Neurocomputing 377 (2020) 73–84

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Comparison of end-to-end and hybrid deep reinforcement learning strategies for controlling cable-driven parallel robots

Hao Xiong, Tianqi Ma, Lin Zhang, Xiumin Diao*

School of Engineering Technology, Purdue University, West Lafayette, IN, USA

ARTICLE INFO

Article history:
Received 1 May 2019
Revised 17 August 2019
Accepted 1 October 2019
Available online 17 October 2019

Communicated by Dr. Grana Manuel

Keywords:
Deep reinforcement learning
End-to-end DRL strategy
Hybrid DRL strategy
Deep deterministic policy gradient
Cable-driven parallel robot

ABSTRACT

Deep reinforcement learning (DRL) has been proven effective in learning policies of high-dimensional states and actions. Recently, a variety of robot manipulation tasks have been accomplished using end-to-end DRL strategies. An end-to-end DRL strategy accomplishes a robot manipulation task as a black box. On the other hand, a robot manipulation task can be divided into multiple subtasks and accomplished by non-learning-based approaches. A hybrid DRL strategy integrates DRL with non-learning-based approaches. The hybrid DRL strategy accomplishes some subtasks of a robot manipulation task by DRL and the rest subtasks by non-learning-based approaches. However, the effects of integrating DRL with non-learning-based approaches on the learning speed and the robustness of DRL to model uncertainties have not been discussed. In this study, an end-to-end DRL strategy and a hybrid DRL strategy are developed and compared in controlling a cable-driven parallel robot. This study shows that, by integrating DRL with non-learning-based approaches, the hybrid DRL strategy learns faster and is more robust to model uncertainties than the end-to-end DRL strategy. This study demonstrates that, by taking advantages of both learning and non-learning-based approaches, the hybrid DRL strategy provides an alternative to accomplish a robot manipulation task.

1. Introduction

Deep reinforcement learning (DRL) introduces deep neural networks (DNNs) to solve reinforcement learning problems. With the help of DNNs, DRL can extract features from high-dimensional data and learn complex policies. DRL is especially suitable for sequential decision-making applications (e.g., robot manipulation) [1]. Many DRL algorithms, such as deep Q-network (DQN) [2], trust region policy optimization (TRPO) [3], mirror descent guided policy search (MDGPS) [4], and deep deterministic policy gradient (DDPG) [5], have recently been developed into end-to-end DRL strategies used for a variety of robot manipulation tasks [6–12]. An off-policy DRL strategy based on DNNs and guided policy search (GPS) was used to control a quadrotor to avoid obstacles in [6]. The off-policy DRL strategy employs the pose and the velocity of the quadrotor and data from 30 laser rangefinders installed on the quadrotor as its states. High-dimensional states are challenging to be integrated into non-learning-based control strategies, such as proportional-integral-derivative (PID) controllers, but can be easily processed by DRL strategies. In [7], TRPO was used to control bio-inspired robots in the MuJoCo simulator [13]. MDGPS was applied to the locomotion control of a tensegrity robot with 36 states in [8]. The high-dimension of states makes it hard to control the tensegrity robot using non-learning-based control strategies. Moreover, in [9], a DQN was applied to control a self-balancing robot with two wheels in the Gazebo simulator [14]. When these DRL strategies were used to control the robots in [6–12], the tasks of controlling the robots were treated as black boxes, without any knowledge of the internal workings of the robots. Therefore, these DRL strategies are regarded as end-to-end DRL strategies [15].

In practice, a task of controlling a robot is not always a complete black box. Besides inputs and outputs of the black box, one usually has certain knowledge about its internal workings. For example, when manipulating a robot, the robot manipulation task can be decomposed into multiple subtasks [16]. If one knows the internal workings of a subtask, this subtask may be accomplished by a non-learning-based approach (e.g., an inverse dynamics equation or a PID algorithm). This raises the question of whether there is any benefit of integrating DRL with non-learning-based approaches in a robot manipulation task. To the best knowledge of the authors, the effects of integrating DRL with non-learning-based approaches on the learning speed and the robustness of DRL to model uncertainties have not been discussed in the literature.

* Corresponding author.
E-mail address: diaox@purdue.edu (X. Diao).

https://doi.org/10.1016/j.neucom.2019.10.020
0925-2312/

To study the effects of integrating DRL with non-learning-based approaches on the learning speed and the robustness of DRL to model uncertainties, an end-to-end DRL strategy and a hybrid DRL strategy are developed and compared in this study in controlling a cable-driven parallel robot (CDPR). The end-to-end DRL strategy, called the end-to-end DDPG strategy, is developed based on a DDPG algorithm. With the end-to-end DDPG strategy, the task of controlling the CDPR is accomplished entirely by the DDPG algorithm. The hybrid DRL strategy, called the hybrid DDPG strategy, is developed by integrating a DDPG algorithm and the inverse dynamics equation of the CDPR. With the hybrid DDPG strategy, some subtasks of the task of controlling the CDPR are accomplished by the DDPG algorithm, while the other subtasks are accomplished by the inverse dynamics equation of the CDPR.

This study has the following two major contributions. Firstly, this study develops and compares the end-to-end DDPG strategy and the hybrid DDPG strategy in controlling CDPRs. It is shown that, the hybrid DDPG strategy learns faster and is more robust to model uncertainties than the end-to-end DDPG strategy. This suggests that, if some subtasks of a robot manipulation task can be accomplished by non-learning-based approaches, the hybrid DDPG strategy provides an alternative to accomplish a robot manipulation task. Secondly, this study shows that the end-to-end DDPG strategy can learn the optimal tension distribution of a CDPR as well as the hybrid DDPG strategy calculates the optimal tension distribution of a CDPR based on the inverse dynamics equation of the CDPR.

The rest of the paper is organized as follows. Section 2 introduces preliminaries of DDPG as well as the kinematics and dynamics of a CDPR. In Section 3, the hybrid DDPG strategy and the end-to-end DDPG strategy are proposed for controlling CDPRs. The training of DDPG in the proposed strategies is demonstrated in Section 4. The ability of the end-to-end DDPG strategy to learn the optimal tension distribution of a CDPR is studied in Section 5. In Section 6, the robustness of the proposed strategies to model uncertainties is discussed. Finally, Section 7 summarizes this paper.

2. Preliminaries

In this section, DDPG [5] is introduced to provide a basis for the development of the hybrid DDPG strategy and the end-to-end DDPG strategy in Section 3. Moreover, the kinematics and dynamics of a CDPR are presented. The hybrid DDPG strategy relies on the inverse dynamics equation of the CDPR to calculate the optimal tension distribution of cables. The end-to-end DDPG strategy relies on the DDPG algorithm to learn the optimal tension distribution of cables from trial-and-error.

2.1. Deep deterministic policy gradient

As a model-free DRL algorithm, DDPG aims to solve reinforcement learning problems with continuous state and action [5]. DDPG has an actor-critic architecture that combines the DQN [17] and the deterministic policy gradient (DPG) [18]. DDPG utilizes four DNNs (i.e., actor network, actor-target network, critic network, and critic-target network) to approximate two policies (i.e., behavior policy and target policy), as shown in Table I.

The actor network of DDPG approximates a behavior policy, denoted by μ. The output of the actor network parameterized by θ^{μ} is

$$\boldsymbol{a}_t = \mu(\boldsymbol{s}_t|\ \theta^{\mu}) \tag{1}$$

where t represents a specific time step. $\boldsymbol{s}_t$ represents the state of the actor network at time step t. The behavior policy function uses a deterministic policy instead of a stochastic policy. The critic network parameterized by θ^{Q} approximates a value function as

Table I
Networks and policies of DDPG.

Network	Input	Output	Policy
Actor Network	$\boldsymbol{s}_t$	$\boldsymbol{a}_t = \mu(\boldsymbol{s}_t\|\ \theta^{\mu})$	Behavior Policy
Actor-Target Network	$\boldsymbol{s}_{t+1}$	$\boldsymbol{a}'_t = \mu'(\boldsymbol{s}_{t+1}\|\ \theta^{\mu'})$	Target Policy
Critic Network	$\boldsymbol{s}_t, \tilde{\boldsymbol{a}}_t$	$Q = Q(\boldsymbol{s}_t, \tilde{\boldsymbol{a}}_t\|\ \theta^{Q})$	Behavior Policy
Critic-Target Network	$\boldsymbol{s}_{t+1}, \boldsymbol{a}'_t$	$Q' = Q'(\boldsymbol{s}_{t+1}, \boldsymbol{a}'_t\|\ \theta^{Q'})$	Target Policy

$$Q = Q(\boldsymbol{s}_t, \tilde{\boldsymbol{a}}_t|\ \theta^{Q}) \tag{2}$$

where $\tilde{\boldsymbol{a}}_t$ can be expressed as

$$\tilde{\boldsymbol{a}}_t = \boldsymbol{a}_t + \boldsymbol{N}_t \tag{3}$$

where $\boldsymbol{N}_t$ represents the exploration noise.

The actor-target network approximates a target policy, denoted by μ'. The output of the actor-target network parameterized by $\theta^{\mu'}$ is

$$\boldsymbol{a}'_t = \mu'(\boldsymbol{s}_{t+1}|\ \theta^{\mu'}) \tag{4}$$

where $\boldsymbol{s}_{t+1}$ represents the state of the actor-target network at time step $t+1$. The critic-target network parameterized by $\theta^{Q'}$ approximates a value function

$$Q' = Q'(\boldsymbol{s}_{t+1}, \boldsymbol{a}'_t|\ \theta^{Q'}) \tag{5}$$

Data of transitions (i.e., state $\boldsymbol{s}_t$, action $\boldsymbol{a}_t$, reward r_t, and next state $\boldsymbol{s}_{t+1}$) are stored in an experience replay buffer. A random mini-batch of N data of transitions ($\boldsymbol{a}_i$, $\boldsymbol{s}_i$, r_i, $\boldsymbol{s}_{i+1}$) is selected from the experience replay buffer to train the four DNNs. The critic network is updated by minimizing the loss L which is defined as

$$L = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - Q\left(\boldsymbol{s}_i, \boldsymbol{a}_i|\theta^{Q}\right)\right)^2 \tag{6}$$

where y_i can be expressed as

$$y_i = r_i + \gamma Q'(\boldsymbol{s}_{i+1}, \boldsymbol{\mu'}(\boldsymbol{s}_{i+1}|\theta^{\mu'})|\theta^{Q'}) \tag{7}$$

where $\gamma \in (0,\ 1)$ denotes a discount factor. The actor network is updated with respect to the gradient of the expected performance objective J as

$$\nabla_{\theta^{\mu}} J \approx \frac{1}{N}\sum_{i=1}^{N}\left[\nabla_a Q\left(\boldsymbol{s}, \boldsymbol{a}|\theta^{Q}\right)|_{\boldsymbol{s}_i, \mu(\boldsymbol{s}_i)} \nabla_{\theta^{\mu}} \mu(\boldsymbol{s}|\theta^{\mu})|_{\boldsymbol{s}_i}\right] \tag{8}$$

After updating the actor network and the critic network, the two corresponding target networks are updated by

$$\theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'} \tag{9}$$

$$\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'} \tag{10}$$

where update rate $\tau \ll 1$.

2.2. Kinematics and dynamics of a CDPR

A CDPR is a robot driven by a set of cables in parallel. An example CDPR for the rehabilitation of ankles is shown in Fig. 1. The end-effector of the CDPR is a brace worn on the foot while the base of the CDPR is a cuff on the shank. Four cables connect the end-effector to the base. Anchor points are points where cables connect the base and the end-effector of the CDPR. The example CDPR has three degrees of freedom (DOFs).

The kinematics architecture of a CDPR with n cables is shown in Fig. 2. The base frame F_b is mounted on the base of the CDPR and the end-effector frame F_e is mounted on the end-effector of the CDPR. The positions of the attaching points A_i and $B_i$are represented by vectors $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ in the base frame, respectively. $\boldsymbol{u}_i$ is

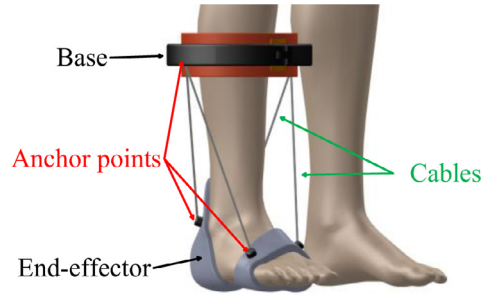

Fig. 1. A CDPR for the rehabilitation of ankles.

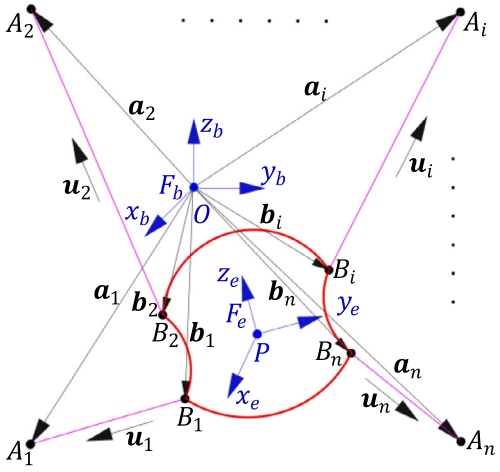

Fig. 2. Kinematics notations of a CDPR.

the unit vector along the ith cable. Based on the above kinematics notations, the Jacobian of the CDPR can be expressed as [19]

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \dots & \boldsymbol{u}_n \\ \boldsymbol{b}_1 \times \boldsymbol{u}_1 & \boldsymbol{b}_2 \times \boldsymbol{u}_2 & \dots & \boldsymbol{b}_n \times \boldsymbol{u}_n \end{bmatrix}^T \tag{11}$$

The equation of motion of the end-effector can be obtained using the Newton–Euler Formulation as [20,21]

$$\boldsymbol{M}(\boldsymbol{X})\ddot{\boldsymbol{X}} + \boldsymbol{C}\left(\boldsymbol{X}, \dot{\boldsymbol{X}}\right)\dot{\boldsymbol{X}} + \boldsymbol{G}(\boldsymbol{X}) + \boldsymbol{E}(\boldsymbol{X}) = \boldsymbol{w} \tag{12}$$

where $\boldsymbol{X}$ denotes the pose (i.e., both position and orientation) of the end-effector. $\dot{\boldsymbol{X}}$ denotes the twist of the end-effector. $\ddot{\boldsymbol{X}}$ represents the acceleration of the end-effector. $\boldsymbol{M}(\boldsymbol{X})$ is the mass matrix. $\boldsymbol{C}(\boldsymbol{X}, \dot{\boldsymbol{X}})$ is the Coriolis and centripetal matrix. $\boldsymbol{G}(\boldsymbol{X})$ represents the gravity matrix. $\boldsymbol{E}(\boldsymbol{X})$ represents the external wrench matrix. $\boldsymbol{w}$ is the wrench vector applied on the end-effector by cables. Assuming there is no collision among cables and the end-effector, one has the inverse dynamics equation of the CDPR

$$\boldsymbol{w} = -\boldsymbol{J}^T\boldsymbol{\tau} \tag{13}$$

where $\boldsymbol{\tau}$ represents the vector of cable tensions. CDPRs are commonly classified as either fully-constrained or under-constrained ones [22]. For a fully-constrained CDPR in its force-closure workspace [23], $\boldsymbol{\tau}$ can be decomposed into two parts [24–26], namely,

$$\boldsymbol{\tau} = \boldsymbol{\tau}_e + \boldsymbol{\tau}_a \tag{14}$$

where $\boldsymbol{\tau}_e$ is the elastic cable tension vector that generates the wrench $\boldsymbol{w}$ acting on the end-effector. $\boldsymbol{\tau}_a$ is the antagonistic cable tension vector that creates a zero wrench on the end-effector

$$-\boldsymbol{J}^T\boldsymbol{\tau}_a = 0 \tag{15}$$

When a fully-constrained CDPR works within its force-closure workspace, an infinite number of feasible tension distributions exist [27,28]. The tension distribution with the minimal Euclidean norm of $\boldsymbol{\tau}$ is the optimal tension distribution of cables [29–31].

3. Control strategies

Fig. 3a illustrates how a CDPR is controlled to move to a target pose. Given the target pose, the controller of the CDPR needs to calculate a set of target cable tensions. Actuators are used to physically deliver the set of target cable tensions to drive the CDPR to the target pose. The task of the controller of the CDPR, calculating a set of target cable tensions from a given target pose, can be divided into two subtasks. The first subtask is to calculate the target wrench in the task space to drive the CDPR to the target pose. Since cables can be pulled only, the target wrench in the task space has to be converted to a set of target cable tensions in the joint space, which is the second subtask. Only when such a set of target cable tensions is available can actuators reel cables in or out to drive the CDPR to the target pose. In this section, a hybrid DDPG strategy and an end-to-end DDPG strategy are developed to control the CDPR. The hybrid DDPG strategy accomplishes the first subtask by DDPG and the second subtask by the inverse dynamics equation of the CDPR in (13). The end-to-end DDPG strategy accomplishes the whole task (i.e., both the first and the second subtasks) of the controller of the CDPR using DDPG.

The flow diagram of the hybrid DDPG strategy is shown in Fig. 3b. The hybrid DDPG strategy integrates DDPG and the inverse dynamics equation of a CDPR. The hybrid DDPG strategy accomplishes the first subtask of the controller of the CDPR by DDPG. To accomplish the first subtask, the DDPG in the hybrid DDPG strategy employs the following state variables: the pose of the CDPR at a certain time step, the difference between the target pose and the pose of the CDPR at a certain time step, and the velocity of the CDPR at a certain time step. Action variables of the DDPG in the hybrid DDPG strategy is the target wrench in the task space that is expected to drive the CDPR to the target pose. Once the first subtask is accomplished, the DDPG of the hybrid DDPG strategy outputs the target wrench in the task space. The second subtask takes such a target wrench in the task space and converts it to a set of target cable tensions. The second subtask is accomplished by solving the inverse dynamics equation of the CDPR in (13). When converting the target wrench in the task space to a set of target cable tensions of a fully-constrained CDPR by solving (13), there is an infinite number of feasible tension distributions. The optimal tension distribution is calculated using the pseudo-inverse of Jacobian [32]. Once the optimal tension distribution is obtained, actuators physically deliver the set of target cable tensions. Algorithm 1 illustrates the implementation of the hybrid DDPG strategy.

The flow diagram of the end-to-end DDPG strategy is shown in Fig. 3c. The end-to-end DDPG strategy accomplishes the whole task (i.e., both the first and the second subtasks) of the controller of the CDPR using DDPG. State variables of the DDPG in this strategy consist of the pose of the CDPR at a certain time step, the difference between the target pose and the pose of the CDPR at a certain time step, and the velocity of the CDPR at a certain time step as well. However, action variables of the DDPG in this strategy are a set of target cable tensions in the joint space, rather than a target wrench in the task space. The set of target cable tensions is derived based on the behavior policy of the DDPG. Algorithm 2 illustrates the implementation of the end-to-end DDPG strategy.

4. Training of DDPG in control strategies

DDPG in the hybrid DDPG strategy and the end-to-end DDPG strategy has to be trained before they can be used to control a

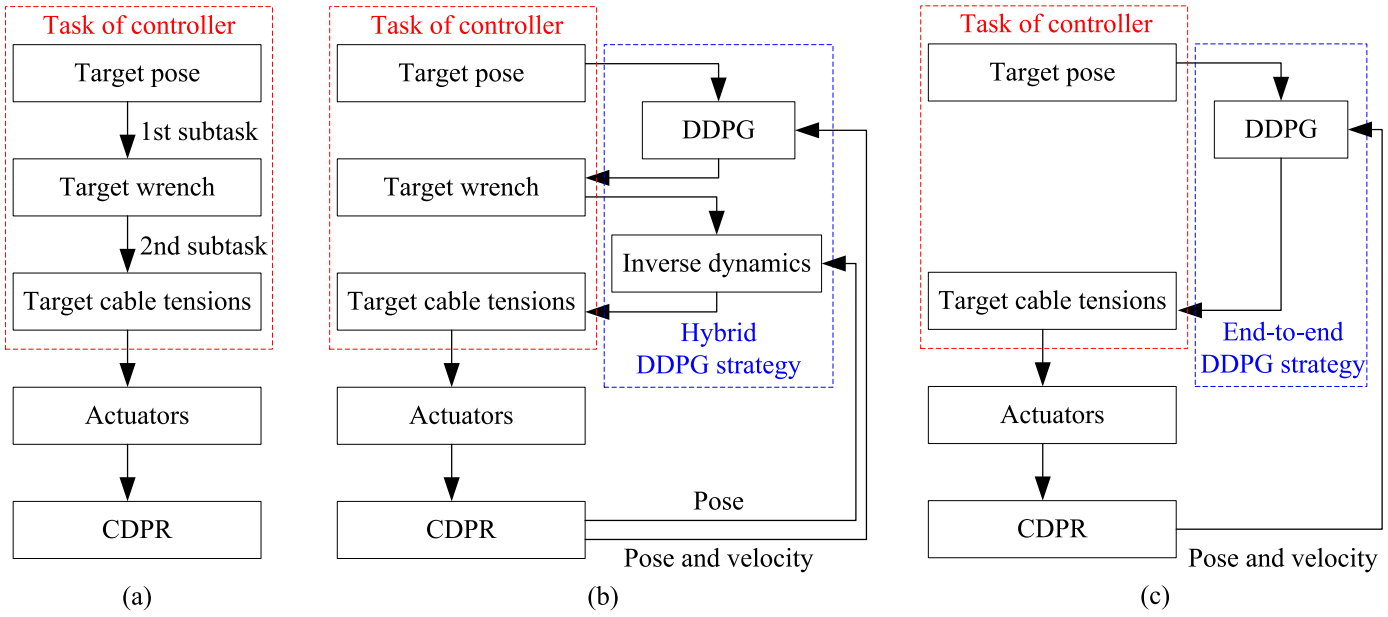

Fig. 3. (a) The task of the controller of a CDPR in controlling the CDPR to a target pose; (b) The flow diagram of the hybrid DDPG strategy; and (c) The flow diagram of the end-to-end DDPG strategy.

Algorithm 1
Implementation of the hybrid DDPG strategy in controlling a CDPR.

Randomly initialize the actor network and the critic network
Initialize the actor-target network and the critic-target network with the weights of the actor network and the critic network, respectively
Initialize an experience replay buffer
for number of episodes **do**
    Reset the CDPR
    Randomly set a target pose
    Record the $\boldsymbol{s}_t$
    for maximum number of time steps **do**
        Calculate $\boldsymbol{a}_t$ (target wrench) according to the current policy and exploration noise
        Calculate the optimal tension distribution based on the inverse dynamics equation of the CDPR according to (13)
        Execute the optimal tension distribution and observe reward and $\boldsymbol{s}_{t+1}$
        if the CDPR is out of the training workspace **then**
            Break;
        if the CDPR reaches the target pose **then**
            Break;
        Stack the data (i.e., states, actions, reward, and next states) in the experience replay buffer
        Select a mini-batch of data from the experience replay buffer
        Update the critic network via minimizing the training loss
        Update the actor network using the sampled policy gradient
        Update the actor- and critic-target networks
    end for
end for

Algorithm 2
Implementation of the end-to-end DDPG strategy in controlling a CDPR.

Randomly initialize the actor network and the critic network
Initialize the actor-target network and the critic-target network with the weights of the actor network and the critic network, respectively
Initialize an experience replay buffer
for number of episodes **do**
    Reset the CDPR
    Randomly set a target pose
    Record the $\boldsymbol{s}_t$
    for maximum number of time steps **do**
        Calculate $\boldsymbol{a}_t$ (target cable tensions) according to the current policy and exploration noise
        Execute $\boldsymbol{a}_t$and observe reward and $\boldsymbol{s}_{t+1}$
        if the CDPR is out of the training workspace **then**
            Break;
        if the CDPR reaches the target pose **then**
            Break;
        Stack the data (i.e., states, actions, reward, and next states) in the experience replay buffer
        Select a mini-batch of data from the experience replay buffer
        Update the critic network via minimizing the training loss
        Update the actor network using the sampled policy gradient
        Update the actor- and critic-target networks
    end for
end for

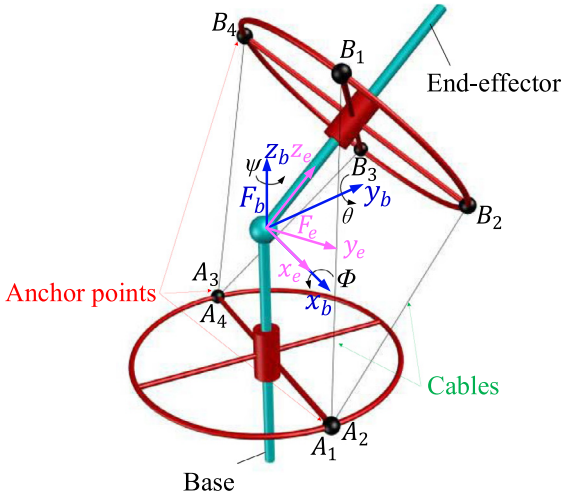

Fig. 4. Notations of the example CDPR.

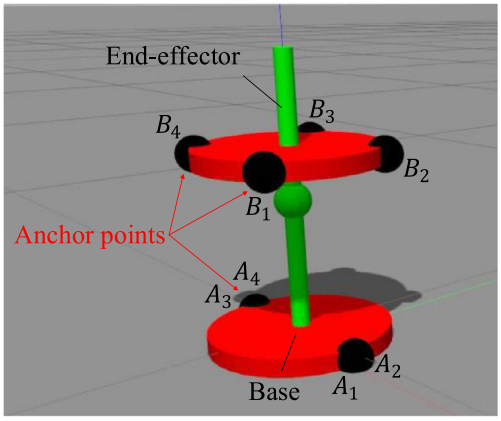

Fig. 5. The model of the CDPR in the Gazebo simulator.

CDPR. Training a DRL algorithm on a real robot may damage the robot, especially when the DRL strategy has not been well-trained in the initial stage. Thus, a robot model in a simulator, rather than a real robot, is highly desirable as simulations are always safe for the real robot [16]. Moreover, a simulator is also helpful in developing and fine-tuning the DRL strategy because the iteration of simulations is much faster than that of real experiments [16]. In this section, DDPG in the proposed strategies is trained to control an example CDPR using the Gazebo simulator.

As shown in Fig. 4, the example CDPR has three rotational DOFs and four cables. F_b represents the base frame and F_e represents the end-effector frame. The pose of the end-effector with respect to the base (i.e., the pose of the CDPR) is described by a vector of three Euler angles $[\phi\,\theta\,\psi]$ with a $\psi-\theta-\phi$ (i.e., yaw-pitch-roll) sequence [33]. DDPG in the proposed strategies is trained within a workspace, called training workspace, of the CDPR. The training workspace in this study is the set of poses that the CDPR can reach with the roll, pitch, and yaw of the CDPR within $[-20,\ 20]$ (unit: deg). Gravity is neglected in this study to simplify the analysis.

A model of the example CDPR is created in the Gazebo simulator, as shown in Fig. 5. The four cables of the CDPR are not shown in Fig. 5. Cable tension is applied between a pair of anchor points of the CDPR. The parameters and the positions of the anchor points of the CDPR are shown in Tables II and III, respectively.

Table II
Parameters of the CDPR.

Parameter	Value
Mass of the end-effector	3 kg
Moment of inertia of end-effector about x axis of F_e	0.02 kg·m^2
Moment of inertia of end-effector about y axis of F_e	0.02 kg·m^2
Moment of inertia of end-effector about z axis of F_e	0.02 kg·m^2

4.1. Training setup

4.1.1. State setup

In this study, the state variables of the hybrid DDPG strategy are the same as those of the end-to-end DDPG strategy. The state variables of the proposed strategies at time step t can be expressed as

$$\boldsymbol{s}_t = \begin{bmatrix} \boldsymbol{X} & \dot{\boldsymbol{X}} & \Delta\boldsymbol{X} \end{bmatrix} \tag{16}$$

where $\boldsymbol{X} = [\ \phi \quad \theta \quad \psi\ ]$ is the pose of the CDPR at time step t. $\dot{\boldsymbol{X}} = [\ \dot{\phi} \quad \dot{\theta} \quad \dot{\psi}\ ]$ is the angular velocity of the CDPR at time step t. $\Delta\boldsymbol{X}$ is the difference between the target pose of the CDPR, denoted as $\boldsymbol{X}^*$, and the pose of the CDPR at time step t. Thus, $\Delta\boldsymbol{X}$ can be expressed as

$$\Delta\boldsymbol{X} = \boldsymbol{X}^* - \boldsymbol{X} \tag{17}$$

It should be noted that $\boldsymbol{s}_t$ is a vector having nine scalar elements in this study.

4.1.2. Action setup

The action variables of the hybrid DDPG strategy are different from those of the end-to-end DDPG strategy. The action variables of the hybrid DDPG strategy at time step t can be expressed as

$$\boldsymbol{a}_t^w = \begin{bmatrix} w_x & w_y & w_z \end{bmatrix} \tag{18}$$

where w_x, w_y, and w_z represent the target torques about the x, y, and z axes of F_e, respectively. The action variables of the end-to-end DDPG strategy at time step t can be expressed as

$$\boldsymbol{a}_t^\tau = \begin{bmatrix} \tau_1 & \tau_2 & \tau_3 & \tau_4 \end{bmatrix} \tag{19}$$

where τ_i $(i = 1,\ 2,\ 3,\ 4)$ is the target cable tension in the ith cable.

4.1.3. Reward setup

Rewards of the hybrid DDPG strategy and the end-to-end DDPG strategy used in this study are defined below. For the hybrid DDPG strategy, the reward is designed as

$$r_t^w = -2\|\Delta\boldsymbol{X}\| - \left\|\dot{\boldsymbol{X}}\right\| \tag{20}$$

where $\|*\|$ represents the Euclidean norm of $*$. r_t^w has two terms. The first term is defined by the difference between the target pose and the pose of the CDPR at time step t. The second term is defined by the velocity of the CDPR at time step t. In this way, a larger reward is granted if the CDPR is closer to the target pose or the velocity of the CDPR is smaller. If the CDPR reaches the target pose with a full stop, the maximum reward (i.e., $r_t^w = 0$) is granted.

For the end-to-end DDPG strategy, whether the strategy can learn the optimal tension distribution of cables or not depends on the reward. Thus, two rewards are designed and tested, aiming to find a reward with which the end-to-end DDPG strategy can learn the optimal tension distribution of cables. The two rewards of the end-to-end DDPG strategy are designed as

$$r_t^{\tau 1} = -2\|\Delta\boldsymbol{X}\| - \left\|\dot{\boldsymbol{X}}\right\| \tag{21}$$

$$r_t^{\tau 2} = -2\|\Delta\boldsymbol{X}\| - \left\|\dot{\boldsymbol{X}}\right\| - 0.2\|\boldsymbol{a}_t^\tau\| \tag{22}$$

Table III
Positions of anchor points of the CDPR (unit: m).

Pair of anchor points	Positions of anchor points on the base	Positions of anchor points on the end-effector
1	$A_1: [0.1, 0.0, -0.15]^T$	$B_1: [0.0707, -0.0707, 0.05]^T$
2	$A_2: [0.1, 0.0, -0.15]^T$	$B_2: [0.0707, 0.0707, 0.05]^T$
3	$A_3: [-0.1, 0.0, -0.15]^T$	$B_3: [-0.0707, 0.0707, 0.05]^T$
4	$A_4: [-0.1, 0.0, -0.15]^T$	$B_4: [-0.0707, -0.0707, 0.05]^T$

Table IV
Architecture of the four networks of the DDPG.

	Actor and actor-target networks	Critic and critic-target networks
Number of inputs	9	$9 + N_a$
Activation function 1	ReLU	ReLU
Number of units in layer 1	90	150
Activation function 2	ReLU	ReLU
Number of units in layer 2	60	120
Activation function 3	tanh	None
Number of outputs	N_a	1

Table V
Hyper-parameters used to train the DDPG.

Hyper-Parameter	Value
Maximum number of time steps for each episode	1000
Learning rate of the actor and actor-target networks	0.001
Learning rate of the critic and critic-target networks	$0.0001 \rightarrow 0.00005$
Discount factor γ	0.95
Update rate (target) τ	0.001
Size of the experience replay buffer	10,000
Size of mini-batch	1024
Maximum magnitude of random exploration noise (i.e., an element of $\boldsymbol{N}_t$)	$\frac{\text{range of an action}}{\text{number of episodes}+2}$

where $r_t^{\tau 1}$ is the same as r_t^w. Compared to $r_t^{\tau 1}$, $r_t^{\tau 2}$ has a third term, namely, the Euclidean norm of the vector of cable tensions or the action variables of the end-to-end DDPG strategy defined in (19). Such a cable tension term gives higher rewards to actions producing smaller tensions. The design of $r_t^{\tau 2}$ is inspired by the reward consisting of both balancing and goal-oriented terms for a simulated bicycle riding task in [34]. With $r_t^{\tau 2}$, a larger reward is granted if the CDPR is closer to the target pose; the velocity of the CDPR is smaller; or the Euclidean norm of the vector of cable tensions is smaller. In this manner, the end-to-end-DDPG strategy's goal is not only to accomplish the task of controlling a CDPR, but also achieve the optimal tension distribution of cables. r_t^w, $r_t^{\tau 1}$, and $r_t^{\tau 2}$ are possible to be further optimized according to studies of multi-objective reinforcement learning that aims to find compromising solutions balancing different objectives to RL problems [35]. Multi-objective reinforcement learning has been successfully applied to control variable speed wind turbines achieving the optimal balance of power generation stability and rotor angular speed in [36].

4.1.4. Networks and hyper-parameters setup

In this study, the four networks of the DDPG used by both proposed strategies are fully connected neural networks with two hidden layers. The architecture of the four DNNs is shown in Table IV and the hyper-parameters used to train the DDPG are shown in Table V. The architecture of the four DNNs of the DDPG and the hyper-parameters used by both proposed strategies are the same except the number of outputs of the actor network and the actor-target network and the number of inputs of the critic network and the critic-target network. N_a in Table IV represents the number of actions. In this study, $N_a = 3$ for the hybrid DDPG strategy and $N_a = 4$ for the end-to-end DDPG strategy. In Table V, the learning rates of the critic network and the critic-target network change from 0.0001 to 0.00005 if the training loss is less than 0.001. The data (i.e., states, actions, reward, and next states) used to train DDPG in both proposed strategies are collected from the Gazebo simulator. For each episode, the CDPR starts at $\boldsymbol{X} = [0 \quad 0 \quad 0]$ (unit: deg). An episode ends if the CDPR reaches a target pose, the maximum number of 1000 time steps is reached, or the CDPR goes out of the training workspace.

4.2. Training

In this study, it is assumed that elements of $\boldsymbol{a}_t^\tau$ in (19) (i.e., the individual cable tensions) are within [0, 1] (unit: N) and elements of $\boldsymbol{a}_t^w$ in (18) (i.e., the target wrenches about the x, y, and z axes of F_e) are within $[-0.1, 0.1]$ (unit: Nm). The Adam optimizer [37] is utilized in training. With the training setup in Section 4.1, DDPG in the proposed strategies is trained based on a CDPR model in the Gazebo simulator. The frequency of the Gazebo simulator is set to 100 Hz and the time step of the DDPG is 0.1 s. The average rewards in every episode are shown in Fig. 6. It is shown that the hybrid DDPG strategy converges within 70 episodes, while the end-to-end DDPG strategy with $r_t^{\tau 1}$ and $r_t^{\tau 2}$ converges within 100 and 400 episodes, respectively. Therefore, the hybrid DDPG strategy learns faster than the end-to-end DDPG strategy even when they use the same reward.

According to [16], a complicated task becomes easier to learn if some of its subtasks have already been accomplished. For the hybrid DDPG strategy, the second subtask of converting the target wrench in the task space to a set of target cable tensions in the joint space is accomplished by solving the inverse dynamics equation in (13). Thus, the DDPG of the hybrid DDPG strategy can focus on the first subtask while the DDPG of the end-to-end DDPG strategy has to deal with both the first and the second subtasks. This explains why the hybrid DDPG strategy learns faster than the end-to-end DDPG strategy. Moreover, the number of actions of the DDPG in the end-to-end DDPG strategy is four while that in the hybrid DDPG strategy is three. The reduced number of actions is another possible reason that the hybrid DDPG strategy learns faster than the end-to-end DDPG strategy.

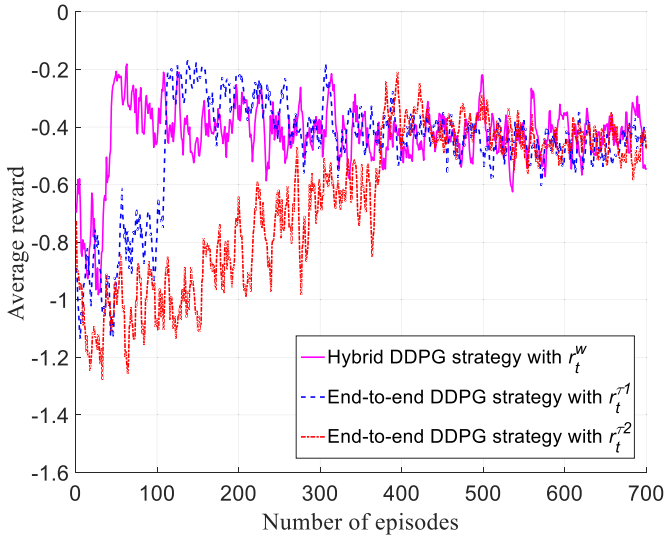

Fig. 6. Average rewards of DDPG in the proposed strategies.

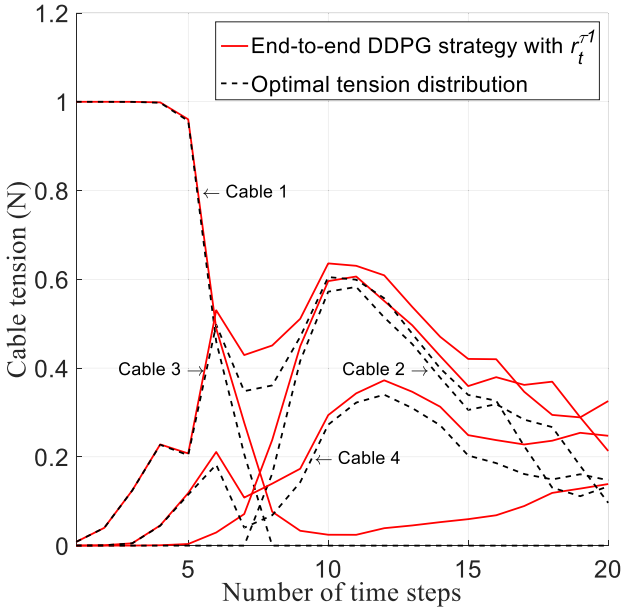

Fig. 7. Cable tensions of the CDPR controlled by the end-to-end DDPG strategy with $r_t^{\tau 1}$.

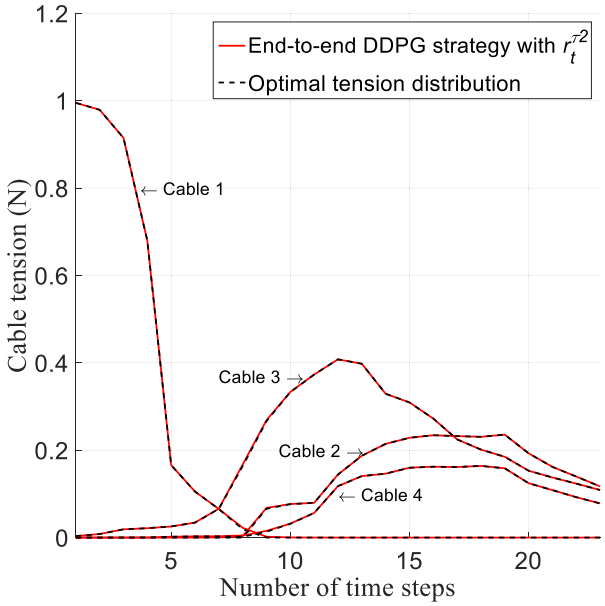

Fig. 8. Cable tensions of the CDPR controlled by the end-to-end DDPG strategy with $r_t^{\tau 2}$.

5. Optimal tension distribution of cables

The example CDPR concerned in this study is a fully-constrained CDPR. A fully-constrained CDPR in its force-closure workspace has an infinite number of feasible tension distributions. Therefore, how to obtain the optimal tension distribution [29–31] of cables is studied for the example CDPR in this section. The hybrid DDPG strategy calculates the optimal tension distribution of cables by solving the inverse dynamics equation in (13). The end-to-end DDPG strategy has to learn the optimal tension distribution of cables on its own. This section demonstrates that, with a proper reward, the end-to-end DDPRG strategy can learn the optimal tension distribution of cables as well as the hybrid DDPG strategy obtains the optimal tension distribution of cables using a non-learning-based approach.

The tension distributions of the CDPR controlled by the end-to-end DDPG strategy with $r_t^{\tau 1}$ and $r_t^{\tau 2}$ in reaching a randomly selected target pose $\boldsymbol{X} = [9\ 12\ 15]$ (unit: deg) is studied. The end-to-end DDPG strategy with $r_t^{\tau 2}$ whose third term is about cable tensions is expected to be able to learn the optimal tension distribution of cables. The tension distributions of the end-to-end DDPG strategy with $r_t^{\tau 1}$ and $r_t^{\tau 2}$ are shown in Figs. 7 and 8, respectively. The optimal tension distribution (i.e., the dash lines in Figs. 7 and 8) is obtained by solving the inverse dynamics equation in (13). It is shown that the end-to-end DDPG strategy with $r_t^{\tau 2}$ is able to learn the optimal tension distribution, while the end-to-end DDPG strategy with $r_t^{\tau 1}$ cannot. Therefore, with a proper reward (e.g., $r_t^{\tau 2}$), the end-to-end DDPG strategy can learn the optimal tension distribution of a fully-constrained CDPR.

6. Robustness to model uncertainty

The model of a robot used to train a DRL algorithm may not capture all the details of the real robot in practice [16]. For example, positions of anchor points of the CDPR in Fig. 1 may change from setup to setup due to the wearing inconsistency of the cuff and the brace. As a result, the models of the CDPR for different setups may be slightly different due to the wearing inconsistency. A setup or model of the CDPR to be controlled by the proposed strategies may be slightly different from the setup or model of the same CDPR based on which DDPG in the proposed strategies is trained. It would be ideal if the proposed strategies whose DDPG is trained using one model of the CDPR can be used to control the CDPR whose model has slightly changed. Such an adaptive capability requires the proposed strategies to be robust to model uncertainties [16].

The robustness of the proposed strategies to model uncertainties is investigated in this section based on a pose-tracking test and a trajectory-tracking test. The proposed strategies whose DDPG is trained using one of the models of the CDPR in the Gazebo simulator are first used to control the CDPR with the same model. In this case, there is no model uncertainty or difference between the model based on which DDPG is trained and the model to be controlled by DDPG. This works as the baseline for the study of the robustness of the proposed strategies to model uncertainties. Then, the proposed strategies are used to control the CDPR whose model has been slightly changed to test the robustness of the proposed strategies to model uncertainties.

Table VI
Outcomes of the pose-tracking test.

Strategy	Number of target poses the CDPR reaches successfully	Number of target poses the CDPR fails to reach	Success rate
Hybrid DDPG strategy with the reward of r_t^w	993	7	99.3%
End-to-end DDPG strategy with the reward of $r_t^{\tau 1}$	978	22	97.8%
End-to-end DDPG strategy with the reward of $r_t^{\tau 2}$	995	5	99.5%

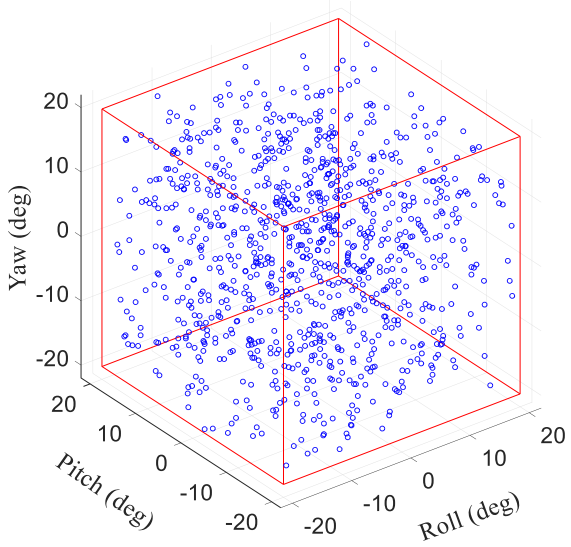

Fig. 9. Randomly selected target poses.

In the pose-tracking test, the CDPR is controlled by the proposed strategies to move from $\boldsymbol{X} = [0\ 0\ 0]$ (unit: deg) to 1000 randomly selected target poses within the training workspace, as shown in Fig. 9. A pose-tracking test is considered successful if the CDPR is able to reach a target pose (i.e., the Euclidean norm of the tracking error defined by $\|\Delta \boldsymbol{X}\|$ is less than 1.8°) and it takes no more than 150 time steps (i.e., 15 s). Otherwise, the pose-tracking test is considered failed.

In the trajectory-tracking test, the target trajectory to be tracked is designed as

$$\begin{cases} \phi = 45\sin(0.01\pi j)/\pi \\ \theta = 22.5\cos(0.01\pi j)/\pi - 22.5/\pi \\ \psi = 45\sin(0.02\pi j)/\pi \end{cases} \tag{23}$$

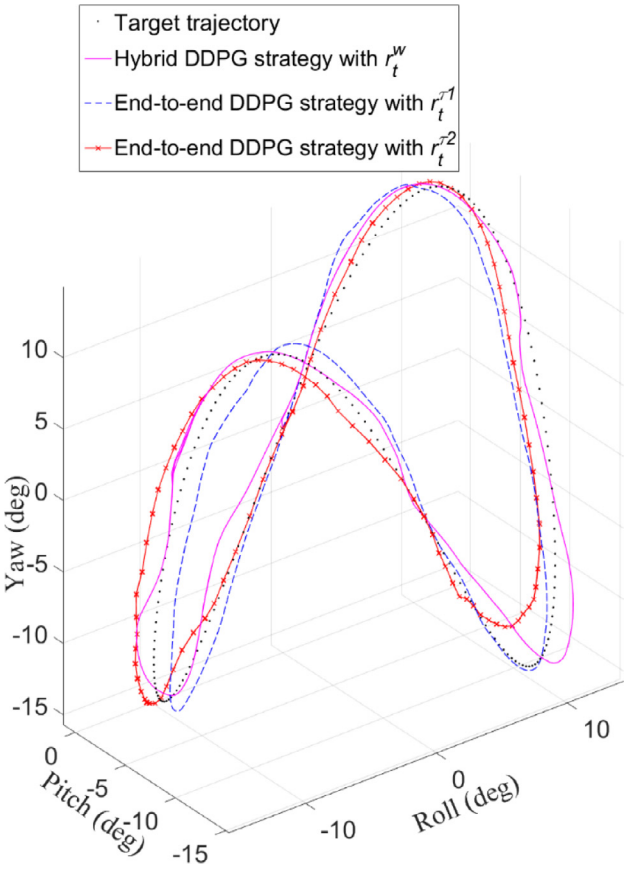

Fig. 11. Trajectories of the CDPR without model uncertainty.

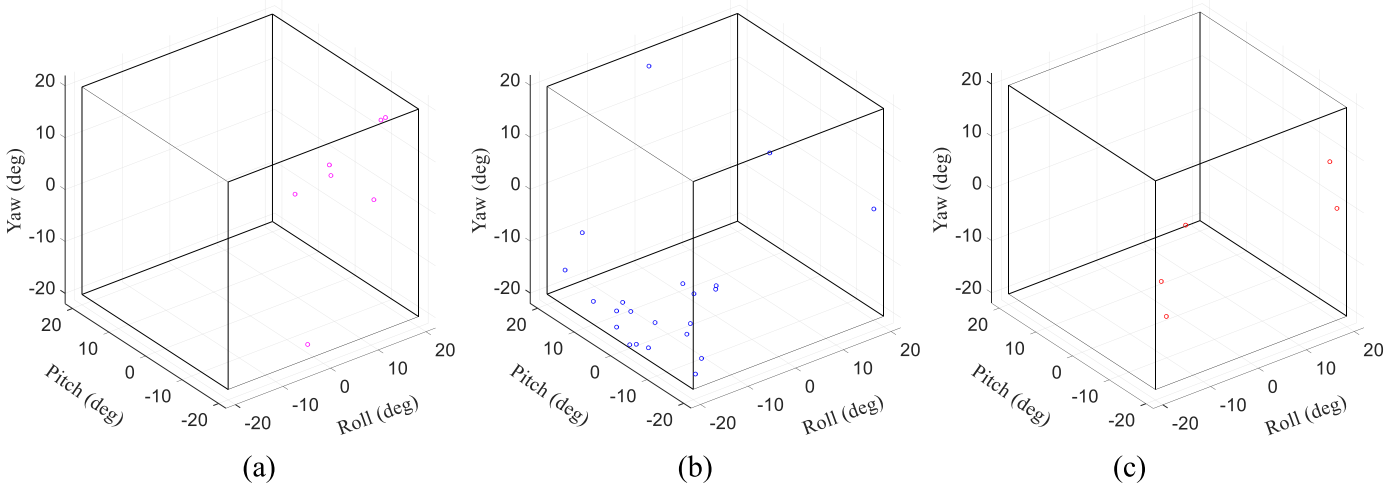

Fig. 10. Target poses of the CDPR fails to be reached using the proposed strategies in controlling the CDPR without model uncertainty: (a) the hybrid DDPG strategy; (b) the end-to-end DDPG strategy with the reward of $r_t^{\tau 1}$; and (c) the end-to-end DDPG strategy with the reward of $r_t^{\tau 2}$.

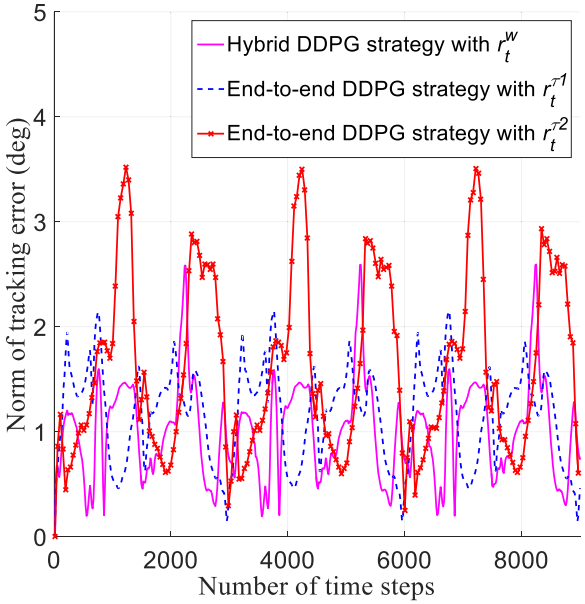

Fig. 12. Tracking error of the CDPR without model uncertainty.

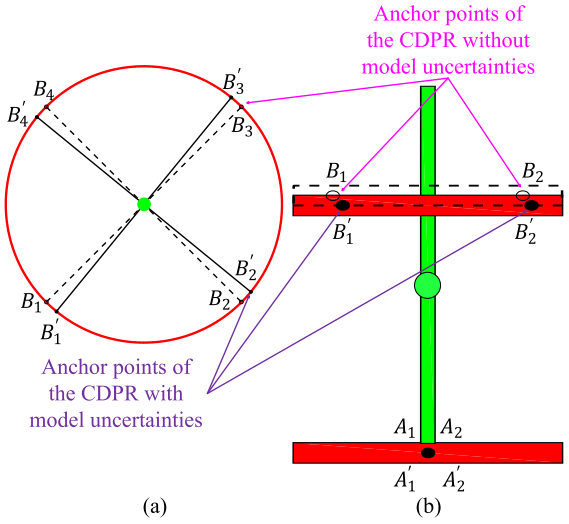

Fig. 13. Position errors of anchor points of the CDPR: a) top view; and b) side view.

where $j = 0, 1, \ldots, 599$ and j is updated in every 15 time steps (i.e., 1.5 s).

6.1. Evaluate control strategies on a CDPR without model uncertainty

The proposed strategies whose DDPG is trained using a model of the CDPR in the Gazebo simulator are used to control the CDPR with the same model. For the proposed strategies, the model of the CDPR based on which DDPG in the strategies is trained is regarded as the model of the CDPR without model uncertainty. The outcomes of the pose-tracking test are shown in Table VI. The hybrid DDPG strategy and the end-to-end DDPG strategy with the reward of $r_t^{\tau 2}$ perform well in the pose-tracking test. The success rate of the end-to-end DDPG strategy with the reward of $r_t^{\tau 1}$ is slightly lower than those of the hybrid DDPG strategy and the end-to-end DDPG strategy with the reward of $r_t^{\tau 2}$. To investigate why the proposed strategies cannot control the CDPR to reach a target pose, the target poses that the proposed strategies cannot control the CDPR to reach are plotted in Fig. 10. It turns out that all these target poses are close to the boundary of the training workspace. If a target pose is close to the boundary of the training workspace, the CDPR may move out of the training workspace when moving towards such a target pose. If the CDPR is out of the training workspace, it is very likely that the CDPR cannot move back to the training workspace because the training of DDPG in the strategies is limited to the training workspace. Thus, poses close to the boundary of the training workspace are more challenging to track than those close to the center of the training workspace. This suggests that the training workspace should be larger than the workspace used in a control task in practice.

Fig. 11 shows the trajectories of the CDPR controlled by the proposed strategies to track the target trajectory defined in (23). The proposed strategies can control the CDPR to track the target trajectory indeed. Fig. 12 shows the Euclidean norm of the tracking error when the CDPR is controlled to track the target trajectory. The hybrid DDPG strategy and the end-to-end DDPG strategy with the reward of $r_t^{\tau 1}$ have smaller tracking error than the end-to-end DDPG strategy with the reward of $r_t^{\tau 2}$. Since the magnitude of cable tensions is included in the reward of $r_t^{\tau 2}$, the end-to-end

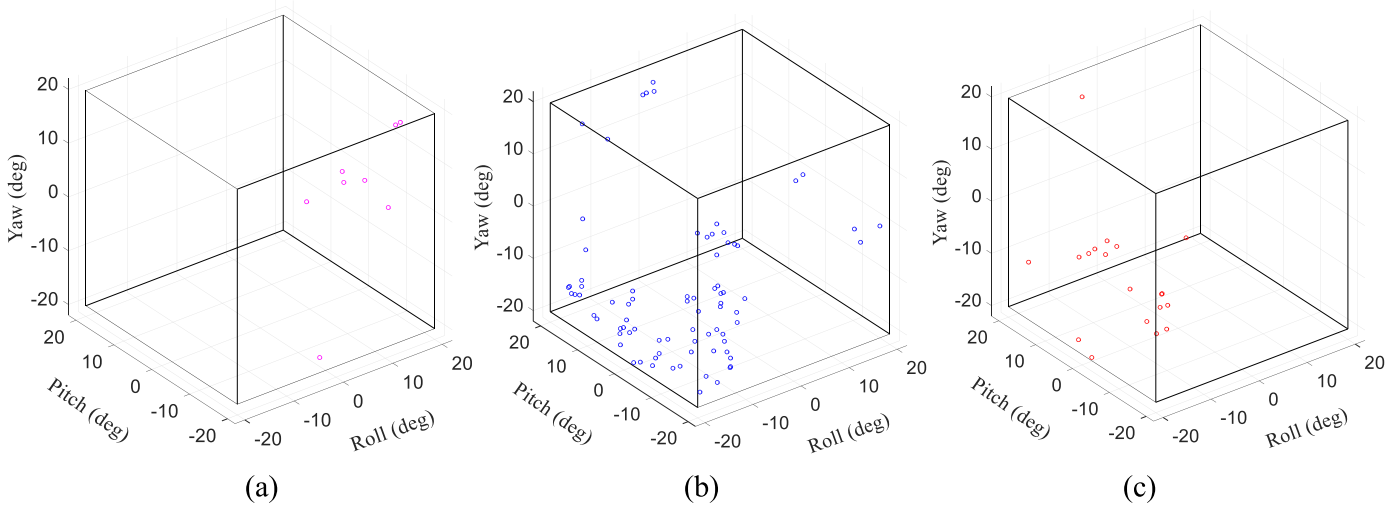

Fig. 14. Target poses of the CDPR fails to be reached using the proposed strategies in controlling the CDPR with model uncertainties: (a) the hybrid DDPG strategy; (b) the end-to-end DDPG strategy with the reward of $r_t^{\tau 1}$; and (c) the end-to-end DDPG strategy with the reward of $r_t^{\tau 2}$.

Table VII
Positions of anchor points of the CDPR with model uncertainties (Unit: m).

Pair of anchor points	Positions of anchor points on the base	Positions of anchor points on the end-effector
$1'$	$A'_1: [0.1,\ 0.0,\ -0.15]^T$	$B'_1: [0.0774,\ -0.0633,\ 0.04]^T$
$2'$	$A'_2: [0.1,\ 0.0,\ -0.15]^T$	$B'_2: [0.0633,\ 0.0774,\ 0.04]^T$
$3'$	$A'_3: [-0.1,\ 0.0,\ -0.15]^T$	$B'_3: [-0.0774,\ 0.0633,\ 0.04]^T$
$4'$	$A'_4: [-0.1,\ 0.0,\ -0.15]^T$	$B'_4: [-0.0633,\ -0.0774,\ 0.04]^T$

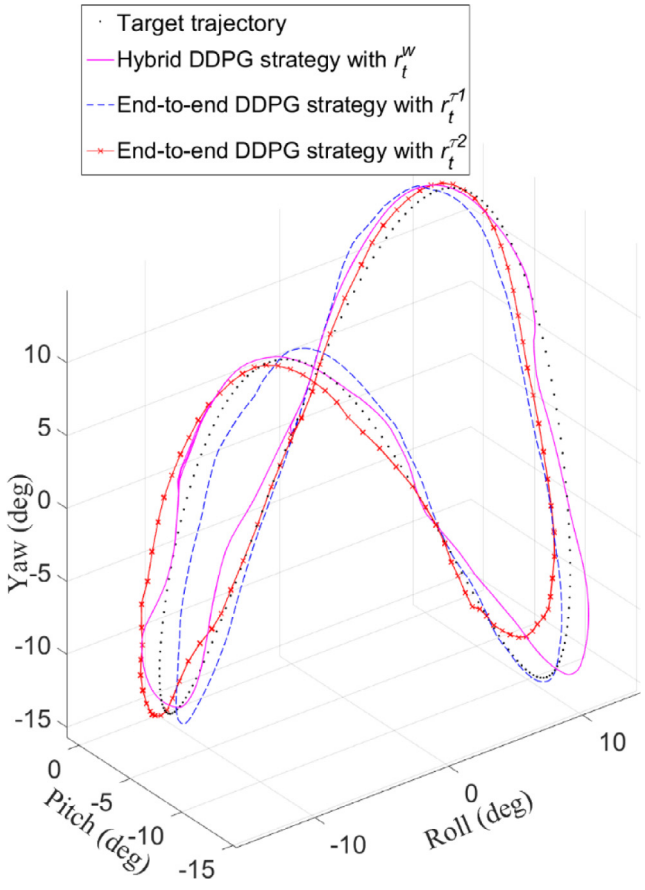

Fig. 15. Trajectories of the CDPR with model uncertainties.

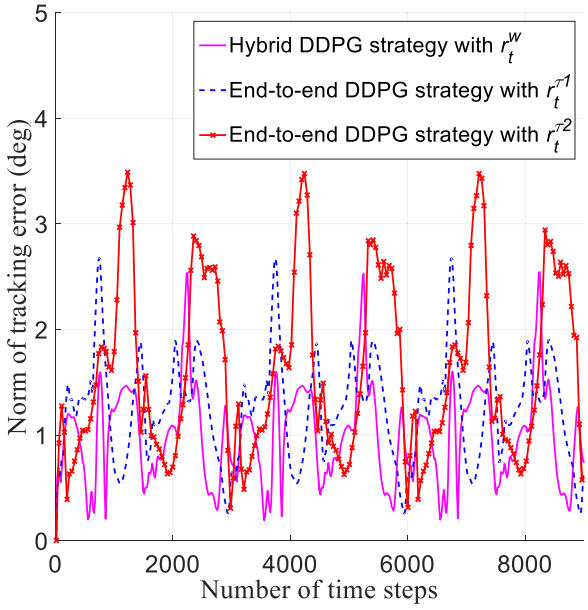

Fig. 16. Tracking errors of the CDPR with model uncertainties.

DDPG strategy with the reward of $r_t^{\tau 2}$ tends to achieve not only smaller tracking error and velocity, but also smaller cable tensions. It makes sense that the end-to-end DDPG strategy with the reward of $r_t^{\tau 2}$ leads to a larger tracking error than the end-to-end DDPG strategy with the reward of $r_t^{\tau 1}$ and the hybrid DDPG strategy with the reward of r_t^w.

6.2. Evaluate control strategies on a CDPR with model uncertainties

In this subsection, the proposed strategies are used to control the CDPR with model uncertainties. In other words, the model of the CDPR to be controlled is slightly different from the model of the CDPR based on which DDPG in the strategies is trained.

In this study, the model uncertainties are supposed to be caused by position errors of anchor points of the CDPR, as shown in Fig. 13. The positions of anchor points of the CDPR without model uncertainty are shown in Table III, while the positions of anchor points of the CDPR with model uncertainties are shown in Table VII. The anchor points on the end-effector of the CDPR with model uncertainties are supposed to rotate 5.73° about the z axis of F_e and slide 0.01 m along the z axis of F_e, compared to those of the CDPR without model uncertainty. Moreover, the moment of inertias and the mass of the CDPR with model uncertainties are assumed the same as those of the CDPR without model uncertainty, as shown in Table II. It should be noticed that the Jacobian of the CDPR used in the inverse dynamics equation in the hybrid DDPG strategy is calculated based on the CDPR without model uncertainty. Thus, the inverse dynamics equation of the CDPR is no longer accurate in controlling the CDPR with model uncertainties due to the position errors of anchor points.

The pose-tracking test and the trajectory-tracking test are conducted on a CDPR with model uncertainties in this subsection. In the pose-tracking test, the target poses are the poses shown in Fig. 9. The outcomes of the pose-tracking test are shown in Table VIII. According to Table VIII, the control performance of the end-to-end DDPG strategy is affected by the model uncertainties. However, the control performance of the hybrid DDPG strategy is not affected much, even though the inverse dynamics equation of the CDPR is not accurate in this case. Therefore, the hybrid DDPG strategy is more robust to model uncertainties than the end-to-end DDPG strategy. Target poses that the CDPR fail to reach are shown in Fig. 14. These target poses are close to the boundary of the training workspace.

In the trajectory-tracking test, the CDPR with model uncertainties is controlled by the proposed strategies to track the target trajectory defined in (23). The trajectories and tracking errors of the CDPR with model uncertainties are shown in Figs. 15 and 16, respectively. According to Fig. 15, the CDPR with model uncertainties can still be controlled by the proposed strategies to track the target trajectory. In other words, the proposed strategies are robust to certain model uncertainties. However, comparing Figs. 12 and 16, one can see that, controlled by the end-to-end DDPG strategy with the reward of $r_t^{\tau 1}$, the CDPR with model uncertainties has larger tracking errors (about 2.7°) than the CDPR without model

Table VIII
Outcomes of the pose-tracking test conducted on a CDPR with model uncertainties.

Strategy	Number of target poses the CDPR reaches successfully	Number of target poses the CDPR fails to reach	Success rate
Hybrid DDPG strategy with the reward of r_t^w	992	8	99.2%
End-to-end DDPG strategy with the reward of $r_t^{\tau 1}$	921	79	92.1%
End-to-end DDPG strategy with the reward of $r_t^{\tau 2}$	981	19	98.1%

uncertainty (about 2.2°). It means a decrease in control performance. Moreover, controlled by the hybrid DDPG strategy and the end-to-end DDPG strategy with the reward of $r_t^{\tau 2}$, the CDPR with model uncertainties has almost the same tracking errors as the CDPR without model uncertainty.

7. Conclusion

In this paper, an end-to-end DDPG strategy and a hybrid DDPG strategy are developed and compared in controlling a CDPR. The hybrid DDPG strategy integrates DDPG and the inverse dynamics equation of a CDPR. Given a target pose of a CDPR, DDPG outputs the target wrench in the task space based on the behavior policy. The inverse dynamics equation takes the target wrench in the task space and solves the optimal tension distribution of cables in the joint space. The end-to-end DDPG strategy accomplishes the entire control task of a CDPR by DDPG. This study shows that the hybrid DDPG strategy learns faster than the end-to-end DDPG strategy in training. Both the hybrid DDPG strategy and the end-to-end DDPG strategy are robust to certain model uncertainties. However, the hybrid DDPG strategy is more robust to model uncertainties than the end-to-end DDPG strategy. Moreover, the end-to-end DDPG strategy can learn the optimal tension distribution of cables as well as the hybrid DDPG strategy calculates it from the inverse dynamics equation of the CDPR. This study demonstrates that, by taking advantages of both learning and non-learning-based approaches, the hybrid DDPG strategy provides an alternative to accomplish a robot manipulation task.

To control a CDPR in practice, the state variables of the proposed strategies can be measured and the action variables can be implemented. State variables (i.e., the pose and velocity of the CDPR) can be measured by instruments such as inertial measurement units [38]. Action variables (i.e., cable tensions of the CDPR) can be delivered by direct-current motors with proper voltage inputs [39]. One should note that controlling a robot with DRL strategies may have safety issues in practice. For example, although controlling a robot with DRL strategies is feasible for non-safety-critical applications such as robotic palletizing, regulations may prevent the implementation of DRL strategies for safety-critical applications such as robotic surgery.

In the future, the authors plan to build a CDPR prototype such that DRL strategies can be applied on the real CDPR. Besides the DDPG algorithm, more DRL algorithms will be used and compared based on both end-to-end and hybrid strategies. The DRL algorithms will be trained based on a model of the real CDPR and then used to control the real CDPR.

Declaration of Competing Interest

None

References

[1] Y. Tsurumine, Y. Cui, E. Uchibe, T. Matsubara, Deep reinforcement learning with smooth policy update: application to robotic cloth manipulation, Robot. Auton. Syst. 112 (2019) 72–83, doi:10.1016/j.robot.2018.11.004.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing atari with deep reinforcement learning, arXiv:1312.5602, (2013).

[3] J. Schulman, S. Levine, P. Abbeel, M. Jordan, P. Moritz, Trust region policy optimization, Int. Conf. Mach. Learn. 37 (2015) 1889–1897.

[4] W.H. Montgomery, S. Levine, Guided policy search via approximate mirror descent, Adv. Neural Inf. Process. Syst. (2016) 4008–4016.

[5] T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, arXiv:1509.02971 (2015).

[6] T. Zhang, G. Kahn, S. Levine, P. Abbeel, Learning deep control policies for autonomous aerial vehicles with MPC-guided policy search, IEEE Int. Conf. Robot. Autom. (2016) 528–535 IEEE, doi:10.1109/ICRA.2016.7487175.

[7] A. Nagabandi, G. Kahn, R.S. Fearing, S. Levine, Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning, IEEE Int. Conf. Robot. Autom. (2018) 7559–7566, doi:10.1109/ICRA.2018.8463189.

[8] M. Zhang, X. Geng, J. Bruce, K. Caluwaerts, M. Vespignani, V. SunSpiral, P. Abbeel, S. Levine, Deep reinforcement learning for tensegrity robot locomotion, IEEE Int. Conf. Robot. Autom. (2017) 634–641 IEEE, doi:10.1109/ICRA.2017.7989079.

[9] M.D.M. Rahman, S.M.H. Rashid, M.M. Hossain, Implementation of Q learning and deep Q network for controlling a self balancing robot model, Robot. Biomim. 5 (2018) 8, doi:10.1186/s40638-018-0091-9.

[10] N. Passalis, A. Tefas, Deep reinforcement learning for controlling frontal person close-up shooting, Neurocomputing (2019), doi:10.1016/j.neucom.2019.01.046.

[11] Z. Li, S. Xue, W. Lin, M. Tong, Training a robust reinforcement learning controller for the uncertain system based on policy gradient method, Neurocomputing 316 (2018) 313–321, doi:10.1016/j.neucom.2018.08.007.

[12] F. Li, Q. Jiang, S. Zhang, M. Wei, R. Song, Robot skill acquisition in assembly process using deep reinforcement learning, Neurocomputing (2019), doi:10.1016/j.neucom.2019.01.087.

[13] E. Todorov, T. Erez, Y. Tassa, Mujoco: a physics engine for model-based control, IEEE/RSJ Int. Conf. Intell. Robot. Syst. (2012) 5026–5033 IEEE, doi:10.1109/IROS.2012.6386109.

[14] N. Koenig, A. Howard, Design and use paradigms for Gazebo, an open-source multi-robot simulator, IEEE/RSJ Int. Conf. Intell. Robot. Syst. 3 (2004) 2149–2154, doi:10.1109/IROS.2004.1389727.

[15] S. Levine, C. Finn, T. Darrell, P. Abbeel, End-to-end training of deep visuomotor policies, J. Mach. Learn. Res. 17 (2016) 1334–1373.

[16] J. Kober, J.A. Bagnell, J. Peters, Reinforcement learning in robotics: a survey, Int. J. Robot. Res. 32 (2013) 1238–1274, doi:10.1177/0278364913495721.

[17] V. Mnih, A.P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, K. Kavukcuoglu, Asynchronous methods for deep reinforcement learning, Int. Conf. Mach. Learn. 48 (2016) 1928–1937.

[18] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, M. Riedmiller, Deterministic policy gradient algorithms, Int. Conf. Mach. Learn. 32 (2014) 387–395.

[19] X. Diao, O. Ma, Vibration analysis of cable-driven parallel manipulators, Multibody Syst. Dyn. 21 (2009) 347–360, doi:10.1007/s11044-008-9144-0.

[20] M.A. Khosravi, H.D. Taghirad, Robust PID control of fully-constrained cable driven parallel robots, Mechatronics 24 (2014) 87–97, doi:10.1016/j.mechatronics.2013.12.001.

[21] X. Weber, L. Cuvillon, J. Gangloff, Active vibration canceling of a cable-driven parallel robot using reaction wheels, IEEE/RSJ Int. Conf. Intell. Robot. Syst. (2014) 1724–1729, doi:10.1109/IROS.2014.6942787.

[22] W. Bin Lim, G. Yang, S.H. Yeo, S.K. Mustafa, A generic force-closure analysis algorithm for cable-driven parallel manipulators, Mech. Mach. Theory 46 (2011) 1265–1275, doi:10.1016/j.mechmachtheory.2011.04.006.

[23] X. Diao, O. Ma, Force-closure analysis of 6-DOF cable manipulators with seven or more cables, Robotica 27 (2008) 209, doi:10.1017/s0263574708004591.

[24] S. Behzadipour, A. Khajepour, Stiffness of cable-based parallel manipulators with application to stability analysis, J. Mech. Des. 128 (2005) 303–310, doi:10.1115/1.2114890.

[25] H. Jamshidifar, A. Khajepour, B. Fidan, M. Rushton, Kinematically-constrained redundant cable-driven parallel robots: modeling, redundancy analysis, and stiffness optimization, IEEE/ASME Trans. Mechatron. 22 (2017) 921–930, doi:10.1109/TMECH.2016.2639053.

[26] M. Azadi, S. Behzadipour, G. Faulkner, Antagonistic variable stiffness elements, Mech. Mach. Theory. 44 (2009) 1746–1758, doi:10.1016/j.mechmachtheory.2009.03.002.

[27] P. Bosscher, A.T. Riechel, I. Ebert-Uphoff, Wrench-feasible workspace generation for cable-driven robots, IEEE Trans. Robot. 22 (2006) 890–902, doi:10.1109/TRO.2006.878967.

[28] M. Gouttefarde, D. Daney, J. Merlet, Interval-analysis-based determination of the wrench-feasible workspace of parallel cable-driven robots, IEEE Trans. Robot. 27 (2011) 1–13, doi:10.1109/TRO.2010.2090064.

[29] S. Fang, D. Franitza, M. Torlo, F. Bekes, M. Hiller, Motion control of a tendon-based parallel manipulator using optimal tension distribution, IEEE/ASME Trans. Mechatron. 9 (2004) 561–568, doi:10.1109/TMECH.2004.835336.

[30] S.-R. Oh, S.K. Agrawal, Cable suspended planar robots with redundant cables: controllers with positive tensions, IEEE Trans. Robot. 21 (2005) 457–465, doi:10.1109/TRO.2004.838029.
[31] M.J.-D. Otis, S. Perreault, T. Nguyen-Dang, P. Lambert, M. Gouttefarde, D. Laurendeau, C. Gosselin, Determination and management of cable interferences between two 6-DOF foot platforms in a cable-driven locomotion interface, IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. 39 (2009) 528–544, doi:10.1109/TSMCA.2009.2013188.
[32] R. Babaghasabha, M.A. Khosravi, H.D. Taghirad, Adaptive robust control of fully-constrained cable driven parallel robots, Mechatronics 25 (2015) 27–36, doi:10.1016/j.mechatronics.2014.11.005.
[33] J.J. Craig, Introduction to Robotics: Mechanics and Control (2009).
[34] J. Randløv, P. Alstrøm, Learning to drive a bicycle using reinforcement learning and shaping, Int. Conf. Mach. Learn. (1998) 463–471.
[35] K. Van Moffaert, M.M. Drugan, A. Nowé, Scalarized multi-objective reinforcement learning: novel design techniques, in: 2013 IEEE Symp. Adapt. Dyn. Program. Reinf. Learn., 2013: pp. 191–199. doi:10.1109/ADPRL.2013.6615007.
[36] B. Fernandez-Gauna, U. Fernandez-Gamiz, M. Grana, Variable speed wind turbine controller adaptation by reinforcement learning, Integr. Comput. Aided Eng. 24 (2017) 27–39, doi:10.3233/ICA-160531.
[37] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015.
[38] A. Aflakian, A. Safaryazdi, M. Tale Masouleh, A. Kalhor, Experimental study on the kinematic control of a cable suspended parallel robot for object tracking purpose, Mechatronics 50 (2018) 160–176, doi:10.1016/j.mechatronics.2018.02.005.
[39] A.S. Niyetkaliyev, S. Hussain, M.H. Ghayesh, G. Alici, Review on design and control aspects of robotic shoulder rehabilitation orthoses, IEEE Trans. Hum. Mach. Syst. 47 (2017) 1134–1145, doi:10.1109/THMS.2017.2700634.

Hao Xiong received his B.S. degree in Automation from Beihang University, China, in 2011 and M.S. degree in Electromagnetics Design from University of Nottingham, United Kingdom, in 2013. He is currently pursuing his Ph.D. degree in Technology in the School of Engineering Technology at Purdue University, United States. His research interests are cable-driven parallel robots, assistive devices, reinforcement learning, and intelligent control.

Tianqi Ma received his B.S. degree in Engineering Mechanics from Tsinghua University, China, in 2019. He is currently pursuing his Ph.D. degree in Automation in the Department of Automation, Tsinghua University, China. His research interests are human-robot interaction and artificial intelligence control.

Lin Zhang received his B.S. degree in Control Science and Automation from Harbin Institute of Technology, China, in 2007, M.S. degree in Mechatronics from Beijing University of Post and Telecommunication, China, in 2010, and Ph.D. degree in Mechanical Engineering from New Mexico State University, United States, in 2016. He is currently a Senior Research Associate at University of Cincinnati, United States. His research interests are robotics, human-robot interaction and deep reinforcement learning.

Xiumin Diao received his B.S. degree in Mechanical Design and Manufacturing from Yantai University, China, in 2000, M.S. degree in Measurement Technology and Automatic Device from Beihang University, China, in 2003, and Ph.D. degree in Mechanical Engineering from New Mexico State University, United States, in 2007. He is currently an Assistant Professor in the School of Engineering Technology at Purdue University, United States. His research interests are robotics and human-robot interaction.