Kinematic Stability based AFG-RRT* Path Planning for Cable-Driven Parallel Robots

Utkarsh A. Mishra, Marceau Métillon, Stéphane Caro

To cite this version:
Utkarsh A. Mishra, Marceau Métillon, Stéphane Caro. Kinematic Stability based AFG-RRT* Path Planning for Cable-Driven Parallel Robots. The 2021 IEEE International Conference on Robotics and Automation (ICRA 2021), May 2021, Xi’an, China. hal-03182298

HAL Id: hal-03182298
https://hal.archives-ouvertes.fr/hal-03182298
Submitted on 29 Mar 2021

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.
Kinematic Stability based AFG-RRT* Path Planning for Cable-Driven Parallel Robots

Utkarsh A. Mishra¹, Marceau Métillon² and Stéphane Caro²*, IEEE Member

Abstract— Motion planning for Cable-Driven Parallel Robots (CDPRs) is a challenging task due to various restrictions on cable tensions, collisions and obstacle avoidance. The presented work aims at proposing an optimal path planning strategy in order to both maximize the wrench capability and the dexterity of the robot in a cluttered environment. First, an asymptotically-optimal path finding method based on a variant of rapidly exploring random trees (RRT) is implemented along with the GilbertJohnsonKeerthi (GJK) algorithm to account for the collision detections. Then, a goal biased Artificial Field Guide (AFG) is employed to reduce convergence time and ensure directional exploration. Finally, a post-processing algorithm is added to get a short and smooth resultant path by fitting appropriate splines. The proposed path planning strategy is analyzed and demonstrated on a simulated and experimental setup of a six-DOF spatial CDPR.

I. INTRODUCTION

Cable-Driven Parallel Robots (CDPRs) belong to the class of parallel robots in which the end-effector (EE), a moving-platform, is connected to a rigid frame by means of cables instead of articulated limbs. CDPRs can have a low inertia, a higher payload to weight ratio and a large workspace which enables a variety of usage of such robots in large scale 3D printing [1], large scale telescopes [2] and in rehabilitation mechanisms [3]. However, there are some challenges with such systems along with the advantages accompanying it. The cables used in such robots can only apply unilateral forces, i.e. they can only pull the moving platform and not push it [4].

Given that, the designing of CDPRs based on manipulability and structural stability to external wrenches plays a vital role in governing the workspace and reliability under varying conditions. Gosselin and Angeles first used the condition number of the kinematic Jacobian matrix in 1990 [5] [6] which was extended by Cui and Hao [7] as a measure of the mechanism dexterity. The problem of non homogeneous jacobian was addressed by Guay et al. [8] which further explored an index used for supporting varying force and motion capabilities of the actuator, called the capacity margin. A similar index [9] was also established for twist feasibility based on cable velocities. Various works [10], [11] have used related indices as a measure of the effectiveness of the mechanism, the design problem being generally formulated as an optimization problem.

Based on a particular configuration, motion planning for CDPRs has been explored by means of various strategies. Cluttered environment with obstacles introduces many challenges related to obstacle avoidance, cable collision and management of cable tensions. A dynamic point-to-point trajectory planning for underactuated CDPRs was proposed in [12]. Previously, various strategies to plan point-to-point trajectories using polynomial and trigonometric functions were introduced in [13], [14] for navigating through waypoints. Point-to-point motion planning is required for pick-and-place like operations. However, these works only validated the wrench feasibility conditions without obstacle avoidance. Motion planning of reconfigurable Mobile CDPRs with Direct Transcription Method while considering the static equilibrium and the tipping conditions of the overall system along the searched path was further explored [15] along with obstacle avoidance.

In context of path planning, sampling based path planning algorithms like rapidly-exploring random trees (RRT) perform effective planning in cluttered environments. RRT*, a variant of general RRT, will guarantee asymptotic optimality [16] in finding path to a goal, if it exists. A modified RRT* along with a fast GJK obstacle avoidance algorithm [17] [18] [19] was proposed in [20] considering cable/cable and cable/moving-platform collisions as well as the wrench capability of the system. A faster convergence criterion based on the condition number of the kinematic Jacobian matrix with the help of an artificial potential field was introduced in [21]. Recently, RRT* has also been used to compute efficient trajectories based on horizontal linear
periodic motion energies of the state [22].

In this paper, RRT* based offline path planning has been explored for pick-and-place operations (point-to-point motions) in a cluttered environment. A pose based capacity margin as a measure of robustness on sustaining external wrenches is used as the cost function in addition to the dexterity. Finally, the AFG-RRT* algorithm is formulated and solved under the guidance of a goal-biased Artificial Field Guide (AFG) and a GJK based detection of cable/cable, cable/obstacles and moving-platform/obstacles collisions. The overall objective is to plan the path to the goal, if it exists, maximizing the structure stability to external wrenches and with proper manipulability. The motion was simulated and demonstrated on a 6-Degrees Of Freedom (DOF) spatial CDPR shown in Fig. 1.

Section II describes the robot model and presents the problem statement. Section III formulates the cost indices. Section IV deals with the implementation of the AFG-RRT* algorithm along with the functions of transition between poses and collision detection. The simulated and experimental results are discussed in Section V. Finally, conclusions are drawn and future work is presented in Section VI.

II. ROBOT MODEL

Let us consider an \( n \)-DOF CDPR with \( m \) cables. Its \( i \)th closed loop is represented in Fig. 2. The frame \( \mathcal{F}_b \) of origin \( O \) is attached to the base. The frame \( \mathcal{F}_p \) of origin \( P \) is attached to the Moving-Platform (MP). \( \mathbf{a}_i \) denotes the Cartesian coordinates vector of exit point \( A_i \) expressed in \( \mathcal{F}_b \). \( \mathbf{b}_i \) denotes the Cartesian coordinates vector of anchor point \( B_i \) expressed in \( \mathcal{F}_p \). The MP pose is defined by the vector \( \mathbf{p} \) pointing from \( O \) to \( P \) and the unit quaternion \( \mathbf{Q}_p \in \text{SU}(2) \) (and equivalent rotation matrix \( \mathbf{R}_p \in \text{SO}(3) \)) from \( \mathcal{F}_b \) to \( \mathcal{F}_p \).

The \( i \)th loop closure equation is expressed as:

\[
\mathbf{b}_i \mathbf{u}_i + \mathbf{b}_i \mathbf{b}_i - \mathbf{b}_i \mathbf{R}_p \mathbf{b}_i - \mathbf{p} = 0
\tag{1}
\]

\( \mathbf{u}_i \) is the unit vector along the \( i \)th cable pointing from \( A_i \) to \( B_i \). The subscript \( \mathbf{b} \) (\( \mathbf{p} \), resp.) means that the corresponding vector is expressed in frame \( \mathcal{F}_b \) (\( \mathcal{F}_p \), resp.).

The upper and lower bounds on cable tensions are denoted as \( \tau \) and \( \tau_s \) respectively (\( \tau = 86 \text{ N}, \tau_s = 1 \text{ N} \)). The MP mass is named \( m_E \).

This paper aims to define a methodology to generate optimal paths in a cluttered environment from a given starting pose to a given ending pose while maximizing the overall robot performances. The maximum and minimum tension impose significant restrictions to the motion of the moving platform. The only external wrench action on the moving platform is due to its own weight. The practical setup shown in Fig. 1 of size 4.24 m x 3.67 m x 2.76 m (l x b x h) has a 5.6 Kg moving platform of size 0.28 m x 0.28 m x 0.20 m.

III. PERFORMANCE INDEX

A. Capacity Margin Formulation

Let an external wrench, \( \mathbf{w}_e \), is applied on the moving platform given by, \( \mathbf{w}_e = [\mathbf{f}_p^T \mathbf{\tau}_p^T]^T \) for an external force \( \mathbf{f}_p \) and moment \( \mathbf{\tau}_p \). For the moving platform to remain in equilibrium, there must exist a \( m \)-dimensional vector of cable tensions, \( \mathbf{t} = [t_1, t_2, ..., t_m] \), in order to satisfy the Newton-Euler equation given by,

\[
-\mathbf{W} \mathbf{t} + \mathbf{w}_e = \mathbf{0}_6
\tag{2}
\]

where, \( \mathbf{W} \) is the Wrench matrix of the mechanism at this particular pose of the moving platform, given by

\[
\mathbf{W} = \begin{bmatrix}
    b_{u_1} \\
    (1/r)b_{R_p} \times b_{u_1} \\
    \vdots \\
    (1/r)b_{R_p} \times b_{u_m}
\end{bmatrix}
\tag{3}
\]

and \( \mathbf{w}_e = [\mathbf{f}_p^T (1/r) \mathbf{\tau}_p^T]^T \) reformulated as dimensionally homogenized matrices with the help of a characteristic length \( r \). This length is determined by imposing isotropy conditions on the wrench matrix. For planar manipulators [10], it is given by \( r^2 = \frac{2}{m} \sum_{i=1}^{m} ||b_{u_i}||_2^2 \), whereas, for spatial manipulators, it is obtained for minimum Euclidean distance between square matrix \( \mathbf{W} \mathbf{W}^T \) and diagonal matrix of squared eigenvalues (\( \sigma_1, ..., \sigma_6 \)) of \( \mathbf{W} \mathbf{W}^T \).

Given the above formulation, the equilibrium in eq. (2) must hold for all the possible external wrenches \( \{\mathbf{w}_e\} \) which can be specified as path planning requirements. Let us denote this set of external wrenches as \( \mathcal{W}_e \in \mathbb{R}^6 \), given by \( \mathcal{W}_e = \{\mathbf{w}_e \in \mathbb{R}^6 \} \) represented as a polytope in the wrench space of the robot. Similarly, due to bounded cable tensions, let the allowable tension set, \( \mathcal{F} \) be composed of all the feasible cable tension vectors such that

\[
\mathcal{F} = \{ \mathbf{t} \in \mathbb{R}^m : \mathbf{t} \leq \mathbf{t} \leq \mathbf{i} \}
\tag{4}
\]

represents another polytope (a hypercube) in the tension space of the robot.

The two sets \( \mathcal{W}_e \) and \( \mathcal{F} \) are defined in two different spaces, namely the wrench space and the tension space, which are connected by eq. (2). The feasible tensions \( \mathbf{t} \in \mathcal{F} \) are linear mapped onto the wrench space using the transformation wrench matrix, \( \mathbf{W} \), to get the set of available feasible wrenches, \( \mathcal{W}_r \). Hence, after transformation, \( \mathcal{F} \) becomes a special type of polytope in wrench space called the zonotope, given by

\[
\mathcal{W}_r = \{ \mathbf{w}_r : \mathbf{w}_r = \mathbf{W} \mathbf{t} \forall \mathbf{t} \in \mathcal{F} \}
\tag{5}
and referred to as the Available Wrench Set (AWS). This can be visualized from Fig. 3 as a low dimensional representation of tension space (3 dim.) and wrench space (2 dim.).

For the system to be able to reach an equilibrium, there must be available cable tensions to balance the external wrench, i.e.,

\[ \mathcal{W}_e \subseteq \mathcal{W}_t \]  

This can be analysed graphically as the degree of inclusion of \( \mathcal{W}_e \) in \( \mathcal{W}_t \). i.e. the distance between two polytopes which is represented as a signed distance, \( s \), that is positive when \( \mathcal{W}_e \) is completely within \( \mathcal{W}_t \) and negative when \( \mathcal{W}_e \) is partly outside \( \mathcal{W}_t \). Mathematically, the Hyperplane Shifting Method (HSM) [8] is implemented to convert the zonotope, \( \mathcal{W}_t \), to a set of hyperplanes satisfying \( Ax = b \), where \( x \in \mathbb{R}^6 \) is a vector in wrench space. Finally, between the \( i \)th hyperplane, \( a_i x = b_i \), and \( j \)th task wrench, \( w_{ej} \), the signed distance, \( s_{ij} \), is given as

\[ s_{ij} = \frac{(b_i - a_i w_{ej})}{|a_i|} \]  

and capacity margin (\( \kappa \)) for such a pose is defined as the minimum degree of such inclusion, given by

\[ \kappa = \min(s_{ij}) \]  

B. Dexterity Margin Formulation

Let the linear and angular velocities of the moving platform in Cartesian space be denoted as \( ^bV \) and \( ^b\omega \) respectively. The Kinematic Jacobian matrix, \( \mathbf{J} \), defined as

\[ \begin{bmatrix} ^bV \\ ^b\omega \end{bmatrix} = \mathbf{J} \hat{\theta} \]  

linearly maps the actuated cable velocities \( \hat{\theta} \) to the twist for a particular pose and by calculation, \( \mathbf{J} = \mathbf{W}^T \). While analyzing transfer of error for the transformation from cable space to the cartesian space in eq. (9), effective manipulability can be well ensured by reducing the magnitude of the transfer of error and can be related to the dexterity of the robot [23], which is the second performance index considered in the paper. The conditioning number of the kinematic Jacobian matrix is defined as,

\[ \kappa = \frac{\lambda_{\max}(\mathbf{J}^T \mathbf{J})}{\lambda_{\min}(\mathbf{J}^T \mathbf{J})} \]

where, \( \lambda_{\max}, \lambda_{\min} \) denote the maximum and minimum eigenvalues of matrix \( \mathbf{J}^T \mathbf{J} \), \( \sigma_{\max}, \sigma_{\min} \) denote the maximum and minimum singular value of the matrix. The Jacobian matrix conditioning number, \( \kappa \) is a measurement for the dexterity and it’s a number no less than 1, i.e. \( 1 \leq \kappa \leq \infty \). Hence, the dexterity margin, \( \gamma_d \), is formulated as the reciprocal of this conditioning number i.e.

\[ \gamma_d = \frac{1}{\kappa} \]  

and thus \( 0 \leq \gamma_d \leq 1 \). When \( \gamma_d = 1 \), the mechanism is at its optimal transmission capacity. Whereas, when \( \gamma_d = 0 \) the mechanism loses its control and attains a singular configuration. Thus \( \gamma_d \) is used to measure a robots dexterity and controlling precision.

IV. MODIFIED AFG-RRT* FORMULATION

A. RRT* Algorithm

The implemented RRT* algorithm, as in Algorithm 1 includes the cost of traveling to each vertex relative to its parent vertex based on the cost function which is modelled as a combination of the capacity, \( \kappa \), and dexterity, \( \gamma_d \), margins defined in the previous section. A simple path

\begin{algorithm}
\begin{algorithmic}
\State Result: Path from Start to Goal Node
\While {True}
\For {V in G do}
\If {Distance(V, Goal) < (\epsilon, \delta)}
\State \text{return} G;
\EndIf
\EndFor
\State X_{rand} = RandomNode(AFG, Obstacle);
\State X_{nearest} = Nearest(G, X_{rand});
\State X_{new} = Transition(X_{rand}, X_{nearest});
\If {PATH(X_{rand}, X_{nearest}) \in Obstacle}
\State \text{Continue};
\Else
\State X_{new}.Cost = Cost(X_{new}, X_{nearest});
\State X_{best}.Xneighbors = Neighbors(G, X_{new}, D_{W});
\State Link = Chain(X_{new}, X_{best});
\For {X \in X_{neighbors} do}
\If {X_{new}.Cost + Cost(X_{new}, X) < X.Cost}
\State X.Cost = X_{new}.Cost + Cost(X_{new}, X);
\State Parent(X) = X_{new};
\State G \leftarrow \{X_{new}, X\};
\EndIf
\EndFor
\State G \leftarrow Link;
\EndIf
\EndIf
\EndWhile
\end{algorithmic}
\end{algorithm}

cost is the cumulative increment in capacity margin and dexterity achieved during the transition. Points are generated over the workspace and connected to the closest available node based on the cost metric. For a graph, \( G(V, E) \), of edges and vertices initialized with a start node, the closest
node \((\mathbf{X}_{\text{new}})\) is obtained. Now, if a node with a better
cost \((\mathbf{X}_{\text{best}})\) is obtained within a neighborhood of existing
vertices \((\mathbf{X}_{\text{neighbours}})\) marked by a fixed distance radius, \(D_N\),
it will replace the proximal node using the distance metric.
The algorithm ends when a node is generated within the
goal region bounded by \(\varepsilon\) Cartesian distance and \(\delta\) angular
distance. The overall transition between intermediate nodes
in governed by a pose transition rule and checked for obstacle
avoidance.

B. Collision Detection

In order to account for the large number of collision checks
in sampling based algorithms like RRT*, the GJK algorithm
is integrated for its faster and more accurate collision detection
behavior. It calculates the distance between objects, with
reference to distance from origin. To deal with the limitation
of the algorithm to convex polytopes, the moving-platform,
obstacles and the cables are approximated as convex objects
[20]. All possible interference between the moving-platform,
cables and obstacles (shown in Fig. 4) are considered to
determine the paths free of collision between two nodes.

![Fig. 4: Convex shape representation of the Moving Platform (MP), Cables (\(A_iB_i\)) and Obstacles (\(O_i\)) for satisfying the prerequisites of GJK Algorithm](image)

For considering collisions between two convex polytopes
A and B, Algorithm 2 defines the support function (getSupport()) as the Minkowski
difference between the farthest A and B. Algorithm 2 defines the support function (getSupport()) as the Minkowski
difference contain the origin, are in a state of collision.

A sub-algorithm is iterated within, which makes a simplex
in Minkowski difference try to contain the origin. If such
an event occurs, the polytopes, that make the Minkowski
difference contain the origin, are in a state of collision.

C. Transition Law

The transition between intermediate poses play a signifi-
cant role in the nature of the solution obtained by the RRT*
exploration policy. The node generated for the above RRT*
algorithm represents a configuration of the mobile platform
i.e. \([p, Q]_N\). The transition from a pose, \([p_1, Q_1]_N\), towards
a randomly sampled pose, \([p_2, Q_2]_N\), is modelled using a
fixed transition ratio, \(\lambda\) and new pose after transition can
be expressed as a linear interpolation of the position vector
and a spherical linear interpolation of the unit quaternions.

For the new pose, \([p_N, Q_N]_N\),

\[
\mathbf{p}_N = \mathbf{p}_1 + \lambda (\mathbf{p}_2 - \mathbf{p}_1) \tag{12}
\]

\[
\mathbf{Q}_N = \frac{\sin[(1 - \lambda)\Omega]}{\sin \Omega} \mathbf{Q}_1 + \frac{\sin[\lambda\Omega]}{\sin \Omega} \mathbf{Q}_2 \tag{13}
\]

where, \(\cos \Omega = \mathbf{Q}_1 \cdot \mathbf{Q}_2\), i.e. the 4-dimensionsal dot product
between the two quaternions of the form \(\mathbf{Q} = [q_0, \mathbf{q}_1 (3\times1)]\).

D. Artificial Field Guide (AFG)

Simple uniform random number generators reaches
asymptotic optimality with RRT*. Such numbers encourage
exploration at the expense of increasing computation time
[21]. Thus, a goal biased artificial field guided formulation
of generating random numbers is used to attract the search
between the goal and, ensure faster convergence. This way the
searching and exploration is much more directional towards
the desired goals. Nodes, \(N(p, Q)\) are generated based on the

\[
N(p) = C(p) + (D(s(\psi)c(\phi), D(s(\psi)s(\phi), Dc(\psi))) \tag{14}
\]

\[
P(D \mid D \in [D_{\text{min}}, D_{\text{max}}]) \propto \frac{1}{\|N(p) - C(p)\|_2} \tag{15}
\]

\[
N(Q) = \frac{\sin[(1 - \lambda_r)\Omega_r]}{\sin \Omega_r} C(Q) \tag{16}
\]

where \(s(\psi) = \sin \psi\) and \(c(\psi) = \cos \psi\). The probability of
choosing \(D\) in eq. (15) governs the attraction phenomenon
where \(D_{\text{min}}\) is a decision variable and \(D_{\text{max}}\) is the maxi-
mum radius coverage of the AFG (in Fig. 5) extended to
\(D_{\text{max}} = \Delta r_{\text{workspace}}. N(p, Q)\) must be within the wrench
feasible workspace (i.e. \(\mathcal{K}_{\psi, \phi, \Omega_r N(p, Q)} \geq 0\)) and outside any
obstacle. Further chosen conditions on decision variables
are \(\psi, \phi, \Omega_r \in \text{Random}(0, 2\pi)\) and \(\lambda_r \in \text{Random}(0, 1)\). The

---

**Algorithm 2: GJK Collision Detection**

**Result:** Collision Flag

1. Direction \(\vec{v}\), Simplex \((\mathcal{F}_0)\) ← Initialize()
2. Next Point \((\vec{w}) = \text{getSupport}(\vec{v})\)
3. \(\mathcal{F} \leftarrow \text{newSimplex}(\mathcal{F}_0, \vec{w})\)
4. \(\vec{v} \leftarrow \text{updateDirection}(-\vec{w})\)
5. **while** False **do**
   1. \(\vec{w} = \text{getSupport}(\vec{v})\)
   2. **if** \(\vec{w}, \vec{v} < 0\) **then**
      1. \(\vec{v} \leftarrow \text{updateDirection}(-\vec{w})\)
      2. **return** False
   3. **else**
      1. \(\mathcal{F} \leftarrow \text{newSimplex}(\mathcal{F}, \vec{w})\)
      2. **if** intersect\((\mathcal{F}, \mathcal{O})\) **then**
         1. **return** True
      3. **else**
         1. \(\mathcal{F} \leftarrow \text{reduceSimplex}(\mathcal{F})\)
         2. \(\vec{v} \leftarrow \text{updateDirection}(-\vec{w})\)
      **return** False
   **end**
**end**

---
probability of $C$ being the goal node or any random node is defined using a probability $P_g$, i.e., $P(C = X_{goal}) = P_g$ and $P(C = X_{rand}) = 1 - P_g$.

The AFG, as visualized from Fig. 5, is modelled as a potential field where the random node generation is based on an attraction phenomenon by the goal node $X_{goal}$. The field also preserves randomness by generating random nodes and the degree of randomness is controlled by adjusting $P_g$. The generated node will be neglected if it lies on the obstacles.

**E. Post Processing**

The intermediate nodes between the start and the goal node, as obtained by the modified RRT* undergo a post-processing strategy while retracing the nodes from the goal to the start. Each subsequent node will be connected to the most furthest node, if the path connecting them lies out of the obstacle region. The final output is provided as the post processed output and a spline is fitted based on the final nodes to get a smoother path.

**V. RESULTS AND DISCUSSIONS**

**A. Simulation Results**

In this section, the above formulated AFG-RRT* algorithm is simulated in a cluttered environment to obtain near optimal path for point to point motions. The complete strategy is simulated on a 6-DOF spatial CDPR shown in Fig. 1. The rotation of the MP about x and y axes is restricted in order to simulate pick-and-place operations. Therefore, a node for RRT* is defined by a 5-dimensional vector composed of $[p, Q]$ where $p = [x, y, z]$ and $Q = [q_0, q_1]$ which forms a unit quaternion $q = q_0 + 0\hat{i} + 0\hat{j} + q_1\hat{k}$ representing MP position and orientation about z-axis (also denoted as $\beta$ in Euler angle notation in paper).

The wrench matrix was calculated for each pose without considering orientation i.e. $\beta = 0$ rad and the tension space was mapped into the wrench space to get $\mathcal{W}_t$. $\mathcal{W}_c$ is considered to be composed of the gravity wrench of the MP, i.e., $w_E = [0, 0, m_E g, 0]$ where, $g = 9.81$ m/s$^2$ is the gravitational acceleration.

Fig. 6a depicts the isocontours of the capacity margin $\gamma_c$ for a constant and a null orientation ($\beta = 0$) of the MP throughout the $xz$ plane and for the mid-plane along $y$-direction. This margin was normalized by considering its upper bound, $\gamma_c^{max} = 2D$ such that $\gamma_c \in (0, 1)$. Similarly, Fig. 6b illustrates the isocontours of the dexterity index $\gamma_d$ throughout the same plane. $\gamma_d$ is bounded between 0 and 1 too. The global performance index $\gamma$ is defined as a convex combination of indices $\gamma_c$ and $\gamma_d$, namely,

$$\gamma = \gamma_c + \gamma_d \in (0, 2)$$  \hspace{1cm} (17)

This metric is used to conduct the AFG-RRT* simulation while analyzing separately for the individual metrics ($\gamma_c$ and $\gamma_d$). All the simulations are performed using ©MATLAB with CPU computations on an Intel ®i7-7500U CPU@2.70GHz.

**TABLE I: Parameter values for AFG-RRT* algorithm**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_N$</td>
<td>0.15 m</td>
</tr>
<tr>
<td>$\varepsilon$</td>
<td>0.15 m</td>
</tr>
<tr>
<td>$\delta$</td>
<td>$\pi/16$ rad</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>0.05</td>
</tr>
<tr>
<td>$D_{min}$</td>
<td>0.05 m</td>
</tr>
<tr>
<td>$P_g$</td>
<td>0.7</td>
</tr>
</tbody>
</table>

The AFG-RRT* was initialized using the parameter values given in Tab. I and the task was formulated
by considering various obstacles, according to Fig. 7, hindering motion of the cables as well as that of moving-platform. The initial and final nodes are defined by \( p_{\text{initial}} = [1.8, 0.4, 0.1], \ Q_{\text{initial}} = [1, 0] \) and \( p_{\text{final}} = [1.8, 3.6, 0.1], \ Q_{\text{final}} = [1, 0] \), respectively. The AFG setup is such that the directional exploration towards the final goal point dominates over randomness.

The obtained path, orientation about z-axis and the rewinded path according to the post-processing algorithm is given in Fig. 8a, Fig. 8b and Fig. 9 respectively. The overall path is subjected to collision avoidance between the cables, MP and the obstacles.

The algorithm was simulated for a number of times and the performance index achieved was recorded along with the number of iterations of AFG-RRT* and the analysis results can be visualized in Tab. II. 215 iterations was required to achieve a mean capacity margin equal to 0.55 and mean dexterity margin of 0.69. The average computation time required for simulating the complete AFG-RRT* simulation was 13 minutes.

### VI. CONCLUSIONS

A structural robustness based path planning was performed based on a goal-biased AFG coupled random sampling RRT* algorithm. The algorithm ensures directional exploration, maximizes kinematic stability, maintains efficient manipulability and achieves fast convergence. The algorithm was simulated and demonstrated for a 6-DOF CDPR setup with 8 cables. An analysis of various performance metrics showed the effectiveness of the proposed methodology in planning a path for a suspended CDPR in a cluttered environment.

Table: Performance metrics obtained from 100 simulations of AFG-RRT* algorithm

<table>
<thead>
<tr>
<th>Metric</th>
<th>Mean Value</th>
<th>Max Value</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \gamma_C )</td>
<td>0.5542</td>
<td>0.7122</td>
<td>0.1104</td>
</tr>
<tr>
<td>( \gamma_D )</td>
<td>0.6934</td>
<td>0.8237</td>
<td>0.0978</td>
</tr>
<tr>
<td>( \gamma_{\theta} )</td>
<td>1.0941</td>
<td>1.4277</td>
<td>0.1507</td>
</tr>
<tr>
<td>No. of Iterations</td>
<td>215</td>
<td>317</td>
<td>22</td>
</tr>
<tr>
<td>Computing time</td>
<td>13.78 min</td>
<td>15.38 min</td>
<td>1.70 min</td>
</tr>
</tbody>
</table>

Fig. 10: CDPR prototype and cluttered environment made up of 18 boxes

---

Table II: Performance metrics obtained from 100 simulations of AFG-RRT* algorithm

---

REFERENCES


