



HAL
open science

Unsupervised Neural Network Based Forward Kinematics for Cable-Driven Parallel Robots with Elastic Cables

Utkarsh A Mishra, Stéphane Caro

► **To cite this version:**

Utkarsh A Mishra, Stéphane Caro. Unsupervised Neural Network Based Forward Kinematics for Cable-Driven Parallel Robots with Elastic Cables. Marc Gouttefarde; Tobias Bruckmann; Andreas Pott. Cable-Driven Parallel Robots. Proceedings of the 5th International Conference on Cable-Driven Parallel Robots, 104, Springer, pp.63-76, 2021, Mechanisms and Machine Science, 978-3-030-75788-5. 10.1007/978-3-030-75789-2_6 . hal-03338553

HAL Id: hal-03338553

<https://hal.science/hal-03338553v1>

Submitted on 8 Sep 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.

Unsupervised Neural Network based Forward Kinematics for Cable-Driven Parallel Robots with Elastic Cables

Utkarsh A. Mishra¹ and Stéphane Caro²

¹ Indian Institute of Technology Roorkee, Uttarakhand 247667, India
umishra@me.iitr.ac.in

² CNRS, Laboratoire des Sciences du Numérique de Nantes, UMR CNRS 6004,
1, rue de la Noë, 44321 Nantes, France
stephane.caro@ls2n.fr

Abstract. Forward Kinematic (FK) analysis of under-constrained Cable-Driven Parallel Robots (CDPRs) deals with 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 fast forward geometrico-static analysis for CDPRs in a suspended configuration with elastic cables. The formulation determines a non-linear function approximation to model the FK and proves to be efficient in solving for consecutive and close waypoints in a path. The methodology is applied on a simulated six-degree-of-freedom (6-DOF) spatial under-constrained suspended cable-driven parallel robot. Specific comparison results to show the effectiveness of the proposed method in tracking a given path are presented against the results obtained from least-square non-linear optimization.

Keywords: Unsupervised Neural Networks · Elastic Cables · Forward Kinematics.

1 Introduction

Cable-Driven Parallel Robots (CDPRs) belong to a class of parallel manipulators where cables are used to manipulate a Moving Platform (MP) and connects it to a fixed base frame. CDPRs are inherently accompanied with various advantages of low inertia, high payload to weight ratio, and a significantly large workspace as compared to their serial and parallel counterparts. Such advantages have been exploited for various tasks like large scale 3D printing [2], in rehabilitation mechanisms [8], as transfer robots for the elderly [11] and rescue robots [12] as well as in large-scale telescopes [22]. The kinematic analysis for CDPRs play a vital role in realizing various fundamental characteristics consisting of the size of the static workspace, maintaining static equilibrium and achieving appropriate stiffness. However, such an analysis turns out to be more complex as compared to parallel manipulators with rigid links, driven by the fact that cables can only apply unilateral forces, i.e., only pull the MP and not push it. Thus, constraining its capability to account for static equilibrium. Furthermore, such an analysis

is critically dependent on the configuration of the complete manipulator as well as on the nature of the cables used.

The robot configuration is generally based on the number of cables modulating the intended degrees of freedom (DOFs) and on the availability of the cables to pull the MP in both upward and downward directions. In this context, we are particularly interested in suspended configurations i.e. when cables can only pull the MP upward. With the reduction in the number of cables, the system becomes under constrained and typically has n -DOFs controlled by $m (< n)$ cables. Thus, 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 [5]. For such systems, even when the cable lengths are fixed, the MP still moves and behaves according to the external wrench. Accordingly, the geometrico-static [4] problem becomes more complex and may have several solutions. The cable lengths tend to be the most feasible proprioceptive measurement for CDPRs as compared to cable tensions or orientations. 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 [6]. In this paper, the cables are supposed to be massless and elastic.

Solving the forward kinematics 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 [3, 10] 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 [18], finding the lowest equilibrium pose [7] or minimizing cable tensions [14]. Iterative (usually, Levenberg-Marquadt) algorithms have found themselves useful in locally minimizing pose errors [18], using Hessian matrix to construct convex problem [17] and Jacobian matrix to solve the linearized approximation of the FK problem [21] 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 [23], and cable [16] robots. Furthermore, as most controllers directly work on joint space applying direct control over motor winches to modulate cable lengths, path planning in joint space [15] 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.

2 Contributions and Paper Organization

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, 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 simulated spatial CDPR (with 6-DOFs, 4 cables). The overall performance for the proposed algorithm and *lsqnonlin* is compared based on the time efficiency and accuracy.

The model of the manipulator and the associated nomenclature are discussed in Section 3. Section 4 describes the elastic cable model used Section 5 formulates the geometrico-static problem of the CDPR under study. The unsupervised neural network is described in Section 6. Section 7 gives the simulation results obtained using the proposed approach. Those results are discussed in Section 8. Conclusions and future work are drawn in Section 9.

3 Parametrization

We consider a CDPR consisting of m cables and n -DOFs. Its i^{th} closed-loop being represented in Fig. 1. The two frames of reference, namely, \mathcal{F}_b and \mathcal{F}_p are attached to the fixed base frame and the MP frame respectively. \mathbf{a}_i represents the Cartesian coordinates vector of exit point A_i on the fixed frame, expressed in \mathcal{F}_b . The cable attachment with the MP, B_i is defined by the Cartesian coordinate vector \mathbf{b}_i expressed in \mathcal{F}_p . The MP pose is defined by the vector \mathbf{p} from O to P and the corresponding orientation of the MP w.r.t. the fixed frame is defined by the rotation matrix ${}^b\mathbf{R}_p \in SO(3)$ from \mathcal{F}_b to \mathcal{F}_p .

The i^{th} loop closure equation is expressed as:

$${}^b\mathbf{a}_i + \ell_i^b \mathbf{u}_i - {}^b\mathbf{R}_p^p \mathbf{b}_i - {}^b\mathbf{p} = \mathbf{0}_3 \quad (1)$$

\mathbf{u}_i is the unit vector along the i th cable pointing from A_i to B_i and $\mathbf{0}_3$ is a three-dimensional zero vector. The subscript b (p , resp.) means that the corresponding vector is expressed in frame \mathcal{F}_b (\mathcal{F}_p , resp.).

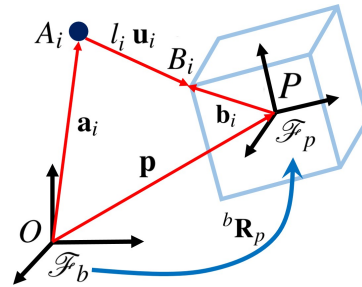


Fig. 1: i^{th} -closed loop of the CDPR

The pre-planned path used for verification purpose is chosen to be within robot workspace (based on CRAFT [13] with an upper and lower bound on cable tensions equal to $\bar{t} = 86$ N and $\underline{t} = 1$ N respectively). The MP mass is named m_E .

4 Elastic Cable Model

We consider the mass-less elastic cable model wherein the cables are modelled as linear springs instead of rigid links. Their elongation is a function of their elasticity and the 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 [1]. However, for simplicity, we consider solving for the final elongation and use the stress-strain relationship [6]. 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:

$$\Delta\ell = t_c \frac{\ell_a}{E_0 A_0} \quad (2)$$

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:

$$\ell_a = \frac{\ell_e}{1 + \frac{t_c}{E_0 A_0}} \quad (3)$$

5 Geometrico-Static Modeling

After realizing the first coupling between kinematics and statics based on elastic cable model in Section 4, 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.

5.1 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, $\mathbf{t} = [t_1, t_2, \dots, t_m]$, satisfying the Newton-Euler equation given by,

$$-\mathbf{W} \mathbf{t} + \mathbf{w}_e = \mathbf{0}_n \quad (4)$$

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

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

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 $r^2 = 1/m \sum_{i=1}^m \|\mathcal{F} \mathbf{b}_i\|_2^2$ (refer [19, 20]).

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 [9], the tension distribution obtained from Eq. (4) holds true for our case.

5.2 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),

$$\ell_i^b \mathbf{u}_i = {}^b\mathbf{p} + {}^b\mathbf{R}_p {}^p\mathbf{b}_i - {}^b\mathbf{a}_i \quad (6)$$

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

where $\boldsymbol{\ell}_a = [\ell_{a1}, \ell_{a2}, \dots, \ell_{am}]$ 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). $\|\cdot\|_2$ 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,

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

5.3 Forward Kinematics

The forward kinematics (FK) of a cable-driven manipulator consists of obtaining the platform 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:

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

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

$$\zeta : \hat{\boldsymbol{\ell}}_a \in \mathbb{R}^m \mapsto \mathbf{s} \in \mathbb{R}^n$$

such that

$$\zeta(\hat{\boldsymbol{\ell}}_a) = \arg \min_{\mathbf{s}} \left\| \hat{\boldsymbol{\ell}}_a - \xi(\mathbf{s}, \mathbf{t}) \right\|_2$$

6 Neural-Network Formulation

The components of robot kinetostatics 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 more 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 $\mathbf{x} = [\hat{\ell}_a^T \mathbf{w}_e^T]^T$. 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:

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

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

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

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:

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

where $g = 9.81m/s^2$ 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.

6.1 Network Architecture

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

Consider a Neural Network (NN) consisting of two hidden layers with weights W of size $h \times (m+n)$, V of size $(m+n) \times h$ and biases B of size $h \times 1$, respectively. Here, 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:

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

The back-propagation error is defined using \mathcal{L} from Eq. (8). From SGD formulation, the weights and biases are optimized with reference to calculated change in weights δ_W of size $h \times (m+n)$, δ_V of size $(m+n) \times h$ and biases δ_B of size $h \times 1$ given by:

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

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

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

The updated weights and biases (\square_{new}) are obtained by using the above δ_{\square} values obtained for the corresponding current weights and biases ($\square_{current}$). Here, \square represents all of W , V (weights) and B (biases). The neural network parameter 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} .

6.2 Derivatives and Gradient Descent

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}^m from the input \mathbf{x} , such that,

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

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

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

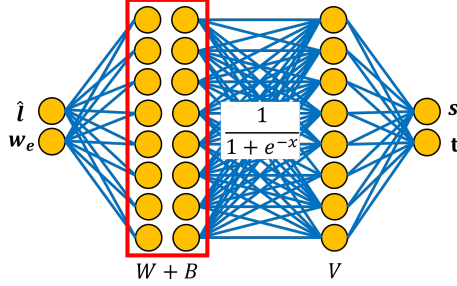


Fig. 2: Neural network architecture and activation function

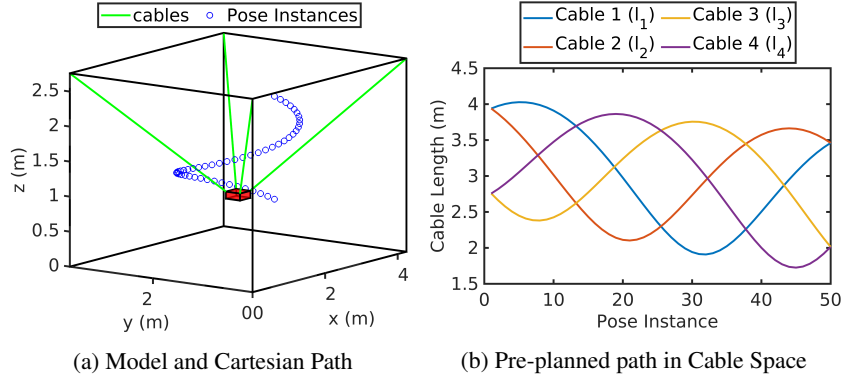


Fig. 3: CDPR configuration and reference path

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

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

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

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

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. Hereafter, we refer this complete algorithm as the Unsupervised Forward Kinematics Neural Network (UFKNN).

7 Simulation Results

UFKNN algorithm is implemented on a 6-DOF spatial robot with four cables. The primary motivation of the methodology is to develop re-usability by making a module to learn the FK for a particular manipulator. Thus, the results are analyzed by making the manipulator follow a planned path in the joint space 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, decreasing the time taken to compute FK for each instance. This makes tracking faster using only proprioceptive sensors. The results are present thereafter considering the elastic cable model.

The suspended underconstrained configuration as shown in Fig. 3a 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 moving platform of size 0.28 m x 0.28 m x 0.20 m. The appropriate learning rate (α) was obtained

iteratively by observing the quality of the results. Table 1 gives the parameters that are used in this illustrative example.

Table 1: Parameters for spatial CDPR-UFKNN algorithm, *lsqnonlin* and elastic cable model

Parameter	Value	Parameter	Value	Parameter	Value
m_E	5.6 Kg	t_1	15 N	itr_{max}^{lsq}	5×10^4
r	0.396 m	t_2	18 N	tol^{lsq}	10^{-8}
μ	50	t_3	25 N	E_0	70 GPa
h	64	t_4	20 N	d_0	5 mm
α	10^{-5}	$\mathcal{L}_{threshold}$	0.0001	A_0	$\pi d_0^2/4$

The reference path for this case is chosen to be a vertical spiral in the center of the fixed frame, as in Fig. 3a, and the corresponding path in joint space is shown in Fig. 3b. Such a path considers the extreme points at varying heights. 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 $\mathbf{t} = [t_1, t_2, t_3, t_4]$, cable tension values being given 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 4 secs to converge, the neural network formulation requires about 1.5-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 computational time objective which is the primary objective. Accuracy, on the other hand, refers to the relative deviation of the cable lengths, cable tensions and Cartesian MP poses from the desired values.

As a performance measure, the overall performance for both the algorithms

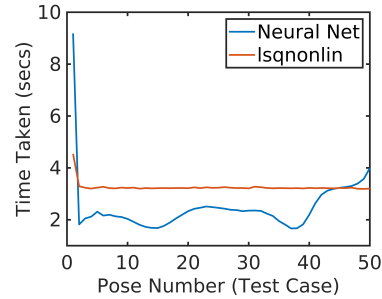


Fig. 4: Computational time efficiency comparison

was compared based on the computation time for each of the 50 pose instances. Figure 4 clearly justifies the learning of the neural network. It should be noted that the pre-iterations for the initial pose for 1.5 min can be considered as the time taken by the algorithm to learn the model-specific kineto-statics. 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, and takes nearly the same time (≈ 3.5 secs). The UFKNN algorithm does not store the guesses, but a non-linear approximation of the kineto-static model itself and progressively decreases the computational time. This is to be noted that the test conducted in the presented work maintains constant tension targets for all the test poses in the reference path. With this consideration, the final points on the reference path lie at the boundary of the wrench feasible workspace and hence the computation time increases in the later stages.

8 Discussion

As we deal with coupled 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,

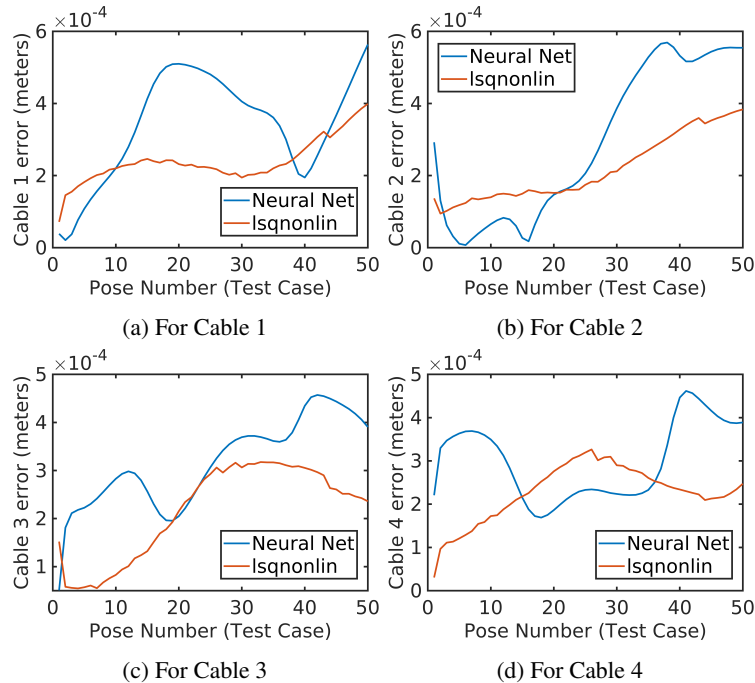


Fig. 5: Comparison of cable length errors

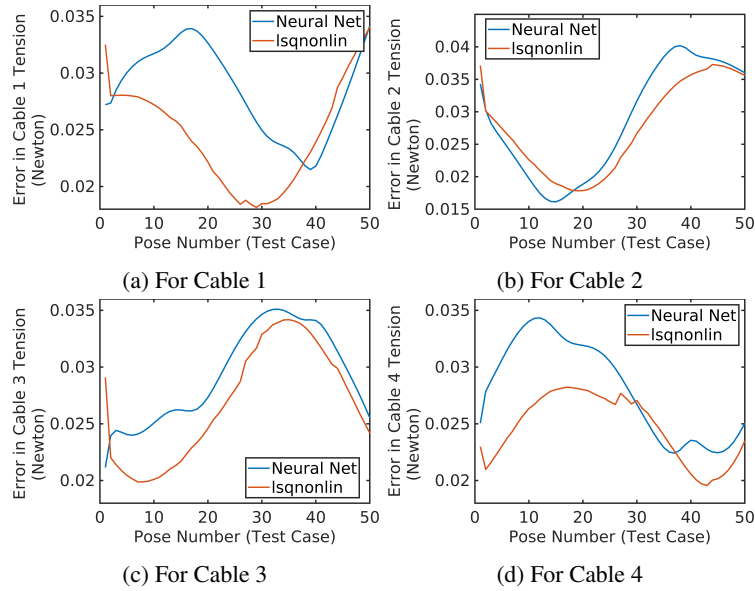


Fig. 6: Comparison of cable tension errors

cable tensions, and the predicted MP poses. Figures 5 and 6 show the accuracy comparison of both the algorithms.

For cable lengths, the desired cable lengths obtained from the planned path are a part of the inputs to the algorithms. After predicting the MP pose by the UFKNN algorithm, the obtained cable lengths are calculated through IK. The final error is the difference between the calculated and the desired cable lengths. The comparison of these cable errors is given in Fig. 5 for all cables. The maximum deviation in cable length is equal to 0.6 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.

For 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. The outputs of the algorithm correspond to the MP pose and cable tensions. The error between the obtained and fixed desired tensions is shown in Fig. 6. The maximum absolute error in cable tensions is 0.035 N for *lsqnonlin* and 0.04 N for the UFKNN algorithm. The corresponding mean errors are 0.020 N and 0.025 N respectively. All these cable tensions usually result in a very high degree of wrench satisfaction i.e. $\frac{\|-\mathbf{W}(\mathbf{s}) \mathbf{t} + \mathbf{w}_e\|_2}{m_E g} \approx 10^{-5}$.

For predicted MP poses, the mean Cartesian error for the MP poses of the complete path is 0.6 mm for UFKNN algorithm and 0.5 mm for *lsqnonlin*. The deviations are mostly due to angular displacements, which should be null, but such a high precision cannot be expected from numerical computations.

9 Conclusions and Future Work

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 simulated 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 hardware and conducting simulated experiments to fully leverage its capabilities. 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.

Acknowledgement This work was supported by the ANR CRAFT project, grant ANR-18-CE10-0004, <https://anr.fr/Project-ANR-18-CE10-0004>

References

1. Baklouti, S., Courteille, E., Caro, S., Dkhil, M.: Dynamic and oscillatory motions of cable-driven parallel robots based on a nonlinear cable tension model. *Journal of Mechanisms and Robotics* (2017). <https://doi.org/10.1115/1.4038068>
2. Barnett, E., Gosselin, C.: Large-scale 3D printing with a cable-suspended robot. *Additive Manufacturing* (2015). <https://doi.org/10.1016/j.addma.2015.05.001>
3. Berti, A., Merlet, J.P., Carricato, M.: Solving the direct geometrico-static problem of underconstrained cable-driven parallel robots by interval analysis. *International Journal of Robotics Research* (2016). <https://doi.org/10.1177/0278364915595277>
4. Carricato, M.: Direct geometrico-static problem of underconstrained cable-driven parallel robots with three cables. *Journal of Mechanisms and Robotics* **5**(3) (2013). <https://doi.org/10.1115/1.4024293>
5. Carricato, M., Merlet, J.P.: Stability Analysis of Underconstrained Cable-Driven Parallel Robots. *IEEE Transactions on Robotics* pp. 1–1 (2012). <https://doi.org/10.1109/vr.2012.6180941>
6. Chawla, I., Pathak, P.M., Notash, L., Samantaray, A.K., Li, Q., Sharma, U.K.: 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* (2021). <https://doi.org/10.1016/j.mechmachtheory.2020.104175>
7. Collard, J.F., Cardou, P.: Computing the lowest equilibrium pose of a cable-suspended rigid body. *Optimization and Engineering* (2013). <https://doi.org/10.1007/s11081-012-9191-5>
8. Homma, K., Fukuda, O., Sugawara, J., Nagata, Y., Usuba, M.: A wire-driven leg rehabilitation system: Development of a 4-DOF experimental system. In: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM* (2003). <https://doi.org/10.1109/AIM.2003.1225463>

9. Maier, T., Woernle, C.: Inverse Kinematics for an Underconstrained Cable Suspension Manipulator. In: *Advances in Robot Kinematics: Analysis and Control* (1998). https://doi.org/10.1007/978-94-015-9064-8_10
10. Merlet, J.P.: Solving the forward kinematics of a gough-type parallel manipulator with interval analysis. *International Journal of Robotics Research* (2004). <https://doi.org/10.1177/0278364904039806>
11. Merlet, J.P.: MARIONET, A Family of Modular Wire-Driven Parallel Robots. In: *Advances in Robot Kinematics: Motion in Man and Machine* (2010). https://doi.org/10.1007/978-90-481-9262-5_6
12. Merlet, J.P., Daney, D.: A portable, modular parallel wire crane for rescue operations. In: *Proceedings - IEEE International Conference on Robotics and Automation* (2010). <https://doi.org/10.1109/ROBOT.2010.5509299>
13. Metillon, M., Cardou, P., Subrin, K., Charron, C., Caro, S.: A Cable-Driven Parallel Robot with Full-Circle End-Effector Rotations. *Journal of Mechanisms and Robotics* (2021). <https://doi.org/10.1115/1.4049631>
14. Miermeister, P., Kraus, W., Pott, A.: Differential kinematics for calibration, system investigation, and force based forward kinematics of cable-driven parallel robots. In: *Mechanisms and Machine Science* (2013). https://doi.org/10.1007/978-3-642-31988-4_20
15. Mishra, U.A., Chawla, I., Pathak, P.M.: 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 2020* (2020). <https://doi.org/10.1109/MED48518.2020.9183198>
16. Notash, L.: Artificial neural network prediction of deflection maps for cable-driven robots. In: *Proceedings of the ASME Design Engineering Technical Conference* (2020). <https://doi.org/10.1115/DETC2020-22513>
17. Pott, A.: Influence of pulley kinematics on cable-driven parallel robots. In: *Latest Advances in Robot Kinematics* (2012). https://doi.org/10.1007/978-94-007-4620-6_25
18. Pott, A., Schmidt, V.: On the forward kinematics of cable-driven parallel robots. In: *IEEE International Conference on Intelligent Robots and Systems* (2015). <https://doi.org/10.1109/IROS.2015.7353818>
19. Rasheed, T., Long, P., Caro, S.: Wrench-Feasible Workspace of Mobile Cable-Driven Parallel Robots. *Journal of Mechanisms and Robotics* **12**(3), 031009 (Jan 2020). <https://doi.org/10.1115/1.4045423>, <https://hal.archives-ouvertes.fr/hal-02379201>
20. Ruiz, A.L.C., Caro, S., Cardou, P., Guay, F.: ARACHNIS: Analysis of robots actuated by cables with handy and neat interface software. In: *Mechanisms and Machine Science* (2015). https://doi.org/10.1007/978-3-319-09489-2_1
21. Santos, J.C., Gouttefarde, M.: A Real-Time Capable Forward Kinematics Algorithm for Cable-Driven Parallel Robots Considering Pulley Kinematics (2021). https://doi.org/10.1007/978-3-030-50975-0_25
22. Su, Y.X., Duan, B.Y., Nan, R.D., Peng, B.: Development of a large parallel-cable manipulator for the feed-supporting system of a next-generation large radio telescope. *Journal of Robotic Systems* (2001). <https://doi.org/10.1002/rob.8102>
23. Tang, H., Notash, L.: Neural network based transfer learning of manipulator inverse displacement analysis. In: *Proceedings of the ASME Design Engineering Technical Conference* (2020). <https://doi.org/10.1115/DETC2020-22508>