



Research paper

# Pose-estimation methods for underactuated cable-driven parallel robots<sup>☆</sup>

Sara Gabaldo, Edoardo Idà<sup>\*</sup>, Marco Carricato

Department of Industrial Engineering, University of Bologna, 40137 Bologna, Italy

## ARTICLE INFO

## Keywords:

Underactuated robots  
Cable-driven parallel robots  
Pose estimation  
Redundant measurements  
Sensor fusion

## ABSTRACT

Cable-Driven Parallel Robots (CDPRs) displace the end-effector (EE) by means of cables configured in a parallel fashion. When a CDPR employs fewer cables than the degrees of freedom (DoFs) of its EE, it is underactuated and underconstrained. As a result, only a limited subset of the EE DoFs can be controlled assigning specific cable lengths, and determining the EE pose through direct kinematics (DK) is unfeasible, since it also depends on mechanical equilibrium. To estimate the EE pose, then, it may be convenient to acquire and combine redundant measurements. This paper introduces a sensor fusion technique that aims at achieving the optimal estimation of the pose of a generic UACDPR. Sensor fusion is achieved by an iterative nonlinear weighted least-square algorithm, which is solved by a Gauss–Newton-like method. A novel criterion for terminating iterations, which is physically sound and straightforward to implement, is put forward. Different redundant measurements are experimentally compared, so as to show the performance of the method with different sensors involved.

## 1. Introduction

Cable-Driven Parallel Robots (CDPRs) are parallel manipulators that employ cables in place of rigid-body extensible legs to control the pose of their end-effector (EE) [1]. CDPRs can cover large workspaces [2] while being modular and easy to assemble and disassemble, thus potentially reconfigurable [3].

Underactuated CDPRs (UACDPRs) are equipped with a number  $n$  of cables that is smaller than the EE degrees of freedoms (DoFs). Accordingly, their workspace is more accessible, especially if the robot is assembled in a suspended configuration, which is usually the case. The drawback is that  $n$  actuated cables can control only  $n$  EE DoFs, and, in a parallel robot, actuation deficiency usually implies constraint deficiency. As a result, the EE of UACDPRs preserves some DoFs even if all actuators are locked, and it is free to oscillate as a multi-DoF parallel pendulum [4]. UACDPRs may be employed in several applications where a limited controllability is acceptable, such as production engineering, logistics, construction, motion simulation, entertainment, search and rescue [5], and maintenance [6].

Model-based control is mostly used in the operation of CDPRs, and, if feedback control is to be used, it is necessary to determine the EE poses in real time [7]. The direct kinematics (DK) of CDPRs, presents inherent challenges due to the nonlinearity of the loop-closure equations with respect to the EE pose, and it admits multiple solutions [8,9]. If a unique solution has to be identified, it is crucial to acknowledge that it is highly sensitive to measurement errors in the acquired data [10]. In order to determine a unique DK solution, minimize solution errors caused by measurement inaccuracies, and speed up the computational processes, it

<sup>☆</sup> A preliminary and partial version of this paper was presented at the Sixth International Conference on Cable-Driven Parallel Robots, that was held in Nantes, France, on June 25–28, 2023 (Gabaldo et al., 2023).

<sup>\*</sup> Corresponding author.

E-mail address: [edoardo.ida2@unibo.it](mailto:edoardo.ida2@unibo.it) (E. Idà).



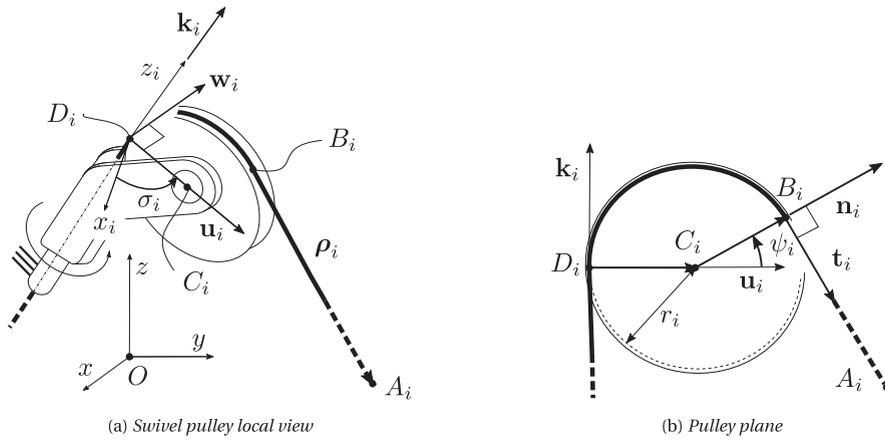


Fig. 2. Swivel pulley geometric model.

The position vectors of  $D_i$  and  $B_i$  in  $Oxyz$  are  $\mathbf{d}_i$  and  $\mathbf{b}_i$ , respectively. While  $\mathbf{d}_i$  is constant,  $\mathbf{b}_i$  varies depending on the geometrical model of the pulley and the pose of the  $EE$ . Each cable is attached to the platform at point  $A_i$ :  ${}^P \mathbf{a}'_i$  is the constant position vector of  $A_i$  in  $Px'y'z'$ ; the position vector of  $A_i$  in  $Oxyz$  is  $\mathbf{a}_i = \mathbf{p} + \mathbf{a}'_i = \mathbf{p} + \mathbf{R}^P \mathbf{a}'_i$ .

To describe the geometry of the pulley, it is convenient to define an additional fixed reference frame  $D_i x_i y_i z_i$ . Assuming points  $A_i$  and  $D_i$  and the swivel axis  $z_i$  to be co-planar, the *pulley plane* is defined by vector  $\boldsymbol{\rho}_i = \mathbf{a}_i - \mathbf{d}_i$  and  $z_i$  (Fig. 2). The *swivel angle*  $\sigma_i \in [-\pi, \pi]$  is the angle between the fixed plane  $x_i z_i$  and the pulley plane (Fig. 2(a)), and it can be computed as [20]:

$$\sigma_i = \text{atan2}(\varrho_{i,x}, \varrho_{i,y}) \quad (1)$$

where  $\varrho_{i,x}$  and  $\varrho_{i,y}$  are the components of  $\boldsymbol{\rho}_i$  along the axes  $x$  and  $y$  of  $Oxyz$ .

The *tangency angle*  $\psi_i \in [-\pi, \pi]$  is the angle between the directions  $C_i - D_i$ , described by the unit vector  $\mathbf{u}_i$ , and  $B_i - C_i$  (Fig. 2(a)). If the cable is clockwise wrapped onto the pulley, the tangency angle can be derived as:

$$\psi_i = 2 \text{atan} \left[ \frac{\varrho_{i,z_i}}{\varrho_{i,u_i}} + \sqrt{\left( \frac{\varrho_{i,z_i}}{\varrho_{i,u_i}} \right)^2 + 1 - \frac{2r_i}{\varrho_{i,u_i}}} \right] \quad (2)$$

where  $\varrho_{i,z_i}$  and  $\varrho_{i,u_i}$  are the components of  $\boldsymbol{\rho}_i$  along directions  $z_i$  and  $\mathbf{u}_i$ , respectively.

Finally, the cable vector  $\boldsymbol{\rho}_i = \mathbf{b}_i - \mathbf{a}_i$ <sup>1</sup> is:

$$l_i = \widehat{D_i B_i} + \sqrt{\boldsymbol{\rho}_i \cdot \boldsymbol{\rho}_i} \quad (3)$$

where  $l_i$  is the cable length, and the arc length  $\widehat{D_i B_i} = r_i(\pi - \psi_i)$  is the portion of cable wrapped onto the pulley.

### 3. Direct kinematics via sensor fusion

In this Section, a sensor fusion method for the solution of the DK of  $UACDPRs$  is introduced. The iterative approach proposed by Garant in [11] for fully-constrained  $CDPRs$  is adapted to  $UACDPRs$ . The Iteration Termination Conditions (ITCs) are discussed and a new ITC is proposed.

The sensor fusion approach requires a set of measures from different sensors to be available. If a  $n$ -cable  $UACDPR$  is considered, the following sets of pose-dependent geometrical variables are considered to be measurable at any time instant [12]:

- cable lengths:  $l_1^* \dots l_n^*$
- swivel angles:  $\sigma_1^* \dots \sigma_n^*$
- $EE$  orientation angles:  $\phi^*, \theta^*, \psi^*$

Cable lengths can be indirectly measured by measuring motor angular rotation through encoders; swivel angles can be directly measured by mounting an encoder on the swivel axis; the  $EE$  orientation angles can be indirectly measured by IMUs equipped with

<sup>1</sup> The use of  $\boldsymbol{\rho}$  and  $\rho$  in defining position vectors  $\mathbf{b}_i - \mathbf{a}_i$  and  $\mathbf{d}_i - \mathbf{a}_i$ , is intentional, as they are equivalent in case the pulley radius is zero, namely cables exit the frame from eyelets [22].

robust estimation algorithms, sometimes referred as AHRs (attitude and heading reference systems) [23]. Pose-dependent variables and measurements can be grouped into vectors  $\mathbf{h}(\zeta)$  and  $\mathbf{z}$ , respectively:

$$\begin{aligned}\mathbf{h}(\zeta) &= [l_1(\zeta) \dots l_n(\zeta) \quad \sigma_1(\zeta) \dots \sigma_n(\zeta) \quad \phi(\zeta) \quad \theta(\zeta) \quad \psi(\zeta)]^T \\ \mathbf{z} &= [l_1^* \dots l_n^* \quad \sigma_1^* \dots \sigma_n^* \quad \phi^* \quad \theta^* \quad \psi^*]^T\end{aligned}\quad (4)$$

If only a subset of measurements is available, it is still included in vector  $\mathbf{z}$ , and the corresponding subset of pose-dependent variables is grouped in vector  $\mathbf{h}$ . The residual associated with a certain measurement is defined as the difference between a modeled pose-dependent variable and its measurement; for convenience, we define the vector of residuals as:

$$\mathbf{r}(\zeta) = \mathbf{h}(\zeta) - \mathbf{z} \quad (6)$$

The DK problem can be stated as the problem of finding the solution vector  $\zeta$  of the non-linear system of equations  $\mathbf{r}(\zeta) = 0$ . For a single solution to exist, we must have at least as many equations as unknowns, namely  $\dim(\mathbf{r}) \geq 6$ . In practice, the use of at least 6 residuals may not be sufficient for determining the correct pose, since the Jacobian matrix of the problem may lose its rank, thus leading to indeterminacy, or the errors affecting some measurements may significantly affect the results.

If we consider a residual  $\mathbf{r}(\zeta) \in \mathbb{R}^w$ , with  $w \geq 6$ , the sensor-fusion DK problem can be formulated as

$$\min_{\zeta} f(\zeta) \quad \text{with} \quad f(\zeta) = \frac{1}{2} \mathbf{r}^T(\zeta) \mathbf{Q}^{-1} \mathbf{r}(\zeta) \quad (7)$$

where  $\mathbf{Q}$  is the covariance matrix of  $\mathbf{r}(\zeta)$  so that the solution  $\zeta$  is the best linear unbiased estimator (BLUE) of the *EE* pose [11]. In the case at hand, we assume that the elements of the residual vector are uncorrelated and its covariance matrix  $\mathbf{Q}$  is therefore diagonal, with the elements on the diagonal being the variance of the error associated with each measurement in  $\mathbf{z}$ . By doing so, we directly account for the sensor noise in the solution process. The search for local minima<sup>2</sup> of this nonlinear least-square problem is equivalent to finding the zeros of the gradient of  $f(\zeta)$  with respect to  $\zeta$ . Let this gradient be  $\nabla f(\zeta) \in \mathbb{R}^6$ , and the Hessian of  $f(\zeta)$  be  $\nabla^2 f(\zeta) \in \mathbb{R}^{6 \times 6}$ . Given an initial approximation of the solution  $\zeta_0$ , Gauss-Newton's iterative method may be used to find, at each step, a new approximation  $\zeta_k$ , with  $k = 0, 1, 2, \dots$  [24,25]:

$$\zeta_{k+1} = \zeta_k - [\nabla^2 f(\zeta_k)]^{-1} \nabla f(\zeta_k) \quad (8)$$

If  $\mathbf{A} \in \mathbb{R}^{q \times q}$  is a symmetric matrix independent of the *EE* pose, and  $\mathbf{c}(\zeta) \in \mathbb{R}^q$  is vector that depends on the *EE* pose,  $\alpha(\zeta) = \mathbf{c}^T(\zeta) \mathbf{A} \mathbf{c}(\zeta)$  is a symmetric quadratic form, then the gradient of  $\alpha(\zeta)$  with respect to  $\zeta$  can be expressed as follows<sup>3</sup>:

$$\nabla \alpha(\zeta) = 2 (\nabla \mathbf{c}^T(\zeta)) \mathbf{A} \mathbf{c}(\zeta) \quad (9)$$

with  $\nabla \mathbf{c}^T(\zeta) \in \mathbb{R}^{6 \times q}$ . Accordingly, given that  $\mathbf{Q}^{-1}$  is diagonal and constant, and vector  $\mathbf{z}$  does not depend on  $\zeta$ , the gradient of  $f(\zeta)$  can be calculated as follows:

$$\nabla f(\zeta) = (\nabla \mathbf{r}^T(\zeta)) \mathbf{Q}^{-1} \mathbf{r}(\zeta) = (\nabla \mathbf{h}^T(\zeta)) \mathbf{Q}^{-1} \mathbf{r}(\zeta) = \mathbf{J}^T \mathbf{Q}^{-1} \mathbf{r} \quad (10)$$

where  $\nabla \mathbf{r}^T(\zeta) = \nabla \mathbf{h}^T(\zeta) = \mathbf{J}^T \in \mathbb{R}^{6 \times w}$ , and the explicit dependence from  $\zeta$  was dropped for the sake of brevity. The Gauss approximation is used for the evaluation of the Hessian of  $f(\zeta)$ , and the term dependent from the second-order derivative of  $\mathbf{r}(\zeta)$ , namely  $\nabla \mathbf{J}^T \mathbf{Q}^{-1} \mathbf{r}$ , is neglected<sup>4</sup>:

$$\nabla^2 f(\zeta) = \nabla \mathbf{J}^T \mathbf{Q}^{-1} \mathbf{r} + \mathbf{J}^T \mathbf{Q}^{-1} \mathbf{J} \simeq \mathbf{J}^T \mathbf{Q}^{-1} \mathbf{J} \quad (11)$$

Finally, if  $\mathbf{r}_k$  and  $\mathbf{J}_k$  denote  $\mathbf{r}$  and  $\mathbf{J}$  calculated in  $\zeta_k$ , respectively, substituting Eqs. (10) and (11) in Eq. (8) yields:

$$\zeta_{k+1} = \zeta_k - [\mathbf{J}_k^T \mathbf{Q}^{-1} \mathbf{J}_k]^{-1} \mathbf{J}_k^T \mathbf{Q}^{-1} \mathbf{r}_k \quad (12)$$

The number of iterations to be performed depends on the algorithm stopping criteria. These criteria include several tolerances assigned a priori, namely thresholds that, if met, stop the iterations. The most frequently assigned tolerances include:

1.  $T_k$  is an upper threshold on the number of iterations. The algorithm is stopped if too many iterations have occurred, namely if

$$k \geq T_k \quad (13)$$

This condition is a safety measure against infinite executions; if met, the algorithm has usually failed.

<sup>2</sup> In the context of nonlinear optimization problems, the likelihood of converging to a global minimum is significantly influenced by the initial starting point selected for the algorithm. Initiating the algorithm in proximity to the global minimum can enhance the chances of successfully identifying it. [24]

<sup>3</sup> For the sake of brevity, we omit the proof, which relies on fundamental matrix product and derivative properties. For similar calculations, please refer to the appendix of [4].

<sup>4</sup> Computing the second-order derivative of a vector yields a 3-dimensional tensor; performing computations involving a 3-dimensional tensor, matrices, and vectors is not straightforward: an example illustrating the procedure can be found in the appendix of [4]. However, in this application, the term is several orders of magnitude smaller compared to the term to which it is added, consequently, it has been neglected. This simplifies the computation without affecting the accuracy.

2.  $T_{\|\Delta\zeta\|}$  is a lower threshold on the size of the step. The algorithm is stopped if

$$\|\Delta\zeta_k\| = \|[\mathbf{J}_k^T \mathbf{Q}^{-1} \mathbf{J}_k]^{-1} \mathbf{J}_k^T \mathbf{Q}^{-1} \mathbf{r}_k\| \leq T_{\|\Delta\zeta\|} \quad (14)$$

This condition regulates the step-size increment. When addressing a numerically singular problem (which can be visualized in a one-dimensional context as a function with an almost vertical tangent near the solution), the function exhibits low sensitivity to variations in the independent variable in the proximity of the solution; if the condition is met, the algorithm has usually stalled and a solution may or may not be found in practice.

3.  $T_{\|\nabla f(\zeta)\|}$  is a lower near-zero threshold on the first-order optimality measure. The algorithm is stopped if

$$\|\nabla f(\zeta_k)\| \leq T_{\|\nabla f(\zeta)\|} \quad (15)$$

If this condition is met, a local extremum of Eq. (7) is usually found, but this could be a local minimum for function  $\mathbf{f}$ , which is in general non convex and may have several minima.

4.  $T_{\|\mathbf{r}\|}$  is a lower near-zero threshold on the norm of the residual vector. The algorithm is stopped if

$$\|\mathbf{r}_k\| \leq T_{\|\mathbf{r}\|} \quad (16)$$

If this condition is satisfied, a global extrema for function  $\mathbf{f}$ , i.e. a solution to the DK problem, is usually found.

If the set tolerances are too loose, the results may be inaccurate. On the other hand, setting arbitrarily small tolerances may yield accuracy but at the expense of a high computational time, which may not be viable in a hard-real-time computational context. Since each element of  $\mathbf{r}_k$  may have different units and possibly different magnitudes, choosing a consistent value for the tolerance  $T_{\|\mathbf{r}\|}$  is not straightforward. We then propose to modify tolerance 4 with the following:

5.  $\mathbf{T}_r$  is a positive and near-zero vector of thresholds, whose components refer to the elements of the residual vector. The algorithm is stopped if the absolute value of each element of  $\mathbf{r}_k$  is below its own threshold, namely if

$$r_{k,i} \leq T_{r,i} \quad \forall i = 1, \dots, m \quad (17)$$

If a ground-truth instrumentation system is available (a laser tracker or a camera system to measure the  $EE$  pose accurately), the value of  $\mathbf{T}_r$  can be estimated, for instance, after calibrating the robot. Through calibration, the probability distributions of the errors affecting the acquired measures can be estimated, and  $\mathbf{T}_r$  can be set accordingly. In this manner, iterations are terminated even when the residuals of highly noisy equations are high, thus ensuring that the method does not stall. Examples will be provided in Section 4.

#### 4. Experimental validation

The Gauss–Newton method presented in Section 3 was tested on the 4-cable 6-DoF UACDPR prototype of the University of Bologna (Fig. 3). We chose a 4-cable system for its known complexity and challenges, as its motion stability is highly influenced by how cables are kept taut and controlled [26]. The prototype was used to acquire the datasets necessary for running the sensor-fusion pose-estimation method. Subsequently, the datasets underwent offline processing using MATLAB, where two distinct analyses were performed:

- We compared the accuracy and computational efficiency of the method when iteration termination conditions (16) and (17) where respectively employed. To this end, specific numerical values were assigned to the thresholds  $T_{\|\mathbf{r}\|}$  and  $\mathbf{T}_r$ . The same datasets and residuals were utilized in both scenarios to ensure a consistent comparison.
- We compared the accuracy and computational efficiency of the method with iteration termination condition (17) when using different sensors, and corresponding datasets and residuals.

Geometrical and inertial properties of the prototype were deduced from the prototype CAD models, and are summarized in Tables 1 and 2, where  $\mathbf{i}$ ,  $\mathbf{j}$ ,  $\mathbf{k}$  are the unit vectors directed along the axes of  $Oxyz$ , the coordinates of vectors  $\mathbf{d}_i$  and  ${}^P\mathbf{a}'_i$  are given, respectively, in  $Oxyz$  and  $Px'y'z'$ ,  $m$  is the mass of the  $EE$ ,  $\mathbf{I}$  its inertia matrix expressed in  $Px'y'z'$  and  $\mathbf{r}_G$  the position vector of the center of mass in  $Px'y'z'$ . To ensure an unbiased evaluation of the performances of the method, we decided to acquire various sets of measurements from different areas within the workspace. For simplicity, we assigned the  $EE$  a quasi-static workspace-scanning trajectory, defined through a series of linear paths that connect the set-points<sup>5</sup>:

$$\mathbf{p}_i = \begin{bmatrix} (-1)^{\lfloor i/22 \rfloor} (0.5 - 0.1 \lfloor i/2 \rfloor) - 1.1 \lfloor i/22 \rfloor + 4.4 \lfloor i/44 \rfloor \\ (-1)^{\lfloor (i+3)/2 \rfloor} 0.5 \\ 0.25(1 + \lfloor i/22 \rfloor) \end{bmatrix} \mathbf{m} \quad \forall i = 0, \dots, 65 \quad (18)$$

where  $\lfloor x \rfloor$  represent the greatest integer less than or equal to  $x$ , known as the floor function. The paths are followed by the  $EE$  reference point at a constant speed of 1 cm/s (by computing the cable lengths with the method proposed in [26]) and are illustrated by red lines in Fig. 4.

<sup>5</sup> For safety reasons, the method is experimentally tested while maintaining a sufficient distance from the boundaries of the workspace. Nevertheless, simulation results confirm that the method performs equally well across all regions of the workspace, see Appendix for a more detailed analysis.

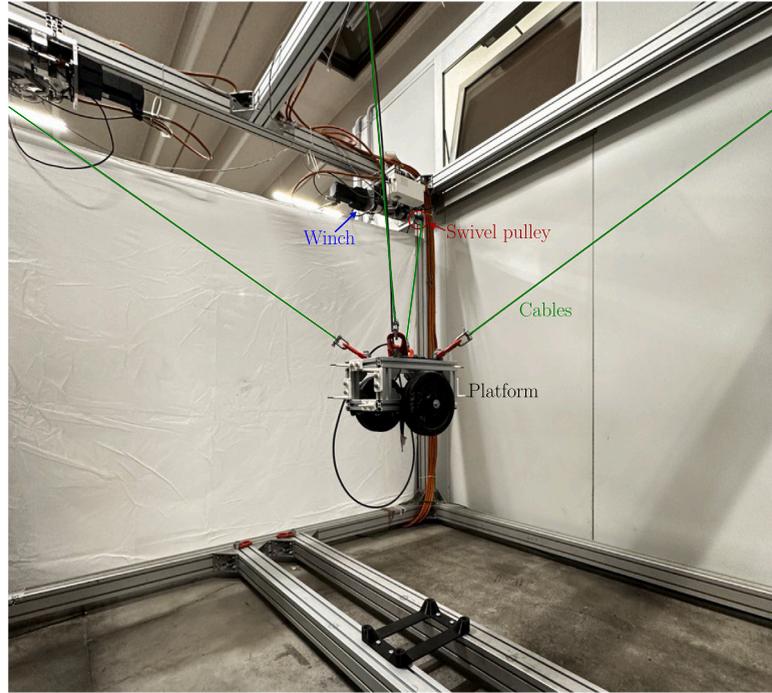


Fig. 3. UACDPR prototype used for experiments.

**Table 1**  
Actuators' properties.

| $i$                      | 1  | 2   | 3  | 4   |
|--------------------------|--|---|--|---|
| $\mathbf{d}_i$ [m]       | $\begin{bmatrix} 1.068 \\ -1.155 \\ 1.839 \end{bmatrix}$ | $\begin{bmatrix} 1.068 \\ 1.155 \\ 1.839 \end{bmatrix}$ | $\begin{bmatrix} -1.068 \\ 1.155 \\ 1.839 \end{bmatrix}$ | $\begin{bmatrix} -1.068 \\ -1.155 \\ 1.839 \end{bmatrix}$ |
| $r_i$ [m]                | 0.03   | 0.03  | 0.03   | 0.03  |
| ${}^P \mathbf{a}'_i$ [m] | $\begin{bmatrix} 0.075 \\ -0.075 \\ 0.280 \end{bmatrix}$ | $\begin{bmatrix} 0.075 \\ 0.075 \\ 0.280 \end{bmatrix}$ | $\begin{bmatrix} -0.075 \\ 0.075 \\ 0.280 \end{bmatrix}$ | $\begin{bmatrix} -0.075 \\ -0.075 \\ 0.280 \end{bmatrix}$ |
| $x_i$                    | $-\mathbf{k}$  | $-\mathbf{k}$   | $-\mathbf{k}$  | $-\mathbf{k}$   |
| $y_i$                    | $\mathbf{j}$   | $\mathbf{j}$  | $-\mathbf{j}$  | $-\mathbf{j}$   |
| $z_i$                    | $\mathbf{i}$   | $\mathbf{i}$  | $-\mathbf{i}$  | $-\mathbf{i}$   |

**Table 2**  
Inertial properties of the  $EE$ .

|                 |   |  |
|-----------------|---|--|
| $m = 12.966$ kg | $\mathbf{I} = \begin{bmatrix} 0.18 & 0 & 0 \\ 0 & 0.12 & 0 \\ 0 & 0 & 0.19 \end{bmatrix}$ kg/m <sup>2</sup> | $\mathbf{r}_G = \begin{bmatrix} 0 \\ 0 \\ 0.173 \end{bmatrix}$ m |
|-----------------|---|--|

During the motion, we acquire the datasets  $\mathbf{z}_l$ ,  $\mathbf{z}_\psi$  and  $\mathbf{z}_\theta$ , grouping the measurements relative to, respectively, cable lengths, pulley swivel angles and  $EE$  orientation angles. The first set of measurements is acquired by the incremental encoders mounted on the motor axes (with a resolution of roughly  $0.1^\circ$ ). These measurements, when combined with the kinematic model of the motors [18], provide estimates of the actual cable lengths  $l_i^*$ , with  $i = 1, \dots, 4$ . Simultaneously, incremental encoders (with a resolution of roughly  $0.02^\circ$ ) mounted on the axes of the swivel pulleys acquire measurements of swivel angles  $\sigma_i^*$ , with  $i = 1, \dots, 4$ . An inclinometer (XSens MTi-630 AHRS), fixed to the  $EE$ , provides the last set of measurements, consisting in roll and pitch angles (with  $0.5^\circ$  accuracy) and yaw angles (with  $3^\circ$  accuracy). The encoders and inclinometer acquisitions are sampled at 100 Hz. We obtain  $N = 73891$  measurement sets, denoted as  $\mathbf{z} = [l_1^* \dots l_4^* \sigma_1^* \dots \sigma_4^* \phi^* \theta^* \psi^*]^T$  along the assigned trajectory.

Ground-truth pose measurements are obtained by recording the position of 9  $EE$ -mounted reflective markers through a Vicon photogrammetry system (the marker position accuracy is  $\pm 0.2$  mm). To improve data quality, a zero-phase finite-impulse response low-pass digital filter with a stop-band frequency of 4 Hz is applied to the marker coordinates. Given the nature of our model, which does not predict any natural oscillation frequency exceeding 4 Hz [4], this filtering process effectively eliminates measurement noise

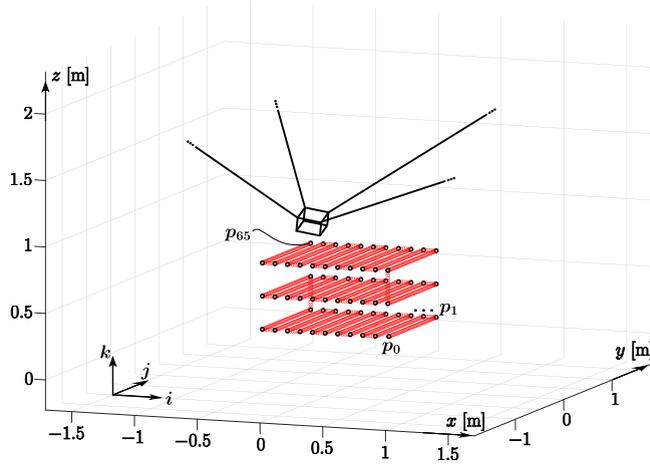


Fig. 4. UACDPR prototype performing the assigned trajectory.

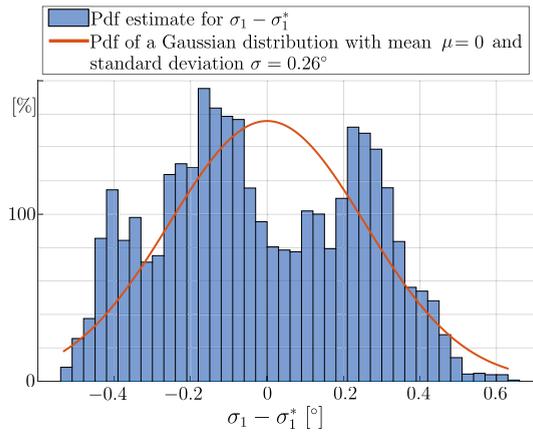


Fig. 5. Probability density function (Pdf) of the error distribution in swivel angle  $\sigma_1$  measurements and its Gaussian distribution approximation.

Table 3

Standard deviation of the measurement errors.

|                            |                                  |                                  |                                  |                                  |
|----------------------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|
| Cable lengths              | $\sigma_{l_1} = 0.39$ cm         | $\sigma_{l_2} = 0.26$ cm         | $\sigma_{l_3} = 0.10$ cm         | $\sigma_{l_4} = 0.15$ cm         |
| Pulley swivel angles       | $\sigma_{\sigma_1} = 0.22^\circ$ | $\sigma_{\sigma_2} = 0.62^\circ$ | $\sigma_{\sigma_3} = 0.28^\circ$ | $\sigma_{\sigma_4} = 0.38^\circ$ |
| Roll, pitch and yaw angles | $\sigma_\phi = 0.30^\circ$       | $\sigma_\theta = 0.16^\circ$     | $\sigma_\psi = 0.35^\circ$       |                                  |

without affecting the recorded rigid-body motion. The position of the markers is then processed to infer the *EE* reference position, and its roll, pitch, and yaw angular parameters.

The Vicon-measured platform poses allow to reconstruct, according to our model and the robot inverse kinematics (cfr. Eqs. (1) and (3)), the actual values of the swivel angles and cable lengths along the trajectory. These values, along with the Vicon-measured roll, pitch, and yaw trends, are considered to be the ground truth, and are then used to characterize the errors in the acquired measurements. It has emerged that errors in the measured variables have probability distributions that are quite dissimilar to the Gaussian distribution (Fig. 5 illustrates the error distribution in the first-pulley swivel-angle measure as an example). However, analyzing the error distributions affecting the employed sensors goes beyond the scope of this paper and, for the sake of simplicity, they have been approximated as Gaussian: the standard deviations of errors in length measurements, swivel angles, and roll, pitch, and yaw angles are given in Table 3 and they are used to estimate the confidence intervals (see [27], Chapter 5, Section 1).

#### 4.1. Comparing Gauss–Newton iteration termination conditions

Two Gauss–Newton Algorithms were implemented, whose only difference consisted in relying on a stopping criteria set including iteration termination conditions (ITCs) 1–4 (stopping criteria set A), or stopping criteria set including ITCs 1–3 and 5 (stopping criteria set B). The pseudocode for the algorithms is showed in 1 and 2, for stopping criteria set A and B, respectively, with the

characteristic ITC of each algorithm highlighted in red. The objective of this section is to assess and compare the performance of the two algorithms in terms of accuracy and computational running time, with the following assumptions:

- We analyze the behavior of the two algorithms by using all the available measurement sets at our disposal. These sets include measurements of cable lengths, swivel angles, and  $EE$  orientation.
- The initial guess for the algorithms in the first trajectory point, in both cases, is:

$$\zeta_0 = [0.5037 \quad -0.5520 \quad 0.5155 \quad -0.1790 \quad -0.1758 \quad 0.0089]^T$$

For the remaining  $n_s - 1$  poses in the trajectory, the initial guesses are the solutions computed at the previous trajectory point.

- The upper threshold on the number of solver iterations, the lower threshold on the step size and the lower near-zero threshold on the first-order optimality measure are respectively set to:  $T_k = 10^2$ ,  $T_{\|\Delta\zeta\|} = 10^{-4}$  and  $T_{\|\nabla f(\zeta)\|} = 10^{-4}$ .
- Matrix  $\mathbf{Q}$  is the diagonal covariance matrix, with its diagonal elements being the squared standard deviations of the measurement errors (reported in Table 3).

As for the other tolerances, assigning a suitable numerical value to the lower threshold of the norm of the residual vector, namely  $T_{\|\mathbf{r}\|}$  of Alg. 1, is not straightforward. The robot was experimentally characterized, and the error distributions of the acquired measurements were approximated as Gaussian distributions, with their standard deviations detailed in Table 3. By considering the vector of standard deviations

$$\sigma = [\sigma_{l_1} \quad \sigma_{l_2} \quad \sigma_{l_3} \quad \sigma_{l_4} \quad \sigma_{\sigma_1} \quad \sigma_{\sigma_2} \quad \sigma_{\sigma_3} \quad \sigma_{\sigma_4} \quad \sigma_{\phi} \quad \sigma_{\theta} \quad \sigma_{\psi}]^T$$

one can attempt to assign a threshold in the form  $T_{\|\mathbf{r}\|} = \|\sigma\|$ . However, since the norm operator forces the addition of quantities with different units, this strategy clearly lacks a physical ground. Moreover, the significance of the assigned tolerance varies based on the selected measurement units.

---

**Algorithm 1** Gauss-Newton algorithm. Stopping criteria set A.

---

**Data:**  $\zeta_0$ ;  $\mathbf{z} = [l_1^* \dots l_4^* \quad \sigma_1^* \dots \sigma_4^* \quad \phi^* \quad \theta^* \quad \psi^*]^T$ ;  $\mathbf{Q}$ ;  
 $T_k$ ;  $T_{\|\Delta\zeta\|}$ ;  $T_{\|\nabla f\|}$ ;  $T_{\|\mathbf{r}\|}$   
 $k \leftarrow 0$ ;  
 $\Delta\zeta \leftarrow T_{\Delta\zeta}$ ;  
 $\zeta \leftarrow \zeta_0$ ;  
 $\mathbf{h} \leftarrow [l_1(\zeta) \dots l_4(\zeta) \quad \sigma_1(\zeta) \dots \sigma_4(\zeta) \quad \phi(\zeta) \quad \theta(\zeta) \quad \psi(\zeta)]^T$ ;  
 $\mathbf{r} \leftarrow \mathbf{h} - \mathbf{z}$ ;  
 $f \leftarrow \frac{1}{2} \mathbf{r}^T \mathbf{Q}^{-1} \mathbf{r}$ ;  
 $\mathbf{J} \leftarrow \nabla \mathbf{r}$ ;  
 $\nabla f \leftarrow (\mathbf{J}^T) \mathbf{Q}^{-1} \mathbf{r}$ ;  
**while**  $k < T_k \wedge \|\Delta\zeta\| \geq T_{\|\Delta\zeta\|} \wedge \|\nabla f\| \geq T_{\|\nabla f\|} \wedge \|\mathbf{r}\| \geq T_{\|\mathbf{r}\|}$  **do**  
 $k \leftarrow k + 1$ ;  
 $\Delta\zeta \leftarrow -[\mathbf{J}^T \mathbf{Q}^{-1} \mathbf{J}]^{-1} \mathbf{J}^T \mathbf{Q}^{-1} \mathbf{r}$ ;  
 $\zeta \leftarrow \zeta + \Delta\zeta$ ;  
 $\mathbf{h} \leftarrow [l_1(\zeta) \dots l_4(\zeta) \quad \sigma_1(\zeta) \dots \sigma_4(\zeta) \quad \phi(\zeta) \quad \theta(\zeta) \quad \psi(\zeta)]^T$ ;  
 $\mathbf{r} \leftarrow \mathbf{h} - \mathbf{z}$ ;  
 $f \leftarrow \frac{1}{2} \mathbf{r}^T \mathbf{Q}^{-1} \mathbf{r}$ ;  
 $\mathbf{J} \leftarrow \nabla \mathbf{r}$ ;  
 $\nabla f \leftarrow (\mathbf{J}^T) \mathbf{Q}^{-1} \mathbf{r}$ ;  
**end**

---

In our computation, we use meters to measure length and radians to measure angular measurements, so that the norm of the vector of standard deviations is  $\|\sigma\| \simeq 0.017$ . If centimeters were used for lengths and degrees for angles, instead, the norm of the standard deviation vector would be  $\|\sigma\| \simeq 1.07$ . This approach, besides being physically unacceptable, can also be numerically ineffective: if some sensors are less accurate or their model is less realistic, the residuals associated with their measurements may be disproportionately high compared to the residuals from more reliable sensors. In the case at hand, for instance, significant measurement errors in the swivel-angle sensors could lead to an excessively permissive overall tolerance, which in turn would lead to pose estimations that are unacceptably imprecise. In general, a high value of  $T_{\|\mathbf{r}\|}$  disfavors accuracy for the benefit of computational time, whereas a small value typically provides the opposite characteristics. To favor accuracy, we decided to set:

$$T_{\|\mathbf{r}\|} = 10^{-8} \quad (19)$$

with lengths measured in meters and angles in radians.

On the other hand, assigning a numerical value to the single components of the residual vector, namely  $T_r$  of Alg. 2, is straightforward. Two different conditions are considered, with the lower threshold for each element of the residual vector being assigned as:

**Algorithm 2** Gauss-Newton algorithm. Stopping criteria set B.

---

**Data:**  $\zeta_0$ ;  $\mathbf{z} = [l_1^* \dots l_4^* \quad \sigma_1^* \dots \sigma_4^* \quad \phi^* \quad \theta^* \quad \psi^*]^T$ ;  $\mathbf{Q}$ ;  
 $T_k$ ;  $T_{\|\Delta\zeta\|}$ ;  $T_{\|\nabla f\|}$ ;  $(T_{r_j} \quad \forall j = 1, \dots, 11)$   
 $k \leftarrow 0$ ;  
 $\Delta\zeta \leftarrow T_{\Delta\zeta}$ ;  
 $\zeta \leftarrow \zeta_0$ ;  
 $\mathbf{h} \leftarrow [l_1(\zeta) \dots l_4(\zeta) \quad \sigma_1(\zeta) \dots \sigma_4(\zeta) \quad \phi(\zeta) \quad \theta(\zeta) \quad \psi(\zeta)]^T$ ;  
 $\mathbf{r} \leftarrow \mathbf{h} - \mathbf{z}$ ;  
 $f \leftarrow \frac{1}{2} \mathbf{r}^T \mathbf{Q}^{-1} \mathbf{r}$ ;  
 $\mathbf{J} \leftarrow \nabla \mathbf{r}$ ;  
 $\nabla f \leftarrow (\mathbf{J}^T) \mathbf{Q}^{-1} \mathbf{r}$ ;  
**while**  $k < T_k \wedge \|\Delta\zeta\| \geq T_{\|\Delta\zeta\|} \wedge \|\nabla f\| \geq T_{\|\nabla f\|} \wedge (r_j \geq T_{r_j} \quad \forall j = 1, \dots, 11)$  **do**  
 $k \leftarrow k + 1$ ;  
 $\Delta\zeta \leftarrow -[\mathbf{J}^T \mathbf{Q}^{-1} \mathbf{J}]^{-1} \mathbf{J}^T \mathbf{Q}^{-1} \mathbf{r}$ ;  
 $\zeta \leftarrow \zeta + \Delta\zeta$ ;  
 $\mathbf{h} \leftarrow [l_1(\zeta) \dots l_4(\zeta) \quad \sigma_1(\zeta) \dots \sigma_4(\zeta) \quad \phi(\zeta) \quad \theta(\zeta) \quad \psi(\zeta)]^T$ ;  
 $\mathbf{r} \leftarrow \mathbf{h} - \mathbf{z}$ ;  
 $f \leftarrow \frac{1}{2} \mathbf{r}^T \mathbf{Q}^{-1} \mathbf{r}$ ;  
 $\mathbf{J} \leftarrow \nabla \mathbf{r}$ ;  
 $\nabla f \leftarrow (\mathbf{J}^T) \mathbf{Q}^{-1} \mathbf{r}$ ;  
**end**

---

**Table 4**  
Average and maximum errors in pose components computation using algorithms 1 and 2.

| Average error:  | Alg. 1 | Alg. 2 ( $\sigma$ ) | Alg. 2 ( $3\sigma$ ) |
|-----------------|--------|---------------------|----------------------|
| Position [cm]   | 0.159  | 0.163               | 0.215                |
| Orientation [°] | 0.216  | 0.217               | 0.229                |
| Maximum error:  | Alg. 1 | Alg. 2 ( $\sigma$ ) | Alg. 2 ( $3\sigma$ ) |
| Position [cm]   | 0.883  | 0.884               | 1.21                 |
| Orientation [°] | 0.924  | 0.921               | 1.19                 |

- the standard deviation of the measurement corresponding to each element, ensuring that 68.3% of the measurements deviate from the Vicon-reconstructed value of the variable by less than the assigned tolerance;
- three times the standard deviation of the measurement corresponding to each element, ensuring that 99.7% of the measurements deviate from the Vicon-reconstructed value of the variable by less than the assigned tolerance.

The maximum and average errors on *EE* position and orientation<sup>6</sup> computed with Algorithm 1 and tolerance equal to  $10^{-8}$ , Algorithm 2 with tolerances equal to the standard deviations (denoted as  $\sigma$ ) and Algorithm 2 with tolerances equal to three times the standard deviations (denoted as  $3\sigma$ ) are reported in Table 4. The accuracies of algorithms 1 and 2 ( $\sigma$ ) are comparable, while algorithm 2 ( $3\sigma$ ) is, on average, 35% less accurate than algorithm 2 ( $\sigma$ ) in determining the position and 6% less accurate in determining the orientation.

Regarding computational time, algorithm 1 and algorithm 2 ( $\sigma$ ) respectively take an average of 6.30 ms and 5.20 ms to find a single pose, namely a single DK solution, whereas algorithm 2 ( $3\sigma$ ) takes an average of 2.10 ms to perform the same operation.

In conclusion, algorithms 1 and 2 ( $\sigma$ ) yield comparable results, both in terms of accuracy and computational time. Compared to these, algorithm 2 ( $3\sigma$ ) significantly speeds up the process, but a slightly lower accuracy is achieved. Algorithm 2, regardless of the numerical values assigned to the termination conditions ( $\sigma$  or  $3\sigma$ ), has the advantage of providing a clear methodology for iteration termination. This methodology is physically meaningful, independent of measurement units, and does not require estimation through trial-and-error.

#### 4.2. Comparing different sensor sets

In this subsection, different sensor sets are considered in order to investigate the relative influence of each sensor type in obtaining a fast and accurate solution. Given the underactuated nature of the 4-cable *UACDPR* used for experiments, it is necessary to consider

<sup>6</sup> The norm of the array of position errors represent the radius of the sphere (centered in the point  $P$  computed by the algorithm), within which the actual point  $P$  lies. The norm of the array of orientation errors indicates the overall rotation required to align the calculated end-effector orientation with the actual orientation. This rotation can be viewed as a single rotation around a fixed axis, equivalent to the cumulative effect of three sequential rotations corresponding to roll, pitch, and yaw errors. This simplification arises from employing the Rodriguez rotation formula, approximated for small angles of roll, pitch, and yaw.

**Table 5**

Average and maximum errors in pose components computation using algorithm B applied to different sensor sets.

| Average error:  | <i>sli</i> | <i>li</i> | <i>sl</i> | <i>si</i> |
|-----------------|------------|-----------|-----------|-----------|
| Position [cm]   | 0.215      | 0.208     | 0.839     | NaN       |
| Orientation [°] | 0.229      | 0.235     | 3.158     | NaN       |
| Maximum error:  | <i>sli</i> | <i>li</i> | <i>sl</i> | <i>si</i> |
| Position [cm]   | 1.21       | 1.16      | 3.62      | NaN       |
| Orientation [°] | 1.19       | 1.17      | 11.1      | NaN       |

**Table 6**

Average and maximum computational running time for determining a single DK solution.

| Average time: | <i>sli</i> | <i>li</i> | <i>sl</i> | <i>si</i> |
|---------------|------------|-----------|-----------|-----------|
| [ms]          | 2.06       | 0.966     | 0.973     | 0.879     |
| Maximum time: | <i>sli</i> | <i>li</i> | <i>sl</i> | <i>si</i> |
| [ms]          | 26.8       | 5.36      | 9.72      | 11.5      |

at least two sensor sets for obtaining a solution to the DK problem. Furthermore, although utilizing a minimum of two sensor sets is a prerequisite for establishing a DK algorithm, it may not be sufficient for accurate pose determination, due to measurement errors, noise, or residual equations being limitedly affected by *EE* pose variations [10]. For simplicity sake, we define the following acronyms to refer to different combinations of sensor sets:

- *sli*, which stands for swivel angles, cable lengths and inclinometer, represents the sensor set composed of incremental encoders mounted on the axes of the swivel pulleys, incremental encoders mounted on the motor axes and the inclinometer mounted on the *EE*;
- *sl*, which stands for swivel angles and cable lengths, represents the sensor set composed of incremental encoders mounted on the axes of the swivel pulleys and incremental encoders mounted on the motor axes;
- *li*, which stands for cable lengths and inclinometer, represents the sensor set composed of incremental encoders mounted on the motor axes and the inclinometer mounted on the *EE*;
- *si*, which stands for swivel angles and inclinometer, represents the sensor set composed of incremental encoders mounted on the axes of the swivel pulleys and the inclinometer mounted on the *EE*.

The Algorithm 2 with tolerances equal to three times the standard deviations are applied to these four cases. We chose to test this algorithm as it is more than twice faster than Alg. 2 with tolerances equal to the standard deviations, and this is crucial in the context of real-time applications. The average and maximum error in determining pose variables, and the computation running time, are reported in Tables 5 and 6, respectively.

Table 5 clearly shows that the accuracies of the results obtained by sensor sets *sli* and *li* are comparable, whereas the results obtained by sensor set *sl* are, on average, 290% less accurate than the ones obtained with sensor set *sli* in position estimation and 1280% less accurate in determining the orientation. The lower accuracies of the algorithm employing swivel angle measurements result from the unreliability of said sensors, which is significantly influenced by cable tensions: the lower the tension, the more difficult is to overcome friction in the swiveling joints, and move the pulley as per the model. The sensor set *si* leads to some form of numerical singularity, causing the algorithm to fail to converge to a solution: this fact may be justified by the orientation of the swivel axes, which motivates us to investigate in the future how to optimally orient them for pose-estimation purposes. Table 6 shows that computational times are more than halved when applying subsets of the complete combination of sensor sets *sli*.

In conclusion, (i) the sensor set *si* is not suitable for estimating the pose of the *EE* of a 4-cables *UACDPR*, (ii) the sensor set *sl* is not the optimal choice as it does not provide a good accuracy, (iii) both the *sli* and *li* sensor sets enable accurate results but (iv) the *li* sensor set is the optimal choice, as it delivers results in less than half the time compared to *sli*, while also requiring fewer sensors, which is an advantage in terms of reducing costs and complexity.

## 5. Conclusions

In this paper, a sensor fusion method designed to obtain the most accurate pose estimation for a general *UACDPR* was presented. This study introduced a novel and physically sound criterion for terminating iterations, which is not only straightforward to implement, but also demonstrated to be effective in limiting computational times by without sacrificing accuracy. This research included an experimental comparison of various redundant measurement configurations to illustrate the method performance with different sensor combinations. It was revealed that measuring the pulley swivel angles offers no apparent benefit, as this approach not only fails to improve accuracy, but also extends the computational time required for solution convergence.

In the future, we intend to investigate how to account for non-Gaussian error probability distributions affecting the sensor measurement, in order to reduce computational time while at the same time maintaining a high accuracy. Additionally, we intend to explore how our sensor-fusion algorithm would impact the trajectory planning of *UACDPRs*, as an uncaredfully planned trajectory may result in the method poor convergence due to kinematic singularities or inappropriate initial guesses.

**Table A.7**

Average and maximum errors in determining the set of poses showed in Fig. A.6(a) when using Alg. 2 ( $\sigma$ ) and sensor set  $li$ .

|                 | Average error: | Maximum error: |
|-----------------|----------------|----------------|
| Position [cm]   | 3.84           | 25.21          |
| Orientation [°] | 1.66           | 3.38           |

### CRedit authorship contribution statement

**Sara Gabaldo:** Writing – original draft, Visualization, Validation, Software, Formal analysis, Data curation. **Edoardo Idà:** Writing – review & editing, Supervision, Project administration, Methodology, Conceptualization. **Marco Carricato:** Writing – review & editing, Supervision.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Data availability

Data will be made available on request.

### Acknowledgments

The authors acknowledge the support of the Italian Ministry of University and Research (MUR) through the PRIN 2020 grant “Extending Robotic Manipulation Capabilities by Cooperative Mobile and Flexible Multi-Robot Systems (Co-MiR)” (No. 2020CMEFPK) and the support of EU/MUR FSE REACT-EU PON R&I 2014–2020 (No. CCI2014IT16M2OP005).

### Appendix. Convergence ability of the algorithm throughout the workspace

When solving a direct kinematic problem, singularities may occur at the boundaries of the workspace, and incorporating additional sensors is often an effective strategy to mitigate them [13]. To this aim, we conducted a thorough evaluation of the performance of our pose estimation method across the reachable workspace of the 4-cable prototype described in Section 4.

The reachable workspace of the robot was determined using the method presented in [26], where lower and upper bounds on each cable tension were set to 10  $N$  and 200  $N$  respectively (Fig. A.6(a)). Given a set of  $N = 85262$  reachable poses, we tested Alg. 2 ( $\sigma$ ) leveraging cable length and platform orientation measurements (sensor set  $li$ ) for reconstructing the platform pose within the workspace. In particular:

1. the cable lengths corresponding to each pose were calculated through inverse kinematics;
2. uniformly distributed random errors were assigned to cable lengths ( $\mu = 4$  cm), platform roll and pitch angles ( $\mu = 1^\circ$ ), and platform yaw angle ( $\mu = 3^\circ$ ) to mimic the inaccuracies and noise of real measurements;
3. the “inaccurate and noisy” cable lengths and orientation angles determined in the previous step were used to determine the  $EE$  pose with Alg. 2 ( $\sigma$ ); the initial guess of the algorithm was set with position and orientation differing from the real ones, on average, by 10 cm and  $20^\circ$ ;
4. the computed poses were compared with the exact ones.

Simulation results show that the algorithm always converges to a solution. The average and maximum errors in determining the platform position and orientation are reported in Table A.7. The average error aligns with our expectations: with thresholds set at 4 cm for cable lengths and  $1^\circ/3^\circ$  for platform orientation, the algorithm halts iterations when the elements of the residual vector, dependent on the computed pose, fall within the specified thresholds.

Fig. A.6(b) shows the errors in determining the platform position in various points within the reachable workspace. This graphically shows that the error in computing the pose does not depend on the proximity to the boundaries of the workspace, except for a very limited number of points in the upper part of the workspace, where the error in determining the platform position can reach up to 25 cm. This is reasonable, as cables become more and more co-planar, thus approaching a well-known direct singularity.

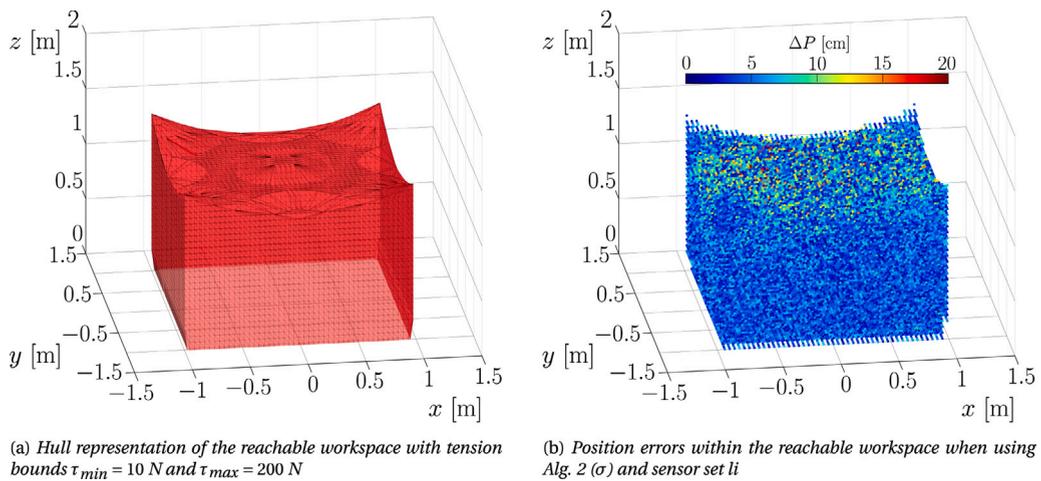


Fig. A.6. IRMA L@B 4-cable UACDPR reachable workspace.

## References

- [1] A. Pott, Cable-Driven Parallel Robots: Theory and Application, Springer, 2018, <http://dx.doi.org/10.1007/978-3-319-76138-1>.
- [2] E. Idà, V. Mattioni, Cable-driven parallel robot actuators: State of the art and novel servowinch concept, *Actuators* 11 (10) (2022) <http://dx.doi.org/10.3390/act11100290>.
- [3] J.B. Izard, M. Gouttefarde, M. Michelin, O. Tempier, C. Baradat, A reconfigurable robot for cable-driven parallel robotic research and industrial scenario proofing, in: T. Bruckmann, A. Pott (Eds.), *Cable-Driven Parallel Robots*, in: *Mechanisms and Machine Science*, vol. 12, Springer, 2013, pp. 135–148, [http://dx.doi.org/10.1007/978-3-642-31988-4\\_9](http://dx.doi.org/10.1007/978-3-642-31988-4_9).
- [4] E. Idà, S. Briot, M. Carricato, Natural oscillations of underactuated cable-driven parallel robots, *IEEE Access* 9 (2021) 71660–71672, <http://dx.doi.org/10.1109/ACCESS.2021.3071014>.
- [5] M. Angelini, E. Idà, D. Bertin, M. Carricato, E. Mantovani, D. Bazzi, V. Orassi, An underactuated cable-driven parallel robot for marine automated launch and recovery operations, in: *Proceedings of the ASME 2023 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. 47th Mechanisms and Robotics Conference (MR)*, 20–23 August 2023, Boston, 2023, <http://dx.doi.org/10.1115/DETC2023-110570>.
- [6] S. Schröder, Under constrained cable-driven parallel robot for vertical green maintenance, in: M. Gouttefarde, T. Bruckmann, A. Pott (Eds.), *Cable-Driven Parallel Robots*, in: *Mechanisms and Machine Science*, vol. 104, Springer, 2021, pp. 389–400, [http://dx.doi.org/10.1007/978-3-030-75789-2\\_31](http://dx.doi.org/10.1007/978-3-030-75789-2_31).
- [7] A. Pott, An algorithm for real-time forward kinematics of cable-driven parallel robots, in: J. Lenarcic, M.M. Stanisic (Eds.), *Advances in Robot Kinematics: Motion in Man and Machine*, Springer, 2010, pp. 529–538, [http://dx.doi.org/10.1007/978-90-481-9262-5\\_57](http://dx.doi.org/10.1007/978-90-481-9262-5_57).
- [8] A. Berti, J.P. Merlet, M. Carricato, Solving the direct geometrico-static problem of underconstrained cable-driven parallel robots by interval analysis, *Int. J. Robot. Res.* 35 (6) (2016) 723–739.
- [9] G. Abbasnejad, M. Carricato, Direct geometrico-static problem of underconstrained cable-driven parallel robots with n cables, *IEEE Trans. Robot.* 31 (2) (2015) 468–478, <http://dx.doi.org/10.1177/02783649155595277>.
- [10] S. Gabaldo, E. Idà, M. Carricato, Sensitivity of the direct kinematics of underactuated cable-driven parallel robots to redundant sensor-measurement errors, in: O. Altuzarra, A. Kecskeméthy (Eds.), *Advances in Robot Kinematics 2022*, in: *Springer Proceedings in Advanced Robotics*, vol. 24, Springer, 2022, pp. 131–138, [http://dx.doi.org/10.1007/978-3-031-08140-8\\_15](http://dx.doi.org/10.1007/978-3-031-08140-8_15).
- [11] X. Garant, A. Campeau-Lecours, P. Cardou, C. Gosselin, Improving the forward kinematics of cable-driven parallel robots through cable angle sensors, in: C. Gosselin, P. Cardou, T. Bruckmann, A. Pott (Eds.), *Cable-Driven Parallel Robots*, in: *Mechanisms and Machine Science*, vol. 53, Springer, 2018, pp. 167–179, [http://dx.doi.org/10.1007/978-3-319-61431-1\\_15](http://dx.doi.org/10.1007/978-3-319-61431-1_15).
- [12] J.P. Merlet, An experimental investigation of extra measurements for solving the direct kinematics of cable-driven parallel robots, in: *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6947–6952. 21–26 May 2018, Brisbane, QLD, Australia, 2018, <http://dx.doi.org/10.1109/ICRA.2018.8460901>.
- [13] A. Fortin-Côté, P. Cardou, A. Campeau-Lecours, Improving cable driven parallel robot accuracy through angular position sensors, in: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4350–4355. 9–14 October 2016, Daejeon, Korea (South), 2016, <http://dx.doi.org/10.1109/IROS.2016.7759640>.
- [14] M. Korayem, M. Yousefzadeh, S. Kian, Precise end-effector pose estimation in spatial cable-driven parallel robots with elastic cables using a data fusion method, *Measurement* 130 (2018) 177–190, <http://dx.doi.org/10.1016/j.measurement.2018.08.009>.
- [15] V. Le Nguyen, R.J. Caverly, Cable-driven parallel robot pose estimation using extended kalman filtering with inertial payload measurements, *IEEE Robot. Autom. Lett.* 6 (2) (2021) 3615–3622, <http://dx.doi.org/10.1109/LRA.2021.3064502>.
- [16] V.L. Nguyen, R.J. Caverly, CDPR forward kinematics with error covariance bounds for unconstrained end-effector attitude parameterizations, in: M. Gouttefarde, T. Bruckmann, A. Pott (Eds.), *Cable-Driven Parallel Robots*, in: *Mechanisms and Machine Science*, vol. 104, Springer, 2021, pp. 37–49, [http://dx.doi.org/10.1007/978-3-030-75789-2\\_4](http://dx.doi.org/10.1007/978-3-030-75789-2_4).
- [17] N. Puri, R.J. Caverly, Pose estimation of a cable-driven parallel robot using Kalman filtering and forward kinematics error covariance bounds, in: P. Larochele, J.M. McCarthy (Eds.), *Proceedings of the 2022 USCToMM Symposium on Mechanical Systems and Robotics*, in: *Mechanisms and Machine Science*, vol. 118, Springer, 2022, pp. 65–75, [http://dx.doi.org/10.1007/978-3-030-99826-4\\_7](http://dx.doi.org/10.1007/978-3-030-99826-4_7).
- [18] E. Idà, T. Bruckmann, M. Carricato, Rest-to-rest trajectory planning for underactuated cable-driven parallel robots, *IEEE Trans. Robot.* 35 (6) (2019) 1338–1351, <http://dx.doi.org/10.1109/TRO.2019.2931483>.
- [19] E. Idà, S. Briot, M. Carricato, Identification of the inertial parameters of underactuated cable-driven parallel robots, *Mech. Mach. Theory* 167 (2022) 104504, <http://dx.doi.org/10.1016/j.mechmachtheory.2021.104504>.

- [20] E. Idá, J.P. Merlet, M. Carricato, Automatic self-calibration of suspended under-actuated cable-driven parallel robot using incremental measurements, in: A. Pott, T. Bruckmann (Eds.), *Cable-Driven Parallel Robots*, in: *Mechanisms and Machine Science*, vol. 74, Springer International Publishing, 2019, pp. 333–344, [http://dx.doi.org/10.1007/978-3-030-20751-9\\_28](http://dx.doi.org/10.1007/978-3-030-20751-9_28).
- [21] S. Gabaldo, E. Idà, M. Carricato, Pose-estimation methods for planar underactuated cable-driven parallel robots, in: S. Caro, A. Pott, Bruckmann T. (Eds.), *Cable-Driven Parallel Robots*, in: *Mechanisms and Machine Science*, vol. 132, Springer, 2023, pp. 3–15, [http://dx.doi.org/10.1007/978-3-031-32322-5\\_1](http://dx.doi.org/10.1007/978-3-031-32322-5_1).
- [22] E. Idà, S. Briot, M. Carricato, Robust trajectory planning of under-actuated cable-driven parallel robot with 3 cables, in: J. Lenarčič, B. Siciliano (Eds.), *Advances in Robot Kinematics 2020*, in: *Springer Proceedings in Advanced Robotics*, vol. 15, Springer, 2021, pp. 65–72, [http://dx.doi.org/10.1007/978-3-030-50975-0\\_9](http://dx.doi.org/10.1007/978-3-030-50975-0_9).
- [23] <https://www.movella.com/products/sensor-modules/xsens-mti-630-ahrs> (Accessed 27 December 2023).
- [24] J. Nocedal, S.J. Wright, *Numerical Optimization*, second ed., Springer, 2006, <http://dx.doi.org/10.1007/978-0-387-40065-5>.
- [25] B.M. Bell, The iterated Kalman filter update as a Gauss–Newton method, *IEEE Trans. Autom. Control* 38 (2) (1993) 294–297, <http://dx.doi.org/10.1109/9.250476>.
- [26] E. Idá, M. Carricato, Static workspace computation for underactuated cable-driven parallel robots, *Mech. Mach. Theory* 193 (2024) 105551, <http://dx.doi.org/10.1016/j.mechmachtheory.2023.105551>.
- [27] W.C. Navidi, *Confidence Intervals*, in: *Statistics for engineers and scientists*, McGraw-Hill, 2015, ISBN13: 9781266672910.