frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 14 October 2022
DOI 10.3389/fnbot.2022.1014163

QPSO-MPC based tracking algorithm for cable-driven continuum robots

Qi Chen[1]*, Yanan Qin[1] and Gelun Li[2]

[1]School of Mechanical Engineering, University of Shanghai for Science and Technology, Shanghai, China, [2]Robotics State Key Laboratory, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China

Check for updates

OPEN ACCESS

EDITED BY
Lianbo Ma,
Northeastern University, China

REVIEWED BY
Gang Wang,
Harbin Engineering University, China
Zhiwei Yu,
Nanjing University of Aeronautics and Astronautics, China
Guoqiang Yin,
Northeastern University, China
Juan Fang,
Bern University of Applied Sciences, Switzerland
Zhaobing Liu,
Wuhan University of Technology, China

*CORRESPONDENCE
Qi Chen
chenqi8@usst.edu.cn

RECEIVED 08 August 2022
ACCEPTED 27 September 2022
PUBLISHED 14 October 2022

CITATION
Chen Q, Qin Y and Li G (2022) QPSO-MPC based tracking algorithm for cable-driven continuum robots. *Front. Neurorobot.* 16:1014163.
doi: 10.3389/fnbot.2022.1014163

Cable-driven continuum robots (CDCRs) can flexibly travel through narrow space for complex workspace tasks. However, it is challenging to design the trajectory tracking algorithm for CDCRs due to their nonlinear dynamic behaviors and cable hysteresis characteristics. In this contribution, a model predictive control (MPC) tracking algorithm based on quantum particle swarm optimization (QPSO) is designed for CDCRs to realize effective trajectory tracking under constraints. In order to make kinematic analysis of a CDCR, the forward and inverse mapping among actuation space, joint space and work space is analyzed by using the piecewise constant curvature method and the homogeneous coordinate transformation. To improve the performance of conventional MPC for complex tracking tasks, QPSO is adopted in the rolling optimization of MPC for its global optimization performance, robustness and fast convergence. Both simulation and operational experiment results demonstrate that the designed QPSO-MPC presents high control stability and trajectory tracking precision. Compared with MPC and particle swarm optimization (PSO) based MPC, the tracking error of QPSO-MPC is reduced by at least 43 and 24%, respectively.

KEYWORDS
cable-driven continuum robots, QPSO, MPC, control stability, trajectory tracking precision

Introduction

The continuum robots, designed by imitating the biological characteristics of snakes and elephant trunks, have multiple degrees of freedom (DOFs) (Guo et al., 2018; Guan et al., 2020). In comparison with conventional industrial robots, continuum robots are more suitable for working in the uncertain environment due to their deformable structures. The actuation mechanism of continuum robots includes flexible fluidic actuations (FFAs) (Garriga-Casanovas et al., 2018; Renda et al., 2020), shape memory alloy materials (SMAs) (Yang et al., 2018; Jiang et al., 2020), electroactive polymers (EAPs) (Chang et al., 2018; Bar-Cohen and Anderson, 2019) and cable-driven actuations (CDAs) (Jin et al., 2018; Hamida et al., 2021). Cable-driven continuum robots (CDCRs) are activated by changing link lengths, which make them easy to operate (Khomami and Najafi, 2021). Compared with other continuum robots, CDCRs have many advantages such as light weight, small moment of inertia and easy to realize variable stiffness control.

CDCRs are supported by passive joints and elastic backbones. By changing the lengths of the driving cables, CDCRs are able to achieve bending, stretching, twisting, grasping and other actions. Therefore, CDCRs are widely used in rescue operations (Kim et al., 2020; Takahashi et al., 2021), minimally invasive surgery (Miyasaka et al., 2015; Hwang et al., 2020), rehabilitation applications (Zhang et al., 2020; Shi et al., 2021) and human–robot interaction (Ma et al., 2021a,b). The trajectory tracking is essential for CDCRs to realize various tasks. However, it is challenging for CDCRs to track the desired trajectory accurately and smoothly due to their nonlinear response, parameters variation and non-rigid structures.

In recent years, a series of researches have emerged in motion control and trajectory tracking of CDCRs. A bio-inspired CDCR was designed in Li and Du (2013), where the kinematic model derived from the piecewise constant curvature method was used to control the motion of the robot. A segmented constant curvature method was employed to derive the forward and inverse kinematic model of a CDCR, which achieved simple shape and posture control of the CDCR (Liu et al., 2021). The kinematic model based controllers are well-understood and easy to apply. However, uncertainties incurred in exact kinematic calculations make kinematic-based tracking controllers suffer from accumulative errors in performing complex trajectory tracking tasks.

Researchers have developed adaptive trajectory tracking methods to solve the problem of modeling errors. Wang et al. (2020) designed an adaptive time-delay approach for the trajectory tracking of CDCRs, which enhances robustness of the controller. Li et al. (2020) introduced a pretension-based adaptive robust controller, which solved the problem of the cable slack in the process of trajectory tracking. However, the performance of adaptive controllers depends on a computational expensive process of updating the controller parameters.

Neural network control method is also used to achieve trajectory tracking by previous training samples. Tan et al. (2021) adopted neurodynamics method to track the preset trajectory of a CDCR. The end position control of a CDCR was implemented based on deep reinforcement learning in Wu et al. (2020). Since the CDCR has a nonlinear structure with continuous bending and infinite DOFs, the training samples are difficult to obtain and the huge amount of computation cannot meet the demand of complex tasks.

Researchers have also focused on sliding mode control (SMC) for its simple structure. Liu and Xia (2020) introduced a trajectory tracking control system based on SMC for a three DOFs CDCR. Abu et al. (2019) designed a multi-surface SMC to solve the nonlinear problem of CDCRs. However, sliding mode controllers produce buffeting due to the cable-related effects.

Fuzzy control calculates tracking strategy from expert knowledge, which has a strong anti-interference ability. Ba et al. (2021) implemented a fuzzy-logic feedback controller to track evenly distributed points. Compared with the piecewise constant curvature (PCC) based controller, the proposed controller provides a solution to the problem of failing to converge. A closed-loop fuzzy PID controller was proposed for position control of a CDCR in Xu et al. (2018). However, the fuzzy rules are subjective and difficult to obtain in practice due to the complex structure of CDCRs.

The shape-changing structures of CDCRs result in control constraints, which make it difficult to control CDCRs to follow the desired trajectory. Moreover, the error of state estimation and external interference can also affect the tracking precision. This makes conventional control methods difficult to achieve stable trajectory tracking for complex tasks. MPC is widely used for trajectory tracking tasks thanks to its ability of dealing with various constraints (Chu et al., 2021). The rolling optimization is essential for MPC based controllers, which can compensate for model uncertainties and disturbances. However, the nonlinear and time-varying dynamic characteristics of CDCRs make the rolling optimization design of MPC challenging. A nonlinear MPC based on the particle swarm optimization (PSO) was designed for a CDCR to track the expected trajectory (Amouri et al., 2022). However, PSO cannot ensure global optimal outputs due to the limitation of the search space of particles. The quantum particle swarm optimization (QPSO) cancels the particle moving direction attribute of PSO to increase the randomness of particle motion, so the update of particle positions is no longer constrained by the previous motion state and the problem of local minima is also alleviated. A QPSO based path optimization approach was used to achieve a smooth trajectory in a complex plane with obstacles, which can avoid local minimum and enhance the probability of searching global optimal control points (Dian et al., 2022). In this paper, a QPSO based MPC is designed to the trajectory tracking control for CDCRs, where the QPSO is used in the rolling optimization process of MPC. Compared with the traditional rolling optimization process of MPC, the QPSO algorithm adopts the average optimal position to improve the cooperation between particles, which can improve global search performance significantly. The proposed MPC controller achieves global optimal performance, robustness and fast convergence. The contribution of our work is that the proposed QPSO algorithm provides optimal control outputs of MPC to compensate the uncertainties caused by model mismatch, distortion and interference. Simulations and experiments show that the QPSO-MPC trajectory tracking controller has high tracking stability and accuracy in complex tasks.

This paper is arranged as follows: Section Modeling of the CDCR designs the structure and model of the CDCR. The QPSO-MPC trajectory tracking controller is designed in section QPSO-MPC trajectory tracking strategy. Section Simulation results presents the simulation and analysis results in detail. In Section Experiment results, three typical experiments are made and discussed. Finally, Section Conclusions draws conclusions.

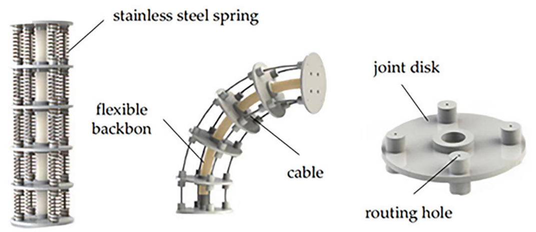

FIGURE 1
The structure of the CDCR.

Modeling of the CDCR

Mechanical structure of the CDCR

The structure of the designed CDCR is shown in Figure 1. The robot is composed of 6 joint disks (diameter, 4 cm; distance between joint disks, 6 cm) and a flexible backbone made of silicone rubber (diameter, 1 cm; length, 30 cm). Four high-strength polyethylene fiber cables are evenly distributed at an interval of 90° along the length of the backbone. Each cable is, respectively, attached to a servo driver and terminated at the last joint disk. The cables are pulled by the drivers to make the robot bend at a particular angle.

Kinematic model of the CDCR

It is difficult to derive the kinematic model of the under-actuated CDCR owing to its complex structure. Before analyzing the kinematic model of the robot, the following key assumptions need to be made and emphasized:

(1) Assume that the robot bends with a constant curvature.
(2) Assume that the flexible backbone of the robot is incompressible.
(3) The robot is driven by four evenly distributed traction cables that are also incompressible.

The CDCR is not a kind of direct driving robot, which is controlled by the length changes of the cables. The actuation space $\boldsymbol{L}$ is denoted as the length changes of the cables. The joint space $\boldsymbol{\Theta}$ is the set of the bending angle and the rotation angle, which is controlled by the length changes of the cables. The work space $\boldsymbol{W}$ represents the position of the end joint disk, which can be obtained from the joint space. Hence, the kinematic models of the robot can be derived from the mapping relationship among $\boldsymbol{L}$, $\boldsymbol{\Theta}$ and $\boldsymbol{W}$.

The single joint geometric model of the CDCR is shown in Figure 2A. The routing holes of the 4 driving cables are marked from 1 to 4. The base frame and end-effector frame of the CDCR are established in the center of the base joint and the end joint disk, respectively. The robot has a bending DOF and a rotation DOF, which are expressed by bending angle θ and rotation angle α, respectively. O is denoted as the center of the bending curve. α is the intersection angle between plane $X_0Z_0O_0$ and plane OO_0O_1, and θ is the center angle of the curvature arc.

Figure 2B provides a vertical view of the base section. Under the premise of the assumption of constant curvature, the radius of curvature for each driving cable is expressed as follows:

$$\zeta_{\,i}=\zeta\ \text{-}r\cos\alpha_{\,i} \tag{1}$$

where ζ stands for the radius of curvature measured from the center of the disk, ζ_i stands for the bending curvature radius of the i-th cable ($i = 1, 2, 3, 4$), r is the distance between the center of the disk and the center of each routing hole, and the rotation angle of the i-th cable is denoted as $\alpha_i= \alpha$ - (i - 1) π/2.

The 4 driving cables are parallel to each other. Multiplying the bending angle θ with Equation (1), the length change of the driving cables is derived as follows:

$$\Delta\ l_i=(\zeta\ \text{-}\zeta_{\,i})\theta=\mathrm{r}\,\theta\,\cos\alpha_i \tag{2}$$

where Δl_i represents the length change of the i-th driving cable. The relationship between joint space and work space can be derived by applying Equation (2) to the i-th and the i+1-th actuators and getting the quotient of them. Since $\alpha_{i+1} = \alpha_i$ - iπ/2, the joint space variables are obtained as follows:

$$\alpha=\arctan\left(\frac{\Delta\ l_{i+1}}{\Delta\ l_i}\right) \tag{3}$$

$$\theta=\frac{\Delta\ l_i}{r\cos\alpha} \tag{4}$$

The relative pose from the base frame to the end-effector frame is represented by the transformation matrix $\boldsymbol{T} \in \mathrm{R}^{4\times4}$, which is composed of the rotation matrix $\boldsymbol{R} \in \mathrm{R}^{3\times3}$ and the position vector $\boldsymbol{Q} \in \mathrm{R}^3$. Based on the geometric analysis method, the rotation matrix $\boldsymbol{R}$ is given by the consecutive rotations about the Z and Y axis as follows:

$$\boldsymbol{R}=\begin{bmatrix} c^2\alpha c\theta+s^2\alpha & c\alpha s\alpha c\theta-c\alpha s\alpha & c\alpha s\theta \\ c\alpha s\alpha c\theta-c\alpha s\alpha & s^2\alpha c\theta+c^2\alpha & s\alpha s\theta \\ -c\alpha s\theta & -s\alpha s\theta & c\theta \end{bmatrix} \tag{5}$$

where c and s represent cos and sin, respectively. The position vector $\boldsymbol{Q}$ is expressed as follows:

$$Q=\left[\tfrac{1}{\theta}c\alpha\,(1-c\theta)\ \ \tfrac{1}{\theta}s\alpha\,(1-c\theta)\ \ \tfrac{1}{\theta}s\theta\right]^{\mathrm{T}} \tag{6}$$

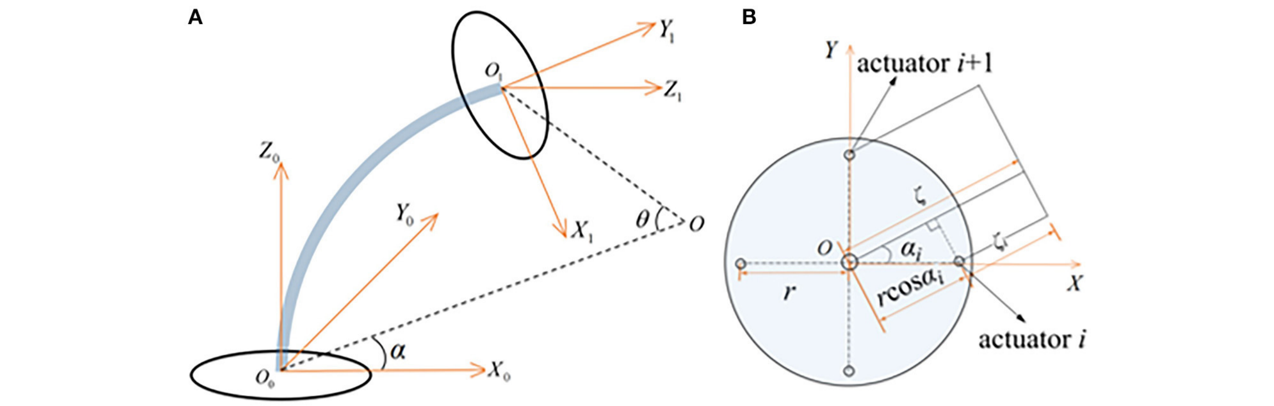

FIGURE 2
Kinematic model of the CDCR: **(A)** Single joint bending geometric; **(B)** Vertical view of the base section.

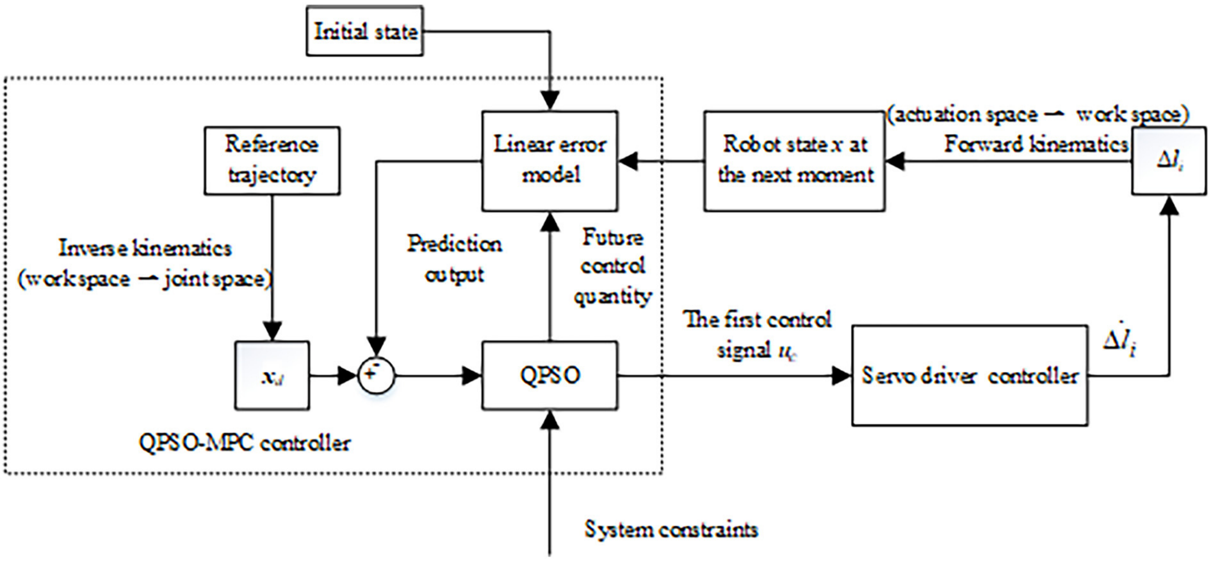

FIGURE 3
The block diagram of the trajectory tracking controller based on QPSO-MPC.

TABLE 1 The control parameters for the three controllers.

Controller	M_q	M_d	N	$Ni_{\max}$
QPSO-MPC	20	1	50	50
PSO-MPC	20	1	50	50
MPC	20	1	/	/

Then the pose transformation matrix $\boldsymbol{T}$ is expressed as follows:

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{R} & Q \\ 0 & 1 \end{bmatrix} \tag{7}$$

The relationship between joint space and work space can be derived by the transformation matrix $\boldsymbol{T}$. In order to solve the inverse kinematic transformation from work space to joint space, the end-effector transformation matrix $\boldsymbol{T}$ is presented as follows:

$$\boldsymbol{T} = \begin{bmatrix} w_x & e_x & b_x & q_x \\ w_y & e_y & b_y & q_y \\ w_z & e_z & b_z & q_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

where $\boldsymbol{w}$, $\boldsymbol{e}$, $\boldsymbol{b}$ are the unit vectors corresponding to the X_1, Y_1, Z_1 axes, respectively, and $\boldsymbol{Q}$ is the position vector. The

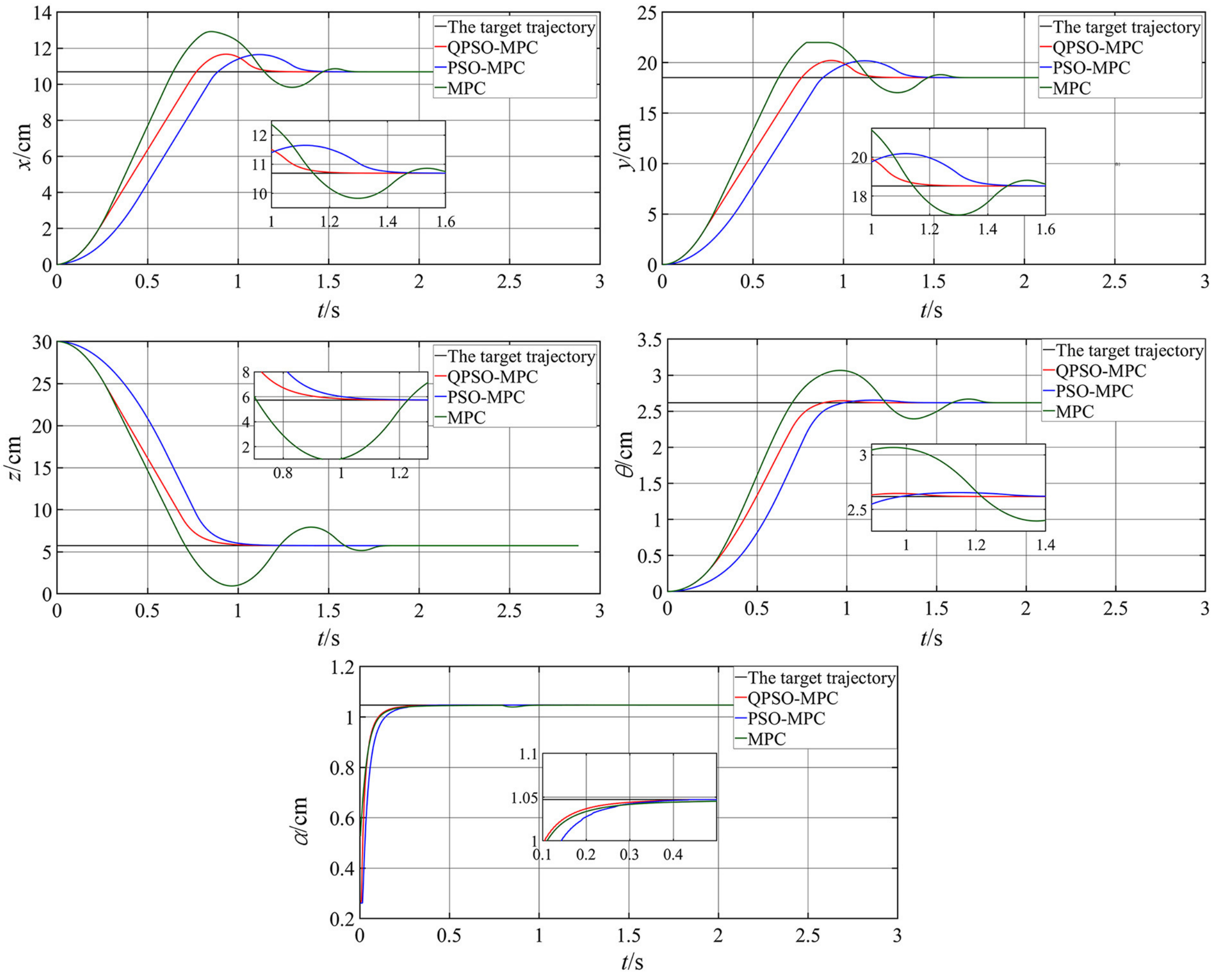

FIGURE 4
Simulation results of setpoint tracking.

bending angle θ and the rotation angle α of the robot is given by Equation (9)-(10):

$$\theta = \arccos(b_z) \tag{9}$$

$$\alpha = \arctan(\frac{q_y}{q_x}) \tag{10}$$

QPSO-MPC trajectory tracking strategy

The rolling optimization process of MPC is essential because it can repeatedly produce online control signals to compensate system uncertainties. To reduce the tracking error of MPC, QPSO is adopted in the rolling optimization process to improve tracking stability and accuracy. The block diagram of the designed tracking algorithm is shown in Figure 3, which is mainly composed of system constraints, the error model and the optimization process based on QPSO. According to the inverse kinematic equation from work space to joint space, the reference trajectory and the real trajectory are transformed into the joint space as input to the QPSO-MPC controller. The linear error model predicts the future position of the CDCR through mathematical description. The optimization process based on QPSO makes the CDCR has high control stability and tracking precision. Compared with the rolling optimization process of MPC, QPSO ensures an optimal feasible control value at every moment, which makes the robot track the target trajectory more stably and accurately.

Linear error model

The configuration state vector of the control system is $\boldsymbol{s} = [x, y, z, \theta, \alpha]^{\mathrm{T}}$, and the control vector is $\boldsymbol{h} =$

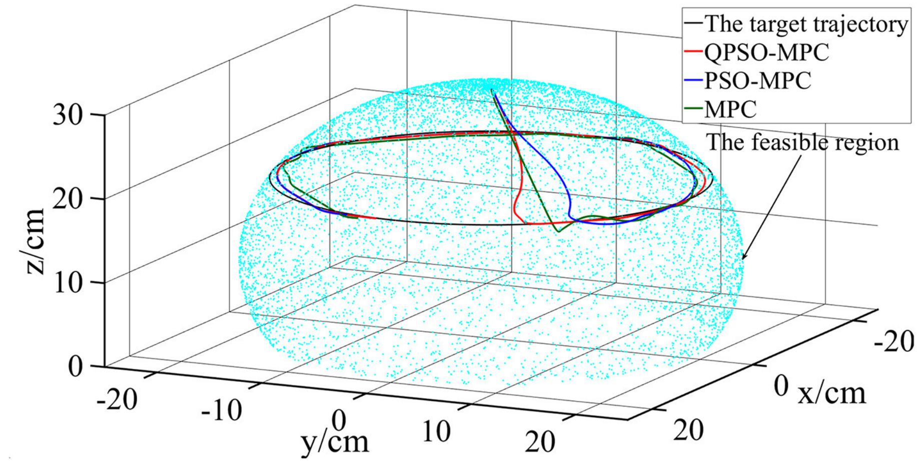

FIGURE 5
Simulation results of circular trajectory tracking.

$[\dot{\theta}, \dot{\alpha}]^{\mathrm{T}}$. The state of the control system can be expressed as follows:

$$\dot{s} = \boldsymbol{f}(s, h) \tag{11}$$

The velocity kinematics of the end-effector is described by taking the Jacobian matrix of the work space with respect to joint space. By taking the derivative of Equation (6), the velocity of the end-effector $\dot{V} = [\dot{x}, \dot{y}, \dot{z}]^{\mathrm{T}}$is obtained as follows:

$$\dot{V} = \boldsymbol{J}(\boldsymbol{\Theta})\dot{\boldsymbol{\Theta}} = \begin{bmatrix} -lc\alpha\frac{1-c\theta-s\theta\theta}{\theta^2} & -ls\alpha\frac{1-c\theta}{\theta} \\ -ls\alpha\frac{1-s\theta\theta-c\theta}{\theta^2} & -lc\alpha\frac{c\theta-1}{\theta} \\ -l\frac{c\theta\theta-s\theta}{\theta^2} & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \end{bmatrix} \tag{12}$$

where $\boldsymbol{J}(\boldsymbol{\Theta}) \in \boldsymbol{R}^{3\times 2}$ is the Jacobian matrix. According to Equation (12), the state space of the system is expressed as follows:

$$\begin{bmatrix} \dot{V} \\ \dot{\boldsymbol{\Theta}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}(\boldsymbol{\Theta})\dot{\boldsymbol{\Theta}} \\ \dot{\boldsymbol{\Theta}} \end{bmatrix} \tag{13}$$

By performing Taylor series expansion of Equation (11) at the reference trajectory $\boldsymbol{s}_d$ and ignoring the higher order terms (11) can be defined as follows:

$$\dot{\boldsymbol{s}} = \boldsymbol{f}(\boldsymbol{s}_d, \boldsymbol{h}_d) + \frac{\delta \boldsymbol{f}}{\delta \boldsymbol{s}}\bigg|_{\substack{s=s_d \\ h=h_d}} (\boldsymbol{s} - \boldsymbol{s}_d) + \frac{\delta \boldsymbol{f}}{\delta \boldsymbol{h}}\bigg|_{\substack{s=s_d \\ h=h_d}} (\boldsymbol{h} - \boldsymbol{h}_d) \tag{14}$$

The linear error model of the robot is obtained by subtracting Equation (11) from (14):

$$\dot{\tilde{\boldsymbol{s}}} = \boldsymbol{E}_t\tilde{\boldsymbol{s}} + \boldsymbol{F}_t\tilde{\boldsymbol{h}} \tag{15}$$

where $\dot{\tilde{\boldsymbol{s}}} = \dot{\boldsymbol{s}}_d - \dot{\boldsymbol{s}}, \tilde{\boldsymbol{s}} = \boldsymbol{s}_{\boldsymbol{d}} - \boldsymbol{s}, \tilde{\boldsymbol{h}} = \boldsymbol{h}_{\boldsymbol{d}} - \boldsymbol{h}, \boldsymbol{E}_t = \frac{\delta \boldsymbol{f}}{\delta \boldsymbol{s}}\bigg|_{\substack{s=s_d \\ h=h_d}}, \boldsymbol{F}_{\boldsymbol{t}} = \frac{\delta \boldsymbol{f}}{\delta \boldsymbol{h}}\bigg|_{\substack{s=s_d \\ h=h_d}}.$

Prediction model

To be adopted by the MPC controller, Equation (15) is discretized as follows:

$$\tilde{\boldsymbol{s}}(m+1) = \boldsymbol{E}_t\tilde{\boldsymbol{s}}(m) + \boldsymbol{F}_t\tilde{\boldsymbol{w}}(m) \tag{16}$$

In order to compute system outputs in the future time horizon, Equation (16) is converted to the state space form $\boldsymbol{\xi}(m|t) = \begin{bmatrix} \tilde{\boldsymbol{s}}(m|t) \\ \tilde{\boldsymbol{w}}(m-1|t) \end{bmatrix}$, and the new state space is as follows:

$$\boldsymbol{\xi}(m+1|t) = \tilde{\boldsymbol{E}}_t\boldsymbol{\xi}(m|t) + \tilde{\boldsymbol{F}}_t\Delta\boldsymbol{w}(m|t) \tag{17}$$

$$\boldsymbol{s}(m|t) = \tilde{\boldsymbol{G}}_t\boldsymbol{\xi}(m|t) \tag{18}$$

where $\tilde{\boldsymbol{E}}_t = \begin{bmatrix} \boldsymbol{E}_t & \boldsymbol{F}_t \\ 0_{i\times j} & \boldsymbol{I}_i \end{bmatrix}, \tilde{\boldsymbol{F}}_t = \begin{bmatrix} \boldsymbol{F}_t \\ \boldsymbol{I}_i \end{bmatrix}, \tilde{\boldsymbol{G}}_t = [\boldsymbol{G}_t \ \ 0], j = 5$ and $i = 2, \boldsymbol{I}_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Thus, the predictive output is described as follows:

$$\boldsymbol{Z}(t) = \boldsymbol{\Psi}_t\boldsymbol{\xi}(t|t) + \boldsymbol{\Gamma}_t\Delta\boldsymbol{W}(t) \tag{19}$$

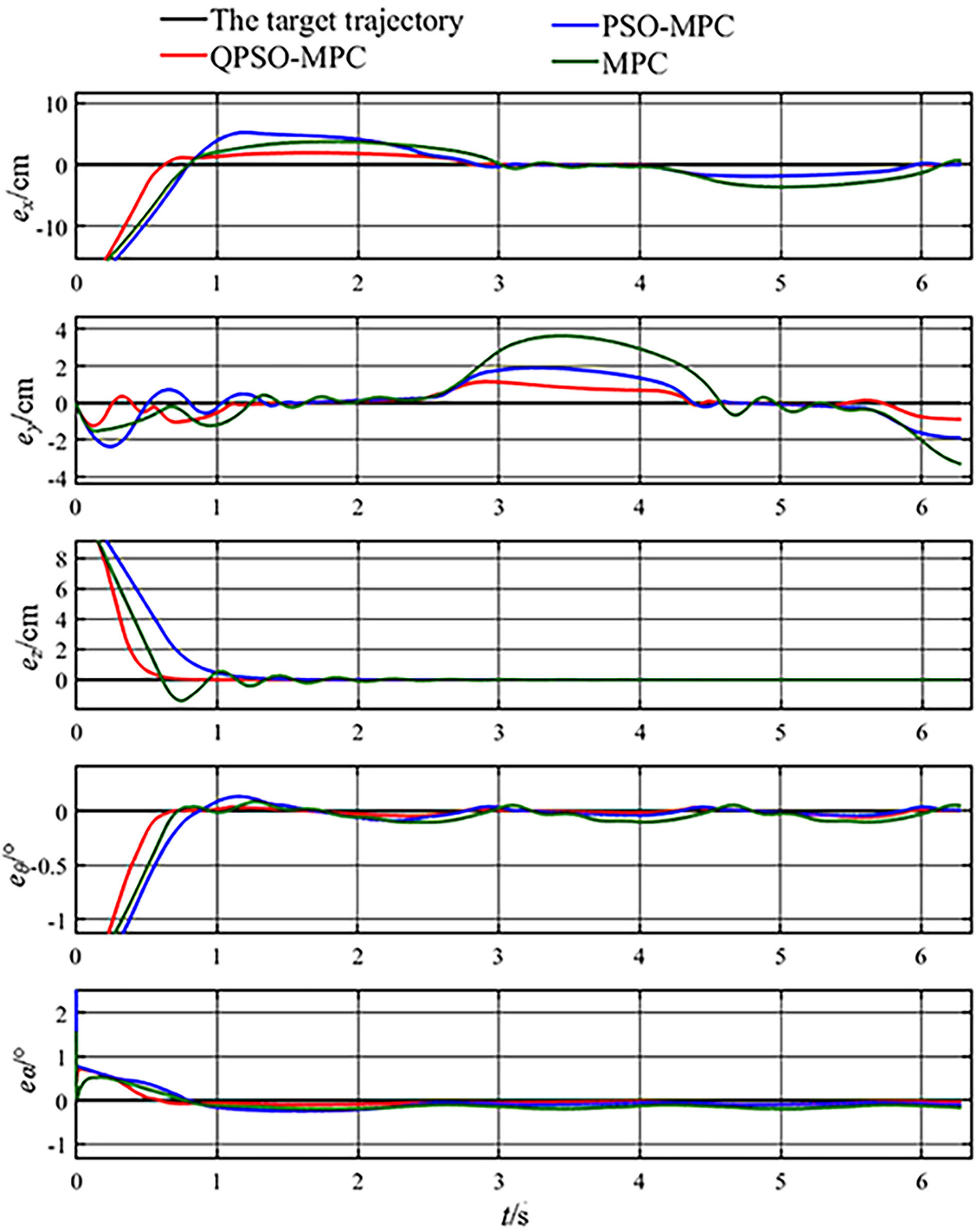

FIGURE 6
Tracking errors of circular trajectory tracking.

$$\text{where } \boldsymbol{Z}(t) = \begin{bmatrix} \boldsymbol{s}(t+1|t) \\ \vdots \\ \boldsymbol{s}(t+M_q|t) \end{bmatrix}, \boldsymbol{\Psi}(t) = \begin{bmatrix} \tilde{\boldsymbol{G}}_t\tilde{\boldsymbol{E}}_t \\ \vdots \\ \tilde{\boldsymbol{G}}_t\tilde{\boldsymbol{E}}_t^{M_q} \end{bmatrix}, \Delta \boldsymbol{W} = \begin{bmatrix} \Delta \boldsymbol{w}(t|t) \\ \vdots \\ \Delta \boldsymbol{w}(t+M_d|t) \end{bmatrix},$$

$$\boldsymbol{\Gamma}_t = \begin{bmatrix} \tilde{\boldsymbol{G}}_t\tilde{\boldsymbol{F}}_t & 0 & 0 & 0 \\ \tilde{\boldsymbol{G}}_t\tilde{\boldsymbol{E}}_t\tilde{\boldsymbol{F}}_t & \tilde{\boldsymbol{G}}_t\tilde{\boldsymbol{F}} & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\boldsymbol{G}}_t\tilde{\boldsymbol{E}}_t^{M_q-1}\tilde{\boldsymbol{F}} & \tilde{\boldsymbol{G}}_t\tilde{\boldsymbol{E}}_t^{M_q-2}\tilde{\boldsymbol{F}}_t & \cdots & \tilde{\boldsymbol{G}}_t\tilde{\boldsymbol{E}}_t^{M_q-M_d-1}\tilde{\boldsymbol{F}}_t \end{bmatrix}.$$

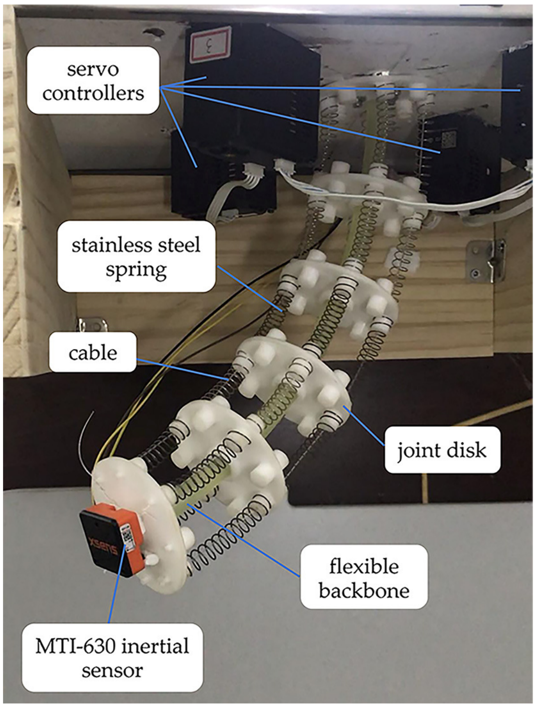

FIGURE 7
Experimental platform of the CDCR.

M_q and M_d are the predictive domain and control domain, respectively.

QPSO rolling optimization

Objective function and constraints

By minimizing the deviation between the reference trajectory and the current trajectory, the CDCR achieves the optimal trajectory that tracks the desired path. The excessive tracking errors are penalized by the objective function.

$$\boldsymbol{F}(m) = \sum_{j=1}^{M_q} \left\| \boldsymbol{s}(m+j|t) - \boldsymbol{s}_{ref}(m+j|t) \right\|_{\boldsymbol{B}}^2 + \sum_{j=1}^{M_d-1} \left\| \Delta \boldsymbol{w}(m+j|t) \right\|_{\boldsymbol{C}}^2 + \rho\varepsilon^2 \tag{20}$$

where $\boldsymbol{B}$, $\boldsymbol{C}$ represent the state increment weighting matrix and the control increment weighting matrix, respectively, $m+j|t$ is the state quantity of $m+j$ step at time t, $\Delta \boldsymbol{w}$ is the control increment, ρ represents weight coefficient, ε stands for relaxation factor.

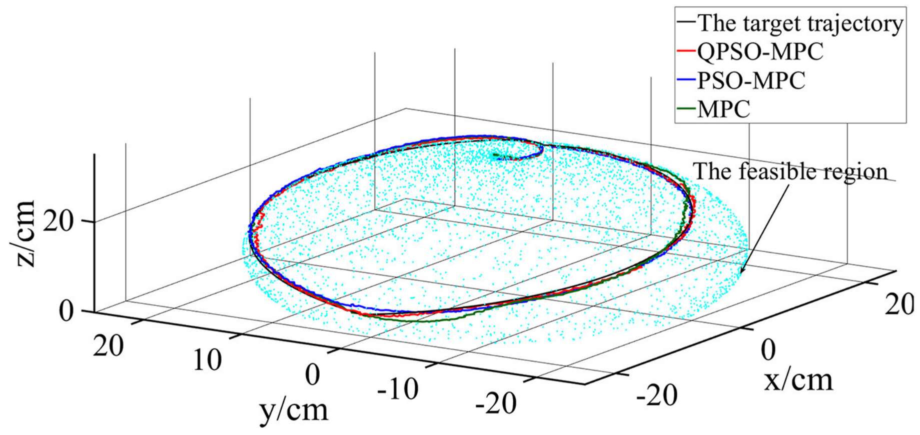

FIGURE 8
Experiment results of the cardioid trajectory tracking.

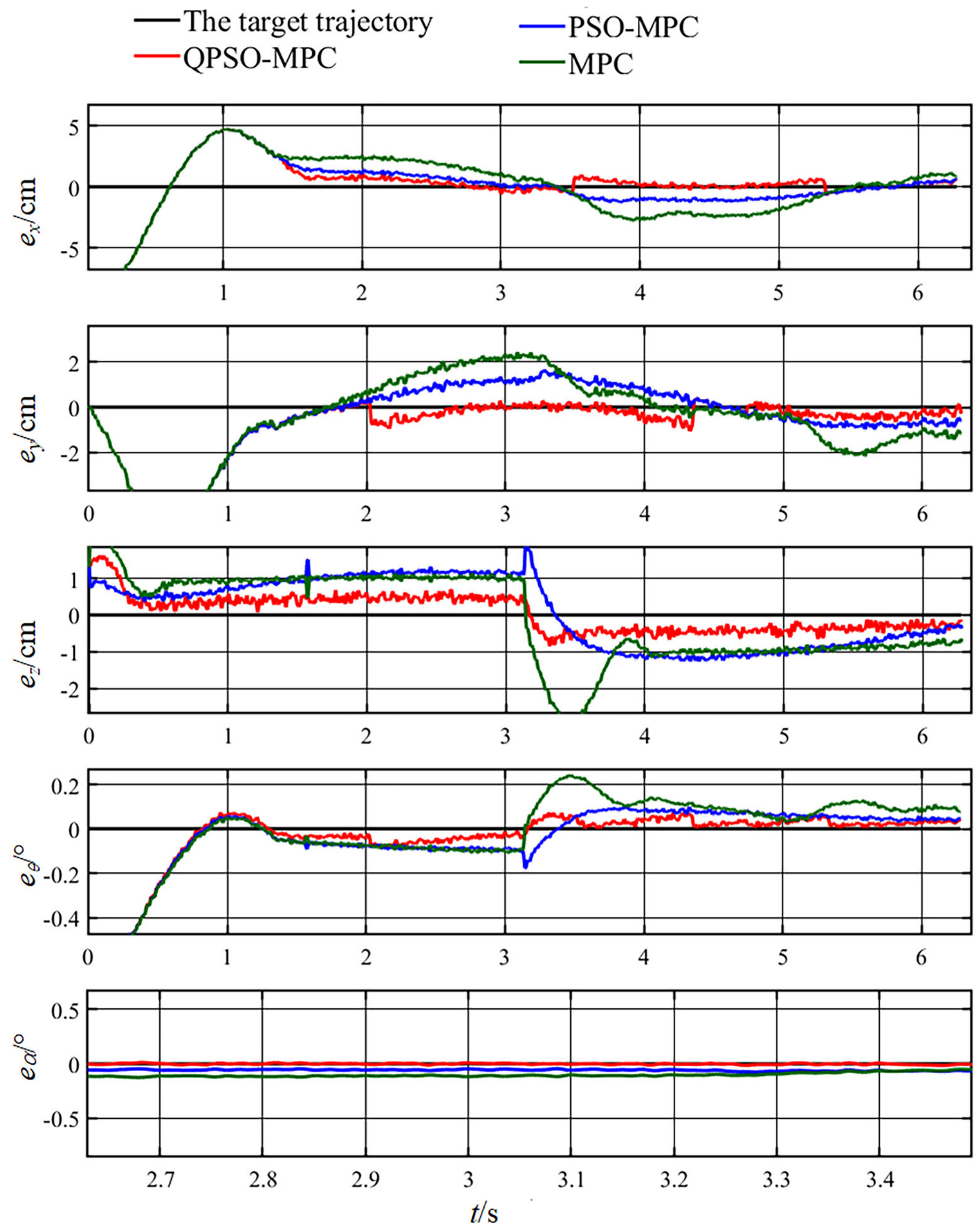

FIGURE 9
Tracking errors of the cardioid trajectory tracking.

The control quantity and control increment constraints need to be appended to avoid the sharp mutation of the speed when the servo controllers pull the cables, which are defined as follows:

$$\boldsymbol{w}_{\min}(t+m) \leq \boldsymbol{w}(t+m) \leq \boldsymbol{w}_{\max}(t+m) \tag{21}$$

$$\Delta\boldsymbol{w}_{\min}(t+m) \leq \Delta\boldsymbol{w}(t+m) \leq \Delta\boldsymbol{w}_{\max}(t+m) \tag{22}$$

where $m = 0, 1, \ldots, M_d$-1.

In addition, the movement of the robot should be limited within its feasible region, which is defined as follows:

$$\boldsymbol{s}_{\min}(t+m) \leq \boldsymbol{s}(t+m) \leq \boldsymbol{s}_{\max}(t+m) \tag{23}$$

where $m = 0, 1, \ldots, M_q$-1.

Therefore, the objective function is as follows:

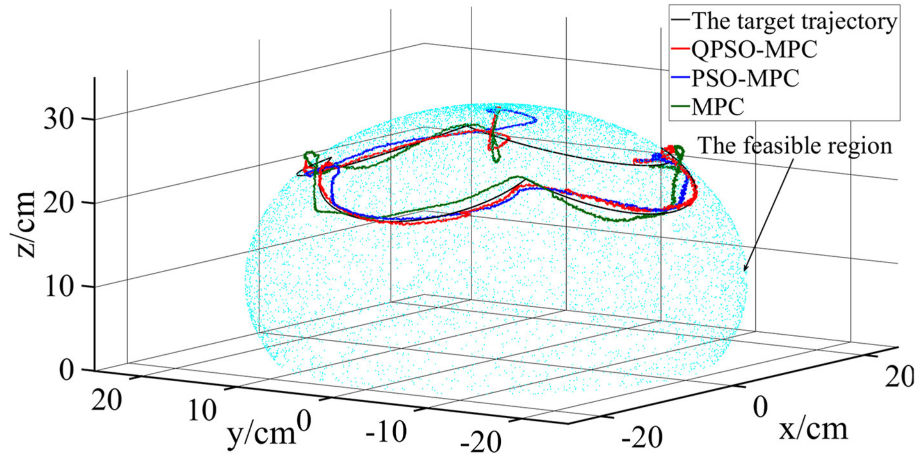

FIGURE 10
Experiment results of the polyline trajectory tracking.

$$\boldsymbol{F}(\boldsymbol{\xi}(t), \boldsymbol{w}(t-1), \Delta \boldsymbol{W}(t)) = \Delta \boldsymbol{W}(t)\boldsymbol{H}_t\Delta \boldsymbol{W}(t)^{\mathrm{T}} + \boldsymbol{L}_t\Delta \boldsymbol{W}(t)^{\mathrm{T}} \tag{24}$$

where the positive definite matrix $\boldsymbol{H}_t = \begin{bmatrix} \boldsymbol{\Gamma}_{\mathrm{t}}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{\Gamma}_t + \boldsymbol{C} & 0 \\ 0 & \rho \end{bmatrix}$ is used to punish the control change rate. $\boldsymbol{L}_t = \begin{bmatrix} 2\boldsymbol{D}(t)^{\mathrm{T}}\boldsymbol{B}\boldsymbol{\Gamma}_t & 0 \end{bmatrix}$, where $D(t)$ is the tracking error.

Sort out Equations (21)–(24), the optimization process under constraints that need to be solve by QPSO can be presented as follows:

$$\begin{aligned} &\min_{\Delta \boldsymbol{W}(t)} \ \Delta \boldsymbol{W}(t)\boldsymbol{H}_t\Delta \boldsymbol{W}(t)^{\mathrm{T}} + \boldsymbol{L}_t\Delta \boldsymbol{W}(t)^{\mathrm{T}} \\ &\Delta \boldsymbol{W}_{\mathrm{min}} \le \Delta \boldsymbol{W}_t \le \Delta \boldsymbol{W}_{\mathrm{max}} \\ s.t.\ &\boldsymbol{W}_{\mathrm{min}} \le \Delta \boldsymbol{W}_t + \boldsymbol{W}_t \le \boldsymbol{W}_{\mathrm{max}} \\ &\boldsymbol{Z}_{\mathrm{min}} \le \boldsymbol{Z}_{\mathrm{t}} \le \boldsymbol{Z}_{\mathrm{max}} \end{aligned} \tag{25}$$

QPSO algorithm

Classical PSO algorithm only allows the optimal particles to appear in a limited space for lack of randomness, which can lead to local minima. In contrast, QPSO allows particles to appear anywhere in the whole space with a certain probability, which improves the performance of optimization. Besides, there is only one parameter of QPSO need to be tuned, which makes it easy to be implemented. In the QPSO model, particles update their positions in accordance with the following equations:

$$mbest_y(t) = \sum_{x=1}^{N} \frac{P_{xy}(t)}{\boldsymbol{N}} = \left(\sum_{x=1}^{N} \frac{P_{x1}(t)}{\boldsymbol{N}}, \ldots, \sum_{x=1}^{N} \frac{P_{xe}(t)}{\boldsymbol{N}} \right) \tag{26}$$

$$PP_{xy}(t) = \sigma_y(t)P_{xy}(t) + [1 - \sigma_y(t)]P_{gy}(t) \tag{27}$$

$$I_{xy}(t+1) = PP_{xy}(t) \pm \eta \left| mbest_y(t) - I_{xy}(t) \right| \ln[\frac{1}{ra_{xy}(t)}] \tag{28}$$

where $mbest(t)$ represents the average of all the particles' optimal positions, vector $P_x(t) = (P_{x1}(t), P_{x2}(t), \ldots, P_{xe}(t))$ represents the optimal individual position of particles, vector $P_g(t) = (P_{g1}(t), P_{g2}(t), \ldots, P_{gy}(t))$ represents the global best position of particles, vector $I_{xy}(t) = (I_{x1}(t), I_{x2}(t), \ldots, I_{xe}(t))$ represents the particle position, $PP_{xy}(t)$ represents the stochastic point between $P_x(t)$ and $P_g(t)$, N *is* the total number of particles and e is the dimension of particles. Random parameters σ and $ra(t)$ are distributed uniformly in [0, 1]. η represents shrinkage and expansion coefficient, which is used to adjust the convergence speed of the controller. The QPSO algorithm is described as follows:

(1) Initialize the particle swarm, where I_{xy} is a random value.
(2) Calculate $mbest(t)$ by Equation (26).
(3) Calculate the loss function value $F(I_{xy}(t+1))$ according to Equation (25). Update $P_x(t)$ and $P_g(t)$.
(4) Update the new position $I_{xy}(t+1)$ of all particles according to Equation (28).
(5) Let $t = t+1$, and skip back to step 2 until the algorithm reaches the maximum number of iterations.

By derivatizing Equation (2), the tensile speed of the driving cables is computed as follows:

$$\Delta \dot{l}_i = d(\dot{\theta}\cos(\Gamma - \alpha) - \theta\dot{\varphi}\sin(\Gamma - \alpha)) \tag{29}$$

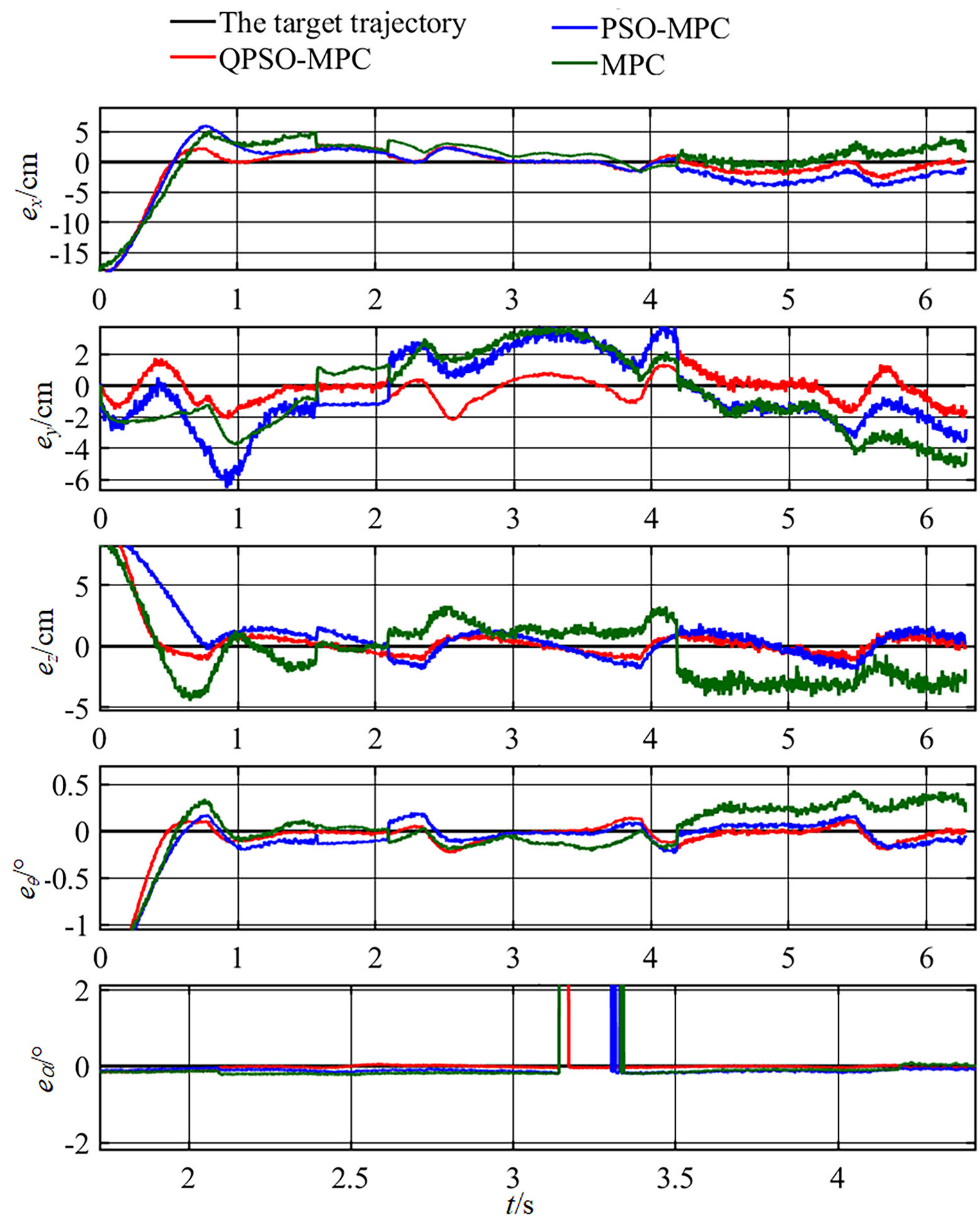

FIGURE 11
Tracking errors of the polyline trajectory tracking.

where $\Delta\dot{l}_i$ stands for the driving speed of i-th cable, $\Gamma = (n - 1)\pi/2$ is the division angle.

Simulation results

A set of simulations have been made to evaluate the effectiveness of the proposed QPSO-MPC controller in comparison with a conventional MPC controller and a PSO-MPC controller. For simulation studies, the initial positions of the robot are set to the natural relaxation state, where the initial position vector is set as $[0, 0, l, 0, 0]^T$ and the flexible support length l is 30 cm. The parameters are set as Table 1, where $Ni_{\max}$ means the maximum number of iterations. In the first simulation, the static performances of the three controllers are verified by the setpoint trajectory tracking. The static target position of the robot is set as $[10.69, 18.52, 5.73, 5\pi/6, \pi/3]^T$ and the sampling period is 0.05 s. The simulation results of the three controllers are shown in Figure 4.

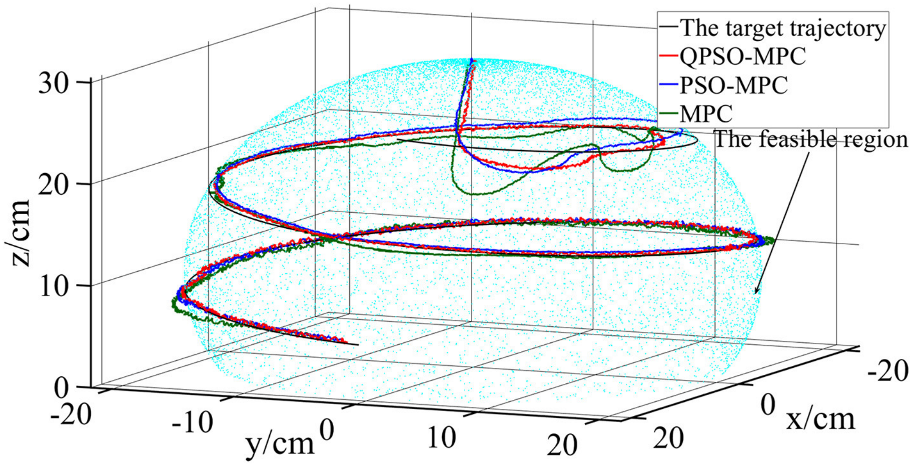

FIGURE 12
Experiment results of the spiral trajectory tracking.

It can be seen that all the three controllers track the desired position accurately within a certain time period and make the robot deformed into the desired shape. However, the QPSO-MPC controller yields smaller overshoots and more accurate tracking performance.

In the second simulation, the dynamic trajectory tracking performance is evaluated by tracking the circular-shaped trajectory, which is defined as follows:

$$\begin{cases} x = l(1 - \cos\theta)\cos\alpha/\theta \\ y = l(1 - \cos\theta)\sin\alpha/\theta \\ z = l\sin\theta/\theta \\ \theta = \pi/2 \\ \alpha = t \quad t \in [0, 2\pi] \end{cases} \tag{30}$$

The simulation results of circular trajectory tracking are shown in Figure 5. The tracking errors of circular trajectory tracking are given in Figure 6. The tracking errors of the QPSO-MPC are smaller than those of other controllers. The tracking errors of QPSO-MPC, PSO-MPC and MPC controllers are 2.64, 3.73, and 4.16 cm, respectively. The response time of QPSO-MPC, PSO-MPC and MPC controllers are 3.0, 3.1, and 3.5 s, respectively. The simulation results verify the effectiveness of the control sequences generated by the QPSO-MPC in suppressing the system overshoot, reducing oscillation, and improving convergence.

Experiment results

Experimental implementations are utilized to verify the trajectory tracking effectiveness of the QPSO-MPC controller in comparison with the PSO-MPC and the conventional MPC controller. The CDCR is shown in Figure 7, which consists of a flexible backbone (diameter, 1 cm; length, 30 cm), a MTI-630 inertial sensor, four anti-winding driving cables (diameter, 0.1 cm), four servo motors with reducers (servo motors type, QDD Plus-NU80-6; reduction ratio, 6:1; maximum torque, 6 N·m; rated full-load speed, 200 rpm), six joint disks (diameter, 4 cm; distance between joint disks, 6 cm) and twenty stainless steel springs. The servo motors are used in Position Mode. The flexible backbone provides bending stiffness for the robot. Each joint disk, mounted on the flexible backbone, has eight uniformly distributed cable holes.

The stainless steel springs (diameter, 1 cm) are used to maintain the external contour of the robot. The servo controllers, using CAN bus communication, are evenly distributed on the operating chassis. The MTI-630 inertial sensor is an industrial inertial measurement unit (IMU) and the direction error is <0.5°, which can produce an accurate measurement of the attitude and position of the robot end-effector. In order to ensure the trajectory tracking performance of the CDCR, the maximal load that the end-effector can support is 300 g. A higher maximal load of the CDCR can be achieved by increasing backbone strength, spring stiffness, cable strength, and motor power. The double-integration of the acceleration measured from the MTI-630 sensor is processed by the Extended Kalman Filter, which can reduce the estimated error of position and provide reliable motion data for trajectory tracking. Besides, when the output of the servo motors is zero, the velocity should also be zero, which can be used for the Extended Kalman Filter to reduce zero-shift effect.

In this paper, the tracking performances of QPSO-MPC, PSO-MPC and MPC controllers are assessed with respect to three typical trajectory tracking tasks: the cardioid curve, the polyline curve and the spiral curve, which can be presented as (31), (32), and (33), respectively.

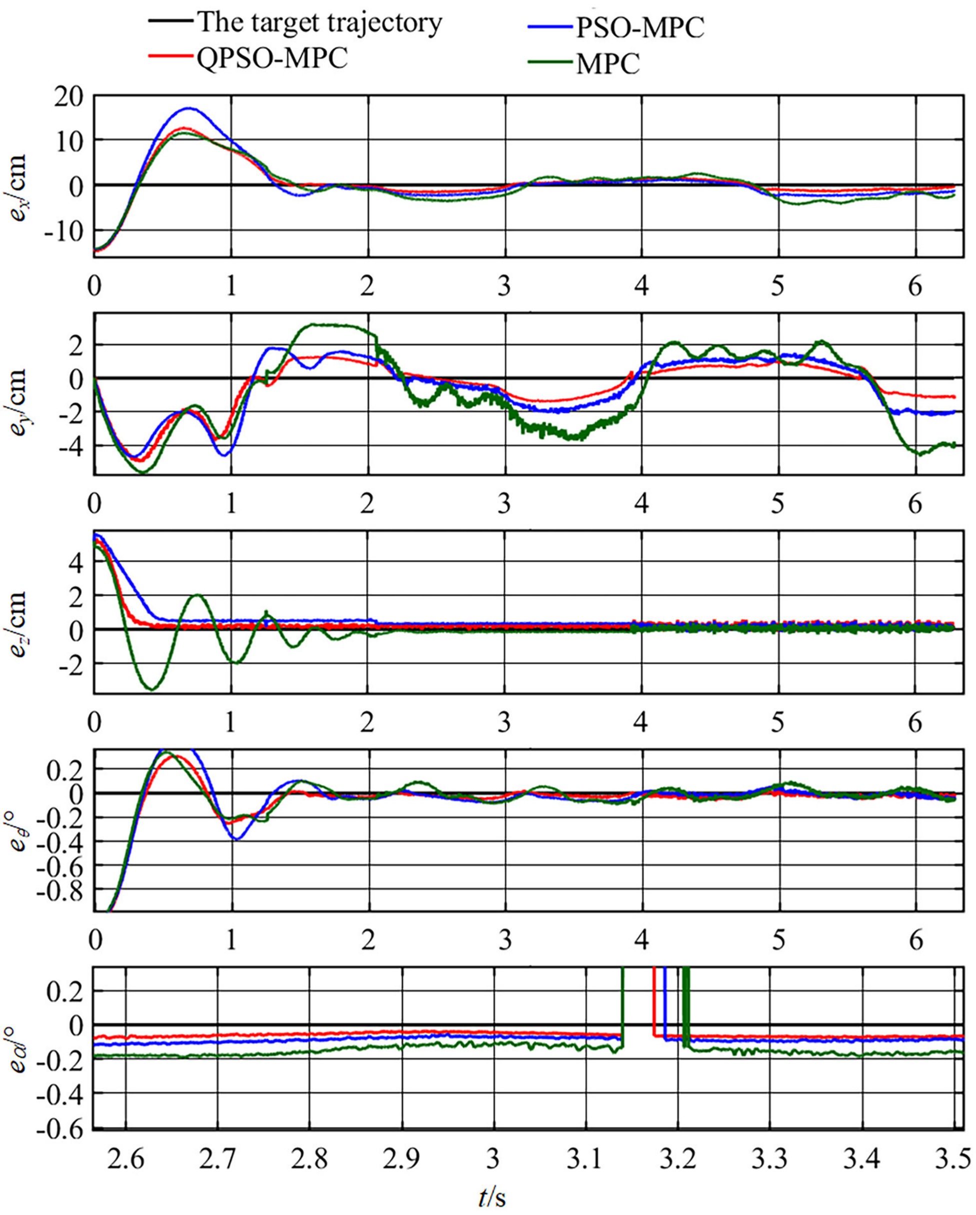

FIGURE 13
Tracking errors of the spiral trajectory tracking.

$$\begin{cases} x = l(1 - \cos\theta)\cos\alpha/\theta \\ y = l(1 \text{ - } \cos\theta)\sin\alpha/\theta \\ z = l\sin\theta/\theta \\ \theta = \begin{cases} \pi/6 + 2\pi \times t/3 \; t \in (0, \pi] \\ 3\pi/2 - 2\pi \times t/3 \; t \in (\pi, 2\pi] \end{cases} \\ \alpha = t \;\; t \in (0, 2\pi] \end{cases} \tag{31}$$

$$\begin{cases} x = l(1 \text{ - } \cos\theta)\cos\alpha/\theta \\ y = l(1 \text{ - } \cos\theta)\sin\alpha/\theta \\ z = l\sin\theta/\theta \\ \theta = \begin{cases} |1.5 \times l\sin(t)| & t \in [\pi/4, 3\pi/4] \cup [5\pi/4, 7\pi/4] \\ |1.5 \times l\cos(t)| & other \end{cases} \\ \alpha = t \;\; t \in (0, 2\pi] \end{cases} \tag{32}$$

$$\begin{cases} x = l(1 - \cos\theta)\cos\alpha/\theta \\ y = l(1 - \cos\theta)\sin\alpha/\theta \\ z = l\sin\theta/\theta \\ \theta = \pi t/4 + \pi/3 \; t \in (0, 2\pi] \\ \alpha = 2t \; t \in (0, 2\pi] \end{cases} \tag{33}$$

The control parameters of all the controllers for the experiments are set as follows in Table 1. The sampling periods of the three tasks are 0.05, 0.02, and 0.01 s, respectively. The tracking results and tracking error curves of the cardioid trajectory are shown in Figures 8, 9, respectively. The tracking results and tracking error of the polyline trajectory are shown in Figures 10, 11, respectively. The tracking results and tracking error of the spiral trajectory are shown in Figures 12, 13, respectively. It is shown that all the controllers could control the robot to move along the reference trajectories within a certain time. However, the QPSO-MPC controller fits the reference trajectory best in all the three tracking tasks.

The tracking errors are quantified as the mean value of the Euclidean distance between the real position and the target trajectory, which are shown in Table 2. In each task, the QPSO-MPC controller achieves the lowest tracking errors. The average tracking errors of QPSO-MPC, PSO-MPC and MPC controllers in all the three tasks are 2.61, 3.25, and 3.74 cm respectively, and the standard deviations are 0.44, 0.48, and 0.62 cm, respectively. Compared with the PSO-MPC and the MPC controllers, the average tracking errors of the QPSO-MPC is reduced by 24.52 and 43.30%, respectively, which further proves the tracking effectiveness of QPSO-MPC outperforms the other controllers. The experiments show that QPSO-MPC controller achieves accurate and stable tracking performance in complex trajectory tracking tasks. The proposed QPSO-MPC controller achieves substantial improved performance by comparing with MPC and PSO-MPC controllers because it ensures the system get a global optimal output at every moment.

Conclusions

In this paper, a QPSO-MPC is applied to trajectory tracking tasks of CDCRs. The kinematic model of the CDCR is built based on the piecewise constant curvature assumptions, which simplifies the mathematical model of the CDCR and reduces the computation of the control system. The QPSO is applied to the rolling optimization process of the MPC based controller, which guarantees stable and accurate trajectory tracking under constraints. The QPSO provides optimal control outputs of MPC to compensate various uncertainties. Moreover, the QPSO solves the local minima problem of PSO algorithm.

The effectiveness of the proposed QPSO-MPC in typical tracking tasks is demonstrated by both simulations and experiments. Compared with PSO-MPC and MPC controllers, the QPSO-MPC algorithm shows greatly improved performance in three typical tracking tasks. It has been proved that the QPSO-MPC controller is more suitable for controlling CDCRs to track complex trajectories. Future studies will focus on the extended applications of the QPSO-MPC algorithm for CDCRs with different structures, such as CDCRs with multi-backbone structures and CDCRs with nested backbone.

TABLE 2 The tracking errors of the experiments.

Controller	Error (cm)			Average error (cm)	Standard deviation (cm)
	Task 1	Task 2	Task 3		
QPSO-MPC	2.19	2.57	3.07	2.61	0.44
PSO-MPC	2.79	3.22	3.74	3.25	0.48
MPC	3.05	3.91	4.26	3.74	0.62

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

QC and YQ conceived the study and put forward the methodology. YQ and GL performed the data collection and pre-processing, and reviewed and edited the manuscript. QC carried out the software for the experiments and wrote the first draft of the manuscript. All authors read and agreed to the published version of the manuscript.

Funding

This research was funded by National Natural Science Foundation of China Grant Numbers 51975565 and 52127813, Science and Technology Innovation Plan of Shanghai Science and Technology Commission Grant Number 20DZ1206700, and Natural Science Foundation of Liaoning Province Grant Number 2020-KF-22-12.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Abu, A. A., Khoo, S., and Norton, M. (2019). Multi-surface sliding mode control of continuum robots with mismatched uncertainties. *Meccanica*. 54, 2307–2316. doi: 10.1007/s11012-019-01072-6

Amouri, A., Cherfia, A., Merabti, H., and Laib, D. L. Y. (2022). Nonlinear model predictive control of a class of continuum robots using kinematic and dynamic models. *FME Trans*. 50, 339–350. doi: 10.5937/fme2201350A

Ba, W. M., Dong, X., Mohammad, A., Wang, M. F., Axinte, D., Norton, A., et al. (2021). Design and validation of a novel fuzzy-logic-based static feedback controller for tendon-driven continuum robots. *IEEE-ASME Trans. Mech*. 26, 3010–3021. doi: 10.1109/TMECH.2021.3050263

Bar-Cohen, Y., and Anderson, I. A. (2019). Electroactive polymer (EAP) actuators—background review. *Mech. Soft Mater*. 1, 1–14. doi: 10.1007/s42558-019-0005-1

Chang, L., Liu, Y., and Yang, Q. (2018). Ionic electroactive polymers used in bionic robots: a review. *J. Bionic. Eng*. 15, 765–782. doi: 10.1007/s42235-018-0065-1

Chu, Z. Z., Wang, D., and Meng, F. (2021). An adaptive RBF-NMPC architecture for trajectory tracking control of underwater vehicles. *Machines*. 9, 105. doi: 10.3390/machines9050105

Dian, S. Y., Zhong, J. N., Guo, B., Liu, J. X., and Guo, R. (2022). A smooth path planning method for mobile robot using a BES-incorporated modified QPSO algorithm. *Exp. Syst. Appl*. 208:118256. doi: 10.1016/j.eswa.2022.118256

Garriga-Casanovas, A., Collison, I., and Baena, F. R. (2018). Toward a common framework for the design of soft robotic manipulators with fluidic actuation. *Soft Robot*. 5, 622–649. doi: 10.1089/soro.2017.0105

Guan, Q., Sun, J., and Liu, Y. (2020). Novel bending and helical extensile/contractile pneumatic artificial muscles inspired by elephant trunk. *Soft Robot*. 7, 597–614. doi: 10.1089/soro.2019.0079

Guo, X., Zhu, W., and Fang, Y. C. (2018). Guided motion planning for snake-like robots base on geometry mechanics and HJB equation. *IEEE Trans. Ind. Electron*. 66, 7120–7130. doi: 10.1109/TIE.2018.2883278

Hamida, I. B., Laribi, M. A., Mlika, A., Romdhane, L., Zeghloul, S., Carbone, G., et al. (2021). Multi-objective optimal design of a cable driven parallel robot for rehabilitation tasks. *Mech. Mach. Theory*. 156, 104141. doi: 10.1016/j.mechmachtheory.2020.104141

Hwang, M., Thananjeyan, B., and Paradis, S. (2020). Efficiently calibrating cable-driven surgical robots with RGBD fiducial sensing and recurrent neural networks. *IEEE Robot. Autom. Let*. 5, 5937–5944. doi: 10.1109/LRA.2020.3010746

Jiang, S., Chen, B., and Qi, F. (2020). A variable-stiffness continuum manipulator by an SMA-based sheath in minimally invasive surgery. *Int. J. Med. Robot. Comp*. 16, 1–11. doi: 10.1002/rcs.2081

Jin, X., Prado, A., and Agrawal, S. K. (2018). Retraining of human gait-are lightweight cable-driven leg exoskeleton designs effective? *IEEE Trans. Neur. Sys. Reh*. 26, 847–855. doi: 10.1109/TNSRE.2018.2815656

Khomami, A. M., and Najafi, F. (2021). A survey on soft lower limb cable-driven wearable robots without rigid links and joints. *Robot. Auton. Syst*. 144, 103846. doi: 10.1016/j.robot.2021.103846

Kim, J., Seo, J., and Park, S. (2020). "Design and implementation of hydraulic-cable driven manipulator for disaster response operation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Paris, France*. Piscataway, NJ: IEEE.

Li, B., Yan, L., Zhang, L. Q., and Gerada, C. (2020). Adaptive robust control of the cable driven system for position tracking. In Proceedings of the 2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA), Kristiansand, Norway. doi: 10.1109/ICIEA48937.2020.9248385

Li, Z., and Du, R. (2013). Design and analysis of a bio-inspired wire-driven multi-section flexible robot. *Int J Adv Robot. Syst*. 10, 1–11. doi: 10.5772/56025

Liu, P., and Xia, Z. (2020). "Trajectory tracking control of a 3-DOF planar wire-driven robot," in *Proceedings of the 2020 5th International Conference on Control: Robotics and Cybernetics (CRC), Wuhan, China* (Piscataway, NJ: IEEE).

Liu, Z., Zhang, X., Cai, Z., Peng, H., and Wu, Z. (2021). Real-time dynamics of cable-driven continuum robots considering the cable constraint and friction effect. *IEEE Robot. Autom. Let*. 6, 6235–6242. doi: 10.1109/LRA.2021.3086413

Ma, L. B., Li, N., Guo, Y. N., Huang, M., Yang, S. X., Wang, X. W., et al. (2021a). *Learning to* optimize: Reference vector reinforcement learning adaption to constrained many-objective optimization of industrial copper burdening system. *IEEE Trans. Cybern*. doi: 10.1109/TCYB.2021.3086501

Ma, L. B., Wang, X. Y., Wang, X. W., Wang, L., Shi, L., Huang, M. T. C. D. A., et al. (2021b). Truthful combinatorial double auctions for mobile edge computing in industrial internet of things. *IEEE Trans Mobile Comput*. 2021, 3064314. doi: 10.1109/TMC.2021.3064314

Miyasaka, M., Matheson, J., Lewis, A., and Hannaford, B. (2015). "Measurement of the cable-pulley Coulomb and viscous friction for a cable-driven surgical robotic system," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robot and Systems (IROS), Hamburg, Germany*. Available online at: https://dl.acm.org/doi/abs/10.1109/IROS.2015.7353464

Renda, F., Armanini, C., and Lebastard, V. A. (2020). geometric variable-strain approach for static modeling of soft manipulators with tendon and fluidic actuation. *IEEE Robot. Autom. Lett*. 5, 4006–4013. doi: 10.1109/LRA.2020.2985620

Shi, K., Song, A. G., Li, Y., Li, H. J., Chen, D. P., Zhu, L. F., et al. (2021). A cable-driven three-DOF wrist rehabilitation exoskeleton with improved performance. *Front. Neurorobot*. 15:664062. doi: 10.3389/fnbot.2021.664062

Takahashi, T., Tadakuma, K., and Watanabe, M. (2021). Eversion robotic mechanism with hydraulic skeletonto realize steering function. *IEEE Robot. Autom. Let*. 6, 5413–5420. doi: 10.1109/LRA.2021.3073653

Tan, N., Yu, P., Zhang, X. Y., and Wang, T. (2021). Model-free motion control of continuum robots based on a zeroing neurodynamic approach. *Neural Netw*. 133, 21–31. doi: 10.1016/j.neunet.2020.10.005

Wang, Y. Y., Liu, L. F., Wang, D., Ju, F., and Chen, B. (2020). Time-delay control using a novel nonlinear adaptive law for accurate trajectory tracking of cable-driven robots. *IEEE Trans. Ind Inform*. 16, 5234–5243. doi: 10.1109/TII.2019.2951741

Wu, Q. X., Gu, Y. Q., Li, Y. C., and Zhang, B. T. (2020). Position control of cable-driven robotic soft arm based on deep reinforcement learning. *Inform (Swi)*. 11, 310. doi: 10.3390/info11060310

Xu, B. G., Song, A. G., Zhao, G. P., Liu, J., Xu, G. Z., Pan, L. Z., et al. (2018). EEG-modulated robotic rehabilitation system for upper extremity. *Biotechnol. Biotec. Eq*. 32, 795–803. doi: 10.1080/13102818.2018.1437569

Yang, H., Xu, M., and Li, W. (2018). Design and implementation of a soft robotic arm driven by SMA coils. *IEEE Trans. Ind. Electron*. 66, 6108–6116. doi: 10.1109/TIE.2018.2872005

Zhang, Q., Sun, D. Y., Qian, W., Xiao, X. H., and Guo, Z. (2020). Modeling and control of a cable-driven rotary series elastic actuator for an upper limb rehabilitation robot. *Front. Neurorobot*. 14:13. doi: 10.3389/fnbot.2020.00013