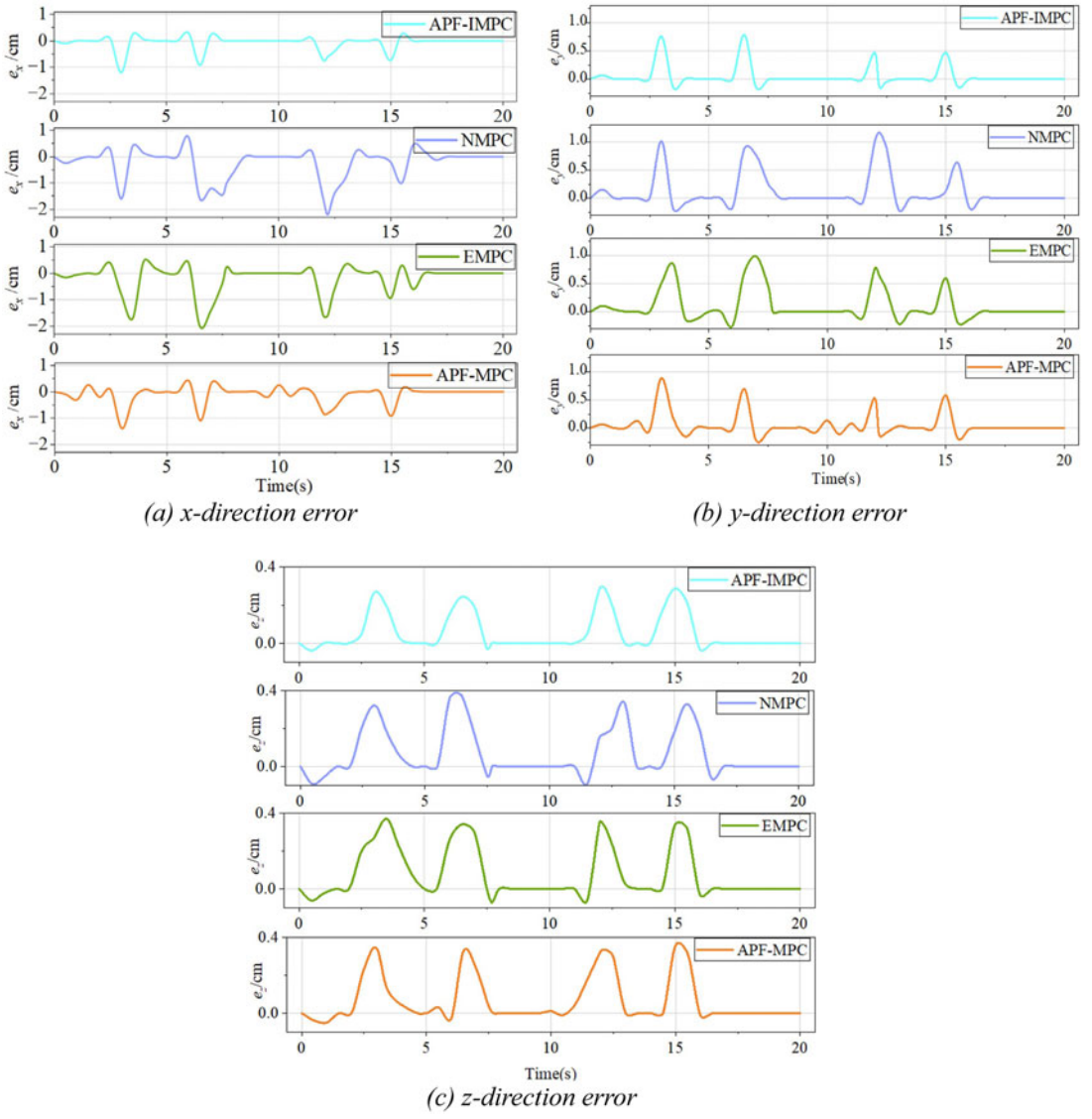


Figure 13. *Tracking error between end effector position and desired position.*

It can be clearly seen from Figure 14 that the trajectory becomes smoother after B-spline optimization. However, two collisions occurred at the blue circle mark when using the traditional B-spline algorithm. The reason is that the traditional B-spline optimization algorithm may cause collision with obstacles after changing the original trajectory. On the contrary, when the two-layer B-spline optimization framework encounters the risk of collision with obstacles after improving trajectory smoothness, it will activate the adjustment strategy of the second layer to modify the trajectory to the collision-free area. Therefore, the trajectory of the two-layer B-spline optimization framework does not collide with obstacles. It demonstrates that the proposed APF-IMPC nested two-layer B-spline optimization framework can effectively reduce the risk of motion mutation and collisions in multi-obstacle environments while improving trajectory smoothness.

https://doi.org/10.1017/S0263574726103701 Published online by Cambridge University Press

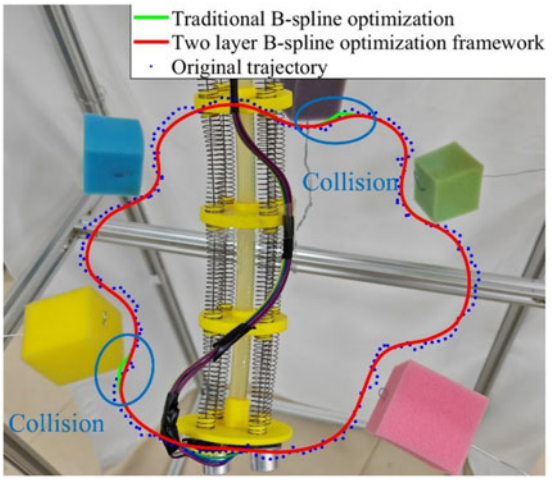


Figure 14. *The planned trajectory by the two algorithms.*

7. Conclusions

This paper proposes an APF-IMPC nested two-layer B-spline optimization framework to achieve safe and efficient obstacle avoidance for the CDCR in a multi-obstacle environment. Adopting APF as the objective function penalty term to further ensure the safety of generated trajectories. In addition, an IWOA is introduced to avoid the local optimal trap of MPC. The proposed two-level B-spline optimization framework enhances the smoothness of the planned trajectory. Simulation and experiment results demonstrate that the proposed algorithm enables the CDCR to safely and smoothly avoid obstacles in multi-obstacle environments while ensuring high-precision trajectory tracking. Although the IWOA-based IMPC effectively mitigates local optima, the current framework has several limitations that define our future work. First, the elite retention strategy will be upgraded with self-adaptive mechanisms for highly dense environments. Second, an event-triggered localized APF will be investigated to enhance robustness against sudden boundary proximity. Third, future work will consider the adaptive scaling of repulsive field parameters to better accommodate irregular physical obstacles. Finally, formalizing the theoretical convergence proofs to explicitly account for physical imperfections–such as data transmission delays and the persistent position drift caused by thermal expansion during long-duration operations–will be critical. Integrating a disturbance observer into the MPC to compensate for these slow-varying thermo-mechanical dynamics will be our next immediate focus.

Author contribution. This article was written by Shishuai Zhao. Qi Chen and Yanan Qin reviewed and helped revise the content.

Financial support. This work was supported by the National Natural Science Foundation of China under grants 52471336, 52127813, and 52431012.

Conflicts of interest. The authors declare no conflicts of interest exist.

References

[1] H. Tian, Z. Gao and Y. Gong, "Rapidly fabricated soft spring actuator via laminated laser-lithographed 2D polyurethane layers," *Mechatronics*. **92**, 102985 (2023).

[2] J. Zhang, Y. Li, Z. Kan, Q. Yuan, H. Rajabi, Z. Wu, H. Peng and J. Wu, "A preprogrammable continuum robot inspired by elephant trunk for dexterous manipulation," *Soft Robot*. **10**(3), 636–646 (2023).

[3] G. Qin, Q. Wang, C. Li, A. Ji, H. Wu, Z. Yang and S. Wen, "Design and development of a cable-driven elephant trunk robot with variable cross-sections," *Indust. Robot: The Int. J. Robot. Res. Appl.* **50**(3), 520–529 (2023).

https://doi.org/10.1017/S0263574726103701 Published online by Cambridge University Press

[4] L. Mao, P. Yang, C. Tian, X. Shen, F. Wang, H. Zhang and H. Xie, "Magnetic steering continuum robot for transluminal procedures with programmable shape and functionalities," *Nat. Commun.* **15**(1), 3759 (2024).
[5] A. M. Zou and Z. Fan, "Distributed fixed-time attitude coordination control for multiple rigid spacecraft," *Int. J. Robust Nonlin.* **30**(1), 266–281 (2020).
[6] P. Polygerinos, Z. Wang, K. C. Galloway, R. J. Wood and C. J. Walsh, "Soft robotic glove for combined assistance and at-home rehabilitation," *Robot. Auton. Syst.* **73**, 135–143 (2015).
[7] C. Wang, M. Yang, Q. Han, P. Li, Z. He, Z. Zhao and F. Wu, "Dynamic litchi fruit trajectory prediction using Extended Kalman Filter in simulated wind environment," *Comput. Electron. Agr.* **247**, 111666 (2026).
[8] H. Wang, G. Zhang, H. Cao, K. Hu, Q. Wang, Y. Deng and Y. Tang, "Geometry-aware 3D point cloud learning for precise cutting-point detection in unstructured field environments," *J. Field Robot.* **42**(7), 3063–3076 (2025).
[9] Y. Huang, H. Wang, L. Han and Y. Xu, "Robot path planning in narrow passages based on improved PRM method," *Intel. Serv. Robot.* **17**, 1–12 (2024).
[10] S. Zhu, Y. Zhang, X. Zong and R. Li, "Finite-time constrained attitude regulation for spacecraft formation with disturbances on SO (3," *Int. J. Robust. Nonlin.* **33**(8), 4459–4489 (2023).
[11] J. F. Duhé, S. Victor and P. Melchior, "Contributions on artificial potential field method for effective obstacle avoidance," *Fract. Calc. Appl. Anal.* **24**, 421–446 (2021).
[12] Y. Shang, F. Liu, P. Qin, Z. Guo and Z. Li, "Research on path planning of autonomous vehicle based on RRT algorithm of Q-learning and obstacle distribution," *Eng. Comput.* **40**(5), 1266–1286 (2023).
[13] S. M. Yang and Y. A. Lin, "Development of an improved rapidly exploring random trees algorithm for static obstacle avoidance in autonomous vehicles," *Sensors.* **21**(6), 2244 (2021).
[14] J. Köhler, P. Kötting, R. Soloperto, F. Allgöwer and M. A. Müller, "A robust adaptive model predictive control framework for nonlinear uncertain systems," *Int. J. Robust. Nonlin.* **31**(18), 8725–8749 (2021).
[15] H. Meng, J. Zhang and S. Li, "Dual-mode robust model predictive control for the tracking control of nonholonomic mobile robot," *Int. J. Robust. Nonlin.* **33**(6), 3627–3639 (2023).
[16] H. Zhang, J. Yao and S. Tian, "Model predictive control for trajectory planning considering constraints on vertical load variation," *Electronics.* **13**(8), 2024 (1488).
[17] K. Zhang and Y. Shi, "Adaptive model predictive control for a class of constrained linear systems with parametric uncertainties," *Automatica.* **117**, 108974 (2020).
[18] C. Lee, D. Chung, J. Kim and J. Kim, "Nonlinear model predictive control with obstacle avoidance constraints for autonomous navigation in a canal environment," *IEEE/ASME Trans. Mechatron.* **29**(3), 2024 (1986–1996).
[19] P. Li, S. Wang, H. Yang and H. Zhao, "Trajectory tracking and obstacle avoidance for wheeled mobile robots based on EMPC with an adaptive prediction horizon," *IEEE Trans. Cybern.* **52**(12), 13536–13545 (2021).
[20] T. Zhu, J. Mao, L. Han, C. Zhang and J. Yang, "Real-time dynamic obstacle avoidance for robot manipulators based on cascaded nonlinear MPC with artificial potential field," *IEEE Trans. Ind. Electron.* **71**(7), 7424–7434 (2024).
[21] Y. Dai, J. Yu, C. Zhang, B. Zhan and X. Zheng, "A novel whale optimization algorithm of path planning strategy for mobile robots," *Appl. Intell.* **53**(9), 10843–10857 (2023).
[22] N. Wan, D. Xu and H. Ye, "Improved Cubic B-spline Curve Method for Path Optimization of Manipulator Obstacle Avoidance," **In:** *2018 Chinese Automation Congress (CAC)* (2018).
[23] E. Tring, S. A. Moosavi, M. Dipoppa and D. L. Ringach, "Contrast gain control is a reparameterization of a population response curve," *J. Neurophysiol.* **132**(5), 1359–1370 (2024).
[24] Y. Wang, S. Tang and M. Deng, "Modeling nonlinear systems using the tensor network B-spline and the multi-innovation identification theory," *Int. J. Robust. Nonlin.* **32**(13), 7304–7318 (2022).
[25] L. Wang and Y. Guo, "Speed adaptive robot trajectory generation based on derivative property of B-spline curve," *IEEE Robot. Autom. Lett.* **8**(4), 1905–1911 (2023).
[26] Ü. Dinçer and M. Çevik, "Improved trajectory planning of an industrial parallel mechanism by a composite polynomial consisting of Bézier curves and cubic polynomials," *Mech. Mach. Theory* **132**, 248–263 (2019).
[27] R. J. Webster III and B. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review. The," *Int. J. Robot. Res.* **29**(13), 1661–1683 (2010).
[28] R. Van Hoek, J. Ploeg and H. Nijmeijer, "Cooperative driving of automated vehicles using B-splines for trajectory planning," *IEEE Trans. Intell. Veh.* **6**(3), 594–560 (2021).
[29] L. Piegl and W. Tiller. *The NURBS Book* (Berlin, Springer Science & Business Media, 2012).
[30] B. Fowler and R. Bartels, "Constraint-based curve manipulation," *IEEE Comput. Graph.* **13**(5), 43–49 (1993).

Cite this article: Q. Chen, S. Zhao and Y. Qin (2026). "Trajectory planning for cable-driven continuum robots using model predictive control and basis spline optimization", Robotica **44**, 1551–1572. https://doi.org/10.1017/S0263574726103701

https://doi.org/10.1017/S0263574726103701 Published online by Cambridge University Press