# Sensor-based design of a Delta parallel robot

Minglei Zhu, Sébastien Briot, Abdelhamid Chriette

# Sensor-based design of a Delta parallel robot

Minglei Zhu[a,c], Sébastien Briot[b,c] and Abdelhamid Chriette[a,c]

[a]Ecole Centrale de Nantes France
[b]Centre National de la Recherche Scientifique (CNRS) France
[c]Laboratoire des Sciences du Numerique de Nantes (LS2N) UMR CNRS 6004 Nantes France

ARTICLE INFO

*Keywords*:
sensor-based design, parallel robots, Optimal design, Controller singularity, singularity

ABSTRACT

This paper proposes a sensor-based design methodology in order to design a Delta robot with guaranteed accuracy performance for a dedicated sensor-based controller. This sensor-based design methodology takes into account the accuracy performance of the controller in the design process in order to find optimal geometric parameters of the robot. Three types of controllers are envisaged to be applied to the Delta robot, leading to three different optimal designs: leg-direction-based visual servoing, line-based visual servoing and image moment visual servoing. Based on these three controllers, positioning error models taking into account the error of observation coming from the camera are developed, and the controller singularities are analyzed. Then, design optimization problems are formulated in order to find the optimal geometric parameters and relevant parameters of the camera for the Delta robot for each type of controller. Prototypes of Delta robots have been manufactured based on the obtained optimum design parameters in order to test the performance of the pair {robot-controller}.

## 1. Introduction

Compared with serial robots, parallel robots have several advantages: they can reach high speeds and accelerations, they have large payload capacities, as well as an interesting payload-to-stiffness ratio Merlet (2006b). However, their control may be troublesome due to their highly non-linear models.

A lot of research works focused on the control of parallel robots (see Merlet (2012) for a long list of references). The large majority of the proposed approaches are model-based controllers. This means that, in order to get high accuracy with a parallel robot, detailed prediction models (e.g., taking into account deformations or joint clearances) must be included in the controller. Nevertheless, these detailed models are complex and not computationally efficient. Moreover, they may still suffer from the problem of inaccuracy in reality because of robot assembly and manufacturing errors. Therefore, the use of sensor-based controllers is an efficient method to bypass the complex modeling approaches and to get good accuracy. Sensor-based controllers use external sensors to estimate the robot configuration, in particular the end-effector pose Chaumette, Hutchinson and Corke (2016); Chaumette and Hutchinson (2006). Visual servoing is a sensor-based closed-loop control that takes one or several cameras as sensors and closes the control loop by using the image features extracted from the camera signal. Several researches focused on using visual servoing for parallel robot control Dallej, Andreff, Mezouar and Martinet (2006); Andreff, Dallej and Martinet (2007); Traslosheros, Sebastian, Angel, Roberti and Carelli (2007); Garrido, Soria and Trujano (2008); Trujano, Garrido and Soria (2012). In past years, different image features have been proposed to have good performance in terms of accuracy for controlling parallel robots, for example, the observation of robot legs Dallej et al. (2006); Andreff et al. (2007) or the observation of features put on the robot platform Traslosheros et al. (2007); Qi and McInroy (2008); Garrido et al. (2008); Trujano et al. (2012).

For the case of a vision-based controller, positioning accuracy is one of the most important internal performances. The controller properties such as the number and types of cameras that are used, together with the kinds of features that are observed, have a great influence on the final observation error Kaci, Boudaud, Briot and Martinet (2018). Some other factors that affect the positioning accuracy are the camera position and also the robot geometric parameters because they are parameters of the interaction model Chaumette et al. (2016) linking the image feature to the estimated robot configuration. In particular, it is known that the interaction model is not free of singularities Michel and Rives (1993) that affect the robot accuracy performance Pascual Escudero, Nayak, Briot, Kermorgant, Martinet and Chaumette (2021). As a result, it is not wise to apply a given type of visual servoing technique on any robot and to expect that the controller will ensure the best accuracy performance for all possible robot configurations. On the contrary, the robot geometric parameters and the relevant parameters of the camera should be optimized in order to ensure the best accuracy performance for the pair {robot-controller}.

Robot optimal design allows finding optimal design parameters that minimize a given objective under constraints. The design parameters are related to the robot performance that will be put as the objectives and constraints of the optimization problem. They are usually of two kinds:

- Primary parameters Briot, Pashkevich and Chablat (2010a); Wu, Yu, Gao and Wang (2018) like the lengths of the links, or the constant offset angles between two linked bodies; these parameters are

minglei.zhu@uestc.edu.cn (M. Zhu); Sebastien.briot@ls2n.fr (S. Briot); Abdelhamid.chriette@ls2n.fr (A. Chriette)
ORCID(s):

usually related to the geometric, kinematic and kinetostatic robot performance (e.g. workspace size Bonev and Ryu (2001); Merlet (2006b); Wang, Dong, Ba, Mohammad, Axinte and Norton (2021), singularities Gosselin and Angeles (1990); Zlatanov, Fenton and Benhabib (1995), accuracy Merlet and Daney (2005); Merlet (2006a); WU, ZHANG, WANG and YU (2021), effort or velocity transmission indices Liu, Wang and Pritschow (2006); Brinker, Corves and Takeda (2019); Aginaga, Iriarte, Plaza and Mata (2018); Meng, Liu and Xie (2022); Qiang, Ehmann and Jian (2014); Wu, Wang, Zhang and Huang (2021)).

- Secondary parameters Briot, Pashkevich and Chablat (2010b) like the link-cross shapes, the material properties; these parameters are usually related to the dynamic and elastic robot performance (e.g. maximal input torques Briot and Khalil (2015); Wu, Wang, Wang and Li (2009); Wu, Ye, Yu and Huang (2022), cycle time Han, Xie, Liu, Meng and Zhang (2020), balancing Wu and Gosselin (2004); Baron, Philippides and Rojas (2021), deformation Pashkevich, Chablat and Wenger (2009), natural frequencies Cammarata, Condorelli and Sinatra (2013)).

The aforementioned performances come from robot physics, not from the robot controller. When visual servoing is applied, the controller singularity and controller positioning accuracy should be considered in the optimal design process, which has never been done before.

Therefore, in this work, we propose a "sensor-based design" approach that takes into account the controller performance (like accuracy and singularity) during the design process. We will apply this design procedure to the case study of the Delta robot Pierrot, Reynaud and Fournier (1990), which is one of the most popular parallel robots, and consider three types of vision-based controllers as dedicated controllers. The resulting designs will be compared altogether. It should be mentioned that, while being focused on visual servoing, The proposed optimal design methodology could be extended to any other sensor-based control approach Andersen (2015).

This paper is organized as follows. In Section 2, the sensor-based design methodology is introduced. Section 3 shows the robot architecture and design requirements. Some necessary recalls on visual servoing are discussed in Section 4. In particular, image features that are observed in the different aforementioned techniques are detailed. Moreover, it is shown that the controller accuracy performance depends on the observed features and camera placement but also on the robot geometric parameters. Section 5 explains how to find the robot configurations leading to controller singularities (and thus instability and positioning inaccuracy), and an error model relating the camera observation error to the robot positioning error is provided. Then, in Section 6, the optimal sensor-based design approach is formulated and solved. This approach takes into account the knowledge of the controller singular configuration and accuracy error model in order to find optimal parameters of the robot and the sensor ensuring controller accuracy performance. These results are validated through co-simulation in Section 7. Experimental validations are provided in Section 8. Finally, some conclusions are drawn in Section 9.

## 2. Sensor-based design methodology

The design process has been formalized by French French (1999). It traditionally starts with the phase called "preliminary design" or "conceptual design". In robot manipulator design, during this phase, some kinematic architectures potentially interesting for specified tasks are selected. Then, the robot geometry is optimized during the second phase called "advanced design" or "optimal design". This phase allows finding the robot's optimal design parameters that minimize a given objective under constraints. As recalled during the introduction, the design parameters are related to the robot's performance and they will be put as the objectives and constraints of the optimization problem. Traditionally, the aforementioned performances come from robot physics, not from robot controller. Once the optimal geometric parameters are found, the design embodiment phase starts: a CAD model is created, and then a prototype is manufactured.

In order to formulate the optimal design problem, let us denote as $\mathbf{p}$ the robot configuration, and by $\mathcal{W}$ all sets of configurations $\mathbf{p}$ for which some performances must be tested. Besides, for each configuration $\mathbf{p} \in \mathcal{W}$, let us define the functions $\varphi_\alpha(\mathbf{p}, \boldsymbol{\pi})$, $\alpha \in \{0, 1, \dots, n_\alpha\}$, that describe various *mechanical properties* of the manipulator (velocity, force transmission, accuracy, etc.) for any given set of the mechanical design parameters $\boldsymbol{\pi}$. Let us also assume that for each function $\varphi_\alpha$, it is possible to define physically consistent scalar measures $\sigma_\beta(\varphi_\alpha(\mathbf{p}, \boldsymbol{\pi}))$, $\beta \in \{0, 1, \dots, n_\beta\}$, that may be directly included in the design objectives or constraints. Some examples of such measures (e.g. transmission indices) are presented in the following sections.

Similarly, for the global evaluation of the manipulator, let us introduce global performance measures $\eta_\gamma(\boldsymbol{\pi})$, $\gamma \in \{0, 1, \dots, n_\gamma\}$, that also depend on the physical parameters of the links $\boldsymbol{\pi}$, but no more on the robot configuration $\mathbf{p}$. Among the numerous examples of global measures, we may cite the robot's total mass, footprint, workspace size, etc. Then, traditionally in optimal robot design, the design optimization problem can be stated as achieving the best value of the global performance indices:

$$\eta_\gamma(\boldsymbol{\pi}) \rightarrow \min_{\boldsymbol{\pi}}, \quad \forall \gamma \tag{1}$$

subject to the constraints:

$$\sigma_\beta(\varphi_\alpha(\mathbf{p}, \boldsymbol{\pi})) \in \mathcal{Q}_{\alpha\beta}, \quad \forall \alpha, \ \beta \text{ and } \forall \mathbf{p} \in \mathcal{W} \tag{2}$$

where $\mathcal{Q}_{\alpha\beta}$ represents the set of admissible values for the mechanical performance $\sigma_\beta(\varphi_\alpha(\mathbf{p}, \boldsymbol{\pi}))$.

When optimizing the robot from the point of view of its mechanical capacities only, all the behavior of the machine

due to the controller performance is neglected. This is problematic, as the controller impacts the global robot capabilities, in particular terms of positioning accuracy, tracking capacities, stiffness, stability, and so on. Therefore it makes sense to include in the robot design process some indices coming from the controller performances: Such a design methodology is called a "sensor-based design" approach.

In the sensor-based design approach, in order to take into account the impact of the sensor-based controller performances, it is necessary to modify the optimization problem (1)–(2) by introducing new constraints (and eventually new objectives) associated with properties no more related to physics but to the controller. Let us define as $\kappa_\epsilon(\mathbf{p}, \boldsymbol{\pi}, \boldsymbol{\zeta})$, $\epsilon \in \{0, 1, \dots, n_\epsilon\}$, various functions that describe *controller properties* (controller singularity, stability, convergence, etc.). These properties depend both on the robot mechanical parameters $\boldsymbol{\pi}$, its configuration $\mathbf{p}$, but also on some other parameters $\boldsymbol{\zeta}$ associated with the sensor (e.g. sensor intrinsic or extrinsic parameters, etc.). To these sensor-based controller properties $\kappa_\epsilon$, we assign some scalar measures $\sigma_\iota(\kappa_\epsilon(\mathbf{p}, \boldsymbol{\pi}, \boldsymbol{\zeta}))$, $\iota \in \{0, 1, \dots, n_\iota\}$, that may be also included in the design objectives or constraints. Some examples of such measures (repeatability, distance to controller singularity, etc.) are shown in what follows.

The sensor-based design optimization problem may be thus defined as (in case the controller performance are only added into the constraints):

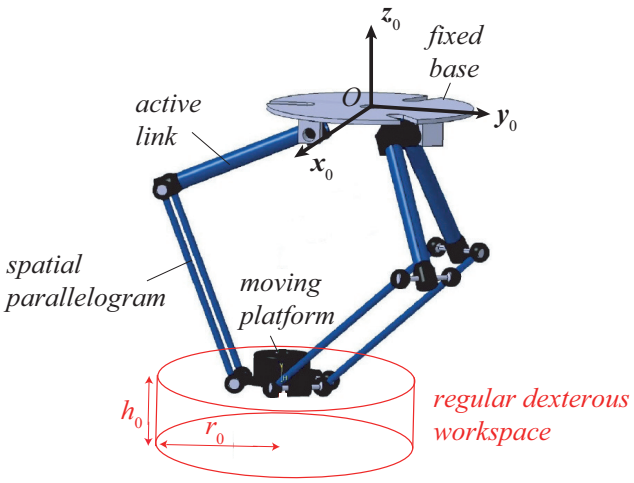
$$\eta_\gamma(\boldsymbol{\pi}\ \boldsymbol{\zeta}) \rightarrow \min_{\boldsymbol{\pi},\ \boldsymbol{\zeta}},\quad \forall\gamma \tag{3}$$

subject to the constraints:

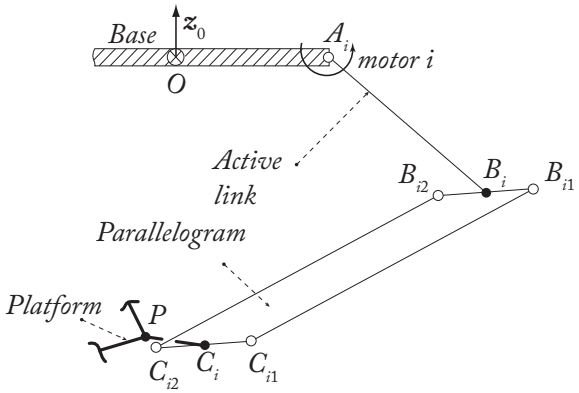
$$\begin{cases} \sigma_\beta(\varphi_\alpha(\mathbf{p}, \boldsymbol{\pi})) \in \mathcal{Q}_{\alpha\beta}, & \forall\alpha,\ \beta \quad \text{and} \quad \forall\mathbf{p} \in \mathcal{W} \\ \sigma_\iota(\kappa_\epsilon(\mathbf{p}, \boldsymbol{\pi}, \boldsymbol{\zeta})) \in \mathcal{Q}_{\epsilon\iota}, & \forall\epsilon,\ \iota \quad \text{and} \quad \forall\mathbf{p} \in \mathcal{W} \end{cases} \tag{4}$$

where $\mathcal{Q}_{\epsilon\iota}$ represents the set of admissible values for the controller performance $\sigma_\iota(\kappa_\epsilon(\mathbf{p}, \boldsymbol{\pi}, \boldsymbol{\zeta}))$. This problem formulation is general, and this sensor-based design methodology can be applied to any robots, as long as controller properties can be defined.

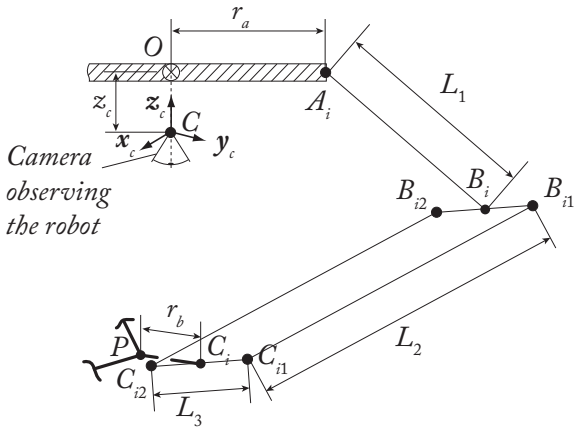
In the following Sections, we define controller performances related to visual servoing approaches, as well as more traditional mechanical performances, that will be used in order to optimize the design of a Delta robot. However, let us start by first defining the robot design specifications.




(a) CAD of the Delta robot with its regular dexterous workspace




(b) Schematics of the Delta leg architecture




(c) Delta geometric design parameters

**Figure 1:** Schematics of the Delta robot.

## 3. Robot Architecture and Specification

In this paper, we propose to optimize the geometry of the Delta robot Pierrot et al. (1990) controlled with vision-based controllers in order to obtain the best performance of the pair {robot-controller}. Let us first describe the Delta robot architecture.

The Delta robot is a parallel manipulator with three translational degrees of freedom (DOF). Its mobile platform can only translate along the three axes of the space with respect to the fixed base. The mechanism consists of a moving platform connected to a fixed base through three identical kinematic chains. Its architecture is illustrated in Fig. 1. Each chain contains an active link actuated by a revolute motor located at $A_i$ $(i = 1, 2, 3)$ in the base platform. The platform and the active links are connected through the use of spatial parallelograms $\boldsymbol{B}_{i1}\boldsymbol{B}_{i2}\boldsymbol{C}_{i2}\boldsymbol{C}_{i1}$ $(i = 1, 2, 3)$

**Table 1**
Requirements of the Delta robot

| | |
|---|---|
| Cylindrical RDW size | radius $r_0 \geqslant 200$ mm<br>height $h_0 \geqslant 100$ mm |
| Positioning repeatability in RDW | $\leqslant 0.5$ mm |
| No singularity in RDW | of the controller<br>of the robot |
| Constraints on geom. param. | will be provided<br>in Section 6.3 |

Laribi, Romdhane and Zeghloul (2007). In this work, we consider a symmetrical Delta robot architecture:

- triangles $A_1A_2A_3$ and $C_1C_2C_3$ are equilateral,
- the links $A_iB_i$ ($i = 1, 2, 3$) are moving in vertical planes containing $OA_i$,
- lengths for each link belonging to the different legs are the same: $\|\overrightarrow{A_iB_i}\| = L_1$, $\|\overrightarrow{B_iC_i}\| = L_2$, $\|\overrightarrow{B_{i1}B_{i2}}\| = \|\overrightarrow{C_{i1}C_{i2}}\| = L_3$ for $i = 1, 2, 3$.

The requirements that must be achieved by the Delta robot in this work are given in Tab. 1. They have been fixed after discussions with some of our industrial partners. The robot should be able to move in at least a cylindrical workspace of radius $r_0 \geq 200$ mm and height $h_0 \geq 100$ mm (see Fig. 1). In this cylinder, a set of performances should be guaranteed, that are detailed hereafter. Therefore this cylinder will be called the regular dexterous workspace (RDW) of the robot.

Throughout the RDW, some geometric and kinematic constraints must be satisfied:

- The RDW should be free of singularity (of the robot but also of the controller),
- The robot repeatability should be lower than 0.5 mm,
- Some distances are constrained in order to avoid collisions or to have unpractical designs: the distance $r_a$ between the origin of the base frame $O$ and the motor position $A_i$, the distance $r_b$ between the origin of the platform frame $P$ and the parallelogram extremity $C_i$, the different link lengths $L_1$, $L_2$ and $L_3$, the radius of the parallelogram link's $B_{ij}C_{ij}$ cross-section denoted as $R'$ and, finally, the camera frame location (Fig. 1(c)). These constraints will be further detailed in Section 6.3.

Additionally, for practical industrial reasons (gain of place), the robot must be as compact as possible.

In order to obtain the desired repeatability of 0.5 mm, visual servoing will be applied. We plan to use a single camera (1920×1200 pixels of resolution and a focal length of 10 mm) mounted onto the ground, which is already present in our lab. In order to get the best robot accuracy, the best idea is to directly observe with the camera some features present on the platform. However, there could be some application cases (milling operations, for instance) for which the end-effector would be difficult to observe. Alternative features to be observed are proposed in Andreff, Marchadier and Martinet (2005): the cylindrical legs of the robot's parallelograms, which are long and made of cylinders, are pretty easy to observe by a camera. Based on these considerations, three types of vision-based controllers are envisaged, two of which are dedicated to the visual servoing of parallel robots, the last one being more generic:

1. Leg-direction-based visual servoing Andreff et al. (2005),
2. Line-based visual servoing Vignolo, Briot, Philippe and Chen (2014),
3. Image-moment-based visual servoing of a feature mounted on the platform Chaumette (2004).

The first two controllers are using the observation of the robot legs, while the last one will be used to observe the platform directly. In what follows, we will find the optimal design parameters of the Delta robot for each type of controller. Then, the obtained results will be compared.

Before going further, we would like to make several comments. First, in these specifications, no dynamic criterion is provided. Indeed, visual servoing is not dedicated for high-speed motion purposes except for a few very specific scenarios Fusco, Kermorgant and Martinet (2020); Dahmouche, Andreff, Mezouar, Ait-Aider and Martinet (2012). Therefore, we focus only on robot geometry and kinematics performance. Studying dynamics with visual servoing is left as future work. Then, repeatability of 0.5 mm could also be obtained by using standard encoder-based controllers. However, the purpose of this paper is not to claim that using visual servoing is better than standard encoder-based control but to show that if someone wants to use visual servoing to control a robot (and, by extension, for any sensor-based controllers), it is necessary to optimize the pair {robot-controller} in order to be sure to obtain the desired accuracy.

Before presenting the optimization problem formulation, let us make some brief recalls on visual servoing.

## 4. Recalls on Visual Servoing

In this section, we provide some necessary recalls on visual servoing in general, and on the three considered approaches in particular Chaumette (2004); Andreff et al. (2005); Vignolo et al. (2014).

### 4.1. Basics of visual servoing

Visual servoing is a controller based on the use of the so-called interaction matrix $\mathbf{L}$ Chaumette and Hutchinson (2006) which relates the twist $\boldsymbol{\tau}_c$ between the camera and the scene, to the time derivative $\dot{\mathbf{s}}$ of the vector $\mathbf{s}$ of the visual primitives observed by the camera through the relationship:

$$\dot{\mathbf{s}} = \mathbf{L}(\mathbf{s}, \mathbf{x})\boldsymbol{\tau}_c \tag{5}$$

The components of the interaction matrix $\mathbf{L}$ are highly nonlinear and are function of both the visual features $\mathbf{s}$ and the robot end-effector configuration $\mathbf{x}$ which is also a function of $\mathbf{s}$, i.e. $\mathbf{x} = \mathbf{x}(\mathbf{s})$, $\mathbf{x}(\mathbf{s})$ being also a highly nonlinear function. Note that, in simulation, the value of $\mathbf{x}$ can be obtained directly by creating a virtual sensor measuring the simulated robot configuration. In the real world, the vector $\mathbf{x}$ must be rebuilt from the measurement $\mathbf{s}$. Several strategies for dealing with this problem are detailed in Chaumette et al. (2016); Chaumette and Hutchinson (2006).

Then, a classical controller based on the observation of the visual features $\mathbf{s}$ is based on the use of a proportional linearizing and decoupling control scheme of the form:

$$\boldsymbol{\tau}_c = -\lambda \mathbf{L}^{+}(\mathbf{s}, \mathbf{x})(\mathbf{s} - \mathbf{s}^{*}) \tag{6}$$

where $\lambda$ is a positive gain, $\boldsymbol{\tau}_c$ is used as a pseudo-control variable for the robot ($\boldsymbol{\tau}_c$ could be related to the motor velocities by $\boldsymbol{\tau}_c = \mathbf{J}\dot{\mathbf{q}}$, $\mathbf{J}$ being the Jacobian matrix of the robot and $\dot{\mathbf{q}}$ being the robot motor velocities) and $\mathbf{s}^{*}$ is the desired feature to be observed in order to reach the desired robot end-effector configuration $\mathbf{x}^{*}$.

From (5), we see that a simple visual servoing error model can emerge:

$$\Delta\mathbf{s} = \mathbf{L}(\mathbf{s}, \mathbf{x})\Delta\mathbf{x} \tag{7}$$

where $\Delta\mathbf{s}$ represents a small error in the observation of the features $\mathbf{s}$ and $\Delta\mathbf{x}$ an error in the position of the robot end-effector. As mentioned above, the matrix $\mathbf{L}$ is a matrix whose components are nonlinear functions depending of the variables $\mathbf{s}$ and $\mathbf{x}$. Therefore, the matrix $\mathbf{L}$ may meet some singularities near which, based on the equation (7), a small error of observation $\Delta\mathbf{s}$ may lead to a large error in robot positioning $\Delta\mathbf{x}$. Error models and singularities of the visual servoings Chaumette (2004); Andreff et al. (2005); Vignolo et al. (2014) will be further detailed in Section 5. Now, let us provide some recalls about the features observed in the three different types of controllers Chaumette (2004); Andreff et al. (2005); Vignolo et al. (2014).

## 4.2. Chosen visual features

In this section, we recall what are the features observed in the three different types of controllers Chaumette (2004); Andreff et al. (2005); Vignolo et al. (2014) in order to rebuild the robot end-effector configuration. Interaction matrices are not necessarily detailed. The interested reader would find all necessary information for their computation in Chaumette (2004); Andreff et al. (2005); Vignolo et al. (2014); Chaumette et al. (2016); Chaumette and Hutchinson (2006).

### *4.2.1. Leg-direction-based and line-based visual servoing*

These two approaches are pretty similar due to the fact that, in both of them, the robot legs are observed by the camera. For the Delta robot, the camera observes the cylindrical links $B_{ij}C_{ij}$ of the spatial parallelograms (Fig. 2). In the image plane, the contour of these cylinders of cross-section radius $R'$ are mapped into the lines $\ell_{ij}^{(k)}$ ($k = 1, 2$) defined as the intersections of the image plane and the plane $\mathcal{S}_{ij}^{(k)}$ with normal ${}^{c}\mathbf{n}_{ij}^{(k)}$ lying on the camera frame origin $C$ and the observed cylinder (see Fig. 3). Note that the superscript ${}^{c}$ in ${}^{c}\mathbf{n}_{ij}^{(k)}$ means that the vector is given in the camera frame. The plane $\mathcal{S}_{ij}^{(k)}$ is the so-called "line interpretation plane" Andreff, Espiau and Horaud (2002). For each cylinder, the normal vectors ${}^{c}\mathbf{n}_{ij}^{(k)}$ are information that can be provided by the camera, as shown in Andreff et al. (2007). Thus, if $n$ cylinders are observed, the vector $\mathbf{s}$ of the observed features is given by: $\mathbf{s} = [{}^{c}\mathbf{n}_{11}^{(1)T}\ {}^{c}\mathbf{n}_{11}^{(2)T}\ \dots\ {}^{c}\mathbf{n}_{n2}^{(1)T}\ {}^{c}\mathbf{n}_{n2}^{(2)T}]^{T}$.

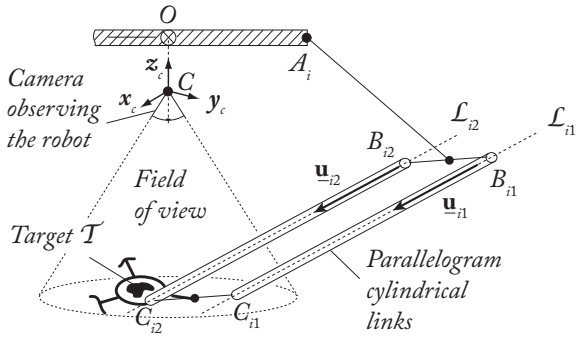



**Figure 2:** A camera observing either the robot parallelogram links or its platform

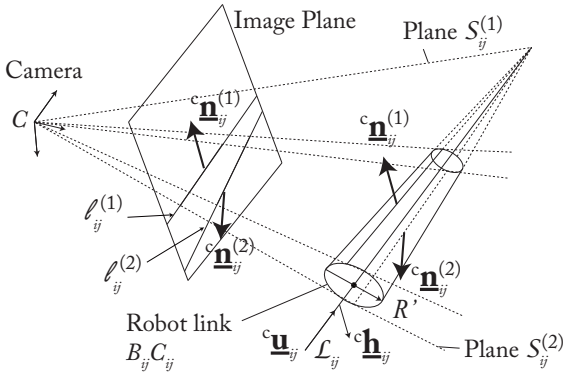



**Figure 3:** Projection of a cylinder in the image plane

This being said, the two approaches Andreff et al. (2005); Vignolo et al. (2014) differ by the following characteristics:

- For line-based visual servoing Vignolo et al. (2014), the vectors ${}^{c}\mathbf{n}_{ij}^{(k)}$ are used in order to rebuild the Plücker coordinates $\mathcal{L}_{ij} = ({}^{c}\underline{\mathbf{u}}_{ij}, {}^{c}\mathbf{h}_{ij})$ of the axis of the cylinder for the link $B_{ij}C_{ij}$ (Fig. 3), where ${}^{c}\underline{\mathbf{u}}_{ij}$ is the unit vector of its direction, and ${}^{c}\mathbf{h}_{ij}$ the moment of ${}^{c}\underline{\mathbf{u}}_{ij}$ with respect to the camera frame origin $C$. Once all Plücker coordinates $\mathcal{L}_{ij}$ are computed, then the function $\mathcal{Q}_{\ell}$ relating the coordinates $\mathcal{L} = (\mathcal{L}_{11}, \dots, \mathcal{L}_{n2})$ and the robot end-effector pose $\mathbf{x}$ can be established Vignolo et al. (2014). This function is different for each robot. It can be shown that, for the

Delta robot, a minimum of two lines coordinates $\mathcal{L}_{ij}$ must be known (belonging to two different parallelograms).

- For leg-direction-based visual servoing Andreff et al. (2005), the vectors ${}^c\mathbf{n}_{ij}^{(k)}$ are used in order to rebuild ${}^c\underline{\mathbf{u}}_{ij}$ only, i.e. the direction of the cylinder axis (Fig. 3). Then the function $\mathcal{Q}_u$ relating the coordinates $\mathbf{u} = (\mathbf{u}_{11}, \dots, \mathbf{u}_{n2})$ and the robot end-effector pose $\mathbf{x}$ can be established Andreff et al. (2007, 2005); Rosenzveig, Briot and Martinet (2013). This function is also different for each robot. It has been shown in Rosenzveig et al. (2013); Briot, Rosenzveig, Martinet, Özgür and Bouton (2016) that, for the Delta robot, a minimum of two independent line directions $\mathbf{u}_{ij}$ must be known.

For both approaches, in the models for estimating the robot platform configuration based on the image features $\mathbf{s} = [{}^c\mathbf{n}_{11}^{(1)T}\ {}^c\mathbf{n}_{12}^{(2)T} \dots {}^c\mathbf{n}_{n2}^{(1)T}\ {}^c\mathbf{n}_{n2}^{(2)T}]^T$, the radii $R'$ of the observed cylinders is involved, as well as the robot geometric parameters (see Andreff et al. (2007, 2005) for instance), and obviously, the camera pose. The value of these parameters will be optimized later during the design optimization process.

#### *4.2.2. Image moment visual servoing*

This approach is different from the previous ones. The idea is to control the robot thanks to the observation of a target $\mathcal{T}$ mounted on the end-effector (Fig. 2). Based on this observation, we extract the projected moments of the target in the image plane Chaumette (2004).

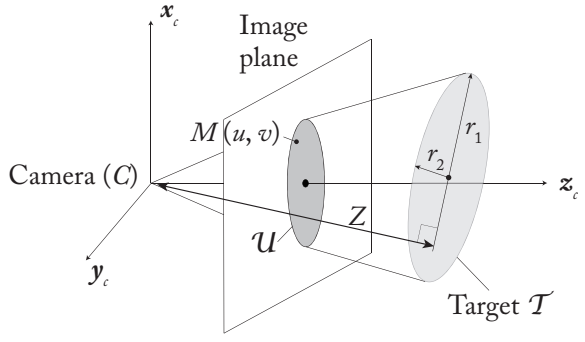



**Figure 4:** Projection of an ellipse in the image plane

Let us then consider a target $\mathcal{T}$ whose contours are closed (i.e. $\mathcal{T}$ is a continuous dense object). We denote as $\mathcal{U}$ the projection of $\mathcal{T}$ in the image plane (Fig. 4). It is then possible to compute the image moments of $\mathcal{U}$: the moment $m_{wt}$ of order $w + t$ is defined by Chaumette (2004)

$$m_{wt} = \iint_{\mathcal{U}} u^w v^t \, dx dy \tag{8}$$

where $u$ and $v$ are the coordinates in the camera frame of any point $M$ belonging to the surface $\mathcal{U}$. The interaction matrix associated with any moment is provided in Chaumette (2004). For a general object moving along six DOF, a set of six independent moments is necessary in order to control it. Since the Delta robot studied in this paper has only three translational DOF, only three independent moments are necessary. Here we select the following classical moments: the coordinates $x_g$, $y_g$ of the center of gravity of $\mathcal{U}$ ($x_g(= m_{10}/m_{00})$, $y_g(= m_{01}/m_{00})$), and the area $A$ of the surface $\mathcal{U}$ ($A = m_{00}$). In this case, and if $\mathcal{T}$ is a planar object, the interaction matrix for these moments has a simple expression which is Chaumette (2004):

$$\mathbf{L} = \begin{bmatrix} -C & 0 & C\,x_g \\ 0 & -C & C\,y_g \\ 0 & 0 & 2\,A\,C \end{bmatrix} \tag{9}$$

where $C = \frac{1}{Z}$, $Z$ is the depth between the camera and the object (Fig. 4).

We decided that the target $\mathcal{T}$ mounted on the platform of the Delta robot will be an ellipse of radii $r_1$ and $r_2$ (Fig. 4). Indeed, as shown in Tahri and Chaumette (2005), observing an ellipse is enough in order to control the position of an object in space. Therefore, we chose this target for the control of our translational robot. In the model for estimating the robot platform configuration based on the image features $\mathbf{s} = [x_g\ y_g\ A]^T$, the radii $r_1$ and $r_2$ are involved, as well as the camera pose. The value of these parameters will be optimized later during the design optimization process.

Note that, even if there is no explicit appearance of the robot geometric parameters in the interaction model of this controller, they still impact its performance: the robot geometric parameters define the location of the robot workspace. If the workspace is far from the camera location, the accuracy performance will be worse than if the workspace is closer to the camera location. Therefore, it is also necessary to optimize the robot's geometric parameters in order to optimize the overall robot accuracy.

Next Section deals with the computation of some performance indices of the vision-based controller.

## 5. Controller Performance

In this section, we define two types of vision-based controller performance:

- the presence (or even proximity) of controller singularities,
- the controller positioning accuracy.

Let us start with the computation of the controller singular configurations.

### 5.1. Controller singularities

As defined in Hutchinson, Hager and Corke (1996), the visual servoing controller singularity appears when the interaction matrix $\mathbf{L}$ is rank deficient. In this section, we study the conditions of rank deficiency of the interaction matrices associated with the controllers defined in Section 4.

#### 5.1.1. Leg-based visual servoing singularities

The singularities of the leg-based visual servoings Andreff et al. (2005); Vignolo et al. (2014) are different depending on the type of controller:

- *Leg-direction-based visual servoing:* The singularities of this controller applied to Delta robot have been studied in Rosenzveig et al. (2013) and Briot et al. (2016). It has been shown that a singularity will appear when all planes $\mathcal{V}_i$ ($i = 1, 2, 3$) containing points $A_i$ and $B_i$ and the axis of the active revolute joint located at point $A_i$ (Fig. 1) are parallel.

- *Line-based visual servoing:* As known from Merlet (2006b); Pascual Escudero et al. (2021), singularities of kinematic models are also singularities of the pose estimation models. When observing two to six lines $\mathcal{L}_{ij}$ passing through the links $B_{ij}C_{ij}$ ($i = 1, 2, 3$, $j = 1, 2$, Fig. 1) of the Delta robot in order to rebuild its end-effector pose, the robot platform orientation being always constant, the pose estimation model is equivalent to finding the common intersection point of the lines $\mathcal{L}'_{ij}$ obtained from lines $\mathcal{L}_{ij}$ by a translation of vectors $\overrightarrow{C_{ij}P}$. The only condition of the degeneracy of this pose estimation model is when the observed lines $\mathcal{L}'_{ij}$ are all parallel to each other. In other words, this means that the controller singularity of line-based visual servoing for the Delta robot will appear when all its links $B_{ij}C_{ij}$ belonging to the spatial parallelograms will be parallel.

#### 5.1.2. Image moment visual servoing singularities

From (9), it is easy to see that the interaction matrix of the selected image moments can be singular if and only if $C = 0$ or $A = 0$. $C$ being equal to $1/Z$ ($Z$ being the depth between the camera and the object), the condition $C = 0$ means that the object is at infinity. $A$ being the projected area in the image of the observed feature, $A = 0$ also means that the object is at infinity or that the plane containing the target $\mathcal{T}$ is lying on the camera center (thus meaning that the feature $\mathcal{T}$ cannot be observed anymore). Thus, these singularity cases will never happen in the image moment visual servoing of the Delta robot.

#### 5.1.3. Proximity to controller singularities

Even if the robot configurations leading to the controller singularities are known from the previous subsections, it is worthy of defining a criterion of "proximity" to controller singularities. Several indices could be adapted from the abundant literature in robot singularity analysis (see for instance, Merlet (2006a)). However, singularities of the controller being before all singularities impacting the numerical stability of the control law (6), we decided to use as a measure of the proximity to singularity the inverse conditioning of the interaction matrix, which is directly related to the numerical stability of the computed control input as shown in Pascual Escudero et al. (2021). More advanced criteria could obviously be used Merlet (2006a); Brinker et al. (2019).

### 5.2. Positioning error model

In order to characterize the controller repeatability, we propose to use the model (7), as it was done in Briot, Caro and Germain (2017). This model may be simplistic because it takes only as potential sources of errors the motor positioning errors. However, it has two main advantages:

- It is computationally efficient, which is extremely important in an optimal design algorithm,

- Even if it cannot handle all sources of errors in a real experimental benchmark, the difference between the model prediction and the real accuracy stays small as it will be shown later.

However, before this, we need to define what should be taken as observed feature errors $\Delta\mathbf{s}$ for each of the controllers.

#### 5.2.1. Observation errors in the leg-based visual servoing

When observing the robot links in the leg-based visual servoing approaches, the link edges are projected into the image plane into lines $\ell_{ij}^{(k)}$ (Fig. 3), which are then pixellated (Fig. 5). We consider here that the error of estimation of the lines $\ell_{ij}^{(k)}$ is due to a random shift of $\pm 1$ px in the pixels corresponding to the intersections of the $\ell_{ij}^{(k)}$ with the image plane boundary (points $P_{ijk}^{(1)}$ and $P_{ijk}^{(2)}$ in Fig. 5).

Let us denote as $\mathbf{p}_{ijk}^{(1)}$, $\mathbf{p}_{ijk}^{(2)}$ the coordinates of the intersections points $P_{ijk}^{(1)}$ and $P_{ijk}^{(2)}$. Without providing the mathematical derivations, we can find a mapping relating the time derivatives of the vectors ${}^c\underline{\mathbf{n}}_{ij}^{(k)}$ (characterizing the line $\ell_{ij}$, as explained in Section 4.2.1) to the time derivative of $\mathbf{p}_{ijk}^{(1)}$, $\mathbf{p}_{ijk}^{(2)}$ which can be written into the form:

$$ {}^c\dot{\underline{\mathbf{n}}}_{ij}^{(k)} = \mathbf{J}_{nijk} \begin{bmatrix} \dot{\mathbf{p}}_{ijk}^{(1)} \\ \dot{\mathbf{p}}_{ijk}^{(2)} \end{bmatrix} \tag{10} $$

or also, for the error model,

$$ \Delta {}^c\underline{\mathbf{n}}_{ij}^{(k)} = \mathbf{J}_{nijk} \begin{bmatrix} \Delta\mathbf{p}_{ijk}^{(1)} \\ \Delta\mathbf{p}_{ijk}^{(2)} \end{bmatrix} \tag{11} $$

where $\Delta {}^c\underline{\mathbf{n}}_{ik}^{(k)}$, $\Delta\mathbf{p}_{ijk}^{(1)}$ and $\Delta\mathbf{p}_{ijk}^{(2)}$ stand for small variations of the vectors ${}^c\underline{\mathbf{n}}_{ik}^{(k)}$, $\mathbf{p}_{ijk}^{(1)}$ and $\mathbf{p}_{ijk}^{(2)}$ respectively. Finally, recalling that for the controllers analyzed in the present subsection, $\Delta\mathbf{s} = [\Delta {}^c\mathbf{n}_{11}^{(1)T} \ \Delta {}^c\mathbf{n}_{11}^{(2)T} \ \dots \ \Delta {}^c\mathbf{n}_{n2}^{(1)T} \ \Delta {}^c\mathbf{n}_{n2}^{(2)T}]^T$,

$$ \Delta\mathbf{s} = \mathbf{J}_n \Delta\mathbf{p} \tag{12} $$

where $\Delta\mathbf{p}$ contains all errors $\Delta\mathbf{p}_{ijk}^{(1)}$ and $\Delta\mathbf{p}_{ijk}^{(2)}$.

We consider here that for the points in the image plane, the resolution of the camera is taken as $\pm 1$ pixel, which

is a typical noise for cameras. Thus every component of vector $\Delta\mathbf{p}_{ijk}^{(1)}$ and $\Delta\mathbf{p}_{ijk}^{(2)}$ can take the values $+1$ or $-1$. This being defined, the expression (11) can be introduced in the equation (7) in order to define the observation error model for the leg-based controllers, which could be written under the generic form:

$$\Delta\mathbf{x} = \mathbf{L}_P \Delta\mathbf{p} \tag{13}$$

where the matrix $\mathbf{L}_P$ is equal to $\mathbf{L}_P = \mathbf{L}^+\mathbf{J}_n$. Recall that the interaction matrix $\mathbf{L}$ is not the same for the leg-direction-based controller as for the line-based controller, leading to different error models for those two approaches.

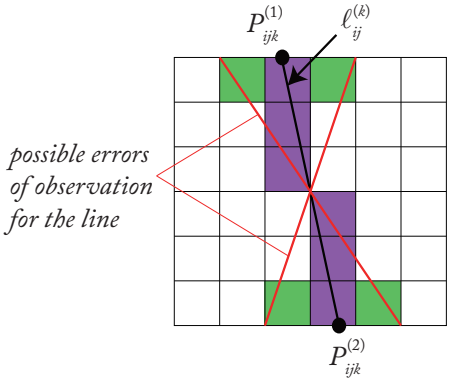



**Figure 5:** Error for the observation of a line

#### 5.2.2. *Observation errors in the image moment visual servoing*

For this controller, the observation error is more tricky to define. Based on our experience with this type of controllers, we defined the following observation errors. We estimate that there is an error of estimation $\Delta x_g = \pm 1$ px, $\Delta y_g = \pm 1$ px for the location of the surface $\mathcal{U}$ center of gravity. Moreover, we consider that it may be up to 1 pixel of error on the estimation of the main radii $d_1$ and $d_2$ of the ellipse $\mathcal{U}$, leading to the fact that the error $\Delta A$ is given by $\Delta A = \pi[(d_1 \pm 1)(d_2 \pm 1) - d_1 d_2]$ px$^2$. Thus, the observation error $\Delta\mathbf{s}$ in the model (7) is provided by $\Delta\mathbf{s} = [\delta x_g, \delta y_g, \delta A]^T$.

It should be mentioned that we fixed the error for the location of the center of gravity of the surface $\mathcal{U}$ at $\pm 1$ px based on the experimental results shown in the paper Dahmouche, Ait-Aider, Andreff and Mezouar (2008). In this work, the authors observed the location of centers of masses of several ellipsoids, and they experimentally estimated the error of observation at 0.54 pixels with a standard deviation of 0.62 pixels. This means that more than 67 % of the observation errors were below 1.16 pixels. Therefore we decided to use this value of $\pm 1$ pixel for the estimated error of observation, which seemed to us realistic in our context.

In the next Section, we show the formulation of the optimal design problem.

## 6. Optimal Design Procedure

The design procedure developed to obtain the optimal parameters of the Delta robot is described thereafter.

### 6.1. Design variables

**Geometric parameters of the robot:** As shown in Section 3, the geometry of the Delta robot is parameterized with the following variables: $L_1, L_2, L_3, r_a, r_b$ as depicted in Fig. 1 ($L_1 = \|\overrightarrow{A_iB_i}\|$, $L_2 = \|\overrightarrow{B_iC_i}\|$, $L_3 = \|\overrightarrow{B_{i1}B_{i2}}\| = \|\overrightarrow{C_{i1}C_{i2}}\|$, $r_a = \|\overrightarrow{OA_i}\|$, $r_b = \|\overrightarrow{PC_i}\|$). All these parameters have an effect either on the size of the robot workspace and its physics performance as well as on the controller performance.

Additionally, as mentioned in Section 4.2.1, the controller interaction models are dependent on another robot parameter: For the leg-based controllers, the interaction model is dependent of the cross-section radius $R'$ of the links $B_iC_i$. As a result, for these controllers, this parameter should be optimized. For the image-based controller, whose interaction model is not affected by this parameter, its value will be fixed at: $R' = 0.01$ m.

**Relevant parameters of the camera:** The visual servoing controller interaction model is dependent on the relative configuration between the camera and the object to be observed. The camera configuration is normally parameterized by six independent parameters. However, the robot architecture is symmetrical (see Section 3). In order to have a homogeneous observation of the workspace by the camera (and thus of the robot legs or end-effector):

- We fix the camera frame orientation to be parallel to the robot fixed frame,
- The camera origin is imposed to stay on a vertical line passing through $O$.

As a result, the camera position is parameterized by a single parameter $z_c$, which is the vertical location of the camera origin in the robot fixed frame (Fig. 1(c)).

Additionally, as mentioned in Section 4.2.2, the interaction model for the image moments is dependent on some other parameters: the radii $r_1$ and $r_2$ of the observed ellipse. These two parameters must be optimized when dealing with this type of controller.

**Design variables:** Based on the following explanations, we define two different set of design variables (grouped in a vector $\mathbf{y}$), depending of the types of controllers:

- For the leg-based controllers, $\mathbf{y} = [r_a, r_b, L_1, L_2, L_3, z_c, R']^T$
- For the moment-based controllers, $\mathbf{y} = [r_a, r_b, L_1, L_2, L_3, z_c, r_1, r_2]^T$.

### 6.2. Objective function

As mentioned in Section 3, the robot should be as compact as possible. We consider in this work that an index

of compacity for the studied robot could be the following volume:

$$V = S \cdot L_3 \tag{14}$$

where $S$ is the surface of the polygon $OA_iB_iC_iB$ when:

- $\overrightarrow{OP}$ is aligned along the vertical of the base frame,
- $\overrightarrow{A_iB_i}$ and $\overrightarrow{B_iC_i}$ are perpendicular.

Then:

$$S = \frac{1}{2}\left((r_a + r_b)\sqrt{L_1^2 + L_2^2 - (r_a - r_b)^2} + L_1 L_2\right) \tag{15}$$

### 6.3. Constraints

The constraints provided in Section 3 are recalled here for reasons of readability. Throughout the RDW, some geometric and kinematic constraints must be satisfied:

- The RDW should be free of singularity (of the robot but also of the controller); Singularities of the controllers are detailed in Section 5.1 while the singularities of the Delta robot are analyzed in Di Gregorio (2004). As explained in Section 5.1.3, for characterizing the proximity to singularity for the controller, we used the inverse condition number of the interaction matrix $\mathbf{L}$, denoted as $\kappa^{-1}(\mathbf{L})$. In the RDW, we want to have

  $$\kappa^{-1}(\mathbf{L}) > 10^{-3} \tag{16}$$

  For the robot "mechanics" singularity, we used the transmission index $\gamma$ defined inLiu, Li and Zhou (2015). In the RDW, we want to have

  $$\gamma > 10^{-3} \tag{17}$$

- The robot repeatability should be lower than 0.5 mm. For the robot being controlled by vision, the positioning error comes from the camera observation errors defined in Section 5.2. The models (7) and (13) being linear in terms of the observation error, the maximal positioning error $\delta_{\max} = \max \|\delta\mathbf{x}\|$ of the robot will be found at one of the corners of the hyper-polyhedron defining the observation errors Briot et al. (2010a). The repeatability constraint can be formulated as:

  $$\delta_{\max} \leq 0.5 \text{ mm} \tag{18}$$

- Some distances or angles are constrained in order to avoid collisions or to have unpractical designs. The numerical values of these constraints are provided here:

  $$\begin{aligned}
  &0 \leqslant r_b \leqslant 0.1 \text{ m},\\
  &r_b \leqslant r_a \leqslant 0.4 \text{ m},\\
  &0.4 \text{ m} \leqslant L_1 \leqslant 0.8 \text{ m},\\
  &0.2 \text{ m} \leqslant L_2 \leqslant 0.6 \text{ m},\\
  &0.08 \text{ m} \leqslant L_3 \leqslant 0.16 \text{ m},\\
  &25 \text{ deg} \leqslant \angle B_iB_{i1}C_{i1} \leqslant 155 \text{ deg},\ i = 1,2,3\\
  &0 \text{ m} \leqslant R' \leqslant 0.04 \text{ m},\\
  &0 \text{ m} \leqslant r_2 \leqslant r_1 \leqslant 0.04 \text{ m},\\
  &0 \text{ m} \leqslant z_c \leqslant 0.15 \text{ m}
  \end{aligned} \tag{19}$$

The base radius $r_a$ should be bigger than the platform radius $r_b$ in order to get a platform design compatible with the need of traditional pick-and-place operations, especially for packaging purposes. The base radius upper value, as well as the limits for the lengths and radii of the different bars, are constrained by the capacities of our machine tools. The extreme values for the angle $\angle B_iB_{i1}C_{i1}$ are set in order to avoid the degeneracy of the parallelograms, which would lead to a poor effort transmission and potential damage of the spherical joints. Finally, the constraints on the ellipsoid target $\mathcal{T}$ are set so that it can be put on the platform without colliding with the spherical joints.

The aforementioned RDW in which all constraints (16) to (19) must be satisfied should be a cylinder of radius $r_0 \geq 200$ mm and height $h_0 \geq 100$ mm (see Fig. 1). The algorithm given in Germain, Caro, Briot and Wenger (2013) is adapted in order to find the cylindrical Largest Regular Dexterous Workspace (LRDW) among the RDW of the manipulator for a given decision variable vector $\mathbf{y}$.

Let us denote as $r_{LRDW}$ and $h_{LRDW}$ the radius and height of the cylindrical LRDW. Because in the LRDW, all constraints (16) to (19) are obligatory true, then only two constraints can be used to replace all the other ones. They are defined by:

$$r_0 \leq r_{LRDW} \tag{20}$$

$$h_0 \leq h_{LRDW} \tag{21}$$

### 6.4. Problem formulation and optimization results

Based on the previous description of the objective and constraint functions, the optimal design problem can be formulated as:

$$\begin{aligned}
\min_{\mathbf{y}} \quad & V\\
\text{subject to} \quad & r_{LRDW} \geqslant 200 \text{ mm}\\
& h_{LRDW} \geqslant 100 \text{ mm}
\end{aligned} \tag{22}$$

We would like to note that, in the leg-based visual servoings, we consider that all six links $B_{ij}C_{ij}$ ($i = 1,2,3$, $j = 1,2$) are observed. The previous optimization algorithm is then applied for the design of the Delta robot for each of the three previously defined controllers. In order to achieve the global minima, these optimization problems have been solved by means of the 'active-set' algorithm implemented in the MATLAB *fmincon* function. Besides, a *multistart* algorithm, combined with random initial points initialized by a Genetic Algorithm was also used to increase the chances of reaching the global minima. The efficiency of this method

**Table 2**
Design parameters and value of the objective function as a function of the chosen controller

| | leg-direction-based visual servoing | line-based visual servoing | image moment visual servoing |
|---|---|---|---|
| $r_a$[m] | 0.2213 | 0.1741 | 0.1292 |
| $r_b$[m] | 0.0631 | 0.0530 | 0.0513 |
| $L_1$[m] | 0.6680 | 0.5614 | 0.5844 |
| $L_2$[m] | 0.4668 | 0.3060 | 0.3742 |
| $L_3$[m] | 0.1066 | 0.0856 | 0.0852 |
| $z_c$[m] | 0.0500 | 0.0500 | 0.1289 |
| $R'$[m] | 0.020 | 0.015 | N/A |
| $r_1$[m] | N/A | N/A | 0.0254 |
| $r_2$[m] | N/A | N/A | 0.0254 |
| $V$[m$^3$] | 0.02873 | 0.01345 | 0.01461 |

has been shown in Germain et al. (2013); Zhu, Chriette and Briot (2021). The optimal design results are given in Tab. 2.

It can be observed that, in terms of the objective function, the Delta robots designed for the line-based visual servoing and image moment visual servoing are very close to each other. The footprint of the Delta robot designed for leg-direction-based visual servoing is more than twice bigger as the other ones, meaning that this design is less attractive for the industrial purpose of compactness. For the other two robots, it is difficult to conclude anything more at this step of the analysis. In order to have a better insight into their intrinsic performances, comparing their footprint is not enough, and other performances must be checked, such as their experimental positioning accuracy capacities. This analysis will be presented in the remaining of the paper.

In the following section, co-simulations with ADAMS and Simulink will be performed in order to cross-validate the accuracy performance of the robots designed with the parameters provided in Tab. 2.

## 7. Results Cross-validations through Simulations

### 7.1. Simulation method

In order to make a first validation of the optimization results, co-simulations are performed in a connected ADAMS-Simulink environment (Fig. 6). Three Delta robot models are created in the software ADAMS by using the results obtained from the optimal design process (one model per controller).

From each model, we extract some information (block "Data acquisition") from the ADAMS simulator:

- For the leg-based controllers, we extract the coordinates of the points $C_i$ and $B_i$ (Fig. 1),
- For image moment visual servoing, we get the coordinates of the centroid point of the observed ellipse and the two points at the extremity of its minimal and maximal radii.

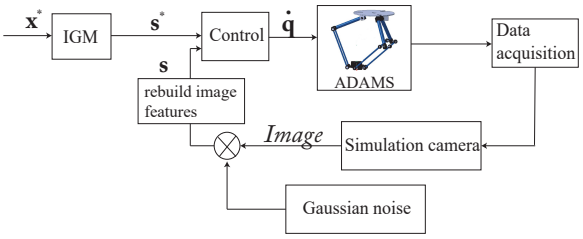



**Figure 6:** Co-simulation control scheme of Delta robot

**Table 3**
Test points coordinates parameterized with respect to the center of the LRDW.

| Point | Coordinate [m] | Point | Coordinate [m] | Point | Coordinate [m] |
|---|---|---|---|---|---|
| $P_1$ | (0.2,0,-0.05) | $P_9$ | (0.2,0,0) | $P_{17}$ | (0.2,0,0.05) |
| $P_2$ | (0.14,0.14,-0.05) | $P_{10}$ | (0.14,0.14,0) | $P_{18}$ | (0.14,0.14,0.05) |
| $P_3$ | (0,0.2,-0.05) | $P_{11}$ | (0,0.2,0) | $P_{19}$ | (00,0.2,0.05) |
| $P_4$ | (-0.14,0.14,-0.05) | $P_{12}$ | (-0.14,0.14,0) | $P_{20}$ | (-0.14,0.14,0.05) |
| $P_5$ | (-0.2,0,-0.05) | $P_{13}$ | (-0.2,0,0) | $P_{21}$ | (-0.2,0,0.05) |
| $P_6$ | (-0.14,-0.14,-0.05) | $P_{14}$ | -0.14,-0.14.2,0) | $P_{22}$ | (-0.14,-0.14,0.05) |
| $P_7$ | (0,-0.2,-0.05) | $P_{15}$ | (0,-0.2,0) | $P_{23}$ | (0,-0.2,0.05) |
| $P_8$ | (0.14,-0.14,-0.05) | $P_{16}$ | (0.14,-0.14,0) | $P_{24}$ | (0.14,-0.14,0.05) |

Based on these extracted data, we rebuild the image seen by the camera (block "Simulation camera") with an additional random noise related to the observation errors detailed in Section 5.2. Then, we extract the observed image features **s** depending on the controller type and use them in order to control the robot based on the controller defined in (6).

In the first set of simulations, we started to simulate each robot with its dedicated controller. In all these simulations, a home position $P_0$ and twenty-four desired positions within the LRDW are defined: all positions are defined in Tab. 3 with respect to the center of the LRDW, which is also the home position $P_0$.

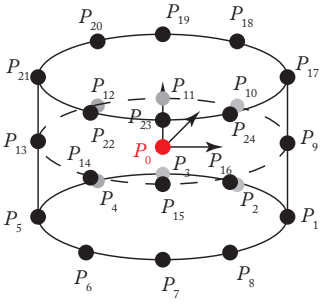



**Figure 7:** Test points in the LRDW

Then, each robot performs motions based on the controller (6) from the home position to the desired positions defined in Tab. 3 (see also Fig. 7). During the simulation, their positioning errors are recorded. After these first simulations, we applied to the designed prototypes another controller, different from the one dedicated during the design process, in order to see the interest in performing sensor-based design.

**Table 4**
Test point coordinates parameterized with respect to the center of the LRDW for the experiments.

| Point | Coordinate [m] | Point | Coordinate [m] | Point | Coordinate [m] |
|---|---|---|---|---|---|
| $P_1$ | (0,0,-0.05) | $P_8$ | (0,0,0) | $P_{15}$ | (0,0,0.05) |
| $P_2$ | (0.17,0.1,-0.05) | $P_9$ | (0.17,0.1,0) | $P_{16}$ | (0.17,0.1,0.05) |
| $P_3$ | (0.17,-0.1,-0.05) | $P_{10}$ | (0.17,-0.1,0) | $P_{17}$ | (0.17,-0.1,0.05) |
| $P_4$ | (-0.17,-0.1,-0.05) | $P_{11}$ | (-0.17,-0.1,0) | $P_{18}$ | (-0.17,-0.1,0.05) |
| $P_5$ | (-0.17,0.1,-0.05) | $P_{12}$ | (-0.17,0.1,0) | $P_{19}$ | (-0.17,0.1,0.05) |
| $P_6$ | (0,0.2,-0.05) | $P_{13}$ | (0,0.2,0) | $P_{20}$ | (0,0.2,0.05) |
| $P_7$ | (0,-0.2,-0.05) | $P_{14}$ | (0,-0.2,0) | $P_{21}$ | (0,-0.2,0.05) |

In what follows, for the purpose of conciseness, we will only show the result of a line-based visual servoing applied to the robot designed for the image-based moments.

Results are shown and analyzed in the next subsection.

## 7.2. Simulation results

In this section, we denote as:

- **[Case A]:** the robot optimized for leg-direction-based visual servoing controlled with its dedicated controller, all six links $B_{ij}C_{ij}$ ($i = 1, 2, 3$, $j = 1, 2$) are observed,
- **[Case B]:** the robot optimized for line-based visual servoing controlled with its dedicated controller, all six links $B_{ij}C_{ij}$ ($i = 1, 2, 3$, $j = 1, 2$) are observed,
- **[Case C]:** the robot optimized for image-moment-based visual servoing controlled with its dedicated controller,
- **[Case D]:** the robot optimized for image-moment-based visual servoing controlled with the line-based visual servoing, all six links $B_{ij}C_{ij}$ ($i = 1, 2, 3$, $j = 1, 2$) are observed.

We played each simulation for five seconds and recorded the position error. An example of the recorded control signals (velocities of the actuators denoted as Vq1, Vq2, and Vq3) and positioning errors are shown in Fig. 8 and Fig. 9 for the robot of **Case A** going to point $P_4$. We see on these pictures that the robot is converging at around 0.5 s and that after convergence, some oscillation errors are still present due to the simulated observation noise. Since the "sensor-based design" method aims at obtaining the optimal geometric parameters of the robot and the best positioning accuracy of the visual servoing controller, then, during the co-simulation, we only focus on the positioning errors. For each simulated motion towards one of the 24 points, and for each **Case A** to **Case D**, we record the maximal positioning error along the time: for point $P_k$ ($k = 1, \ldots, 24$) simulated in case $\alpha$ ($\alpha = A, B, C, D$), this maximal error is denoted as $\delta_{k\alpha}$. Then, for **Case A** to **Case D**, we summarize the results in Tab. 5: in this Table, we show, for a given value of $\alpha$, the min, max, standard deviation and mean values for the error $\delta_{k\alpha}$ obtained for $k = 1, \ldots, 24$.

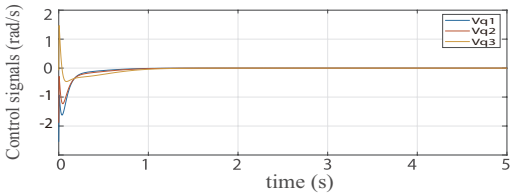



**Figure 8:** Control signals in **Case A** when driven to $P_4$.

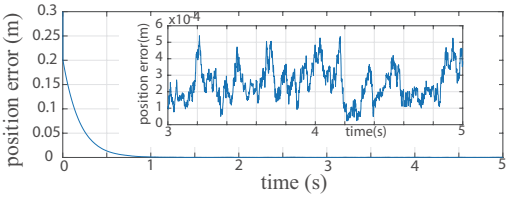



**Figure 9:** Positioning error at point $P_4$ in **Case A**.

**Table 5**
Results of co-simulation in terms of end-effector accuracy: min, max, standard deviation and mean values for the error recorded on the tested 24 points.

| Case | min [mm] | max [mm] | mean [m] | std. deviation [m] |
|---|---|---|---|---|
| A | 0.41 | 0.56 | 0.47 | 0.04 |
| B | 0.36 | 0.56 | 0.45 | 0.05 |
| C | 0.20 | 0.27 | 0.24 | 0.02 |
| D | 0.46 | 0.63 | 0.55 | 0.05 |

To demonstrate the optimal design results, the positioning error is set to be the performance index in the comparison between these visual servoing controllers. From the results, we see that the robot **Case C** leads to the minimal positioning error (both min, max, mean, and standard deviation values). Considering the results of **Case A** and **Case B**, the mean values of the maximal positioning errors are lower than the requested value of 0.5 mm, but there are some points in the workspace for which the error is slightly upper than this limit (maximal error of 0.56 mm in both cases). Indeed, as already mentioned, the accuracy model used during the optimal design process (Section 6) was simplistic and was not able to take into account all potential sources of inaccuracies during the optimal design process. However, even with this simplistic model, the maximal robot positioning error (0.56 mm) is only slightly upper than the threshold of 0.5 mm, while their mean values stay below the accuracy threshold of 0.5 mm. A possibility in order to take into account all the unmodelled effects during the design process would have been to put a safety coefficient on the value of the maximal threshold for positioning error, as it is commonly done in aeronautics for the design of planes.

The results of **Case D** are probably the most interesting. For the robot that optimized for image-moment-based visual servoing controller but controlled with the line-based visual servoing, the mean value of the maximal positioning

error is bigger than the requested value of 0.5 mm, and the maximal value grows up to 0.63 mm, which is bigger than the corresponding values of **Case B** and **Case C**. This result by itself confirms the importance of optimizing a robot for a dedicated sensor-based controller: in other words, the sensor-based design approach of the pair {robot-controller} is necessary in order to ensure the vision-based control accuracy performance.

Based on the optimal design results and the comparison co-simulation results, we see that the Delta robot designed for image moment visual servoing is more compact and has better accuracy performance than the other two robots optimized for other control techniques. However, the differences in robot size and accuracy between the image moment controller and line-based controller are not significant enough to draw general conclusions. Therefore, in the next section, we perform experiments on real prototypes to confirm the results we obtain from the simulations.

## 8. Delta Robot Prototypes and Experimental Validations

This section describes the prototypes of the Delta robot and presents experimental validations on the characterization of their accuracy.

### 8.1. Benchmarks

Based on the remark at the end of Section 6.4 that the footprint of the Delta robot designed for leg-direction-based visual servoing (Case A) is more than twice bigger than the others, meaning that this design is less attractive for the industrial purpose of compactness, we did not prototype it. Instead, we prototyped the robots of Case B (designed for line-based controller, called "Delta L") and Case C (designed for image moments, called "Delta I").

Both prototypes share the same mechanical design concepts and only differ in their geometric parameters. Figure 10 shows the front view of both prototypes. They have been made at low cost: especially, the joints at points $B_{ij}$ and $C_{ij}$ are spherical joints made of ABS plastics, and the internal rotations of the bars $B_{ij}C_{ij}$ are constrained through the use of tension springs. Moreover, a rectified square is mounted on the end-effector of the Delta robot prototype and is used to realize the positioning accuracy measurement (See Fig. 11).

In order to control the motors in the experiments, a dSPACE DS1103 control board is used. The layout of the control board and the way it interacts with the robot is shown in Fig. 12. The motor converters of the prototypes are controlled by the dSPACE control board. This control board is interfaced with an INTERFACE PC through optics fiber which allows to development of the robot controller and sends the reference points to be reached by the robot. Additionally, the control board also communicates with a CAMERA PC which extracts the camera images and processes them.

**Figure 10:** Real prototype in front view; left: Prototype Delta L for line-based visual servoing, right: Prototype Delta I for image moment visual servoing

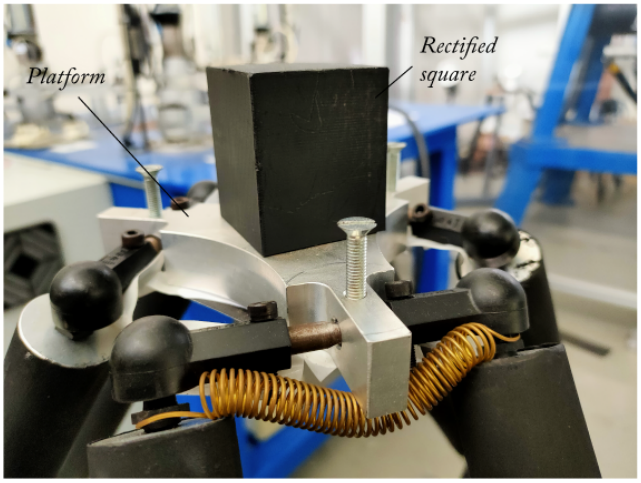



**Figure 11:** Rectified square mounted on the prototype

### 8.2. Camera parameters and data processing

Before performing the accuracy measurement test, it is necessary to properly characterize the camera's intrinsic parameters. The whole calibration process was based on the available VISP library Marchand, Spindler and Chaumette (2005) which implements a camera calibration procedure based on virtual visual servoing Marchand and Chaumette (2001). The camera used in the experiments was a Toshiba Teli BU238M whose resolution is 1920×1200 pixels with a maximum sampling frequency of 165 Hz, with identified intrinsic parameters: $[f_u,\ f_v,\ u_0,\ v_0] = [1395,\ 1396,\ 939,\ 523]$ (see Chaumette and Hutchinson (2006) for more information on those parameters).

Once the camera parameters are found, it is necessary to define some feature trackers in order to extract the necessary information from the image. For line-based visual servoing, a dedicated tracker was implemented, which tracks the links limbs and allows to fit lines to the border subsets of pixels (Fig. 13). For image moment visual servoing, an ellipse tracker is provided by the VISP library Marchand et al.

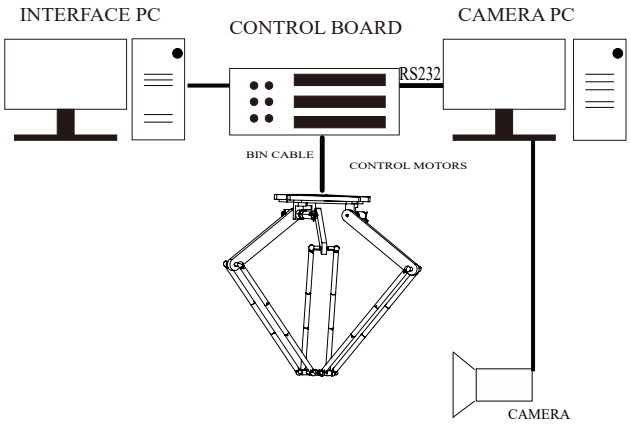



**Figure 12:** Interaction layout for Delta robot prototype

(2005): the coordinate of the centroid point of the ellipse and its area can be computed directly from this library (Fig. 14).

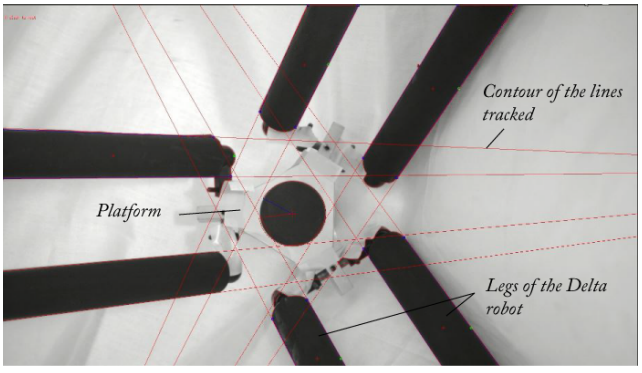



**Figure 13:** Line tracker: The robot legs have been painted in red for a better image processing

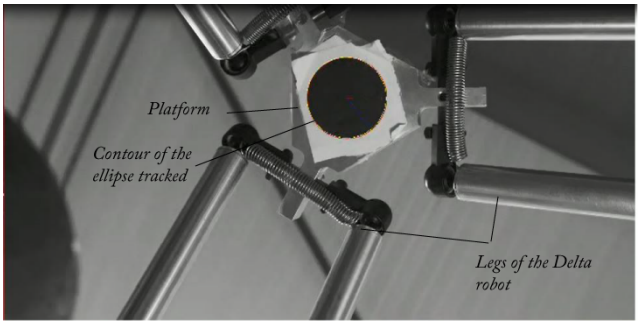



**Figure 14:** Ellipse tracker

## 8.3. Repeatability experiments

In order to measure the prototypes' repeatability, a rectified square was attached to the end-effector of each prototype. The rectified square has its faces parallel to the fixed base main axes, such that three orthogonal micrometers can be used in order to measure its repeatability.

**Table 6**
Results of experiments in terms of repeatability: min, max, standard deviation and mean values for the repeatability computed on the tested 21 points.

| Robot | min [mm] | max [mm] | mean [mm] | std. deviation [mm] |
|---|---|---|---|---|
| Delta L | 0.50 | 0.56 | 0.53 | 0.02 |
| Delta I | 0.22 | 0.39 | 0.30 | 0.05 |

In order to characterize the repeatability performance throughout the workspace, twenty-one points have been selected. Their coordinates parameterized with respect to the LRDW center are given in Tab. 4. The process of all repeatability experiments is the same for both prototypes and is given here:

1. Place the Delta robot prototype in the desired position to be reached using a position controller
2. In this desired final position, record the micrometers values and the visual features that will become the desired visual features $\mathbf{s}^*$.
3. Move the robot to the initial position with the position controller.
4. Return back to the desired position using the visual servoing controller.
5. Read the measures in the micrometers.
   Repeat 3 times the steps from 3 to 5 for an initial position. A total of three initial positions are tested: a first one for approaching the point along X direction, a second one for approaching along Y direction and a last one for approaching along Z axis. Thus we have 9 measurements in total for a given desired final position $\mathbf{s}^*$. With these 9 measurements, we compute an estimation of the robot repeatability for the desired final position $\mathbf{s}^*$. The repeatability for the point $k$ ($k = 1, \ldots, 21$) is denoted as $\delta_k$
6. Repeat all the steps above for all the twenty remaining points in Tab. 4.

We summarize the results in Tab. 6: in this Table, we show, for both prototypes, the min, max, standard deviation, and mean values for the repeatability $\delta_k$ obtained for $k = 1, \ldots, 21$.

The results obtained in Tab. 6 are very close to the simulated one (see Tab. 5, Cases B and C). Note again that, for prototype Delta L, all six parallelogram links are observed by the camera. Analyzing the results, we see that the robot Delta I (designed for image moments) leads to minimal positioning error. For the robot Delta L (design for line-based visual servoing), the mean error is now slightly higher than the requested value of 0.5 mm, even if it stays close to it (0.53 mm). This greater error may come from several factors: as mentioned, the spherical joints used for robot design are very low cost, so they are not free of manufacturing errors (sphericity issues, clearance, etc.). Because line-based visual servoing rebuilds the end-effector pose

from the leg observation, the defects in the spherical joints may partially explain the difference between the model and reality. Note, however, that the error between the simulation and the experiment is small (30 microns). Moreover, once again, the accuracy model used during the optimal design process (Section 6) was really simplistic and was thus the source of computing inaccuracy. However, even with this simplistic model, the maximal robot Delta-L repeatability (0.56 mm) is slightly upper than the threshold of 0.5 mm.

These experimental results confirm those of the simulations: it is clear that robot Delta I has a better repeatability performance than robot Delta L. The results also show that the image-moment-based visual servoing being directly related to the observation of the end-effector, it is less sensitive to errors coming from the robot manufacturing, contrary to the line-based visual servoing. Finally, even if we did not exactly reach our goal of 0.5 mm of repeatability with the Delta L, the final results are close to it, thus confirming the interest in taking into account the controller performance during the design process. A way to improve this last issue would be to use a better error model during the optimization process or to use a safety factor on it.

## 9. Conclusion

In this paper, we performed the "sensor-based design" of a DELTA robot in order to obtain the best accuracy performance of the pair {robot-controller}. We explained that the sensor-based controller performances (accuracy, singularity) are affected by the robot's geometric design parameters. Therefore, when designing a robot, it is necessary to find the optimized geometric parameters of the robot that will allow the best performance of the pair {robot-controller}.

Three different types of sensor-based controllers were proposed to be applied to the DELTA robot: leg-direction-based visual servoing, line-based visual servoing, and image moment visual servoing. For each of these three controllers, positioning error models considering the error of observation coming from the camera were detailed, as well as the controller singularities. Then, the design optimization problem was formulated in order to find the optimal geometric parameters and the relevant parameters of the camera for the DELTA robot for each type of controller. Co-simulations of the robots optimized for the three controllers were performed. In particular, it was shown that, when using a visual servoing controller on a robot designed for another one, the positioning error performance was not guaranteed anymore, confirming the importance of the sensor-based design approach.

Then, prototypes have been manufactured based on the obtained optimum design parameters and experiments have been performed in order to verify the simulation results. The experiment results showed one of the prototypes reached the desired repeatability performance while the other one was really near to getting the desired accuracy. However, even if we did not exactly reach our goal of 0.5 mm of repeatability with the second robot, the final results were close to it, thus confirming the interest in taking into account the controller performance during the design process.

## Acknowledgements

This work has been partially funded by the French ANR project SESAME (funding ID: ANR-18-CE33-0011).

## References

Aginaga, J., Iriarte, X., Plaza, A., Mata, V., 2018. Kinematic Design of a New Four Degree-of-Freedom Parallel Robot for Knee Rehabilitation. ASME Journal of Mechanical Design 140.

Andersen, T., 2015. Sensor based real-time control of robots. Ph.D. thesis. Technical University of Denmark, Departmentof Electrical Engineering.

Andreff, N., Dallej, T., Martinet, P., 2007. Image-based visual servoing of a gough-stewart parallel manipulator using leg observations. The International Journal of Robotics Research 26, 677–687.

Andreff, N., Espiau, B., Horaud, R., 2002. Visual servoing from lines. The International Journal of Robotics Research 21, 679–699.

Andreff, N., Marchadier, A., Martinet, P., 2005. Vision-based control of a gough-stewart parallel mechanism using legs observation, in: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp. 2535–2540. doi:10.1109/ROBOT.2005.1570494.

Baron, N., Philippides, A., Rojas, N., 2021. A Dynamically Balanced Kinematically Redundant Planar Parallel Robot. ASME Journal of Mechanical Design 143.

Bonev, I.A., Ryu, J., 2001. A new approach to orientation workspace analysis of 6-dof parallel manipulators. Mechanism and machine theory 36, 15–28.

Brinker, J., Corves, B., Takeda, Y., 2019. Kinematic performance evaluation of high-speed delta parallel robots based on motion/force transmission indices. Mechanism and Machine Theory 125, 111–125.

Briot, S., Caro, S., Germain, C., 2017. Design procedure for a fast and accurate parallel manipulator. Journal of Mechanisms and Robotics 9, 061012.

Briot, S., Khalil, W., 2015. Dynamics of Parallel Robots – From Rigid Links to Flexible Elements. Springer. ISBN: 978-3-319-19787-6.

Briot, S., Pashkevich, A., Chablat, D., 2010a. Optimal technology-oriented design of parallel robots for high-speed machining applications, in: 2010 IEEE International Conference on Robotics and Automation, pp. 1155–1161. doi:10.1109/ROBOT.2010.5509543.

Briot, S., Pashkevich, A., Chablat, D., 2010b. Technology-Oriented Optimization of the Secondary Design Parameters of Robots for High-Speed Machining Applications, pp. 753–762. URL: https://doi.org/10.1115/DETC2010-28107, doi:10.1115/DETC2010-28107.

Briot, S., Rosenzveig, V., Martinet, P., Özgür, E., Bouton, N., 2016. Minimal representation for the control of parallel robots via leg observation considering a hidden robot model. Mechanism and Machine Theory 106, 115–147.

Cammarata, A., Condorelli, D., Sinatra, R., 2013. An algorithm to study the elastodynamics of parallel kinematic machines with lower kinematic pairs. ASME Transactions Journal of Mechanisms and Robotics 5.

Chaumette, F., 2004. Image moments: a general and useful set of features for visual servoing. IEEE Transactions on Robotics 20, 713–723.

Chaumette, F., Hutchinson, S., 2006. Visual servo control. i. basic approaches. IEEE Robotics & Automation Magazine 13, 82–90.

Chaumette, F., Hutchinson, S., Corke, P., 2016. Visual servoing, in: Springer Handbook of Robotics. Springer, pp. 841–866.

Dahmouche, R., Ait-Aider, O., Andreff, N., Mezouar, Y., 2008. High-speed pose and velocity measurement from vision, in: 2008 IEEE International Conference on Robotics and Automation, pp. 107–112. doi:10.1109/ROBOT.2008.4543194.

Dahmouche, R., Andreff, N., Mezouar, Y., Ait-Aider, O., Martinet, P., 2012. Dynamic visual servoing from sequential regions of interest acquisition. The International Journal of Robotics Research 31, 520–537.

Dallej, T., Andreff, N., Mezouar, Y., Martinet, P., 2006. 3D pose visual servoing relieves parallel robot control from joint sensing, in: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, Beijing,China. pp. 4291–4296.

Di Gregorio, R., 2004. Determination of singularities in delta-like manipulators. The International Journal of Robotics Research 23, 89–96.

French, M., 1999. Conceptual design for engineers. 3rd ed., Springer.

Fusco, F., Kermorgant, O., Martinet, P., 2020. Integrating features acceleration in visual predictive control. IEEE Robotics and Automation Letters 5, 5197–5204.

Garrido, R., Soria, A., Trujano, M., 2008. Visual pid control of a redundant parallel robot, in: 2008 5th International Conference on Electrical Engineering, Computing Science and Automatic Control, pp. 91–96. doi:10.1109/ICEEE.2008.4723397.

Germain, C., Caro, S., Briot, S., Wenger, P., 2013. Optimal Design of the IRSBot-2 Based on an Optimized Test Trajectory. URL: https://doi.org/10.1115/DETC2013-13037, doi:10.1115/DETC2013-13037. v06AT07A056.

Gosselin, C., Angeles, J., 1990. Singularity analysis of closed-loop kinematic chains. IEEE transactions on robotics and automation 6, 281–290.

Han, G., Xie, F., Liu, X.J., Meng, Q., Zhang, S., 2020. Technology-Oriented Synchronous Optimal Design of a 4-Degrees-of-Freedom High-Speed Parallel Robot. ASME Journal of Mechanical Design 142.

Hutchinson, S., Hager, G.D., Corke, P.I., 1996. A tutorial on visual servo control. IEEE Transactions on Robotics and Automation 12, 651–670.

Kaci, L., Boudaud, C., Briot, S., Martinet, P., 2018. Elastostatic modelling of a wooden parallel robot, in: Computational Kinematics. Springer, pp. 53–61.

Laribi, M., Romdhane, L., Zeghloul, S., 2007. Analysis and dimensional synthesis of the delta robot for a prescribed workspace. Mechanism and Machine Theory 42, 859–870.

Liu, X.J., Li, J., Zhou, Y., 2015. Kinematic optimal design of a 2-degree-of-freedom 3-parallelogram planar parallel manipulator. Mechanism and Machine Theory 87, 1–17.

Liu, X.J., Wang, J., Pritschow, G., 2006. Kinematics, singularity and workspace of planar 5r symmetrical parallel mechanisms. Mechanism and Machine Theory 41, 145–169.

Marchand, E., Chaumette, F., 2001. A New Formulation for Non-Linear Camera Calibration Using Virtual Visual Servoing. Research Report RR-4096. INRIA. URL: https://hal.inria.fr/inria-00072535.

Marchand, E., Spindler, F., Chaumette, F., 2005. Visp for visual servoing: a generic software platform with a wide class of robot control skills. IEEE Robotics and Automation Magazine 12, 40–52.

Meng, Q., Liu, X.J., Xie, F., 2022. Design and development of a schönflies-motion parallel robot with articulated platforms and closed-loop passive limbs. Robotics and Computer-Integrated Manufacturing 77, 102352.

Merlet, J., 2006a. Jacobian, manipulability, condition number, and accuracy of parallel robots. ASME Journal of Mechanical Design 128, 199–206.

Merlet, J., Daney, D., 2005. Dimensional synthesis of parallel robots with a guaranteed given accuracy over a specific workspace, in: Proceedings of the 2005 IEEE Conference on Robotics and Automation (ICRA 2005), pp. 942–947.

Merlet, J.P., 2006b. Parallel robots. volume 128. Springer Science & Business Media.

Merlet, J.P., 2012. http://www-sop.inria.fr/members/Jean-Pierre.Merlet/merlet.html.

Michel, H., Rives, P., 1993. Singularities in the determination of the situation of a robot effector from the perspective view of 3 points. Research Report RR-1850. INRIA. URL: https://hal.inria.fr/inria-00074822.

Pascual Escudero, B., Nayak, A., Briot, S., Kermorgant, O., Martinet, P., Chaumette, F., 2021. Complete singularity analysis for the perspective-four-point problem. International Journal of Computer Vision Accepted.

Pashkevich, A., Chablat, D., Wenger, P., 2009. Stiffness analysis of overconstrained parallel manipulators. Mechanism and Machine Theory 44, 966–982.

Pierrot, F., Reynaud, C., Fournier, A., 1990. Delta: A simple and efficient parallel robot. Robotica 8, 105–109.

Qi, Z., McInroy, J., 2008. Improved image based visual servoing with parallel robot. Journal of Intelligent Robot Systems 53, 359–379.

Qiang, Z., Ehmann, K.F., Jian, C., 2014. Tri-pyramid robot: Design and kinematic analysis of a 3-dof translational parallel manipulator. Robotics and Computer Integrated Manufacturing 30, 648–657.

Rosenzveig, V., Briot, S., Martinet, P., 2013. Minimal representation for the control of the adept quattro with rigid platform via leg observation considering a hidden robot model, in: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 430–435. doi:10.1109/IROS.2013.6696387.

Tahri, O., Chaumette, F., 2005. Point-based and region-based image moments for visual servoing of planar objects. IEEE Transactions on Robotics 21, 1116–1127.

Traslosheros, A., Sebastian, J.M., Angel, L., Roberti, F., Carelli, R., 2007. Visual servoing of a parallel robot system, in: 2007 IEEE International Symposium on Intelligent Signal Processing, pp. 1–6. doi:10.1109/WISP.2007.4447643.

Trujano, M., Garrido, R., Soria, A., 2012. Robust visual control of parallel robots under uncertain camera orientation. International Journal of Advanced Robotic Systems 9.

Vignolo, A., Briot, S., Philippe, M., Chen, C., 2014. Comparative Analysis of two Types of Leg-observation-based Visual Servoing Approaches for the Control of the Five-bar Mechanism, in: 2014 Australasian Conference on Robotics and Automation (ACRA 2014), Melbourne, Australia. URL: https://hal.archives-ouvertes.fr/hal-01074335.

Wang, M., Dong, X., Ba, W., Mohammad, A., Axinte, D., Norton, A., 2021. Design, modelling and validation of a novel extra slender continuum robot for in-situ inspection and repair in aeroengine. Robotics and Computer-Integrated Manufacturing 67, 102054.

Wu, J., Wang, J., Wang, L., Li, T., 2009. Dynamics and control of a planar 3-dof parallel manipulator with actuation redundancy. Mechanism and Machine Theory 44, 835–849.

Wu, J., Wang, X., Zhang, B., Huang, T., 2021. Multi-objective optimal design of a novel 6-dof spray-painting robot. Robotica 39, 2268–2282. doi:10.1017/S026357472100031X.

Wu, J., Ye, H., Yu, G., Huang, T., 2022. A novel dynamic evaluation method and its application to a 4-dof parallel manipulator. Mechanism and Machine Theory 168, 104627.

Wu, J., Yu, G., Gao, Y., Wang, L., 2018. Mechatronics modeling and vibration analysis of a 2-dof parallel manipulator in a 5-dof hybrid machine tool. Mechanism and Machine Theory 121, 430–445.

WU, J., ZHANG, B., WANG, L., YU, G., 2021. An iterative learning method for realizing accurate dynamic feedforward control of an industrial hybrid robot. SCIENCE CHINA Technological Sciences 64, 1177–1188.

Wu, Y., Gosselin, C.M., 2004. Synthesis of reactionless spatial 3-dof and 6-dof mechanisms without separate counter-rotations. The International Journal of Robotics Research 23, 625–642.

Zhu, M., Chriette, A., Briot, S., 2021. Control-based design of a delta robot, in: Venture, G., Solis, J., Takeda, Y., Konno, A. (Eds.), ROMANSY 23 - Robot Design, Dynamics and Control, Springer International Publishing, Cham. pp. 204–212.

Zlatanov, D., Fenton, R., Benhabib, B., 1995. A unifying framework for classification and interpretation of mechanism singularities. ASME Journal of Mechanical Design 117, 566–572.