



**HAL**  
open science

# Constant Distance and Orientation Following of an Unknown Surface with a Cable-Driven Parallel Robot

Thomas Rousseau, Nicolò Pedemonte, Stéphane Caro, François Chaumette

► **To cite this version:**

Thomas Rousseau, Nicolò Pedemonte, Stéphane Caro, François Chaumette. Constant Distance and Orientation Following of an Unknown Surface with a Cable-Driven Parallel Robot. ICRA 2023 - IEEE International Conference on Robotics and Automation, IEEE, May 2023, London, United Kingdom. pp.1-7. hal-04025937

**HAL Id: hal-04025937**

**<https://hal.science/hal-04025937>**

Submitted on 13 Mar 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Constant Distance and Orientation Following of an Unknown Surface with a Cable-Driven Parallel Robot

Thomas Rousseau<sup>1</sup>, Nicolò Pedemonte<sup>2</sup>, Stéphane Caro<sup>3</sup>, François Chaumette<sup>4</sup>

**Abstract**—Cable-Driven Parallel Robots (CDPRs) are well-adapted to large workspaces since they replace rigid links by cables. However, they lack in positioning accuracy and new control methods are necessary to achieve profile-following tasks. This paper presents a control scheme designed for these tasks, relying on a combination of accurate boarded distance sensors and of a less accurate remote camera. The profile-following task is divided into two subtasks that are partially conflicting: maintaining a parallel orientation and a constant distance with the surface to follow, and following a trajectory between two points on the surface. The data fusion to solve the redundancy is based on the Gradient Projection Method. This control scheme is validated experimentally on a CDPR prototype and shown to provide the expected behaviour.

## I. INTRODUCTION

Cable-Driven Parallel Robots (CDPRs) are robots able to carry loads in large workspaces, and thus hold a potential for industrial applications in large assembly operations, for instance in the aeronautics or naval sectors. However, these robots are still lacking in precision, including in tasks requiring an accurate positioning with regards to a given object. This adds to the complexity of profile-following tasks. Most of the research works focus on improving the CDPR model, especially regarding cable elasticity and sagging [1], [2], [3], [4], [5], [6], [7] but other approaches such as vision-based control have also been proposed to improve the precision of CDPRs [8], [9]. The robustness of trajectory planning and tracking for CDPRs was evaluated in [10].

As the accuracy of vision-based pose estimation decreases with the distance to the cameras, external vision-based systems have limited performance in large workspaces. For large-workspace applications, this results in often needing multiple and high-resolution cameras, costly systems that require a high computing power to operate in real-time. In order to achieve precise relative positioning of the robot with regard to a given object, both camera/robot and camera/object poses need to be measured precisely. Sensors boarded on the end-effector (EE) allow a more accurate relative positioning with respect to such an object. For

instance, visual servoing was used in [11] to perform an unknown profile following task with a camera mounted on the EE of a robot arm. This method however requires that three parallel curves to follow are drawn on the surface. Untextured surfaces are common in industrial applications, and other solutions such as using structured light [12] or distance sensors [13] can prove effective in this case. These boarded sensors cannot however position the moving-platform (MP) of a CDPR in a large workspace alone, because of their restrained range and field of view. Conversely, one or multiple camera(s) fixed in the base reference frame can observe a large workspace and estimate the MP pose, with a lesser accuracy. Combining boarded sensors and external cameras allows to benefit from the advantages of both local and precise measurements around the end-effector and of less precise measurements but a global view of the workspace. This combination enables profile following tasks. However, the two information sources are redundant and may conflict with each other. A solution to correctly combine their measurements is thus necessary. Multiple sensor fusion approaches to solve the redundancy exist [14], notably the well-documented Gradient Projection Method (GPM) designed for prioritizing tasks [15], [16], [17]. In particular, this method is used in [18] to perform a cooperation between eye-to-hand and eye-in-hand cameras to control the pose of a robot EE.

The purpose of this work is to design a control architecture able to closely follow at a constant distance and relative orientation the profile of a curved surface without any prior information on its shape. The available information when launching the robot is limited to the current location of the MP, a starting point and a target point, given by a remote vision system observing hand-placed AprilTags marking these points in the robot base reference frame. The trajectory following has to be played between these starting and target points. During the profile following, three boarded distance sensors and the remote camera provide the system with three distances to the surface and the MP pose in the robot base frame. In order to perform the task, an appropriate data fusion method based on the GPM is implemented to combine the measurements from both sensors. While this work was designed for and experimented with a CDPR, the results are applicable to any kind of robots.

The robot control strategy is presented in Section II: two subtasks are first presented and the fusion using GPM is then detailed. In Section III, an experiment designed to assess the performance of this control scheme is presented. The results are analyzed and discussed, and conclusions are drawn in Section IV.

\*This work is supported by IRT Jules Verne in the framework of the PERFORM project.

<sup>1</sup> Thomas Rousseau works at IRT Jules Verne and is a member of Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, Nantes, France [thomas.rousseau@ls2n.fr](mailto:thomas.rousseau@ls2n.fr)

<sup>2</sup> Nicolò Pedemonte is with IRT Jules Verne, Nantes, France [Nicolo.Pedemonte@irt-jules-verne.fr](mailto:Nicolo.Pedemonte@irt-jules-verne.fr)

<sup>3</sup> Stéphane Caro is with CNRS at LS2N, Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, Nantes, France [Stephane.Caro@ls2n.fr](mailto:Stephane.Caro@ls2n.fr)

<sup>4</sup> François Chaumette is with Inria, Univ Rennes, CNRS, IRISA, Rennes, France [Francois.Chaumette@inria.fr](mailto:Francois.Chaumette@inria.fr)

## II. ROBOT CONTROL

The profile following task can be divided into two sub-tasks: maintaining the MP parallel to the followed surface at a constant distance and moving from the starting pose to the target pose. The control of the robot stems from sensor-based control, especially visual servoing, and combines measurements provided by two types of sensors: an array of three distance sensors boarded on the MP, and an external camera placed in front of the robot. These measurements allow to complete the two subtasks that need to be conciliated.

The MP to control and the reference frames used in this work are shown in Fig. 1. The three distance sensors  $S_1, S_2, S_3$  and the external camera are also represented. Several AprilTags are placed in the workspace. The pose of the reference tag in  $\mathcal{F}_c$  is measured precisely offline in order to compute the transformation matrix from  $\mathcal{F}_c$  to  $\mathcal{F}_w$ . The AprilTag placed on the MP allows its pose estimation in  $\mathcal{F}_c$  and then in  $\mathcal{F}_w$ . The trajectory start and trajectory target tags mark the points between which the profile following is played. The trajectory planner computes a straight line trajectory between the starting pose of the platform, above the start tag, towards the target tag. The robot then has to follow this trajectory while also respecting the distance and orientation constraints above the ground then above the curved surface to follow, as shown in Fig. 1.

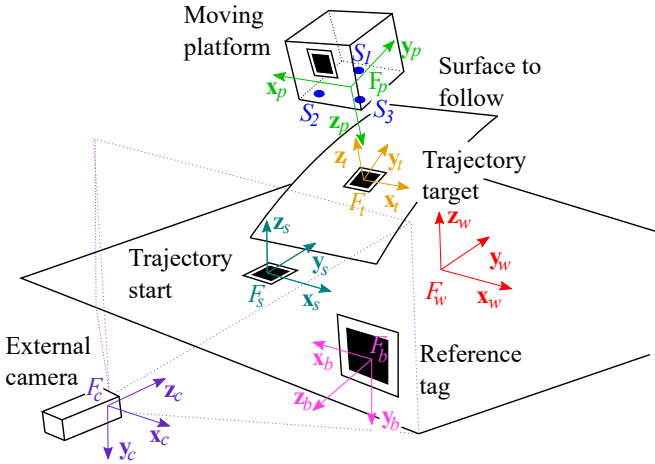


Fig. 1: Parametrization and main components of the robotic platform. The AprilTags are represented by black squares.

### A. Sensors

1) *Distance sensors*: Three SICK UM18 ultrasonic distance sensors are placed below the MP, facing downwards along the  $z$ -axis of  $\mathcal{F}_p$ . These sensors have a resolution of 0.2 mm and an accuracy at 200 mm (working distance) of 0.3 mm. They provide the system with three distance measurements each 80 ms. Their current output is processed by an analogic-digital converter (ADC) connected to the robot control. The ADC introduces a quantum error corresponding to  $\sigma_{s,q} = 0.04$  mm. Their calibration also introduces an error, estimated to 2 mm. The combined distance error for a distance sensor is then estimated to  $\sigma_d = 2.01$  mm.

The relative distance between the three sensors was chosen to minimize the perturbations from reflections. Since all sensor directions are parallel, for the MP to be locally parallel to the surface, the three sensors must return the same distance (see Fig. 2).

In the eye-in-hand configuration used for the boarded sensors, the time variation of the measured distance is given by the proximity sensor model presented in [17]:

$$\dot{s}_i = -\frac{1}{\mathbf{n}_{T_i} \cdot \mathbf{n}_{S_i}} (\mathbf{n}_{T_i} \cdot \boldsymbol{\nu}_S - ([s_i \mathbf{n}_{S_i}]_{\times} \mathbf{n}_{T_i}) \cdot \boldsymbol{\omega}_S) \quad (1)$$

with  $\mathbf{n}_{T_i}$  the unit vector normal to the surface at the target feature point and  $\mathbf{n}_{S_i}$  the unit vector defining the axis of the sensor (see Fig. 2).  $\boldsymbol{\nu}_s$  is the linear velocity of the sensor and  $\boldsymbol{\omega}_s$  is its angular velocity, both expressed in the sensor frame. Note that due to the chosen configuration of the sensors,  $\mathbf{n}_{S_i} = (0, 0, 1)$  in  $\mathcal{F}_p, i = 1, \dots, 3$ .

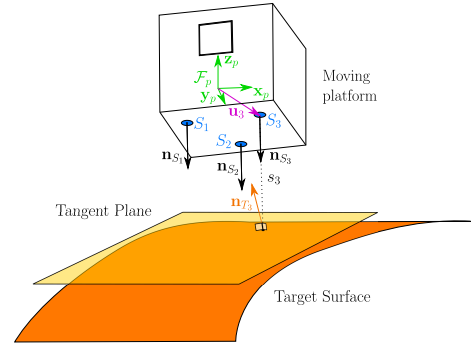


Fig. 2: Sensor model: only the bottom plane of the MP is shown.

Knowing the position  $\mathbf{O}_p \mathbf{S}_i = (x_i, y_i, 0)$  of the sensors in  $\mathcal{F}_p$ , it is possible to express their time variation in function of the instantaneous velocity  $\mathbf{v}_p$  of the MP, expressed in  $\mathcal{F}_p$  from Eq. (1):

$$\dot{s}_i = -\frac{1}{\mathbf{n}_{T_i} \cdot \mathbf{n}_S} (\mathbf{n}_{T_i} \cdot \mathbf{v}_p - ([s_i \mathbf{n}_S + \mathbf{O}_p \mathbf{S}_i]_{\times} \mathbf{n}_{T_i}) \cdot \boldsymbol{\omega}_P) \quad (2)$$

for  $i = 1, \dots, 3$ . Doing so, the feature variation, noted  $\dot{\mathbf{s}}$  where  $\mathbf{s} = (s_1, s_2, s_3)$  is related to the instantaneous velocity  $\mathbf{v}_p = (\mathbf{v}_p, \boldsymbol{\omega}_p)$  of the MP expressed in  $\mathcal{F}_p$  through:

$$\dot{\mathbf{s}} = {}^p \mathbf{L}_1 \mathbf{v}_p \quad (3)$$

${}^p \mathbf{L}_1$  is the interaction matrix of the distance sensors [17], [19]. Eq. (2) can be reshaped into the form of (3) and evaluated for the particular case where  $\mathbf{s} = \mathbf{s}^*$ , for which  $\mathbf{n}_{T_i} = (0, 0, 1)$  in  $\mathcal{F}_p$  for  $i = 1, \dots, 3$ :

$${}^p \mathbf{L}_1 |_{\mathbf{s}=\mathbf{s}^*} = \begin{bmatrix} 0 & 0 & -1 & -y_1 & x_1 & 0 \\ 0 & 0 & -1 & -y_2 & x_2 & 0 \\ 0 & 0 & -1 & -y_3 & x_3 & 0 \end{bmatrix} \quad (4)$$

Using a constant value of the interaction matrix corresponding to the desired configuration is indeed a usual choice in sensor-based control [19].

2) *External camera*: The external camera is an IDS uEye 3240CP with a resolution of  $640 \times 480$  px at 30 images per second. The images acquired are processed by an algorithm recognising the AprilTags and estimating their relative pose with regard to the camera frame  $\mathcal{F}_c$ . April Tags were chosen because they offer a simple solution for pinpointing locations on an unknown and untextured surface. Then, this pose is expressed in  $\mathcal{F}_w$ , and, as for Pose-Based Visual Servoing (PBVS), the visual features are selected as:

$$\mathbf{s}_v = \begin{pmatrix} {}^w \mathbf{t}_p \\ \theta \mathbf{u} \end{pmatrix} \quad (5)$$

where  ${}^w \mathbf{t}_p$  is the position of the platform in the world reference frame and  $\theta \mathbf{u}$  is the axis-angle representation of the rotation matrix  ${}^p \mathbf{R}_p$ . The target pose is defined by:

$$\mathbf{s}_v^* = \begin{pmatrix} {}^w \mathbf{t}_p^* \\ \mathbf{0}_3 \end{pmatrix} \quad (6)$$

where  ${}^w \mathbf{t}_p^*$  is the position of the target AprilTag in  $\mathcal{F}_w$ . The interaction matrix of these visual features is well known and given by [19]:

$${}^w \mathbf{L}_2 = \begin{bmatrix} \mathbf{I}_3 & -[{}^w \mathbf{t}_p]_{\times} \\ \mathbf{0}_3 & \mathbf{L}_\omega \end{bmatrix} \quad (7)$$

with  $[{}^w \mathbf{t}_p]_{\times}$  the skew-symmetric matrix of  ${}^w \mathbf{t}_p$ ,  $\mathbf{I}_3$  the three-dimensional identity, and  $\mathbf{L}_\omega$  can be approximated by  $\mathbf{I}_3$ . This interaction matrix is then expressed in  $\mathcal{F}_p$  with:

$${}^p \mathbf{L}_2 = {}^w \mathbf{L}_2 {}^c \mathbf{V}_p \quad (8)$$

such that:

$${}^w \mathbf{V}_p = \begin{bmatrix} {}^w \mathbf{R}_p & [{}^w \mathbf{t}_p]_{\times} {}^w \mathbf{R}_p \\ \mathbf{0}_3 & {}^w \mathbf{R}_p \end{bmatrix} \quad (9)$$

The accuracy of the pose estimation by the vision system depends on several factors: the resolution of the camera, but also the size and the pose of the observed object. Here, all AprilTags have a side of 80 mm, except for the one attached to the MP, which is 64 mm, while the observed objects are placed within a range of 1–1.5 m away from the camera. Angles between the image plane and the AprilTags are comprised between  $-90^\circ$  and  $+10^\circ$ . The accuracy along the optical axis of the camera is worse because the depth estimation is based on the perceived size of the tracked object, accumulating the errors of several measures. The depth error increases with the distance. In the working area, the accuracy of the position estimation (neglecting the rotation errors) was measured using a Creaform C-Track (see Section III-A) to  $\sigma_v = 70$  mm.

### B. Choice of a fusion method

On one hand, performing the first task with the three distance sensors locks three degrees of freedom (DoF) : two rotations and one translation. On the other hand, controlling the MP pose with the remote camera, required to complete the trajectory tracking involves six DoF. Since the MP has only  $n = 6$  DoF, the degree of measurement redundancy  $r$  is equal to three considering all available measurements. Various sensor fusion techniques allow to deal with this

redundancy, and two main approaches are available: feature-level fusion (fusing errors) and control-level fusion (fusing velocities).

The respective precision of the two sensor sets needs to be taken into account by the fusion algorithm. For instance, inverse variance weighting gives for the  $z_p$  coordinate of the platform:

$$z_p = (\sigma_d^{-2} z_{p,d} + \sigma_v^{-2} z_{p,v}) / \sigma^2 \quad (10)$$

with  $\sigma^2 = (\sigma_d^{-2} + \sigma_v^{-2})$ . The variances can be calculated from the precision of the distance sensors and the vision-based pose estimation.  $\sigma$  is the standard deviation of the combined measurements:

$$\sigma_d = 2.0 \text{ mm}; \quad \sigma_v = 70 \text{ mm}; \quad \sigma = 4.0 \text{ mm} \quad (11)$$

While values of  $z_{p,d}$  and  $z_{p,v}$  have the same order of magnitude, it follows that  $\sigma_d^{-2} z_{p,d} \gg \sigma_v^{-2} z_{p,v}$  and the resulting standard deviation  $\sigma$  is twice superior to the theoretical precision  $\sigma_d$  when taking into account only the distance sensors. Combining information with such a technique is thus detrimental to the precision of the system and a strict decoupling of the two sensor sets is more adapted to control this DoF with the best precision possible. In other words, the vision input can be neglected for this component  $z_p$ , which is equivalent to using binary weights (1 for  $z_{p,d}$  and 0 for  $z_{p,v}$ ). This is the same for the other two rotational DoFs that can be controlled by the three distance sensors. The same reasoning is valid for a fusion at the control level. The best is thus to use the well-known Gradient Projection Method that is perfectly suited in this case. Indeed, it consists in projecting a secondary task on the kernel of a main task, which ensures that the main task will not suffer from any perturbation induced by the secondary task.

### C. Fusion using the Gradient Projection Method

With this method, the main task  $\mathcal{T}_1$  is considered as a priority, while only the part of the secondary task  $\mathcal{T}_2$  compatible with the realization of the main one is retained.

We naturally select for the main task the measurements  $\mathbf{e}_1$  provided by the distance sensors, that is:

$$\mathbf{e}_1 = (s_1 - s^*, s_2 - s^*, s_3 - s^*)^T \quad (12)$$

where  $s_i$  is the distance measured by the sensor  $S_i$  and  $s^*$  is the common reference distance so that the MP is locally parallel to the followed surface (in practice, we select  $s^* = 0.2$  m).

The next step is to determine the Jacobian of the main task. As already seen in Section II-A.1, the time variation of the main task is related to the MP twist  $\mathbf{v}_p$  through (3). As the inputs of the low-level controller of the CDPR are the velocities  $\dot{\mathbf{l}}$  of the  $m$  cable lengths, we use:

$$\dot{\mathbf{l}} = \mathbf{A} \mathbf{v}_p \quad (13)$$

where the forward Jacobian matrix  $\mathbf{A}$  of the CDPR is given by [20]:

$$\mathbf{A} = \begin{bmatrix} \mathbf{u}_1 & \dots & \mathbf{u}_i & \dots & \mathbf{u}_m \\ \mathbf{b}_1 \times \mathbf{u}_1 & \dots & \mathbf{b}_i \times \mathbf{u}_i & \dots & \mathbf{b}_m \times \mathbf{u}_m \end{bmatrix}^T \quad (14)$$

where  $\mathbf{u}_i$  are the cable direction unit vectors, pointing from the exit points (pulleys) of the CDPR to the anchor points placed on the MP, and where  $\mathbf{b}_i$  are the position vectors of the anchors points, known from the platform design. All these vectors are expressed in  $\mathcal{F}_p$ .  $\mathbf{u}_i$  vectors are calculated geometrically from the MP pose expressed in  $\mathcal{F}_w$  using the straight and inelastic cable model [20]. We recall that the MP pose in  $\mathcal{F}_w$  is estimated from its pose in  $\mathcal{F}_c$  and the known transform  ${}^w\mathbf{T}_b$  between the reference AprilTag (whose pose is also estimated in  $\mathcal{F}_c$ ) and  $\mathcal{F}_w$ .

Finally, from (3) and (13), we obtain:

$$\dot{\mathbf{e}}_1 = \mathbf{J}_1 \dot{\mathbf{i}} \quad (15)$$

where the Jacobian  $\mathbf{J}_1$  of the main task is given by

$$\mathbf{J}_1 = {}^p\mathbf{L}_1 \mathbf{A}^+ \quad (16)$$

$\mathbf{A}^+$  being the pseudo-inverse of  $\mathbf{A}$ .

Regarding the secondary task  $\mathcal{T}_2$ , it is given by:

$$\mathbf{e}_2 = \mathbf{s}_v - \mathbf{s}_v^* \quad (17)$$

where  $\mathbf{s}_v$  and  $\mathbf{s}_v^*$  are given by (5) and (6). From (8) and (13), we have:

$$\dot{\mathbf{e}}_2 = \mathbf{J}_2 \dot{\mathbf{i}} \quad (18)$$

where the Jacobian  $\mathbf{J}_2$  of  $\mathcal{T}_2$  is given by:

$$\mathbf{J}_2 = {}^p\mathbf{L}_2 \mathbf{A}^+ \quad (19)$$

In practice, the exact value of  $\mathbf{J}_1$  and  $\mathbf{J}_2$  is not perfectly known, that is why approximations  $\hat{\mathbf{J}}_1$  and  $\hat{\mathbf{J}}_2$  have to be used in the control scheme. Doing so and applying the GPM, the cable velocity vector  $\dot{\mathbf{i}}$  to be sent to the low-level robot controller has the following form:

$$\dot{\mathbf{i}} = \hat{\mathbf{J}}_1^+ \dot{\mathbf{e}}_1 + \hat{\mathbf{P}}_1 \mathbf{z} \quad (20)$$

where  $\hat{\mathbf{P}}_1 = (\mathbf{I}_m - \hat{\mathbf{J}}_1^+ \hat{\mathbf{J}}_1)$  is a projection operator on the kernel of  $\hat{\mathbf{J}}_1$ , and  $\hat{\mathbf{P}}_1 \mathbf{z}$  is the part of the secondary task that can be realized without disturbing the main one. For considering the potential part of  $\mathcal{T}_2$  achieved by  $\mathcal{T}_1$ , the term  $\mathbf{z}$  can be chosen as [21]:

$$\mathbf{z} = (\hat{\mathbf{J}}_2 \hat{\mathbf{P}}_1)^+ (\dot{\mathbf{e}}_2 - \hat{\mathbf{J}}_2 \hat{\mathbf{J}}_1^+ \dot{\mathbf{e}}_1) \quad (21)$$

Since  $\hat{\mathbf{P}}_1$  is Hermitian and idempotent, we have  $\hat{\mathbf{P}}_1 (\hat{\mathbf{J}}_2 \hat{\mathbf{P}}_1)^+ = (\hat{\mathbf{J}}_2 \hat{\mathbf{P}}_1)^+$ . Eq. (20) can thus be rewritten as:

$$\dot{\mathbf{i}} = \hat{\mathbf{J}}_1^+ \dot{\mathbf{e}}_1 + (\hat{\mathbf{J}}_2 \hat{\mathbf{P}}_1)^+ (\dot{\mathbf{e}}_2 - \hat{\mathbf{J}}_2 \hat{\mathbf{J}}_1^+ \dot{\mathbf{e}}_1) \quad (22)$$

To end with the design of the control law, an exponential decoupled decrease of each error  $\mathbf{e}_i$  is specified, that is,  $\dot{\mathbf{e}}_i = -\lambda_i \mathbf{e}_i$ ,  $\lambda_i > 0$ ,  $i = \{1, 2\}$ . The control law can then be expressed in terms of  $\mathbf{e}_1$  and  $\mathbf{e}_2$ :

$$\dot{\mathbf{i}} = -\lambda_1 \hat{\mathbf{J}}_1^+ \mathbf{e}_1 - \lambda_1 (\mathbf{J}_2 \hat{\mathbf{P}}_1)^+ (r_\lambda \mathbf{e}_2 - \mathbf{J}_2 \hat{\mathbf{J}}_1^+ \mathbf{e}_1) \quad (23)$$

where  $r_\lambda = \lambda_2 / \lambda_1$  is the gain ratio. To simplify the notations, the term  $\tilde{\mathbf{e}}_2 = r_\lambda \mathbf{e}_2 - \mathbf{J}_2 \hat{\mathbf{J}}_1^+ \mathbf{e}_1$  and matrix  $\hat{\mathbf{J}}_2 = \mathbf{J}_2 \hat{\mathbf{P}}_1$  are defined such that the control law has finally the simple form (see Fig. 3):

$$\dot{\mathbf{i}} = -\lambda_1 (\hat{\mathbf{J}}_1^+ \mathbf{e}_1 + \hat{\mathbf{J}}_2^+ \tilde{\mathbf{e}}_2) \quad (24)$$

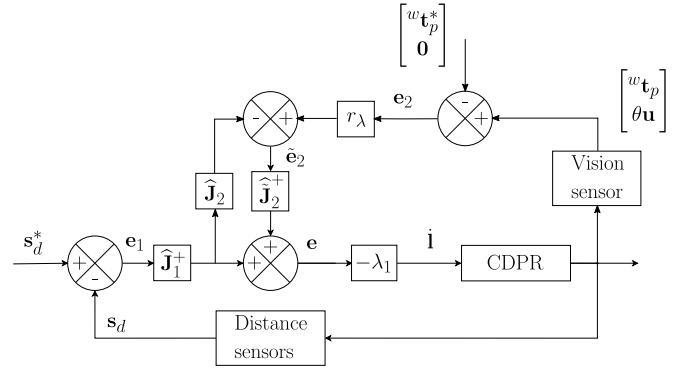


Fig. 3: Control scheme of the robot with the GPM.

Note that, in practice, reaching the desired pose  ${}^w\mathbf{t}_p^*$  is done by subdividing the trajectory from the starting pose to  ${}^w\mathbf{t}_p^*$  by a series of intermediate poses.

### III. EXPERIMENTS

#### A. Experimental setup

The robot used in the experimental platform is a six degrees of freedom Cable-Driven Parallel Robot named ACROBOT, located at IRT Jules Verne, Nantes, France. It consists of a cubic cell within a frame measuring  $1.2 \text{ m} \times 1.2 \text{ m} \times 1.2 \text{ m}$  in which a platform is suspended to 8 actuated cables. The moving-platform of the robot is  $0.18 \text{ m}$  long,  $0.18 \text{ m}$  wide and  $0.150 \text{ m}$  high. It boards 3 SICK UM18 ultrasonic distance sensors, placed on the bottom and facing downwards, measuring the distance between the MP and the surface below. The external camera is placed in front of the cell so as to observe the MP inside the robot workspace and estimate its pose using a set of AprilTags. The setup is shown Fig.4.

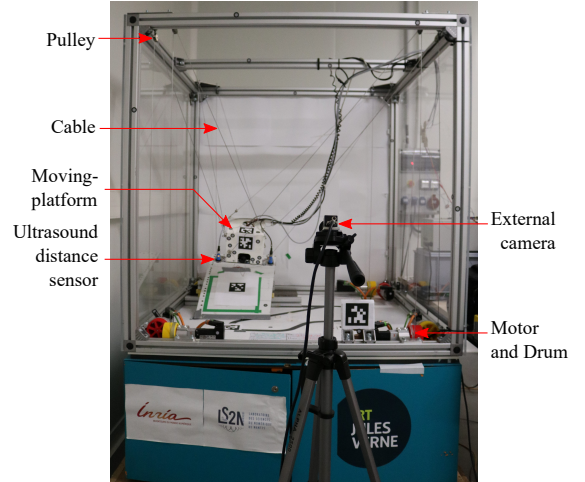


Fig. 4: The ACROBOT prototype, an eight-cable suspended CDPR located in IRT Jules Verne, Nantes, France.

The robot is controlled by a B&R automation interface connected to a PC where a ROS system computes and transmits the input velocities to the robot through an Ethernet link. Trajectories are generated with a trapezoidal speed

profile and, as already said, are composed of intermediary waypoints placed at each timestep (0.1 s) so that the desired path is a straight line. The starting point of the trajectory is located above an AprilTag placed on the bottom plane of the CDPR workspace and the target AprilTag is placed on an unknown piece of fuselage with a small curvature. The trajectory between these points allows testing the surface following and the steep change of curvature between two surfaces. A view of the MP above the curved surface at the end of the trajectory is shown Fig 5.

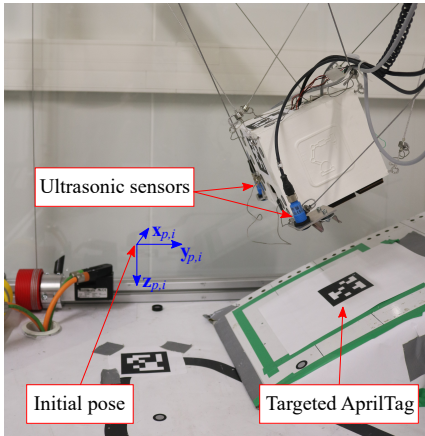


Fig. 5: View of the moving-platform reaching the targeted pose, marked with the AprilTag placed on the curved surface.

All the data exchanged between the various ROS control modules is recorded for the analysis, and the trajectories are also filmed using an additional camera. To assess the MP positioning accuracy obtained with the vision system, a Creaform C-Track device observes the scene and tracks the MP pose in the target frame  $\mathcal{F}_t$  (see Fig. 1). The measurement accuracy and repeatability of this optical metrology device are equal to 0.01 mm and 0.02 mm, respectively.

### B. Experimental results

The behavior of the robot with regard to the main task (profile following) can be assessed by observing the curves of the distance sensors output along the trajectory shown in Fig. 6.

From Fig. 6, it should be noted that  $s_1$  value decreases below the target when the MP starts crossing the threshold of the curved surface. The distances  $s_2$  and  $s_3$  briefly increase (peaks at  $t = 13$  s) while the MP is pivoting around its x-axis until all the sensors correctly face the surface to follow for the remainder of the trajectory ( $t \geq 15$  s). The settling time on the plane surface, measured between  $t = 2$  s and  $t = 8$  s, is  $\tau_r = 3.4$  s. Figure 6 shows that the control method is able to correctly solve the task conflict and to satisfy both the orientation and the distance constraints since all sensors measurements correctly converge to zero, namely,  $s - s^*_{|t \geq 17 s} = 0$ .

Results regarding the secondary task can be analyzed from Fig. 7. The Cartesian coordinates of the MP position measured by the external camera are expressed in the target

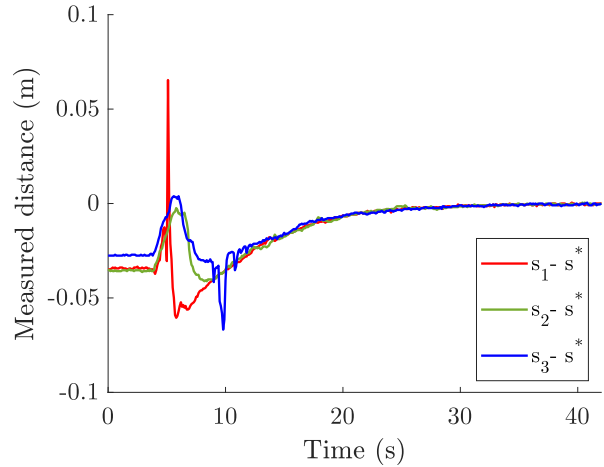
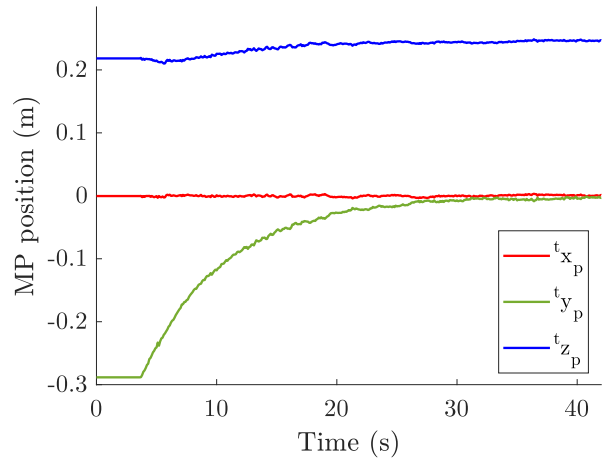
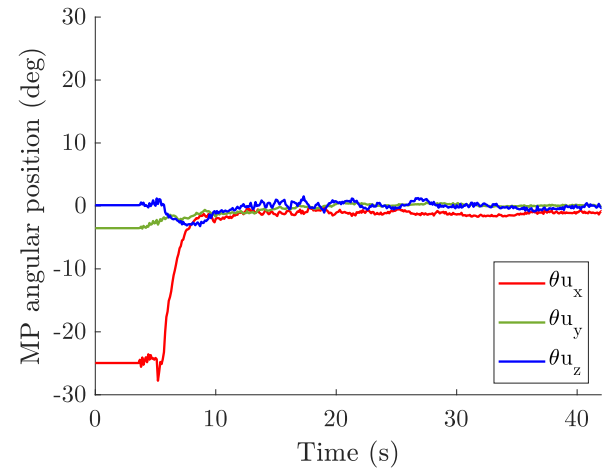


Fig. 6: Distance error  $e_1$  measured during a trajectory.



(a) Position of the MP in  $\mathcal{F}_t$ .



(b) Angular position (axis-angle) of the MP in  $\mathcal{F}_t$ .

Fig. 7: Translational and rotational displacements of the MP along a trajectory.



frame  $\mathcal{F}_t$ . Figure 7a shows that  ${}^t x_p$  and  ${}^t y_p$  coordinates correctly converge towards zero ( $\pm 10$  mm) and the shape of the  ${}^t y_p$  coordinate corresponds to the variation in position resulting of a trapezoidal velocity profile. The  ${}^t z_p$  coordinate value reaches 238 mm, higher than the reference target distance of 200 mm but within the margin of error for the vision-based estimation. This demonstrates the inaccuracy of the vision sensor to estimate this component that is hopefully not used in the control scheme thanks to the GPM that prioritizes the distance sensors. Figure 7b shows that the rotation of the platform along the  $y$  and  $z$  axes remains close to zero during the whole trajectory, and the only significant rotation is about  $x_t$ . Finally, the rotation components correctly converge towards zero, satisfying the orientation constraint given by the main task.

### C. Accuracy

The ground truth measurements obtained with the C-track are shown Fig. 8. They show that all the coordinates from the vision-based position estimation share the same evolution with their ground truth counterpart, with an offset. The offset is approximately 35 mm on the  $x$  axis, 50 mm for the  $y$  axis and 36 mm on the  $z$  axis.

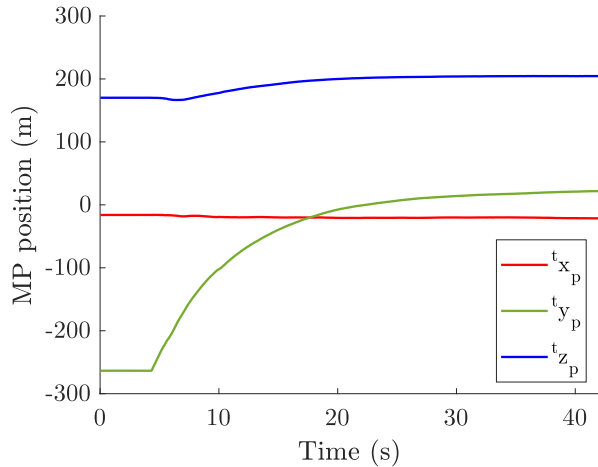


Fig. 8: Position of the MP in  $\mathcal{F}_t$ , ground truth obtained with the C-Track measurement system.

The vision-based position estimation error at time  $t$  is given by:

$$e_p(t) = \|\mathbf{x}_{p,v}(t) - \mathbf{x}_{p,c}(t)\|_2 \quad (25)$$

where  $\mathbf{x}_{p,v}$  is the position of the platform estimated by the system and  $\mathbf{x}_{p,c}$  is the position obtained from the C-track. The final vision-based position estimation error of the MP is  $e_p = 50$  mm. The distance error when the platform is parallel to the surface is given by:

$$e_{dist} = \bar{s}_i - z_{p,c} \quad (26)$$

The final distance error is  $e_{dist} = 3$  mm. This error could be reduced by a better calibration of the distance sensors. While the error due to the inaccuracy of the vision system

is significant, the proposed control scheme manages to keep an error on the main task more than 15 times lower than the position estimation error.

## IV. CONCLUSIONS

This paper proposed a control scheme to perform a profile following with a CDPR using a combination of boarded sensors and an external camera while implementing the Gradient Projection Method to deal with the measurement redundancy. The validity and the performance of this control scheme were assessed experimentally.

In spite of the poor precision of the external camera and thanks to the distance sensors, the proposed method is able to follow the surface with a positioning accuracy 15 times better than the precision provided by the external camera. The robot is thus able to follow a trajectory on an unknown profile with the vision system, while a purely distance-based approach would not be possible, but with an increase in precision compared to a pure visual control performed with the same camera. This validates the proposed approach, opening the possibility to implement this control strategy on a larger CDPR to really benefit from the advantage of the sensor cooperation.

However, several limitations exist and could be worth exploring for improvements. An assessment of the robustness of the system, as well as the maximum admissible curvature of the followed surface are still required. This curvature is limited by the size of the CDPR moving-platform and by potential interference between the ultrasound sensors. A possible solution to handle this problem would be to replace the latter by more costly, but more precise and more directive LASER rangefinders to reduce interference. These sensors could then be placed closer to each other and thus follow higher and/or more localized variations in curvature. The precision of the cameras does not hamper the surface following on simple profiles, but a better resolution is likely needed for any potential industrial applications, especially in a larger workspace. Another possibility would be to use a pan-tilt-zoom camera to follow the MP in the workspace.

## REFERENCES

- [1] D. Q. Nguyen, M. Gouttefarde, O. Company, and F. Pierrot, "On the simplifications of cable model in static analysis of large-dimension cable-driven parallel robots," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'13)*, 2013, pp. 928–934.
- [2] J.-P. Merlet, "The kinematics of cable-driven parallel robots with sagging cables: preliminary results," in *IEEE Int. Conf. on Robotics and Automation (ICRA'15)*, 2015, pp. 1593–1598.
- [3] A. Fortin-Côté, P. Cardou, and A. Campeau-Lecours, "Improving cable driven parallel robot accuracy through angular position sensors," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'16)*, 2016, pp. 4350–4355.
- [4] F. Xie, W. Shang, B. Zhang, S. Cong, and Z. Li, "High-precision trajectory tracking control of cable-driven parallel robots using robust synchronization," *IEEE Trans. on Industrial Informatics*, vol. 17, no. 4, pp. 2488–2499, 2020.
- [5] J.-P. Merlet and R. Tissot, "A panorama of methods for dealing with sagging cables in cable-driven parallel robots," in *Advances in Robots Kinematics*, 2022.
- [6] U. Mishra and S. Caro, "Forward Kinematics for Suspended Under-Actuated Cable-Driven Parallel Robots With Elastic Cables: A Neural Network Approach," *Journal of Mechanisms and Robotics*, vol. 14, no. 4, Aug. 2022.

- [7] M. Métilion, C. Charron, K. Subrin, and S. Caro, "Geometrical modelling of a cable-driven parallel robot winch," in *25<sup>e</sup> Congrès Français de Mécanique, Nantes, 2022*.
- [8] T. Dallej, M. Gouttefarde, N. Andreff, P.-E. Hervé, and P. Martinet, "Modeling and vision-based control of large-dimension cable-driven parallel robots using a multiple-camera setup," *Mechatronics*, vol. 61, pp. 20–36, 2019.
- [9] M. Zavatta, M. Chianura, A. Pott, and M. Carricato, "A vision-based referencing procedure for cable-driven parallel manipulators," *Journal of Mechanisms and Robotics*, vol. 12, no. 4, 2020.
- [10] Z. Zake, F. Chaumette, N. Pedemonte, and S. Caro, "Robust 2 1/2D Visual Servoing of a Cable-Driven Parallel Robot Thanks to Trajectory Tracking," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 660–667, Jan. 2020.
- [11] J. A. Gangloff and M. F. de Mathelin, "Visual servoing of a 6-dof manipulator for unknown 3-d profile following," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 4, pp. 511–520, 2002.
- [12] J. Pages, C. Collewet, F. Chaumette, and J. Salvi, "Optimizing plane-to-plane positioning tasks by image-based visual servoing and structured light," *IEEE Trans. on Robotics*, vol. 22, no. 5, pp. 1000–1010, 2006.
- [13] J. Thomas, F. Pasteau, and F. Chaumette, "Plane-to-plane positioning by proximity-based control," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'22), 2022*.
- [14] B. Siciliano, "Kinematic control of redundant robot manipulators: A tutorial," *Journal of Intelligent and Robotic Systems*, vol. 3, no. 3, pp. 201–212, 1990.
- [15] A. Liegeois *et al.*, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 7, no. 12, pp. 868–871, 1977.
- [16] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *The Int. Journal of Robotics Research*, vol. 6, no. 2, pp. 3–15, 1987.
- [17] C. Samson, B. Espiau, and M. Le Borgne, *Robot control: the task function approach*. Oxford University Press, Inc., 1991.
- [18] G. Flandin, F. Chaumette, and E. Marchand, "Eye-in-hand/eye-to-hand cooperation for visual servoing," in *IEEE Int. Conf. on Robotics and Automation (ICRA'00)*, vol. 3, 2000, pp. 2741–2746.
- [19] F. Chaumette, S. Hutchinson, and P. Corke, "Visual servoing," in *Springer Handbook of Robotics*. Springer, 2016, pp. 841–866.
- [20] Z. Zake, F. Chaumette, N. Pedemonte, and S. Caro, "Vision-based control and stability analysis of a cable-driven parallel robot," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1029–1036, 2019.
- [21] N. Mansard and F. Chaumette, "Task sequencing for high-level sensor-based control," *IEEE Trans. on Robotics*, vol. 23, no. 1, pp. 60–72, 2007.