Complex & Intelligent Systems (2023) 9:6985–6998
https://doi.org/10.1007/s40747-023-01114-3

ORIGINAL ARTICLE

Optimal sampling-based path planning for mobile cable-driven parallel robots in highly constrained environment

Jiajun Xu[1] · Byeong-Geon Kim[1] · Yuzhen Lu[2] · Kyoung-Su Park[1]

Received: 29 December 2022 / Accepted: 16 May 2023 / Published online: 9 June 2023

Abstract

Mobile cable-driven parallel robots (MCDPRs) is a novel concept of cable-driven parallel robots (CDPRs) developed by mounting several mobile bases to discrete the conventional fixed frame. However, the additional mobile bases introduce more degree-of-freedom (DoF), thereby causing the kinematic redundancy. Moreover, mobile bases are susceptible to disturbances causing inconsistent performance. Hence, path planning of MCDPRs becomes a challenging issue due to various internal and external constraints involved. In this article, an optimization-based path planning method is proposed for MCDPRs in highly constrained environments with considering kinematic stability. The proposed approach quickly generates feasible paths for coupled mobile bases by using the developed multi-agent rapidly exploring random tree (MA-RRT). In this process, the tree can share information through the heuristics method to optimize the path, and the adaptive sampling strategy is thus proposed to increase the tree growth efficiency by self-adjusting sampling space. Moreover, the developed dynamic control checking method (DCC) is integrated with MA-RRT to smooth the path and the kinodynamic constraints of mobile bases can be satisfied. To generate the path of the end-effector, two performance metrics are designed considering the kinematic and stability of the MCDPR. Based on the performance metrics, the grid-based search method is developed to generate the path for the end-effector. Finally, the convincing performance of the proposed method is revealed through the dynamic simulation software (CoppeliaSim) and real-world experiments based on a self-built MCDPR prototype.

Keywords Mobile cable-driven parallel robot · Rapidly exploring random tree · Stability and kinematic performance · Optimal sampling-based path planning

Introduction

Cable-driven parallel robots (CDPRs) are categorized as a particular type of parallel manipulator. In CDPRs, multiple flexible cables that reel around a fixed winch-pulley system are responsible for actuating the end-effector to the desired position. Owing to the large workspace and higher payload-to-weight ratio, CDPRs are promising for several industrial applications such as picking and placing heavy loads, grasping, space and terrestrial applications [1–3].

Although CDPRs provide significant advantages in numerous applications, several challenges still exist. As an example, CDPRs are generally available with a large workspace, which imposes a strong requirement for the safety issue [4]. The radiate structure of cables increases the risks of cable–cable and cable–environment interferences, especially when CDPRs have a large number of cables in a cluttered environment [5]. Due to the conventional CDPRs having a fixed frame, the cable span dominates a large space during the motion of the end-effector [6], which results in restricted flexibility for CDPR.

Consequently, CDPRs require further development to improve adaptation to environmental and task requirements [7]. Based on the concept of reconfiguration, reconfigurable

✉ Kyoung-Su Park
pks6348@gachon.ac.kr

Jiajun Xu
jilinxujiajun@163.com

Byeong-Geon Kim
bk0725@naver.com

Yuzhen Lu
luyuzhen@msu.edu

[1] Department of Mechanical Engineering, Gachon University, 1342 Seongnamdaero, Sujeong-gu, Seongnam-si, Gyeonggi-do 461-701, Korea

[2] Department of Biosystems & Agricultural Engineering, Michigan State University, East Lansing, MI 48824, USA

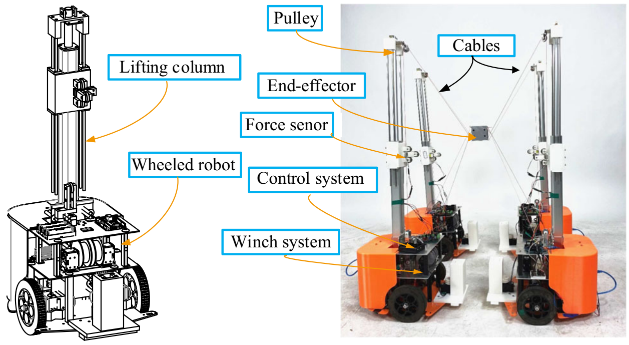


Fig. 1 A self-built MCDPR prototype that is designed to perform a variety of manipulation tasks in a complex environment

cable-driven parallel robots (RCDPRs) were developed, in which the cable's attachment points on the frame and the cable's anchor points on the end-effector of pulleys can be altered [8]. However, the existing RCDPR systems have limited reconfigurability, such as having a guideway with limited length or requiring manual intervention. These drawbacks of conventional RCDPRs lead the reconfigurable operations to be time-consuming and costly to use.

To extend the reconfigurability of RCDPRs, as shown in Fig. 1, a mobile cable-driven parallel robot (MCDPR) is developed in this work to make it automatically be configured according to the desired task. The developed MCDPR is composed of a six degree-of-freedom (DoF) end-effector pulled by eight cables mounted on four mobile bases. A mobile base consists of a differential wheeled robot and a lifting column. Thus, MCDPR can be moved freely with the wheeled robots, and the position of the pulley can be adjusted by using the lifting column.

However, the kinematical redundancy of MCDPRs is increased due to more DoF given by additional mobile bases. These mobile bases are coupled to each other by multiple cables, resulting in complex constraints and a high-dimension regime. In addition, the mobile base is susceptible to cable tension and becomes unstable, which also affects the available workspace for MCDPRs [9]. Hence, the path planning of MCDPRs becomes a challenging problem and requires to be addressed. In consequence, for displacing the end-effector from the starting point to its destination, feasible and optimal paths are required to be generated for each mobile base and the end-effector simultaneously.

For CDPRs, path planning is a challenging issue due to various constraints such as cable tension, workspace, collisions, and obstacle avoidance. In particular, MCDPRs involve in the high-dimensional cooperative pathfinding problem as a result of the coupled mobile bases and end-effector. Accordingly, Jiang et al [10] developed a dynamic point-to-point trajectory planning point trajectory planning method for a CDPR with six DoF. More recently, Idà et al. [11] proposed a rest-to-rest trajectory planning method of CDPRs with the given motion time and path profile. However, these studies only considered an underactuated CDPR in an environment without obstacles.

In a cluttered environment, it is critical for CDPRs to avoid obstacles. However, MCDPRs involve multiple moving parts such as cables and the end-effector must avoid the obstacles simultaneously. Furthermore, the multiple mobile bases of MCDPRs introduce more constraints increasing the complexity of the path planning problem. On the other hand, the sampling-based path planning algorithms perform well to deal with various constraints by randomly sampling the state space [12, 13]. Compared with other path planning techniques, they need less time to conduct the computation and can avoid obstacle accurately. The most representative sampling-based path planning algorithm is the Rapidly exploring Random Tree (RRT) method, which relies on a feasible checking module to generate a series of candidate trajectories for constructing a graph framework [14]. However, the conventional RRT method generally generates a path that comprises a significant number of redundant nodes. To address the optimality issue, some extensions of RRT like RRT* and informed RRT* are developed to obtain an asymptotically optimal solution [15, 16].

Accordingly, Bak et al. [17] proposed a modified RRT method to generate feasible paths for CDPRs, which takes advantage of the goal-biased sampling method and post-processing to mitigate the length of the path. Zhang et al. [18] designed a new dexterity-based cost function to accomplish collision-free path planning for CDPR by combining RRT* and potential field guidance (PFG) methods. Recently, Mishra et al. [19] proposed an AFG-RRT* method to generate optimized paths for CDPRs in cluttered environments. However, the above-cited methods only deal with the path planning for the end-effector, the multiple mobile bases and additional constraints are not involved.

For MCDPRs, the additional mobile bases lead to kinematical redundancy, and the path planning problem for MCDPRs becomes incrementally challenging. Rasheed et al. [20] developed an iterative algorithm to generate available paths for MCDPR. This approach is straightforward to traverse all possible pre-defined states of each mobile base, and the next state is identified by maximizing the wrench capacity. However, the traversal time cost increases exponentially with the possible states of the mobile bases and the quality of the obtained path is not desirable. To improve the path quality, the Direct Transcription method was proposed in [21], in which the trajectory is discretized according to the specific time step and the resulting paths are obtained by minimizing the goal function with a set of given constraints, but the optimization-based method is commonly time-consuming in terms of convergence.

In this work, we present an optimal sampling-based path planning algorithm for MCDPR. The proposed approach is capable of efficiently handling the multiple constraints and kinematic redundancies associated with the robot. In summary, the contributions of this paper are listed as follows:

(1) First, an innovative high-dimensional tree structure is proposed in this paper. A distinctive characteristic is that we maintain the tree growth in the high-dimensional state space while preserving the tree to share information in the low-dimensional domain. Based on the growth state of the tree, we deploy a heuristic cost that allows the tree to be sampled adaptively in the subspace. Hence, the proposed method can generate the feasible path in a short time.

(2) Second, various constraints of MCDPR such as mobile base/mobile base, mobile base/obstacle, cable/obstacle collisions, and the available configuration are considered. The dynamic control checking method is developed to smooth the path and satisfy kinodynamic constraints of the robot. Moreover, the heuristic method is designed to optimize the path and ensure the continuity of parent nodes. The optimal and smooth path can thus be obtained.

(3) Third, two performance metrics are developed considering the stability and kinematics of MCDPR. Based on the performance metrics, the path of the end-effector is generated by using the proposed grid-search method. Finally, the resulting paths are simulated in complex scenarios using the MCDPR dynamic model developed by CoppeliaSim software and validated by prototype experiments.

The remainder of this paper is organized as follows: Robot parameterization and problem formulation are introduced in Section "System modeling" . The kinematics and stability model for MCDPRs is established in Section "Consideration" . In Section "Method description", the proposed path planning method for MCDPRs is introduced in detail. The method validation is carried out in Section "Method validation". Finally, Section "Conclusion and future work" concludes the research work.

System modeling

MCDPR parameterization

The self-bulit MCDPR in this work is composed of a classical redundantly constrained CDPR with eight cables ($m = 8$) and a six DoF moving-platform ($n = 6$) mounted on four mobile bases ($p = 4$). As depicted in Fig. 2, the origin of the global coordinate system is represented as O_0 and the local coordinate system located at the center of the end-effector is denoted as O_e. The j-th mobile base, denoted as $\mathscr{M}_j$, $j = 1, ..., 4$ carries two cables ($m_j = 2$) and has three wheels ($c = 3$) including one caster wheel and two driven wheels. The local coordinate system attached to $\mathscr{M}_j$ is denoted as O_{pj}. The mobile base is characterized as the non-holonomic differential drive mechanism, it can provide two DoF translational motions and one DoF rotational motion.

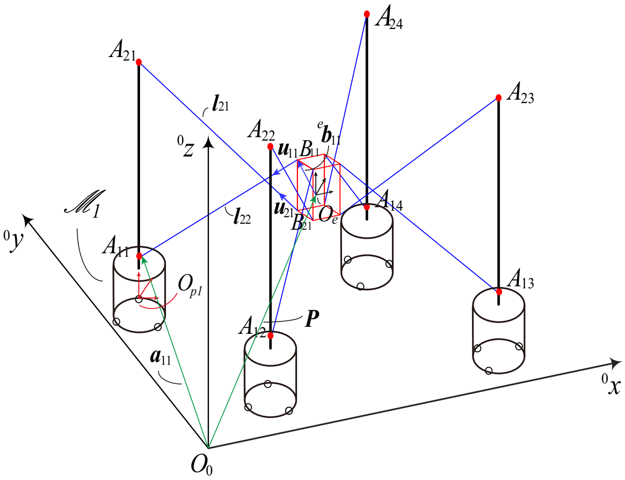


Fig. 2 Structural diagram of MCDPR

Cables are assumed to be straight and massless. The end-effector is modeled as a cube. The cable exit point and the anchor point are represented as A_{ij} and B_{ij}, respectively. Thus, i-th cable vector attached onto $\mathscr{M}_j$ is denoted as $\boldsymbol{l}_{ij}$, $i = 1, ...m_j$.

Selecting the i-th closed-loop vector branch, $\boldsymbol{l}_{ij}$ can be expressed as follows

$$\boldsymbol{l}_{ij} = \boldsymbol{a}_{ij} - \boldsymbol{P} - \boldsymbol{R} \cdot {}^e\boldsymbol{b}_{ij} \tag{1}$$

where $\boldsymbol{a}_{ij}$ is the position vector of A_{ij} in O_0 coordinate system. $\boldsymbol{P}$ and $\boldsymbol{R}$ is a three-dimensional position vector and rotation matrix of the end-effector, respectively. ${}^e\boldsymbol{b}_{ij}$ represents the position vector of B_{ij} in the moving coordinate system O_e. To simplify the problem, we restrict the end-effector only perform the translational motion in this work.

Problem formulation

Due to the MCDPR is designed such that A_{ij} is lie on the axis ${}^{pj}z$ of $\mathscr{M}_j$, the rotational of $\mathscr{M}_j$ has no effect on $\boldsymbol{u}_{ij} = \boldsymbol{l}_{ij}/\|\boldsymbol{l}_{ij}\|$, which is the unit vector along the cable $\boldsymbol{l}_{ij}$. We consider the state of mobile bases $\mathscr{M}_j$ at k-th time step is $\boldsymbol{m}_{j,k} = [{}^0x_{j,k}, {}^0y_{j,k}]$, $k = 1, ..., N$. The state of the muti-robot system constructed by mobile bases at k-th time step is represented as $\boldsymbol{M}_k = [\boldsymbol{m}^T_{1,k}, \boldsymbol{m}^T_{2,k}, \boldsymbol{m}^T_{3,k}, \boldsymbol{m}^T_{4,k}]^T$ in $\mathbb{R}^{2p}$ and the state of the end-effector at k-th time step is $\boldsymbol{P}_k =$

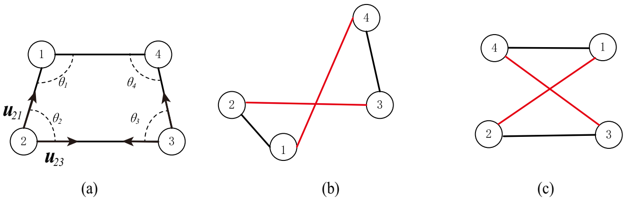


Fig. 3 MCDPR configurations

$[{}^0x_k, {}^0y_k, {}^0z_k]$. Consequently, the state of MCDPR at k-th time step can be denoted as $\boldsymbol{X}_k = [\boldsymbol{P}_k^T, \boldsymbol{M}_k^T]^T$.

The MCDPR involves various internal and external constraints that require to be considered during the path planning. To facilitate the computation, the mobile base and the obstacles are modeled as cylindrical structures. Hence, at each time step, the following constraints are imposed for j-th mobile base

$$\left\|\boldsymbol{m}_{h,k} - \boldsymbol{m}_{j,k}\right\| > L_m, \text{ for } h = 1, .., 4, \; h \neq j \tag{2}$$

$$\left\|\boldsymbol{m}_{j,k} - \boldsymbol{o}_q\right\| > r_m + r_q, \text{ for } q = 1, .., s, \tag{3}$$

$$\|\boldsymbol{c}_k - \boldsymbol{b}_k\| > L_c \tag{4}$$

$$l_{min} \leq \left\|[\boldsymbol{m}_{j,k}{}^T \; h_i]^T - \boldsymbol{P}_k\right\| \leq l_{max}, \text{ for } i = 1, 2 \tag{5}$$

$$\boldsymbol{P}_k \in \mathcal{W}_k \tag{6}$$

Where (2–4) impose the distance constraints of the MCDPR in terms of mobile base/mobile base, mobile base/obstacle and cable/obstacle. L_m denotes the safety distance between two mobile bases. r_m and $r_q, q = 1, ..., s$, represent the radius of the mobile base and the q-th obstacle, respectively. $\boldsymbol{o}_q$ represents the position of q-th obstacle. $\boldsymbol{c_k}$ and $\boldsymbol{b_k}$ denotes the two closest point between MCDPR and obstacles at kth time step, respectively. Hence, (2) and (3) prevent the collision between mobile bases/mobile bases and mobile bases/obstacles. (4) prevent the collision between cables/end-effector with obstacles by using the minimum acceptable distance L_c. Furthermore, (5) and (6) indicate that the CDPR have limited workspace and cable length, respectively. Let l_{min} and l_{max} denote the minimum and maximum length of cable. $h_l, l = 1, 2$, denotes two constant heights of exit point $A_{i,j}$ in the O_0 coordinate system. For each $\boldsymbol{M}_k$, the available workspace of the end-effector is given by $\mathcal{W}_k$.

Additionally, the MCDPR also requires to satisfy the non-holonomic and configuration constraints. Let $\boldsymbol{d}_k = \boldsymbol{m}_{j,k} - \boldsymbol{m}_{j,k-1}$ denotes the directional vector of $\mathscr{M}_j$ between two adjacent states, (7) prevents the abrupt direction change of the mobile base to satisfy the non-holonomic constraints by using the maximum turning angle β_{max}. As a result of mobile bases are coupled with cables, the MCDPR should maintain the initial formation shown in Fig. 3a, and the infeasible formation illustrated in Fig. 3b and c should be prevented to cause internal interferences, which indicates that the intersection of the line segment formed by connecting two mobile bases. We consider $\theta_j \in [0, 2\pi]$ is the interior angle of $\mathscr{M}_j$ shown in Fig. 3a is defined by (9), and $\theta_j = \theta_j + 2\pi$ if $\theta_j < 0$, (8) ensures the mobile bases do not cross each other during the motion.

$$\beta_{j,k} = \cos^{-1}\frac{\boldsymbol{d}_{j,k-1} \cdot \boldsymbol{d}_{j,k}}{\|\boldsymbol{d}_{j,k-1}\|\|\boldsymbol{d}_{j,k}\|} < \beta_{max} \tag{7}$$

$$\theta_{j,k} \in [0, \pi] \tag{8}$$

$$\theta_j = \text{atan2}(\boldsymbol{u}_{j,y}, \boldsymbol{u}_{j,x}) - \text{atan2}(\boldsymbol{u}_{h,y}, \boldsymbol{u}_{h,x}) \tag{9}$$

The composite state space of MCDPR $\boldsymbol{X} = \boldsymbol{M} \times \boldsymbol{P} \subset \mathbb{R}^{2p+3}$ is the Cartesian product of the individual spaces. We denote the free composite state space that satisfies constraints defined in (2–8) is $\boldsymbol{X}_{free} \subseteq \boldsymbol{X}$. Then, the optimal path planning problem for MCDPR is to find the feasible trajectory $\pi : [0, t] \rightarrow \boldsymbol{X}_{free}$ from an initial state $\pi(0) = \boldsymbol{X}_{init}$ to the goal region $\pi(t) \in \boldsymbol{X}_{goal}$ that minimizes a given optimal function and obeys the system dynamics. In this work, the optimal function is defined to minimize the total path length.

Consideration

Kinematics performance

Due to additional mobile bases lead the kinematical redundancy of the MCDPR, the kinematics performance indicator is designed in this subsection to ensure excellent kinematics performance. Derivative of (1) with respect to time and lead the end-effector only perform translational motion, the velocity of the end-effector $\dot{\boldsymbol{P}} = [{}^0\dot{x}_e, {}^0\dot{y}_e, {}^0\dot{z}_e]^T$ can be expressed as follows

$$\begin{bmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \\ \vdots \\ \boldsymbol{J}_j \end{bmatrix} \dot{\boldsymbol{P}} = \begin{bmatrix} \dot{\boldsymbol{l}}_1 \\ \dot{\boldsymbol{l}}_2 \\ \vdots \\ \dot{\boldsymbol{l}}_j \end{bmatrix} \tag{10}$$

where $\dot{\boldsymbol{l}}_j = [\dot{\boldsymbol{l}}_{1j}, ..., \dot{\boldsymbol{l}}_{\boldsymbol{m}_j j}]^T$ denotes the speed vector of the cables on $\mathscr{M}_j$, and $\boldsymbol{J}_j$ can be expressed as

$$\boldsymbol{J}_j = \begin{bmatrix} \boldsymbol{u}_{1,j}{}^T \\ \vdots \\ \boldsymbol{u}_{m_j j}{}^T \end{bmatrix} \tag{11}$$

where $\boldsymbol{J}_j \in \mathbb{R}^{m_j \times 2}$ is the sub-Jacobian matrix associated with the cable actuation wrench on $\mathscr{M}_j$. The kinematic jacobian matrix of the carried CDPR can be expressed as $\boldsymbol{J} = [\boldsymbol{J}_1, ..., \boldsymbol{J}_j]^T$, which is associated with a specific position of the end-effector to map relationship from the end-effector velocity to the cable velocities.

Based on the kinematic jacobian matrix of the carried CDPR, the kinematic performance indicator is defined as the dexterity of the mechanism, which measures the accuracy of transmission between cable space and the Cartesian space of the end-effector. The dexterity of the robot can be obtained by the condition number of $\boldsymbol{J}$, denoted as

$$\kappa = ||\boldsymbol{J}|| \cdot ||\boldsymbol{J}^{-1}|| \tag{12}$$

$$||\boldsymbol{J}|| = \sqrt{\lambda_{max}(\boldsymbol{J}^T \boldsymbol{J})} = \sigma_{max}(\boldsymbol{J}) \tag{13}$$

where, λ_{max} is the maximum eigenvalues of square matrix $\boldsymbol{J}^T \boldsymbol{J}$, and σ_{max} is the maximum singular value of $\boldsymbol{J}$. The condition number of kinematic Jacobian matrix $\kappa \in [1, \infty)$. Thus, κ is normalized to define the kinematics performance index γ_k expressed as

$$\gamma_k = 1/\kappa \in [0, 1] \tag{14}$$

The better the kinematics performance can be obtained when γ_k is closer to 1, and the robot loses its control when $\gamma_k = 0$. Consequently, the larger γ_k should be remained during the path planning.

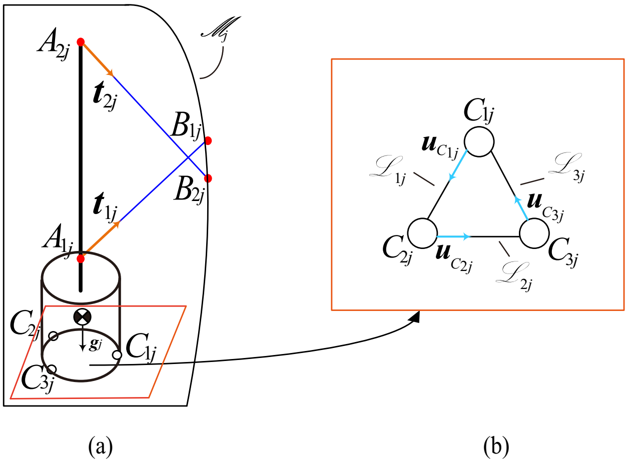
Stability performance

The additional mobile bases lead the MCDPR highly flexible. However, the mobile base is susceptible to cable tension and becomes unstable. Thus, the static equilibrium of the mobile base with its tipping condition is analyzed in this section to guarantee the better stability performance.

Figure (4) illustrates the free body diagram of j-th mobile base. $\boldsymbol{t}_{ij} = -t_{ij}\boldsymbol{u}_{ij}$ denote the cable tension vector along the cable $\boldsymbol{l}_{ij}$, and t_{ij} is the cable tension. We define the wheel contact points of the mobile base $\mathscr{M}_j$ as C_{nj}, $n = 1, 2, 3,$ in a counter-clockwise direction, and $\boldsymbol{c}_{nj}$ denotes the Cartesian coordinate vector of C_{nj}. The boundary between two successive contact points C_{nj} and C_{n+1j} is expressed as the $\mathscr{L}_{C_{nj}}$, and the direction is described by $\boldsymbol{u}_{C_{nj}}$. The equilibrium towards the tipping of a wheeled robot can be calculated by using Zero-Moment Point (ZMP) method [22], which depends on the moments generated at the boundaries. We consider $\boldsymbol{m}_{C_{nj}}$ is the moment when $\mathscr{M}_j$ loses contact with the ground that does not constitute the boundary $\mathscr{L}_{C_{nj}}$,

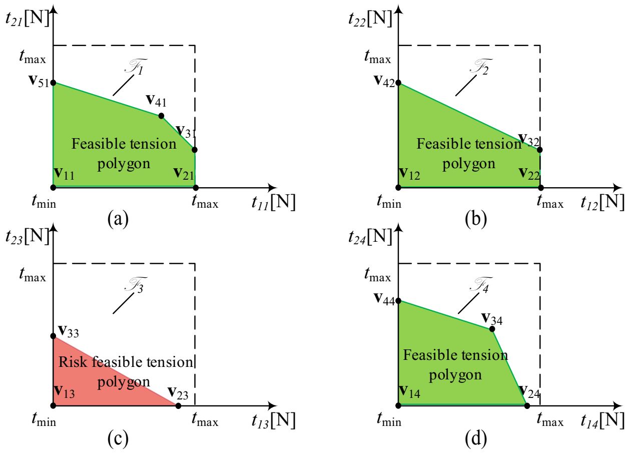
$$\begin{aligned} \boldsymbol{m}_{C_{nj}} =& \boldsymbol{u}_{c_{nj}}^T((\boldsymbol{g}_j - \boldsymbol{c}_{kj}) \times \boldsymbol{w}_{gj}) \\ &+ \sum_{i=1}^{m_j} \boldsymbol{u}_{C_{nj}}^T((\boldsymbol{c}_{kj} - \boldsymbol{P}) \times \boldsymbol{u}_{ij})t_{ij} \end{aligned} \tag{15}$$

where $\boldsymbol{g}_j$ denotes the Cartesian coordinate vector of the center of gravity in $\mathscr{M}_j$. $\boldsymbol{w}_{gj}$ is the weight vector of $\mathscr{M}_j$. To guarantee $\mathscr{M}_j$ in static equilibrium, the following constraints are required to be satisfied.



(a) (b)

Fig. 4 **a** Free body diagram of j-th mobile base. **b** Distribution of wheel contact points



(a) (b) (c) (d)

Fig. 5 Feasible cable tension polygon of $\mathscr{M}_1$, ..., $\mathscr{M}_4$

$$m_{C_{nj}} < 0, \ n = 1, 2, 3 \tag{16}$$

Equation (16) denotes the tipping conditions of $\mathscr{M}_j$, it can be regarded as linear constraints in terms of the cable tensions t_{ij}. In general, the cable tensions are all bounded between a minimum and positive tension t_{min} and a maximum tension t_{max}. The feasible tension space of $\mathscr{M}_j$ is represented as a square with side length $\Delta t = t_{max} - t_{min}$. However, as shown in Fig. 5, the linear constraints in (16) reduce the feasible tension space to form the feasible tension polygon. We denote the area of $\mathscr{M}_j$'s feasible tension polygon is $\mathscr{F}_j$. Since the cable tension has infinite solutions due to redundant actuation, we define the riskiest feasible tension polygon as

$$\mathscr{F}_r = \min(\mathscr{F}_r), \ j = 1, ..., 4 \tag{17}$$

where $\mathscr{F}_r$ indicates the feasible tension polygon with the smallest area, which can be obtained by corresponding vertices(i.e., $\mathbf{v}_{13}$, $\mathbf{v}_{23}$, $\mathbf{v}_{33}$ in Fig. 5c). This polygon corresponds to a mobile base that is the most unstable to the cable tension and has a risk of tipping phenomenon. Hence, the stability performance of MCDPR can be defined as

$$\gamma_s = \mathscr{F}_r / \Delta t^2 \in [0, 1] \tag{18}$$

It can be observed from (18) that MCDPR is at the nest stability performance when $\gamma_s = 1$. However, when $\gamma_s = 0$, the MCDPR will tip and lose control due to the cable tension solution not satisfying the static equilibrium.

Method description

In this section, the path planning method of MCDPRs is described in detail. In Algorithm 1, the muti-agent RRT is proposed to generate feasible path $\boldsymbol{M}_{path} = ({}^p\boldsymbol{M}_1, {}^p\boldsymbol{M}_2, ..., {}^p\boldsymbol{M}_N)$ for mobile bases. For each state of mobile bases on the path ${}^p\boldsymbol{M}_k$, the position of end-effector ${}^p\boldsymbol{P}_k$ is determined by using the grid-based search method presented in Algorithm 3 with considering kinematic stability. Moreover, the corresponding high-level diagram of the proposed method is depicted in Fig. 6.

Algorithm 1 Multi-agent RRT

```
Input: An initial state $\boldsymbol{M}_{init}$, a goal state $\boldsymbol{M}_{goal}$, mobile base radius
    $r_m$ obstacles position $\boldsymbol{o}_q$, obstacles radius $r_o$, optimization radius $r*$
Output: RRT tree $\mathcal{T}$
 1: Check the feasibility of the initial state and goal state
 2: Initialize tree $\mathcal{T} \leftarrow \boldsymbol{M}_{init}$
 3: $\boldsymbol{M}_{new} \leftarrow \boldsymbol{M}_{init}$
 4: $\boldsymbol{M}_{new}$.GoalArrived = $[0, 0, 0, 0]^T$,
    $\boldsymbol{M}_{new}$.FailureRate = $[0, 0, 0, 0]^T$,
    $\boldsymbol{M}_{new}$.GoalExtend = $[0, 0, 0, 0]^T$
 5: while $\boldsymbol{M}_{new}$.GoalArrived $\neq [1, 1, 1, 1]^T$ do
 6:   $\boldsymbol{M}_{rand} \leftarrow$ AdaptiveSampling($\boldsymbol{M}_{goal}$,$\boldsymbol{M}_{parent}$,$\mathcal{T}$)
 7:   $\boldsymbol{M}_{nearest} \leftarrow$ Nearest( $\boldsymbol{M}_{rand}$,$\mathcal{T}$,$V_j$)
 8:   $u_{new}$, $\boldsymbol{M}_{new} \leftarrow$Steer($\boldsymbol{M}_{nearest}$,$\boldsymbol{M}_{rand}$)
 9:   if $\boldsymbol{M}_{new}$ is feasible then
10:     Update $\boldsymbol{M}_{nearest}$.GoalExtended
11:     $\mathcal{M}_{near} \leftarrow$ NearestNeighbors($\boldsymbol{M}_{new}$,$\mathcal{T}$,$r*$)
12:     $c_{min} = h(\boldsymbol{M}_{new}) + dist(\boldsymbol{M}_{new}, \boldsymbol{M}_{nearest})$
13:     for each $M_{near} \in \mathcal{M}_{near}$ do
14:       $(V_{new,j}$,$E_{new,j}) \leftarrow$ Connect($\boldsymbol{m}_{near,j}$,$\boldsymbol{m}_{new,j}$)
15:       if Path($\boldsymbol{m}_{near,j}$,$\boldsymbol{m}_{new,j}$) is feasible then
16:         $c_{new,j} = h(\boldsymbol{m}_{near,j}) + dist(\boldsymbol{m}_{new,j}, \boldsymbol{m}_{near,j})$
17:         if $c_{new,j} < c_{min,j}$ then
18:           $c_{min,j} \leftarrow c_{new,j}$,
              $\boldsymbol{M}_{near} \leftarrow$ReSetParent($m_{near,j}$,$\mathcal{T}$)
19:           if $\boldsymbol{M}_{near}$ is feasible then
20:             Continue
21:           end if
22:         end if
23:       end if
24:     end for
25:     $\mathcal{T}.\mathcal{V} \leftarrow$InsertVertexToTree($\boldsymbol{M}_{new}$, $\mathcal{T}$)
26:     $\mathcal{T}.\mathcal{E} \leftarrow$InsertEdgeToTree($\boldsymbol{M}_{new}$,$\boldsymbol{M}_{parent}$,$\mathcal{T}$)
27:     $\boldsymbol{M}_{new}$.GoalArrived $\leftarrow$CheckDistance($\boldsymbol{M}_{new}$,$\boldsymbol{M}_{goal}$)
28:     if $\boldsymbol{M}_{new}$.GoalArrived is changed then
29:       Record the current number of vertices $V_j$, $j = 1, ...4$ to
          construct a remained tree.
30:     end if
31:   else
32:     Update $\boldsymbol{M}_{new}$.Failure_rate
33:   end if
34: end while
```

Path planning for mobile bases

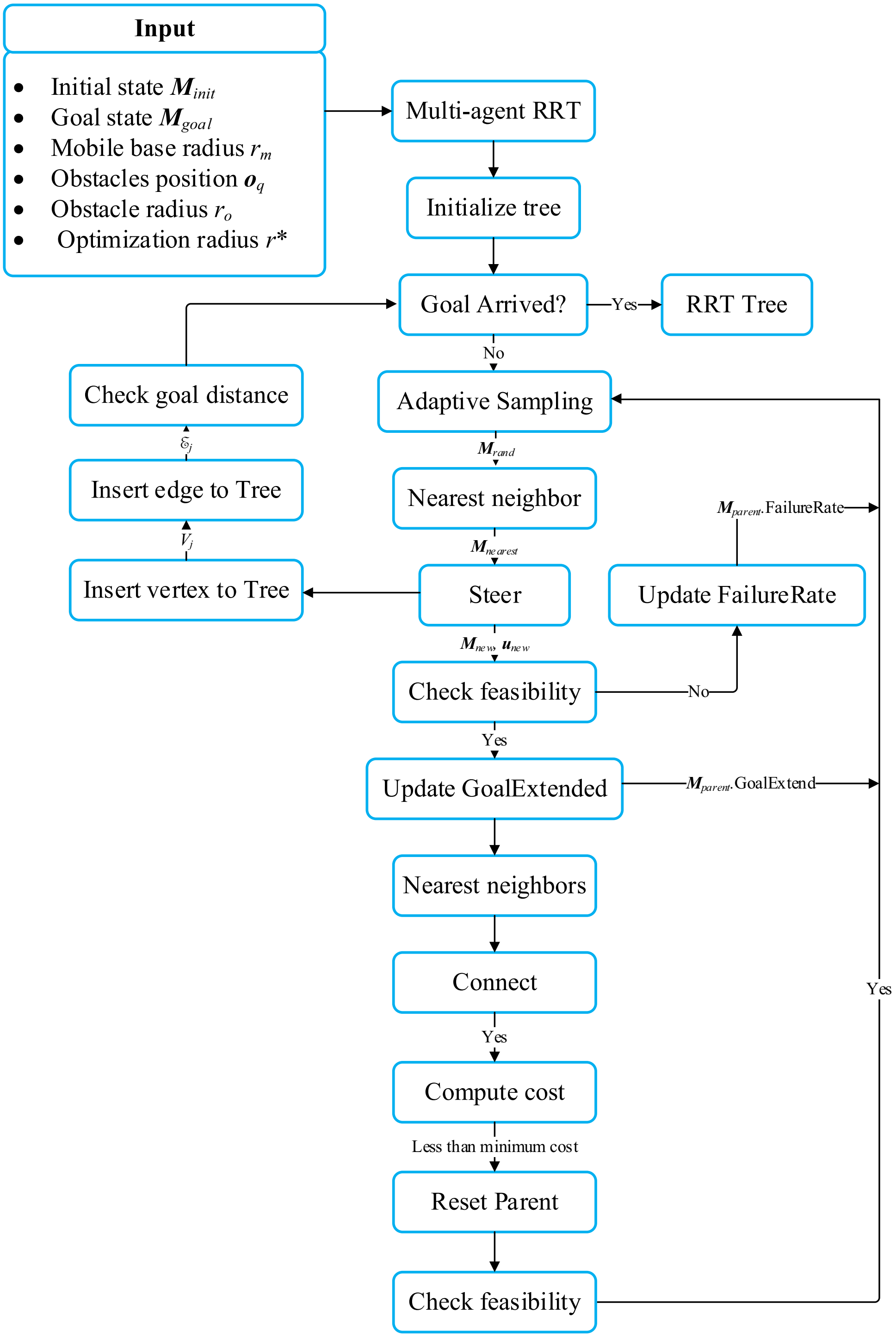
The proposed multi-agent method aims to build a tree-structured graph $\mathcal{T}(\mathcal{V}, \mathcal{E}) = (\mathcal{T}_1, ..., \mathcal{T}_4)$ by randomly sampling the state space. The vertices $\mathcal{V}$ represent feasible states of mobile bases $\boldsymbol{M}_k$, and the edge $\mathcal{E}$ denotes the feasible transitions of two adjacent states. $\mathcal{T}_j$, $j = 1, ..., 4$, denotes a sub-tree of $\mathscr{M}_j$.

In Algorithm 1, the feasibility of the initial and goal state of mobile bases are checked first. Then, some attributes are assigned to vertices. $\boldsymbol{M}_{new}$.GoalArrived is used to determine the arrival status of four mobile bases. The j-th element of $\boldsymbol{M}_{new}$.GoalArrived is changed to 1 when $\mathscr{M}_j$ arrive its goal point, and $\mathscr{M}_j$ becomes fixed. M_{new}.FailureRate is the ratio of the number of failed expansions to total expansions. $\boldsymbol{M}_{new}$.GoalExtend is used to record whether the node can expand toward the goal point, and it changes to 1 if the expansion fails.

To solve the high-dimensional state space and increase efficiency, the adaptive sampling method (line 6 of Algorithm 1) is designed in Algorithm 2. The random node of $\mathscr{M}_j$,$\boldsymbol{m}_{rand,j}$, is assigned as the goal point if the parent node can successfully expand to the goal point. Otherwise, the proposed method can adaptively select two sampling methods based on $\boldsymbol{M}_{parent}$.GoalExtend and $\boldsymbol{M}_{parent}$.FaiureRate. It should be noted that the proposed method is sampling in the subspace for each mobile base and integrated into high-dimensional space.

The ForwardSector($\boldsymbol{m}_{goal,j}$, r_s,θ_s) expands the sampling space to a sector to guide the tree towards the goal space in which the vertex is the goal point and $\boldsymbol{m}_{c,j}$ denotes the subnode with maximum cost. r_s and θ_s represent the radius and angle of the sector, representatively.The BackwardBall($\boldsymbol{m}_{centor}$, r_b) represents uniformly sampling in a circular area centered on the $\boldsymbol{m}_{centor}$ with radius r_b, and $\boldsymbol{m}_f$ is the sub-node with maximum failure rate. Thus, the proposed method favors BackwardBall($\boldsymbol{m}_{centor}$, r_b) with the $\boldsymbol{M}_{parent}$.FaiureRate increasing.

Fig. 6 High-level diagram of the proposed method



Dynamic control checking and heuristic path optimization

In contrast to the conventional RRT, the Nearest($\boldsymbol{M}_{rand}$,$\mathcal{T}$,V_j) function (line 7 of Algorithm 1) returns a node $\boldsymbol{M}_{nearest}$ closest to $\boldsymbol{M}_{rand}$ by searching the remained tree to lead the mobile base fixed after it reaches the target, and the starting node of the remained tree is recorded as V_j (line 29 of Algorithm 1). In addition, to smooth the path and satisfy the kinodynamic constraints, the Steer($\boldsymbol{M}_{nearest}$,$\boldsymbol{M}_{rand}$) function in Algorithm 1 is developed based on the dynamic control checking method. As shown in Fig. 7a, each node on the tree has a set of possible control output $\mathcal{U}(u_k, \epsilon)$, which is constrained by the control input from the parent node and the

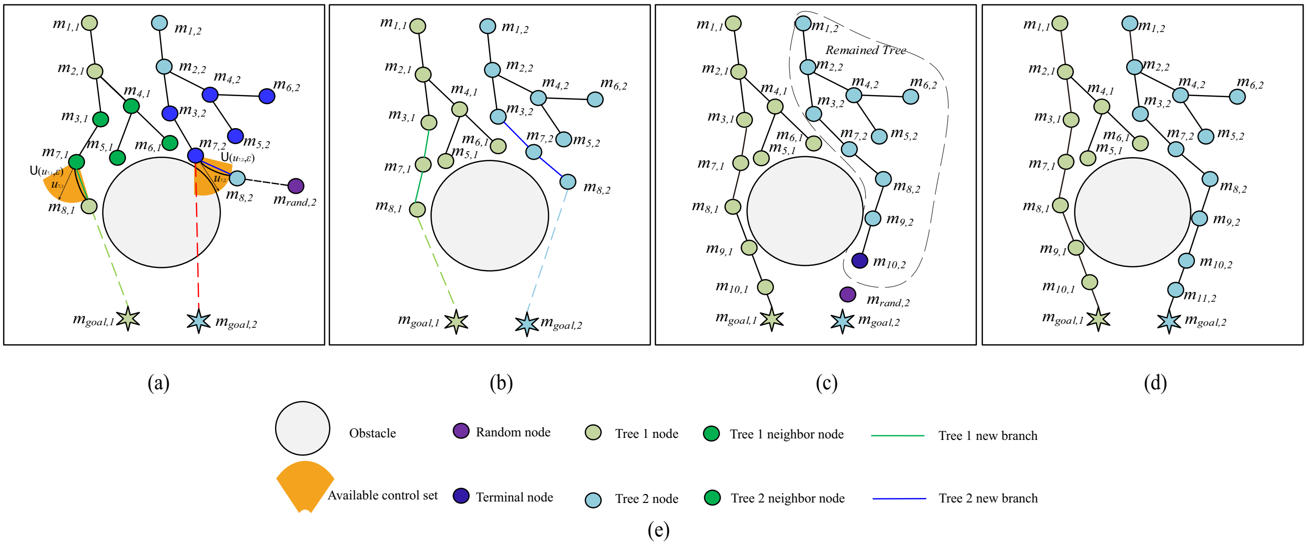


Fig. 7 Illustrations of proposed sampling-based path planning method for MCDPR. Every node on the tree is feasible in its time stamp according to the prediction of the moving obstacle. **a** Tree growth by using adaptive sampling and DCC method. **b** Tree was optimized by using the heuristic method. **c** One subtree reached its target. **d** Tree is constructed successfully after all subtrees reached the target

```
Algorithm 2 AdaptiveSampling
Input: $M_{goal}, M_{parent}$
Output: $M_{rand}$
for $j = 1$ to 4 do
    if $M_{parent}$.GoalExtended[$j$] $= 0$ then
        $m_{rand,j} \leftarrow m_{goal,j}$
    else
        $r_{rand} \leftarrow rand()$
        if $r_{rand} > M_{parent}$.FailureRate[$j$] then
            $s = (m_{start,j} - m_{goal,j})/\|m_{start,j} - m_{goal,j}\|$
            $a = m_{c,j} - m_{goal,j}, r_s = \|a\|$
            $\theta_s = \mathrm{arcos}(s \cdot a/r_s)$
            $m_{rand,j} \leftarrow$ForwardSector($m_{goal,j}, r_s, \theta_s$)
        else
            $m_{centor} = \sum_{i=1}^{N} m_i/N$
            $r_b = \|m_{centor} - m_f\|$
            $m_{rand,j} \leftarrow$BackwardBall($m_{centor}, r_b$)
        end if
    end if
end for
```

inherent control constraints of the mobile base. According to the kinematic model of the mobile base [23], each control output u_k corresponds to an expected state M_k, which is determined by

$$m_{check,j} = \underset{m_{k,j}}{\mathrm{argminCost}} \left(m_{k,j}, m_{rand,j}\right) \tag{19}$$

$$\mathrm{Cost}(m_1, m_2) = \omega_1 \|m_1 - m_2\| + \omega_2 \mathrm{arccos} \frac{\overrightarrow{v_1} \cdot \overrightarrow{m_1 m_2}}{\|\overrightarrow{v_1}\| \|\overrightarrow{m_1 m_2}\|} \tag{20}$$

where $m_{check,j}$ is a temporary control checking node associated with $\mathscr{M}_j$ in the new node M_{new}. m_1 and m_2 are the two arbitrary node, v_1 is the unit vector with angle of m_1. This cost function evaluates the distance and smoothness between two nodes. Hence, by using (19) and (20), M_{new} and the optimal control output u_{new} is obtained. One should note that $\mathcal{U}(u_k, \epsilon)$ is assigned as null if M_{parent}.GoalArrived[j] $= 1$ to ensure the mobile base is fixed once it reaches goal points.

In line 11-20 of Algorithm 1, the Heuristic method is proposed to optimize the path by using neighbor vertices. After a feasible new node M_{new} is generated. The neighbor vertices of M_{new}, $\mathcal{M}_{near}$ are obtained by using a given search radius r_c. Each $m_{near,j}$ directly connected to $m_{new,j}$ and the $m_{near,j}$ with minimum heuristic cost remained, in which $h(M_{new})$ indicates the total path length from M_{init} to $h(M_{new})$. However, setting $m_{near,j}$ as the parent node could change the structure of the tree causing vertices disjunction. In this work, we propose ReSetParent() to reset the passed vertices on the edge ($m_{near,j}, m_{new,j}$), which is not change the parent node and thereby ensure the stability of the tree structure.

The proposed heuristic method in this study differs from the conventional neighborhood optimization method used in the RRT* algorithm in two key aspects. Firstly, the RRT* algorithm typically utilizes path length as its cost function without considering the feasibility of robot control, but the MCDPR is significantly more flexible and incorporates multiple constraints. As a result, the cost function in this study was enhanced, and control feasibility was taken into account. Secondly, the rewiring process in RRT* has the potential to change the parent node and thus alter the timestamp. To ensure the stability of the tree structure, the ReSetParent() function was introduced. This function enables us to reset the passed vertices on the edge, preventing changes to the parent node while maintaining the integrity of the tree structure. By implementing this additional step, we can effectively address the issue of changing timestamps that may arise during the rewiring process.

Figure 7 illustrates the proposed method, we use two subtrees to represent two mobile bases for simplicity. As shown in Fig. 7a, the neighbor vertices of $\boldsymbol{m}_{8,1}$ and $\boldsymbol{m}_{8,2}$ is obtained to optimize path. Thus, two path ($\boldsymbol{m}_{3,1}$,$\boldsymbol{m}_{8,1}$) and ($\boldsymbol{m}_{3,2}$,$\boldsymbol{m}_{8,2}$) with minimum cost can be found. The parent vertices $\boldsymbol{m}_{7,1}$, $\boldsymbol{m}_{7,2}$ is reset and check in Fig. 7b by using $\boldsymbol{m}_{reset,j} = \boldsymbol{m}_{reset,j}.parent + \eta(\boldsymbol{m}_{reset,j}.child - \boldsymbol{m}_{reset,j}.parent)$, in which η is initialized to $1/2$. However, if $\boldsymbol{m}_{reset,j}$ is not feasible, η is a random number ranging from 0 to 1. Moreover, due to the adaptive sampling method, the tree tends to grow toward the goal. As shown in Fig. 7c, the first tree has arrived but the second tree is not. In this case, the next sampling will be expanded on the remained vertices $\boldsymbol{m}_{10,2}$ on the second tree. Finally, the algorithm is terminated when each tree has reached its target point shown in Fig. 7d.

Generation of the end-effector optimal path

After the RRT tree $\mathcal{T}$ is obtained from Algorithm 1, the path for mobile bases $\boldsymbol{M}_{path}$ can be generated by consecutively finding $\boldsymbol{M}_{parent}$ from the root of the tree. In this work, a kinematic stability index γ is used to determine the position of the end-effector, which is expressed as follows

$$\gamma = \gamma_k + \gamma_s \in [0, 2] \tag{21}$$

Based on the kinematic stability index γ, a grid-based search method is developed in Algorithm 3 to generate the path for the end-effector $\boldsymbol{P}_{path,p}$, $p = 1, ..., N$. The proposed algorithm traverses each state on the path of mobile bases $\boldsymbol{M}_{path,p}$ that corresponds to a specific CDPR configuration to maximize γ. For each configuration, we first determine a search area $\mathcal{S}_p$ with radius r_s, and the center of the $\mathcal{S}_p$ depends on the last end-effector position $\boldsymbol{P}_{path,n-1}$. Thus, $\mathcal{S}_p$ is different in each $\boldsymbol{M}_p$, which ensures smoothness and continuity of the path. For each $\mathcal{S}_p$, we discrete $\mathcal{S}_p$ to obtain Q_p cells $\{c_1, ..., c_Q\}$, and traverse each cell find the largest γ. It should be noted that some cells may not be feasible due to the cable/end-effector colliding with obstacles, which is considered in line 6 of Algorithm 3.

The path obtained through the end-effector is optimal due to the use of the grid-based search method and a kinematic stability index. Specifically, the kinematic stability index γ is used to determine the position of the end-effector, and a grid-based search method is developed to generate the path for the end-effector by maximizing γ at each configuration along the path of the mobile bases. By maximizing γ, the algorithm ensures that the end-effector is positioned in a kinematically stable configuration at each point along the path. Additionally, the use of the grid-based search method ensures that the path is smooth and continuous. Together, these factors suggest that the path obtained through the end-effector is optimal in terms of kinematic stability and smoothness/continuity.

Algorithm 3 Grid-based search method

```
Input: ${}^{p}\boldsymbol{M}_{path}$, $Q$, $c$
Output: $\boldsymbol{P}_{path,p}$
1: $\gamma_{best} = 2$
2: for $n = 1$ to $N$ do
3:   Determine the search area $\mathcal{S}_p$ with radius $r_s$
4:   Discrete $\mathcal{S}_p$ to obtain $Q_p$ cells $\{c_1, ..., c_Q\}$
5:   for $n = 1$ to $Q$ do
6:     if not feasible $(c_n)$ then
7:       Continue
8:     else
9:       Calculate $\gamma$ by using (21)
10:       if $\gamma \geq \gamma_{best}$ then
11:         $\boldsymbol{P}_{path,n} \leftarrow c_n$
12:         $\gamma_{best} \leftarrow \gamma$
13:       end if
14:     end if
15:     $\boldsymbol{P}_{path,p} \leftarrow \boldsymbol{P}_{path,p} \cup \boldsymbol{P}_{path,n}$
16:   end for
17: end for
```

Method validation

Initialization set-up and results

In this section, we conduct a number of simulations to demonstrate the effectiveness and efficiency of the proposed algorithm. Figure 8 illustrates the cluttered environment with the dynamic model of our MCDPR prototype developed using CoppeliaSim (formerly known as V-REP) robot simulator software [24]. The cluttered environment is composed of ten obstacles and these obstacles are modeled as cylinders

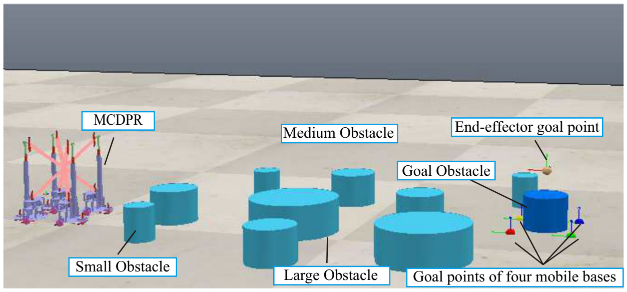


Fig. 8 Simulation environment developed using CoppeliaSim software [24]

Table 1 Simulation parameter

Parameters	Value	Parameters	Value
L_m	0.38 m	l_{min}	0.5 m
r_m	0.2 m	l_{max}	4 m
β_{max}	65 $\circ$	r_c	0.5 m
L_c	0.05 m	$r_{gc,j}$	0.3 m
$r*$	0.5 m	η	0.1 m
ω_1	0.5	ω_2	0.5

with three different radii, 0.15 m, 0.25 m, and 0.45 m. These obstacles have an identical height 0.4 m, and the position of obstacles is randomly assigned. The five goal points for the MCDPR are given around the dark blue obstacle that is denoted as a target obstacle in which the robot can perform the task on it, i.e., picking and/or releasing operations.

The proposed method is initialized using parameters given in Table 1. For the consecutive state transition from $\boldsymbol{X}_k$ to $\boldsymbol{X}_{k+1}$, ΔT should be relatively small to obtain high accuracy, thereby $\Delta T = 0.2$ is used in this work. For each ${}^{p}\boldsymbol{M}_{path}$, it could determine the coordinates of a set of pulleys by using $h_1 = 0.285$ m and $h_2 = 0.926$ m. Accordingly, the mass center of the mobile base can be directly obtained from the MCDPR dynamic model in CoppeliaSim given by $[0.12, 0.34, 0.36]^T$ with respect to O_{pj} coordinate system. Hence, γ_k and γ_s can be computed in each ${}^{p}\boldsymbol{M}_{path}$. The simulation result is illustrated in Fig. 9, it is found that a feasible path is generated for each mobile base to reach their goal points and the path of the end-effector is determined by maximizing γ. Due to the limitation of the length of the cable making MCDPR impossible to separate through large obstacles, one should note that the MCDPR adaptively performs integration mode and separation modes in Fig. 9.

To simplify the problem, the end-effector is modeled as a point to perform collision detection. Due to the end-effector was restricted to only perform the translational motion in x, y, and z direction, and the rotation of the end-effector is not considered. The x and y motion of the end-effector were depicted in Fig. 9 and the z-coordinate of the end-effector was shown in Fig. 10. The MCDPR is a type of parallel robot for which the inverse kinematics, as expressed in equations (1) and (10), can be readily determined. As a result, when given a set of three-dimensional coordinates for the end-effector, it is possible to calculate the corresponding cable length and cable speed using these equations. This enables precise control of the end-effector's translational motion through the manipulation of cable length and speed.

The three minimum distances about mobile bases/mobile bases, mobile bases/obstacles, and cables/obstacles is illustrated in Fig. 11, it can be found that the distance constraints of (2), (3), and (4) can be satisfied. Moreover, the change of four interior angles is shown in Fig. 12, it can be seen that $\theta_1, \theta_2, \theta_3, \theta_4 \in [0, \pi]$, thereby the MCDPR configuration constraints defined by (4) is thus fulfilled.

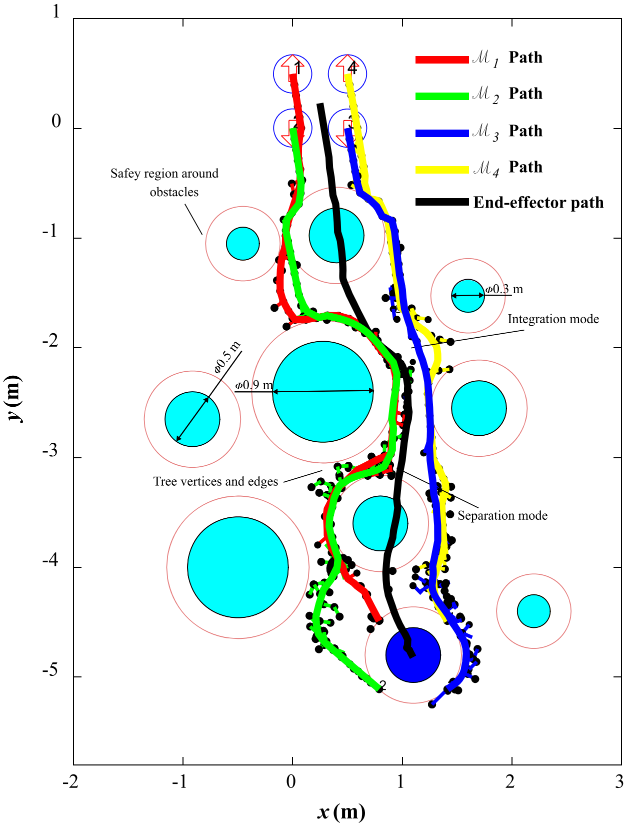


Fig. 9 Resulted path in a constrainted environment

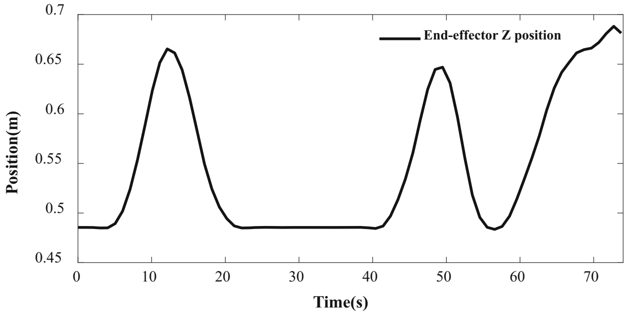


Fig. 10 z-coordinate of the end-effector

MCDPR Verification in CoppeliaSim and experiments

In this section, the resulting MCDPR trajectories obtained from the last section are simulated in CoppeliaSim environment to verify the proposed method and the corresponding experiment is carried out. The software simulation framework is depicted in Fig. 14. For controlling the mobile bases, the continuous velocity profiles of the mobile bases are converted into the rotational velocities of the mobile base's

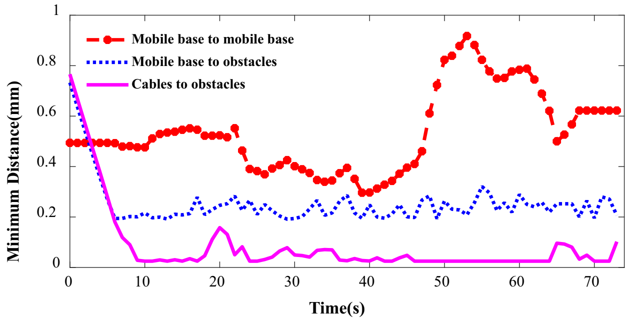


Fig. 11 Relative distance change with time

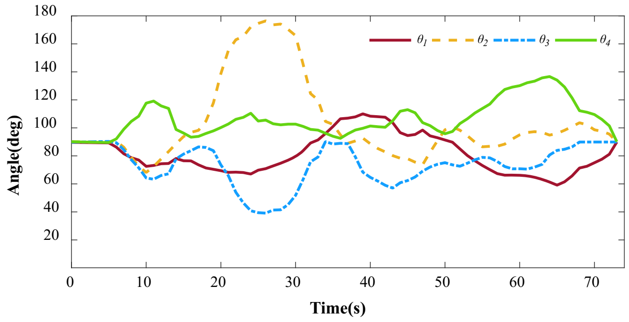


Fig. 12 Interior angles change with time

wheels by using its kinematic model. As shown in Fig. 13, the continuous velocity profiles of the mobile bases and the end-effector are guaranteed due to the post-processing. One should note that these mobile bases do not arrive at the goal points at the same time. Moreover, for controlling the end-effector, we utilize (1) and (10) to adjust the length and speed of cables shown in Fig. 15.

The obtained wheels' rotational velocities are sent to the revolute joints associated with wheels to control the mobile bases in CoppeliaSim. Additionally, the length of the cable can be adjusted by prismatic joints associated with cables. Hence, the MCDPR model is actuated by these joints. The state of MCDPR can be directly obtained from the CoppeliaSim during the motion, it will be compared with the desired path to obtain the error.

The clips of the simulation in CoppeliaSim are shown in Fig. 17, it can be seen that the MCDPR can follow the desired paths and avoid the collision. Moreover, Fig. 16 illustrates the error between the actual path of the MCDPR from the simulation and the desired path, the maximum error about the mobile base and the end-effector is approximately 3.8 cm. Thus, the feasibility and stability of the proposed path planning method can be verified.

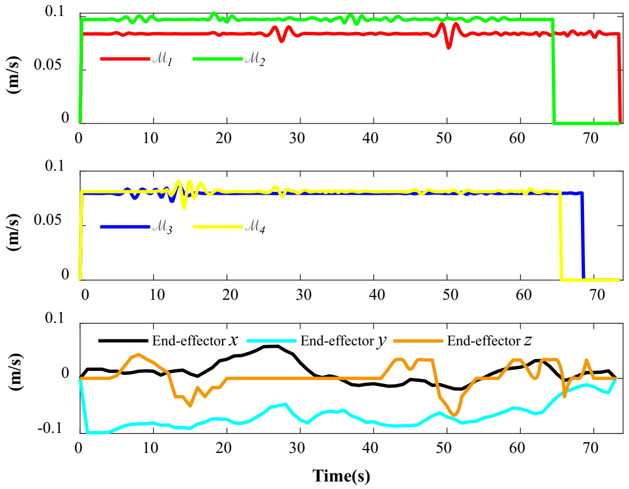


Fig. 13 Velocity profiles of mobile bases

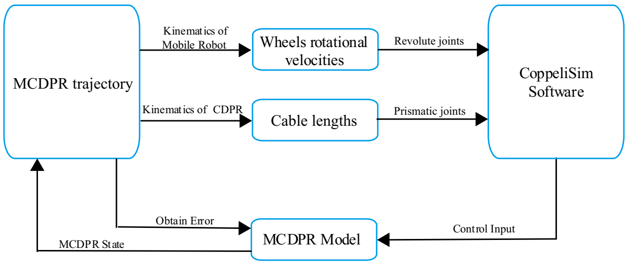


Fig. 14 Software simulation framework

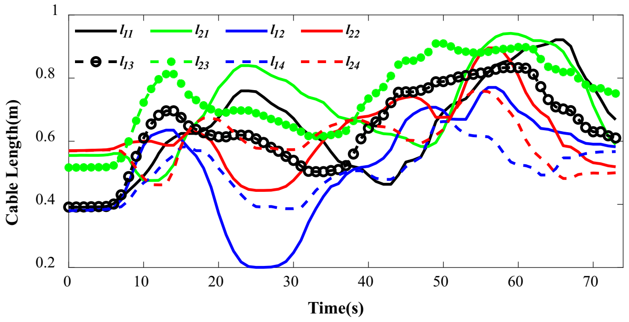


Fig. 15 Cable length change

The proposed approach is validated experimentally on our self-built MCDPR, a scenario is designed as the MCDPR pass through the first obstacle shown in Fig. 9. The NVIDIA Jetson Nano$^T M$ Developer Kit is adopted as a control computer to operate our MCDPR system, in which the designed cable length and mobile base velocities are sent to control MCDPR. The experimental results are shown in Fig. 18, it can be seen that the MCDPR is smooth and stable passing the obstacle and the total time is 24 s. Before the experiment, the cable was pre-tensed to 20 N, and the cable tension change measured from the load cell sensor during the experiment is shown in Fig. 19. It can be seen that the cable tension remains positive in the experiment and has a continuous profile. These results verify that the proposed method is feasible and effective.

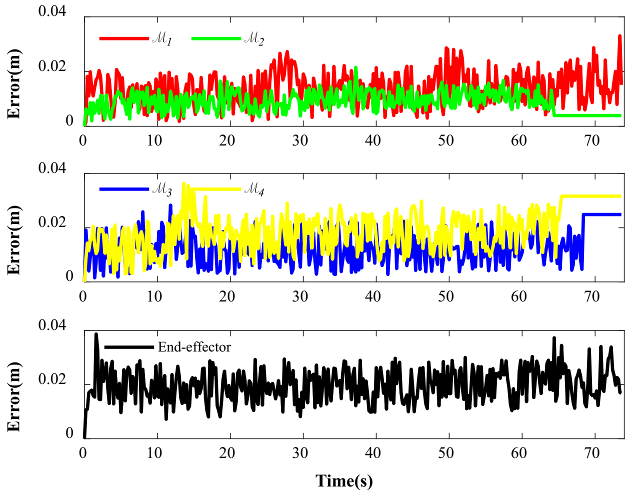


Fig. 16 Error between the actual and the desired position

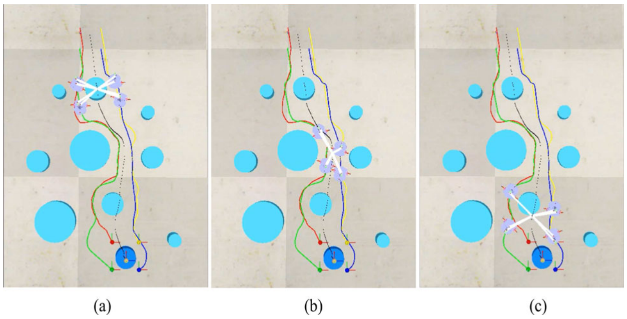
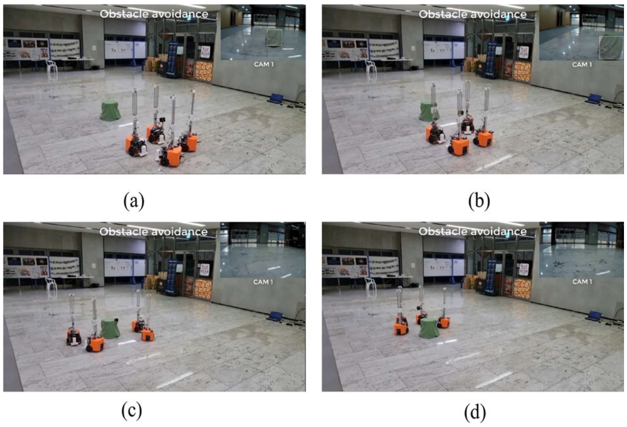

(a) (b) (c)

Fig. 17 Simulation in CoppeliaSim. **a** $t = 21$ s **b** $t = 49$ s **c** $t = 61$ s



(a) (b)

(c) (d)

Fig. 18 Experimental results. **a** $t = 6$ s **b** $t = 12$ s **a** $t = 18$ s **b** $t = 24$ s

Batch evaluation

In this section, the proposed path planning method for MCDPRs is compared with other existing methods. Due to the stochastic nature of the proposed method, a batch evaluation of 1000 simulations is performed and the proposed method is evaluated using the average value. All the simulations are

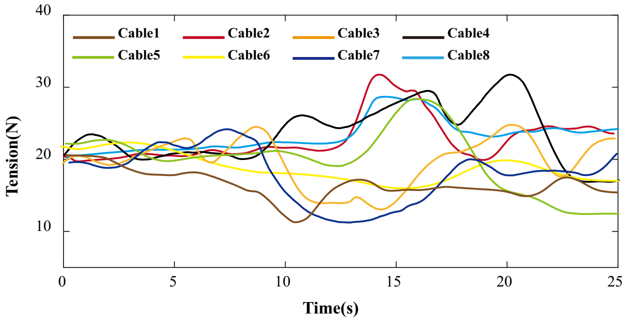


Fig. 19 Cable tension change in experiment

performed using ©MATLAB with CPU computations on an Intel ®i7-9750 CPU@2.60GHz and 32GB RAM Windows 10 system. The identical initialization parameters defined in the previous simulations are used in these simulations.

The results of the batch evaluation are summarized in Table 2. Comparing the proposed method with the Iterative Algorithm and Direct Transcription, the proposed method reduced the CPU time by 88.7% and 79.8%, respectively. This is due to the developed adaptive sampling strategy leading the tree to grow towards the goal points. Iterative Algorithm and Direct Transcription require complex iterative processes for optimization which may become computationally more expensive with the increase of obstacles in the environment. However, the proposed method does not require an iterative search but constructs the feasible tree by adaptively exploring the state space. It should be noted that the Iterative algorithm and the Direct Transcription method resulted in the mobile base and the end-effector reaching the target point simultaneously. In our method, the growth of the subtree is terminated when it reaches the goal point. Furthermore, it can be observed that the average global performance index γ of the total path is improved by 56.12% and 34.21% with respect to the Iterative Algorithm and Direct Transcription. Therefore, the proposed path planning method can contribute to better stability and kinematic performance of MCDPRs.

The proposed algorithm incorporates a heuristic optimization approach to enhance the efficiency of the MCDPR path. The results demonstrate a 46.81% decrease in path length compared to the Iterative Algorithm. Although the Direct Transcription method outperforms the proposed algorithm in terms of path length due to its broader optimization radius and iterative process, the proposed method only differs by 4.51% from Direct Transcription. Moreover, the proposed algorithm offers several advantages, including faster CPU time and improved average γ, which offset the minor difference in path length. These results highlight the ability of the proposed method to optimize path, enhance computational efficiency, and ensure kinematics stability simultaneously.

Table 2 The evaluation result

Method	CPU Time	Path Length	Average γ
Iterative Algorithm [20]	120.94 min	11.62 m	0.98
Direct Transcription [21]	67.81 min	5.91 m	1.14
Proposed method	**13.63** min	**6.18** m	**1.53**

The results of the proposed method are in bold to highlight the effectiveness

Conclusion and future work

In this work, an optimal path planning method is proposed for MCDPRs to generate feasible paths in constrainted environments with considering kinematic stability. The proposed method utilizes a heuristic approach to optimize the path, and the adaptive sampling method is developed to enhance efficiency. Hence, the various constraints of MCDPR can be considered in detail and the kinematic stability is maximized by using designed performance indicators. The proposed method was validated by simulations and subject to real-world testing on a self-built MCDPR prototype. The results show that the proposed approach can generate feasible paths for mobile bases and the end-effector in a shorter time. Compared to other existing methods, the proposed method provides better kinematic stability performance and efficiency on the premise of minimizing the path length. Therefore, the reliability of the proposed method was verified.

One should note that the proposed method is developed in a static environment in which the position of the obstacle is known. Moreover, the proposed method can be extended to a variety of multi-agent systems, such as UAVs. However, it is imperative for individuals to tailor the algorithm to the specific system by adjusting the cost function and constraint equations as necessary. This enables the algorithm to effectively accommodate the distinct characteristics and demands of the system. In future work, it is desirable to develop the real-time path planning method for MCDPR in a dynamic environment with the moving obstacle. To fulfill this purpose, a continuous localization system for MCDPR is also required to be designed in the future.

Acknowledgements This study was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government. (MSIT) (2021R1A2C2013053).

Data availability The data that support the findings of this study are available upon request from the corresponding author. Restrictions apply to the availability of these data, which were used under license for the current study, and so are not publicly available. Data are however available from the authors upon reasonable request and with permission of Kyoung-Su Park (pks6348@gachon.ac.kr).

Declarations

Conflict of interest There is no conflict of interest.

References

1. Zhang Z, Shao Z, Wang L, Shih AJ (2018) Optimal design of a high-speed pick-and-place cable-driven parallel robot. In: cable-driven parallel robots: proceedings of the third international conference on cable-driven parallel pobots, Springer International Publishing, pp 340–352
2. Chawla I, Pathak P, Notash L, Samantaray A, Li Q, Sharma U (2021) Workspace analysis and design of large-scale cable-driven printing robot considering cable mass and mobile platform orientation. Mechanism and machine theory 165:104426
3. Piao J, Jin X, Choi E, Park JO, Kim CS, Jung J (2018) A polymer cable creep modeling for a cable-driven parallel robot in a heavy payload application. In: Cable-Driven Parallel Robots: Proceedings of the Third International Conference on Cable-Driven Parallel Robots, Springer International Publishing, pp 62–72
4. Winter DL, Ament C (2021) Development of Safety Concepts for Cable-Driven Parallel Robots. In: Cable-driven parallel robots: Proceedings of the 5th International Conference on Cable-Driven Parallel Robots, Springer International Publishing, Cham, pp 360–371
5. Bury D, Izard J-B, Gouttefarde M, Lamiraux F (2019) "Continuous collision detection for a robotic arm mounted on a cable-driven parallel robot," arXiv:1909.10857
6. Lesellier M, Gouttefarde M (2019)"A bounding volume of the cable span for fast collision avoidance verification," In: International Conference on Cable-Driven Parallel Robots. Springer, pp. 173–183
7. Pedemonte N, Rasheed T, Marquez-Gamez D, Long P, Hocquard É, Babin F, Fouché C, Caverot G, Girin A, Caro S (2020) Fastkit: a mobile cable-driven parallel robot for logistics. Advances in robotics research: from lab to market. Springer, New York, pp 141–163
8. Gagliardini L, Gouttefarde M, Caro S (2018) Design of reconfigurable cable-driven parallel robots". Mechatronics for cultural heritage and civil engineering. Springer, New York, pp 85–113
9. Rasheed T, Long P, Marquez-Gamez D, Caro S (2018) Tension distribution algorithm for planar mobile cable-driven parallel robots. In: Cable-driven parallel robots: Proceedings of the Third Inter-

national Conference on Cable-Driven Parallel Robots, Springer International Publishing, pp 268–279

10. Xiang S, Gao H, Liu Z, Gosselin C (2020) Dynamic point-to-point trajectory planning for three degrees-of-freedom cable-suspended parallel robots using rapidly exploring random tree search. J Mech Robot 12:4
11. Idà E, Bruckmann T, Carricato M (2019) Rest-to-rest trajectory planning for underactuated cable-driven parallel robots. IEEE Trans Robot 35(6):1338–1351
12. Lin Y, Saripalli S (2017) Sampling-based path planning for uav collision avoidance. IEEE Trans Intell Transp Syst 18(11):3179–3192
13. Jaillet L, Cortés J, Siméon T (2010) Sampling-based path planning on configuration-space costmaps. IEEE Trans Robot 26(4):635–646
14. LaValle Steven M (1998) Rapidly-exploring random trees: A new tool for path planning 98–11
15. Karaman S, Frazzoli E (2011) Sampling-based algorithms for optimal motion planning. Int J Robot Res 30(7):846–894
16. Gammell JD, Srinivasa SS, Barfoot TD (2014)"Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, pp. 2997–3004
17. Bak J-H, Hwang SW, Yoon J, Park JH, Park J-O (2019) Collision-free path planning of cable-driven parallel robots in cluttered environments. Intell Service Robot 12(3):243–253
18. Zhang B, Shang W, Cong S (2018) "Optimal rrt* planning and synchronous control of cable-driven parallel robots," In: 2018 3rd International Conference on Advanced Robotics and Mechatronics (ICARM). IEEE, pp. 95–100
19. Mishra UA, Mishra U, Métillon M, Caro S (2021) *et al.*, "Kinematic stability based afg-rrt* path planning for cable-driven parallel robots," In: The 2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
20. Rasheed T, Long P, Marquez-Gamez D, Caro S (2019) "Path planning of a mobile cable-driven parallel robot in a constrained environment," In: International Conference on Cable-Driven Parallel Robots. Springer. pp 257–268
21. Rasheed T, Long P, Roos AS, Caro S (2019) "Optimization based trajectory planning of mobile cable-driven parallel robots," In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 6788–6793
22. Kajita S, Kanehiro F, Kaneko K, Fujiwara K, Harada K, Yokoi K, Hirukawa H (2003) "Biped walking pattern generation by using preview control of zero-moment point," In: 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422). vol. 2. IEEE, pp. 1620–1626
23. Malu SK, Majumdar J (2014) Kinematics, localization and control of differential drive mobile robot. Glob J Res Eng 14(H1):1–7
24. Rohmer E, Singh SP, Freese M (2013) "V-rep: A versatile and scalable robot simulation framework," In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, pp. 1321–1326