# Robust 2 1/2D Visual Servoing of a Cable-Driven Parallel Robot Thanks to Trajectory Tracking

Zane Zake[1,2], *Student Member, IEEE*, François Chaumette[3], *Fellow, IEEE*, Nicolò Pedemonte[2], and Stéphane Caro[1,4], *Member, IEEE*

***Abstract*—Cable-Driven Parallel Robots (CDPRs) are a kind of parallel robots that have cables instead of rigid links. Implementing vision-based control on CDPRs leads to a good final accuracy despite modeling errors and other perturbations in the system. However, unlike final accuracy, the trajectory to the goal can be affected by the perturbations in the system. This paper proposes the use of trajectory tracking to improve the robustness of 2½D visual servoing control of CDPRs. Lyapunov stability analysis is performed and, as a result, a novel workspace, named control stability workspace, is defined. This workspace defines the set of moving-platform poses where the robot is able to execute its task while being stable. The improvement of robustness is clearly shown in experimental validation.**

***Index Terms*—Parallel Robots, Visual Servoing, Motion Control, Sensor-based Control, Tendon/Wire Mechanism**

## I. INTRODUCTION

A special kind of parallel robots named Cable-Driven Parallel Robots (CDPRs) has cables instead of rigid links. The main advantages of CDPRs are their large workspace, low mass in motion, high velocity and acceleration capacity, and reconfigurability [1]. The main drawback of CDPRs is their poor positioning accuracy. Multiple approaches to deal with this drawback can be found in the literature. The most common one is the improvement of the CDPR model. Since cables are not rigid bodies, creating a precise CDPR model is a tedious task, because it needs to include, for example, pulley kinematics, cable sag, elongation and creep [2] [3] [4]. Besides, cable-cable and cable-platform interferences can affect the accuracy of a CDPR. To avoid those interferences, studies have been done on the definition of CDPR workspace [1]. When modeling has been deemed unsuitable or insufficient, sensors have been used to gain knowledge about some of the system parameters. For example, angular sensors can be used to retrieve the cable angle [5]; cable tension sensors can be used to assess the current payload mass and the location of center of gravity [6]; color sensors can be used to detect regularly spaced color marks on cables to improve cable length measurement [7]. Of course, exteroceptive sensors can be used to measure the moving-platform (MP) pose accurately. To the best of our knowledge, few studies exist on the use of vision to control CDPRs and improve their accuracy. For instance, four cameras are used in [8] to precisely detect the MP pose of a large-scale CDPR. Furthermore, additional stereo-camera pairs were used to detect cable sagging at their exit points from the CDPR base structure. Similarly, [9] used a six infrared camera system to detect the MP pose of a CDPR used in a haptic application. A camera can also be mounted on the MP to see the object of interest. In this case, control is performed with respect to the object of interest. Thus the MP pose is not directly observed. Such a control algorithm for a three-DOF translational CDPR has been introduced in [10], and it has been extended to six-DOF CDPRs in [11], where the authors used a pose-based visual servoing (PBVS) control scheme. The robustness of this control scheme to perturbations and uncertainties in the robot model was analyzed. The stability analysis of this controller was extended in [12] to find the limits of perturbations that do not yield the system unstable. As a conclusion, as long as the perturbations are kept within these limits, they do not affect the MP accuracy at its final pose. However, even if perturbation levels are kept within the boundaries, they have an undesirable effect along the trajectory to the goal.

To further improve the robustness and the achievement of the expected trajectory, planning and tracking of a trajectory can be used. Trajectory planning and tracking take advantage of stability and robustness to large perturbations of classical visual servoing approaches in the vicinity of the goal [18]. Indeed, when the difference between current and desired visual features is small, the behavior of the system approaches the ideal one, no matter the perturbations. With the implementation of trajectory planning and tracking, the desired features are varying along the planned trajectory keeping the difference between current and desired visual features small at all times.

Under perfect conditions, the PBVS control used in [11] and [12] leads to a straight-line trajectory of the target center-point in the image, which means that the target is likely not to be lost during task execution. Unfortunately, even under perfect conditions the camera trajectory is not a straight line. To have a straight-line trajectory for both the target center-point in the image and the camera in the robot frame, a hybrid visual servoing control, named 2½D visual

Manuscript received: September, 10, 2019; Revised December, 5, 2019; Accepted January, 2, 2020.

This paper was recommended for publication by Editor Dezhen Song upon evaluation of the Associate Editor and Reviewers' comments.

This work is supported by IRT Jules Verne (French Institute in Research and Technology in Advanced Manufacturing Technologies for Composite, Metallic and Hybrid Structures) in the framework of the PERFORM project.

[1]Laboratoire des Sciences du Numérique de Nantes, UMR CNRS 6004, 1, rue de la Noë, 44321 Nantes, France, Zane.Zake@ls2n.fr

[2]IRT Jules Verne, Chemin du Chaffault, 44340, Bouguenais, France, nicolo.pedemonte@irt-jules-verne.fr

[3]Inria, Univ Rennes, CNRS, IRISA, Rennes, France, Francois.Chaumette@inria.fr

[4]Centre National de la Recherche Scientifique (CNRS), 1, rue de la Noë, 44321 Nantes, France, stephane.caro@ls2n.fr

Digital Object Identifier (DOI): see top of this page.

servoing (2½D VS) [13] [14], has been selected in this paper. It combines the use of 2D and 3D features in order to profit from the benefits of PBVS and Image-Based Visual Servoing (IBVS), while suffering the drawbacks of neither.

Accordingly, this paper deals with robust 2½D VS of a CDPR thanks to trajectory tracking. It allows us to ensure the predictability of the trajectory to the goal. Furthermore, it improves the overall robustness of the system. In addition, it was found in [12] that the stability of the system with given perturbations depends also on the MP pose in the base frame. As a consequence, a novel workspace, named Control Stability Workspace (CSW), is defined. This workspace gives the set of MP poses where the robot is able to execute its task, while being stable from a control viewpoint.

This paper is organized as follows. Section II presents the vision-based control strategy for a CDPR. Section III is dedicated to the addition of trajectory planning and tracking in the control strategy. Stability of both control types is analyzed in Section IV. Section V is dedicated to the definition of a novel workspace named Control Stability Workspace. Section VI describes the experimental results obtained on a small-scale CDPR. Finally, conclusions are drawn in Section VII.

## II. 2½D Visual Servoing of Cable-Driven Parallel Robots

### A. CDPR Kinematics

The schematic of a spatial CDPR is shown in Fig. 1. The camera is mounted on the MP, therefore the homogeneous transformation matrix ${}^{p}\mathbf{T}_{c}$ between the MP frame $\mathcal{F}_p$ and the camera frame $\mathcal{F}_c$ does not change with time. On the contrary, the homogeneous transformation matrices ${}^{b}\mathbf{T}_{p}$ between the base frame $\mathcal{F}_b$ and the MP frame $\mathcal{F}_p$, and ${}^{c}\mathbf{T}_{o}$ between the camera frame $\mathcal{F}_c$ and the object frame $\mathcal{F}_o$ change with time.

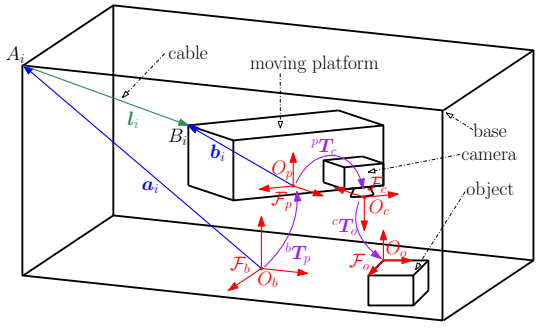


Fig. 1. Schematic of a spatial CDPR with eight cables, a camera mounted on its MP and an object in the workspace

The length $l_i$ of the $i$th cable is the 2-norm of the vector $\overrightarrow{A_iB_i}$ pointing from cable exit point $A_i$ to cable anchor point $B_i$, namely,

$$l_i = \left\| \overrightarrow{A_iB_i} \right\|_2 \tag{1}$$

with

$$l_i {}^{p}\mathbf{u}_i = {}^{p}\overrightarrow{A_iB_i} = {}^{p}\mathbf{b}_i - {}^{p}\mathbf{a}_i = {}^{p}\mathbf{b}_i - {}^{p}\mathbf{R}_b\, {}^{b}\mathbf{a}_i - {}^{p}\mathbf{t}_b \tag{2}$$

where ${}^{p}\mathbf{u}_i$ is the unit vector of ${}^{p}\overrightarrow{A_iB_i}$ that is expressed as:

$${}^{p}\mathbf{u}_i = \frac{{}^{p}\overrightarrow{A_iB_i}}{\left\| {}^{p}\overrightarrow{A_iB_i} \right\|_2} = \frac{{}^{p}\mathbf{b}_i - {}^{p}\mathbf{a}_i}{\left\| {}^{p}\overrightarrow{A_iB_i} \right\|_2} = \frac{{}^{p}\mathbf{b}_i - {}^{p}\mathbf{R}_b\, {}^{b}\mathbf{a}_i - {}^{p}\mathbf{t}_b}{\left\| {}^{p}\overrightarrow{A_iB_i} \right\|_2} \tag{3}$$

${}^{b}\mathbf{a}_i$ is the Cartesian coordinates vector of cable exit point $A_i$ expressed in $\mathcal{F}_b$; ${}^{p}\mathbf{b}_i$ is the Cartesian coordinates vector of cable anchor point $B_i$ expressed in $\mathcal{F}_p$; ${}^{p}\mathbf{R}_b$ and ${}^{p}\mathbf{t}_b$ are the rotation matrix and translation vector from $\mathcal{F}_p$ to $\mathcal{F}_b$.

The cable velocities $\dot{l}_i$ are obtained upon differentiation of Eq. (2) with respect to (w.r.t.) time:

$$\dot{\mathbf{l}} = \mathbf{A}\, {}^{p}\mathbf{v}_p \tag{4}$$

where ${}^{p}\mathbf{v}_p$ is the MP twist expressed in its own frame $\mathcal{F}_p$, $\dot{\mathbf{l}}$ is the cable velocity vector, and $\mathbf{A}$ is the Forward Jacobian matrix of the CDPR, defined as [15]:

$$\mathbf{A} = \begin{bmatrix} {}^{p}\mathbf{u}_1^T & ({}^{p}\mathbf{b}_1 \times {}^{p}\mathbf{u}_1)^T \\ \vdots & \vdots \\ {}^{p}\mathbf{u}_m^T & ({}^{p}\mathbf{b}_m \times {}^{p}\mathbf{u}_m)^T \end{bmatrix} \tag{5}$$

where $m = 8$ for a spatial CDPR with eight cables. Thus the Jacobian $\mathbf{A}$ is a $(8 \times 6)$–matrix.

### B. 2½D Visual Servoing

The control scheme considered in this paper is shown in Fig. 2. An image is retrieved from the camera and processed with a computer vision algorithm, from which the current feature vector is defined as $\mathbf{s} = [{}^{c^*}\mathbf{t}_c^T \;\; x_o \;\; y_o \;\; \theta u_z]^T$ [13] [14]. Here, ${}^{c^*}\mathbf{t}_c$ is the translation vector between the desired camera frame $\mathcal{F}_{c^*}$[1] and the current camera frame $\mathcal{F}_c$; $x_o$ and $y_o$ are the image coordinates of the object center $\mathbf{o}$; $\theta u_z$ is the third component of $\theta\mathbf{u}$ vector, where $\mathbf{u}$ is the axis and $\theta$ is the angle of the rotation matrix ${}^{c^*}\mathbf{R}_c$. An error vector $\mathbf{e}$ is defined by comparing $\mathbf{s}$ to $\mathbf{s}^*$, namely

$$\mathbf{e} = \mathbf{s} - \mathbf{s}^* \tag{6}$$

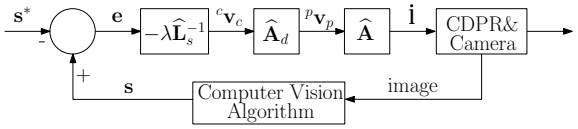


Fig. 2. Control scheme for visual servoing of a CDPR

As mentioned in the introduction, in perfect conditions, this choice of visual features leads to a straight-line trajectory of the camera (because ${}^{c^*}\mathbf{t}_c$ is part of $\mathbf{s}$), as well as a straight-line trajectory of object center-point $\mathbf{o}$ in the image (as $(x_o, y_o)$ is also part of $\mathbf{s}$). The translational degrees of freedom are used to realize the 3D straight line of the camera, while the rotational degrees of freedom are devoted to the realization of the 2D straight line of point $\mathbf{o}$.

To decrease the error $\mathbf{e}$, an exponential decoupled form is selected

$$\dot{\mathbf{e}} = -\lambda\mathbf{e} \tag{7}$$

with a positive adaptive gain $\lambda$, that is computed at each iteration, depending on the current value of $\|\mathbf{e}\|_2$ [10]. The derivative of the error $\dot{\mathbf{e}}$ can be written as a function of the Cartesian velocity of the camera ${}^{c}\mathbf{v}_c$, expressed in $\mathcal{F}_c$:

$$\dot{\mathbf{e}} = \mathbf{L}_{\mathbf{s}}\, {}^{c}\mathbf{v}_c \tag{8}$$

[1]In this paper, the superscript $*$ denotes the desired value, e.g. desired feature vector $\mathbf{s}^*$. Similarly, $c^*$ in ${}^{c^*}\mathbf{t}_c$ refers to desired camera frame $\mathcal{F}_{c^*}$

where $\mathbf{L_s}$ is the interaction matrix given by [13] [14] [16]:

$$\mathbf{L_s} = \begin{bmatrix} {}^{c^*}\mathbf{R}_c & \mathbf{0}_3 \\ \frac{1}{Z}\mathbf{L}_v & \mathbf{L}_{v\omega} \end{bmatrix} \tag{9}$$

with:

$$\mathbf{L}_v = \begin{bmatrix} -1 & 0 & x_o \\ 0 & -1 & y_o \\ 0 & 0 & 0 \end{bmatrix} \tag{10}$$

$$\mathbf{L}_{v\omega} = \begin{bmatrix} x_o y_o & -(1+x_o^2) & y_o \\ (1+y_o^2) & -x_o y_o & -x_o \\ l_1 & l_2 & l_3 \end{bmatrix} \tag{11}$$

$l_1, l_2, l_3$ being the components of the third row of matrix $\mathbf{L}_\omega$:

$$\mathbf{L}_\omega = \mathbf{I}_3 - \frac{\theta}{2}[\mathbf{u}]_\times + \Big(1 - \frac{\mathrm{sinc}(\theta)}{\mathrm{sinc}^2(\frac{\theta}{2})}\Big)[\mathbf{u}]_\times^2 \tag{12}$$

where $\mathrm{sinc}(\theta) = \sin(\theta)/\theta$

Finally, injecting (7) into (8) the instantaneous velocity of the camera in its own frame can be expressed as:

$${}^c\mathbf{v}_c = -\lambda\, \widehat{\mathbf{L}}_\mathbf{s}^{-1}\, \mathbf{e} \tag{13}$$

where $\widehat{\mathbf{L}}_\mathbf{s}^{-1}$ is the inverse of the estimation of the interaction matrix $\widehat{\mathbf{L}}_\mathbf{s}$. Note that the inverse is directly used, because $\mathbf{L_s}$ is a $(6 \times 6)$–matrix that is of full rank for 2½D VS [16].

### C. Kinematics and Vision

To control the CDPR by 2½D VS, it is necessary to combine the modeling shown in Sections II-A and II-B. It is done by expressing the MP twist ${}^p\mathbf{v}_p$ as a function of camera velocity ${}^c\mathbf{v}_c$:

$${}^p\mathbf{v}_p = \mathbf{A}_d\, {}^c\mathbf{v}_c \tag{14}$$

where $\mathbf{A}_d$ is the adjoint matrix that is expressed as [17]:

$$\mathbf{A}_d = \begin{bmatrix} {}^p\mathbf{R}_c & [{}^p\mathbf{t}_c]_\times\, {}^p\mathbf{R}_c \\ \mathbf{0}_3 & {}^p\mathbf{R}_c \end{bmatrix} \tag{15}$$

Finally, the model of the system shown in Fig. 2 is written from Eqs. (4), (8) and (14):

$$\dot{\mathbf{e}} = \mathbf{L_s}\, \mathbf{A}_d^{-1} \mathbf{A}^\dagger\, \dot{\mathbf{l}} \tag{16}$$

where $\mathbf{A}^\dagger$ is the Moore-Penrose pseudo-inverse of the Jacobian matrix $\mathbf{A}$.

Upon injecting (14) and (13) into (4), the output of the control scheme, i.e. the cable velocity vector $\dot{\mathbf{l}}$, takes the form:

$$\dot{\mathbf{l}} = -\lambda \widehat{\mathbf{A}} \widehat{\mathbf{A}}_d \widehat{\mathbf{L}}_\mathbf{s}^{-1} \mathbf{e} \tag{17}$$

where $\widehat{\mathbf{A}}$ and $\widehat{\mathbf{A}}_d$ are the estimations of $\mathbf{A}$ and $\mathbf{A}_d$, resp.

## III. Trajectory Planning and Tracking

It is well known that having perturbations in the system, which do not cause loss of stability, has an undesirable effect on the trajectory. This was shown in [11] [12] for PBVS and it is also true for the 2½D VS controller. Trajectory planning and following can be used to increase the robustness of the chosen control w.r.t. modeling errors [18] and to preserve the straight-line shape of the trajectory [19].

Indeed, the larger $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$, the bigger the effect of modeling errors on system behavior. When tracking a chosen trajectory, at each iteration $i$ the error becomes $\mathbf{e}(t) = \mathbf{s}(t) - \mathbf{s}^*(t)$. Consequently, when $t = 0\,\mathrm{s}$ we have $\mathbf{s}^*(0) = \mathbf{s}(0)$. Since $\mathbf{s}^*(t)$ is now time varying, the control scheme needs to be slightly changed. More precisely, instead of (8) we now have [16] [19]:

$$\dot{\mathbf{e}} = \dot{\mathbf{s}} - \dot{\mathbf{s}}^* = \mathbf{L_s}\, {}^c\mathbf{v}_c - \dot{\mathbf{s}}^* \tag{18}$$

Hence, the new control scheme is shown in Fig. 3.

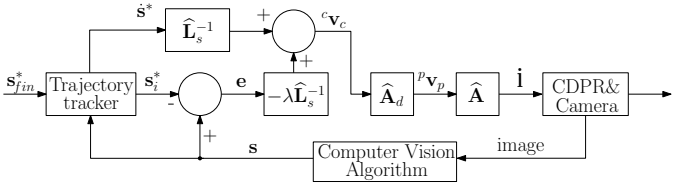


Fig. 3. Control scheme for VS with trajectory tracking of a CDPR

The new model of the system shown in Fig. 3 is written from Eqs. (4), (18) and (14):

$$\dot{\mathbf{e}} = \mathbf{L_s} \mathbf{A}_d^{-1} \mathbf{A}^\dagger\, \dot{\mathbf{l}} - \dot{\mathbf{s}}^* \tag{19}$$

Injecting (14) and (18) into (4) and expressing cable velocity vector $\dot{\mathbf{l}}$ leads to :

$$\dot{\mathbf{l}} = \widehat{\mathbf{A}} \widehat{\mathbf{A}}_d \widehat{\mathbf{L}}_\mathbf{s}^{-1} (-\lambda \mathbf{e} + \widehat{\dot{\mathbf{s}}^*}) \tag{20}$$

The success of any trajectory tracking is based on the time available to complete the task. The higher the trajectory time $t_{full}$, the more accurate the trajectory tracking. Indeed, the larger $t_{full}$, the lower the MP velocity, and the smaller the path step between two iterations. This leads to a smaller difference between $\mathbf{s}^*(t)$ and $\mathbf{s}^*(t+\Delta t)$, which in turn means a smaller difference between $\mathbf{s}(t)$ and $\mathbf{s}^*(t+\Delta t)$, thus a better path following.

### A. Implementation for 2½D VS

The implementation of the trajectory planning and tracking for 2½D VS is shown in Algorithm 1. There are three distinct phases, the first being the initialization, the second being the trajectory planning, and the third being the trajectory tracking. During the initialization phase, the final desired object pose ${}^{c^*}\mathbf{p}^*_{o_{fin}}$ and center-point $\mathbf{o}^*_{fin}$ are defined. They are used to compute the final feature vector $\mathbf{s}^*_{fin}$. Similarly, the initial feature vector $\mathbf{s}_{init}$ is defined based on the initial pose ${}^c\mathbf{p}_{o_{init}}$ and center-point $\mathbf{o}_{init}$ of the object of interest that are measured and recorded. This allows us to compute the full error:

$$\mathbf{e}_{full} = \mathbf{s}_{init} - \mathbf{s}^*_{fin} \tag{21}$$

and trajectory time:

$$t_{full} = \max\Big(\frac{\mathrm{e}_n}{\mathrm{v}_n}, n = 1 \text{ to } 6\Big) \tag{22}$$

where $\mathrm{e}_n$ stands for the $n$-th component of $\mathbf{e}_{full}$; $\mathrm{v}_n$ stands for the $n$-th component of the desired average velocity $\mathbf{v}$.

The current desired feature vector $\mathbf{s}^*(t)$ varies at a constant velocity $\mathbf{c}$ that is expressed as:

$$\mathbf{c} = \frac{\mathbf{e}_{full}}{t_{full}} \tag{23}$$

At the trajectory planning phase, we define $\mathbf{s}^*(t)$. At the beginning, when $t = 0\,\text{s}$, it is clear that $\mathbf{s}^*(0) = \mathbf{s}_{init}$. Then for $i = 1, \ldots, k$, where $k = t_{full}/\Delta t$ and $\Delta t$ is the time interval between two iterations, the trajectory planning is expressed as:

$$\mathbf{s}^*(i\Delta t) = \mathbf{s}_{init} + i\Delta t\,\mathbf{c} \tag{24}$$

As a consequence, we can set in (20):

$$\begin{cases} \dot{\mathbf{s}}^* = \widehat{\dot{\mathbf{s}}^*} = \mathbf{c} \text{ when } t < t_{full} \\ \dot{\mathbf{s}}^* = \widehat{\dot{\mathbf{s}}^*} = \mathbf{0} \text{ when } t \geq t_{full} \end{cases} \tag{25}$$

The third phase iterates until the difference $\left\|\mathbf{s}(t) - \mathbf{s}^*_{fin}\right\|_2$ reaches a defined threshold. At each iteration, the current feature vector $\mathbf{s}(t)$ is computed from the current object pose and the current object center-point coordinates. The current desired feature vector $\mathbf{s}^*(t)$ is retrieved from trajectory planning algorithm. This allows us to compute the current error $\mathbf{e}(t) = \mathbf{s}(t) - \mathbf{s}^*(t)$, which is then used as input of the control scheme.

**Algorithm 1:** Trajectory planning and tracking

**Initialization**
  Set the desired object pose ${}^{c^*}\mathbf{p}^*_{o_{fin}}$ and center-point coordinates $\mathbf{o}^*_{fin}$
  Define final feature vector $\mathbf{s}^*_{fin}$
  Read and record initial object pose ${}^{c}\mathbf{p}_{o_{init}}$ and center-point coordinates $\mathbf{o}_{init}$
  Define initial feature vector $\mathbf{s}_{init}$
  Compute trajectory time $t_{full}$ from (22)
  Compute the constant velocity $\mathbf{c}$ as in (23)
**End of Initialization**

**Trajectory Planning**
  $\mathbf{s}^*(0) = \mathbf{s}_{init}$
  $k = t_{full}/\Delta t$
  **for** $i = 1 : k$ **record**
    $\mathbf{s}^*(i\Delta t) = \mathbf{s}_{init} + i\Delta t\,\mathbf{c}$
  **end for**
**End of Trajectory Planning**

**Trajectory Tracking**
  **while** $\left\|\mathbf{s}(t) - \mathbf{s}^*_{fin}\right\|_2 >$ threshold **do**
    Retrieve current desired feature vector $\mathbf{s}^*(t)$
    Compute current feature vector $\mathbf{s}(t)$
    Compute current error $\mathbf{e}(t) = \mathbf{s}(t) - \mathbf{s}^*(t)$
    Compute current $\widehat{\mathbf{L}}_\mathbf{s}$, $\widehat{\mathbf{A}}$ and $\widehat{\mathbf{A}}_d$
    Compute $\dot{\mathbf{l}}$ using (20) and send to CDPR
  **end while**
**End of Trajectory Tracking**

## IV. Stability Analysis

The ability of a system to successfully complete its tasks can be characterized by its stability. By analyzing system stability, it is possible to find the limits of perturbation on different variables that the system is able to withstand, that is, to determine whether the system is able to converge accurately to its goal despite the perturbations [20].

In this paper, Lyapunov analysis is used to determine the stability of the closed-loop system.

### A. 2½D Visual Servoing

The following closed-loop equation is obtained from (16) and (17):

$$\dot{\mathbf{e}} = -\lambda \mathbf{L}_\mathbf{s} \mathbf{A}_d^{-1} \mathbf{A}^\dagger \widehat{\mathbf{A}} \widehat{\mathbf{A}}_d \widehat{\mathbf{L}}_\mathbf{s}^{-1} \mathbf{e} \tag{26}$$

From (26), a sufficient condition to ensure the system stability is [20]:

$$\mathbf{\Pi} = \mathbf{L}_\mathbf{s} \mathbf{A}_d^{-1} \mathbf{A}^\dagger \widehat{\mathbf{A}} \widehat{\mathbf{A}}_d \widehat{\mathbf{L}}_\mathbf{s}^{-1} > \mathbf{0}, \forall t \tag{27}$$

Indeed, if this condition is satisfied, the error $\mathbf{e}$ will always decrease to finally reach $\mathbf{0}$.

### B. Trajectory tracking with 2½D Visual Servoing

When trajectory tracking is involved, the closed-loop equation is written by injecting (20) into (19). Then, by using (25), we obtain:

$$\dot{\mathbf{e}} = -\lambda \mathbf{L}_\mathbf{s} \mathbf{A}_d^{-1} \mathbf{A}^\dagger \widehat{\mathbf{A}} \widehat{\mathbf{A}}_d \widehat{\mathbf{L}}_\mathbf{s}^{-1} \mathbf{e} + \mathbf{L}_\mathbf{s} \mathbf{A}_d^{-1} \mathbf{A}^\dagger \widehat{\mathbf{A}} \widehat{\mathbf{A}}_d \widehat{\mathbf{L}}_\mathbf{s}^{-1} \dot{\mathbf{s}}^* - \dot{\mathbf{s}}^* \tag{28}$$

The stability criterion $\mathbf{\Pi}$ keeps the form defined in (27). However, even if $\mathbf{\Pi}$ is positive definite, the error $\mathbf{e}$ will decrease iff the estimations are sufficiently accurate so that

$$\mathbf{L}_\mathbf{s} \mathbf{A}_d^{-1} \mathbf{A}^\dagger \widehat{\mathbf{A}} \widehat{\mathbf{A}}_d \widehat{\mathbf{L}}_\mathbf{s}^{-1} \dot{\mathbf{s}}^* \approx \dot{\mathbf{s}}^* \tag{29}$$

Otherwise tracking errors will be observed. This can be explained by a simple example from [16], where a scalar differential equation $\dot{e} = -\lambda e + b$, which is a simplification of (28), is analyzed. The solution is $e(t) = e(0)\exp(-\lambda t) + b/\lambda$, which converges towards $b/\lambda$. Increasing $\lambda$ reduces the tracking error. However, if it is too high, it can yield the system unstable. Therefore, it is necessary to keep $b$ as small as possible.

Most importantly, as the current desired feature vector $\mathbf{s}^*(t)$ approaches regularly the final desired feature vector $\mathbf{s}^*$, the desired feature vector velocity $\dot{\mathbf{s}}^*$ will become $\mathbf{0}$ as stated in (25), which makes the tracking errors vanish at the end.

## V. Control Stability Workspace

Before using a CDPR, one needs to know its workspace. Among the existing workspaces [21] [22], the static feasible workspace (SFW) is the simplest one and is formally expressed as [1]:

$$\mathcal{F} = \{\mathbf{p}_p \in SE(3) : \exists \boldsymbol{\tau} \in \mathcal{T},\ \mathbf{W}\boldsymbol{\tau} + \mathbf{w}_g = \mathbf{0}_6\} \tag{30}$$

Namely, the workspace $\mathcal{F}$ is the set of all MP poses $\mathbf{p}_p$ for which there exists a vector of cable tensions $\boldsymbol{\tau}$ within the cable tension space $\mathcal{T}$ such that the CDPR can balance the gravity wrench $\mathbf{w}_g$, and $\mathbf{W}\boldsymbol{\tau} + \mathbf{w}_g = \mathbf{0}_6$. Here, $\mathbf{W}$ is the wrench matrix and it is related to the robot Jacobian as $\mathbf{W} = -\mathbf{A}^T$.

This workspace is a kineto-static workspace that shows all the poses that the MP is physically able to attain. In addition, it is important to evaluate the CDPR ability to reach a pose from a control perspective.

In [12] it was concluded that the results of stability analysis were dependent on the size of the MP workspace. The smaller the desired workspace, the larger the tolerated perturbations within system stability. The MP pose and stability analysis are related to each other, because the MP pose shows up in the stability criterion $\mathbf{\Pi}$ through the Jacobian matrix $\mathbf{A}$ in the form of rotation matrix ${}^{b}\mathbf{R}_{p}$ and translation vector ${}^{b}\mathbf{t}_{p}$.

According to the stability analysis of 2½D VS control, presented in Section IV, the corresponding workspace, named Control Stability Workspace (CSW), is defined as follows:

$$\mathcal{C} = \{\mathbf{p}_p \in SE(3) : \forall \mathbf{d} \in \mathcal{D}, \mathbf{\Pi} > \mathbf{0}\} \tag{31}$$

The workspace $\mathcal{C}$ is the set of all MP poses $\mathbf{p}_p$, for which the stability criterion $\mathbf{\Pi}$ is positive definite for any vector of perturbations $\mathbf{d}$ that is within bounds $\mathcal{D}$. It means that for any MP pose within its CSW, the robot controller will be able to guide the MP to its goal.

It is of interest to create a compound workspace, that takes into account the controller and the kineto-static performance of the robot. Indeed, on the one hand, a MP pose can belong to $\mathcal{F}$ while being outside of $\mathcal{C}$, namely, it is in a static equilibrium, but it will fail to reach the goal. On the other hand, a MP pose can belong to $\mathcal{C}$ while being outside of $\mathcal{F}$, namely, the robot controller will make the MP reach the goal although the MP is not in a static equilibrium. Thus we define a compound workspace, named $\mathcal{FC}$, as the intersection of $\mathcal{F}$ and $\mathcal{C}$:

$$\begin{aligned}\mathcal{FC} = \{&\mathbf{p}_p \in SE(3) : \exists \boldsymbol{\tau} \in \mathcal{T},\ \forall \mathbf{d} \in \mathcal{D},\\ &\mathbf{W}\boldsymbol{\tau} + \mathbf{w}_g = \mathbf{0}_6,\ \mathbf{\Pi} > \mathbf{0}\}\end{aligned} \tag{32}$$

The compound workspace $\mathcal{FC}$ is the set of all MP poses $\mathbf{p}_p$ for which there exists a vector of cable tensions $\boldsymbol{\tau}$ within the cable tension space $\mathcal{T}$ such that the CDPR can balance the gravity wrench $\mathbf{w}_g$ leading to $\mathbf{W}\boldsymbol{\tau} + \mathbf{w}_g = \mathbf{0}_6$, and for which for any vector of perturbations $\mathbf{d}$ that is within bounds $\mathcal{D}$, the stability criterion $\mathbf{\Pi}$ is positive definite.

## VI. Experimental Setup and Validation

Stability criterion (27) is robot model dependent. Thus, ACROBOT, the CDPR prototype used for experimental validation, is presented in Section VI-A. Workspace $\mathcal{C}$ is computed in Section VI-C based on the numerical analysis of the stability criterion (27). Finally, experimental results are shown in Section VI-D.

### A. CDPR prototype ACROBOT

CDPR prototype ACROBOT is shown in Fig. 4. It is assembled in a suspended configuration, so that all the cable exit points are located at the four corners above the MP. Cables are 1.5 mm in diameter, assumed to be massless and nonelastic. The frame of the robot is a $1.2\,\text{m} \times 1.2\,\text{m} \times 1.2\,\text{m}$ cube. The MP size is $0.1\,\text{m} \times 0.1\,\text{m} \times 0.07\,\text{m}$ and its mass is 1.5 kg.

A camera is mounted on the MP facing the ground. As a simplification of the vision part, AprilTags [23] are used as objects and are put in various places on the ground. Their recognition and localization are done by algorithms available in the ViSP library [24]. The robot is controlled to arrive directly above a chosen AprilTag.

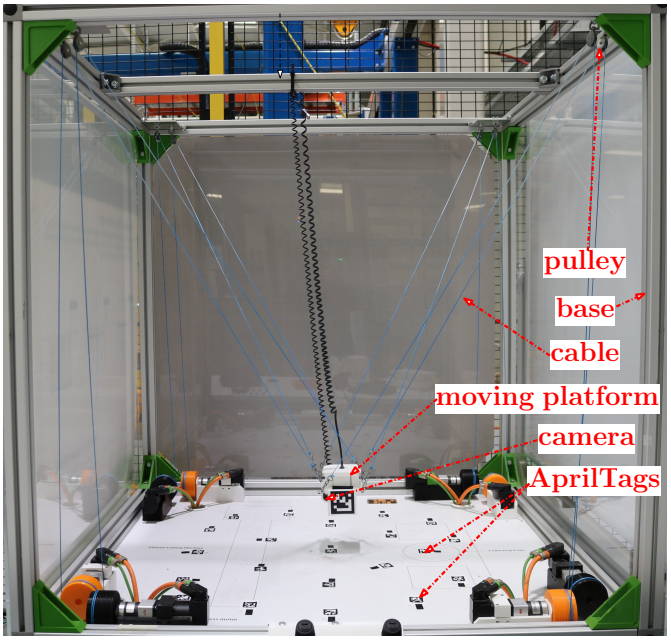


Fig. 4. ACROBOT: a CDPR prototype located at IRT Jules Verne, Nantes

### B. Constant and varying perturbations in the system

Two types of perturbations are considered depending on whether they change during task execution or not. Here is a list of perturbed parameters that do not change during the task execution:

- ${}^{p}\widehat{\mathbf{T}}_{c}$ - the pose of the camera in the MP frame $\mathcal{F}_p$ can be perturbed due to hand-eye calibration errors. It affects the adjoint matrix $\widehat{\mathbf{A}}_{d}$;
- ${}^{p}\hat{\mathbf{b}}_{i}$ - the Cartesian coordinates vector of cable anchor points expressed in $\mathcal{F}_p$ can be perturbed due to manufacturing errors. It affects the estimation of Jacobian matrix $\widehat{\mathbf{A}}$;
- ${}^{b}\hat{\mathbf{a}}_{i}$ - the Cartesian coordinates vector of cable exit points expressed in $\mathcal{F}_b$. Since pulleys are not modeled, there is a small difference between the modeled and the actual cable exit points. It affects the estimation of Jacobian matrix $\widehat{\mathbf{A}}$.

Here is a list of the perturbed parameters that vary during the task execution:

- $\mathbf{s}$ - the feature vector requires current AprilTag Cartesian pose in $\mathcal{F}_c$ and the image coordinates of its center-point $\mathbf{o}$. Those terms are computed from image features and are thus corrupted by noise. The smaller the AprilTag in the image, the larger the estimation error. It affects the interaction matrix $\widehat{\mathbf{L}}_{\mathbf{s}}$;
- ${}^{b}\widehat{\mathbf{T}}_{p}$ - the transformation matrix between frames $\mathcal{F}_b$ and $\mathcal{F}_p$ is estimated by exponential mapping:

$$({}^{b}\mathbf{T}_{p})_{i+1} = ({}^{b}\mathbf{T}_{p})_{i} \exp^{({}^{p}\mathbf{v}_{p}, \Delta t)} \tag{33}$$

Since ${}^{p}\mathbf{v}_{p}$ is computed from ${}^{c}\mathbf{v}_{c}$, which is perturbed by errors in ${}^{p}\widehat{\mathbf{T}}_{c}$, and since computed ${}^{c}\mathbf{v}_{c}$ does not correspond exactly to achieved ${}^{c}\mathbf{v}_{c}$ due to errors in $\widehat{\mathbf{A}}$ and due to the time-response of the low-level controller,

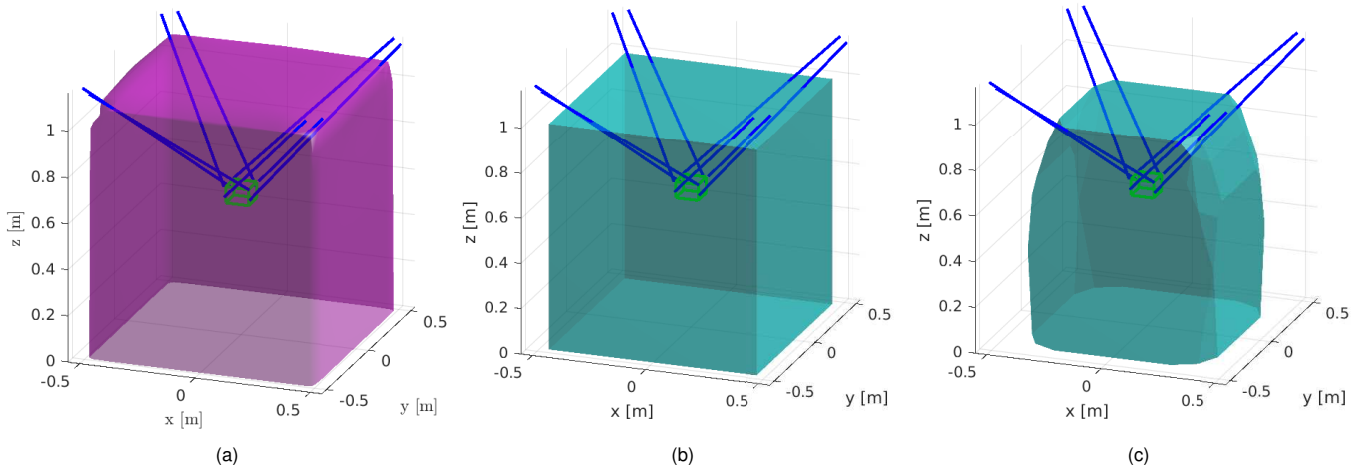


Fig. 5. Workspace visualizations for ACROBOT: a) SFW; b) CSW for 2½D VS with minimal perturbations in the system and constant MP orientation; c) CSW for 2½D VS with non-negligible perturbations in the system and MP rotation up to 30° about any arbitrary axis.

then ${}^b\widehat{\mathbf{T}}_p \neq {}^b\mathbf{T}_p$. Furthermore, the initial position is only coarsely known[2]. It affects the Jacobian matrix $\widehat{\mathbf{A}}$.

### C. Workspace of ACROBOT

The constant orientation static feasible workspace of ACROBOT was traced thanks to ARACHNIS software [25] and is shown in Fig. 5a.

CSW for ACROBOT is shown in Fig. 5b. Here, for the sake of comparison we also constrain the MP to the same constant orientation. Furthermore, we also take into account hand-eye calibration errors in camera pose in the MP frame $\mathcal{F}_p$, which are simulated as 0.01 m along and 3° about any arbitrary axis. Finally, the MP pose is assumed to be estimated coarsely, allowing for an error of 0.05 m in translation and 10° in rotation along and about any arbitrary axis.

Figure 5c shows a smaller CSW, where the system will remain stable with non-negligible perturbations. Namely, we add 0.19 m translational error and 8.5° rotational error along and about any arbitrary axis to the initial MP pose. Furthermore, we also simulate a bad hand-eye calibration by adding a 30° error to the camera pose in $\mathcal{F}_p$. Finally, since we are interested in changing the orientation of the MP, CSW shown in Fig. 5c allows for up to ±30° rotation of the MP about any arbitrary axis.

### D. Experimental Validation

An experimental setup was designed to validate the proposed approach. For 2½D VS we used an adaptive gain $\lambda$ [10]:

$$\lambda(x) = (\lambda_0 - \lambda_\infty)\mathrm{e}^{-(\dot{\lambda}_0/(\lambda_0-\lambda_\infty))x} + \lambda_\infty \qquad (34)$$

where:

- $x = ||\mathbf{e}||_2$ is the 2–norm of error $\mathbf{e}$ at the current iteration
- $\lambda_0 = \lambda(0)$ is the gain tuned for very small values of $x$
- $\lambda_\infty = \lambda(\infty)$ is the gain tuned for very high values of $x$
- $\dot{\lambda}_0$ is the slope of $\lambda$ at $x = 0$

These coefficients have been tuned at the following values: $\lambda_0 = 2.0$, $\lambda_\infty = 0.4$ and $\dot{\lambda} = 30$.

For the controller with trajectory tracker, $\lambda = \lambda_0 = 2.0$ has been set, since the error is always small. Additionally, for the planner $t_{full}$ is set to be equal to the execution time of the classic 2½D VS in order to ease comparability of the results. Finally, $\Delta t = 0.05$ s.

The initial values are the following:

$$\begin{cases} {}^b\mathbf{p}_p = \left[0.107\,m;\ -0.026\,m;\ 0.35\,m;\ +20°;\ -20°;\ 0°\right] \\ {}^c\mathbf{p}_o = \left[-0.022\,m; 0.136\,m; 0.449\,m; -157°; -18°; -176°\right] \\ \mathbf{o} = \left[-0.043\,m;\ 0.301\,m\right] \end{cases}$$

and final desired values are selected to be:

$$\begin{cases} {}^b\mathbf{p}_p^* = \left[0.30\,m;\ 0.25\,m;\ 0.12\,m;\ 0°;\ 0°;\ 0°\right] \\ {}^c\mathbf{p}_o^* = \left[0\,m;\ 0\,m;\ 0.09\,m;\ -180°;\ 0°;\ -180°\right] \\ \mathbf{o}^* = \left[0\,m;\ 0\,m\right] \end{cases}$$

where ${}^b\mathbf{p}_p$ denotes the MP pose in the base frame $\mathcal{F}_b$; ${}^c\mathbf{p}_o$ denotes the AprilTag pose in the camera frame $\mathcal{F}_c$; and $\mathbf{o}$ stands for the AprilTag center-point coordinates in the image. Note that ${}^c\mathbf{p}_o$ and $\mathbf{o}$ were measured, while ${}^b\mathbf{p}_p$ was estimated through the ${}^b\mathbf{T}_p$ exponential mapping explained in Section VI-B. Therefore, the ${}^b\mathbf{p}_p$ and ${}^b\mathbf{p}_p^*$ are shown as a reference to Fig. 5c, but are not used in the control.

Two perturbation sets are defined as $V1$ and $V2$. The former corresponds to the CSW shown in Fig. 5c and includes: $(i)$ a perturbation of initial MP pose of 0.19 m along axis $\mathbf{u} = \left[0.56; 0.64; 0.52\right]$ and 8.4° about axis $\mathbf{u} = \left[0.73; 0.67; -0.14\right]$; $(ii)$ and a perturbation on the camera orientation expressed in $\mathcal{F}_p$ of 18° about axis $\mathbf{u} = \left[0.61; -0.51; -0.61\right]$. The set $V2$ includes: $(i)$ a perturbation of initial MP pose of 0.13 m along axis $\mathbf{u} = \left[-0.57; -0.52; 0.63\right]$ and 9.5° about axis

[2]The knowledge of initial MP pose is usually difficult to acquire when working with CDPRs. The usual approach is to always finish a task at a known home pose. This can be impossible due to a failed experiment or an emergency stop. Furthermore, great care must be taken when measuring the home pose, which in case of ACROBOT was done by hand.

$\mathbf{u} = \big[-0.52; 0.85; -0.04\big]$; $(ii)$ a perturbation of camera pose in $\mathcal{F}_p$ of 0.05 m along y axis and 12.5° about axis $\mathbf{u} = \big[0.78; -0.51; -0.35\big]$; $(iii)$ and a perturbation of 0.005 m in a random direction for each cable exit point $A_i$ and anchor point $B_i$.

Figure 6 shows the experimental results (see also the accompanying video). Figure 6a shows the trajectories of the AprilTag center-point in the image, while Fig. 6c shows the 3D trajectories of the camera in the frame $\mathcal{F}_b$. Additionally, the deviation from the straight-line trajectory in the image and in $\mathcal{F}_b$ is shown in Figs. 6b and 6d, respectively. Each controller, the classic 2½D VS and the one with trajectory tracking (named "Traj. tracking" in Fig. 6) was tested without added perturbations and under the effect of each perturbation set $V1$ and $V2$. Each experiment was repeated 15 times and the results are combined in a bar graph shown in Fig. 7.

Under good conditions, the behavior is as expected, namely, we see straight-line trajectories both in 3D and in the image. When no perturbation is added, the behavior of 2½D VS controller with and without trajectory tracking is similar. For both controllers the deviation does not surpass 0.01 m and 10 pixels. The superiority of trajectory tracking can be clearly seen when the system is perturbed. Each of the perturbation sets forces the classic 2½D VS to produce deviations from the ideal trajectories. $V2$ leads to higher deviation on the 3D trajectory (orange line in Fig. 6d), while $V1$ has a more pronounced effect on the trajectory in the image (brown line in Fig. 6b). On the contrary, the perturbation sets have a minimal effect on the trajectories produced by the controller with trajectory tracker as depicted by the gray and cyan lines in Fig. 6 for $V1$ and $V2$, resp. Indeed, for the 3D trajectory three lines corresponding to the trajectory tracking controller remain very near. The behavior is slightly worse in the image, where perturbation set $V1$ leads to about 18 pixel error (gray line). However, it is three times smaller than the almost 55 pixel error (brown line) obtained with the classic 2½D VS under the same perturbations in Fig. 6b.

Figure 7 shows the max and mean deviation from the ideal 2D and 3D trajectories for both controllers subject to the three perturbation sets. When there is no perturbation, the behavior of the controller without and with trajectory tracker is similar (groups A and B). No matter the perturbation set, the errors are at least three times smaller when the trajectory tracker is used: groups C and D for $V1$; groups E and F for $V2$. Furthermore, the 3D trajectory deviation (and the deviation of the trajectory in image for $V2$) remains similar to the trajectory tracker without perturbation.

## VII. Conclusions

This paper dealt with the use of trajectory planning and tracking with 2½D Visual Servoing for the control of Cable-Driven Parallel Robots. First, the proposed controller aims to increase the robustness of the system with respect to perturbations and errors in the robot model. Furthermore, it ensures the straight-line motion of both the center-point of the AprilTag in the image and the camera in the base frame.

Furthermore, a Control Stability Workspace (CSW) was defined and computed for a CDPR prototype ACROBOT, based on the stability analysis of the full system under 2½D visual servoing control. The effect of perturbations on CSW size was highlighted.

The improvement of robustness due to the use of trajectory planning and tracking was clearly shown in experimental validation. While both systems, namely, without and with trajectory tracking, remain stable and achieve the set goal, the trajectory produced by the former is clearly affected by perturbations.

A further improvement would be developing a control law that allows us to detect and counteract the modeling errors, instead of increasing robustness to these errors.

## References


[1] L. Gagliardini, S. Caro, M. Gouttefarde, A. Girin, "Discrete Reconfiguration Planning for Cable-Driven Parallel Robots", in *Mechanism and Machine Theory*, vol. 100, pp. 313–337, 2016.

[2] V. L. Schmidt, "Modeling Techniques and Reliable Real-Time Implementation of Kinematics for Cable-Driven Parallel Robots using Polymer Fiber Cables", Ph.D. dissertation, Fraunhofer Verlag, Stuttgart, Germany, 2017.

[3] J. P. Merlet, "Singularity of Cable-Driven Parallel Robot With Sagging Cables: Preliminary Investigation", in *ICRA*, pp. 504–509, 2019.

[4] N. Riehl, M. Gouttefarde, S. Krut, C. Baradat, F. Pierrot. "Effects of non-negligible cable mass on the static behavior of large workspace cable-driven parallel mechanisms", in *ICRA*, pp. 2193–2198, 2009.

[5] A. Fortin-Côté, P. Cardou, A. Campeau-Lecours, "Improving Cable-Driven Parallel Robot Accuracy Through Angular Position Sensors", in *IEEE/RSJ Int. Conf on Intelligent Robots and Systems (IROS)*, pp. 4350–4355, 2016.

[6] E. Picard, S. Caro, F. Claveau, F. Plestan, "Pulleys and Force Sensors Influence on Payload Estimation of Cable-Driven Parallel Robots", in *Proceedings - IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Madrid, Spain, October, 1–5 2018.

[7] J. P. Merlet, "Improving cable length measurements for large CDPR using the Vernier principle", in *International Conference on Cable-Driven Parallel Robots*, pp. 47–58, Springer, Cham, 2019.

[8] T. Dallej, M. Gouttefarde, N. Andreff, P.-E. Hervé, P. Martinet, "Modeling and Vision-Based Control of Large-Dimension Cable-Driven Parallel Robots Using a Multiple-Camera Setup", in *Mechatronics*, vol. 61, pp. 20–36, 2019.

[9] R. Chellal, L. Cuvillon, E. Laroche, "A Kinematic Vision-Based Position Control of a 6-DoF Cable-Driven Parallel Robot", in *Cable-Driven Parallel Robots*, pp. 213–225, Springer, Cham, 2015.

[10] R. Ramadour, F. Chaumette, J.-P. Merlet, "Grasping Objects With a Cable-Driven Parallel Robot Designed for Transfer Operation by Visual Servoing", in *ICRA*, pp. 4463–4468, IEEE, 2014.

[11] Z. Zake, F. Chaumette, N. Pedemonte, S. Caro, "Vision-Based Control and Stability Analysis of a Cable-Driven Parallel Robot", in *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1029–1036, 2019.

[12] Z. Zake, S. Caro, A. Suarez Roos, F. Chaumette, N. Pedemonte, "Stability Analysis of Pose-Based Visual Servoing Control of Cable-Driven Parallel Robots" in *International Conference on Cable-Driven Parallel Robots*, pp. 73-84. Springer, Cham, 2019.

[13] F. Chaumette, E. Malis, "2 1/2 D visual servoing: a possible solution to improve image-based and position-based visual servoings", in *ICRA*, pp. 630-635, IEEE, 2000.

[14] V. Kyrki, D. Kragic, H. Christensen, "New shortest-path approaches to visual servoing", in *IEEE/RSJ Int. Conf on Intelligent Robots and Systems (IROS)*, pp. 349–354, 2004

[15] A. Pott, "Cable-Driven Parallel Robots: Theory and Application", vol. 120., Springer, Cham, 2018.

[16] F. Chaumette, S. Hutchinson, P. Corke, "Visual Servoing", in *Handbook of Robotics*, 2nd edition, O. Khatib B. Siciliano (ed.), pp. 841–866, Springer, 2016.

[17] W. Khalil, E. Dombre, "Modeling, Identification and Control of Robots", Butterworth-Heinemann, 2004, pp. 13–29.

[18] Y. Mezouar, F. Chaumette, "Path planning for robust image-based control", in *IEEE Trans. on Robotics and Automation*, vol. 18, no. 4, pp. 534–549, August 2002.

[19] F. Berry, P. Martinet, J. Gallice, "Trajectory generation by visual servoing", in *IROS*, Grenoble, France, September 1997.

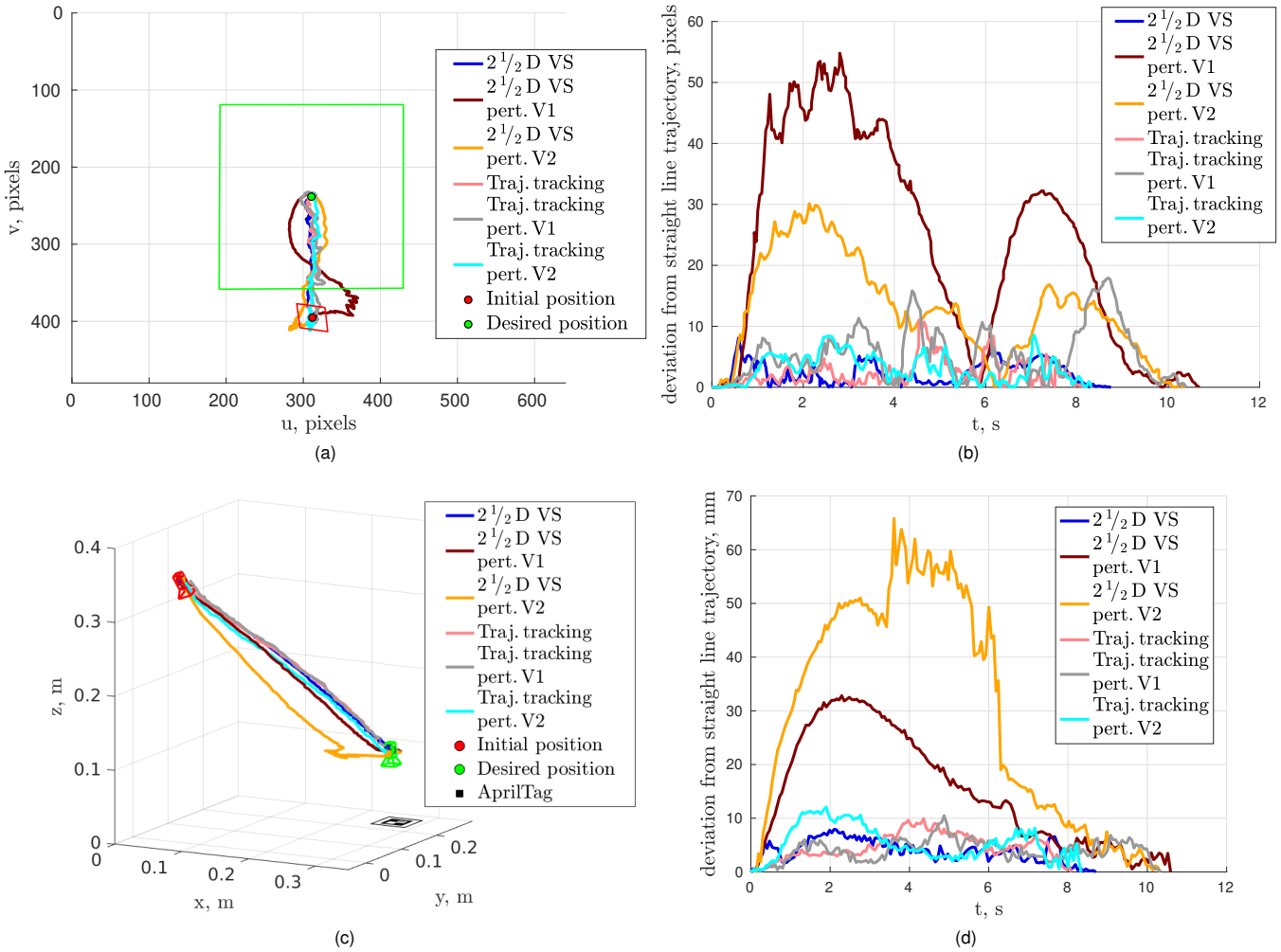


Fig. 6. 2½D VS experiments on ACROBOT: a) the trajectory of AprilTag center-point in the image; b) The pixel deviation from the ideal straight-line trajectory; c) the trajectory of the camera in the frame $\mathcal{F}_b$; d) the deviation from the ideal straight-line 3D trajectory.

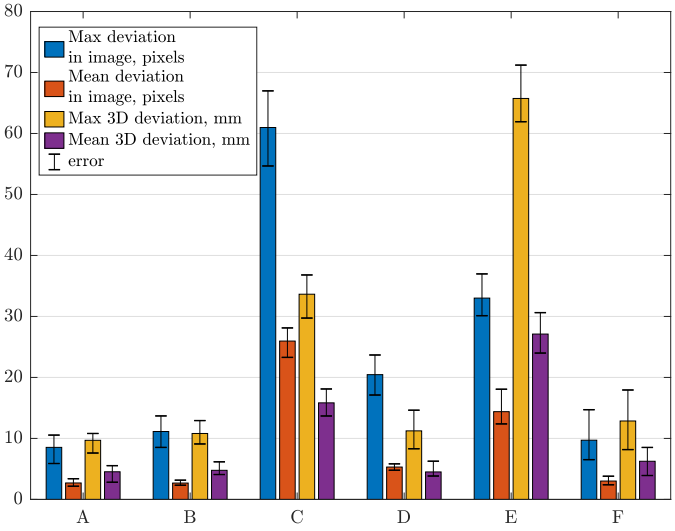


Fig. 7. Bar graph showing the max and mean deviation from the ideal object center-point $\mathbf{c}$ trajectory in image and the ideal camera trajectory in $\mathcal{F}_b$ with and without voluntarily added perturbations. Classical 2½D VS without added perturbation (A), under the effect of perturbation set $V1$ (C) and $V2$ (E);2½D VS with trajectory tracker without added perturbation (B), under the effect of perturbation set $V1$ (D) and $V2$ (F).


[20] H. K. Khalil, Nonlinear systems, Macmillan publishing Co., 2nd ed., New York 1996.

[21] E. Stump, V. Kumar, "Workspaces of Cable-Actuated Parallel Manipulators", in *Journal of Mechanical Design*, vol. 128, no. 1, pp. 159–167, 2006.

[22] R. Verhoeven, "Analysis of the workspace of tendon-based stewart platforms", Ph.D. dissertation, Univ. Duisburg-Essen, 2004.

[23] E. Olson, "AprilTag: A robust and flexible visual fiducial system", in *ICRA*, pp. 3400–3407, IEEE, 2011.

[24] É. Marchand, F. Spindler, F. Chaumette, "ViSP for visual servoing: a generic software platform with a wide class of robot control skills", in *IEEE Robotics & Automation Magazine*, vol. 12, no. 4, pp. 40–52, 2005.

[25] A. L. C. Ruiz, S. Caro, P. Cardou, F. Guay, "ARACHNIS: Analysis of Robots Actuated by Cables with Handy and Neat Interface Software", in *Cable-Driven Parallel Robots*, pp. 293–305, Springer, Cham, 2015.