

# Exit Point, Initial Length and Pose Self-Calibration Method for Cable-Driven Parallel Robots

Bozhao Wang<sup>1</sup>, Stéphane Caro<sup>2</sup>

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

`Bozhao.Wang@ls2n.fr`

`Stephane.Caro@ls2n.fr`

**Abstract.** A calibration process simulation for the interest variable values, including Cartesian coordinates of exit points, moving platform poses and initial cable lengths is performed. The simulation considers the modelling of the pulleys at exit points, and is carried out using non-linear least square method. The effects of calibration tuning parameters and of measurement pose number on calibration quality are analyzed. As a result, the calibration quality increases with the decrease of tuning parameters and the increase of measurement pose number. The use of Jacobian matrix of the interest function  $f_{i,j}$  also leads to a better calibration quality.

**Keywords:** cable-driven parallel robot, calibration, optimization.

## 1 Introduction

Cable-Driven Parallel Robots (CDPRs) are a type of parallel robot that are actuated by cables. [1] studied reconfigurable Cable-Driven Parallel Robots (RCDPRs) whose cable connection points on the base frame can be positioned at a discrete set of possible locations, and the ways to optimize the sequence of discrete reconfigurations allowing the moving-platform to follow a prescribed path. A novel concept of RCDPRs that consists of a classical CDPR mounted on multiple Mobile Bases is known as Mobile CDPR. In [2], the authors developed a methodology to trace the wrench-feasible-workspace of mobile CDPRs.

Robot calibration is performed in purpose of increasing the accuracy of robot positioning by sensor data without changing the robot design. Among different robot configurations, the positioning accuracy can be affected by a large number of sources. Most previous works on the model-based calibration are dedicated to the kinematic process [3, 4]. Other than the importance on robot positioning accuracy, calibration will also contribute on several parts, for example motion control, robot production evaluation, etc [5]. Well-designed experiments can help

calibrate quality and efficiency. [6] developed a method to enhance the measurements in geometric parameter identification, the method implemented irreducible geometric model and took into account of different sources of errors. [7] proposed a method to select manipulator configurations that allow the user to essentially improve calibration quality. [8] studied the geometric and elastostatic calibration of robotic manipulator using partial pose measurements, which do not provide the end-effector orientation, the proposed method improved the calibration efficiency. [9] proposed a concept of the user-defined test-pose to evaluate the calibration experiments quality. Sensitivity analysis is critical to model-based control robustness, in [10], the author performed sensitivity analysis of the elasto-geometrical model to their geometric and mechanical uncertainties, and carried out a case study with a reconfigurable CDPR 'CAROCA'.

Among different calibration approaches for model parameters, the nonlinear least square (NLLS) method used in this work is a basic parameter identification method. In [4], this method is used for kinematic calibration for a new developed CDPR with large workspace. Several different methods derived from NLLS are tested in [11]. Another similar method is orthogonal distance regression (ODR) [11]. In addition to parameter identification, a sensitivity analysis of CDPR is performed in [12] to decrease the computing time of the calibration process.

In order to perform CDPR calibration, measurements are required. Two types of sensors exist. The first one is the internal sensors, including motor encoders and cable force sensors. Self-calibrations, as the case in this work, are realised by the help of internal sensors [11]. The moving-platform of a CDPR is finally connected to the motors through the cables, and encoders can provide the angular position and velocity of the motors. The force sensors provide directly the cable tensions. The second category is the external sensors, for example cameras [13], which can directly measure the position and orientation of the MP. In [14], a look-and-move calibration procedure is developed based on a wireless camera. A low-cost, and efficient calibration method using camera is described in [15].

The sensors could be used alone or in various combinations [16]. In [3], the calibration is carried out with only motor encoders, on a 3-degree-of-freedom CDPR with a tetrahedral platform, which is used as a haptic interface mechanism. In [17], the robot calibration is carried out while the Cartesian coordinates of the exit points expressed in the moving-platform frame are supposed to be known. Some other contributions using multiple sensors have been made [14, 15, 18, 19]. Self-calibrations, as the case in this work, are realised by the help of internal sensors [11].

In this paper, a calibration process simulation for the interest variable values, including Cartesian coordinates of exit points, moving platform poses and initial cable lengths is performed, the robot studied has a 3-DoF, point-mass moving-platform and 4 cables. The calibration simulation takes into account the modelling of pulleys at the exit points, and is performed with encoders alone.

The effects of both calibration tuning parameters and measurement pose number on the calibration quality are analyzed. The calibrations with and without Jacobian matrix are compared. The cable sagging, elasticity and mass are neglected in the calibration process.

In Section 2, the modelling of the CDPR used in this paper is introduced, the identification problem performed is presented. In Section 3, the robot calibration methodology is introduced and new formulas to represent the calibration results are defined. Some simulation results are analyzed and discussed in Section 4. Finally, conclusions and possible future works are presented in Section 5..

## 2 CDPR Modelling

The CDPR considered in this work is illustrated in Fig. 1. The mechanism consists of a 3 DoF point-mass end-effector  $M$ , controlled by the  $m = 4$  cables; each cable will pass around a pulley fixed at an arbitrary exit point  $A_i, i = 1, 2, 3, 4$ . The cable lengths are controlled by motors. Each motor is equipped with an encoder that measures its angular position and velocity. A number of MP poses are to be defined within the base frame, including one home pose  $P_h$ . The cable exit points positions are also not known exactly. Both the poses and exit points are to be identified by the identification method presented in this paper.

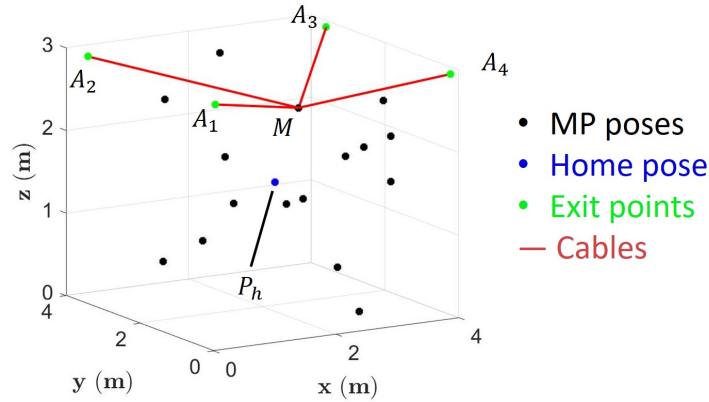
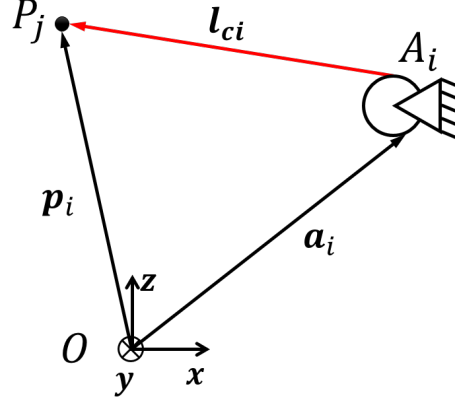


Fig. 1: Architecture of the Cable-Driven Parallel Robot under study

The  $i$ th loop of the CDPR mechanism under study is represented in Fig. 2,  $i = 1, 2, 3, 4$ . The vectors from the origin  $O$  to the pulley and the platform are denoted as  $\mathbf{a}_i$  and  $\mathbf{p}$  respectively. The distance from the pulley exit point to the platform is denoted as  $l_{ci}$ , or the  $i^{th}$  estimated cable length.

Fig. 2: The  $i$ th loop of the cable-driven parallel robot

The parameters associated to the  $i$ th pulley are described in Fig. 3. The cable starts from the encoder to the exit point  $A_i$ , then wrapped on part of the pulley sheave with the length  $l_{si}$  to  $A'_i$ , then ends on the moving platform  $P_i$ . The radius of the pulley used is  $r_p = 0.075m$ . The total cable length from exit point to the moving platform calculated as [20]:

$$l_{ti} = l_{si} + l_{ci} \quad (1)$$

where  $l_{si}$  is the length of the cable wrapped on pulley sheave, from  $A_i$  to  $A'_i$ ,  $l_{ci}$  is the cable length from  $A'_i$  to  $P_i$ , and  $l_{ti}$  is the total cable length from exit point to the moving-platform. Eq. (1) can be calculated directly from the Cartesian coordinates of cable exit points and poses with the following steps. First,  $l_{si}$  is calculated by:

$$l_{si} = r_p(\pi - \alpha_i) \quad (2)$$

Then, the pulley center  $o_i$  can be described by  $\mathbf{a}_i$  and the pulley radius  $r_p$ :

$$\mathbf{o}_i = \mathbf{a}_i + r_p {}^b\mathbf{R}_i \mathbf{x}_p \quad (3)$$

with

$${}^b\mathbf{R}_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 \\ \sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

where  $\mathbf{o}_i$  is the vector pointing from the base frame origin to the pulley center,  ${}^b\mathbf{R}_i$  is the rotation matrix from base frame to the pulley frame  $\{x, y, z\}$  shown in Fig. 3, and  $\mathbf{x}_p$  is the unit vector along the x-axis of the mentioned pulley frame.  $\mathbf{L}_i$  is the vector pointing from the exit point directly to the cable anchor point,  $\theta_i$  is the angle between x-axis of base frame and the pulley, and it can be calculated by:

$$\theta_i = \text{atan2}(L_{yi}, L_{xi}) \quad (5)$$

where  $L_{xi}$  and  $L_{yi}$  are the  $x$  and  $y$  components of  $\mathbf{L}_i$  respectively, expressed in the base frame.

The vector pointing from the pulley center to the anchor point is denoted as  $\mathbf{m}_i$ :

$$\mathbf{m}_i = \mathbf{p}_i - \mathbf{o}_i \quad (6)$$

The length  $L_i$  is expressed as:

$$l_{ci} = \sqrt{\mathbf{m}_i \mathbf{m}_i^T - r_p^2} \quad (7)$$

The angle relations around the pulley are:

$$\alpha_i = -\beta_i + \gamma_i \quad (8)$$

where  $\beta_i$  and  $\gamma_i$  angles take the form:

$$\beta_i = -\text{atan2}(l_{ci}, r_p) \quad (9)$$

and

$$\gamma_i = \arcsin\left(\frac{a_{iz} - p_{iz}}{\|\mathbf{m}_i\|_2}\right) \quad (10)$$

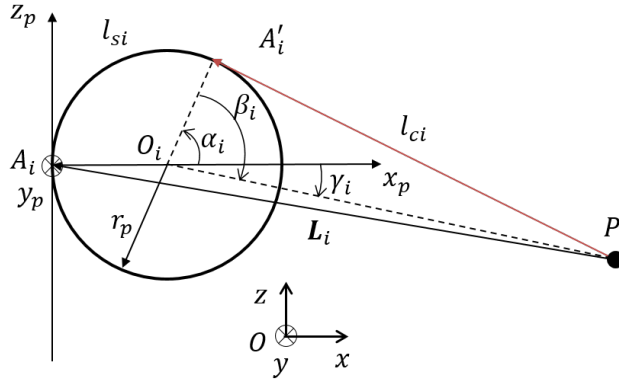


Fig. 3: Parameterization of the  $i$ th pulley

As a result, Eq. (1) can be expressed as:

$$\begin{aligned}
l_{ti} = & r_p \left[ \pi - \operatorname{atan2} \left( \sqrt{(\mathbf{p}_i - \mathbf{a}_i + r_p {}^b\mathbf{R}_i \mathbf{x}_p)(\mathbf{p}_i - \mathbf{a}_i + r_p {}^b\mathbf{R}_i \mathbf{x}_p)^T - r_p^2}, r_p \right) \right. \\
& \left. - \operatorname{arcsin} \left( \frac{a_{iz} - p_{iz}}{\|\mathbf{p}_i - \mathbf{a}_i + r_p {}^b\mathbf{R}_i \mathbf{x}_p\|_2} \right) \right] \\
& + \sqrt{(\mathbf{p}_i - \mathbf{a}_i + r_p {}^b\mathbf{R}_i \mathbf{x}_p)(\mathbf{p}_i - \mathbf{a}_i + r_p {}^b\mathbf{R}_i \mathbf{x}_p)^T - r_p^2}
\end{aligned} \tag{11}$$

### 3 Identification Methodology

As introduced before, the end-effector is connected to the motors by cables. The information provided by the encoders (the motor angular positions) is used to perform the CDPR calibration. Here,  $m = 4$  cables control the point-mass end-effector. At first, the initial cable lengths from exit points to the end-effector at home pose are denoted as  $l_{i,0}$ . It should be noted that  $l_{i,0}$  are not known exactly, and are to be identified. In order to implement calibration process, the point-mass end-effector is moved to  $n$  different positions within the workspace, meanwhile the variations in the cable lengths are obtained from the motor encoder measurements. Thus, the real cable lengths  $l_{i,j}$  are calculated by adding the cable length variations  $\Delta l_{i,j}$  to the initial cable lengths  $l_{i,0}$ , namely,

$$l_{i,j} = \Delta l_{i,j} + l_{i,0} \tag{12}$$

Similarly to [20], in real experiments the variations in the cable lengths obtained from the motor encoder measurements  $\Delta l_{i,j}$  are inputs of the identification problem. In this simulation,  $\Delta l_{i,j}$  is calculated from the assumed actual Cartesian coordinates of home pose and measurement poses. For the Cartesian coordinates of cable exit points, MP poses and the initial cable lengths, only approximate values are known in the beginning. Therefore, they are considered as the outputs of the identification problem. The known values of the identification problem are:  $mn$  cable length variations  $\Delta l_{i,j}$ . The unknowns of the problem:  $m$  initial cable lengths  $l_{i,0}$ ,  $3m$  Cartesian coordinates of cable exit points  $a_{i,x}$ ,  $a_{i,y}$ ,  $a_{i,z}$ , and  $3n$  Cartesian coordinates for the point-mass end-effector position  $p_{j,x}$ ,  $p_{j,y}$ ,  $p_{j,z}$ . The unknowns form a vector  $\mathbf{x}$  for the variables of the calibration problem:

$$\begin{aligned}
\mathbf{x} = & [a_{1,x}, a_{1,y}, a_{1,z}, \dots, a_{m,x}, a_{m,y}, a_{m,z}, l_{1,0}, \dots, l_{m,0}, \\
& p_{1,x}, p_{1,y}, p_{1,z}, \dots, p_{n,x}, p_{n,y}, p_{n,z}] \\
= & [\mathbf{a}_1, \dots, \mathbf{a}_m, \mathbf{l}_0, \mathbf{p}_1, \dots, \mathbf{p}_n]
\end{aligned} \tag{13}$$

The size of  $\mathbf{x}$  is dependent of the number of measurement poses. If  $n$  poses are used,  $\mathbf{x}$  will be a  $(4m + 3n)$ -dimensional vector.

Compared to the real cable length  $l_{i,j}$ , Eq. (1) gives the estimated cable lengths derived from the geometric model of the robot mechanism. As a consequence, a system composed of  $mn$  equations is obtained [3, 20]:

$$\begin{aligned}
f_{i,j}(\mathbf{x}) &= (l_{si} + l_{ci})^2 - (\Delta l_{i,j} + l_{i,0})^2 \\
&= \left( r_p \left[ \pi - \operatorname{atan2} \left( \sqrt{(\mathbf{p}_i - \mathbf{a}_i + r_p {}^b \mathbf{R}_i \mathbf{x}_p)(\mathbf{p}_i - \mathbf{a}_i + r_p {}^b \mathbf{R}_i \mathbf{x}_p)^T - r_p^2}, r_p \right) \right. \right. \\
&\quad \left. \left. - \operatorname{arcsin} \left( \frac{a_{iz} - p_{iz}}{\|\mathbf{p}_i - \mathbf{a}_i + r_p {}^b \mathbf{R}_i \mathbf{x}_p\|_2} \right) \right] \right. \\
&\quad \left. + \sqrt{(\mathbf{p}_i - \mathbf{a}_i + r_p {}^b \mathbf{R}_i \mathbf{x}_p)(\mathbf{p}_i - \mathbf{a}_i + r_p {}^b \mathbf{R}_i \mathbf{x}_p)^T - r_p^2} \right)^2 - (\Delta l_{i,j} + l_{i,0})^2, \\
&\quad i = 1, \dots, m, \quad j = 1, \dots, n
\end{aligned} \tag{14}$$

In order to solve the nonlinear system of equations expressed in Eq. (14), the number of equations should be larger than or equal to the number of unknowns, namely,

$$mn \geq 3m + m + 3n \tag{15}$$

From Eq. (15), for a CDPR with  $m = 4$  cables and a point-mass end-effector, the number of measurement poses should be at least equal to  $n = 16$ . Eq. (14) provides the squared difference of estimated cable length and real cable length. The nonlinear least-square solver in Matlab is used to solve the curve fitting problems of the following form, which contains the defined function Eq. (14):

$$\min_{\mathbf{x}} \left( \sum_{i=1}^m \sum_{j=1}^n f_{i,j}^2 \right) \tag{16}$$

In order to simulate the calibration process, arbitrary errors  $\mathbf{err}$  are added on each *real* variable value, to obtain  $\mathbf{x}_{\mathbf{epsi}}$ :

$$\mathbf{x}_{\mathbf{epsi}} = \mathbf{x} + \mathbf{err} = [a_{1,x} + \operatorname{err}_1, a_{1,y} + \operatorname{err}_2, \dots, p_{n,z} + \operatorname{err}_{16+3n}] \tag{17}$$

The *lsqnonlin* function takes the starting value  $\mathbf{x}_{\mathbf{epsi}}$  as input, and outputs the calibrated values  $\mathbf{x}^*$  that satisfies Eq. (16):

$$\mathbf{x}^* = [\mathbf{a}_1^*, \dots, \mathbf{a}_m^*, \mathbf{l}_0^*, \mathbf{p}_1^*, \dots, \mathbf{p}_n^*] \tag{18}$$

where  $\mathbf{a}_1^*, \dots, \mathbf{p}_n^*$  are all the simulated identified values. The other input arguments include the objective function  $f_{i,j}$ , the lower bound *lb* and upper bound *ub* for  $\mathbf{x}_{\mathbf{epsi}}$ , and optimization options that could be tuned to have a better control

of the calculation process. Apart from  $\mathbf{x}_c$ , the other output arguments used are the final solution values of  $f_{i,j}$ , saved in a vector  $\mathbf{f}_{\text{resi}}$ :

$$\mathbf{f}_{\text{resi}} = \mathbf{F}(\mathbf{x}^*) = [\mathbf{F}_1(\mathbf{x}^*), \dots, \mathbf{F}_j(\mathbf{x}^*), \dots, \mathbf{F}_n(\mathbf{x}^*)] \quad (19)$$

$$\text{with } \mathbf{F}_j(\mathbf{x}^*) = [f_{1,j}(\mathbf{x}^*), \dots, f_{i,j}(\mathbf{x}^*), \dots, f_{m,j}(\mathbf{x}^*)] \quad (20)$$

The size of  $\mathbf{f}_{\text{resi}}$  is the same as the number of interest function  $f_{i,j}$ , therefore  $\mathbf{f}_{\text{resi}}$  is  $mn$ -dimensional.

The following criterion, named  $\mu$ , is defined to assess the identification quality:

$$\mu = \frac{\sum_{h=1}^{mn} |f_{\text{resi},h}|}{mn} \quad (21)$$

For each calibration, the process is performed 500 times with different **err** applied. Then 500 calibrated results  $\mathbf{x}_l^*$ ,  $l = 1, 2, \dots, 500$ , are obtained.

$$\mathbf{x}_l^* = [\mathbf{a}_{1l}^*, \dots, \mathbf{a}_{im}^*, \mathbf{l}_{i0}^*, \mathbf{p}_{1l}^*, \dots, \mathbf{p}_{in}^*], \quad l = 1, 2, \dots, 500 \quad (22)$$

To prevent any single deviated value to produce a large effect on the final result, the mean value of them is taken, noted as  $\mathbf{x}_M^*$ :

$$x_{M,k}^* = \frac{\sum_{l=1}^{500} x_{l,k}^*}{500}, \quad k = 1, 2, \dots, 4m + 3n \quad (23)$$

To examine the calibration quality,  $\mathbf{x}_M^*$  is compared with the real variable values  $\mathbf{x}$ , noted as  $\boldsymbol{\epsilon}$ . Smaller  $\boldsymbol{\epsilon}$  values indicate better calibration quality.

$$\boldsymbol{\epsilon} = [\epsilon_1, \dots, \epsilon_k, \dots, \epsilon_{4m+3n}] \quad (24)$$

$$\epsilon_k = |x_{M,k}^* - x_k|, \quad k = 1, 2, \dots, 4m + 3n \quad (25)$$

In order to speed up the convergence of the algorithm, the following Jacobian matrix can be used in *lsqnonlin* [3]:

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \mathbf{J} = \begin{bmatrix} \frac{\partial f_{1,1}}{\partial \mathbf{a}_1}^\top & \cdots & \frac{\partial f_{1,1}}{\partial l_{1,0}} & \cdots & \frac{\partial f_{1,1}}{\partial \mathbf{p}_n}^\top \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial f_{n,m}}{\partial \mathbf{a}_1}^\top & \cdots & \frac{\partial f_{n,m}}{\partial l_{1,0}} & \cdots & \frac{\partial f_{n,m}}{\partial \mathbf{p}_n}^\top \end{bmatrix} \quad (26)$$



## 4 Simulation Results

### 4.1 Effect of Tuning Parameters

There are a list of tuning parameters inside the *option* argument of *lsqnonlin* function. Among those parameters, 'MaxFunctionEvaluations', 'MaxIterations', 'FunctionTolerance' and 'StepTolerance' have obvious and important influences on the calculation. 'MaxFunctionEvaluations' and 'MaxIterations' are the maximum number of function evaluations and iterations allowed respectively, and with both the default value of 200 times of starting vector length. If the number is not high enough, the optimization will end prematurely without reaching the desired termination tolerance. Here the two parameters are set to be 120000 and 8000 respectively. 'FunctionTolerance' and 'StepTolerance' are the termination tolerances for the function value and interest variables respectively, with both with the default values of 1e-4. The two parameters are set to be 1e-20 and 1e-12 according to their effect on the identification, measured by identification quality  $\mu$ . The result is shown in Fig. 4.

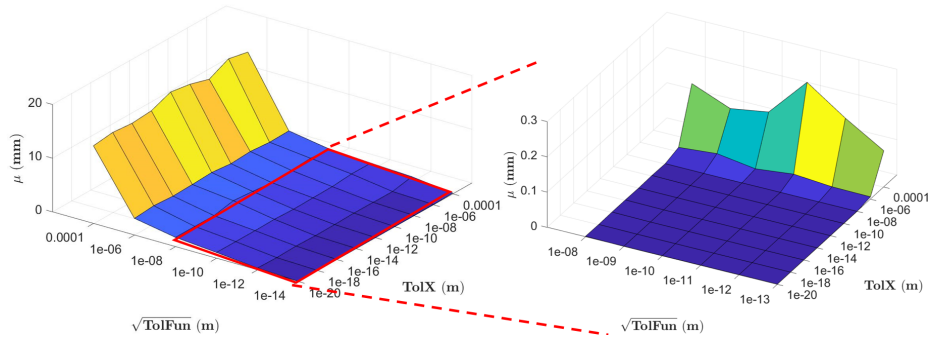


Fig. 4: TolFun, TolX influence on identification error

Identification error  $\mu$  decreases with the decrease of 'FunctionTolerance' and 'StepTolerance'. 'FunctionTolerance' causes the result to decrease rapidly while 'StepTolerance' have a less stronger effect on  $\mu$ . At the area where the square root of 'FunctionTolerance' is less than 1e-8 (m) and 'StepTolerance' is less than 1e-6 (m),  $\mu$  remains at the same level of magnitude, around 1e-5 (mm). When 'FunctionTolerance' and 'StepTolerance' equal to 1e-20 and 1e-12 respectively, the value of  $\mu$  is favourable.

### 4.2 Effect of Number of Poses

For the identification problem studied in this paper, the minimum number of measurement poses is 16. In this paper, up to 40 poses were performed to

obtain more sets of cable length variations from the encoders, in order to achieve higher calibration quality. The result is shown in Fig. 5. The identification quality increases with the increase of number of poses.

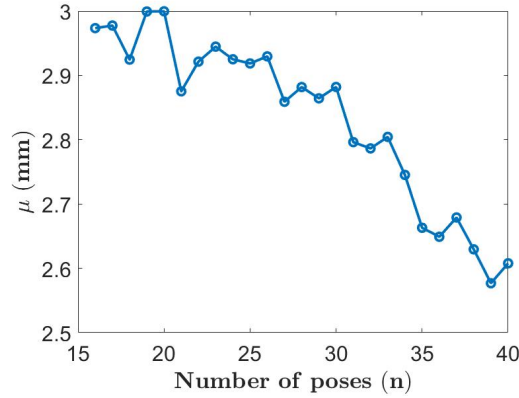


Fig. 5: Influence of number of measurement moving-platform poses on the identification quality

### 4.3 Results analysis

The difference between real and calibrated variable values  $\epsilon$  is calculated. For the Cartesian coordinates of exit points and moving platform poses, almost all the result values locate around or under 0.1 (mm), and have a decreasing trend. Examples of the y coordinate of exit point  $A_1$ , for the case of using Jacobian matrix are shown in Fig. 6. Noted by Matlab, the convergence times are 4327 s and 17278 s for calibration simulation process with and without Jacobian Matrix, respectively. The results show that Jacobian matrix contributes to a faster convergence and higher calibration quality.

## 5 Conclusions and Future Work

A simulation of the calibration process of interest variables using *lsqnonlin* function was carried out in this paper. The robot, which includes a 3-DoF, point-mass moving-platform and 4 cables, was studied considering the pulley modelling. The calibration process was assumed to be performed with encoders, the only one type of sensor alone. The effect of calibration tuning parameters on the calibration quality is analyzed. Identification error  $\mu$  decreases with the decrease of 'FunctionTolerance' and 'StepTolerance'. The effects of measurement pose number on calibration quality and calibrated interest variable values are calculated. With more poses used, the calibration quality would improve, and the errors on the calibrated interest variable values would decrease. The use of

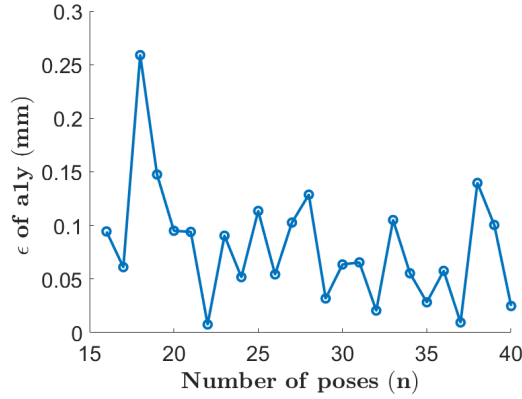


Fig. 6: The error in y-coordinate of the first exit point as a function of the number of measurement poses, with Jacobian Matrix

Jacobian matrix would result in a better interest variable error. There are uncertainties in the geometric parameters, for example geometrical errors on pulley radius, the mechanical errors concerning cable elasticity etc. Sensitivity analysis of these uncertainties will be performed to further improve the calibration quality. This work considered the pulley modelling at the exit points, however the cable elasticity and cable sagging effects were not. The CDPR studied in this work contained only 4 cables and a simple point-mass moving-platform. In future works, real experiments will be carried out, and more complex CDPRs will be considered.

## 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. L. Gagliardini, S. Caro, M. Gouttefarde, and A. Girin, “Discrete reconfiguration planning for Cable-Driven Parallel Robots,” *Mechanism and Machine Theory*, vol. 100, pp. 313–337, 2016.
2. T. Rasheed, P. Long, and S. Caro, “Wrench-Feasible Workspace of Mobile Cable-Driven Parallel Robots,” *Journal of Mechanisms and Robotics*, vol. 12, p. 031009, Jan. 2020.
3. A. Fortin-Côté, P. Cardou, and C. Gosselin, “An admittance control scheme for haptic interfaces based on cable-driven parallel mechanisms,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 819–825, 2014.
4. S. Qian, K. Bao, B. Zi, and N. Wang, “Kinematic calibration of a cable-driven parallel robot for 3D printing,” *Sensors (Switzerland)*, vol. 18, no. 9, 2018.

5. A. Elatta, L. Gen, F. Zhi, Y. Daoyuan, and L. Fei, "An overview of robot calibration," *Information Technology Journal*, vol. 3, pp. 74–78, 01 2004.
6. Y. Wu, A. Klimchik, S. Caro, B. FURET, and A. Pashkevich, "Geometric calibration of industrial robots using enhanced partial pose measurements and design of experiments," *Robotics and Computer-Integrated Manufacturing*, vol. 35, pp. 151–168, 2015.
7. A. Klimchik, S. Caro, and A. Pashkevich, "Optimal pose selection for calibration of planar anthropomorphic manipulators," *Precision Engineering*, vol. 40, pp. 214–229, 2015.
8. A. Klimchik, Y. Wu, S. Caro, B. FURET, and A. Pashkevich, "Geometric and elastostatic calibration of robotic manipulator using partial pose measurements," *Advanced Robotics*, vol. 28, pp. 1419 – 1429, Nov. 2014.
9. A. Klimchik, A. Pashkevich, Y. Wu, S. Caro, and B. Furet, "Design of Calibration Experiments for Identification of Manipulator Elastostatic Parameters," *Journal of Mechanics Engineering and Automation*, vol. 2, pp. 531–542, Sept. 2012.
10. S. Baklouti, S. Caro, and E. Courteille, "Sensitivity analysis of the elasto-geometrical model of cable-driven parallel robots," in *3rd International Conference on Cable-Driven Parallel Robots, CableCon 2017*, (Quebec City, Canada), Aug. 2017.
11. J. A. dit Sandretto, D. Daney, and M. Gouttefarde, "Calibration of a Fully-Constrained Parallel Cable-Driven Robot," *CISM International Centre for Mechanical Sciences, Courses and Lectures*, vol. 544, no. 2007, pp. 77–84, 2013.
12. S. Reicherts, S. Blume, C. Reichert, and Schramm, Dr.-Ing. Dr. h.c. Dieter, "Sensitivity Analysis of the Design Parameters for the Calibration of Cable-driven Parallel Robots," *Pamm*, vol. 16, no. 1, pp. 859–860, 2016.
13. S. Lessanibahri, P. Cardou, and S. Caro, "A Cable-Driven Parallel Robot with an Embedded Tilt-Roll Wrist," *Journal of Mechanisms and Robotics*, vol. 12, Feb. 2020.
14. M. Zavatta, M. Chianura, A. Pott, and M. Carricato, "A Vision-Based Referencing Procedure for Cable-Driven Parallel Manipulators," *Journal of Mechanisms and Robotics*, vol. 12, no. 4, pp. 1–7, 2020.
15. N. Tremblay, K. Kamali, P. Cardou, C. Desrosiers, M. Gouttefarde, and M. J. Otis, "Eye-on-hand calibration method for cable-driven parallel robots," *Mechanisms and Machine Science*, vol. 74, pp. 345–356, 2019.
16. S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and R. Medina-Carnicer, "Generation of fiducial marker dictionaries using Mixed Integer Linear Programming," *Pattern Recognition*, vol. 51, pp. 481–491, 2016.
17. D. Lau, "Initial length and pose calibration for cable-driven parallel robots with relative length feedback," *Mechanisms and Machine Science*, vol. 53, pp. 140–151, 2018.
18. G. R. Tang and B. W. Mooring, "Plane-motion approach to manipulator calibration," *The International Journal of Advanced Manufacturing Technology*, vol. 7, no. 1, pp. 21–28, 1992.
19. H. Zhuang, L. Liu, and O. Masory, "Autonomous calibration of hexapod machine tools," *Journal of Manufacturing Science and Engineering, Transactions of the ASME*, vol. 122, no. 1, pp. 140–148, 2000.
20. E. Picard, S. Caro, F. Claveau, and F. Plestan, "Pulleys and Force Sensors Influence on Payload Estimation of Cable-Driven Parallel Robots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2018)*, (Madrid, Spain), Oct. 2018.