# Dynamic Obstacle Avoidance for Cable-Driven Parallel Robots With Mobile Bases via Sim-to-Real Reinforcement Learning

Yuming Liu, Zhihao Cao, Hao Xiong, Junfeng Du, Huanhui Cao, and Lin Zhang

***Abstract*—A Cable-Driven Parallel Robot (CDPR) with Mobile Bases (MBs) can modify its geometric architecture and is suitable for manipulation tasks in constrained environments. In manipulation tasks, a CDPR with MBs inevitably encounters obstacles, including dynamic obstacles. However, the high dimensional state space and a considerable number of constraints caused by multiple cables and MBs make the real-time dynamic obstacle avoidance of a CDPR with MBs challenging. This letter proposes a Reinforcement Learning (RL)-based dynamic obstacle avoidance method for a CDPR with MBs to deal with dynamic obstacles in real time. To explain the RL-based dynamic obstacle avoidance method, this letter focuses on a CDPR with four fixed-length cables connected to four MBs. An RL-based Obstacle Avoidance Controller (OAC) is developed and integrated into a trajectory tracking controller to address the dynamic obstacle avoidance problem of a CDPR with MBs tracking a target trajectory. To explain and evaluate the RL-based dynamic obstacle avoidance method further, an RL-based OAC is trained in a Mujoco simulator and transferred to a CDPR with four fixed-length cables connected to four MBs in the real world.**

***Index Terms*—Collision avoidance, machine learning for robot control, parallel robots, wire mechanism.**

## I. Introduction

THE Cable-Driven Parallel Robot (CDPR) has two platforms - the base and the Moving Platform (MP). Anchor points on both platforms are paired and connected via cables [1]. A CDPR usually has large workspace, low inertia, and high payload-to-weight ratio [2]. Research on CDPRs focuses mainly on three major classes [3] - (i) CDPRs with variable-length cables and winches fixed in the inertial frame [4], [5], [6], (ii) CDPRs with fixed-length cables connected to Mobile Bases (MBs) [7], [8], and (iii) CDPRs with variable-length cables and winches attached to MBs [9], [10], [11]. With MBs, a CDPR can move throughout a warehouse to address tasks that a CDPR with fixed winches is hard to address (e.g., transportation of goods in a cluttered warehouse) [12]. However, to move throughout a warehouse safely in practice, a CDPR with MBs is preferred to be able to provide safe actions (i.e., control inputs) that lead to safe trajectories without collisions with static obstacles as well as dynamic obstacles [13], as shown in Fig. 1.

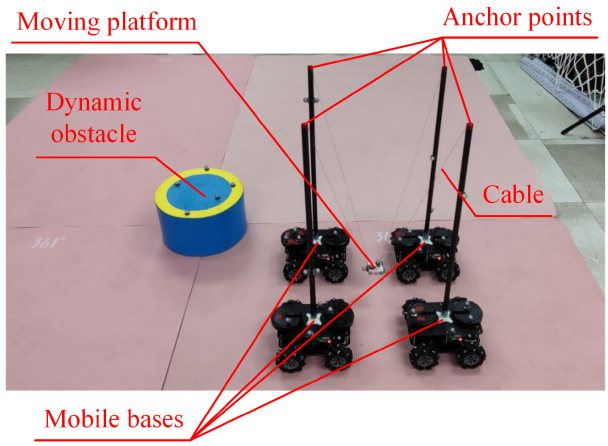


Fig. 1. A CDPR with four mobile bases.

To achieve safe trajectories for CDPRs, researchers have proposed several methods in recent years. For CDPRs with variable-length cables and winches fixed in the inertial frame, the combination of the Artificial Potential Field (APF) method and Rapidly-exploring Random Tree (RRT) was used to achieve trajectory planning methods for CDPRs in [14]. Gilbert–Johnson–Keerthi algorithm was applied to improve the RRT-based collision detection of CDPRs in [15]. Dynamic trajectory planning methods based on RRT [16] have been developed for three degrees-of-freedom (DOFs) suspended CDPRs by Xiang et al. Passarini et al. proposed a dynamic trajectory planning method for CDPRs that suffers a broken cable to drive the MP to a safe pose [17]. To reduce the computational time of trajectory planning, the dynamic trajectory planning problem with multiple variables and nonlinear constraints was reduced to a lower-dimension optimization problem based on approximate strategies. For CDPRs with variable-length cables and winches attached to MBs, a direct transcription optimization method was proposed for the trajectory planning of a CDPR with MBs by

Manuscript received 3 September 2022; accepted 17 January 2023. Date of publication 2 February 2023; date of current version 9 February 2023. This letter was recommended for publication by Associate Editor S. Caro and Editor J. P. Desai upon evaluation of the reviewers' comments. This work was supported in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2021A1515110021, in part by the Shenzhen Science and Technology Program under Grant RCBS20210609103819024, and in part by the Research Foundation for Advanced Talents, the Harbin Institute of Technology Shenzhen under Grant CA11409019. *(Yuming Liu and Zhihao Cao are co-first authors.) (Corresponding author: Hao Xiong.)*

Yuming Liu, Zhihao Cao, Hao Xiong, Junfeng Du, and Huanhui Cao are with the Guangdong Key Laboratory of Intelligent Morphing Mechanisms and Adaptive Robotics and the School of Mechanical Engineering and Automation, the Harbin Institute of Technology Shenzhen, Shenzhen 518055, China (e-mail: llyymm2828@gmail.com; m.zhihaocao@gmail.com; xionghao@hit.edu.cn; janphoondu@outlook.com; 20S153121@stu.hit.edu.cn).

Lin Zhang is with the Department of Physics & Astronomy, University of Central Arkansas, Conway, AR 72035 USA (e-mail: linzhank@gmail.com).

Digital Object Identifier 10.1109/LRA.2023.3241801

Rasheed et al. [18]. The method uses an optimization-based approach to integrate the constraints associated with CDPRs into a trajectory planning problem. Direct transcription is applied to attenuate the reliance on the initial guess. However, for a CDPR with MBs navigating in an obstacle-rich environment, the above-mentioned methods that usually employ re-planning to search for an alternative trajectory [19] are still too time-consuming to provide safe trajectories or safe actions to avoid obstacles in real time with the increase of the number of cables, according to [1].

Reinforcement Learning (RL) is a Machine Learning (ML) technique that can deal with complicated problems and determine actions in real time. ML techniques, such as supervised learning, have been applied to address the kinematics [20], [21], [22] and reconfiguration planning [1] of CDPRs. RL has achieved dramatic successes in several areas, including the manipulation of CDPRs [23], [24], [25]. Researchers have also applied RL to address the obstacle avoidance of robots. Wang et al. proposed a method based on RL to exploit environmental spatio-temporal information and achieved trajectory planning for mobile robots in a planar dynamic environment [19]. With this method, mobile robots can efficiently reach target positions and avoid potential collisions with other mobile robots or dynamic obstacles simultaneously. A decentralized sensor-level collision-avoidance method for a system of multiple mobile robots was developed in [26]. The method directly maps raw sensor measurements to steering commands in terms of target velocity for mobile robots. It is shown that methods based on RL, especially deep RL, have achieved success in the obstacle avoidance of robots in recent years. However, to the best knowledge of the authors, a method for the dynamic obstacle avoidance of a CDPR with MBs based on deep RL is not available yet.

This letter focuses on the dynamic obstacle avoidance of a CDPR with fixed-length cables connected to MBs and develops an RL-based Obstacle Avoidance Controller (OAC) for a CDPR with MBs following a given target trajectory. The main contributions of this letter are as follows.

- This letter proposes an RL-based OAC for a CDPR with fixed-length cables connected to MBs to address dynamic obstacles in real time. Then, the RL-based OAC is integrated into a Trajectory Tracking Controller (TTC) for the CDPR with MBs.
- This letter develops an OAC based on Soft Actor Critic (SAC) algorithm and an attention module to provide a realization method for the RL-based OAC. The design, training, and sim-to-real transfer of the SAC-based OAC are presented based on a CDPR with a three-DOF point-mass MP and four fixed-length cables connected to four two-DOF MBs.

The rest of this letter is organized as follows. Section II introduces the preliminaries of this study, including problem statement, notations of a CDPR with MBs, and Reinforcement Learning. In Section III, an RL-based OAC is proposed and integrated into a TTC for a CDPR with MBs. Section IV conducts experiments to explain and evaluate the developed RL-based OAC. Finally, this letter is summarized in Section V.

## II. Preliminaries

This section introduces the preliminaries of the present letter, including problem statement, notations of a CDPR with MBs, and Reinforcement Learning.

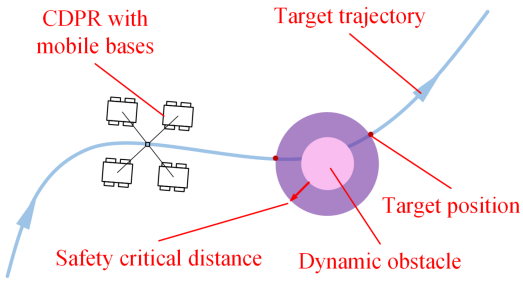


Fig. 2. A CDPR with four mobile bases encounters a dynamic obstacle in trajectory tracking.

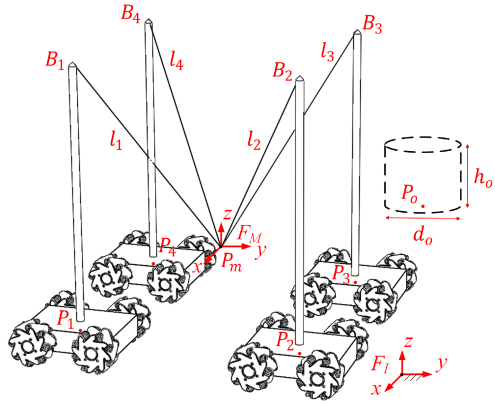


Fig. 3. Notations of a CDPR with four mobile bases encountering an obstacle.

### A. Problem Statement

Researchers have proposed trajectory planning methods for a CDPR with MBs in a cluttered environment with static obstacles (e.g., [18]). In view of these methods, this letter tends to address the dynamic obstacle avoidance for a CDPR with MBs when tracking a target trajectory determined by a trajectory planning method. It is assumed that when tracking a target trajectory, the CDPR with MBs may encounter dynamic obstacles that the trajectory planning method doesn't take into account, as shown in Fig. 2. A dynamic obstacle may move continuously or move to and stop at a certain position to obstruct the movement of the CDPR with MBs. The CDPR with MBs is supposed to bypass or go over a dynamic obstacle in real time and avoid collisions between the MP and the obstacle, collisions between an MB and the obstacle, collisions between two MBs, collisions between the MP and an MB, and collisions between a cable and the obstacle. The CDPR with MBs needs to move back to the target trajectory then. To solve the dynamic obstacle avoidance problem described above, a dynamic obstacle avoidance method is needed to provide actions, in real time, to make a CDPR with MBs avoid a dynamic obstacle and move back to the target trajectory eventually.

### B. Notations of a Cable-Driven Parallel Robot With Mobile Bases

This letter focuses on a CDPR with fixed-length cables connected to MBs. Notations of a CDPR with three translational DOFs and four cables connected to four MBs encountering a dynamic obstacle are shown in Fig. 3. $l_i$ $(i = 1, 2, 3, 4)$ is the length of the $i$th cable. In this study, $l_i$ is a constant. $B_i$

$(i = 1, 2, 3, 4)$ is the anchor point of the $i$th cable on an MB. $P_m$ is the anchor point of cables on the MP. The positions of the anchor points $P_m$ and $B_i$ are represented by the vectors $\boldsymbol{p}_m$ and $\boldsymbol{b}_i$ $(i = 1, 2, 3, 4)$, respectively. $\boldsymbol{b}_i$ can be adjusted by the movement of the $i$th MB. $P_i$ $(i = 1, 2, 3, 4)$ is attached to the bottom of the $i$th MB. The position of the $i$th MB is represented by the position of $P_i$, denoted as $\boldsymbol{p}_i$. The inertial frame is represented by $F_I$. The moving platform frame $F_M$ is attached to the MP and is with a constant orientation in the inertial frame. It is assumed that a dynamic obstacle is contained within a cylinder. $P_o$ is attached to the centroid of the bottom surface of the cylinder. The position of the cylinder is represented by the position of $P_o$, denoted as $\boldsymbol{p}_o = [p_{o,x}, p_{o,y}, p_{o,z}]^T$. The diameter of the bottom surface and the height of the cylinder are $d_o$ and $h_o$, respectively.

### C. Reinforcement Learning

RL is a technique for an agent to achieve a learned policy when interacting with the environment by maximizing the expected cumulative reward [27]. A problem addressed by RL can be defined as a Markov Decision Process (MDP), $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$. $\mathcal{S}$ is a set of states. $\mathcal{A}$ denotes a set of actions. $\mathcal{P}$ is a state transition probability. $r$ represents a reward function and $\gamma \in (0, 1)$ is a discount factor. Based on the current state $s \in \mathcal{S}$ with respect to its policy $\pi : \mathcal{S} \mapsto \mathcal{A}$, the agent selects an action $a \in \mathcal{A}$ at time step $t$. The state transfers to a new state $s'$ and the agent receives a reward $r$. The agent is supposed to maximize the accumulated reward in $T$ time steps $\sum_{t=0}^{T} \gamma^{T-t} r(s_t, a_t)$.

## III. Reinforcement Learning-Based Dynamic Obstacle Avoidance

To address the dynamic obstacle avoidance for a CDPR with MBs, this section develops an RL-based OAC and integrates the OAC into a TTC. The flow diagram of the integration of the OAC and the TTC is presented in Fig. 4. The TTC determines the target velocities of MBs based on the target position and target velocity of the MP. The TTC can be designed according to an arbitrary trajectory tracking method for CDPRs with MBs. Since different trajectory tracking methods for CDPRs with MBs have been proposed and introduced in detail in several previous studies (e.g., [11], [18], [28]), this section doesn't provide the details of a TTC. The RL-based OAC provides the target velocities for MBs to avoid dynamic obstacles in real time, based on the position and the shape of a dynamic obstacle, the position of MBs, the position of the MP, and the target trajectory. The details of the RL-based OAC are presented in Section III-A. The target velocities of MBs determined by the TTC and the RL-based OAC are fused in a velocity fusion module and are set for MBs then. The details of the velocity fusion module are presented in Section III-B. To explain the RL-based OAC further, an OAC based on the SAC algorithm is proposed in Section III-C.

### A. Reinforcement Learning-Based Obstacle Avoidance Controller

An RL-based OAC is designed to control a CDPR with MBs to avoid a dynamic obstacle and move back to a target trajectory. The RL-based OAC is explained based on a CDPR with four MBs in this section. It is assumed that the CDPR with MBs can avoid the dynamic obstacle based on proper maneuvers. The

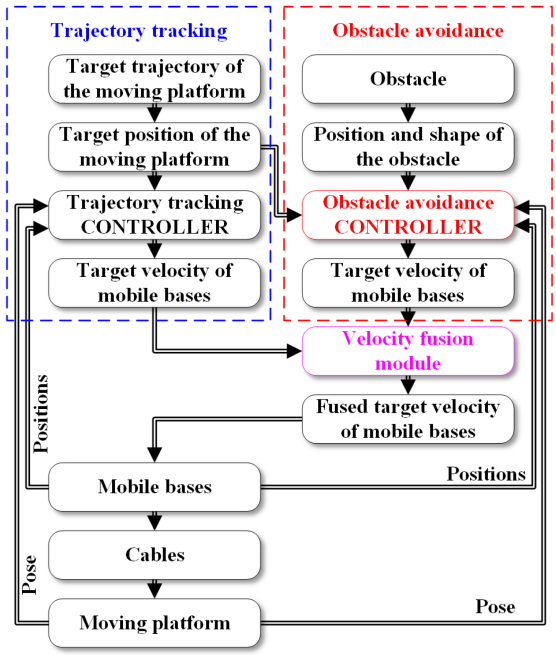


Fig. 4. Flow diagram of the integration of obstacle avoidance and trajectory tracking for a CDPR with mobile bases.

position and shape of the obstacle can be observed by sensors such as cameras, lidars, or a motion capture system.

An OAC assumes that an obstacle is contained within a cylinder. If a target trajectory passes through the cylinder and an original target position is within the cylinder, the OAC will reset the target position to a position that is on the target trajectory but outside the cylinder. The OAC takes the position of MBs denoted as $\boldsymbol{p}_i$ $(i = 1, 2, 3, 4)$, the position of the MP denoted as $\boldsymbol{p}_m$, the target position of the MP denoted as $\boldsymbol{p}_m^*$, the previous outputs of the OAC denoted as $\dot{\boldsymbol{p}}_{t-1}^{RL}$, and the position $\boldsymbol{p}_o$, height $h_o$, and the diameter of the bottom surface $d_o$ of the cylinder into account. The state utilized by the OAC is defined in the moving platform frame $F_M$ as $\boldsymbol{s}_t = [(\boldsymbol{p}_m^*)^T, \boldsymbol{p}_1^T, \boldsymbol{p}_2^T, \boldsymbol{p}_3^T, \boldsymbol{p}_4^T, \boldsymbol{p}_o^T, d_o, h_o, (\dot{\boldsymbol{p}}_{t-1}^{RL})^T]^T$. The outputs of the OAC are the target velocities of MBs, denoted as $(\dot{\boldsymbol{p}}^{RL})^T = [(\dot{\boldsymbol{p}}_1^{RL})^T, (\dot{\boldsymbol{p}}_2^{RL})^T, (\dot{\boldsymbol{p}}_3^{RL})^T, (\dot{\boldsymbol{p}}_4^{RL})^T]^T$.

Inspired by [1], [9], the dynamic obstacle avoidance problem of a CDPR with MBs is defined as a multi-objective optimization problem for the OAC

$$\begin{aligned} \min \sum_{t=0}^{T} \gamma^{T-t} \{ & \lambda_{target} f_{target}(\boldsymbol{s}_t) + \lambda_{bb} f_{bb}(\boldsymbol{s}_t) \\ & + \lambda_{bo} f_{bo}(\boldsymbol{s}_t) + \lambda_{\min} f_{\min}(\boldsymbol{s}_t) \\ & + \lambda_{\max} f_{\max}(\boldsymbol{s}_t) + \lambda_{po} f_{po}(\boldsymbol{s}_t) \\ & + \lambda_{co} f_{co}(\boldsymbol{s}_t) \}, \end{aligned} \tag{1}$$

where $T$ represents the number of time steps of the problem and $\gamma$ is a discount factor. The objectives of the optimization problem include approaching a target position behind an obstacle, obstacle avoidance, and maintaining a feasible configuration. It should be noticed that to make the dynamic obstacle avoidance method more feasible for one who is not a specialist in CDPRs, straightforward cost functions (e.g., $f_{bb}$, $f_{\min}$, and $f_{\max}$), rather

than cost functions based on the workspace of a CDPR, are designed to implicitly maintain the CDPR in a feasible configuration.

Specifically, $f_{target}$ is a target tracking cost function, aiming to lead the MP to a target position. $\lambda_{target}$ is the weight of the function. The target tracking cost function is defined as

$$f_{target}\left(\boldsymbol{s}_t\right) = \|\boldsymbol{p}_m^*\|_2 \tag{2}$$

$f_{bb}$ is a base-to-base interference cost function and $\lambda_{bb}$ is the weight of the function. The base-to-base interference cost function is defined as

$$f_{bb}\left(\boldsymbol{s}_t\right) = \sum_{i=1}^{4}\sum_{\substack{j=1\\ j\neq i}}^{4} f_{bb}^{ij}\left(\boldsymbol{s}_t\right), \tag{3}$$

$f_{bb}^{ij}(\boldsymbol{s}_t)$ is the base-to-base interference cost function of the $i$th MB and the $j$th MB. $f_{bb}^{ij}(\boldsymbol{s}_t)$ is defined as

$$f_{bb}^{ij}\left(\boldsymbol{s}_t\right) = \begin{cases} d_{bb} - \|\boldsymbol{p}_i - \boldsymbol{p}_j\|_2 & \text{if } \|\boldsymbol{p}_i - \boldsymbol{p}_j\|_2 < d_{bb} \\ 0 & \text{otherwise} \end{cases}, \tag{4}$$

where $d_{bb}$ is a critical distance between two MBs.

$f_{bo}$ is a base-to-obstacle interference cost function and $\lambda_{bo}$ is the weight of the function. The base-to-obstacle interference cost function is defined as

$$f_{bo}\left(\boldsymbol{s}_t\right) = \sum_{i=1}^{4} f_{bo}^{i}\left(\boldsymbol{s}_t\right) \tag{5}$$

where $f_{bo}^{i}(\boldsymbol{s}_t)$ is the base-to-obstacle interference cost function of the $i$th MB. $f_{bo}^{i}(\boldsymbol{s}_t)$ is defined as

$$f_{bo}^{i}\left(\boldsymbol{s}_t\right) = \begin{cases} d_{bo} - \|\boldsymbol{p}_o - \boldsymbol{p}_i\|_2 & \text{if } \|\boldsymbol{p}_o - \boldsymbol{p}_i\|_2 < d_{bo} \\ 0 & \text{otherwise} \end{cases}, \tag{6}$$

where $d_{bo}$ is the critical distance between an MB and the obstacle.

$f_{\min}$ is a minimum base-to-platform distance cost function and $\lambda_{\min}$ is the weight of the function. The minimum base-to-platform distance cost function is defined as

$$f_{\min}\left(\boldsymbol{s}_t\right) = \sum_{i=1}^{4} f_{\min}^{i}\left(\boldsymbol{s}_t\right) \tag{7}$$

where $f_{\min}^{i}(\boldsymbol{s}_t)$ is the minimum base-to-platform distance cost function of the $i$th MB. $f_{\min}^{i}(\boldsymbol{s}_t)$ is defined as

$$f_{\min}^{i}\left(\boldsymbol{s}_t\right) = \begin{cases} d_{\min} - \|\boldsymbol{p}_i\|_2 & \text{if } \|\boldsymbol{p}_i\|_2 < d_{\min} \\ 0 & \text{otherwise} \end{cases}, \tag{8}$$

where $d_{\min}$ is a critical minimum distance between an MB and the MP.

$f_{\max}$ is a maximum base-to-platform distance cost function and $\lambda_{\max}$ is the weight of the function. The maximum base-to-platform distance cost function is defined as

$$f_{\max}\left(\boldsymbol{s}_t\right) = \sum_{i=1}^{4} f_{\max}^{i}\left(\boldsymbol{s}_t\right) \tag{9}$$

where $f_{\max}^{i}(\boldsymbol{s}_t)$ is the maximum base-to-platform distance cost function of the $i$th MB. $f_{\max}^{i}(\boldsymbol{s}_t)$ is defined as

$$f_{\max}^{i}\left(\boldsymbol{s}_t\right) = \begin{cases} \|\boldsymbol{p}_i\|_2 - d_{\max} & \text{if } \|\boldsymbol{p}_i\|_2 > d_{\max} \\ 0 & \text{otherwise} \end{cases}, \tag{10}$$

where $d_{\max}$ is the critical maximum distance between an MB and the MP.

$f_{po}$ is a platform-to-obstacle interference cost function and $\lambda_{po}$ is the weight of the function. The platform-to-obstacle interference cost function is defined as

$$f_{po}\left(\boldsymbol{s}_t\right) = \begin{cases} p_{o,z} + h + d_{po} & \text{if } \begin{array}{l} p_{o,z} + h + d_{po} > 0 \ and \\ \|\left[p_{o,x}, p_{o,y}\right]^T\|_2 < \frac{d_o}{2} \end{array} \\ 0 & \text{otherwise} \end{cases}, \tag{11}$$

where $d_{po}$ is a critical distance between the MP and the obstacle. $[p_{o,x}, p_{o,y}]^T$ represents the position of the obstacle in the horizontal plane in $F_M$.

$f_{co}$ is a cable-to-obstacle interference cost function and $\lambda_{co}$ is the weight of the function. The cable-to-obstacle interference cost function is defined as

$$f_{co}\left(\boldsymbol{s}_t\right) = \begin{cases} 1 & \text{if } \begin{array}{l} \text{the distance between an arbitrary} \\ \text{cable and the cylinder is zero} \end{array} \\ 0 & \text{otherwise} \end{cases}, \tag{12}$$

### B. Velocity Fusion Module

According to Fig. 4, the TTC provides a target velocity, denoted as $\dot{\boldsymbol{p}}^{TT}$ and the RL-based OAC provides another target velocity, denoted as $\dot{\boldsymbol{p}}^{RL}$. To fuse these two target velocities, a velocity fusion module is developed. The velocity fusion module switches between these two target velocities smoothly and achieves a fused target velocity according to

$$\dot{\boldsymbol{p}}^{*} = \alpha\dot{\boldsymbol{p}}^{RL} + \left(1-\alpha\right)\dot{\boldsymbol{p}}^{TT}, \tag{13}$$

where $\alpha \in [0,1]$ is designed to make $\dot{\boldsymbol{p}}^{TT}$ dominant if the MP is far from an obstacle or $\dot{\boldsymbol{p}}^{RL}$ dominant if the distance between the MP and an obstacle is smaller than a critical distance $d_\alpha$. In this study, $\alpha$ can be expressed as

$$\alpha = \sigma\left(\frac{\|\boldsymbol{p}_o\|_2 - 0.5d_o - d_\alpha}{k_\alpha}\right) \tag{14}$$

where $\sigma(*) = \frac{1}{1+e^{-(*)}}$ is sigmoid function. $k_\alpha$ is a factor that adjusts the smoothness of the switching between $\dot{\boldsymbol{p}}^{TT}$ and $\dot{\boldsymbol{p}}^{RL}$.

### C. Obstacle Avoidance Controller Based on Soft Actor Critic

According to the OAC presented in Section III-A, SAC algorithm [29] can be used to achieve an OAC. The architecture of an OAC based on the SAC algorithm is shown in Fig. 5. According to the SAC algorithm, the OAC includes a policy network, a value network, and corresponding target networks. The target networks are not shown in Fig. 5. The policy network inputs the state observed by a CDPR with MBs. An attention module and a feature extractor are designed and added to a conventional policy network. The details of the attention module and feature extractor are presented in III-C1 and III-C2. According to the SAC algorithm, an SAC-Head based on Fully-Connected (FC)

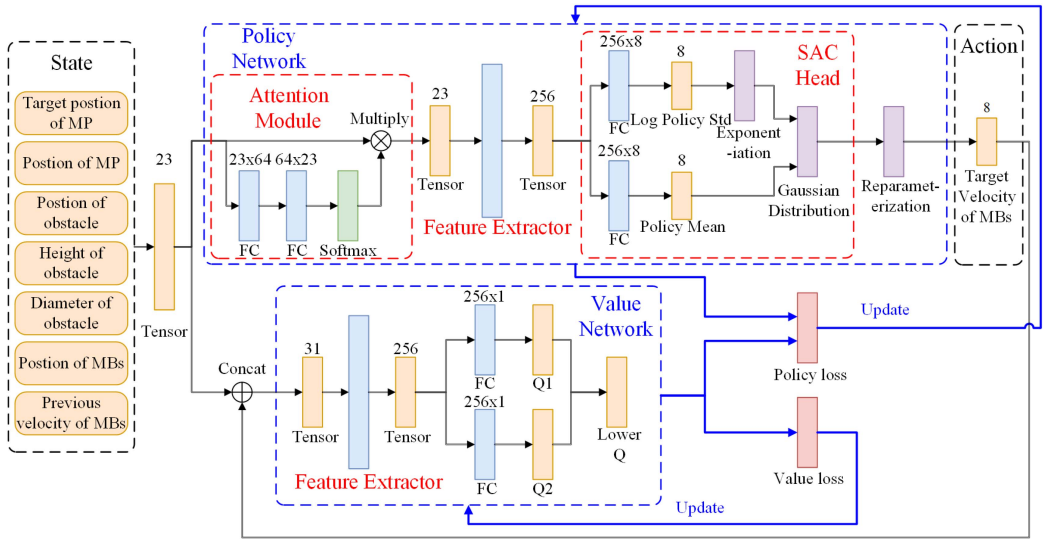


Fig. 5. Architecture of an obstacle avoidance controller based on Soft Actor Critic algorithm.

layers is used to determine the joint distribution of the target velocity of MBs. The target velocity of MBs determined by the OAC (i.e., $\dot{\boldsymbol{p}}^{RL}$) is calculated based on the output of the SAC-Head and reparameterization. The policy network is used to calculate a policy loss and is updated based on the policy loss. The value network inputs a combination of the state observed by the CDPR with MBs and the action (i.e., the target velocity of MBs) determined by the policy network. The feature extractor presented in III-C1 is also applied to the value network. The value network achieves two Q values and outputs the smaller one. The value network is used to calculate a policy loss and a value loss and is updated based on the value loss.

*1) Attention Module:* An attention module is designed inspired by [30], aiming to enable the policy network to focus on information that contributes to the determination of the target velocity of a CDPR with MBs. The attention module has two FC layers and a softmax operator. The attention module adjusts the weight of the elements of the input state to enable the policy network to focus on certain informative elements.

*2) Feature Extractor:* A feature extractor is designed based on a Multi-Layer Perceptron (MLP) that includes three FC layers with ReLu activation functions. The three FC layers have 128 units, 256 units, and 256 units, respectively. The feature extractor maps the elements of the state into a high-dimensional space, aiming to decouple the features represented by the elements of the state.

## IV. Simulation and Experiments

To explain and evaluate the proposed OAC for a CDPR with MBs, the OAC is applied to a CDPR with four fixed-length cables connected to four MBs, as shown in Fig. 1. A model of the CDPR with MBs is established in a Mujoco simulator [31], as shown in Fig. 6. A TTC is designed and an RL-based OAC is achieved in simulation. Then, the effectiveness of the OAC is evaluated based on dynamic obstacle avoidance tests in the real world. The design, training, and evaluation of the RL-based OAC are explained in a video.[1]

[1]Video: https://youtu.be/GRsL08lmYvA

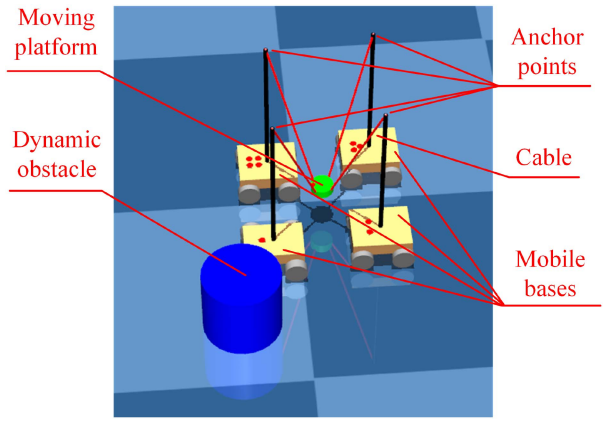


Fig. 6. A CDPR with four fixed-length cables connected to four omnidirectional mobile robots in a Mujoco simulator.

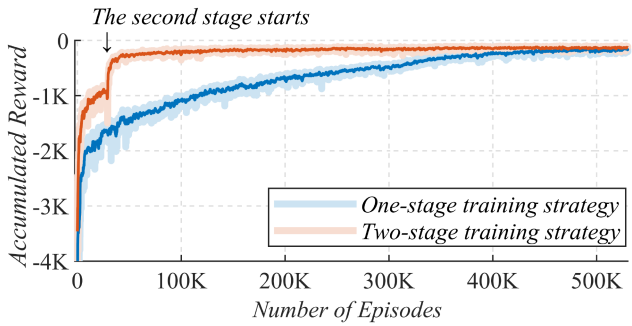


Fig. 7. Accumulated rewards achieved by training strategies.

### A. Setups

The MBs are omnidirectional mobile robots with a size of 0.30 m × 0.20 m. The maximum feasible velocity of MBs is 0.20 m/s. The length of the four cables is 0.70 m. A cable connects to an anchor point that is 0.90 m directly above an MB. A personal computer with an Intel i5-9400F Central Processing

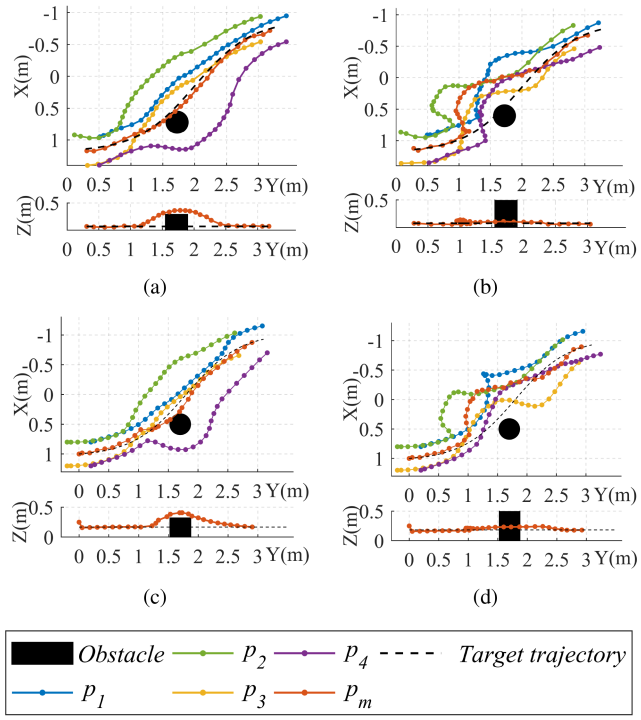


Fig. 8. Trajectories of the moving platform and mobile bases of a CDPR in the inertial frame $F_I$ when encountering (a) a low obstacle with $h_o = 0.33$ m and $d_o = 0.32$ m in the real world, and (b) a high obstacle with $h_o = 0.92$ m and $d_o = 0.32$ m in the real world, (c) a low obstacle with $h_o = 0.33$ m and $d_o = 0.32$ m in simulation, and (d) a high obstacle with $h_o = 0.92$ m and $d_o = 0.32$ m in simulation.

TABLE I
PARAMETERS OF REWARD FUNCTIONS

| Parameters | Value | Parameters | Value | Parameters | Value |
|---|---|---|---|---|---|
| $\lambda_{target}$ | 10 | $\lambda_{bb}$ | 200 | $d_{bb}$ | 0.38 |
| $\lambda_{bo}$ | 500 | $d_{bo}$ | 0.20 | $\lambda_{min}$ | 20 |
| $d_{min}$ | 0.20 | $\lambda_{max}$ | 20 | $d_{max}$ | 0.50 |
| $\lambda_{po}$ | 150 | $d_{po}$ | 0.20 | $\gamma$ | 0.99 |
| $\lambda_{co}$ | 1000 | | | | |

Unit (CPU) and 16 gigabits memory is used in simulation, and in experiments. The Graphics Processing Unit (GPU) of the personal computer is not used in simulation or in experiments. In experiments, the position of the MP, the position of MBs, and the position and shape of a dynamic obstacle are observed based on a NOKOV motion capture system.

### B. Trajectory Tracking Controller

A TTC is designed to derive the target velocity $\dot{\boldsymbol{p}}^{TT}$ based on a waypoint on a target trajectory, denoted as $\boldsymbol{p}_m^*$, by solving an optimization problem [32]. To determine the target position of anchor points on MBs, denoted as $\boldsymbol{b}_i' = [b_{i,x}', b_{i,y}', b_{i,z}']^T$, an optimization problem is defined as

$$\begin{aligned}\min_{\boldsymbol{b}_i'} \quad & w_1 \|\boldsymbol{p}_m^* - \boldsymbol{p}_m'\|_2 \\ & + w_2 \sum_{i=1,3} \| (\boldsymbol{b}_i' - \boldsymbol{b}_{i+1}) \times (\boldsymbol{p}_m' - \boldsymbol{p}_m) \|_2 \\ & + w_2 \sum_{i=2,4} \| (\boldsymbol{b}_i' - \boldsymbol{b}_{i-1}) \times (\boldsymbol{p}_m' - \boldsymbol{p}_m) \|_2 \\ & + w_3 \sum_{i=1}^{4} \sum_{\substack{j=1 \\ j\neq i}}^{4} \|\boldsymbol{b}_i' - \boldsymbol{b}_j'\|_2 \end{aligned} \tag{15}$$

$$s.t. \begin{cases} \|\boldsymbol{b}_i' - \boldsymbol{p}_m'\|_2 = \epsilon_1 & i = 1,2,3,4 \\ \|\boldsymbol{b}_i' - \boldsymbol{b}_j'\|_2 \leq \epsilon_2 & i,j = 1,2,3,4 \text{ and } i \neq j \\ \|\boldsymbol{b}_i - \boldsymbol{b}_i'\|_2 \leq \epsilon_3 & i = 1,2,3,4 \\ b_{i,z}' = \epsilon_4 & i = 1,2,3,4 \end{cases}$$

where $\boldsymbol{p}_m'$ represents a position that is supposed to be reached by the MP. The cost functions of the optimization problem are designed for moving the MP to the waypoint $\boldsymbol{p}_m^*$, guiding MBs to move in the same direction as the MP, and keeping MBs away from each other. The value of the parameters of the cost functions is determined to be $w_1 = 0.70$, $w_2 = 0.30$, and $w_3 = -0.01$ based on multiple attempts. The constraint functions of the optimization problem are designed to guarantee that the length of cables (i.e., $\epsilon_1 = 0.70$ m), the maximum distance between pairs of MBs (i.e., $\epsilon_2 = 0.60$ m), the maximum step size of MBs (i.e., $\epsilon_3 = 0.004$ m), and the height of anchor points on MBs (i.e., $\epsilon_4 = 0.90$ m) are satisfied. The value of the parameters of the constraint functions is determined according to the parameters of the CDPR with MBs, such as the length of cables, the size of MBs, and the length of vertical bars.

The target position to be reached by the $i$th MB is

$$\boldsymbol{p}_i' = \left[b_{i,x}', b_{i,y}', 0\right]^T \quad i = 1,2,3,4. \tag{16}$$

Based on (16), the target velocity of MBs is set to

$$\begin{aligned} &\left(\dot{\boldsymbol{p}}^{TT}\right)^T = \\ &K_p \left[ \left(\boldsymbol{p}_1' - \boldsymbol{p}_1\right)^T, \left(\boldsymbol{p}_2' - \boldsymbol{p}_2\right)^T, \left(\boldsymbol{p}_3' - \boldsymbol{p}_3\right)^T, \left(\boldsymbol{p}_4' - \boldsymbol{p}_4\right)^T \right]^T \end{aligned} \tag{17}$$

where $K_p = 4.00$ in the experiments.

### C. Obstacle Avoidance Controller Training

A training process is conducted in simulation to achieve an OAC. To improve the performance of an OAC that will be transferred to the real world, on one hand, one should improve the accuracy of the model of a CDPR with MBs in simulation as possible. On the other hand, one can apply techniques (e.g., normalization and noise) to enhance the generalization and robustness of the OAC. In the training process, the position of a CDPR with MBs is randomly generated. A cylinder obstacle with $d_o$ ranging from 0.25 m to 0.40 m and $h_o$ ranging from 0.20 m to 1.00 m is set to be close to the CDPR with MBs, randomly. This study randomly sets a target position that will result in a collision if the CDPR with MBs moves directly to the target position. To accelerate the training, this letter proposes a two-stage training strategy.

- In the first stage, an OAC is trained based on a reward function modified from (1) according to the reward shaping technique [27]. The reward function can be expressed as

$$r_1 = -\{\lambda_{target} f_{target}(\boldsymbol{s}_t) + \lambda_{bb} f_{bb}(\boldsymbol{s}_t)$$

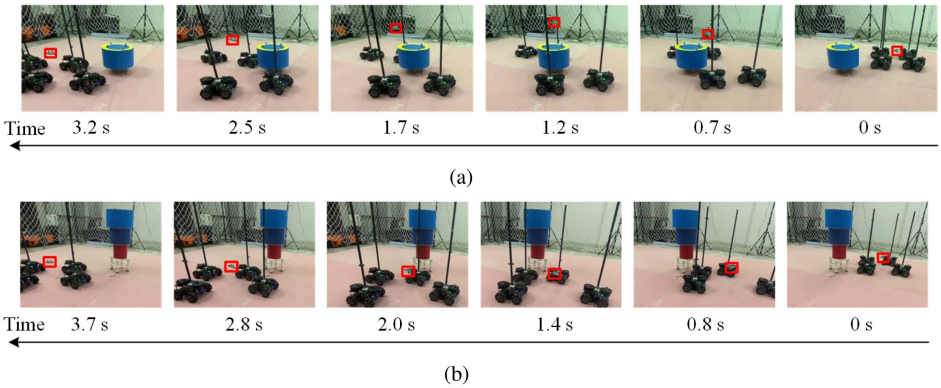


Fig. 9. Behaviors of the CDPR with mobile bases when encountering (a) a low obstacle with $h_o = 0.33$ m and $d_o = 0.32$ m in the real world, and (b) a high obstacle with $h_o = 0.92$ m and $d_o = 0.32$ m in the real world.

$$+\lambda_{\min} f_{\min}(\boldsymbol{s}_t) + \lambda_{co} f_{co}(\boldsymbol{s}_t)\}, \tag{18}$$

The first stage mainly aims to guide the OAC to move the CDPR with MBs toward a target position. If the OAC has learned to move the CDPR toward a target position, the first stage will be terminated.

- In the second stage, the OAC is trained based on a reward function defined according to (1) as

$$\begin{aligned} r_2 = & -\{\lambda_{target} f_{target}(\boldsymbol{s}_t) + \lambda_{bb} f_{bb}(\boldsymbol{s}_t) \\ & + \lambda_{bo} f_{bo}(\boldsymbol{s}_t) + \lambda_{\min} f_{\min}(\boldsymbol{s}_t) \\ & + \lambda_{\max} f_{\max}(\boldsymbol{s}_t) + \lambda_{po} f_{po}(\boldsymbol{s}_t) \\ & + \lambda_{co} f_{co}(\boldsymbol{s}_t)\}, \end{aligned} \tag{19}$$

The second stage aims to achieve an OAC that can address the dynamic obstacle avoidance problem defined in (1).

Based on the parameters listed in Table I, the two-stage training strategy is applied to achieve an OAC. As a reference, a one-stage training strategy that trains an OAC based on $r_2$ only is applied to achieve an OAC as well. The accumulated rewards achieved by the two-stage training strategy and one-stage training strategy are presented in Fig. 7. The OAC trained based on the two-stage training strategy converges within 50,000 episodes and the training of the OAC takes about 35 minutes. The OAC trained based on the one-stage training strategy converges within 500,000 episodes and the training of the OAC takes about 5.5 hours. The two OACs achieve almost the same accumulated reward eventually. It is shown that the two-stage training strategy using the reward shaping technique [27] can accelerate the training of an OAC.

### D. Dynamic Obstacle Avoidance in the Real World

To evaluate the effectiveness of the proposed RL-based dynamic obstacle avoidance method, integration of the OAC achieved in Section IV-C and the TTC presented in Section IV-B is sim-to-real transferred to a CDPR with MBs in the real world. To fuse the target velocities of MBs provided by the OAC and the TTC according to the velocity fusion module, the critical distance of switching between $\dot{\boldsymbol{p}}^{TT}$ and $\dot{\boldsymbol{p}}^{RL}$ is set to $d_\alpha = 0.80$ m and the adjustable factor is set to $k_\alpha = 0.15$. Two types of obstacles are utilized in experiments - 1) a low obstacle with $h_o = 0.33$ m and $d_o = 0.32$ m and 2) a high obstacle with $h_o = 0.92$ m and $d_o = 0.32$ m. The MP of the CDPR can go over the low obstacle from above but cannot go over the high obstacle from above. The trajectories of the MP and MBs of the CDPR when encountering the low obstacle and the high obstacle that move to the front of the CDPR and stop suddenly are shown in Fig. 8(a) and Fig. 8(b), respectively. The behaviors of the CDPR with MBs when encountering the obstacles are presented in Fig. 9. The obstacle avoidance behaviors of the CDPR with MBs in more scenarios are presented in the attached video. According to Fig. 8(a), when encountering the low obstacle, the MP of the CDPR goes over the low obstacle and the MBs of the CDPR bypass the low obstacle. The MP of the CDPR moves back to the target trajectory eventually. Fig. 8(b) shows that when encountering the high obstacle, the MP and the MBs of the CDPR bypass the high obstacle that the MP cannot go over. It is shown that with the RL-based dynamic obstacle avoidance method, the CDPR with MBs can perform obstacle avoidance behaviors in real time to avoid the obstacles and move back to target trajectories.

### E. Dynamic Obstacle Avoidance in Simulation

To investigate the difference in the performance of CDPRs with MBs in the real world and in simulation, the experiments performed in the real world are repeated in an ideal environment without time lag and inaccuracy in simulation. In the repeated experiments, the movement of obstacles in the real world is repeated in simulation. The trajectories of the MP and MBs of the CDPR when encountering 1) a low obstacle with $h_o = 0.33$ m and $d_o = 0.32$ m and 2) a high obstacle with $h_o = 0.92$ m and $d_o = 0.32$ m that move to the front of the CDPR and stop suddenly are shown in Fig. 8(c) and (d), respectively. It can be seen from Fig. 8(c) that when encountering the low obstacle in simulation, the MP of the CDPR goes over the low obstacle and the MBs of the CDPR bypass the low obstacle also. Fig. 8(d) shows that when encountering the high obstacle in simulation, the MP and the MBs of the CDPR also bypass the high obstacle that the MP cannot go over. In simulation, the CDPR with MBs performs obstacle avoidance behaviors that are similar to the obstacle avoidance behaviors performed in the real world.

## V. Conclusion

This letter proposed an RL-based dynamic obstacle avoidance method for a CDPR with MBs to deal with dynamic obstacles.

To explain the RL-based dynamic obstacle avoidance method, this letter focused on a CDPR with four fixed-length cables connected to four MBs. An RL-based OAC that addresses collisions between the MP and an obstacle, collisions between an MB and an obstacle, collisions between two MBs, collisions between the MP and an MB, and collisions between a cable and an obstacle was developed and integrated into the TTC of the CDPR with MBs. To explain and evaluate the RL-based dynamic obstacle avoidance method further, an RL-based OAC has been trained in a Mujoco simulator and has been sim-to-real transferred to a CDPR with four fixed-length cables connected to four MBs in the real world. By conducting dynamic obstacle avoidance tests in the real world, the effectiveness of the RL-based dynamic obstacle avoidance method has been verified. An RL-based OAC for a CDPR with variable-length cables connected to MBs is the next step of this study, which will be researched in the future.

## REFERENCES


[1] H. Xiong et al., "Real-time reconfiguration planning for the dynamic control of reconfigurable cable-driven parallel robots," *J. Mechanisms Robot.*, vol. 14, no. 6, 2022, Art. no. 060913.

[2] X. Tang, "An overview of the development for cable-driven parallel manipulator," *Adv. Mech. Eng.*, vol. 6, 2014, Art. no. 823028.

[3] X. Zhou, S.-K. Jun, and V. Krovi, "Tension distribution shaping via reconfigurable attachment in planar mobile cable robots," *Robotica*, vol. 32, no. 2, pp. 245–256, 2014.

[4] A. Ameri, A. Molaei, M. A. Khosravi, and M. Hassani, "Control-based tension distribution scheme for fully constrained cable-driven robots," *IEEE Trans. Ind. Electron.*, vol. 69, no. 11, pp. 11383–11393, Nov. 2022.

[5] J. Begey, L. Cuvillon, M. Lesellier, M. Gouttefarde, and J. Gangloff, "Dynamic control of parallel robots driven by flexible cables and actuated by position-controlled winches," *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 286–293, Feb. 2019.

[6] Y. Sugahara, T. Ueki, D. Matsuura, Y. Takeda, and M. Yoshida, "Offline reference trajectory shaping for a cable-driven earthquake simulator based on a viscoelastic cable model," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 2415–2422, Apr. 2022.

[7] C. Yang et al., "Collaborative navigation and manipulation of a cable-towed load by multiple quadrupedal robots," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 10041–10048, Oct. 2022.

[8] D. Sanalitro, M. Tognon, A. J. Cano, J. Cortes, and A. Franchi, "Indirect force control of a cable-suspended aerial multi-robot manipulator," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 6726–6733, Jul. 2022.

[9] L. Gagliardini, S. Caro, M. Gouttefarde, and A. Girin, "Discrete reconfiguration planning for cable-driven parallel robots," *Mechanism Mach. Theory*, vol. 100, pp. 313–337, 2016.

[10] K. Youssef and M. J.-D. Otis, "Reconfigurable fully constrained cable driven parallel mechanism for avoiding interference between cables," *Mechanism Mach. Theory*, vol. 148, 2020, Art. no. 103781.

[11] Z. Li, J. Erskine, S. Caro, and A. Chriette, "Design and control of a variable aerial cable towed system," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 636–643, Apr. 2020.

[12] T. Rasheed, P. Long, and S. Caro, "Wrench-feasible workspace of mobile cable-driven parallel robots," *J. Mechanisms Robot.*, vol. 12, no. 3, 2020, Art. no. 031009.

[13] Z. Zhang, H. H. Cheng, and D. Lau, "Efficient wrench-closure and interference-free conditions verification for cable-driven parallel robot trajectories using a ray-based method," *IEEE Robot. Automat. Lett.*, vol. 5, no. 1, pp. 8–15, Jan. 2020.

[14] J. Xu and K. S. Park, "A real-time path planning algorithm for cable-driven parallel robots in dynamic environment based on artificial potential guided RRT," *Microsyst. Technol.*, vol. 26, no. 11, pp. 3533–3546, 2020.

[15] J.-H. Bak, S. W. Hwang, J. Yoon, J. H. Park, and J.-O. Park, "Collision-free path planning of cable-driven parallel robots in cluttered environments," *Intell. Service Robot.*, vol. 12, no. 3, pp. 243–253, 2019.

[16] S. Xiang, H. Gao, Z. Liu, and C. Gosselin, "Dynamic point-to-point trajectory planning for three degrees-of-freedom cable-suspended parallel robots using rapidly exploring random tree search," *J. Mechanisms Robot.*, vol. 12, no. 4, 2020, Art. no. 041007.

[17] C. Passarini, D. Zanotto, and G. Boschetti, "Dynamic trajectory planning for failure recovery in cable-suspended camera systems," *J. Mechanisms Robot.*, vol. 11, no. 2, 2019, Art. no. 021001.

[18] T. Rasheed, P. Long, A. S. Roos, and S. Caro, "Optimization based trajectory planning of mobile cable-driven parallel robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 6788–6793.

[19] B. Wang, Z. Liu, Q. Li, and A. Prorok, "Mobile robot path planning in dynamic environments through globally guided reinforcement learning," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6932–6939, Oct. 2020.

[20] U. A. Mishra and S. Caro, "Unsupervised neural network based forward kinematics for cable-driven parallel robots with elastic cables," *Mechanisms Mach. Sci.*, vol. 104, pp. 63–76, 2021.

[21] A. Aflakian, A. Safaryazdi, M. Tale Masouleh, and A. Kalhor, "Experimental study on the kinematic control of a cable suspended parallel robot for object tracking purpose," *Mechatronics*, vol. 50, pp. 160–176, 2018.

[22] I. Chawla, P. M. Pathak, L. Notash, A. K. Samantaray, Q. Li, and U. K. Sharma, "Inverse and forward Kineto-static solution of a large-scale cable-driven parallel robot using neural networks," *Mechanism Mach. Theory*, vol. 179, 2023, Art. no. 105107.

[23] A. Grimshaw and J. Oyekan, "Applying deep reinforcement learning to cable driven parallel robots for balancing unstable loads: A. ball case study," *Front. Robot. AI*, vol. 7, no. 2, pp. 1–16, 2021.

[24] H. Xiong, T. Ma, L. Zhang, and X. Diao, "Comparison of end-to-end and hybrid deep reinforcement learning strategies for controlling cable-driven parallel robots," *Neurocomputing*, vol. 377, pp. 73–84, 2020.

[25] C. Sancak, F. Yamac, and M. Itik, "Position control of a planar cable-driven parallel robot using reinforcement learning," *Robotica*, vol. 40, pp. 3378–3395, 2022.

[26] T. Fan, P. Long, W. Liu, and J. Pan, "Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios," *Int. J. Robot. Res.*, vol. 39, no. 7, pp. 856–892, 2020.

[27] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, 2013.

[28] D. Sanalitro, H. J. Savino, M. Tognon, J. Cortés, and A. Franchi, "Full-pose manipulation control of a cable-suspended load with multiple UAVs under uncertainties," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 2185–2191, Apr. 2020.

[29] T. Haarnoja, K. Hartikainen, P. Abbeel, and S. Levine, "Latent space policies for hierarchical reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 1851–1860.

[30] M.-H. Guo et al., "Attention mechanisms in computer vision: A survey," *Comput. Vis. Media*, vol. 8, no. 3, pp. 331–368, 2022.

[31] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 5026–5033.

[32] J. M. Filho, E. Lucet, and D. Filliat, "Real-time distributed receding horizon motion planning and control for mobile multi-robot dynamic systems," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 657–663.