



HAL
open science

A Real-Time Capable Forward Kinematics Algorithm for Cable-Driven Parallel Robots Considering Pulley Kinematics

João Cavalcanti Santos, Marc Gouttefarde

► **To cite this version:**

João Cavalcanti Santos, Marc Gouttefarde. A Real-Time Capable Forward Kinematics Algorithm for Cable-Driven Parallel Robots Considering Pulley Kinematics. ARK 2020 - 17th International Symposium on Advances in Robot Kinematics, Jun 2020, Ljubljana, Slovenia. pp.199-208, 10.1007/978-3-030-50975-0_25 . lirmm-02749091

HAL Id: lirmm-02749091

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-02749091v1>

Submitted on 7 Jun 2020

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.

A Real-Time Capable Forward Kinematics Algorithm for Cable-Driven Parallel Robots Considering Pulley Kinematics

João Cavalcanti Santos and Marc Gouttefarde

Abstract A real-time capable Forward Kinematics (FK) algorithm for Cable-Driven Parallel Robots (CDPRs) considering the pulley kinematics is proposed. The algorithm applies iteratively QR decomposition to solve a linearized version of the least squares problem representing the FK. Differential kinematics delivers an analytical expression for the Jacobian matrix of CDPRs considering the pulley kinematics. This Jacobian matrix is used to construct the linearization of the FK problem. Experimental and numerical results address the convergence capabilities of the proposed algorithm.

Key words: Cable-Driven Parallel Robots, Forward Kinematics, Pulley Kinematics

1 Introduction

Cable Driven Parallel Robots (CDPRs) may present several advantages compared to rigid links parallel robots such as large workspace, high payload capabilities, reduced moving mass and high dynamic capabilities. Therefore, CDPRs have been used as flight simulators [12], robotic cranes [2], large radio telescopes [13], rehabilitation devices [19], among others.

In spite of these advantages, positioning precision may be one prominent drawback of CDPRs. In comparison to rigid-link robots, the computation of the end-effector pose relying solely on the joint (winch) positions may be troublesome and inaccurate [10]. A solution for this issue would be the application of an external measurement system capturing the pose of the platform. This can be done, for instance, using a motion-capture system Bonita developed by Vicon [4] or a multi-cameras setup [5]. Nevertheless, the implementation of such measurement systems within uncontrolled environments may lead to issues related to occlusions

The authors are with LIRMM, Univ. Montpellier, CNRS, Montpellier, France e-mail: joao.cavalcanti-santos@lirmm.fr, marc.gouttefarde@lirmm.fr

and lighting variations. Furthermore, these solutions also depend on additional sensors, which may be costly.

Therefore, several studies sought a reliable and accurate algorithm able to compute the CDPR platform pose for a given set of winch positions, i.e., Forward Kinematic (FK) Algorithms. The FK problem may be solved with interval analysis [1, 8]. This approach allows one to find all the possible FK problem solutions, taking into account that some cables may be slack. Even though this method is considered mathematically complex, real-time capability has been presented [9]. Iterative numerical schemes are another typical approach [3, 11, 14]. In general, the FK problem is formulated as a nonlinear optimization problem. The iterative algorithm (usually, Levenberg-Marquadt) delivers a platform pose that locally minimize modeling errors, taking the winch positions as input. To this end, a differential kinematic model is necessary.

The previous works discussed above do not address the influence of the pulley kinematics. Indeed, Pott showed in [15] that relevant errors may be obtained assuming fixed proximal anchor points. In [17], the FK problem considering the pulley kinematics is solved but the equations are not derived explicitly. In this paper, these equations are obtained (in section 3) and QR decomposition is used to iteratively solve the linearized version of the nonlinear FK problem (section 4). This linearization is based on the differential kinematic model obtained with the differentiation of the Inverse Kinematics (IK) model (section 2). Therefore, the contributions of this paper may be summarized as follows: (i) an analytical expression for the differential kinematics of CDPRs taking into account the pulley kinematics and (ii) the solution and implementation of the resultant least squares problem using QR decomposition on a CDPR prototype operated with industry-grade control software. Section 5 presents experimental results that demonstrate its applicability in a industrial real-time environment and numerical studies that address its convergence capability.

2 Inverse Kinematic Model

A six degree-of-freedom (DoF) CDPR is considered but the proposed algorithm can be applied to CDPRs with fewer DoFs. The platform is driven by m cables. The main variables and parameters of the kinematic model are introduced in Figure 1. The position of each cable attachment point B_i on the platform is given by the vector \mathbf{b}_i , with $i = 1, \dots, m$. Each point A_i is a fixed point which lies on the circumference of pulley i . Each pulley i is able to rotate around the line passing through A_i tangent to the pulley circumference. The constant position of A_i is given by vector \mathbf{a}_i in the fixed reference frame.

The platform pose is given by the vector $\mathbf{x} = [\mathbf{p}^T \ \boldsymbol{\varphi}^T]^T$. The position of the platform is $\mathbf{p} = [p_x \ p_y \ p_z]^T$. Its orientation is represented by the vector $\boldsymbol{\varphi} = [\alpha \ \beta \ \gamma]^T$. The elements of this vector are Euler angles so that $\mathbf{b}_i = \mathbf{p} + \mathbf{R}_z(\gamma) \mathbf{R}_y(\beta) \mathbf{R}_x(\alpha) \mathbf{b}_{pi}$, where \mathbf{b}_{pi} is the vector of the platform attachment point i written in the coordinate system \mathcal{O}_p , attached to the platform, and \mathbf{R}_x , \mathbf{R}_y and \mathbf{R}_z are the rotation matrices

around x , y and z axis, respectively. The vector of cable lengths is $\mathbf{l} = [l_1 \dots l_m]^T$, where l_i is the length of cable i and the cable elongations are neglected. The IK model gives the motor positions for a given pose of the platform, i.e. compute \mathbf{l} for a given \mathbf{x} . The remainder of this section recalls the IK considering the influence of the pulley geometry. Similar models were notably presented in [3, 15].

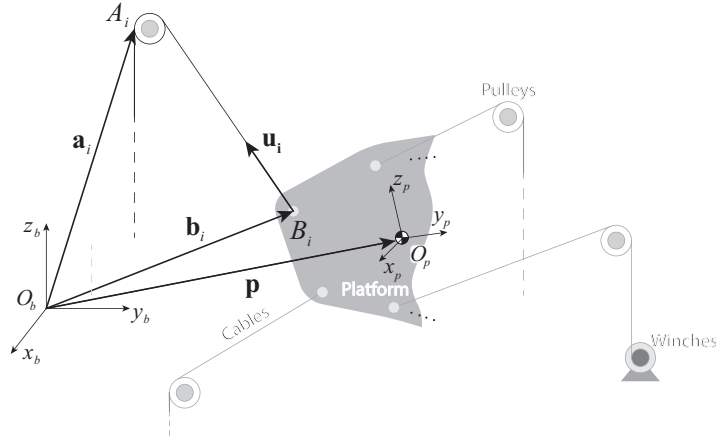


Fig. 1 Main geometric notation

The studied pulley geometry is illustrated in Figure 2. One drawing pulley i among the m CDPR pulleys is considered and subscript i is dropped in the remainder of this section. Consider the definition of coordinate systems $\mathbf{E}_0 = [\mathbf{e}_1^0 \ \mathbf{e}_2^0 \ \mathbf{e}_3^0]$ and $\mathbf{E}_1 = [\mathbf{e}_1^1 \ \mathbf{e}_2^1 \ \mathbf{e}_3^1]$. The pulley may rotate over the fixed axis \mathbf{e}_3^0 (with angle ρ). \mathbf{E}_0 is any fixed coordinate system with \mathbf{e}_3^0 aligned with the rotation axis of the pulley. \mathbf{E}_1 rotates with the pulley and $\mathbf{e}_3^1 = \mathbf{e}_3^0$. \mathbf{E}_1 is thus pose dependent and its orientation is computed in the inverse kinematics.

Define $\mathbf{v} = \mathbf{b}_i - \mathbf{a}_i$, written in the CDPR fixed reference frame. Vector \mathbf{a}_i is constant for a given CDPR and \mathbf{b}_i depends on the platform pose. Vector \mathbf{v} can be written in the coordinate system \mathbf{E}_0 as $\mathbf{v}^0 = \mathbf{E}_0^T \mathbf{v}$. The angle ρ shown in Figure 2 is obtained as $\text{atan2}(v_2^0, v_1^0)$, with v_1^0 and v_2^0 the first and second components of \mathbf{v}^0 . Therefore, the coordinate system \mathbf{E}_1 is obtained as $\mathbf{E}_1 = \mathbf{E}_0 \mathbf{R}_z(\rho)$, where \mathbf{R}_z is the rotation matrix around the z axis.

The vector \mathbf{v}^1 written in the coordinate system \mathbf{E}_1 is $\mathbf{v}^1 = \mathbf{E}_1^T \mathbf{v} = [v_1^1 \ 0 \ v_3^1]^T$. The component $v_2^1 = 0$ because the coordinate system is aligned to the pulley orientation.

Analysing the geometry presented in Figure 2, the following system of trigonometric equations is obtained

$$\mathbf{v}^{1r} = \begin{bmatrix} v_1^1 \\ v_3^1 \end{bmatrix} = l_f \begin{bmatrix} \sin \phi \\ \cos \phi \end{bmatrix} + r_p \begin{bmatrix} 1 - \cos \phi \\ \sin \phi \end{bmatrix} \quad (1)$$

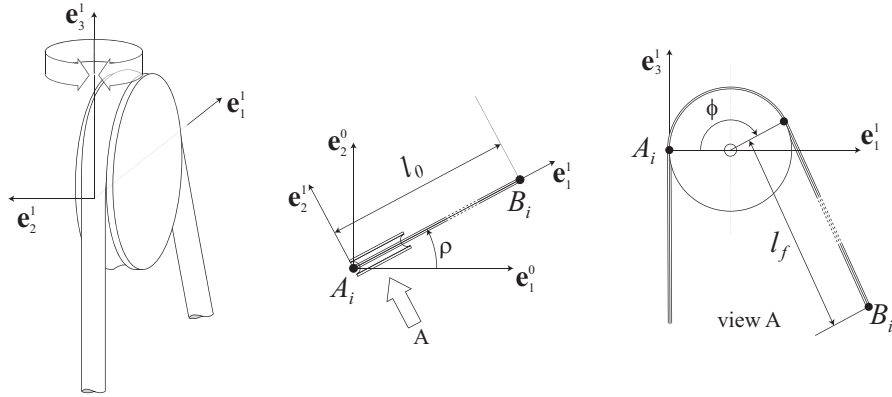


Fig. 2 Pulley geometry and coordinate systems

where r_p is the primitive radius of the pulley and l_f is the length depicted in the Figure 1. Eliminating the variable l_f , the following equation is obtained:

$$v_3^1 \sin \phi + (r_p - v_1^1) \cos \phi - r_p = 0 \quad (2)$$

The following transformations can be used to simplify (2):

$$\sin \phi = \frac{2 \tan(\phi/2)}{1 + \tan^2(\phi/2)}, \quad \cos \phi = \frac{1 - \tan^2(\phi/2)}{1 + \tan^2(\phi/2)} \quad (3)$$

which leads to

$$\tan^2(\phi/2) (v_1^1 - 2r_p) + \tan(\phi/2) (2v_3^1) - v_1^1 = 0 \quad (4)$$

Equation (4) can be solved as a quadratic equation in $\tan(\phi/2)$. Then, using the inverse tangent function, two solutions are obtained in the interval $[-\pi, \pi]$:

$$\phi_1 = 2 \operatorname{atan} \left(\frac{\sqrt{v_1^{12} - 2v_1^1 r_p + v_3^{12} - v_3^1}}{v_1^1 - 2r_p} \right) \quad (5)$$

$$\phi_2 = 2 \operatorname{atan} \left(\frac{-\sqrt{v_1^{12} - 2v_1^1 r_p + v_3^{12} - v_3^1}}{v_1^1 - 2r_p} \right) \quad (6)$$

If $v_1^1 > 2r_p$ (typical situation), $\phi_1 > 0$ and $\phi_2 < 0$. In this case the angle respecting the geometry of the problem is ϕ_1 . Otherwise, ϕ_2 should be taken.

The length l_f may be easily obtained from (1) once ϕ is known. The cable length l_i is given by

$$l_i = r_p \phi + l_f + l_{ci} \quad (7)$$

Forward Kinematics of CDPRs

5

where l_{ci} is a constant additional length. This constant additional length may take into account the total distance between fixed pulleys for instance.

3 Differential Kinematics

The Jacobian matrix \mathbf{J} is used in the solution of the FK problem since it relates $\delta \mathbf{l}$ to $\delta \mathbf{x}$ as

$$\delta \mathbf{l} = \mathbf{J} \delta \mathbf{x} \quad (8)$$

where $\delta(\cdot)$ denotes the differential variation of (\cdot) . This section presents the analytical expression of the matrix \mathbf{J} obtained from the equations in section 2. As in section 2, subscript i is dropped in the remainder of this section. The vector $\mathbf{v} = \mathbf{b}_i - \mathbf{a}_i$ is computed as

$$\mathbf{v} = \mathbf{p} + \mathbf{R}(\varphi) \mathbf{b}_{pi} - \mathbf{a}_i \quad (9)$$

Differentiating this expression with respect to the platform pose, the following equation is obtained

$$\delta \mathbf{v} = \underbrace{\left[\mathbf{I}_3 \frac{\partial}{\partial \varphi} (\mathbf{R}(\varphi) \mathbf{b}_{pi}) \right]}_{\frac{\partial \mathbf{v}}{\partial \mathbf{x}}} \delta \mathbf{x} \quad (10)$$

where

$$\begin{aligned} \frac{\partial}{\partial \varphi} (\mathbf{R}(\varphi) \mathbf{b}_{pi}) &= \\ &= \begin{bmatrix} b_{piy} \sigma_1 + b_{piz} \sigma_2 & b_{piz} c_\alpha c_\beta c_\gamma - b_{pix} c_\gamma s_\beta + b_{piy} c_\beta c_\gamma s_\alpha & b_{piz} \sigma_3 - b_{piy} \sigma_4 - b_{pix} c_\beta s_\gamma \\ -b_{piy} \sigma_3 - b_{piz} \sigma_4 & b_{piz} c_\alpha c_\beta s_\gamma - b_{pix} s_\beta s_\gamma + b_{piy} c_\beta s_\alpha s_\gamma & b_{piz} \sigma_3 - b_{piy} \sigma_2 + b_{pix} c_\beta c_\gamma \\ b_{piy} c_\alpha c_\beta - b_{piz} c_\beta s_\alpha & -b_{pix} c_\beta - b_{piz} c_\alpha s_\beta - b_{piy} s_\alpha s_\beta & 0 \end{bmatrix} \end{aligned}$$

with

$$\sigma_1 = (s_\alpha s_\gamma + c_\alpha c_\gamma s_\beta), \quad \sigma_2 = (c_\alpha s_\gamma - c_\gamma s_\alpha s_\beta) \quad (11)$$

$$\sigma_3 = (c_\gamma s_\alpha - c_\alpha s_\beta s_\gamma), \quad \sigma_4 = (c_\alpha c_\gamma + s_\alpha s_\beta s_\gamma) \quad (12)$$

and $c_\theta = \cos \theta$ and $s_\theta = \sin \theta$. Defining $\mathbf{v}^{0r} = [v_1^0 \ v_2^0]^T$ and $\mathbf{E}_{0r} = [\mathbf{e}_1^0 \ \mathbf{e}_2^0]$, one may write

$$\mathbf{v}^{0r} = \mathbf{E}_{0r}^T \mathbf{v} = l_0 \begin{bmatrix} \cos \rho \\ \sin \rho \end{bmatrix} \quad (13)$$

with l_0 the length depicted in Figure 2. Differentiating (13) and isolating $\delta \rho$, this equation leads to the derivative of ρ with respect to \mathbf{x}

$$\delta \rho = \underbrace{\left(\frac{1}{l_0} [-\sin \rho \ \cos \rho] \mathbf{E}_{0r}^T \frac{\partial \mathbf{v}}{\partial \mathbf{x}} \right)}_{\frac{\partial \rho}{\partial \mathbf{x}}} \delta \mathbf{x} \quad (14)$$

Defining

$$\mathbf{R}_{zr}^T(\rho) = \begin{bmatrix} \cos \rho & \sin \rho & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

equation (1) may be rewritten as follows

$$\mathbf{v}^{1r} = \mathbf{R}_{zr}^T(\rho) \mathbf{E}_0^T \mathbf{v} = l_f \begin{bmatrix} \sin \phi \\ \cos \phi \end{bmatrix} + r_p \begin{bmatrix} 1 - \cos \phi \\ \sin \phi \end{bmatrix} \quad (15)$$

Differentiating this equation, the following relation is obtained

$$\begin{aligned} \overbrace{\left(\mathbf{D}_{Rz}(\rho) \mathbf{E}_0^T \mathbf{v} \frac{\partial \rho}{\partial \mathbf{x}} + \mathbf{R}_{zr}^T(\rho) \mathbf{E}_0^T \frac{\partial \mathbf{v}}{\partial \mathbf{x}} \right)}^{\frac{\partial \mathbf{v}^{1r}}{\partial \mathbf{x}}} \delta \mathbf{x} = \\ = \left(r_p \begin{bmatrix} \sin \phi \\ \cos \phi \end{bmatrix} + l_f \begin{bmatrix} \cos \phi \\ -\sin \phi \end{bmatrix} \right) \delta \phi + \begin{bmatrix} \sin \phi \\ \cos \phi \end{bmatrix} \delta l_f \end{aligned} \quad (16)$$

with

$$\mathbf{D}_{Rz}(\rho) = \begin{bmatrix} -\sin \rho & \cos \rho & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (17)$$

The differential $\delta \phi$ can be isolated in (16) leading to the derivative of ϕ

$$\delta \phi = \underbrace{\left(\frac{1}{l_f} [\cos \phi \ -\sin \phi] \frac{\partial \mathbf{v}^{1r}}{\partial \mathbf{x}} \right)}_{\frac{\partial \phi}{\partial \mathbf{x}}} \delta \mathbf{x} \quad (18)$$

Moreover, this expression can be substituted for $\delta \phi$ in (16) and δl_f is thereby obtained as

$$\delta l_f = \underbrace{\left([\sin \phi \ \cos \phi] \frac{\partial \mathbf{v}^{1r}}{\partial \mathbf{x}} - r_p \frac{\partial \phi}{\partial \mathbf{x}} \right)}_{\frac{\partial l_f}{\partial \mathbf{x}}} \delta \mathbf{x} \quad (19)$$

Differentiation of (7) leads to

$$\delta l_i = \underbrace{\left(r_p \frac{\partial \phi}{\partial \mathbf{x}} + \frac{\partial l_f}{\partial \mathbf{x}} \right)}_{\frac{\partial l_i}{\partial \mathbf{x}}} \delta \mathbf{x} \quad (20)$$

Finally, the computations detailed above can be applied to each kinematic chain $i = 1, \dots, m$, and the derivatives are concatenated to form the expression of the Jacobian matrix

$$\delta \mathbf{l} = \begin{bmatrix} \delta l_1 \\ \vdots \\ \delta l_m \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\partial l_1}{\partial \mathbf{x}} \\ \vdots \\ \frac{\partial l_m}{\partial \mathbf{x}} \end{bmatrix}}_{\mathbf{J}} \delta \mathbf{x} \quad (21)$$

4 Forward Kinematic Model

The FK consists in computing the platform pose for a given set of winch motor positions. In the case of redundant CDPRs with $m > n$, n being the number of degrees of freedom of the mobile platform, the kinematic model is overdetermined. In general, due to modeling and measurement inaccuracies, it is impossible to find a platform pose that satisfies exactly the kinematic constraints set by the m kinematic chains. Therefore, the FK may be formulated as the problem of minimizing the error between the measured cable lengths and the cable lengths computed by the IK (eq. (7)). Let \mathbf{l}_m be the set of measured cable lengths and $\hat{\mathbf{l}}(\mathbf{x})$ the cable lengths obtained with the IK model for a given pose \mathbf{x} . The error to be minimized is defined as $e(\mathbf{x}) = \|\hat{\mathbf{l}}(\mathbf{x}) - \mathbf{l}_m\|_2$, where the 2-norm is denoted by $\|\cdot\|_2$. Therefore, for a given \mathbf{l}_m , the FK Algorithm should produce the solution $\mathbf{x}^* = \arg \min_{\mathbf{x}} \|\hat{\mathbf{l}}(\mathbf{x}) - \mathbf{l}_m\|_2$.

In general, the function $e(\mathbf{x}) = \|\hat{\mathbf{l}}(\mathbf{x}) - \mathbf{l}_m\|_2$ possesses several local minima. The proposed FK algorithm takes an initial guess \mathbf{x}_g and find a \mathbf{x}^* that locally minimizes the function $e(\mathbf{x})$. The proposed algorithm assumes that \mathbf{x}_g is sufficiently close to the current platform pose and consists of an iterative scheme. For an iteration k with platform pose \mathbf{x}_k , the next iteration takes $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta \mathbf{x}_k$. Considering the approximation $\Delta \hat{\mathbf{l}} \approx \mathbf{J}(\mathbf{x}_k) \Delta \mathbf{x}$, a reasonable choice for $\Delta \mathbf{x}_k$ is

$$\Delta \mathbf{x}_k = \arg \min_{\Delta \mathbf{x}} \left\| \mathbf{J}(\mathbf{x}_k) \Delta \mathbf{x} - (\mathbf{l}_m - \hat{\mathbf{l}}(\mathbf{x}_k)) \right\|_2 \quad (22)$$

The minimization problem (22) is a Linear Least Squares problem. The solution of (22) might be computed solving $\mathbf{J}(\mathbf{x}_k)^T \mathbf{J}(\mathbf{x}_k) \Delta \mathbf{x} = \mathbf{J}(\mathbf{x}_k)^T (\mathbf{l}_m - \hat{\mathbf{l}}(\mathbf{x}_k))$ for $\Delta \mathbf{x}$. However, this solution may lead to numerical instability [16]. Accordingly, this problem is typically solved using SVD or QR factorizations [6, 16]. The SVD may be preferred because of its greater diagnostic capability in pathological cases. Nevertheless, QR factorization presents faster computing time. The latter being critical in real-time applications, the QR factorization is preferred. Therefore, the Jacobian matrix \mathbf{J} is decomposed using the QR factorisation such that $\mathbf{J} = \mathbf{Q}\mathbf{R}$, with \mathbf{Q} orthogonal and \mathbf{R} upper triangular. The solution of (22) is then obtained from the back substitution of the system $\mathbf{R}\Delta \mathbf{x} = \mathbf{Q}^T (\hat{\mathbf{l}}(\mathbf{x}_k) - \mathbf{l}_m)$. Note that the proposed algorithm does not present numerical damping. For the studied scenarios, this was not necessary.

Once $\Delta \mathbf{x}_k$ is obtained, the next pose $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta \mathbf{x}_k$ is computed. This procedure is repeated until $\|\Delta \mathbf{x}\|_2 < \varepsilon$, for ε the desired tolerance.

5 Experimental Results

The proposed FK algorithm has been implemented in the HRPCable prototype shown in Figure 3. This prototype has a 6-DoF platform fully constrained by 8 cables. The control was programmed in C++ and runs in an industrial PC Beckhoff C6920 equipped with 2.4GHz i7 core processor. The platform pose control loop runs at 125 Hz. An inner feedback loop running at 2 kHz control the cable tensions, the latter being measured by means of force sensors (load pins) placed in the drawing pulleys. Figure 3 depicts the performed trajectory. The threshold ε was defined as $\varepsilon = 1 \times 10^{-6}$.

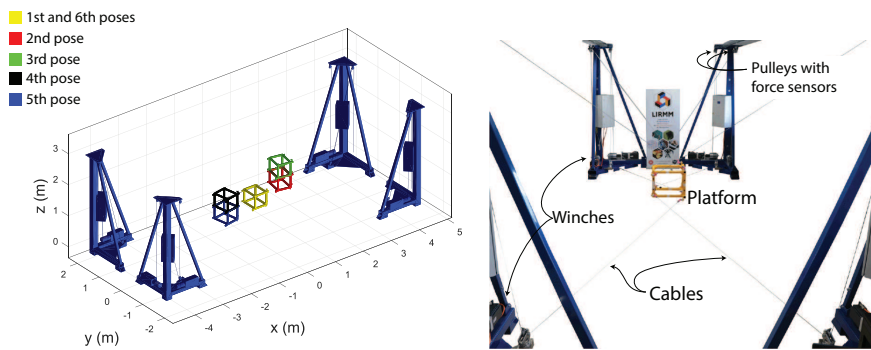


Fig. 3 HRPCable prototype and its CAD view presenting the performed trajectory

In addition to the components shown in Figure 3, a Metris K600 camera system is used to measure the pose (6-DoF) of the platform in real time with a precision of $70 \mu\text{m}$. The measurements obtained with the K600 camera system were used as reference to compare with the estimations obtained with the proposed FK algorithm. These results are shown in Figure 4. This figure also shows the necessary number of iterations during the trajectory.

For a given pair of vectors \mathbf{x}^* and \mathbf{l}^* consistent with the constraints presented in section 2, strictly evaluating the capability (denoted here as convergence capability) of an FK iterative algorithm to quickly and reliably find \mathbf{x}^* for given \mathbf{l}^* is not a trivial task and is out of the scope of this paper. The FK problem of a Stewart-Gough platform (topologically very similar to CDPRs) may have up to 40 solutions [18] and redundant configurations do not necessarily present a reduced number of possible solutions compared to the non-redundant case [7]. Therefore, it is necessary to prove that the algorithm is able to converge to \mathbf{x}^* , and not just to one of the (potentially) many \mathbf{x} that are consistent with \mathbf{l}^* . Nevertheless, it is still interesting to address this capability as follows.

The FK algorithm has been applied to IK solutions corresponding to more than 30×10^3 poses equally spaced across the workspace keeping the initial guess constant equal to $\mathbf{x}_g = [0 \ 0 \ .8 \ 0 \ 0 \ 0]^T$ for all the poses. The workspace of the prototype

Forward Kinematics of CDPRs

9

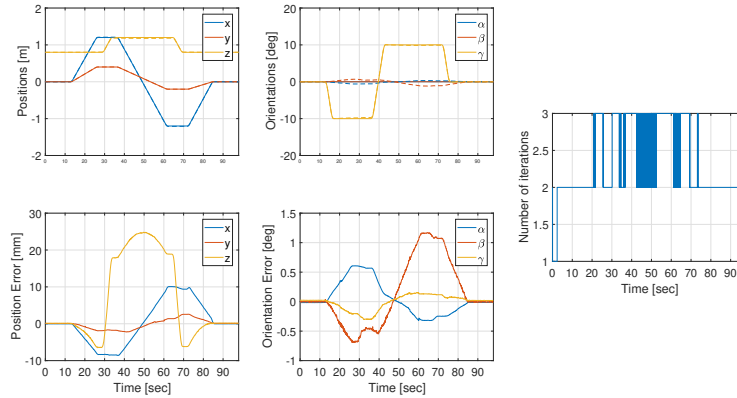


Fig. 4 Experimental results: Solid lines represent the poses estimated with the proposed FK algorithm and dashed lines represent the poses measured by the Metris K600 camera system

is a $4 \times 2 \times 1.4 \text{ m}^3$ (x, y, z) cuboid, considering $-10^\circ \leq \gamma \leq 10^\circ$, $\alpha = \beta = 0$. The algorithm obtained a pose with errors smaller than the proposed tolerance within 7 iterations for all these poses.

6 Conclusions

The present work introduced an explicit expression for the Jacobian matrix of CDPRs considering the pulley kinematics. Furthermore, a Forward Kinematics (FK) algorithm has been implemented based on this explicit expression. The Jacobian matrix is used to linearize the FK problem formulating it as a least squares problem, the latter being solved iteratively by means of QR decomposition. The algorithm was implemented in a prototype with industrial real-time environment. The proposed algorithm was successfully tested in numerous poses distributed in the workspace of the robot. Nevertheless, the measured position and orientation accuracy indicates that there is room for improvements.

Acknowledgements

This work was supported by the European Union's H2020 Program (H2020/2014-2020) under the grant agreement No. 732513 (HEPHAESTUS project).

This work also has been sponsored by the French government research program Investissements d'avenir through the Robotex Equipment of Excellence (ANR-10-EQPX-44)

References

1. Berti, A., Merlet, J.P., Carricato, M.: Solving the direct geometrico-static problem of underconstrained cable-driven parallel robots by interval analysis. *The International Journal of Robotics Research* **35**(6), 723–739 (2016). DOI 10.1177/0278364915595277. URL <https://doi.org/10.1177/0278364915595277>
2. Bostelman, R., Albus, J., Dagalakis, N., Jacoff, A., Gross, J.: Applications of the NIST robocrane. In: *Proceedings of the 5th International Symposium on Robotics and Manufacturing*, pp. 14–18 (1994)
3. Bruckmann, T., Mikelsons, L., Brandt, T., Hiller, M., Schramm, D.: *Wire robots part I: Kinematics, analysis & design*. I-Tech Education and Publishing, Vienna, Austria (2008)
4. Chellal, R., Cuvillon, L., Laroche, E.: A kinematic vision-based position control of a 6-dof cable-driven parallel robot. In: *Cable-Driven Parallel Robots*, pp. 213–225. Springer (2015)
5. Dallej, T., Gouttefarde, M., Andreff, N., Hervé, P.E., Martinet, P.: Modeling and vision-based control of large-dimension cable-driven parallel robots using a multiple-camera setup. *Mechatronics* **61**, 20–36 (2019)
6. Golub, G.H., Loan, C.F.V.: *Matrix computations*, Fourth Edition (2013)
7. Husty, M., Mielczarek, S., Hiller, M.: A redundant spatial Stewart-Gough platform with a maximal forward kinematics solution set. In: *Advances in Robot Kinematics*, pp. 147–154. Springer (2002)
8. Merlet, J.P.: Solving the forward kinematics of a Gough-type parallel manipulator with interval analysis. *The International Journal of Robotics Research* **23**(3), 221–235 (2004)
9. Merlet, J.P.: On the real-time calculation of the forward kinematics of suspended cable-driven parallel robots. In: *4th IFToMM World Congress on the Theory of Machines and Mechanisms*, Taipei, Taiwan (2015)
10. Merlet, J.P.: Simulation of discrete-time controlled cable-driven parallel robots on a trajectory. *IEEE Transactions on Robotics* **33**(3), 675–688 (2017). DOI 10.1109/TRO.2017.2664888
11. Miermeister, P., Kraus, W., Pott, A.: Differential kinematics for calibration, system investigation, and force based forward kinematics of cable-driven parallel robots. In: *Cable-Driven Parallel Robots*, pp. 319–333. Springer (2013)
12. Miermeister, P., Lächele, M., Boss, R., Masone, C., Schenk, C., Tesch, J., Kerger, M., Teufel, H., Pott, A., Bühlhoff, H.H.: The CableRobot simulator large scale motion platform based on cable robot technology. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3024–3029. IEEE (2016)
13. Nan, R., Li, D., Jin, C., Wang, Q., Zhu, L., Zhu, W., Zhang, H., Yue, Y., Qian, L.: The five-hundred-meter aperture spherical radio telescope (FAST) project. *International Journal of Modern Physics D* **20**(06), 989–1024 (2011)
14. Pott, A.: An algorithm for real-time forward kinematics of cable-driven parallel robots. In: *Advances in Robot Kinematics: Motion in man and machine*, pp. 529–538. Springer (2010)
15. Pott, A.: Influence of pulley kinematics on cable-driven parallel robots. In: *Latest Advances in Robot Kinematics*, pp. 197–204. Springer (2012)
16. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: *Numerical recipes in C++*. The art of scientific computing **2**, 1002 (1992)
17. Schmidt, V.L.: *Modeling techniques and reliable real-time implementation of kinematics for cable-driven parallel robots using polymer fiber cables*. Stuttgart: Fraunhofer Verlag (2017)
18. Wampler, C.W.: Forward displacement analysis of general six-in-parallel SPS (Stewart) platform manipulators using soma coordinates. *Mechanism and Machine Theory* **31**(3), 331–337 (1996)
19. Zanutto, D., Rosati, G., Minto, S., Rossi, A.: Sophia-3: A semiadaptive cable-driven rehabilitation device with a tilting working plane. *IEEE Transactions on Robotics* **30**(4), 974–979 (2014). DOI 10.1109/TRO.2014.2301532