Robotica (2024), **42**, pp. 2677–2689
doi:10.1017/S0263574724001048

RESEARCH ARTICLE

DDPG-based path planning for cable-driven manipulators in multi-obstacle environments

Dong Zhang[1], Renjie Ju[1] and Zhengcai Cao[2]

[1]College of Information Science and Technology, Beijing University of Chemical Technology, Beijing, China
[2]The State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, Harbin, China
Corresponding author: Zhengcai Cao; Email: caozc@hit.edu.cn

Received: 31 October 2023; **Revised:** 26 March 2024; **Accepted:** 8 May 2024; **First published online:** 13 September 2024

Keywords: cable-driven manipulator; path planning; deep deterministic policy gradient

Abstract

Hyper-redundant cable-driven manipulators (CDMs) are widely used for operations in confined spaces due to their slender bodies and multiple degrees of freedom. Most research focuses on their path following but not path planning. This work investigates a deep deterministic policy gradient (DDPG)-based path-planning algorithm for CDMs in multi-obstacle environments. To plan passable paths under many constraints, a DDPG algorithm is modified according to features of CDMs. To improve adaptability of planned paths, a specialized reward function is newly designed. In this function, such factors as smoothness, arrival time and distance are taken into account. Results of simulations and physical experiments are presented to demonstrate the performances of the proposed methods for planning paths of CDMs.

1. Introduction

Cable-driven manipulators (CDMs) are a novel class of robots. Similar to snake robots [1], they can move flexibly in confined environments due to their slender bodies and a large number of degrees of freedom (DOFs) [2]. Unlike snake robots, CDMs have separate driving systems and all DOFs can be independently controlled. This makes them more suitable than snake robots for executing dangerous tasks such as operating in nuclear facilities and intricate industrial devices [3, 4].

In the past decades, many hyper-redundant manipulators were designed, which can be roughly classified into two categories: rigid-backbone and continuum-backbone manipulators [5, 6]. As a main form of these robots, a rigid-backbone robot is connected by a number of joints and links. As a pioneer, Shigeo designs a multi-segment manipulator [7]. Several commercial manipulators of this structure are developed by OC robotics [8]. Kang and Dai designed a pneumatic muscle-based continuum robot with embedded tendons in ref. [9] and a novel continuum robot with shape memory alloy initiated variable stiffness in [10]. In refs. [11, 12], mechanism and control of a manipulator that can coil and uncoil are investigated. Such robots as the hyper-redundant manipulator developed by OC Robotics [13] and a collaborative continuous robot system developed by Dong [14] are used in challenging environments.

Since CDMs execute tasks in confined spaces with multi-obstacle, they require high performances in terms of safety and precision. However, it is difficult to control them due to their hyper-redundancy DOFs, nonlinear and multi-level relationships among motors, joints and end effectors [15]. To solve these problems, several motion control methods including path following and path planning of CDMs were proposed in the past years.

Path following of CDMs is to control their slender bodies to track desired paths without collision. Since high DOFs of CDMs, their path following is usually solved by using an iterative Jacobian-based method [16]. Conkur computes link's position relative to a given B-spline curve path by using a numerical approach [17]. To control a snake arm robot following desired trajectories with minimal errors, Palmer et al. propose a tip-following approach by using a sequential quadratic programing optimization method [18]. In ref. [19], a tip-following method is proposed for CDMs and its performance is analyzed. A path-following method with prediction lookup and interpolation algorithms is investigated for CDMs working in confined spaces [20]. With these methods, CDMs can move along desired paths accurately. However, these paths are given by operators according to constraints of tasks and CDMs. When they are required to automatically execute special tasks in multi-obstacle environments, their path planning is essential.

Unlike path following, path-planning methods are rarely researched. In the recent years, path planning for CDMs has emerged as a prominent research area. In ref. [21], bilevel optimization and genetic algorithms are used to plan paths for redundant manipulators. It maximizes the manipulability of the robot but fails to consider smoothness of the planned paths. In ref. [22], a motion planning method that considers constraints of joint limits and environments is designed for a novel coiled CDM. However, this motion planning method depends on the kinematic model. Wei et al. propose a specialized rapidly exploring random tree (RRT) path-planning method for achieving a follow-the-leader motion of CDMs [23]. A CDM controlled by this method can move in porous plate environments. Jia et al. propose an MDA + RRT path-planning approach for CDMs that considers the maximum deflection angle of joints [24]. To quickly plan suitable paths for CDMs, an RRT-A* method is proposed and verified in ref. [25]. Yang et al. designed an ellipsoid-shaped RRT*-based path-planning method for CDMs with angle limitations [26]. The mentioned methods are based on research conducted on accurate models of CDMs. In reality, it is difficult to establish such accurate models because of their complex mechanisms and coupling relationships.

Different from these model-based methods, reinforcement learning (RL) algorithm can plan suitable paths without depending on any models [27, 28]. In ref. [27], Sangiovanni et al. propose a collision avoidance method for a 6-DOF manipulator based on a deep RL algorithm. The manipulator can complete tasks with an unpredictable obstacle invasive its workspace. To plan collision-free path in duct-entry space for an 8-DOF manipulator, an RL-based collision-free path planner is designed [28]. The generality of the planner is not enough for different ducts. This work is significant for control manipulators. However, their manipulators' DOF is lesser than CDMs' and their environments are not complex enough. Driven by this work, it is valuable to research RL-based path-planning methods for hyper-redundant CDMs in multi-obstacle environments.

In this paper, path planning of CDMs in multi-obstacle environments is tackled with a deep deterministic policy gradient (DDPG) algorithm. To meet pose constraints of hyper-redundant CDMs, DDPG is modified according to CDMs' features. A specialized reward function considering smoothness, arrival time and distance is designed to improve the adaptability of planned paths. A tip-following method is used to control CDMs to move along a planned path. Extensive simulations and experiments about path planning and following a 17-DOF CDM are conducted. To the best of our knowledge, there has been little research on using DDPG for path planning of CDM.

The rest of this paper is organized as follows: design and kinematics of CDM are described in Section 2. A DDPG-based path planning and tip-following methods are given in Section 3. Simulations and experiments are conducted to verify its efficiency in Section 4. Section 5 concludes the paper.

2. Design and mathematical model of CDM

As shown in Fig. 1, a CDM with 17 DOFs is designed. It consists of a robotic manipulator and an actuation system. The manipulator is connected by 8 rotational sections. Each section is driven by 3 cables and it can rotate in pitch and yaw directions through coupling control of corresponding cables. There are 24 groups of motors and driving modules arranged circularly on a pedestal. In order to expand the robot's workspace, the pedestal is fixed on a platform that can move forward and backward.

Table I. *Mathematical notation.*

Symbol	Description
d_l	The length of each link
d_e	The distance between the end effector and joint 8
d_b	Translation distance of $\mathbb{B}$ relative to $\mathbb{S}$
$d_{\mathbb{S},\mathbb{B}}$	Transformation matrix from $\mathbb{S}$ to $\mathbb{B}$
$d_{\mathbb{S},\mathbb{E}}$	Transformation matrix from $\mathbb{S}$ to $\mathbb{E}$
$d_{\mathbb{B},\mathbb{E}}$	Transformation matrix from $\mathbb{B}$ to $\mathbb{E}$
$\hat{\xi}_i$	Twist of joint i
ω_i	Unit direction vector of axis i
q_i	Center of mass of the rotation axis i
$\mathbb{S} = (X_\mathbb{S}, Y_\mathbb{S}, Z_\mathbb{S})$	Reference coordinate system
$\mathbb{B} = (X_\mathbb{B}, Y_\mathbb{B}, Z_\mathbb{B})$	Base coordinate system
$\mathbb{E} = (X_\mathbb{E}, Y_\mathbb{E}, Z_\mathbb{E})$	End coordinate system
$\theta_{2i-1}, i = 1, 2, \ldots, 8$	Pitch angle of joint i
$\theta_{2i}, i = 1, 2, \ldots, 8$	Yaw angle of joint i
C_θ, S_θ	$\cos\theta$, $\sin\theta$

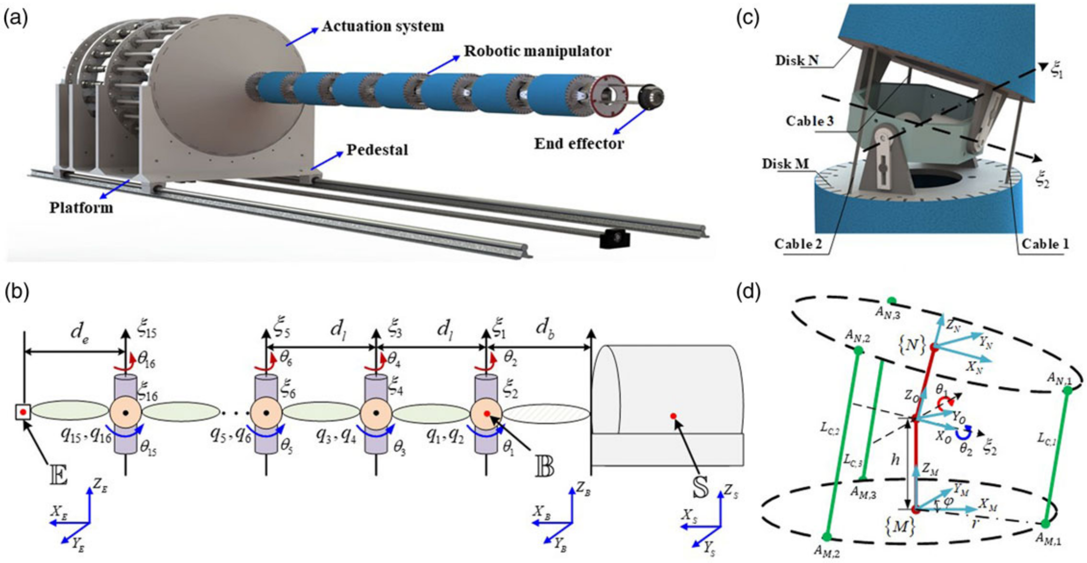

Figure 1. *Design and model of a 17-degree of freedom cable-driven manipulator (CDM). (a) The designed hyper-redundant CDM. (b) A simplified geometric model of the CDM. (c) Model of a joint. (d) Geometric model.*

2.1. Kinematics analysis of CDM

In order to control CDM, we establish and analyze its kinematics model and mapping relationships by using a product of exponentials (POEs) formula method. As shown in Fig. 1(d), the robotic manipulator has 8 sections. Each section is composed of a link and a universal joint. Mathematical symbols used in this paper are listed in Table I.

The kinematics analysis of CDM includes relationships among task space, joint space and driving space. Since the focus of this work is path planning of CDM, we mainly derive kinematics between task space and joint one. For the task-joint kinematics, a mutual transformation between rotational joints' angles and end effector's posture is necessary.

Table II. *Product of exponential parameters of cable-driven manipulator.*

Index *i*	ω_i	q_i
1	$[0, 0, 1]^T$	$[0, 0, 0]^T$
2	$[0, -1, 0]^T$	$[0, 0, 0]^T$
3	$[0, -1, 0]^T$	$[d_l, 0, 0]^T$
4	$[0, 0, 1]^T$	$[d_l, 0, 0]^T$
$\vdots$	$\vdots$	$\vdots$
15	$[0, -1, 0]^T$	$[7d_l, 0, 0]^T$
16	$[0, 0, 1]^T$	$[7d_l, 0, 0]^T$

As defined in our previous work [29], f_1 and f_1^{-1} are the forward and inverse kinematics between task space and joint space, respectively. We have:

$$f_1 = T_{\mathbb{S},\mathbb{E}} = T_{\mathbb{S},\mathbb{B}}(dB)T_{\mathbb{B},\mathbb{E}}(\theta_1, \theta_2, \theta_3, \ldots, \theta_{16}). \tag{1}$$

The twist $\hat{\xi}_i$ can be obtained from:

$$\hat{\xi}_i = \begin{bmatrix} \hat{\omega}_i & v_i \\ 0 & 0 \end{bmatrix} \in SE(3), \tag{2}$$

where $\hat{\omega}_i$ is a skew symmetric matrix of ω_i, $\xi_i = [v_i^T, \omega_i^T]^T$, $v_i = -\omega_i \times q_i$, ω_i and q_i are set as Table II.

In a POE formula method, a rotation motion of CDM can be described in the form of an exponential product as:

$$e^{\hat{\xi}\theta} = \begin{bmatrix} R & \rho \\ 0 & 1 \end{bmatrix} \in SE(3), \tag{3}$$

where $R = I + \sin\theta\hat{\omega} + (1 - \cos\theta)\hat{\omega}^2$, $\rho = (I\theta + (1 - \cos\theta)\hat{\omega} + (\theta - \sin\theta)\hat{\omega}^2)v$ and $I \in R^{3\times3}$ is an identity matrix. The posture of the end effector is:

$$T_{\mathbb{B},\mathbb{E}}(\theta_1, \theta_2, \ldots, \theta_{16}) = e^{\hat{\xi}_1\theta_1} e^{\hat{\xi}_2\theta_2} \ldots e^{\hat{\xi}_{16}\theta_{16}} T_{\mathbb{B},\mathbb{E}}(0), \tag{4}$$

where $T_{\mathbb{B},\mathbb{E}}(0)$ is an initial end posture of CDM.

2.2. Kinematics analysis of a joint

As shown in Fig. 1(c), position and orientation of each section are determined by rotation of its joints. Rotation angles are controlled by corresponding three cables. To analyze the relationship between cable lengths and rotation angles of joints, a geometric model of a joint is simplified as shown in Fig. 1(d). In the schematic diagram of the first universal joint, $i = 1$, the transformation relationship between coordinate systems is:

$${}^M_O T = Trans\ (0, 0, h)\ Rot\ (X, \theta_2), \tag{5}$$

$${}^O_N T = Rot\ (Y, \theta_1)\ Trans\ (0, 0, h). \tag{6}$$

The transformation relationship between $\{M\}$ and $\{N\}$ is:

$${}^N_M T = {}^O_N T {}^M_O T = \begin{bmatrix} C_{\beta_i} & S_{\alpha_i}S_{\beta_i} & C_{\alpha_i}S_{\beta_i} & 2S_{\beta_i}h \\ 0 & C_{\alpha_i} & -S_{\alpha_i} & 0 \\ -S_{\beta_i} & C_{\beta_i}S_{\alpha_i} & C_{\alpha_i}C_{\beta_i} & 2C_{\beta_i}h \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{7}$$

In addition, coordinates of cables' holes in the corresponding disk can be computed.

The driven cables of the first joint can be computed as:

$$L_{c,j}=\begin{bmatrix} 2hS_{\theta_1}-rC_{\varphi+(j-1)\frac{2\pi}{3}}+rC_{\varphi+(j-1)\frac{2\pi}{3}}C_{\theta_1}+rS_{\theta_2}S_{\theta_1}S_{\varphi+(j-1)\frac{2\pi}{3}} \\ rC_{\theta_2}S_{\varphi+(j-1)\frac{2\pi}{3}}-rS_{\varphi+(j-1)\frac{2\pi}{3}} \\ 2hC_{\theta_1}-rC_{\varphi+(j-1)\frac{2\pi}{3}}S_{\theta_1}+rC_{\theta_1}S_{\theta_2}S_{\varphi+(j-1)\frac{2\pi}{3}} \\ 0 \end{bmatrix},\ j=1,2,3. \tag{8}$$

Their lengths are:

$$\begin{aligned} |L_{c,j}| &= ((rC_{\varphi+(j-1)\frac{2\pi}{3}}(C_{\theta_1}-1)+2hS_{\theta_1}C_{\theta_2}+rS_{\varphi+(j-1)\frac{2\pi}{3}}S_{\theta_1}S_{\theta_2})^2 \\ &\quad +(2h(C_{\theta_1}C_{\theta_2}+1)-rC_{\varphi+(j-1)\frac{2\pi}{3}}S_{\theta_1}+rS_{\varphi+(j-1)\frac{2\pi}{3}}C_{\theta_1}S_{\theta_2})^2 \qquad j=1,2,3. \\ &\quad +(rS_{\varphi+(j-1)\frac{2\pi}{3}}(C_{\theta_2}-1)-2hS_{\theta_2})^2)^{1/2} \end{aligned} \tag{9}$$

where r is radius of the circle where the holes are distributed on the disk.

Using above kinematics model, desired motion of other section can be realized with given pitch and yaw angles θ_{2i-1} and θ_{2i}.

3. Path planning based on DDPG

Path planning is important for CDMs to move in complex environments with multiple obstacles. Objectives of CDM path planning can be presented as:

- Followed ability: The curvature of a planned path should be within the limitation of CDM joints.
- Path performances: The planned path should be shorter and smoother.
- Efficiency: Passable paths should be planned rapidly in different environments.

For the path planning of CDMs, the states of CDMs at a certain moment only depend on the previous states and actions. They are independent of the previous states and actions. Due to this feature, path planning of CDM can be modeled as a Markov decision problem. RL is particularly suitable for solving MDPs due to its advantages, such as model independence, exploration-exploitation balance, dynamic decision-making ability and reward-driven learning. Compared with other DRL algorithms, DDPG performs better in processing continuous action spaces, expressing deep learning, as well as algorithm stability and convergence, making it suitable for solving MDPs. In this work, a DDPG-based method is investigated to solve path planning of CDM.

3.1. DDPG

DDPG is a model-free algorithm that can learn competitive policies for many tasks by using same hyperparameters and network structures [30]. It is illustrated in Fig. 2 and realized in Algorithm 1.

DDPG maintains a parameterized actor function $\mu(s \mid \theta^{\mu})$ that specifies the current policy by deterministically mapping states to a specific action. A critic $Q(s, a)$ is learned by using a Bellman equation. An actor is updated by using a chain rule to an expected return from a start distribution J according to parameters of the actor:

$$\begin{aligned} \nabla_{\theta_\mu} J &\approx \mathbb{E}_{s_t\sim\rho^\beta}[\nabla_{\theta^\mu} Q(s,a \mid \theta^Q)) \mid_{s=s_t,a=\mu(s_t\mid\theta^\mu)}] \\ &= \mathbb{E}_{s_t\sim\rho^\beta}[\nabla_a Q(s,a \mid \theta^Q) \mid_{s=s_t,a=\mu(s_t)} \nabla_{\theta_\mu}\mu(s \mid \theta^\mu) \mid s=s_t]. \end{aligned} \tag{10}$$

To accelerate the convergence of DDPG, weights of the network are updated by a back propagation algorithm based on an experience replay technology. Transitions are sampled from environments

Algorithm 1: The DDPG Algorithm.

```
Initialize target network $Q'$ and $\mu'$ with parameters $\theta^{Q'} \leftarrow \theta^{Q}$, $\theta^{\mu'} \leftarrow \theta^{\mu}$;
Initialize target network $Q'$ and $\mu'$ with parameters; $\theta^{Q'} \leftarrow \theta^{Q}$, $\theta^{\mu'} \leftarrow \theta^{\mu}$;
Initialize the replay buffer.
for $episode = 1, \ldots, M$ do
    Initialize a random process $\mathcal{N}$ for action exploration;
    Observe $s_1$;
    for $t = 1, 2, \ldots, T$ do
        Obtain $a_t = \mu(s_t \mid \theta^{\mu}) + \mathcal{N}_t$;
        Execute $a_t$, observe $r_t$;
        Observe $s_{t+1}$.
        $\mathfrak{R} \leftarrow (s_t, a_t, r_t, s_{t+1})$;
        Sample a random minibatch of $\mathcal{N}_t$ transitions $(s_i, a_i, r_i, s_{i+1})$ from $\mathfrak{R}$.
        Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1} \mid \theta^{\mu'}) \mid \theta^{Q'})$;
        $L = \frac{1}{N} \sum_i (y_i - Q(s_i, a_i \mid \theta^{Q}))^2$;
        Update the actor policy according to the sampled policy gradient:
        $\nabla_{\theta^{\mu}} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a \mid \theta^{Q}) \mid_{s=s_i, a=\mu(s_i)} \nabla_{\theta^{\mu}} \mu(s|\theta^{\mu})|_{s_i}$,
        $\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'}$,
        $\theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'}$.
    end
end
```

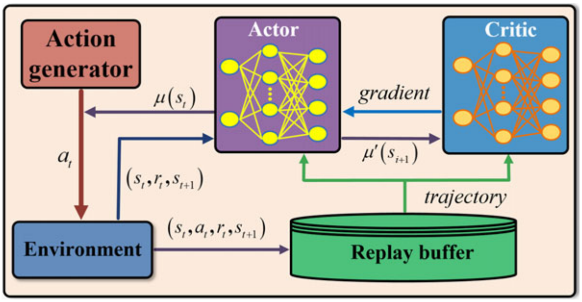

Figure 2. The deep deterministic policy gradient path-planning algorithm.

according to the exploration policy and a tuple (s_t, a_t, r_t, s_{t+1}) is stored in a replay buffer $\mathfrak{R}$. The oldest samples are discarded when $\mathfrak{R}$ is full. The actor and critic networks are updated by sampling a minibatch uniformly from $\mathfrak{R}$ at each timestep. Since DDPG is an off-line policy algorithm, $\mathfrak{R}$ should be large enough to improve its benefit.

3.2. Reward function

A proper reward function is key to a successful reinforcement learning algorithm. This work proposes the following reward function to plan a safe path toward the target for the CDM, i.e.,

Table III. *Parameters of deep deterministic policy gradient.*

Symbol	**Value**
The maximum iteration count M_{max}	5000
The maximum step count T_{max}	200
Discount factor γ	0.99
Update factor τ	0.005
Learning rate η	0.0001
Noise Δ	$Ornstein - Uhlenbeck$
Exploration noise	0.025
Weights of reward function c_1, c_2	$c_1 = 1000$, $c_2 = 60$
δ	0.1
p	8
d_{ref}	0.2

$$R = \begin{cases} 500, & case\ 1 \\ c_1R_1 + c_2R_2, & case\ 2 \\ -50, & case\ 3 \end{cases} \tag{11}$$

where $case1$ is the robot arrives at the target, $case2$ is the robot moves toward the target, $case3$ is the robot collides with obstacles, R_1 is the distance reward between the end effector and the target, R_2 is a distance reward between the end effector and the obstacle and c_1 and c_2 are weights that decide priority during training. The distance reward R_1 is computed by using a Huber-Loss function:

$$R_1 = \begin{cases} \frac{1}{2}d^2, & |d| < \delta \\ \delta(|d| - \frac{1}{2}\delta), & otherwise \end{cases}, \tag{12}$$

where d is the Euclidean distance between CDM's end effector and the target and δ is a parameter that determines the smoothness. The second reward R_2 relates to the distance between the robot and the nearest obstacle. It is computed as:

$$R_2 = \left(\frac{d_o}{\check{d} + d_o}\right)^p, \tag{13}$$

where d_o is set as a constant to ensure $R_2 \in (0, 1)$, $\check{d}$ is the minimum distance between the end effector and the obstacle and p is an exponential decay of the negative reward. Parameters of DDPG used in this work are listed in Table III.

3.3. *Path following of CDMs*

To control CDM to move along a planned path, a tip-following method as shown in Fig. 3 is adopted. Joints and links of CDM should be matched with the planned path. The planned path is discretized into some waypoints. As shown in Fig. 3(a), k_i is a key point located at the center of mass of universal joint i, the pedestal is fixed with k_1 and M_i is the *i-th* mark point located on the path. In this work, a set of points denoted as M are generated by discretizing the path. The length of a curve between adjacent points is set as a constant s_d.

When k_1 is located at a discrete point of the path, the point is defined as p_{k_1}. Other key points p_{k_i} need satisfy the constraint:

$$\min |\overrightarrow{p_{k_i}p_{k_{i-1}}}| - d_l, k_i \in (k_{i-1}, n \times p), \tag{14}$$

where $n \times p$ is the number of all discretization points.

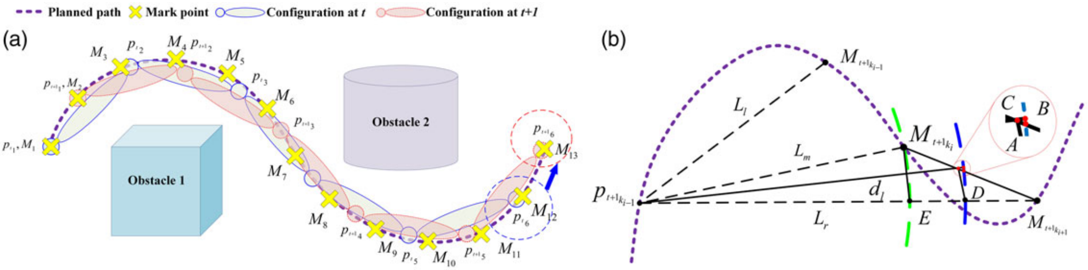

Figure 3. The tip-following method of cable-driven manipulators. (a) Iteration of links. (b) The interpolation algorithm.

To find p_{k_i}, this work adopts an iteratively sequential searching method proposed in refs. [19, 20]. As shown in Fig. 3(a), $p_{^tk_i}$ is the nearest mark point to the key point k_i at time t. In the next time step, key points move forward along the direction of discrete points:

$$^{t+1}k_i -^{t+1} k_{i-1} \approx^t k_i -^t k_{i-1} = \Delta k, \tag{15}$$

where $\Delta k = d_l/s_d$. Index $^{t+1}k_i$ can be predicted.

Algorithm 2: The Key Point Searching Algorithm.

```
$\Delta k = d_l/s_d$,
$^{t+1}\bar{k}_i =^{t+1} k_{i-1} + \Delta k$.
Flag$=0$,
$L_l = |\overrightarrow{p_{^{t+1}k_i} M_{^{t+1}k_i-1}}|$,
$L_m = |\overrightarrow{p_{^{t+1}k_i} M_{^{t+1}k_i}}|$,
$L_r = |\overrightarrow{p_{^{t+1}k_i} M_{^{t+1}k_i+1}}|$.
$D_l = ||L_l - d_l||$, $D_m = ||L_m - d_l||$, $D_r = ||L_r - d_l||$.
$D_{min} = \min(D_l, D_m, D_r)$
while $\sim$ Flag do
    case 1,
    $D_{min} = D_m$,
    $p_{^{t+1}k_i} = M_{^{t+1}k_i}$,
    Flag$=1$.
    case 2,
    $D_{min} = D_l$,
    $L_r = L_m, L_m = L_l, L_l = |\overrightarrow{p_{^{t+1}k_i} M_{^{t+1}k_i-2}}|$.
    case 3,
    $D_{min} = D_r$,
    $L_l = L_m, L_m = L_r, L_r = |\overrightarrow{p_{^{t+1}k_i} M_{^{t+1}k_i+1}}|$.
end
Return $p_{^{t+1}k_i}$.
```

With these predictions, the points $p_{^{t+1}k_i}$ corresponding to $^{t+1}k_i$ need to be verified whether satisfy (14) by using Algorithm 2. Although $p_{^tk_i}$ is the nearest mark point to the key point k_i, the distance between $p_{^tk_i}$ and $p_{^{t+1}k_{i-1}}$ is unequal to the link's length. A point certifies Equation (14) needs to be found around k_i. As shown in Fig. 3(b), $M_{^{t+1}k_i}$ is the predicted $p_{^{t+1}k_i}$, $M_{^{t+1}k_{i-1}}$ and $M_{^{t+1}k_{i+1}}$ are its adjacent mark points.

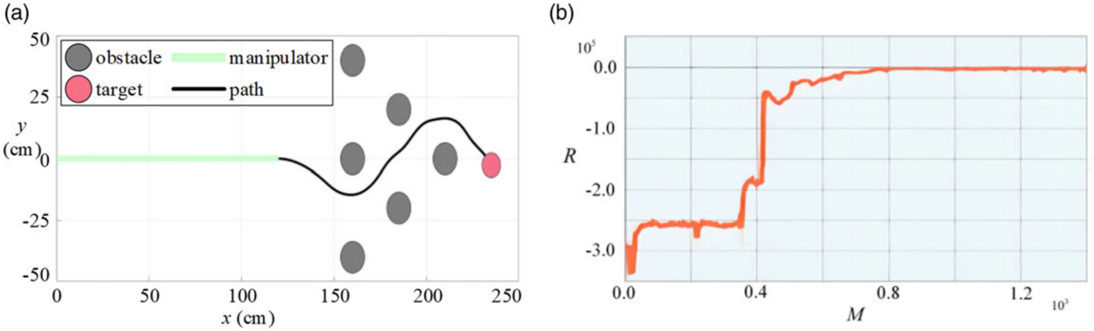

Figure 4. *Simulations of path planning by using the proposed deep deterministic policy gradient algorithm. (a) The planned path. (b) The reward converges to a stable state.*

The expected $p_{t+1_{k_i}}$ locates the blue arc. The center and the radius of the arc are $p_{t+1_{k_{i-1}}}$ and d_l, respectively. The expected $p_{t+1_{k_i}}$ can be computed by using the geometric relationships. Positions of all joints can be computed based on the planned path after this iteration.

4. Evaluation

To test the proposed DDPG-based path-planning method, we have conducted a group of simulations and experiments. In these experiments, the end effector of the CDM is required to move from a start point (130, 0, 0) to a target position (235, 0, 0) while avoiding six cylinders with a radius of 5 cm and a height of 20 cm. Their coordinates are (160,-4,0), (160,0,0), (160,4,0), (185,-2,0), (185,2,0) and (210,0,0), respectively. The initial pitch and yaw angles of all joints of the CDM are 0 degrees. Through multiple experiments, parameters of DDPG are compared and selected as shown in Table III.

4.1. Simulations

In our simulations, Simulink, CoppeliaSim and Pycharm are used. Communication among different platforms is implemented through a remote API. Two groups of simulation are conducted in Simulink and CoppeliaSim, respectively. In Simulink and CoppeliaSim, a virtual robot is built according to the size of our prototype and a multi-obstacle environment is conducted.

The passable path is planned by DDPG algorithm and CDM is controlled by the tip-following method to move toward the target. As shown in Fig. 4(a), DDPG algorithm can plan a passable path (black curve) in a multi-obstacle environment. The reward can converge to a stable state as shown in Fig. 4(b). Angles and positions of links are computed by using the tip-following method. As shown in Fig. 5(a) and (b), the CDM can move along the planned path (black curve) without collision in Simulink and CoppeliaSim. Joint angles are measured and saved to analyze the motion controlled by the proposed methods. As shown in Fig. 5(c), all the angles are always less than 40 degrees while moving from the start point to the target point. They are smooth enough and satisfy to mechanical constraints of the joints.

4.2. Physical experiments

Prototype experiences are used to test our path planning and following methods of CDMs. As shown in Fig. 6(a), the robotic manipulator is connected by 8 sections. The platform is powered by a stepper motor (57TB6600). Each section can rotate between [−45°, +45°]. Motors communicate with a PC via an analyzer (Kvaser Leaf Light v2). Geometric parameters of our CDM are summarized in Table IV.

Table IV. *Geometric parameters of cable-driven manipulator.*

Parameter	**Value**
Total length of the CDM (mm)	1500
Size of the robotic manipulator (mm)	$\Phi 100 \times 1375$
Displacement of the platform (mm)	1200
Number of sections	8
Size of the driving cable (mm)	Φ 1
Number of motors	25
Number of DOFs	17

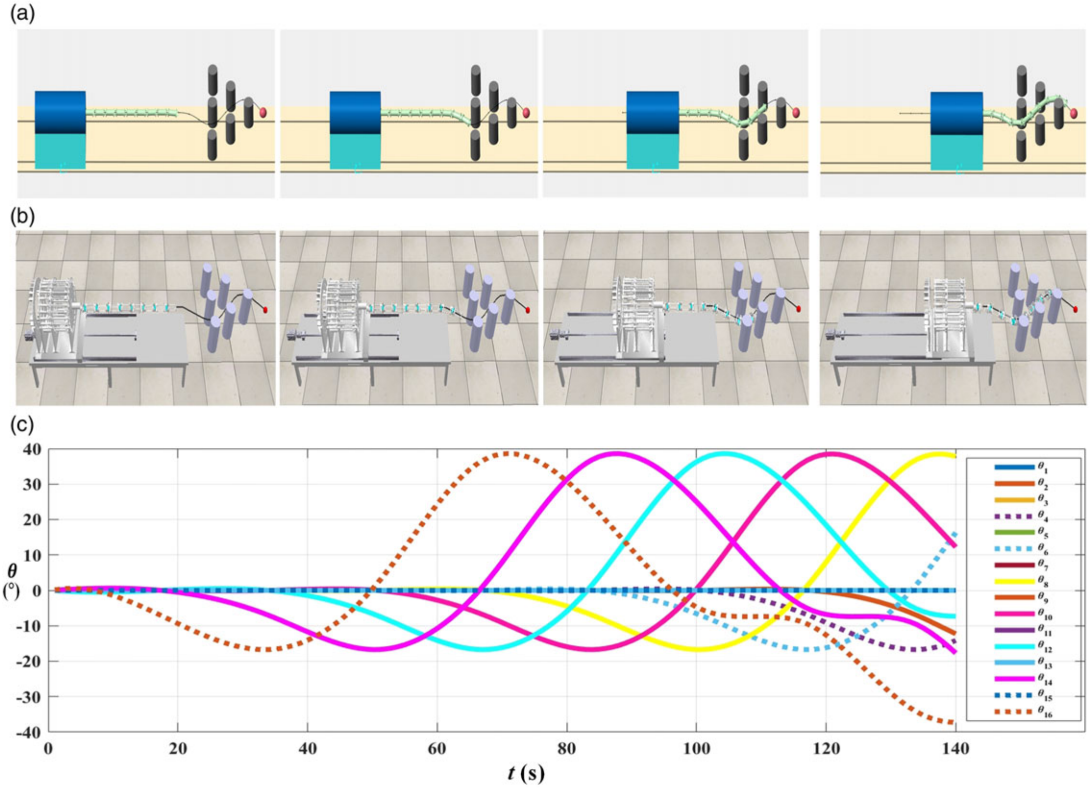

Figure 5. *Simulations of path planning and following of a cable-driven manipulator by using the proposed methods. (a) Simulations in Simulink. (b) Simulations in CoppeliaSim. (c) Joint angles.*

To control CDM move toward the target along a desired path, a closed-loop control framework is designed. Links and obstacles are wrapped by tapes to fix markers. To measure states of CDM accurately, each link attached with 3 markers is defined as a rigid body. Positions of all markers can be measured by a 3-D OptiTrack motion capture system. States of CDM can be computed by using these data and its kinematics model. Passable paths are planned by using DDPG. Positions of all links are computed by the tip-following method. Joint angles and cable lengths are computed according to multi-level relationships among motors, joints and the end effector of the robot. As shown in Fig. 6(b), our CDM controlled by the proposed methods can plan a possible path and move toward a target position smoothly without collision in a multi-obstacle environment.

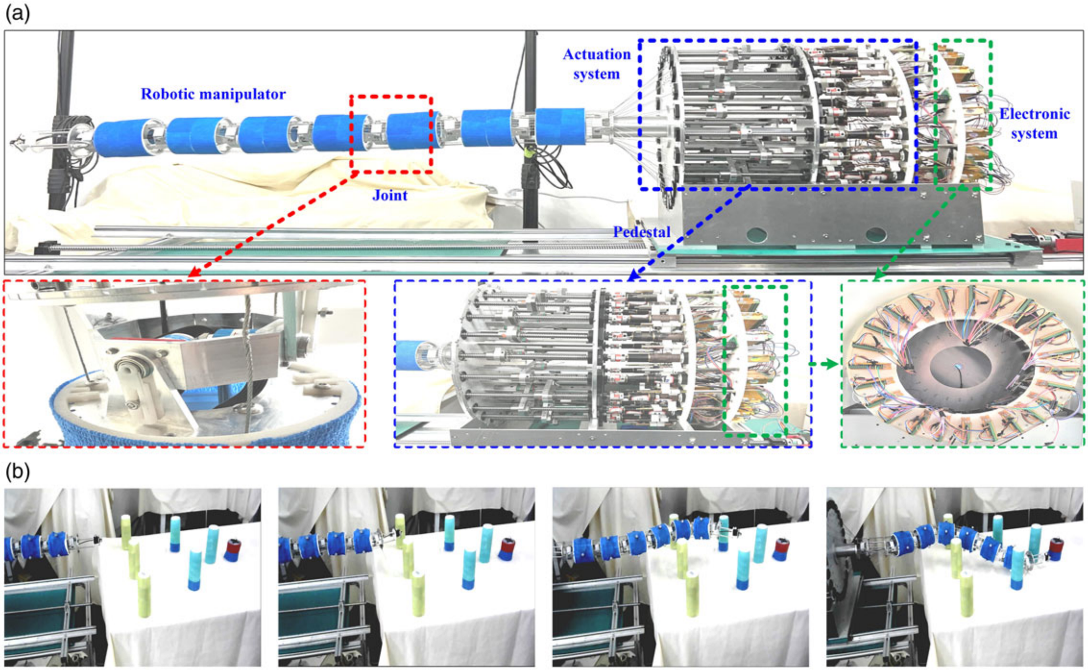

Figure 6. The cable-driven manipulator (CDM) prototype and experiments. (a) The structure of CDM. (b) Experiments of path planning and following of CDM by using the proposed methods.

5. Conclusion and future work

In this work, a specialized DDPG-based path-planning algorithm is designed for CDMs. A reward function under constraint of CDMs' features and multi obstacles is designed to train the DDPG path-planning network. To control the robot moving along the planned path without collisions, a tip-following method is used. Various simulations and experiments are conducted in CoppeliaSim and Simulink to validate the effectiveness of the proposed algorithms. In order to demonstrate the practicality of the motion control methods, they are verified on a 17-DOF CDM prototype through simulations and experiments. The results validate the effectiveness of our path planning and following methods.

The CDM designed in this work can only move in environments with prior static obstacles automatically. However, in most real tasks, there are many dynamic obstacles. In future work, we will improve its mechanisms using methods proposed in refs. [31, 32] and optimize its control by combining environmental perception and understanding algorithms proposed in refs. [33, 34].

Author contributions. Dong Zhang, Renjie Ju and Zhengcai Cao conceived and designed the study and wrote the article.

Financial support. This work is supported by National Natural Science Foundation of China (52105005), the Beijing Natural Science Foundation (L223019), Open Foundation of State Key Laboratory of High-end Compressor and System Technology (SKL-YSJ202311) and Fundamental Research Funds for the Central Universities (ZY2415).

Competing interests. The authors declare no competing interests exist.

Ethical approval. The authors declare none.

References

[1] D. Li, B. Zhang, Y. Xiu, H. Deng, M. Zhang, W. Tong, R. Law, G. Zhu, E. Q. Wu and L. Zhu, "Snake robots play an important role in social services and military needs," *Innovation* **3**(6), 100333 (2022).

[2] J. W. Robert and A. Bryan, "Design and kinematic modeling of constant curvature continuum robots: A review," *Int J Robot Res* **29**(13), 1661–1683 (2010).

[3] T. Li, S. G. Ma, B. Li, M. H. Wang and Y. C. Wang, "Axiomatic design method to design a screw drive in-pipe robot passing through varied curved pipes," *Sci China Technol Sci* **59**(2), 191–202 (2016).

[4] J. Peng, C. Zhang, D. Meng and B. Liang, "Trajectory optimization methods of a space hyper-redundant robot based on effective arm-shape measurement," *IEEE Trans Instrum Meas* **72**(5017914), 1–14 (2023).

[5] I. D. Walker, H. Choset and G. S. Chirikjian, "Snake-Like and Continuum Robots," **In:** *Springer Handbook of Robotics* (B. Siciliano and O. Khatib, eds.) (Springer International Publishing, Berlin, Heidelberg, 2016) pp. 481–498.

[6] X. Dong, M. Raffles, S. Cobos-Guzman, D. Axinte and J. Kell, "A novel continuum robot using twin-pivot compliant joints: Design, modeling, and validation," *J Mech Robot* **8**(2), 021010 (2015).

[7] H. Shigeo, C. Peter and G. Charles, *Biologically Inspired Robots: Snake-Like Locomotors and Manipulators* (Oxford University Press, Oxford, 1993).

[8] R. Buckingham, V. Chitrakaran, R. Conkie, G. Ferguson, A. Graham, A. Lazell, M. Lichon, N. Parry, F. Pollard, A. Kayani, M. Redman, M. Summers and B. Green, Snake-arm robots: A new approach to aircraft assembly, (2007). SAE technical paper, 01–3870.

[9] R. Kang, Y. Guo, L. Chen, D. Branson and J. Dai, "Design of a pneumatic muscle based continuum robot with embedded tendons," *IEEE/ASME Trans Mechatro* **22**(2), 751–761 (2017).

[10] C. Yang, S. Geng, I. Walker, D. T. Branson, J. Liu, J. S. Dai and R. Kang, "Geometric constraint-based modeling and analysis of a novel continuum robot with shape memory alloy initiated variable stiffness," *Int J Robot Res* **39**(14), 1620–1634 (2020). doi: 10.1177//0278364920913929.

[11] X. Dong, D. Axinte, D. Palmer, S. Cobos, M. Raffles, A. Rabani and J. Kell, "Development of a slender continuum robotic system for on-wing inspection/repair of gas turbine engines," *Robot Comp Integ Manuf* **44**, 218–229 (2017).

[12] A. Mohammad, M. Russo, Y. Fang, X. Dong, D. Axinte and J. Kell, "An efficient follow-the-leader strategy for continuum robot navigation and coiling," *IEEE Robot Automa Lett* **6**(4), 7493–7500 (2021).

[13] R. Buckingham and A. Graham, "Dexterous Manipulators for Nuclear Inspection and Maintenance; Case Study," **In:** *IEEE International Conference on Applied Robotics for the Power Industry*, (2010) pp. 1–6.

[14] X. Dong, M. Wang, A. Mohammad, W. Ba, M. Russo, A. Norton, J. Kell and D. Axinte, "Continuum robots collaborate for safe manipulation of high-temperature flame to enable repairs in extreme environments," *IEEE/ASME Trans Mechatro* **27**(5), 4217–4220 (2022).

[15] D. Lau, D. Oetomo and S. K. Halgamuge, "Inverse dynamics of multilink cable-driven manipulators with the consideration of joint interaction forces and moments," *IEEE Trans Robot* **31**(2), 479–488 (2015).

[16] L. Tang, L. Z. Ji, X. Zhu and G. Gu, "Path tracking of a cable-driven snake robot with a two-level motion planning method," *IEEE/ASME Trans Mechatro* **24**(3), 935–946 (2019).

[17] E. S. Conkur, "Path following algorithm for highly redundant manipulators," *Robot Auton Syst* **45**(1), 1–22 (2003).

[18] D. Palmer, S. Cobos-Guzman and D. Axinte, "Real-time method for tip following navigation of continuum snake arm robots," *Robot Auton Syst* **62**(10), 1478–1485 (2014).

[19] J. Wang, L. Tang, G. Gu and X. Zhu, "Tip-following path planning and its performance analysis for hyper-redundant manipulators," *J Mech Eng* **54**(3), 18–25 (2018).

[20] L. Tang, L. M. Zhu, X. Y. Zhu and G. Y. Gu, "Confined spaces path following for cable-driven snake robots with prediction lookup and interpolation algorithms," *Sci China Technol Sci* **63**(2), 255–264 (2020).

[21] S. M. LaValle, *Planning Algorithms* (Cambridge University Press, 2006). doi: 10.1017/CBO9780511546877.

[22] M. Luo, E. Li, A. Zhang, M. Tan and Z. Liang, "A bioinspired coiled cable-driven manipulator: Mechatronic design and kinematics planning with multiconstraints," *IEEE/ASME Trans Mechatro* **28**(6), 3155–3166 (2023).

[23] H. Wei, Y. Zheng and G. Gu, "RRT-Based Path Planning for Follow-the-Leader Motion of Hyper-Redundant Manipulators," **In:** *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (2021) pp. 3198–3204.

[24] L. Jia, Y. Huang, T. Chen, Y. Guo, Y. Yin and J. Chen, "MDA+RRT: A general approach for resolving the problem of angle constraint for hyper-redundant manipulator," *Exp Syst Appl* **193**, 116379 (2021).

[25] D. Zhang, Y. Gai, R. Ju, Z. Miao and J. Lao, "RRT-A* Path Planning Algorithm for Cable-Driven Manipulators," **In:** *IEEE International Conference on Robotics and Biomimetics*, (2022) pp. 451–456.

[26] H. Ji, H. Xie, C. Wang and H. Yang, "E-RRT*: Path planning for hyper-redundant manipulators," *IEEE Robotics and Automa Lett* **8**(12), 8128–8135 (2023).

[27] B. Sangiovanni, A. Rendiniello, G. P. Incremona, A. Ferrara and M. Piastra, "Deep Reinforcement Learning for Collision Avoidance of Robotic Manipulators," **In:** *European Control Conference*, (2018) pp.2063–2068.

[28] X. Hua, G. Wang, J. Xu and K. Chen, "Reinforcement learning-based collision-free path planner for redundant robot in narrow duct," *J Intell Manuf* **32**(2), 471–482 (2021).

[29] R. Ju, D. Zhang, J. Xu, H. Yuan, Z. Miao, M. Zhou and Z. Cao, "Design, modeling, and kinematics analysis of a modular cable-driven manipulator," *J Mech Robot* **14**(6), 064501 (2022).

[30] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver and D. Wierstra, Continuous Control with Deep Reinforcement Learning, Computer Science, (2015).

[31] F. Aimedee, G. Gogu, J. S. Dai, C. Bouzgarrou and N. Bouton, "Systematization of morphing in reconfigurable mechanisms," *Mech Mach Theory* **96**, 215–224 (2016).

[32] M. Salerno, K. Zhang, A. Menciassi and J. S. Dai, "A novel 4-DOFs Origami Enabled, SMA Actuated, Robotic End-Effector for Minimally Invasive Surgery," **In:** *2014 IEEE International Conference on Robotics and Automation*, (2014) pp. 2844–2849.

[33] Z. Cao, J. Li, D. Zhang, M. Zhou and A. Abusorrah, "A multi-object tracking algorithm with center-based feature extraction and occlusion handling," *IEEE Trans Intell Transp* **24**(4), 4464–4473 (2023).

[34] H. Mu, G. Zhang, Z. Ma, M. Zhou and Z. Cao, "Dynamic obstacle avoidance system based on rapid instance segmentation network," *IEEE Trans Intell Transp* **25**(5), 4578–4592 (2023). doi: 10.1109/TITS.2023.3323210.

Cite this article: D. Zhang, R. Ju and Z. Cao (2024). "DDPG-based path planning for cable-driven manipulators in multi-obstacle environments", Robotica **42**, 2677–2689. https://doi.org/10.1017/S0263574724001048