Forward Kinematics for Suspended Under-Actuated Cable-Driven Parallel Robots With Elastic Cables: A Neural Network Approach

Utkarsh Mishra, Stéphane Caro

Forward Kinematics for Suspended Under-Actuated Cable-Driven Parallel Robots with Elastic Cables: A Neural Network Approach

Utkarsh A. Mishra
Mechanical and Industrial Engineering Department
Indian Institute of Technology Roorkee
Uttarakhand, India 247667
Email: utkarsh75477@gmail.com

Stéphane Caro*
CNRS, Laboratoire des Sciences du Numérique de Nantes
UMR CNRS 6004,
1, rue de la Noë, 44321 Nantes, France
Email: stephane.caro@ls2n.fr

ABSTRACT

Kinematic analysis of under-constrained Cable-Driven Parallel Robots has been a topic of interest because of the inherent coupling between the loop-closure and static equilibrium equations. The non-linearity of the problem is magnified with the addition of the coupling between the cable lengths and their tensions based on the elastic cable model. The paper proposes an unsupervised neural network algorithm to perform real-time forward geometrico-static analysis of such robots in a suspended configuration under the action of gravity. The formulation determines a non-linear function approximation to model the problem and proves to be efficient in solving consecutive and close waypoints in a path. The methodology is applied on a six-degree-of-freedom (6-DOF) spatial under-constrained suspended cable-driven parallel robot. Specific comparison results in simulation and hardware to show the effectiveness of the proposed method in tracking a given path are illustrated. Finally, the degree of constraint satisfaction is presented against the results obtained from non-linear least-square optimization.

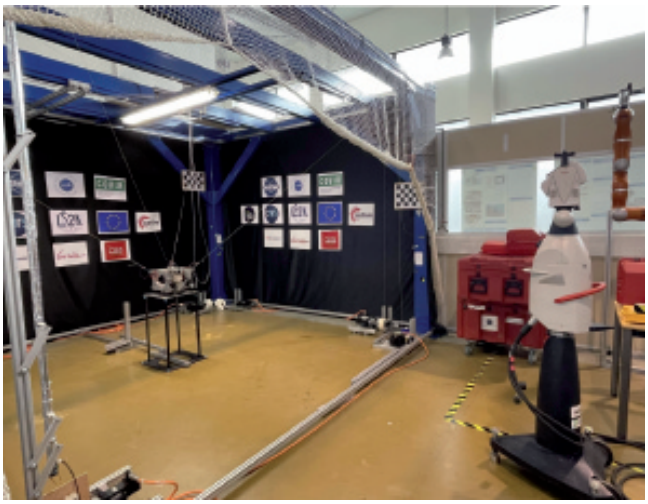


Fig. 1: The complete setup for hardware experiments with the CDPR (left) and the laser tracking system (right).

1 INTRODUCTION

Cable-Driven Parallel Robots (CDPRs) are a particular class of parallel robots whose moving-platform (MP) is connected to the robot fixed base frame by a number of cables. Hereafter, the connection points between the cables and the base frame will be referred to as exit points. Such a design brings advantages of low inertia, high payload to weight ratio, and a significantly large workspace as compared to their serial and parallel counterparts. Numerous applications of such robots have been demonstrated, for instance, large scale 3D printing [1], rescue robots [2], large scale telescopes [3], in rehabilitation mechanisms [4] and transfer robots for the elderly [5].

The kinematic analysis for CDPRs proves to be more complex as compared to the classical parallel robots with rigid links. This is generally driven by the fact that cables can only apply unilateral forces, i.e., only pull the MP and not push it, making it fairly difficult to account for static equilibrium. Many researchers have extensively contributed to the kinematic analysis of such robots. The cable lengths tend to be the most feasible proprioceptive measurement for

*Address all correspondence to this author.

such robots as compared to other physical quantities like cable tensions or orientations. There has been considerable work including angular sensors [6], camera-based pose estimations [7] and tension sensors. However, such incorporations come at their own cost of assembly and accuracy. If only cable lengths, i.e., pulley motor position measurements, are considered, the kinematic analysis is significantly influenced by the cable model used. Cables can be considered mass-less and non-elastic, mass-less but elastic, or with non-negligible mass and elastic [8]. For the introduction of a novel algorithm to solve the Direct Geometrico-Static Problem (DGSP) of CDPRs, this paper considers massless, elastic cables and suspended CDPRs.

Apart from the nature of cables and the robot configuration, the number of cables modulating the intended degrees of freedom (DoFs) plays a vital role in influencing the system kinematic analysis. Fully-constrained configurations consist of the same number of cables as the degrees of freedom, the external wrench being the final virtual cable to guarantee tautness of all the cables [9]. However, with the reduction in the number of cables, the system becomes under constrained. Under-constrained CDPRs typically have n-DoFs controlled by $m(< n)$ cables allowing only m DoFs to be controlled. The kinematic analysis of such robots is inherently coupled with static analysis and cannot be solved without solving the combined kinetostatic problem [10]. For such systems, even when the cable lengths are fixed, the MP still moves and adjusts according to the external wrench. Accordingly, the geometrico-static [11] problem becomes more complex and may have several solutions.

The forward kinematics analysis for under constrained suspended CDPRs has been approached as finding all the possible solutions to the geometrico-static equations or incorporating various iterative strategies to solve in real-time. Interval analysis [12,13] is one possible approach to solve for the complete analysis taking into account that some cables may be slack and have been explored for underconstrained systems. Various works have led to optimization problems for real-time analysis based on minimizing the potential energy [14], finding the lowest equilibrium pose [15] or minimizing cable tensions [16]. Iterative (usually, Levenberg-Marquadt) algorithms have found themselves useful in locally minimizing pose errors [17], using Hessian matrix to construct convex problem [14] and Jacobian matrix to solve the linearized approximation of the FK problem [18] at hand.

The optimization problem discussed for the real-time analysis can be seen to be a highly non-linear system and can be easily correlated to neural network architecture. Artificial Neural Network (ANN) has been used extensively in recent research to solve the inverse analysis of serial robots and forward analysis of parallel robots. Supervised methods are employed by collecting ground truth data and training to find a function approximation. Such practices have been extensively applied to serial [19], and cable [20] robots As most controllers directly work on joint space applying direct control over motor winches to modulate cable lengths, path planning in joint space [21] is a better option. In such a situation, processing Cartesian poses for accurate tracking of a planned path can take advantage of a real-time forward kinematics module.

In this paper, we propose an unsupervised feed-forward neural network algorithm based forward kinematics of under-constrained suspended CDPRs with elastic cables. On the one hand, the under-constrained configuration couples the kinematics and statics of the robot, whereas, on the other hand, the elastic cable model couples the cable lengths with their tensions. This makes the problem formulation highly non-linear and coupled. The algorithm is motivated by the aim of performing forward kinematics for such situations while tracking a pre-planned path and is fulfilled using the unsupervised neural network weight adaptation strategy. Assuming that the waypoints belonging to the path are close to each other, our model does not require significant weight updates. Thus, the convergence turns out to be slower for the starting position and faster for the waypoints as compared to the popular least-square non-linear (*lsqnonlin* by ®MATLAB) optimization framework. Finally, a pre-planned path is tracked using a six-degree-of-freedom (6-DOF) spatial suspended CDPR with four driving cables. The overall performance for the proposed algorithm and *lsqnonlin* is compared based on the time efficiency and accuracy. Extending from our previous contributions [22, 23], we extend the approach to validate on hardware with the setup as shown in Fig. 1 for a particular set of poses from the pre-planned simulated trajectory.

The model of the manipulator and the associated nomenclature are discussed in Section 2. Section 3 describes the elastic cable model used in the paper. Section 4 formulates the geometrico-static problem of the CDPR under study. The unsupervised neural network is described in Section 5. Section 6 and Section 7 give the simulation and hardware results, respectively, obtained using the proposed approach. Conclusions and future work are drawn in Section 9. Accompanying video can be found here. [1]

2 PARAMETRIZATION

Let us consider an n-DOF CDPR with m cables. Its ith 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 rotation matrix ${}^b\mathbf{R}_p \in SO(3)$ from $\mathcal{F}_b$ to $\mathcal{F}_p$.

The ith loop closure equation is expressed as:

$$ {}^b\mathbf{a}_i + l_i^b \mathbf{u}_i - {}^b\mathbf{R}_p {}^p\mathbf{b}_i - {}^b\mathbf{p} = \mathbf{0}_3 \tag{1} $$

$\mathbf{u}_i$ is the unit vector along the ith cable pointing from A_i to B_i and $\mathbf{0_3}$ is a three dimensional zero vector. The subscript ^{b} (^{p},

[1] utkarshmishra04.github.io/redirects/jmr2022.html.

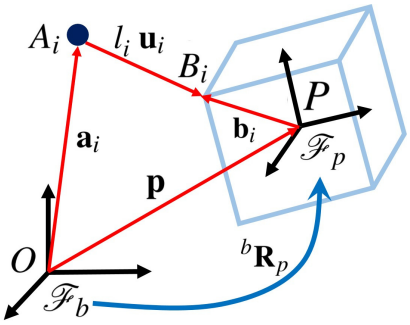


Fig. 2: Schematic of the ith closed-loop of the CDPR.

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 $\bar{t}$ and $\underline{t}$, respectively ($\bar{t}$ = 86 N, $\underline{t}$ = 1 N). The pre-planned path used for the verification purposes are chosen to be within the wrench feasible workspace of the robot (based on CRAFT [24]). The MP mass is named m_E.

3 ELASTIC CABLE MODEL

We consider the massless elastic cable model wherein the cables are modelled as linear springs instead of rigid links. Their elongation is a function of their elasticiy and tension. This eventually couples the kinematic analysis of the robot with the statics. Naturally, for a cable to get into a stable elongation, it is a logarithmic function of time [25]. However, for simplicity, we consider solving for the final elongation and use the stress-strain relationship [8]. For a cable, let the parameters be A_0, E_0, t_c, ℓ_a and ℓ_e defined as the undeformed cross-sectional area, elastic modulus, cable tension, length from the actuator-encoder and the final elongated length, respectively. Then, the elastic elongation, $\Delta\ell$, is given by:

$$\Delta\ell = t_c \frac{\ell_a}{E_0\, A_0} \tag{2}$$

and along with the relation, $\ell_e = \ell_a + \Delta\ell$, the undeformed length can be obtained as a function of cable tension and elongated length, given by:

$$\ell_a = \frac{\ell_e}{1 + \frac{t_c}{E_0\, A_0}} \tag{3}$$

4 GEOMETRICO-STATIC MODELING

After realizing the first coupling between kinematics and statics based on elastic cable model in Section 3, this section formulates the second coupling between them based on the under-constrained suspended configuration. Specifically, this section presents the static equilibrium formulations with the inverse and forward kinematics. It should be noted that all occurrences of ℓ_i correspond to the elongated cable length.

4.1 Static Equilibrium

Let an external force $\mathbf{f}_p$ and moment $\boldsymbol{\tau}_p$ be applied on the moving platform. For the moving platform to remain in equilibrium, there must exist a m-dimensional vector of cable tensions, $\mathbf{t} = [t_1, t_2, \ldots, t_m]$, satisfying the Newton-Euler equation given by,

$$-\mathbf{W}\,\mathbf{t} + \mathbf{w_e} = \mathbf{0}_n \tag{4}$$

where $\mathbf{W}$ is the normalized wrench matrix of the mechanism at this particular MP pose, given by

$$\mathbf{W} = \begin{bmatrix} {}^b\mathbf{u}_1 & \ldots & {}^b\mathbf{u}_m \\ (1/r)\,{}^b\mathbf{R}_p\,{}^p\mathbf{b}_1 \times {}^b\mathbf{u}_1 & \ldots & (1/r)\,{}^b\mathbf{R}_p\,{}^p\mathbf{b}_m \times {}^b\mathbf{u}_m \end{bmatrix} \tag{5}$$

and $\mathbf{w}_e = [\mathbf{f}_p^T \;\; (1/r)\boldsymbol{\tau}_p^T]^T$ as dimensionless homogenized matrices with the help of a characteristic length r which is defined as $r^2 = 1/m \sum_{i=1}^{m} \left\| {}^{\mathcal{F}}\mathbf{b}_i \right\|_2^2$ (refer [26, 27]). Hereafter, we denote $\mathbf{W}$ as $\mathbf{W}(\mathbf{s})$, because wrench matrix is a function of the MP pose, $\mathbf{s}$ (defined in the next subsection).

Now, as the wrench matrix formulation is similar for both massless rigid and elastic cables [8], the tension distribution obtained from Eq. (4) holds true for our case.

4.2 Inverse Kinematics

The inverse kinematics (IK) formulation for CDPRs is the mapping from the Cartesian space to the cable space. For a given pose (position + orientation) of the moving-platform (MP) of a n-DOF CDPR, IK computes the undeformed lengths of the m cables.

The MP pose, $\mathbf{s} = [{}^b\mathbf{p}, {}^b\boldsymbol{\varphi}]^T$, is given by the position vector of the geometric center of the moving platform ${}^b\mathbf{p}$ and its orientation with respect to the base frame ${}^b\boldsymbol{\varphi} = [\phi, \theta, \psi]$. Thus, ${}^b\mathbf{R}_p = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi)$ and from Eq. (1),

$$\ell_i\,{}^b\mathbf{u}_i = {}^b\mathbf{p} + {}^b\mathbf{R}_p\,{}^p\mathbf{b}_i - {}^b\mathbf{a}_i \tag{6}$$

$$\ell_i = \left\| {}^b\mathbf{p} + {}^b\mathbf{R}_p\,{}^p\mathbf{b}_i - {}^b\mathbf{a}_i \right\|_2 \tag{7}$$

where $\boldsymbol{\ell}_a = [\ell_{a1}, \ell_{a2}, \ldots, \ell_{am}]$ is the vector of undeformed actuated cable lengths obtained using the corresponding elongated cable lengths, ℓ_i (from Eq. (7)), and cable tensions, $\mathbf{t}$ (from Eq. (4) using MP pose $\mathbf{s}$) through Eq. (3). $\|.\|_2$ denotes the two-norm of a vector. This IK mapping from $\mathbf{s}$, $\mathbf{t}$ to $\boldsymbol{\ell}_a$ is referred to the mapping function defined as:

$$\xi : \mathbf{s} \in \mathbb{R}^n, \mathbf{t} \in \mathbb{R}^m \mapsto \boldsymbol{\ell}_a \in \mathbb{R}^m$$

4.3 Forward Kinematics

The forward kinematics (FK) of a cable-driven manipulator consists of obtaining the MP pose, $\mathbf{s}$, based on given

cable lengths $\boldsymbol{\ell}_a$ and external wrench. For the under-actuated CDPR studied in this paper, the kinematic model is under-determined. Thus, the FK problem is set up as a minimization problem where the relative error between the given actuated cable lengths, $\hat{\boldsymbol{\ell}}_a$, and the lengths obtained from the IK at the current pose, $\boldsymbol{\ell}_a = \xi(\mathbf{s},\ \mathbf{t})$ is a minimum. The error required to be minimized is given by:

$$e(\mathbf{s}) = \left\| \hat{\boldsymbol{\ell}}_a - \xi(\mathbf{s},\ \mathbf{t}) \right\|_2 \tag{8}$$

Thus, for a given $\hat{\boldsymbol{\ell}}_a$, the FK formulation is expressed as:

$$\zeta : \hat{\boldsymbol{\ell}}_a \in \mathbb{R}^m \mapsto \mathbf{s} \in \mathbb{R}^n$$
$$\text{such that}$$
$$\zeta(\hat{\boldsymbol{\ell}}_a) = \arg\min_{\mathbf{s}} \left\| \hat{\boldsymbol{\ell}}_a - \xi(\mathbf{s},\ \mathbf{t}) \right\|_2$$

5 NEURAL NETWORK FORMULATION

The components of robot kineto-statics play a vital role in defining the solution of the forward kinematics and the feasibility of the solution as a measure of satisfaction of the static equilibrium. Unlike serial robots, solving the forward kinematics is relatively difficult than solving the inverse kinematics for such parallel robots. The proposed neural network (NN) formulation is guided by an unsupervised iterative strategy to solve the surrogate objective of finding a suitable MP pose for given cable lengths.

The complete NN framework is built to take the cable lengths from the trajectory in joint space along with the external wrench being applied on the MP. Thus, the input vector is given by $\mathbf{x} = [\hat{\boldsymbol{\ell}}_a^T\ \mathbf{w_e}^T]^T$. Finally, the framework solves for the MP pose and required cable tensions to maintain the MP in a static equilibrium while considering elastic cable elongations. Accordingly, the output vector is given by $\mathbf{y} = [\mathbf{s}^T\ \mathbf{t}^T]^T$. The overall proposed strategy is formulated as solving:

$$NN : \mathbf{x} \in \mathbb{R}^{m+n} \mapsto \mathbf{y} \in \mathbb{R}^{m+n}$$
$$\text{s.t. for } \hat{\boldsymbol{\ell}}_a, \mathbf{w_e} \in \mathbf{x} \text{ and } \mathbf{s}, \mathbf{t} \in \mathbf{y}$$
$$NN(\mathbf{x}) = \mathbf{y} = \arg\min_{\mathbf{s},\ \mathbf{t}} \left\| \hat{\boldsymbol{\ell}}_a - \xi(\mathbf{s},\ \mathbf{t}) \right\|_2$$
$$-\mathbf{W}(\mathbf{s})\ \mathbf{t} + \mathbf{w_e} = \mathbf{0}_n$$

The above quadratic optimization problem was formulated as a surrogate objective ($\mathcal{L}$) based on penalty formation from the constraints such that

$$\mathcal{L}(\mathbf{y} = [\mathbf{s}^T, \mathbf{t}^T]^T) = \left\| \hat{\boldsymbol{\ell}}_a - \xi(\mathbf{s},\ \mathbf{t}) \right\|_2^2 + \mu \| -\mathbf{W}(\mathbf{s})\ \mathbf{t} + \mathbf{w_e} \|_2^2 \tag{9}$$

However, it can be observed that the summation holds two different measurement units, namely, square meter and Newton, respectively. In order to homogenize the surrogate objective, the characteristic length (meter) and the weight (Newton) of the MP were used respectively. Hence, the homogenized objective function is expressed as:

$$\hat{\mathcal{L}}(\mathbf{y}) = \frac{1}{r^2} \left\| \hat{\boldsymbol{\ell}}_a - \xi(\mathbf{s},\ \mathbf{t}) \right\|_2^2 + \mu \left(\frac{\| -\mathbf{W}(\mathbf{s})\ \mathbf{t} + \mathbf{w_e} \|_2}{m_E\ g} \right)^2 \tag{10}$$

where $g = 9.81 m/s^2$ is the acceleration due to gravity. Such a penalty function formulation can be solved using Gradient Descent, Newton or Quasi-Newton method and Trust Region methods.

5.1 Network Architecture

The formulated input-output problem is solved by means of an unsupervised neural network scheme where the error is calculated using the inverse kinematic lengths of the predicted MP pose vector. The gradients are calculated based on the Stochastic Gradient Descent (SGD) algorithm and the weights are updated based on those gradients.

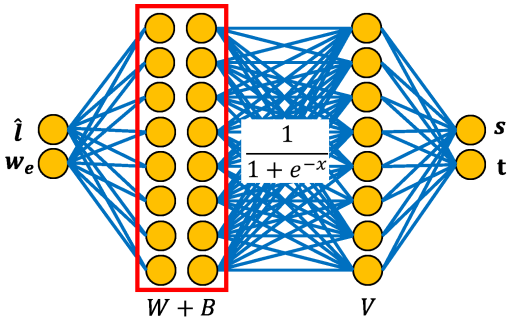


Fig. 3: Considered neural network architecture with the activation function.

Consider a Neural Network (NN) consisting of two hidden layers with weights W of shape $h \times (m+n)$, V of shape $(m+n) \times h$ and biases B of shape $h \times 1$ respectively, where h is the number of hidden nodes in the hidden layers. The activation functions used for the hidden layers is the sigmoid function, $\sigma(x)$, defined by:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{11}$$

The back-propagation error is defined using $\hat{\mathcal{L}}$ from Eq. (8). From SGD formulation, the weights and biases are optimized with reference to calculated change in weights δ_W of shape $h \times (m+n)$, δ_V of shape $(m+n) \times h$ and biases δ_B

of shape $h \times 1$ given by:

$$\delta_W(j,i) = \frac{\partial \hat{\mathcal{L}}}{\partial W(j,i)} \quad \forall \quad i = [1\ (m+n)], \quad j = [1\ h] \quad (12)$$

$$\delta_V(k,j) = \frac{\partial \hat{\mathcal{L}}}{\partial V(k,j)} \quad \forall \quad j = [1\ h], \quad k = [1\ (m+n)] \quad (13)$$

$$\delta_B(j,1) = \frac{\partial \hat{\mathcal{L}}}{\partial B(j,1)} \quad \forall \quad j = [1\ h] \quad (14)$$

The updated weights and biases (W_{new}, V_{new} and B_{new}) are obtained by using the above $\delta_{\square}$ values obtained for the corresponding current weights and biases ($W_{current}, V_{current}$ and $B_{current}$). The update equations are given by:

$$W_{new} = W_{current} - \alpha\delta_W \quad (15)$$

$$V_{new} = V_{current} - \alpha\delta_V \quad (16)$$

$$B_{new} = B_{current} - \alpha\delta_B \quad (17)$$

where α is defined as the learning rate, typically in the range of 10^{-4}.

5.2 Calculation of Derivatives and Weight Updates

The final problem gets reduced to calculation of the gradients, i.e, the changes in weights. To solve this problem, some pre-calculations are done to represent the predicted output vector $\mathbf{y}^{nn}$ from the input $\mathbf{x}$, such that,

$$\mathbf{y}^{nn}{}_{(m+n)\times 1} = V_{(m+n)\times h}\sigma(W_{h\times(m+n)}\mathbf{x}_{(m+n)\times 1} + B_{h\times 1}) \quad (18)$$

Each u^{th} individual term in the output vector is a function of the components of the input vector:

$$\mathbf{y}^{nn}_u = \sum_{j=1}^{h}\left(V(u,j)\sigma\left(\sum_{i=1}^{m+n} W(j,i)\mathbf{x}(i) + B(j,1)\right)\right) \quad (19)$$

As a result, the gradients defined in Eqs. (10) to (12) take the form:

$$\frac{\partial \hat{\mathcal{L}}}{\partial W(j,i)} = \sum_{u=1}^{m+n}\frac{\partial \hat{\mathcal{L}}}{\partial \mathbf{y}^{nn}_u}\frac{\partial \mathbf{y}^{nn}_u}{\partial W(j,i)} \quad (20)$$

$$\frac{\partial \hat{\mathcal{L}}}{\partial V(k,j)} = \sum_{u=1}^{m+n}\frac{\partial \hat{\mathcal{L}}}{\partial \mathbf{y}^{nn}_u}\frac{\partial \mathbf{y}^{nn}_u}{\partial V(k,j)} \quad (21)$$

$$\frac{\partial \hat{\mathcal{L}}}{\partial B(j,1)} = \sum_{u=1}^{m+n}\frac{\partial \hat{\mathcal{L}}}{\partial \mathbf{y}^{nn}_u}\frac{\partial \mathbf{y}^{nn}_u}{\partial B(j,1)} \quad (22)$$

The individual segments can be calculated based on derivative of the error with respect to the outputs and the derivatives of outputs with respect to the weights. Finally, Algorithm 1 describes the Unsupervised Forward Kinematics Neural Network (UFKNN).

Algorithm 1: Unsupervised Forward Kinematics Neural Network (UFKNN)

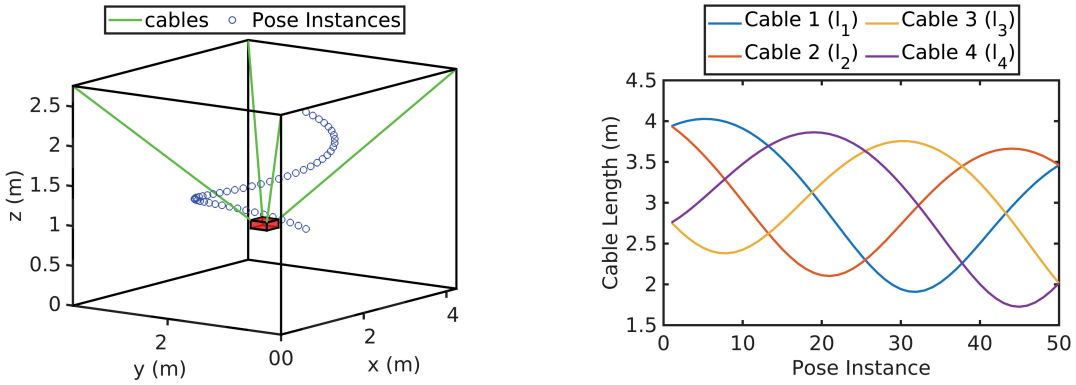
Result: MP pose and cable tension vector
Determine input vector $[l_1,\ldots,l_m,w_{e1},\ldots,w_{en}]$;
$W_{ij} \leftarrow$ *Hidden Layer Weights*;
$B_j \leftarrow$ *Hidden Layer Biases*;
$V_{jk} \leftarrow$ *Output Layer Weights*;
$W_{ij},\ B_j,\ B_{jk} \leftarrow$ *Random Initializing*;
$\sigma() \leftarrow$ *Sigmoid Activation function*;
$\alpha \leftarrow$ **learning rate**;
while *iteration* $\leq$ *max_iterations* **do**
  $inp \leftarrow (A)[l_1,\ldots,l_m,w_{e1},\ldots,w_{en}]$;
  $out[q_1,\ldots,q_n,t_1,\ldots,t_m] \leftarrow$
   $V_{jk}(\sigma(W_{ij} * inp + B_j))$;
  Loss$(\hat{\mathcal{L}}) \leftarrow function(out)$ **from (8)**;
  if $\hat{\mathcal{L}} \leq \hat{\mathcal{L}}_{threshold}$ **then**
    Stop;
  else
    $\delta_{wij},\delta_{bj},\delta_{vjk},\leftarrow \partial\hat{\mathcal{L}}/W_{ij},\partial\hat{\mathcal{L}}/B_j,\partial\hat{\mathcal{L}}/V_{jk}$;
    $W_{ij} \leftarrow W_{ij} - \alpha\delta_{wij}$;
    $B_j \leftarrow B_j - \alpha\delta_{bj}$;
    $V_{jk} \leftarrow V_{jk} - \alpha\delta_{vjk}$;
  end
end

6 SIMULATION RESULTS

Algorithm 1 is implemented on a 6-DOF spatial robot with four cables. The primary motivation of the methodology is to develop some form of re-usability by making a module to learn the FK for a particular manipulator about which the module has no clue beforehand. Thus, the results are analyzed by making the manipulator traverse a planned joint space trajectory and are compared against the results obtained with *lsqnonlin*, a non-linear optimizer using the Levenberg-Marquardt algorithm. It is expected that the weights estimated for one instance do not vary much when the MP moves to the next waypoint. Hence, the time taken to calculate the forward kinematics for each instance possibly takes less time than the previous one. This conclusively makes trajectory tracking faster by only using proprioceptive sensors. The results are presented thereafter considering the rigid cable model.

The spatial 6-DOF suspended underconstrained configuration as shown in Fig. 4a is considered. The setup with a size of 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. As the problem domain increases and becomes more complex, the general intuition that the number of non-linear parameters will increase leads us to choose a higher number of hidden nodes. However, the appropriate learning rate was obtained iteratively by observing the quality of the results. Table 1 gives the parameters that are used in this illustrative example.

The reference path for this case is chosen to be a vertical spiral in the center of the fixed frame, as in Fig. 4a, and the



(a) Model and Cartesian path

(b) Path in cable space

Fig. 4: CDPR configuration and Cartesian reference path.

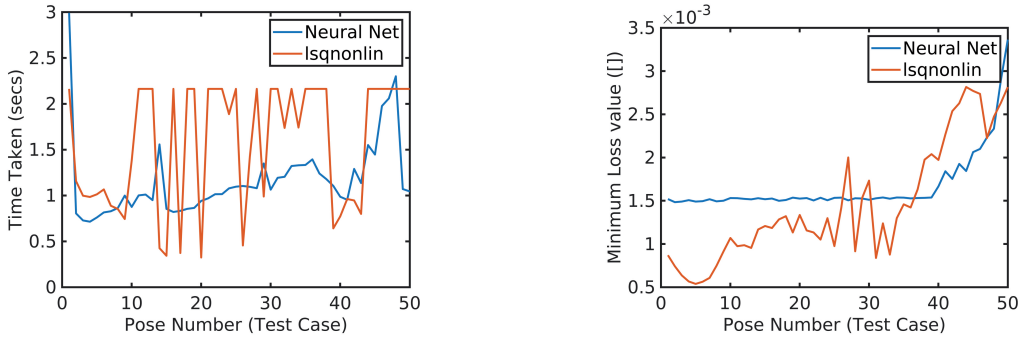
Table 1: Parameters for spatial CDPR-UFKNN algorithm and *lsqnonlin* comparison.

Parameter	Value	Parameter	Value
m_E	7.515 Kg	h	160
r	0.396 m	α	10^{-4}
μ	50	$\hat{\mathcal{L}}_{threshold}$	0.0015
itr_{max}^{lsq}	2×10^5	tol^{lsq}	10^{-10}
E_0	45 GPa	d_0	0.7 mm
A_0	$\pi d_0^2/4$		

corresponding path in joint space is shown in Fig. 4b. Such a path is within the wrench feasible workspace (WFW) of the CDPR and considers the points at the boundaries of the WFW. We base our decision under the assumption that due to increasing non-linearities and difficulty in finding solutions at the boundary of the WFW, the closer the MP to workspace boundary, the less accurate the algorithm. Now, to track the accuracy of the algorithm in finding an appropriate solution, we prepare ground-truth outputs using the Cartesian data and fixed tension vector equal to $\mathbf{t} = [t_1, t_2, t_3, t_4]$, respective values can be seen in Table 1. The wrench to be applied on the MP at every pose instance is calculated by considering the fixed tension vector and the wrench matrix corresponding to that MP pose.

The comparison setup for both the algorithms was based on an iterative rule based on the fact that the right initial solution should speed up the convergence of the algorithms. The ground truth output belonging to the first pose instance was fed into the algorithms. During this initial setup process, while *lsqnonlin* takes only 10 secs to converge, the neural network formulation requires about 1 min to tune the initial network weights. All the calculations were performed using ©MATLAB with CPU computations on an INTEL ®i7-7500U CPU@2.70GHz.

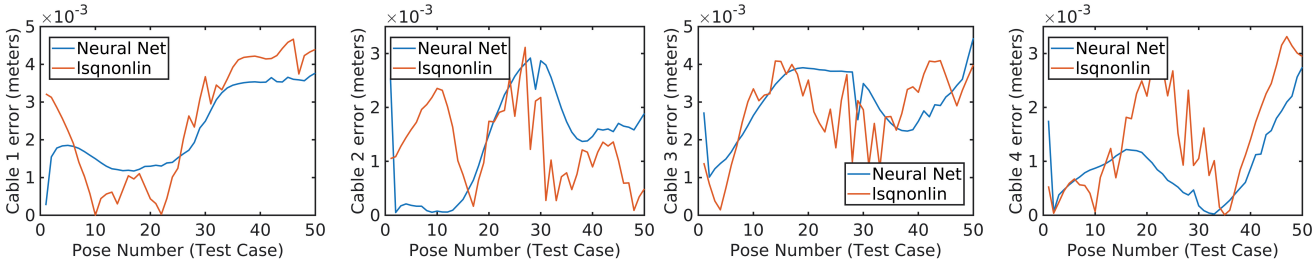
After both the algorithms perfectly fit into the starting pose, the previous outputs from the algorithms were used as initial guesses (for *lsqnonlin*) or weights (for neural network) for each of the next pose instances. The performance and accuracy are measured on a variety of factors as described below. Here, performance is referred to the overall time and minimization objectives which are the primary optimization objectives. Accuracy, on the other hand, refers to the relative deviation of the cable length, cable tensions and Cartesian poses from the desired values.



(a) Time Comparison

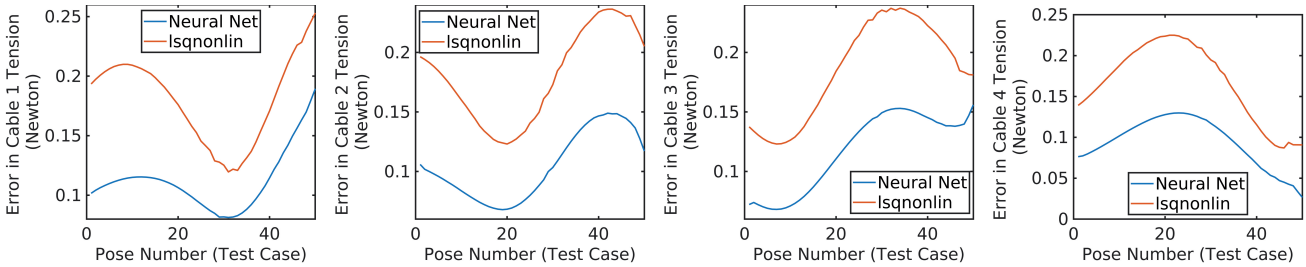
(b) Minimum surrogate objective value

Fig. 5: Comparison of optimization objectives.



(a) Cable 1 length error (b) Cable 2 length error (c) Cable 3 length error (d) Cable 4 length error

Fig. 6: Comparison of cable length errors for spatial CDPR.



(a) Cable 1 tension error (b) Cable 2 tension error (c) Cable 3 tension error (d) Cable 4 tension error

Fig. 7: Comparison of tension errors for spatial CDPR.

6.1 Performance

The overall performance for both the algorithms was compared based on the computation time for each of the 50 pose instances and the extent of minimization of the surrogate objective. The results are shown in Fig. 5. Figure 5a clearly justifies the learning of the neural network. It should be noted that the pre-iterations for the initial pose for 1 min can be considered as the time taken by the algorithm to learn the model-specific kineto-statics. The time taken by the counterpart is relatively higher, which quantitatively corresponds to twice on average. Though *lsqnonlin* gets the privilege of having a very close initial guess (as the waypoints are very close to each other), it has to perform a black-box optimization every time due to its model-free approach. The UFKNN algorithm does not store the guesses but a non-linear approximation of the kineto-static model itself. This model-based behavior plays a vital role in this advantage.

The minimization of the loss function was clipped at the $\hat{\mathcal{L}}_{threshold}$ value for the UFKNN algorithm, and hence the values less than that cannot be shown (even if they might be possible). This threshold value was chosen such that the algorithm marginally reaches the optimization performance of *lsqnonlin* in lesser time. Figure 5b shows the negligible difference between the minimized objective values for both the algorithms.

6.2 Accuracy

As the problem at hand is a coupling of the kinematics and statics of the under-constrained CDPR model, the pose can be changed to satisfy wrench feasibility and decrease the wrench satisfaction error. Thus, it is essential to judge the formulation's accuracy by comparing it with the ground truth reference poses. The metrics for comparison are the cable lengths, cable tensions, and the predicted Cartesian poses. Figures 6 and 7 cumulatively show the accuracy comparison of both the algorithms.

6.2.1 Cable Lengths

The desired cable lengths corresponding to the prescribed path in cable space are a part of the inputs that are given to the algorithms. After predicting the MP pose by the UFKNN algorithm, the obtained cable lengths are calculated through inverse kinematics. The final error is the difference between the calculated and the desired cable lengths. The comparison of these cable errors is given in Fig. 6 for all cables. The maximum deviation in cable length is equal to 5 mm, which is negligible with respect to four-meter frame length. It is apparent that UFKNN reaches the same accuracy as *lsqnonlin* in terms of cable lengths, but in lesser time.

6.2.2 Cable Tensions

The cable tension values were fixed for all MP pose instances, whereas the external wrench, which is also an input to the algorithm, is varied such that the wrench feasibility condition holds. While this is not possible for realization in practical experiments, such conditions can be used to validate the algorithm, which is the main goal of the paper. The outputs of the algorithm correspond to the MP pose and cable tensions. The obtained tensions are compared with the fixed desired tensions, and the error plot is shown in Fig. 7. The maximum absolute error in cable tensions is 0.25 N for *lsqnonlin* and 0.20 N for the UFKNN algorithm. The corresponding mean errors are 0.17 N and 0.10 N respectively.

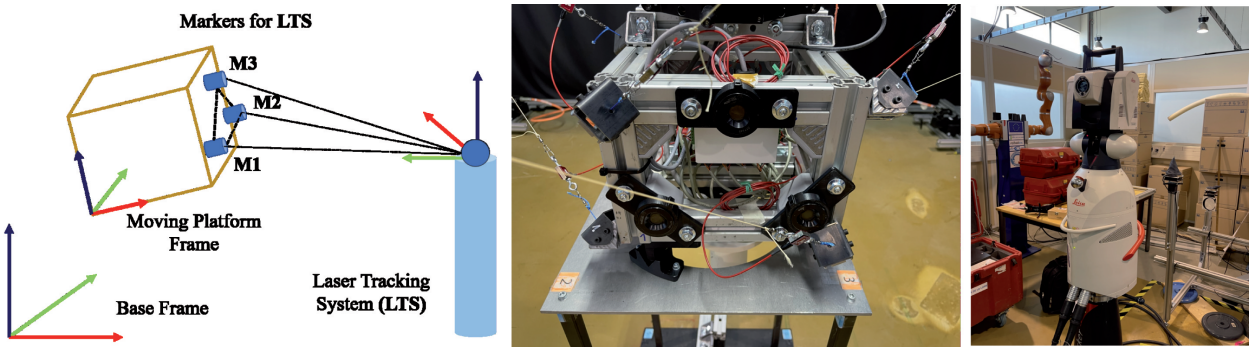


Fig. 8: (left) The hardware setup description and main components. (middle) the three markers on MP along with the stand used as a base position for calculating cable lengths. (right) The laser tracker to measure the cartesian position and orientation of the MP.

The error plot follows a common pattern which might be because of the fixed tension values for all MP pose instances. All these cable tensions usually result in a very high degree of wrench satisfaction i.e. $\frac{\|-\mathbf{W}(\mathbf{s})\ \mathbf{t}+\mathbf{w_e}\|_2}{m_E\ gr} \approx 10^{-3}$.

Additionally, to complete the comparison analysis, the relative deviation of the predicted MP pose from the reference poses is analyzed. The predicted poses are found to be considerably close to real values; however, if observed closely, the relative deviation caused by the UFKNN algorithm is less than that of *lsqnonlin*. The mean Cartesian error for the MP poses of the complete path is 5.7 mm for UFKNN algorithm and 6.7 mm for *lsqnonlin*. The deviations are mostly due to angular displacements, which are desired to be at 0 degrees, but such high precision cannot be expected from numerical computations.

7 EXPERIMENTAL RESULTS

The algorithm described in Section 5 is extended to hardware with the CRAFT [24, 28] CDPR setup after validating the approach with the simulated CDPR model as shown in Section 6. The main hardware of the setup is shown in Fig. 9 which shows the primary components: a PC (equipped with © MATLAB and © CONTROLDESK software), four © PARKER SME60 motors and TPD-M drivers, a © DSPACE DS1007-based real-time controller and four custom-made winch, each comprising a servomotor, a gearbox and a drum. Based on the reading of the motor encoders, the rolled cable lengths are estimated. Here, it is worth to be noted that the calculated length is the unelongated cable length where the effects of elasticity is needed to be incorporated to get the actual cable lengths. We consider that the only external wrench acting on the system is due to the gravity wrench. Using the proposed approach, the approximated Cartesian position, orientation and cable tensions are calculated. To validated the calculated entities, we incorporate a © LEICA AT901-MR Laser Tracker System (LTS) to track the Cartesian MP position with a precision of $\pm 10\mu m + 5\mu m/m$ and © FUTEK LSB205 load/tension sensors in order to calculate the cable tensions with an accuracy of $\pm 2\ N$.

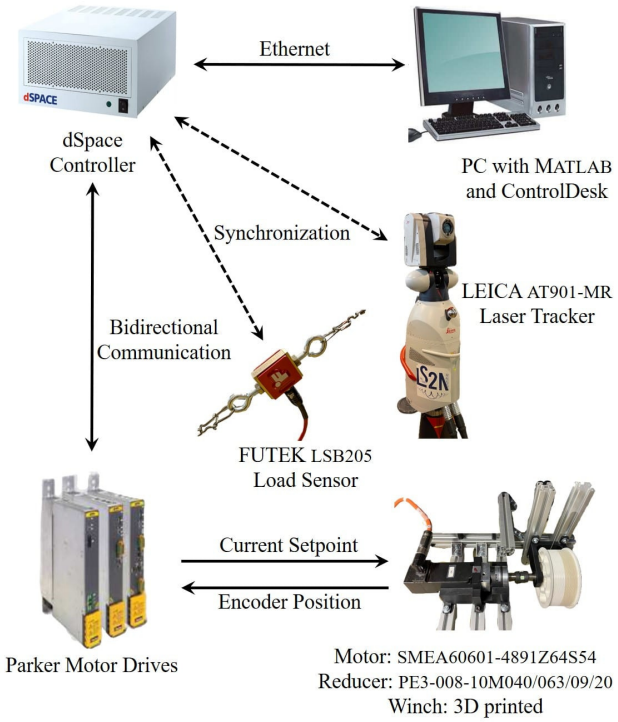


Fig. 9: Communication between main hardware components.

The operation of the LTS is facilitated by assembling three markers on one plane of the moving platform as shown in Fig 8 (left and middle). The Cartesian positions of these markers are calculated based on the frame transformation to correlate the schematic in Fig 8 (left) to Fig 2 in frame $\mathcal{F}_b$. The normal to the plane spanned by the three markers is used to determine the moving-platform orientation. This method is used to obtain the inputs (cable lengths and external wrench) and the ground truth outputs (Cartesian pose and cable tensions) for 10 poses (4 are shown in Fig. 10). The obtained solutions were compared against the actual sensor data and the calculated cable lengths. Considering elasticity in the cable between the cable length between the pulley and

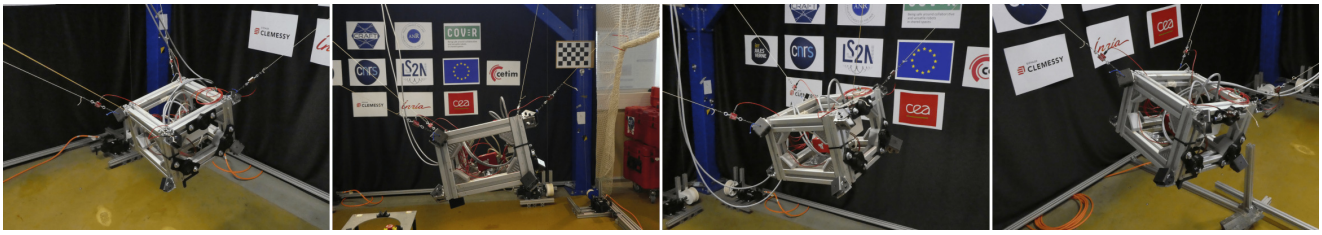


Fig. 10: Four moving platform poses used for validating the proposed methodology experimentally.

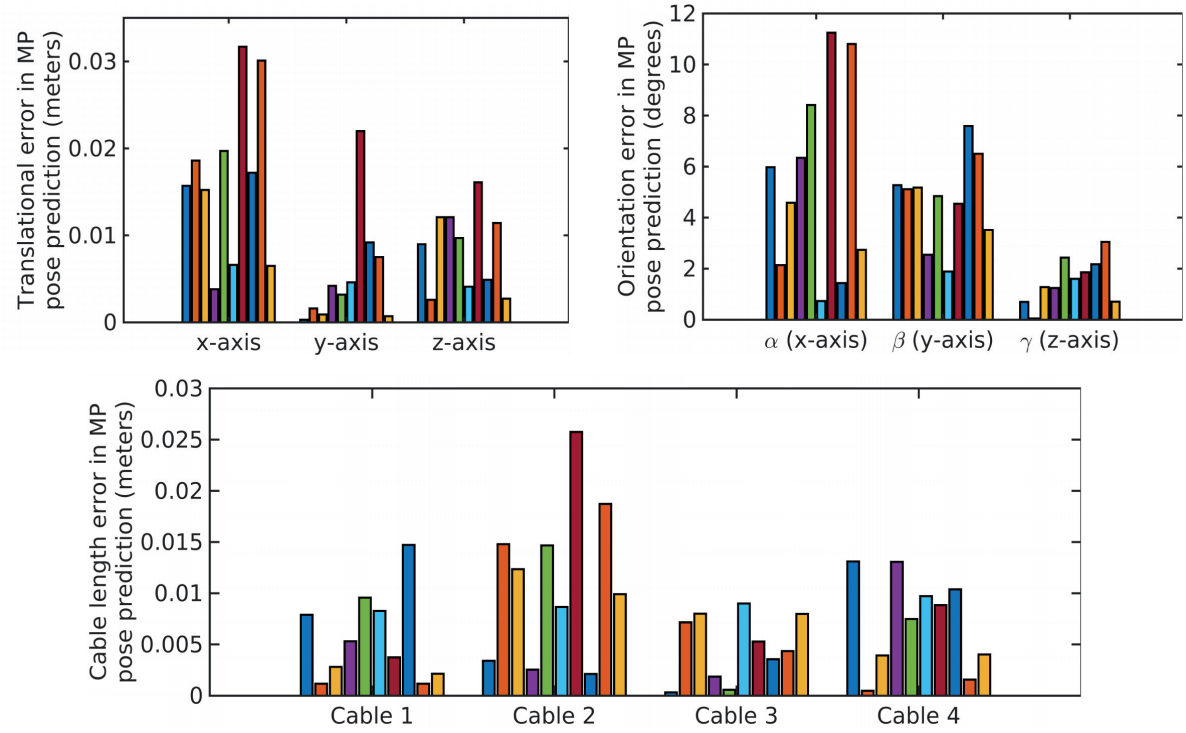


Fig. 11: Error in predicted and laser-tracker calculated translational pose (top left) and orientation (top right) corresponding to the predicted MP poses. (bottom) Error in known and calculated cable lengths for all four cables.

the drum, Eq (3) was modified as follows:

$$\ell_a = \frac{\ell_e - \frac{t_c \, H_{\mathcal{T}_b}}{E_0 \, A_0}}{1 + \frac{t_c}{E_0 \, A_0}} \tag{23}$$

The MP poses measured by the laser tracker, namely, the ground truth, are compared to the MP poses predicted by the NN based FK module. The error bar plots for the 10 poses are shown in Fig. 11 (top). As the system is under-actuated, we observed higher deviations up to 12 degrees in the orientations about the x and y-axes. However, as shown in the Fig. 11 (top right), the orientation error about z-axis is less than 3 degrees. Similarly, we observed less than 3% error in Cartesian position prediction relative to the dimensions of the frame (the MP translational errors in meters are shown). Finally, the maximum deviation in cable lengths is shown in Fig 11 (bottom).

8 DISCUSSION

Artificial Neural Network (ANN) is an elegant approximation tool, however it is typically data hungry and also fails to capture epistemic uncertainties in real systems. Although, previous works have been dependent on extensive data collection and supervised learning, there are still no hardware guaranteeing the use of NN based approaches for CDPRs. In a work like the one presented in this paper, hyperparameter tuning plays a vital role in the performance as overfitting and underfitting are common problems occuring with NNs. Further, the update rule relies on gradient calculation which becomes exponentially difficult with increasing system complexity. Prior works have optimized in the output space of the problem at hand, but the proposed approach optimizes in parameter space without using a huge amount of data. Furthermore, it is worth to be mentioned that we are under an assumption that the local approximation of forward kinematics formulation does not change significantly if consecutive reference nodes are sufficiently close to each other.

We also analyse the real-time performance of the algorithm based on the time taken to solve for the considered pose instant and, the time taken for the moving platform (MP) to traverse between the current and the considered pose instant. If the algorithm is able to solve for the Cartesian pose and tensions as soon as the MP reaches that particular pose, the algorithm can be performed in real time. With a control frequency of 1 KHz, an average velocity of approximately 1 m/s can be achieved. Now, if the algorithm is able to solve for a pose instant at a distance of 1 m from current pose in 1 sec, it is real time. As we observe a similar analysis i.e. calculation time of nearly 1 sec, we can claim an approximate real-time performance. Finally, it is worth noting that the computation time for the next pose increases if the distance to reach is increased. This increased time-demand is compensated by the fact that the MP also takes longer to reach the pose, thus, validating our explanation.

9 CONCLUSION

This paper presented an unsupervised neural network weight adaptation framework according to the kineto-statics of under-constrained Cable-Driven Parallel Robots (CDPRs) with elastic cables and solves the FK problem for a suspended configuration. The approach is validated with both simulated and hardware based results in a 6-DOF spatial CDPR setup and compared with a non-linear least-square optimization-based *lsqnonlin* algorithm. The comparison demonstrates the advantages of the model-based UFKNN algorithm to approximate the robot's kineto-statics compared with model-free black-box optimization techniques currently in practice. This method is beneficial to track paths in cable space, namely in the actuation space. The method is computation time-efficient and works at the same accuracy as other related optimization techniques. Future work includes implementing the proposed algorithm on conducting further experiments to incorporate the overall workspace of the robot and evaluate the capabilities of the approach. Finally, the algorithm will be extended to non-negligible mass and sagging-based cable models to determine the solutions to the forward kinematics of more complex and realistic CDPRs.

Acknowledgements

This work was supported by the ANR CRAFT project, grant ANR-18-CE10-0004 (https://anr.fr/Project-ANR-18-CE10-0004). The research work was motivated by the EquipEx+ TIRREX project, grant ANR-21-ESRE-0015, too. Assistance provided by Mr. Marceau Métillon through the experimentation process is highly appreciated.

References

[1] Barnett, E., and Gosselin, C., 2015. "Large-scale 3d printing with a cable-suspended robot". *Additive Manufacturing,* **7**, pp. 27–44.

[2] Merlet, J.-P., and Daney, D., 2010. "A portable, modular parallel wire crane for rescue operations". In 2010 IEEE International Conference on Robotics and Automation, pp. 2834–2839.

[3] Su, Y. X., Duan, B. Y., Nan, R. D., and Peng, B., 2001. "Development of a large parallel-cable manipulator for the feed-supporting system of a next-generation large radio telescope". *Journal of Robotic Systems,* **18**(11), pp. 633–643.

[4] Homma, K., Fukuda, O., Sugawara, J., Nagata, Y., and Usuba, M., 2003. "A wire-driven leg rehabilitation system: development of a 4-dof experimental system". In Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003), Vol. 2, pp. 908–913 vol.2.

[5] Merlet, J.-P., 2010. "Marionet, a family of modular wire-driven parallel robots". In Advances in Robot Kinematics: Motion in Man and Machine, J. Lenarcic and M. M. Stanisic, eds., Springer Netherlands, pp. 53–61.

[6] Garant, X., Campeau-Lecours, A., Cardou, P., and Gosselin, C., 2018. "Improving the forward kinematics of cable-driven parallel robots through cable angle sensors". In Cable-Driven Parallel Robots, C. Gosselin, P. Cardou, T. Bruckmann, and A. Pott, eds., Springer International Publishing, pp. 167–179.

[7] Dallej, T., Gouttefarde, M., Andreff, N., Hervé, P.-E., and Martinet, P., 2019. "Modeling and vision-based control of large-dimension cable-driven parallel robots using a multiple-camera setup". *Mechatronics,* **61**, pp. 20–36.

[8] Chawla, I., Pathak, P., Notash, L., Samantaray, A., Li, Q., and Sharma, U., 2021. "Effect of selection criterion on the kineto-static solution of a redundant cable-driven parallel robot considering cable mass and elasticity". *Mechanism and Machine Theory,* **156**, p. 104175.

[9] Behzadipour, S., and Khajepour, A., 2005. "A new cable-based parallel robot with three degrees of freedom". *Multibody System Dynamics,* **13**(4), May, pp. 371–383.

[10] Carricato, M., and Merlet, J.-P., 2013. "Stability analysis of underconstrained cable-driven parallel robots". *IEEE Transactions on Robotics,* **29**(1), pp. 288–296.

[11] Carricato, M., 2013. "Direct geometrico-static problem of underconstrained cable-driven parallel robots with three cables". *Journal of Mechanisms and Robotics,* **5**(3).

[12] Berti, A., Merlet, J.-P., and Carricato, M., 2016. "Solving the direct geometrico-static problem of underconstrained cable-driven parallel robots by interval analysis". *The International Journal of Robotics Research,* **35**(6), pp. 723–739.

[13] Merlet, J. P., 2004. "Solving the forward kinematics of a gough-type parallel manipulator with interval analy-

sis". *The International Journal of Robotics Research,* **23**(3), pp. 221–235.

[14] Pott, A., and Schmidt, V., 2015. "On the forward kinematics of cable-driven parallel robots". In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3182–3187.

[15] Collard, J.-F., and Cardou, P., 2013. "Computing the lowest equilibrium pose of a cable-suspended rigid body". *Optimization and Engineering,* **14**(3), Sep, pp. 457–476.

[16] Miermeister, P., Kraus, W., and Pott, A., 2013. *Differential Kinematics for Calibration, System Investigation, and Force Based Forward Kinematics of Cable-Driven Parallel Robots*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 319–333.

[17] Pott, A., 2012. "Influence of pulley kinematics on cable-driven parallel robots". In Latest Advances in Robot Kinematics, J. Lenarcic and M. Husty, eds., Springer Netherlands, pp. 197–204.

[18] Santos, J. C., and Gouttefarde, M., 2021. "A real-time capable forward kinematics algorithm for cable-driven parallel robots considering pulley kinematics". In Advances in Robot Kinematics 2020, J. Lenarčič and B. Siciliano, eds., Springer International Publishing, pp. 199–208.

[19] Tang, H., and Notash, L., 2021. "Neural Network-Based Transfer Learning of Manipulator Inverse Displacement Analysis". *Journal of Mechanisms and Robotics,* **13**(3), 04. 035004.

[20] Notash, L., 2020. "Artificial neural network prediction of deflection maps for cable-driven robots". In Proceedings of the ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC-CIE2020), Vol. 10: 44th Mechanisms and Robotics Conference (MR).

[21] Mishra, U. A., Chawla, I., and Pathak, P. M., 2020. "On determining shortest path in joint space of a cable-driven parallel robot for point-to-point motion". In 2020 28th Mediterranean Conference on Control and Automation (MED), pp. 984–989.

[22] Mishra, U. A., and Caro, S., 2021. "Forward Kinematics for Suspended Under-Actuated Cable-Driven Parallel Robots: A Neural Network Approach". In The ASME 2021 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2021, 45th Mechanisms and Robotics Conference (MR).

[23] Mishra, U. A., and Caro, S., 2021. "Unsupervised neural network based forward kinematics for cable-driven parallel robots with elastic cables". In Cable-Driven Parallel Robots, M. Gouttefarde, T. Bruckmann, and A. Pott, eds., Springer International Publishing, pp. 63–76.

[24] Métillon, M., Cardou, P., Subrin, K., Charron, C., and Caro, S., 2021. "A Cable-Driven Parallel Robot With Full-Circle End-Effector Rotations". *Journal of Mechanisms and Robotics,* **13**(3), 03. 031115.

[25] Baklouti, S., Courteille, E., Caro, S., and DKHIL, M., 2017. "Dynamic and Oscillatory Motions of Cable-Driven Parallel Robots Based on a Nonlinear Cable Tension Model". *Journal of Mechanisms and Robotics,* **9**(6), pp. 1–14.

[26] 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, A. Pott and T. Bruckmann, eds., Springer International Publishing, pp. 293–305.

[27] Rasheed, T., Long, P., and Caro, S., 2020. "Wrench-Feasible Workspace of Mobile Cable-Driven Parallel Robots". *Journal of Mechanisms and Robotics,* **12**(3), Jan., p. 031009.

[28] Etienne, L., Cardou, P., Métillon, M., and Caro, S., 2021. "Design of a Planar Cable-Driven Parallel Crane without Parasitic Tilt". In The ASME 2021 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2021, 45th Mechanisms and Robotics Conference (MR), pp. 1–10.