

Alma Mater Studiorum Università di Bologna  
Archivio istituzionale della ricerca

A Vision-Based Referencing Procedure for Cable-Driven Parallel Manipulators

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

*Published Version:*

Zavatta, M., Chianura, M., Pott, A., Carricato, M. (2020). A Vision-Based Referencing Procedure for Cable-Driven Parallel Manipulators. JOURNAL OF MECHANISMS AND ROBOTICS, 12(4), 1-7 [10.1115/1.4045745].

*Availability:*

This version is available at: <https://hdl.handle.net/11585/758210> since: 2020-11-16

*Published:*

DOI: <http://doi.org/10.1115/1.4045745>

*Terms of use:*

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).  
When citing, please refer to the published version.

(Article begins on next page)

This is the final peer-reviewed accepted manuscript of:

**Zavatta, M., Chianura, M., Pott, A., and Carricato, M. (March 6, 2020). "A Vision-Based Referencing Procedure for Cable-Driven Parallel Manipulators." ASME. *J. Mechanisms Robotics*. August 2020; 12(4): 044502**

The final published version is available online at:

<https://doi.org/10.1115/1.4045745>

#### Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

*This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>)*

***When citing, please refer to the published version.***

# A vision-based referencing procedure for cable-driven parallel manipulators

**Michele Zavatta\***

University of Bologna  
Department of Industrial Engineering, DIN  
Viale Risorgimento 2, 40136, Bologna, Italy  
Email: michele.zavatta@gmail.com

**Mattia Chianura\***

University of Bologna  
Department of Industrial Engineering, DIN  
Viale Risorgimento 2, 40136, Bologna, Italy  
Email: mattiazchianura@gmail.com

**Andreas Pott**

University of Stuttgart  
Institute for control engineering of machine tools and manufacturing units, ISW  
Seidenstr. 36 , D-70174, Stuttgart, Germany  
Email: andreas.pott@isw.uni-stuttgart.de

**Marco Carricato**

University of Bologna  
Department of Industrial Engineering, DIN  
Viale Risorgimento 2, 40136, Bologna, Italy  
Email: marco.carricato@unibo.it

*In the last three decades, Cable-Driven Parallel Robots (CDPRs) have captured a growing attention in the robotics field. Indeed, they promise to bring automation in fields where it is not affirmed yet, granting ease of scaling and re-configurability. For large-workspace cable robots, accuracy is an important issue. In this paper, a look-and-move procedure is proposed, based on a wireless camera, to refer the coordinate frame of the CDPR platform to another known coordinate frame. Two sample cases are studied and presented. In the first, the proposed vision-based system is employed to let the platform precisely attain its home position. In the second, the platform is referenced to an external coordinate frame, in order to accurately accomplish an assigned task. For both cases, experiments are successfully carried out.*

## 1 Introduction

A cable-driven parallel robot (CDPR) consists of a mobile platform connected to a fixed frame by means of cables whose lengths are controlled by electric winches. This simple structure confers many appealing characteristics to CDPRs, such as large workspace and high reconfigurability. CDPRs may help to introduce or foster automation in fields where it is not affirmed yet, such as the construction industry. The potential contribution of CDPRs in this field was briefly explored in [1, 2].

However, when a CDPR is operated for large-scale manipulation, its accuracy is seriously challenged. Due to model uncertainties, high accuracy in CDPRs can only be achieved through a precise estimation of the platform pose, in order to perform a feedback correction. Although considerable research was spent in the last decade, when it comes to a large workspace, forward kinematics algorithms show their limits [3]. Therefore, the integration of additional sensors, such as cameras, is in order. Visual control of manipulators promises advantages when it comes to targets whose positions are not precisely known, or with manipulators which may be inac-

---

\* Both authors contributed equally to this manuscript

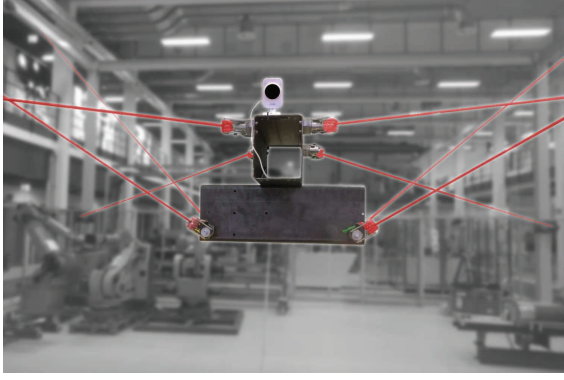


Fig. 1: An optical device mounted on IPAnema 3 platform, allowing the implementation of a vision-based system for improving accuracy

curate. Traditionally, visual sensing and manipulation can be accomplished according two approaches [4, 5]. The *look-and-move* approach uses visual acquisition to generate joint set-points, with the system feedback being realized in the joint space based on the joint position measurements. On the other hand, *visual servoing* does not use conventional position controller/sensors, but directly uses images captured in real time to correct for joint errors, thus dealing with communication and image processing delay. For the particular case of CDPRs, vision-based control was proposed and tested, but only relative to simplified cases, like planar or small workspaces [6–8]. In [9] the stability of CDPRs under vision-based control with only one camera was analyzed and assessed. Up to now, the vision-based control of spatial large CDPRs has been mainly proposed and simulated [10], mostly due to difficulties in embedding sensors in such big environments. A very recent practical implementation can be found in [11], which appeared during the review process of this contribution.

The precise placement of the platform with respect to (w.r.t.) a known coordinate frame, namely a *referencing procedure*, is crucial for the robot accuracy. The platform pose is commanded by six parameters, three for position and three for orientation. Nowadays, the position accuracy achieved by the most developed large CDPR prototypes is in the order of magnitude of some centimeters [3, 12]. This paper presents a novel procedure for CDPRs, aiming at referring the platform pose with respect to a known coordinate frame, which exploits a vision-based algorithm. The algorithm provides a correction to the pose of the robot attained through the model-based control. The implemented correction phase drives the evaluated error to become less than a set threshold. To take advantage of high repeatability of vision-based measurements, the desired pose needs to be recorded and stored. Due to the delay of wireless signal transmission and the time requested for image processing, the non real-time look-and-move strategy is used. The procedure allows precise information to be obtained about the pose of the platform without exploiting forward kinematics. Thus, it is not limited in accuracy by modelling uncertainties regarding cable elongations

and sagging, mechanical imperfections, modifications in the payload, etc.

The proposed procedure can also be used to match the platform frame to the robot fixed one in the home configuration, thus realizing a fast and accurate *homing procedure* that can replace expensive and time-consuming re-calibration processes. Indeed, due to the effects of temperature changes, creep, hysteresis, friction, etc., the home pose may drift in time w.r.t. to the nominal one (which the kinematic parameters implemented in the robot control models refer to), thus leading to a deterioration in the robot accuracy. The referencing procedure presented in the paper was tested on the large scale demonstrator IPAnema 3, available at Fraunhofer IPA in Stuttgart [13, 14]. It represents one of the first inquiries corroborated by experiments on the improvement of the performance of large spatial CDPRs, by virtue of external measurements obtained through optical devices (Fig. 1). In particular, it is the first application of the look-and-move strategy to this type of manipulators.

This paper is organized as follows. Section 2 reports the models of the CDPR inverse kinematics and the imaging formation in an optical device. Section 3 illustrates the novel referencing procedure. Section 4 describes the implementation of the procedure in the prototype IPAnema 3. Section 5 presents two applications with corresponding experimental validation. Section 6 draws conclusions. A list of the symbols and abbreviations used in the paper is reported in the Nomenclature.

## 2 Models

### 2.1 Inverse kinematic model

The six Cartesian pose variables that have to be commanded to the robot in order to achieve the desired pose are projected in the joint space through the inverse kinematic model. Referring to Fig. 2, the  $i$ -th cable vector  $\mathbf{l}_i$  is

$$\mathbf{l}_i = \mathbf{a}_i - \mathbf{r} - \mathbf{R} \mathbf{b}_i, \quad (1)$$

where  $\mathbf{a}_i$  and  $\mathbf{b}_i$  are the position vectors of the  $i$ -th cable anchor points on the base and the platform, expressed in the corresponding frames. More sophisticated formulas including the kinematics of the swivelling pulleys guiding the cables can be expressed as [15]

$$\mathbf{l} = \boldsymbol{\psi}_{IK}(\mathbf{r}, \mathbf{R}), \quad (2)$$

where  $\mathbf{l} = [l_1, \dots, l_m]^T$  is a vector containing all cable lengths.

### 2.2 Camera model

Among the models of a camera with a finite center, one of the most used is the *pinhole camera model*. Referring to Fig. 3,  $uv$  is the image plane,  $u$  and  $v$  are the axes of the bidimensional coordinate frame  $\mathcal{H}_I$ ,  $f$  is the focal length,  $c_x$  and  $c_y$  are the coordinates identifying the camera principal point,



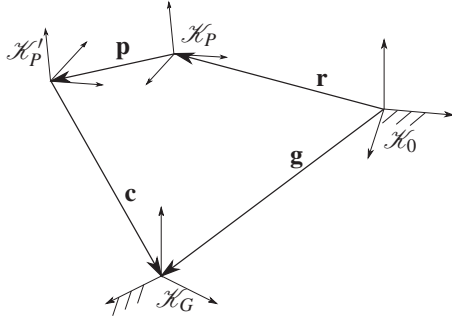


Fig. 4: The four coordinate frames involved in the referencing procedure

the vision-based referencing procedure is to re-match these values. In more detail, the platform, starting from a generic pose, is commanded to reach the aforementioned target pose, usually unsuccessfully. At this point, the vision-based control is activated and drives the platform so that the current values of the external feedback  $\{\mathbf{c}\}_G$  and  $\mathbf{R}_{GP'}$  draw closer and closer to the target values  $\{\mathbf{c}^*\}_G$  and  $\mathbf{R}_{GP'}^*$ , until the error is smaller than a settled threshold.

Both in the case the camera is mounted on the platform and in the case the camera is fixed to the ground, the camera is likely to be placed far from the control unit, especially when it comes to a large scale CDPR. Thus, wireless signal transmission is needed, which introduces a delay and makes a real-time control unfeasible. For this reason, the error is corrected by an iterative “look-and-move” strategy. Two phases are distinguished in every iteration. Firstly, the measurement phase takes place, computing the error of the reciprocal pose between  $\mathcal{K}_{P'}$  and  $\mathcal{K}_G$ . Secondly, if the error is above the threshold, a correction movement is assigned, according to the computed error.

By exploiting the repeatability of the measurement provided by the camera rather than its accuracy, the above procedure can be accomplished by simple and inexpensive 2D cameras.

### 3.2 Error computation

The pose error is computed at every measurement phase, distinguishing between orientation and position. The error must be referred to the inertial frame  $\mathcal{K}_O$ , for the platform is usually controlled w.r.t. it.

**Orientation** The platform orientation is defined by three consecutive rotations about the fixed axes  $x$ ,  $y$  and  $z$  of  $\mathcal{K}_O$ . With the measurement available by the pose sensor, the orientation error referred to  $\mathcal{K}_G$  can be computed as

$$\mathbf{B} = \mathbf{R}_{GP'} \mathbf{R}_{GP'}^{*T} . \quad (6)$$

Through  $\mathbf{R}_{OG}$  (which is assumed given or measurable for each particular application), matrix  $\mathbf{B}$  can be referred to  $\mathcal{K}_O$  as

$$\mathbf{E} = \mathbf{R}_{OG} \mathbf{B} \mathbf{R}_{OG}^T = \mathbf{R}_{OG} \mathbf{R}_{GP'} \mathbf{R}_{GP'}^{*T} \mathbf{R}_{OG}^T = \mathbf{R}_{OP'} \mathbf{R}_{OP'}^{*T} . \quad (7)$$

**Position** Likewise, the position error in  $\mathcal{K}_G$  is:

$$\{\Delta \mathbf{c}\}_G = \{\mathbf{c}\}_G - \{\mathbf{c}^*\}_G . \quad (8)$$

Eventually, by means of  $\mathbf{R}_{OG}$ , the position error is projected into the inertial frame as

$$\{\Delta \mathbf{c}\}_O = \{\mathbf{c}\}_O - \{\mathbf{c}^*\}_O . \quad (9)$$

The most relevant uncertainty in the procedure concerns  $\mathbf{R}_{OG}$ , which should be measured by precise external measurement systems that are not always available. Since  $\mathbf{R}_{OG}$  is used to project the components of the position error in the inertial frame (in which the correction takes place), an error in  $\mathbf{R}_{OG}$  leads  $\{\Delta \mathbf{c}\}_O$  to have a wrong direction (this would occur even if the magnitude of the correction,  $\|\{\Delta \mathbf{c}\}_G\| = \|\{\Delta \mathbf{c}\}_O\|$ , were perfectly computed). If the correction direction is only slightly diverted, the procedure still converges, though more iterations are needed. On the contrary, the procedure fails if the error in  $\mathbf{R}_{OG}$  leads some component in  $\{\Delta \mathbf{c}\}_O$  to change sign w.r.t. the ideal value, thus causing a relevant direction error. The simulations presented in [21] show that for errors up to  $5^\circ$  in the parameters of  $\mathbf{R}_{OG}$  (unlikely scenario), the procedure still converges.

### 3.3 Corrections

The desired pose change is executed by means of seventh-order polynomial interpolations in the time domain, which ensure limited and continuous jerk. This interpolation can be performed either in the operational space or in the joint space. In the former case, the entire motion of the platform is assigned, but inverse kinematics has to be computed for every step of the interpolation, in order to obtain the corresponding cable lengths. In the latter case, instead, the computation of the inverse kinematics has to be done only once, in order to determine the target pose. Then, only time interpolation of cable lengths is left to be performed. This results in less computations, but at the expense of the pose control in the operational space, which only preserves initial and final desired poses. In this paper, operational-space interpolation was implemented, since priority was given to pose control rather than correction speed.

**Orientation** One triplet of Euler angles can be extracted from matrix  $\mathbf{E}^T$ . It includes three successive rotations about the  $\mathcal{K}_O$  axes that the platform must perform in order to reach the target orientation. Euler representation was mainly chosen due to the control structure, since the control interface of the robot IPAnema 3 is designed to receive three angles as input. Moreover, Euler representation provides a clear relationship between the computed corrections and the robot movements, which helps preventing errors and crashes during the tests on the prototype.<sup>1</sup> For a given matrix  $\mathbf{E}^T$ , the

<sup>1</sup>For future developments of the project, the Quaternion representation will be considered for implementation, because it allows representation sin-



target orientation to reach is:

$$\mathbf{R}_{OP'}^* = \mathbf{E}^T \mathbf{R}_{OP'} . \quad (10)$$

Since the desired movement is performed through a polynomial interpolation in the operational space, during each orientation correction only the rotation around one  $\mathcal{H}_O$  axis can be commanded.<sup>2</sup> Therefore, only an elementary rotation matrix  $\mathbf{E}_{el}$  is applied to  $\mathcal{H}_{P'}$  (choosing, for instance, the one corresponding to the axis where the highest error is registered).

At the  $\eta$ -th iteration, the input to be sent to the controller to correct the platform orientation consists in the contribution of the current iteration, the contribution of all previous iterations, and the last input  $\mathbf{R}_{OP_{d,CNC}}$ , given to the controller before activating the vision-based control:

$$\mathbf{R}_{tot,\eta} = \mathbf{E}_{el,\eta}^T \prod_{s=1}^{\eta-1} \mathbf{E}_{el,(\eta-s)}^T \mathbf{R}_{OP_{d,CNC}} . \quad (12)$$

It is worth emphasizing that the contribution of all previous iterations has to be taken into account since the robot is commanded in orientation by matrix  $\mathbf{R}_{OP_{d,CNC}}$ , in accordance to the user input. Once the vision-based correction begins,  $\mathbf{R}_{OP_{d,CNC}}$  remains constant until the end of the process (it stops representing the orientation of the platform as soon as the vision-based correction begins). Therefore, the orientation of the platform is expressed by  $\mathbf{R}_{tot,\eta}$ , which is obtained every cycle by computing a correction  $\mathbf{E}_{el,\eta}$  with respect to the current state  $s$ . The current state  $s$  is described by  $\mathbf{R}_{OP_{d,CNC}}$ , updated with all corrections occurred in the previous iterations.

**Position** Likewise, the position error computed in Eq. (9) represents the translation

$$\{\mathbf{c}^*\}_O = \{\mathbf{c}\}_O - \{\Delta\mathbf{c}\}_O \quad (13)$$

that the platform must perform in order to reach the target position. Also, the position correction takes place in multiple iterations. At the  $\sigma$ -th iteration, the position to be fed to the control consists in the contribution of the current iteration,

gularities to be avoided and it is numerically more efficient. Computations will be performed in Quaternion representation, then results will be sent to the controller after being converted in Euler representation

<sup>2</sup>As the matrix product is noncommutative, no multiple angles can be corrected together. In fact, this would imply the simultaneous change of two (or three) angles and thus the multiplication of the two (or three) corresponding elementary rotation matrices each PLC cycle constituting the correction phase. In formulas, the simultaneous correction of multiple angles requires the following to hold:

$$\mathbf{R}_j(\Delta\theta_j) \mathbf{R}_i(\Delta\theta_i) = \prod_{cycles=1}^n \mathbf{R}_j\left(\frac{\Delta\theta_j}{n}\right) \mathbf{R}_i\left(\frac{\Delta\theta_i}{n}\right) . \quad (11)$$

with  $i, j$  arbitrary axes and  $n > 1$ . Obviously, Eq. (11) is not true.

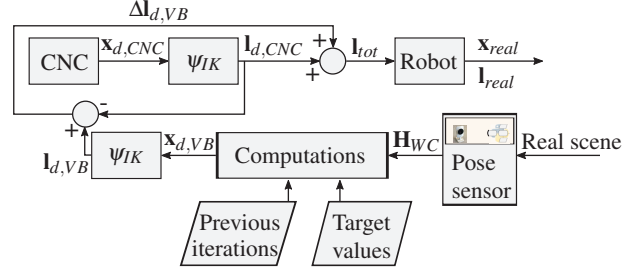


Fig. 5: A block scheme of one correction iteration

all previous iterations, and the last input  $\{\mathbf{r}_{d,CNC}\}_O$ , given before activating the vision-based control:

$$\{\mathbf{r}_\sigma\}_O = \{\mathbf{r}_{d,CNC}\}_O - \sum_{s=1}^{\sigma-1} \{\Delta\mathbf{c}_s\}_O - \{\Delta\mathbf{c}_\sigma\}_O . \quad (14)$$

After the computation of orientation and position of the platform, in Eqs. (14) and (12), cable lengths are calculated by inverse kinematics (see (2)). Figure 5 shows a block scheme of one iteration correction, according to the control structure of IPAnema 3. The final result of the vision-based control consists in a controlled change of cable lengths.

As a consequence of the very nature of the correction, the orientation correction must be as precise as possible to achieve an overall high-quality outcome. Thus, when both orientation and position need to be corrected, the orientation has the priority. To understand what occurs when a correction position is run with a relevant orientation error, it is sufficient to conceive the following scenario. Let us assume that the vector  $\mathbf{c}$  matches the target  $\mathbf{c}^*$ , but  $\mathbf{R}_{GP'}$  does not. The correction position module would end the correction. However, because of the finite size of  $\mathbf{p}$ , the origin of  $\mathcal{H}_P$  is not in the desired position yet. Referring to Eq. (4) and Fig. 4, the origin of  $\mathcal{H}_P$  expressed w.r.t  $\mathcal{H}_O$  is:

$$\begin{aligned} \{\mathbf{r}\}_O &= \{\mathbf{g}\}_O - \{\mathbf{c}^*\}_O - \{\mathbf{p}\}_O \\ &= \{\mathbf{g}\}_O - \mathbf{R}_{OG}\{\mathbf{c}^*\}_G - \mathbf{R}_{OG}\mathbf{R}_{GP'}\mathbf{R}_{P'P}\{\mathbf{p}\}_P \\ &\neq \{\mathbf{g}\}_O - \mathbf{R}_{OG}\{\mathbf{c}^*\}_G - \mathbf{R}_{OG}\mathbf{R}_{GP'}^*\mathbf{R}_{P'P}\{\mathbf{p}\}_P. \end{aligned} \quad (15)$$

Equation (15) shows the consequent change in position of the origin of  $\mathcal{H}_P$  due to an orientation error ( $\mathbf{R}_{GP'}^* \neq \mathbf{R}_{GP'}$ ).

## 4 Implementation

### 4.1 System architecture

The architecture of the system implemented for the experimental validations is illustrated in Fig. 6. As for the imaging device, an inexpensive wireless IP camera DCS-935L (by D-Link) was employed. The camera communicates with the router TL-WDR4900 (by TP-link) which forwards the data to the control PC via Ethernet cable. By means of a python script, the MJPG video streamed over an *http* protocol is accessed. The JPG frames are sent via this protocol and then decoded by the use of OpenCV library functions. The communication between the non real-time applications and

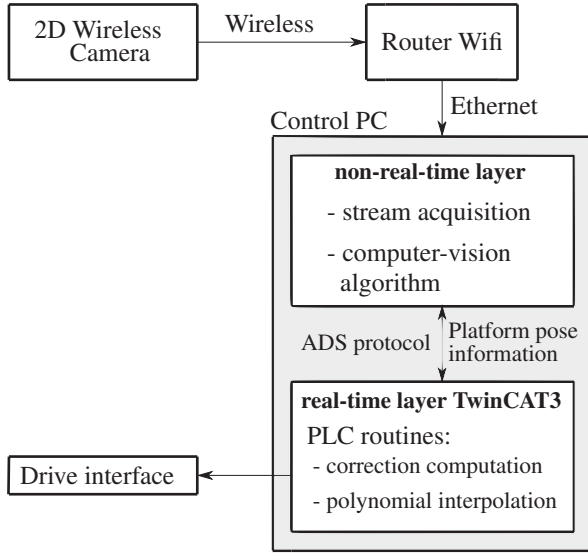


Fig. 6: System architecture - The real-time control is based on a Computerized Numerical Control block (CNC) and a Programmable Logic Controller (PLC). Through the latter, it is possible to process data from external sensors and feed corrections to the drive interface. The communication is realized using an EtherCAT bus with a 1 ms cycle time.

the real-time robot control based on TwinCAT3 is made by virtue of the ADS protocol [22]. The information provided through the aforementioned transport layer is elaborated in a PLC routine programmed in Structured Text. Its final output is sent to the drive interface.

## 4.2 Optical pose sensor

**Camera calibration** The camera was calibrated in order to determine its intrinsic parameters. By means of a known planar chessboard pattern, made up of a 10x7 grid with square side of 26.1 mm, Zhang's method [23] enriched by Bouget's<sup>3</sup> work was utilized. The algorithm is based on solving an optimization problem of the 3D-2D correspondence described in Eq. (3) for each view of the pattern. This technique was implemented through a python script with a considerable use of the OpenCV library.

**Pose detection** A python script was conceived to provide the needed measurements. Its core is the solution of the Perspective-n-Point problem [24]. By this python script, the calibrated camera can be thought of as a pose sensor, whose output is the inverse of matrix  $\mathbf{H}_{CW}$  in Eq. (3). The user-defined coordinate frame was defined through the same chessboard pattern used for camera calibration. This choice is not compelling. Indeed, through further computer vision elaborations, a coordinate frame can also be defined by using the features of the surrounding environment.

<sup>3</sup>Current information about J.Y. Bouget's work are available at [http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/)

The camera was mounted in a location from which it could frame the chessboard pattern at a distance of  $400 \div 600$  mm (i.e. between 400 and 600 mm), which represents a compromise between the requirements of the application and the camera resolution. In general, the closer the camera to the chessboard pattern, the higher the accuracy of the correction imparted by the control: as a result, less iterations are needed to reach convergence.

However, the accuracy of the developed system is intrinsically low, because of the inexpensive camera hardware, but also because the measurement system is not based on a stereo principle. The same cannot be said for repeatability. The latter was assessed to be, in the given working conditions, in the order of magnitude of 0.1 mm for each position component, and  $0.01 \div 0.1^\circ$  for each orientation component.

A digital low-pass WMA (Weighted Moving Average) filter was applied to the pose-sensor output to reduce noise. Indeed, measurements are taken in quasi-static conditions (low-band signals) and, as a consequence, the WMA is effective against noise (high-band signal) without losing important information.

The time needed by the system to provide a measurement is in the order of magnitude of  $0.01 \div 0.1$  s. The frame rate of the camera (30 frames per second) was considered to take into account a sufficient number of measurements for the WMA. A higher frame rate of the camera would guarantee an increase in the procedure performance.

## 4.3 Additional parameters

In the experiments presented in Section 5, the duration of the measurement phase was set to 2.5 s, as the average delay due to the wireless communication was about 1.5 s. The duration of the motion phase varies from 0.2 to 1 s, depending upon the amplitude of the motion to be executed. In order to deem whether the actual orientation has reached the target one, each Euler angle extracted by  $\mathbf{E}^T$  is checked to be below  $0.01^\circ$ . To judge the actual position as the target position, each component of the error vector  $\{\Delta \mathbf{c}\}_O$  is verified to be below 0.1 mm. Moreover, for safety reasons and for limiting the peaks in the motion profile, the maximum amplitude of the correction in one iteration is limited to 5 mm for each component of  $\{\Delta \mathbf{c}\}_O$  and  $3^\circ$  for the elementary rotation.

## 5 Experimental validation

Two examples are presented hereafter to show the effectiveness of the referencing procedure: matching the platform frame  $\mathcal{H}_P$  to the robot fixed frame  $\mathcal{H}_O$  (*homing*), and referring the platform to a coordinate frame unrelated to the robot.

### 5.1 Homing procedure

Here, the camera is fixed w.r.t.  $\mathcal{H}_O$  and its location is identified by  $\mathcal{H}_G$ . The pattern is mounted on the platform and its location is identified by  $\mathcal{H}_P$ . Firstly, the camera performs a unique measurement of the ideal home pose, right after the calibration of the robot is done. Successively, every time a new referencing is needed, the actual reached home pose is



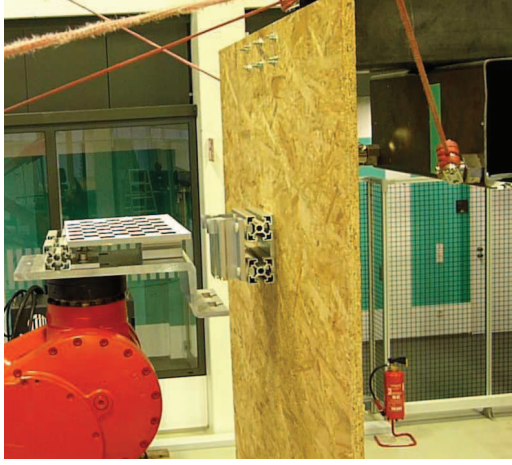


Fig. 7: Demonstration of a façade panel installation by a cable robot endowed with a vision-based external feedback

measured through the camera and compared to the ideal one for the correction.

The procedure accuracy was evaluated by means of a Leica Absolute Laser Tracker, by comparing the ideal home pose (defined during the calibration) with the one reached after the commanded corrections [25]. An array  $\varepsilon$  is stored expressing the error of the home pose in position (mm) and orientation ( $^\circ$ ).  $\varepsilon$  is measured before and after executing the procedure. Then, standard deviation is used to express the scattering of data w.r.t. the mean value. The values of  $\varepsilon$  before ( $\varepsilon_b$ ) and after ( $\varepsilon_a$ ) the homing procedure are, respectively:

$$\varepsilon_b = \begin{bmatrix} 0.445 \pm 1.790 \\ -1.195 \pm 0.450 \\ -9.890 \pm 8.854 \\ -0.2021 \pm 0.2988 \\ 0.1304 \pm 0.1169 \\ 0.3084 \pm 0.3490 \end{bmatrix}; \varepsilon_a = \begin{bmatrix} 0.229 \pm 0.107 \\ -0.270 \pm 0.022 \\ -0.176 \pm 0.136 \\ 0.0055 \pm 0.0105 \\ 0.0020 \pm 0.0208 \\ 0.0181 \pm 0.0122 \end{bmatrix} \quad (16)$$

where the first three components correspond to position, and the following three to orientation. The results are remarkable, with an error norm in position of 0.395 mm and less than  $0.02^\circ$  in orientation. Data scattering is contained in narrow ranges (in the order of magnitude of 0.1 mm for position and  $0.01^\circ$  for orientation), which ensures excellent reliability of the procedure.

## 5.2 Improving accuracy in the task space

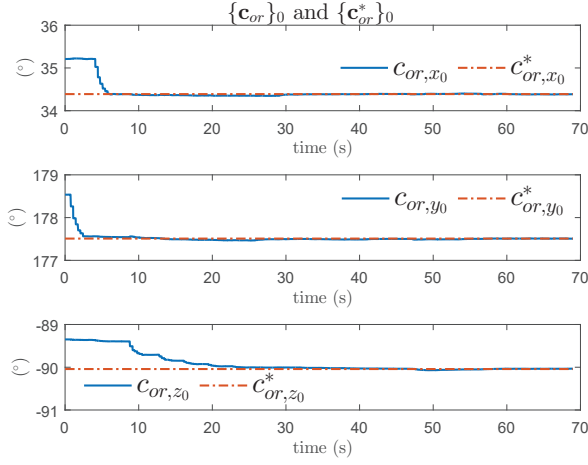
The application described in the following deals with automation in curtain wall installation and maintenance in high-rises [26, 27]. In this scenario, a façade is mainly constituted by almost identical elements that need identical operations for installation and maintenance. Thus, a known coordinate frame may be defined on every aluminium plate by detecting physical instances such as edges, so that the platform of the cable robot can be referenced to each plate in the same way, one after another, every operation. Also the curtain wall module may be detected and fixed to the platform

analogously, in order to consider the potential perturbations in the object placement). For testing the procedure, a simulation of a curtain wall module was executed. As shown in Fig. 7, an aluminium bracket that hosts the panel was attached to a large-scale serial robot, simulating the building. The camera was fixed to the platform. Thus, in this case,  $\mathcal{H}_G$  is the frame defined by the chessboard pattern, while  $\mathcal{H}_{p'}$  is integral with the camera. A first manual installation was performed, recording and storing the target pose, defined as the pose after which the curtain wall installation can be fulfilled by a small translation. Later, the platform was commanded to reach the neighbourhood of the aluminium bracket and the vision-based procedure was activated, allowing IPAnema 3 to attain the desired pose with a precision two order of magnitude greater than its own (which is in the order of magnitude of 10 mm). After the target pose was reached, a small translation provided by a G-Code command allowed installation to be completed. The demonstration of the installation of a façade panel was successfully performed several times. Figure 8 shows the corrections that gradually lead the platform to its target pose (where  $\{\mathbf{c}_{or}\}_O$  is the array containing the triplet of Euler angles corresponding to  $\mathbf{R}_{Op'}$ ). The movement phases can be easily recognized by the little steps in the graphs, consequence of the low measurement frequency. It is evident that the first iterations are dedicated to correct the orientation of the platform. Furthermore, very high resolution in correction is available from the camera measurement, as shown in Fig. 9. The overall average duration is in the order of one minute. However, it strongly depends on the set saturation values for the maximum allowed movement, the initial error, and the space in which the correction is performed. Further sources, such as photos, videos, and graphs are available as complementary material at [28].

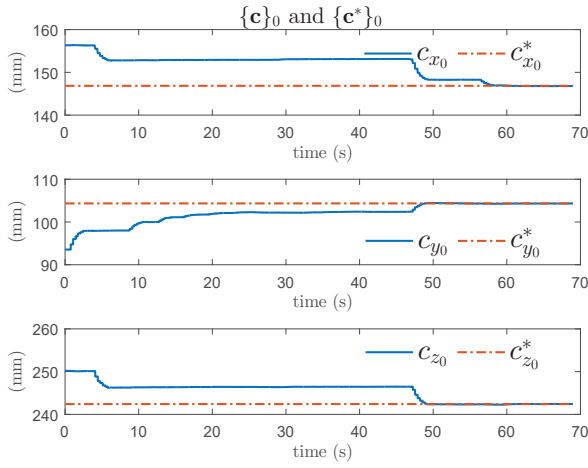
## 6 Conclusions

This paper presented a novel procedure for referring the pose of the platform of a cable-driven parallel robot with respect to a known coordinate frame. A look-and-move vision-based algorithm provides a correction to the pose of the robot attained through the model-based control, leading the evaluated pose error under a given threshold. The proposed procedure was tested on two different applications on the demonstrator IPAnema 3 available at Fraunhofer IPA. Its control architecture was studied in order to conceive control algorithms that could be implemented in the TwinCAT3 protocol. For this reason, the operational space measurement provided by the vision system was used to compute a correction to the commanded cable lengths. The experimental validation showed reliability and robustness of the computer vision algorithm, and precise computation of the platform desired pose.

Cable robots, endowed with external sensor feedback, were proved to be suitable for large scale applications where the accuracy of the platform must be preserved, as in the construction field. The vision-based feedback allowed corrections to be performed with high resolution, which made it possible to obtain better final pose accuracy compared to



(a)



(b)

Fig. 8: Time evolution of the measured arrays ( $\{\mathbf{c}_{or}\}_O$  and  $\{\mathbf{c}\}_O$ ) versus the target arrays ( $\{\mathbf{c}_{or}^*\}_O$  and  $\{\mathbf{c}^*\}_O$ ) during the vision-based control

most of the large-scale CDPRs developed so far. Eventually, this paper represents the first inquiry corroborated by experiments on the improvement of the performance of a large-scale CDPR by means of a vision system based on a look-and-move strategy. Further developments of this project will mainly be related to the speed of data transmission and the accuracy of the pose measurement.

### Acknowledgment

The authors would like to thank the European project Hephaestus (Grant Agreement No 732513) for providing the use-case scenario for testing the procedure described in Section 5.B, as well as Edoardo Idà and Christoph Martin for very useful discussions on the topic and co-supervising the Master Theses of the first two authors.

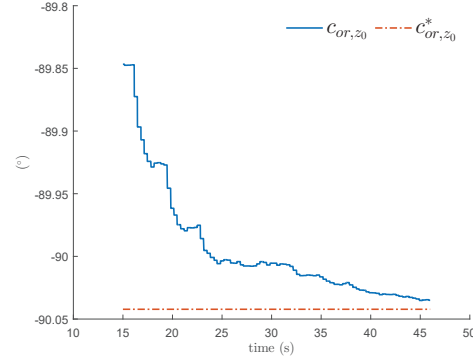


Fig. 9: Enlargement of rotation correction about the z axis

### Nomenclature

**A** Generic matrix.

**H<sub>AB</sub>** 4x4 matrix transforming the homogeneous coordinates of a point from  $\mathcal{K}_B$  to  $\mathcal{K}_A$ .

**I<sub>3x3</sub>** 3x3 identity matrix.

**K<sub>O</sub>** Coordinate frame attached to the fixed base at  $O$ .

**K<sub>P</sub>** Coordinate frame attached to the moving platform at  $P$ .

**K<sub>W</sub>** Generic coordinate frame attached at  $W$ .

**R** Rotation matrix describing the platform orientation.

**R<sub>AB</sub>** 3x3 rotation matrix transforming the array of components of a vector from  $\mathcal{K}_B$  to  $\mathcal{K}_A$ .

$a_{ij}$  Element placed at the  $i$ -th row and the  $j$ -th column of **A**.

$\{\tilde{\mathbf{p}}\}_W$  Homogeneous coordinates of point **p** in  $\mathcal{K}_W$ .

**r** Position vector of the platform reference point  $P$ .

**v** Generic geometric vector.

$\{\mathbf{v}\}_W$  Vector **v** expressed in  $\mathcal{K}_W$ .

$v_{xW}, v_{yW}, v_{zW}$  Components of vector **v** in  $\mathcal{K}_W$ .

**x** Six-element array describing the platform pose.

**0<sub>3x1</sub>** 3x1 null vector.

**A<sub>d,CNC</sub>**, **v<sub>d,CNC</sub>** Desired generic matrix or array, as computed by the CNC.

**A<sub>d,VB</sub>**, **v<sub>d,VB</sub>** Desired generic matrix or array, as computed by the visual-based control.

### References

- [1] Izard, J., Gouttefarde, M., Baradat, C., Culla, D., and Salle, D., 2013. "Integration of a parallel cable-driven robot on an existing building facade". In *Cable-Driven Parallel Robots*. Springer, pp. 149–164.
- [2] Gagliardini, L., Caro, S., Gouttefarde, M., Wenger, P., and Girin, A., 2015. "A reconfigurable cable-driven parallel robot for sandblasting and painting of large structures". In *Cable-Driven Parallel Robots*. Springer, pp. 275–291.
- [3] Schmidt, V., 2016. "Modeling Techniques and Reliable Real-Time Implementation of Kinematic for Cable-Driven Parallel Robots using Polymer Fiber Cables". PhD Thesis, Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen, Universität Stuttgart.

- [4] Corke, P., 1995. "Visual control of robot manipulators – a review". *Visual Servoing*, **7**, July.
- [5] Nigri, I., Meggiolaro, M. A., and Feitosa, R., 2009. "Comparison between look-and-move and visual servo control using sift transform in eye-in-hand manipulator system". In Proceedings of COBEM 2009 - 20th International Congress of Mechanical Engineering, Gramado-RS, Brazil, November 15-20, 2009.
- [6] Emmens, A., Spanjer, S., and Herder, J., 2014. "Modeling and control of a large-span redundant surface constrained cable robot with a vision sensor on the platform". *Mechanisms and Machine Science*, **32**, August.
- [7] Chellal, R., Cuvillon, L., and Laroche, E., 2015. "A kinematic vision-based position control of a 6-dof cable-driven parallel robot". In *Cable-Driven Parallel Robots*. Springer, pp. 213–225.
- [8] Begey, J., Cuvillon, L., Lesellier, M., Gouttefarde, M., and Gangloff, J., 2019. "Dynamic control of parallel robots driven by flexible cables and actuated by position-controlled winches". *IEEE Transactions on Robotics*, **35**, pp. 286–293.
- [9] Zake, Z., Chaumette, F., Pedemonte, N., and Caro, S., 2019. "Vision-based control and stability analysis of a cable-driven parallel robot". *IEEE Robotics and Automation Letters*, **4**(2), April, pp. 1029–1036.
- [10] Dallej, T., Gouttefarde, M., Andreff, N., Dahmouche, R., and Martinet, P., 2012. "Vision-based modeling and control of large-dimension cable-driven parallel robots". In IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal, October 7-12, 2012, pp. 1581–1586.
- [11] Dallej, T., Gouttefarde, M., Andreff, N., Hervé, P.-E., and Martinet, P., 2019. "Modeling and vision-based control of large-dimension cable-driven parallel robots using a multiple-camera setup". *Mechatronics*, **61**, May, pp. 20 – 36.
- [12] Nguyen, D. Q., Gouttefarde, M., Company, O., and Pierrot, F., 2013. "On the simplifications of cable model in static analysis of large-dimension cable-driven parallel robots". In IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, November 3-7, 2013, pp. 928–934.
- [13] Pott, A., Mütterich, H., Kraus, W., Schmidt, V., Miermeister, P., Dietz, T., and Verl, A., 2013. "IPAnema: A family of cable-driven parallel robots for industrial applications". In *Cable-Driven Parallel Robots*, Springer, pp. 119–134.
- [14] Tempel, P., Miermeister, P., and Pott, A., 2015. "Kinematics and dynamics modeling for real-time simulation of the cable-driven parallel robot ipanema 3". In Proceedings of the 14th IFToMM World Congress, Taipei, Taiwan, October 25-30, 2015, pp. 117–123.
- [15] Kraus, W., 2015. "Force Control of Cable-Driven Parallel Robots". PhD Thesis, Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen, Universität Stuttgart.
- [16] Hartley, R., and Zisserman, A., 2003. *Multiple View Geometry in Computer Vision*. Cambridge University Press.
- [17] Heikkilä, J., and Silven, O., 1997. "A four step camera calibration procedure with implicit image correction". In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, USA, June 17-19, 1997, pp. 1106–1112.
- [18] Levenberg, K., 1944. "A method for the solution of certain non-linear problems in least squares". *The Quarterly of Applied Mathematics*, **2**, pp. 164–168.
- [19] Marquardt, D. W., 1963. "An algorithm for least-squares estimation of nonlinear parameters". *Journal of the Society for Industrial and Applied Mathematics*, **11**(2), pp. 431–441.
- [20] Lourakis, M., 2005. "A brief description of the levenberg-marquardt algorithm implemented by levmar". *Foundation of Research and Technology*.
- [21] Chianura, M., 2018. "Vision-based enhancement of the pose accuracy of a redundantly actuated cable-driven parallel robot". Master Thesis, Alma Mater Studiorum - University of Bologna.
- [22] Beckhoff fieldbus components, fieldbussystem, ADS communication. <https://infosys.beckhoff.com/>.
- [23] Zhang, Z., 2000. "A flexible new technique for camera calibration". *IEEE Transactions on pattern analysis and machine intelligence*, **22**(11), pp. 1330–1334.
- [24] Wu, Y., and Hu, Z., 2006. "Pnp problem revisited". *Journal of Mathematical Imaging and Vision*, **24**(1), pp. 131–141.
- [25] Zavatta, M., 2018. "Vision-based homing procedure for an overconstrained cable-driven parallel robot". Master Thesis, Alma Mater Studiorum - University of Bologna.
- [26] Hephaestus european project. <http://www.hephaestus-project.eu/>.
- [27] Taghavi, M., Heredia, H., Iturralde, K., Halvorsen, H., and Bock, T., 2018. "Development of a modular end effector for the installation of curtain walls with cable-robots". *Journal of Facade Design and Engineering*, **6**, 06, pp. 1–8.
- [28] [https://www.dropbox.com/sh/pa4g3b6zicqbbcf/AADPd6R6sWX7NWYF1v8\\_Y8Zka?dl=0](https://www.dropbox.com/sh/pa4g3b6zicqbbcf/AADPd6R6sWX7NWYF1v8_Y8Zka?dl=0).

## List of Figures

1	An optical device mounted on IPAnema 3 platform, allowing the implementation of a vision-based system for improving accuracy . . . . .	2
2	Representation of CDPR kinematic parameters . . . . .	3
3	Pinhole camera model parameters and transformations among the three coordinate frames . . . . .	3
4	The four coordinate frames involved in the referencing procedure . . . . .	4
5	A block scheme of one correction iteration . . . . .	5
6	System architecture - The real-time control is based on a Computerized Numerical Control block (CNC) and a Programmable Logic Controller (PLC). Through the latter, it is possible to process data from external sensors and feed corrections to the drive interface. The communication is realized using an EtherCAT bus with a 1 ms cycle time. . . . .	6
7	Demonstration of a façade panel installation by a cable robot endowed with a vision-based external feedback	7
8	Time evolution of the measured arrays ( $\{\mathbf{c}_{or}\}_O$ and $\{\mathbf{c}\}_O$ ) versus the target arrays ( $\{\mathbf{c}_{or}^*\}_O$ and $\{\mathbf{c}^*\}_O$ ) during the vision-based control . . . . .	8
9	Enlargement of rotation correction about the $z$ axis . . . . .	8