



HAL
open science

Path Planning of a Mobile Cable-Driven Parallel Robot in a Constrained Environment

Tahir Rasheed, Philip Long, David Marquez-Gamez, Stéphane Caro

► **To cite this version:**

Tahir Rasheed, Philip Long, David Marquez-Gamez, Stéphane Caro. Path Planning of a Mobile Cable-Driven Parallel Robot in a Constrained Environment. In: Pott A., Bruckmann T. (eds) Cable-Driven Parallel Robots. CableCon 2019. Mechanisms and Machine Science, vol 74. Springer, Cham, pp.257-268, 2019, 10.1007/978-3-030-20751-9_22 . hal-02404441

HAL Id: hal-02404441

<https://hal.science/hal-02404441>

Submitted on 11 Dec 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Path Planning of a Mobile Cable-Driven Parallel Robot in a Constrained Environment

Tahir Rasheed¹, Philip Long², David Marquez-Gamez³, and Stéphane Caro⁴

¹ École Centrale de Nantes, Laboratoire des Sciences du Numérique de Nantes, UMR CNRS 6004, 1, rue de la Noë, 44321 Nantes, France,

Tahir.Rasheed@ls2n.fr,

² RIVeR Lab, Department of electrical and computing engineering, Northeastern University, USA p.long@northeastern.edu

³ IRT Jules Verne, Chemin du Chaffault, 44340, Bouguenais, France,

david.marquez-gamez@irt-jules-verne.fr

⁴ CNRS, Laboratoire des Sciences du Numérique de Nantes, UMR CNRS 6004, 1, rue de la Noë, 44321 Nantes, France, stephane.caro@ls2n.fr

Abstract. A Mobile Cable-Driven Parallel Robot (MCDPR) is a special type of Reconfigurable Cable-Driven Parallel Robot (RCDPR) composed of a classical Cable-Driven Parallel Robot (CDPR) mounted on multiple mobile bases. The additional mobility of the mobile bases allows such systems to autonomously modify their geometric architecture, and thus make them suitable for multiple manipulative tasks in constrained environments. Moreover, these additional mobilities make MCDPRs kinematically redundant. Therefore, the subject of this paper is to introduce a two stage path planning algorithm for MCDPRs. The first stage searches for a feasible and collision free path of mobile bases. The second stage deals with generating an optimal path of the moving-platform to displace it from an initial to a desired pose. The proposed algorithm is validated through simulation on a three degree-of-freedom (DoF) point mass moving-platform displaced by four cables with each cable carried by an independent mobile base.

Keywords: Mobile Cable-Driven Parallel Robot, Reconfigurability, Kinematic Redundancy, Path Planning, Wrench Analysis

1 Introduction

Cable-Driven Parallel Robots (CDPRs) are a particular class of parallel robots whose actuated limbs are cables, connecting the moving-platform with a fixed base frame. The platform is operated by changing the cable lengths between the moving-platform and the base. CDPRs are interesting for various applications, such as construction [1], high speed tasks [2], operations over large workspace [3] and as haptic devices [4].

In spite of numerous applications of CDPRs, several challenges are still unresolved. For example, CDPRs require free circulation of the cables without any interference with the environment [19]. Furthermore, CDPRs have fixed cable layout, *i.e.*, fixed exit points and cable configuration, which must be chosen carefully in order to maximize its workspace. Therefore, it is reasonable to alter robot's geometric structure based on

its environment and the required task. CDPRs with the ability to change their geometric architecture are known as Reconfigurable Cable-Driven Parallel Robots (RCDPRs). A recent study on RCDPRs [10–12] proposes different approaches for maximizing the robot workspace and increasing platform stiffness. However, for most existing RCDPRs, reconfiguration is usually discrete and performed manually.

To achieve autonomous reconfigurability of RCDPRs, a novel concept of Mobile Cable-Driven Parallel Robots (MCDPRs) has been introduced in [5] which uses a unique combination of CDPR and mobile bases. The first MCDPR prototype, FASTKIT, was designed and built in the framework of ECHORD++ project⁵ for logistic applications. Recently, another MCDPR prototype named MoPICK⁶ has been developed composed of a three DoF point mass moving-platform pulled by four cables mounted on four mobile bases, as shown in Fig. 1(a). The targeted applications of MoPICK are mobile tasks in a constrained environment, for example, a workshop or logistic operations in a warehouse.

Generally MCDPRs are kinematically redundant due to the additional mobility of the mobile bases [7, 8]. As a consequence, there exist multiple paths for the MCDPR mobile bases to displace the moving-platform from one pose to another and to perform a desired task. [7] addresses the problem of determining an optimal path also referred to as a redundancy planning of FASTKIT with one degree of kinematic redundancy. On the contrary, MoPICK has eight degrees of kinematic redundancy. In order to perform the desired tasks in a potentially cluttered environments, safe and collision free trajectories are required for the mobile bases and the moving-platform of MoPICK.

*A** and *Dijkstra* are search algorithms which can find globally optimal paths in a discretized workspace. Yet, such systematic algorithms scale poorly to higher dimensions [20]. In contrast by exploiting randomization, a feasible path can be obtained quickly, for instance, using sampling based motion planning algorithms [21]. However, the resulting paths are often of poor quality and thus the methods require larger computation time to find reasonable solutions in complex environments. A common method to reduce the complexity is by obstacle inflation [18] with the objective of reducing the planning problem to a moving point in a 2D environment. Finally, an additional problem with these algorithms is the difficulty of incorporating the complex kinodynamic constraints associated with CDPRs.

In this paper, we present a sampling based iterative path planning algorithm for MoPICK by optimizing the wrench capabilities of its moving-platform. The proposed algorithm searches for a feasible and continuous path of its mobile bases between the initial and desired pose of the moving-platform by making a locally optimal choice at each step. Thus the system’s kinodynamic constraints can be enforced at each instant of the trajectory. The proposed algorithm decomposes the problem in two parts. The first part aims to find a feasible, continuous and collision free path for the mobile bases. A path is generated using an iterative procedure by generating a sequence of straight line paths. The paths are then smoothed using B-Splines. The second part of the algorithm takes as input the smoothed B-Splines and locally optimizes the moving-platform’s wrench capability.

⁵FASTKIT Videos: <https://www.youtube.com/channel/UCJ8QRs818MBc8YSbn-bZVjA>

⁶Demonstration of MoPICK: https://youtu.be/_zfqtNsrpHI

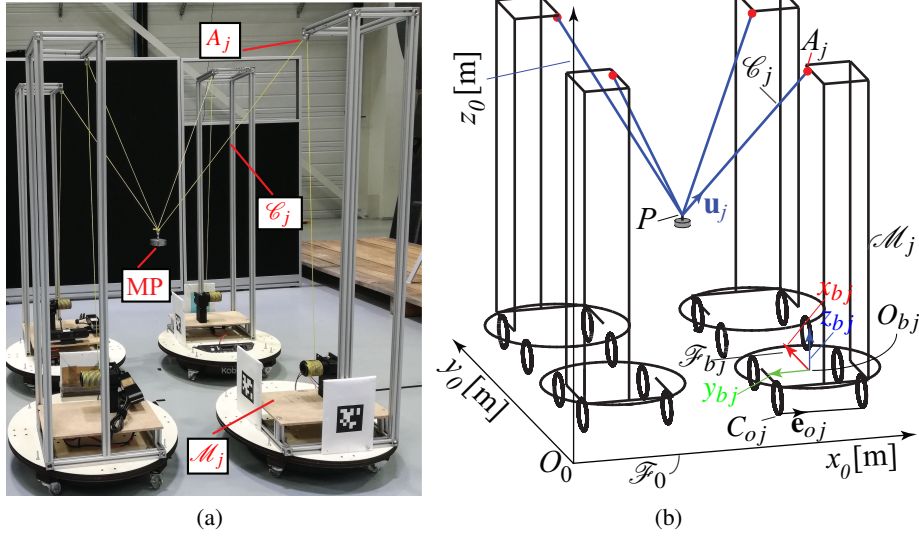


Fig. 1. (a) MoPICK prototype and its (b) Parameterization

The paper is organized as follows. Section 2 deals with the description and parameterization of MoPICK. Section 3 details the Static Equilibrium (SE) conditions of MCDPR under study that is used to determine its wrench feasible poses. Section 4 deals with the formulation of the task which is to be performed by MoPICK. Section 5 proposes a path planning algorithm for MoPICK to displace its moving-platform from the initial pose to a desired pose while respecting its SE conditions. Section 6 discusses the results acquired from the path planning algorithm for the required task. Finally, conclusions are drawn and future work is presented in Section 7.

2 Manipulator Description and Parameterization

The MoPICK prototype consists of four mobile bases ($p = 4$) that carry the exit points of the CDPR itself composed of four cables ($q = 4$) and a three degree-of-freedom (DoF) point mass moving-platform ($n = 3$) as shown in Fig. 1(a). The j th mobile base, denoted as \mathcal{M}_j , $j = 1, \dots, 4$, is carrying a single cable named as \mathcal{C}_j . Let \mathbf{u}_j be the directional vector of \mathcal{C}_j pointing from the the point-mass P to the cable exit point A_j . Let \mathbf{t}_j be the \mathcal{C}_j 's cable tension vector expressed as,

$$\mathbf{t}_j = t_j \mathbf{u}_j, \quad (1)$$

where t_j denotes the tension in the cable \mathcal{C}_j .

Each mobile base has a cylindrical shape of radius r_{mb} equal to 0.25 m. The support structure for the cables is composed of a 1.2 m high aluminum frame mounted on a cylindrical base. Let \mathcal{F}_0 be the base frame of origin O_0 and axes x_0 , y_0 and z_0 . Let \mathcal{F}_{bj} be the frame attached to \mathcal{M}_j of origin O_{bj} and axes x_{bj} , y_{bj} and z_{bj} , respectively.

Each mobile base is composed of four wheels, with two motorized wheels and two supporting wheels at the front and rear. It uses a differential drive mechanism through the motorized wheels to move, thus is subjected to non-holonomic constraints [17], *i.e.*, it can only generate a translational motion along x_{bj} and a rotational motion about z_{bj} . Straight line motion is achieved by turning the motorized wheels at the same rate in the same direction, however, pure rotational motion is accomplished by turning the motorized wheels at the same rate in the opposite direction. The exit point A_j lies on the axis z_{bj} , thus, the rotational motion of \mathcal{M}_j does not alter \mathbf{u}_j . As a consequence, for a given position of P , \mathbf{u}_j can be determined by localizing the reference point O_{bj} of \mathcal{M}_j in x_0y_0 plane (see Fig. 1(b)).

3 Wrench Feasibility

In this section the Wrench Feasible Workspace (WFW) for MoPICK is studied. It is defined as the set of platform poses for which the required set of wrenches can be balanced with wrenches generated by the cables while keeping the MCDPR in Static Equilibrium (SE) [6]. A MCDPR is in the state of equilibrium if and only if its mobile bases and moving-platform are all in SE. Therefore, first the SE conditions of the moving-platform and the mobile bases of MoPICK are formulated. From this, the wrench capabilities of the moving-platform can be studied.

3.1 Static Equilibrium of the MCDPR

The SE of the moving-platform is expressed as [9]

$$\mathbf{W}\mathbf{t} = \mathbf{f}, \quad (2)$$

where \mathbf{W} is a $(n \times m)$ wrench matrix mapping the cable tension vector $\mathbf{t} \in \mathbb{R}^m$ onto the wrenches $\mathbf{f} \in \mathbb{R}^n$ applied by the cables onto the moving-platform.

$$\mathbf{W} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \mathbf{u}_3 \ \mathbf{u}_4], \quad \mathbf{f} = \begin{bmatrix} f^x \\ f^y \\ f^z \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{bmatrix}. \quad (3)$$

The cable tensions are all bounded between a minimum tension \underline{t}_j and a maximum tension \bar{t}_j

$$\underline{t}_j \leq t_j \leq \bar{t}_j, \quad j = \{1, \dots, 4\}, \quad (4)$$

The SE of a wheeled mobile base is characterized by its tipping conditions, which depend on the moments generated at the boundaries of the mobile base footprint. The footprint of \mathcal{M}_j is formed by joining the wheel contact points, denoted by C_{oj} , $o = 1, \dots, 4$ in anticlockwise direction. The directional vector of the footprint boundary between the two consecutive contact points (C_{oj}, C_{o+1j}) is denoted as $\mathbf{e}_{C_{oj}}$ as shown in Fig. 1(b). Let $m_{C_{oj}}$ be the moment generated about the footprint boundary between C_{oj}

An index called Capacity Margin [15, 16] determines if a given pose is wrench feasible using the facets of the AWS and the vertices of the Required Wrench Set (RWS). It is a measure of the robustness of the equilibrium of the robot, denoted as μ , namely,

$$\mu = \min (\min s_{d,l}), \quad (8)$$

where $s_{d,l}$ is the signed distance from d th vertex of the RWS to the l th facet of the AWS. μ is positive as long as all the vertices of RWS are inscribed by \mathcal{A} , *i.e.* RWS can be generated by the cables while respecting all the SE conditions of a MCDPR.

4 Task Formulation

The task is defined as displacing the moving-platform from an initial position P_1 to a desired position P_f while ensuring that the moving-platform passing through a set of way-points. The task is to be performed in a constrained environment having numerous tables and obstacles in it. The way-points on the tables require a task action, for example grasping and/or releasing an object. Some intermediate way-points are placed in order to guide the system to navigate between the two consecutive task locations. These intermediate way-points are interpolated between the two task locations while maximizing the distance from the nearest obstacle.

Let l be the total number of way-points. Each way-point is denoted as $P_i, i = 1, \dots, l$. The Cartesian coordinate vector of the i th way point is denoted as \mathbf{p}_i . The obstacles are defined as cylinders. As discussed earlier, the mobile bases are also modeled as cylinders with radius r_{mb} . A common technique [18] used to ensure no collision between the obstacles and the mobile bases is to inflate the obstacles by at least r_{mb} in x_0y_0 plane, denoted as ‘safe region around obstacles’, as depicted in Fig. 2. As a consequence, the mobile bases are treated as a point.

In order to perform the desired task, a feasible and collision free path of the mobile bases is required. Accordingly, the path of the moving-platform is also required to displace it from P_1 to P_f by sequentially following the intermediate way-points. Therefore, Section 5 addresses the aforementioned problem and presents a path planning algorithm for MCDPR under study.

5 Path planning Algorithm

This section presents a Path Planning algorithm for MoPICK. In order to find a path between i th and $(i+1)$ th way-points, the following steps are taken. First, let’s assume that a feasible solution has been obtained for the mobile bases with moving-platform located at P_i . The wrench capability of the system is calculated when the moving-platform is located at P_{i+1} while the mobile bases are still located at the solution of *previous* way-point (P_i). The mobile bases are then iteratively displaced from P_i to P_{i+1} such that each displacement maximizes the wrench capability while avoiding collisions. This results in a feasible path for the mobile bases. Given this feasible path for the mobile bases, the path of the moving-platform is optimized to further increase the wrench capability during the transition.

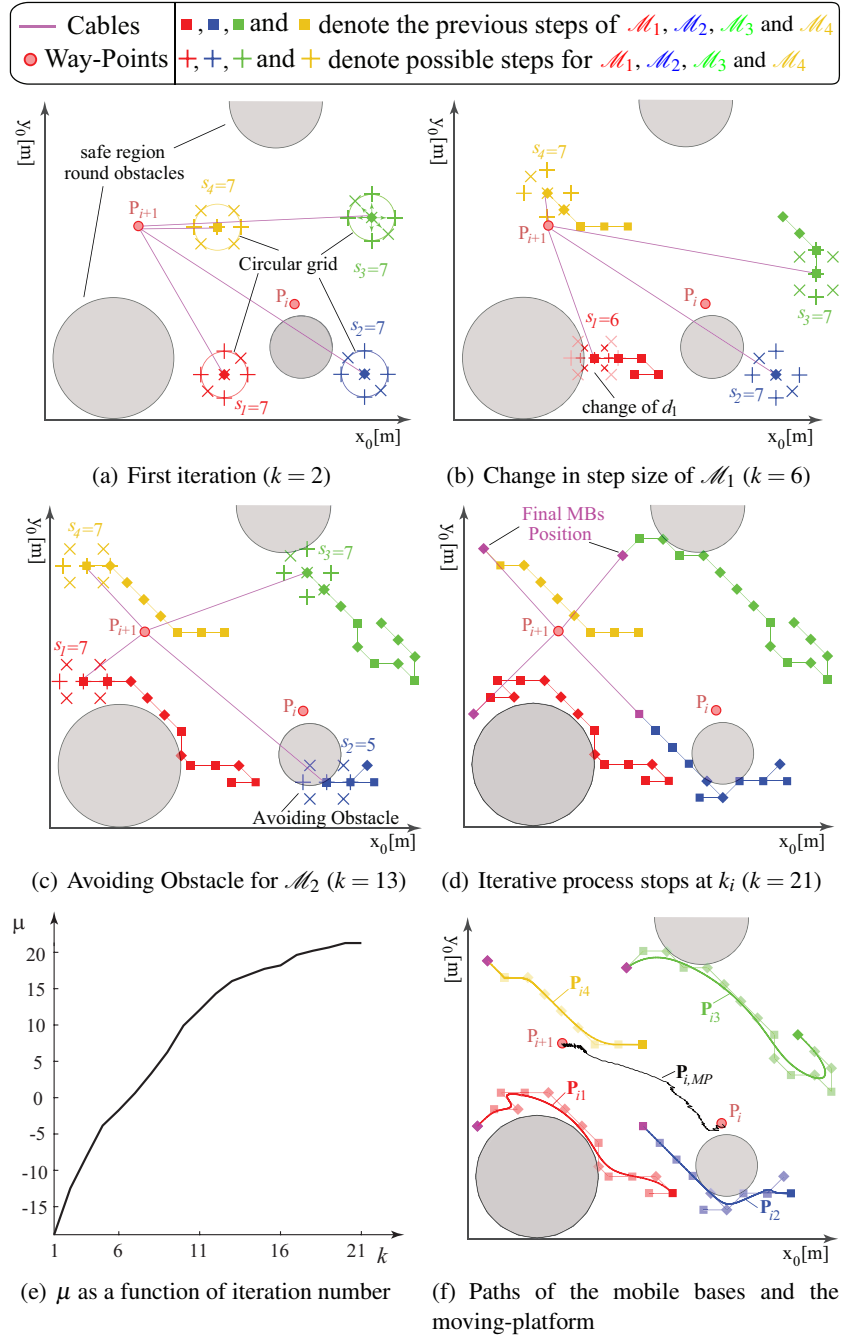


Fig. 3. Illustration of MCDPR path planning

5.1 Generation of feasible path for mobile bases

The first phase of the algorithm iteratively searches for a collision free continuous path for the mobile bases between any two way-points of the moving-platform. As an illustrative example, we will show how the algorithm evolves and calculates a continuous path for $\mathcal{M}_j, j = 1, \dots, 4$, between i th and $(i + 1)$ th way-points shown in Fig. 3(a). Let k represents an iteration. Let \mathbf{M}_{ij} be a matrix whose k th column contains the Cartesian coordinate vector of \mathcal{M}_j in x_0y_0 plane at the k th iteration of the algorithm.

From the initial configuration of the mobile bases at $k = 1$, the iterative process starts at $k = 2$ by displacing the moving-platform at P_{i+1} . This results in the drastic decrease of the moving-platform wrench capabilities (μ). At each k th iteration, the algorithm searches for the best step of \mathcal{M}_j on a circular grid with a step size d_j . On the corresponding search grid, \mathcal{M}_j can either retain its current position or due to the non-holonomic constraints, can move in the forward, backward, or diagonal directions, all denoted by '+' in Figs. 3(a), 3(b) and 3(c). Let \mathcal{M}_j have s_j possibilities of collision free steps. There exists $s_1 \times s_2 \times s_3 \times s_4$ combinations for four mobile bases. The algorithm chooses the combination that results in the maximum increase in μ . The matrix \mathbf{M}_{ij} is updated with the Cartesian coordinates of the new step taken by \mathcal{M}_j and a line segment is created between its locations at k th and $(k - 1)$ th iteration. If there are no feasible steps for \mathcal{M}_j due to the blockage around any obstacle, the step size (d_j) is reduced and the search is repeated until a feasible step is obtained as illustrated in Fig. 3(b). The procedure stops at $k = k_i$ when μ does not increase any further (see Fig. 3(d)). For the illustrative example, the evaluation of μ as a function of iteration number is shown in Fig. 3(e).

The next phase is to smooth the sequence of straight line segments generated between the Cartesian coordinates of \mathcal{M}_j during the iterative process using B-Splines [13]. The following function in MATLAB *bspline_footpoint* [14] is used which requires the two parameters *knot sequence* and *control points* to be tuned. It takes k_i Cartesian coordinates of \mathcal{M}_j in \mathbf{M}_{ij} as input, and generates ten times the smooth sequence of Cartesian coordinates, denoted as \mathbf{P}_{ij} , as depicted in Fig. 3(f).

The above procedure is repeated to find a collision free continuous path for the mobile bases between all the way-points of the moving-platform. It is noteworthy to mention that location of \mathcal{M}_j at k_i th iteration is used as an initial configuration to compute $\mathbf{P}_{(i+1)j}$ *i.e.* path of \mathcal{M}_j between $(i + 1)$ th and $(i + 2)$ th way-points. The Pseudo-code for generating the path of mobile bases is presented in Algorithm. 1.

5.2 Generation of the moving-platform optimal path

The second phase of the path planning algorithm generates an optimal path of the moving-platform, denoted as $\mathbf{P}_{i,MP}$ between i th and $(i + 1)$ th way-points. Similar to the first phase, the second phase of the algorithm is also an iterative process which computes an optimal moving-platform pose for each location of the mobile bases in $\mathbf{P}_{ij}, j = 1, \dots, 4$. Let r represents an iteration. Total number of iterations are fixed for the second phase *i.e.* equal to the number of Cartesian coordinates of \mathcal{M}_j in \mathbf{P}_{ij} .

Given the initial (\mathbf{p}_i) moving-platform pose at $r = 1$, the iterative process begins at $r = 2$. It builds $\mathbf{P}_{i,MP}$ by searching for the best pose of the moving-platform on a

Algorithm 1: Generation of MB feasible paths

Input : A matrix with initial Cartesian coordinates of mobile bases $\mathbf{M}_{ij}(1)$
Maximum number of iterations k_{max}
Cartesian coordinates of the moving-platform at $(i+1)$ th way-point \mathbf{p}_{i+1}

Output : Feasible Path of mobile bases \mathbf{P}_{ij}

Notations : j represents the MB number $j = 1, \dots, 4$
Cartesian Coordinates of \mathcal{M}_j at the k th iteration $\mathbf{M}_{ij}(k)$
A vector with Wrench Capability (μ) of the moving-platform at the k th iteration $\bar{\mu}(k)$

Invoked functions : Determine smooth path of \mathcal{M}_j $BSpline(\mathbf{M}_{ij})$

```

1  $k = 1$ ;
2  $\bar{\mu}(1) =$  Determine  $\mu$  with  $\mathcal{M}_j$  at  $\mathbf{M}_{ij}(1)$  and moving-platform at  $\mathbf{p}_{i+1}$ 
   // Section 3.2
3 repeat
4    $k = k + 1$ ;
5   Determine the new step of  $\mathcal{M}_j$  // Section 5.1
6    $\mathbf{M}_{ij}(k) =$  Cartesian coordinates for the new location of  $\mathcal{M}_j$ 
7    $\bar{\mu}(k) =$  Determine  $\mu$  with  $\mathcal{M}_j$  at  $\mathbf{M}_{ij}(k)$  and MP at  $\mathbf{p}_{i+1}$ 
8 until ( $\bar{\mu}(k) = \bar{\mu}(k-1)$  or  $k > k_{max}$ );
9  $\mathbf{P}_{ij} = BSpline(\mathbf{M}_{ij})$ 
10 return ( $\mathbf{P}_{ij}$ )

```

circular grid with a step size d_{MP} . In contrast to mobile bases, the moving-platform can either retain its previous pose or take a step in all the neighboring eight directions *i.e.*, forward, back, left, right and diagonals. The algorithm chooses a step with a maximum μ among all the possible steps. For the illustrative example, $\mathbf{P}_{i,MP}$ is shown in Fig. 3(f). The above process is repeated to find an optimal moving-platform path between all the way-points. The Pseudo-code used to generate the moving-platform path is presented in Algorithm. 2.

6 Results and Discussion

As discussed in Section 4, the required task is to displace the MoPICK moving-platform from point P_1 to P_l by sequentially following the intermediate way-points as depicted in Fig. 2. In order to perform the desired task, the output of the proposed path planning algorithm is illustrated in Fig. 4. The simulation showing the complete process of searching for a feasible path of mobile bases and the generation of the optimal path of the moving-platform between all the way-points can be seen at⁷. The video also shows the resultant motion of the complete system along with the pick-and-place operations performed at the task locations. The first step of the algorithm took 109.9 minutes to generate the path of the mobile bases. The second step of the algorithm took 15.92 minutes to determine the path of the moving platform.

⁷<https://youtu.be/0wrdLBvM9-s>

Algorithm 2: Generation of the moving-platform optimal path

Input : Path of mobile bases \mathbf{P}_{ij}
 Cartesian coordinates of the moving-platform at the i th way-point \mathbf{p}_i

Output : Optimal path of moving-platform $\mathbf{P}_{i,MP}$

Notations : j denotes number of mobile bases $j = 1, \dots, 4$
 Cartesian Coordinates of \mathcal{M}_j at r th iteration $\mathbf{P}_{ij}(r)$
 Optimal moving-platform pose at r th iteration $\mathbf{P}_{i,MP}(r)$

Invoked functions : Determine the number of Cartesian coordinates of \mathcal{M}_j in \mathbf{P}_{ij}
 $size(\mathbf{P}_{ij})$

```

1  $\mathbf{P}_{i,MP}(1) = \mathbf{p}_i$ ;
2 for  $r = 2 : size(\mathbf{P}_{i1})$  do
3    $\mathbf{P}_{i,MP}(r) =$  Compute optimal moving-platform pose with  $\mathcal{M}_j$  at  $\mathbf{P}_{ij}(r)$  // see
   Section 5.2
4 end

```

The cable tension lower bound is null. The cable tension upper bound depends on the actuation system used to actuate the cables of the MCDPR, *i.e.*, motors, winches etc. In MoPICK prototype, Dynamixel MX-64AT actuators and winches whose drum diameter is equal to 0.2 m, are used, to pull the cables. Based on the hardware specification and safety issues, the cable tension upper bound is set to 15 N.

The tuning parameters for the proposed MCDPR algorithm are the step sizes of the search grids, $d_j, j = 1 \dots, 4$ and d_{MP} . It is important to tune these parameters in order to obtain a feasible path of the MCDPR mobile bases and the moving-platform. For example, during the first phase of the algorithm, the step size d_j must stay smaller than the diameter of the smallest obstacle in the environment in order to detect its collision with the mobile bases. On the contrary, making d_j smaller also increases the computation time to calculate the feasible paths of the mobile bases. During the second phase of the algorithm, a very small step size d_{MP} can result in not achieving the desired moving-platform position while a big d_{MP} can generate large discontinuities in the moving-platform path.

7 Conclusion and Future Work

The aim of this study is to propose a path planning algorithm for kinematically redundant Mobile Cable-Driven Parallel Robots (MCDPRs). The proposed algorithm plans the MCDPR path in two subsequent stages. In the first stage, the algorithm searches for a feasible and collision free path of mobile bases. The second stage generates an optimal path of the moving-platform to reach at the desired pose. Although the obtained path between the initial and final poses may not be the shortest one, it leads to a feasible path, when it exists. The proposed algorithm is validated by simulations on MoPICK, which is a MCDPR with a three degree-of-freedom point mass end-effector displaced by four cables mounted on four mobile bases. Future work will deal with the experimental validation and the extension of the proposed methodology for the trajectory planning of MCDPRs while taking into account the cable and mobile base velocity limits.

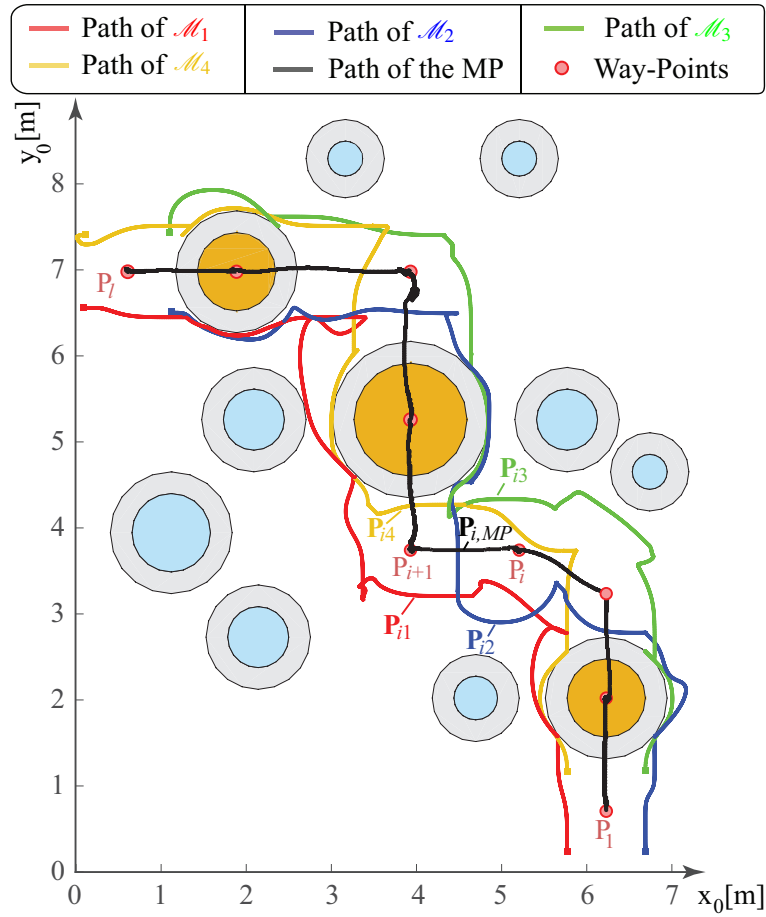


Fig. 4. Path of MoPICK for the required task

References

1. J. Albus, R. Bostelman, and N. Dagalakis. "The nist spider, a robot crane". *Journal of research of the National Institute of Standards and Technology*, 97(3):373–385, 1992.
2. S. Kawamura, H. Kino, and C. Won. "High-speed manipulation by using parallel wire-driven robots". *Robotica*, 18(1):13–21, 2000.
3. Lambert, C., Nahon, M., and Chalmers, D., 2007. "Implementation of an aerostat positioning system with cable control". *IEEE/ASME Transactions on Mechatronics*, 12(1), pp. 32–40.
4. P. Gallina, G. Rosati, and A. Rossi. "3-dof wire driven planar haptic interface". In *Journal of Intelligent and Robotic Systems*, 32(1):23–36, 2001.
5. Rasheed, T., Long, P., Marquez-Gamez, D. and Caro, S., "Tension Distribution Algorithm for Planar Mobile Cable-Driven Parallel Robots", *The Third International Conference on Cable-Driven Parallel Robots (CableCon 2017)*, Quebec City, Canada, August 24 2017.

6. Rasheed, T., Long, P., Gamez, D. M., and Caro, S. "Available wrench set for planar mobile cable-driven parallel robots". In *2018 IEEE International Conference on Robotics and Automation (ICRA)*.
7. Rasheed, T., Long, P., Marquez-Gamez, D. and Caro, S., 2018, August. "Optimal Kinematic Redundancy Planning for Planar Mobile Cable-Driven Parallel Robots". In *ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (pp. V05BT07A025-V05BT07A025). American Society of Mechanical Engineers.
8. Rasheed, T., Long, P., Gamez, D. M., and Caro, S.: "Kinematic modeling and twist feasibility of mobile cable-driven parallel robots". In *Advances in Robot Kinematics 2018*.
9. Hiller, M., Fang, S., Mielczarek, S., Verhoeven, R., and Franitza, D. "Design, analysis and realization of tendonbased parallel manipulators". In *Mechanism and Machine Theory*, 40(4), pp. 429–445, 2005.
10. L. Gagliardini, S. Caro, M. Gouttefarde, and A. Girin. "A reconfiguration strategy for reconfigurable cable-driven parallel robots". In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1613–1620, IEEE, 2015.
11. L. Gagliardini, S. Caro, M. Gouttefarde, and A. Girin. "Discrete reconfiguration planning for cable-driven parallel robots". In *Mechanism and Machine Theory*, pages 313–337, 2016.
12. G. Rosati, D. Zanotto, and S. K. Agrawal. "On the design of adaptive cable-driven systems". In *Journal of mechanisms and robotics*, 3(2):021004, 2011.
13. Pottmann, H., Leopoldseder, S. and Hofer, M., 2002. "Approximation with active B-spline curves and surfaces". In *Computer Graphics and Applications, 2002. Proceedings. 10th Pacific Conference on* (pp. 8–25). IEEE.
14. <https://www.mathworks.com/matlabcentral/fileexchange/27374-b-splines>
15. Guay, F., Cardou, P., Cruz-Ruiz, A.L. and Caro, S., 2014. "Measuring how well a structure supports varying external wrenches". In *New Advances in Mechanisms, Transmissions and Applications* (pp. 385–392). Springer, Dordrecht
16. Ruiz, A.L.C., Caro, S., Cardou, P. and Guay, F., 2015. Arachnis: "Analysis of robots actuated by cables with handy and neat interface software". In *Cable-Driven Parallel Robots* (pp. 293–305). Springer International Publishing.
17. Barraquand, J. and Latombe, J.C., "On nonholonomic mobile robots and optimal maneuvering". In *Proceedings. IEEE International Symposium on Intelligent Control*, pp. 340–347, September 1989, IEEE.
18. Connors, J. and Elkaim, G., "Analysis of a spline based, obstacle avoiding path planning algorithm". In *Vehicular Technology Conference, VTC 2007-Spring*, pp. 2565–2569, April 2007, IEEE.
19. Y. Wischnitzer, N. Shvalb, and M. Shoham, "Wire-driven parallel robot: Permitting collisions between wires," *Int. J. Rob. Res.*, vol. 27, no. 9, pp. 1007–1026, Sep. 2008
20. S. M. LaValle, *IPlanning algorithms*. Cambridge university press, 2006.
21. S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.