Control-based design of Robots

Minglei Zhu

THESE DE DOCTORAT DE

L'ÉCOLE CENTRALE DE NANTES

ECOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Automatique, productique et robotique*

Par

Minglei ZHU

Control-based design of Robots

Thèse présentée et soutenue à Nantes, le 15 Decembre 2020
Unité de recherche : UMR 6004, Laboratoire des Sciences du Numérique de Nantes (LS2N)

Rapporteurs avant soutenance

Nicolas Andreff	Professeur des universités – Université de Franche-Comté, Besançon
Philippe Martinet	Directeur de recherche – INRIA Sophia Antipolis

Composition du Jury

Président:	Philippe Wenger	Directeur de recherche – École Centrale de Nantes
Examinateurs:	Claire Dune-Maillard	Maître de conférences – Université de Toulon
Dir. de thèse:	Sébastien Briot	Chargé de recherche HDR –École Centrale de Nantes
Co-encadrant:	Abdelhamid Chriette	Maître de conférences – École Centrale de Nantes

Abstract

It is well-known that parallel robots have a lot of applications in industry for their high stiffness, high payload, can reach higher acceleration and speed. However, because of their complex structure, their control may be troublesome. When high accuracy is needed, the detailed robot model is necessary. However, even detailed models still suffer from the problem of inaccuracy in reality because of robot assembly and manufacturing errors. Sensor-based control approaches have been proven to be more efficient than model-based controllers in terms of accuracy since they overcome the complex robot models and inconsistency errors. Nevertheless, when applying the visual servoing, there are always some problems in the control process, such as the controller singularities. Thus, this thesis proposes a control-based design methodology which takes into account the accuracy performance of the controller in the design process to get the optimal geometric parameters of the robot.

This thesis applied the control-based design methodology to the optimal design of three types of parallel robots: Five-bar mechanisms, DELTA robots, Gough-Stewart platforms. Two types of controllers are envisaged for the control of the motions of the Five-bar mechanisms: leg-direction based visual servoing and line-based visual servoing. For DELTA robots and Gough-Stewart platforms, three types of controllers are selected: leg-direction-based visual servoing, line-based visual servoing and image moment visual servoing. Based on these selected controllers, positioning error models taking into account the error of observation coming from the camera are developed and the controller singularities are studied. Then, design optimization problems are formulated in order to find the optimal geometric parameters and camera placement for these three types of parallel robots for each type of controller. Co-simulations of the robots optimized for the corresponding controllers are performed to check the accuracy performance of the robots obtained from the optimization.

Two DELTA robot prototypes are designed and the experiments are performed with these two robots in order to validate the controller accuracy. The experiment results confirm the controller performance obtained from the co-simulation and prove that the image moment visual servoing is the best controller for the control of DELTA robot

compared with leg-based visual servoing.

Keywords: parallel robots, visual servoing, control-based design, hidden robot, image moment

Résumé

Il est bien connu que les robots parallèles ont de nombreuses applications dans l'industrie grâce à leur rigidité élevée, leur charge utile élevée et leur capacité à atteindre des accélérations et vitesses élevées. Cependant, en raison de leur structure complexe, leur contrôle peut être difficile. Lorsqu'une précision élevée est nécessaire, un modèle complet du robot détaillé est nécessaire. Cependant, même un modèle détaillé soffre toujours d'inexactitudes par rapport à la réalié à cause d'erreurs d'assemblage et de fabrication du robot. Les approches de contrôle référencées capteurs se sont avérées plus efficaces, en termes de précision; que les contrôleurs basés modèles puisqu'elles s'affranchissent des modèles de robots complexes et des erreurs de modélisation associées. Néanmoins, lors de l'application d'un asservissement visuel, il y a toujours des problèmes dans le processus de contrôle, tels que les singularités du contrôle. Cette thèse propose une méthodologie de conception orientée commande qui prend en compte les performances de précision du contrôle dans le processus de conception du robot pour obenir les paramètres géométriques optimaux de ce dernier. Dans le cadre ce travail de thèse, il a été question d'appiquer la méthodologie de conception orientée commande à la conception optimale de trois types de robots parallèles: le mécanisme cinq barres, le robot DELTA, et, enfin, la plate-forme de Gough-Stewart. Deux types de contrôleurs ont été envisagés pour le contrôle des mouvements des mécanismes cinq barres : les commandes basées sur l'observation des directions des jambes du robot et les commandes basées sur l'observation des lignes droites. Pour les robots DELTA et les plates-forme de Gough-Stewart, trois contôleurs ont été sélectionnés: les commandes basées sur l'observation des directions des jambes, les commandes basées sur l'observation des lignes et les commandes basées sur des moments dans l'image. A partir de ces contrôleurs, des modèles d'erreur de positionnement prenant en compe l'erreur d'observation provenant de la caméra ont été développés et les singularités des contrôleurs ont été étudiées. Ensuite, les problèmes d'optimisation de la conception ont été formulés afin de trouver à la fois les paramètres géométriques optimaux et le placement optimal de la caméra pour ces trois types de robots parallèles et pour chaque type de contrôleur. Pour vérifier les performances en terme de précision des robots optimisés, nous avons effectué des co-simulations des robots optimisés avec les contrôleurs correspondants.

En terme d'expérimentation, deux prototypes de robots DELTA ont été conçus et expérimentés afin de valider la précision du contrôleur. Les résultats des expériences menées ont permis la validation des performances du contrôleur obtenues à partir de la co-simulation et ont prouvé que l'asservissement visuel basé moment dans l'image est le meilleur contrôleur pour le contrôle du robot DELTA en comparaison des commandes basées sur l'observation des jambes.

Mots-clés: robots parallèles, asservissement visuel, conception orientée command, robot caché, moment dans l'image

Acknowledgements

Before beginning this manuscript, I would like to express some words to the people who have guided me, shared time with me, and supported me during the past three years.

Firstly, I would like to express my great appreciation to my thesis director, Sébastien Briot for his valuable and constructive guidance during the planning and development of this thesis. With his useful critiques and comprehensive experience on dynamics, he has been a great mentor throughout this journey. I am extremely grateful for the balance he has put between guiding me and encouraging me to be autonomous during this research.

I would like to thank my co-supervisors, Abdhelhamid Chriette, for the time they have shared with me, their patience and their wise advises provided for the development of this work and during the editing of the research articles. Abdhelhamid Chriette is the most friendly supervisor I've ever meet and gave me a lot of encouragement when I have difficulty in my research.

I would like to express my gratitude also to Stéphane Jolivet, David Llevat and Philippe Lemoine for their patience to help me to build and to implement the prototype for the experimental validations. Their combined advises helped me to improve the different parts of the prototype, including the mechanics and electronics.

My PhD life would not have been the same without my colleages. I would like to address special thanks to Zhongmou, Julian, Franco, David, Guillaume, Shiyu, Haixin, Song Ming for having interesting discussions and sharing beautiful moments with me. And the most important, Rafeal, who helps me a lot in my research. He is the most accommodating friend.

I would like to express a special thank to the China Scholarship Council (CSC) for sponsoring my PhD studies, their support was very important during these three years of PhD studies.

Finally, I would like to express all my gratitude and love to my family: my wife Chen Zhong, my son Zhu ti, my parents and my wife's parents. My wife gave up her career in China and stay with me during the last two years. My son Zhuti likes an angel, he makes my life more warm. My parents and my wife's parents support me a lot in finance and encourage me during the past time. I am eternally grateful to them for their love, education and never ending support.

Table of Contents

List of Figures

List of Tables

Introduction

Context of the thesis

Parallel robots have several advantages compared with serial robots, such as they can reach high speed and acceleration, better payload and are stiffer. Nevertheless, the classical model-based control methodology takes the robot model to approximate the relationship between the joint and the end-effector and it is always troublesome because of the complex structure of the parallel robot and its highly non-linear input/output relationships. In general, the only way to get high accuracy is to get a detailed robot model. However, even detailed models still suffer from the problem of inaccuracy in reality because of robot assembly and manufacturing errors. To bypass the complex kinematic architecture and to get a better accuracy compared with the model-based controllers, the external sensor-based controller is proposed to control the parallel robot. Visual servoing is such an external sensor-based controller which takes the camera as the sensor and closes the control loop by using the image features obtained from the camera.

In recent years, several types of image features have been proposed to have good performance in terms of accuracy for controlling the parallel robot, for example the observation of the robot legs and the application of the image moments. When we observe the robot legs, the features that can be extracted from the observation of the legs are their directions or the lines passing through the leg axes. The 2-D image moments of a dense object or a discrete model can be set as the image features when we use the image moment visual servoing.

When visual servoing is applied to the control of parallel robots, there are several problems which affect the accuracy. One problem is that the controller singularity [Chaumette 1998] appears when the interaction matrix is rank deficient. This singularity problem of the mapping between the observed image feature space and the Cartesian space has great influence on the accuracy of visual servoing. Thanks to the work of [Briot 2013], a tool named "Hidden robot" was developed in order to simplify the study of the controller singularity problem when visual servoing is applied on the control of the parallel robot. It reduces the study of the singularities of the interaction matrix to the

study of the singularities of the virtual parallel robot hidden in the controller.

Another problem is that the positioning error of visual servoing which comes from the camera observation capacities and the interaction model. It is one of the most important internal performances of the visual servoing controller.

The optimal design methodology of the robot aims at getting the optimal geometric parameters of the robot. When visual servoing is applied to the control of parallel robots, these two problems should be considered in advance: the controller singularity (which depends on the type of the controller and the robot structure), internal performance (especially the positioning accuracy) (which depends on the type of the controller and the geometric parameters of the robot). In addition, the vision-based controller is never taken into account in the optimal design process before. Thus we should develop a novel optimal design methodology "Control-based design", which is an optimal design methodology taking the controller performance as the indices in the design process in order to get the optimal geometric parameters of parallel robots and avoid the instability issues appeared in the control process.

In this thesis work, the control-based design methodology is applied in order to design three types of parallel robots with the best accuracy performance for the dedicated controller. The three types of parallel robots are the Five-bar mechanism, the DELTA robot and the Gough-Stewart platform. In addition, prototypes of DELTA robot are designed and experiments are performed in order to validate the controller performance.

Contributions of the thesis

This manuscript presents several major contributions which are listed as follows:

- **The optimal design of Five-bar mechanisms:** The optimal design of a Five-bar mechanism has been performed in [Kaci 2017], [Kaci 2018] and in her PhD Thesis on the control-based design of a Five-bar mechanism. In chapter 2, we revisit the optimization of the Five-bar mechanism for several values of the pixel noise to see the impact on the design. We tested several controllers based on different vision information to see which one is the best for the control of the Five-bar mechanism. The positioning error models (see definition in 2.4) and the hidden robot (see definition in 1.2.5) of the Five-bar mechanism are taken into account to avoid the instability issues appeared in the final design of the robot. Then co-simulations are performed for the robots optimized to check the accuracy.

- **The optimal design of DELTA robots and its experimental validation of the design approach:** DELTA robot is a spatial parallel robot with three degree of freedoms. In chapter 3, the control-based design methodology is applied to the design of a DELTA robot with the best accuracy performance for a dedicated controller. Three visual servoing controllers are selected in this work: 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. With the help of the hidden robot, the controller singularities of the DELTA robot have been studied. Then the design optimization problems are formulated in order to find the optimal geometric parameters and camera placement for the DELTA robot for each type of controller. Co-simulations of the robots optimized for these three types of controllers are performed to check the accuracy performance of the three robots obtained from the optimization. Compared to the previous work, this is the first time that the control-based design methodology is applied to the optimal design of a spatial parallel robot and this is the first time that image moment visual servoing is taken into account as one of the controllers in control-based design. In addition to the numerical simulations, experimental prototypes of the DELTA robot obtained from the optimization were designed and commissioned to perform the experiments in order to validate the controller accuracy. The mechanical design of the prototypes, the control system, experiment processes and results are shown in chapter 4. The experiment results confirmed the controller performance obtained from the co-simulation and proved that the image moment visual servoing is the better controller for the control of DELTA robot compared with line-based visual servoing.

- **Optimal design of Gough-Stewart platforms:** The Gough-Stewart platform is a spatial parallel robot with six degree of freedoms. In chapter 5, we perform control-based design of a Gough-Stewart platform in order to obtain the best accuracy performance of the robot with its controller. In this work, three visual servoing controllers are selected: leg-direction-based visual servoing, line-based visual servoing and image moment visual servoing. Positioning error models taking into account the observation error and the corresponding hidden robot are studied to be sure that no instability issues appeared in the final design of the Gough-Stewart platform. Then we formulate the optimal design problem to find the optimal geometric parameters, the camera placement for the Gough-Stewart platform for each type of

controller and the configuration of the discrete three-point-model to be observed in the image moment visual servoing. Then, we perform co-simulations of the robots optimized for these three types of controllers to check the accuracy performance of the three robots obtained from the optimization. In addition, in order to test the robustness of the accuracy of the model with geometry errors, we perform the same co-simulations on the models with errors on the joints. Compared to the previous work, this is the first time that the control-based design methodology is applied to the optimal design of a spatial parallel robot with six DOFs and this is the first time that the configuration of the discrete three-point-model to be observed in image moment visual servoing is taken into account in control-based design.

This thesis work focus on the optimal design of parallel robots using the control-based design methodology and takes the visual servoing controller as the dedicated controller. Then, let us now start with a bibliographical overview of parallel robots, visual servoing and optimal design of parallel robots.

Chapter 1

State of the art

This Chapter is dedicated to present review on the parallel robot. Introducing the definition of the parallel robot and cetain bibliographical reviews on the well-known parallel robots. In addition, the geometric model of the parallel robot, inverse kinematic and direct kinematic models of the parallel robot and different kinds of singularities configurations will be reviewed. After that, different classical control methods of the parallel robot are presented.

In the second part of this Chapter, it will be a presentation on the visual servoing. First, we will recall the advantages of applying visual servoing on the control of the parallel robot compared with the classical control method. Then presenting different kinds of image features (such as the points, lines, leg-directions and image moment) and how to get the interaction matrix related to these features. Additionally, the recalls of leg-based visual servoing , line-based visual servoing and image moment visual servoing will be proposed. Finally, the controller singularity problem is presented and a powerful tool "hidden robot" which helps simplify the study of controller singularity is presented in details.

In the end, there is a presentation on the optimal design of parallel robots.

1.1 Parallel robot

1.1.1 Basics of parallel robots

The robot is playing a more and more important role in our everyday life, it can be found in Aeronautics and Astronautics, industrial manufacture, medical service and daily life. More and more research works focus on designing a robot with better performance that can serve people's life more efficiently.

For a mechanical system, the end-effector moves with respect to the fixed base in various ways, in translation or rotary motion, which are called its degrees of freedom (DOF). The total number of degrees of freedom of a rigid body in space cannot exceed 6 (three translations along the mutually orthogonal axes and three rotations around these axes). A robot is such a mechanical system that control several degree of freedoms of the end-effector.

In terms of the structure, we can categorise robots into serial robots and parallel robots. The serial robot is the mechanical system designed as a series of links connected by motor-actuated joints that extend from a base to an end-effector. Different from the serial robot, a generalized parallel robot is a closed-loop kinematic chain mechanism whose end-effector is linked to the base by several independent kinematic chains [Merlet 2006]. In addition, parallel robots for which the number of chains is strictly equal to the number of DOF of the end-effector are called fully parallel manipulators [Gosselin 1988].

Parallel robots have several advantages in features compared with serial robots, such as they have high stiffness, high payload, can reach higher acceleration and speed. Nevertheless, the main disadvantage of parallel robots is abundant singularity problems. In addition, the small workspace, complex structure, complex forward kinematics, difficulties of control and complex dynamics [Merlet 2006] when compared with serial structures.

The planar parallel robot is the parallel robot whose end-effector moves within a plan. Five-bar mechanism is such a planar parallel robot with 2 DOFs and its industrial application is the dual-arm SCARA academic robot (see Fig. 1.1). This SCARA robot is designed for finishing high speed pick-and-place job.

The spatial parallel robot is the parallel robot whose end-effector moves in space. In 1986 the Swiss scientist Reymond Clavel conceived the so-called Delta robot [Clavel 1990]. (see Fig. 1.2). The key concept of the delta robot is the application of parallelograms which restrict the movement of the end platform to pure translation, i.e. only movements in the $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ directions with no rotation. The robot's base is mounted above the workspace and

Figure 1.1: Dual-arm SCARA robot

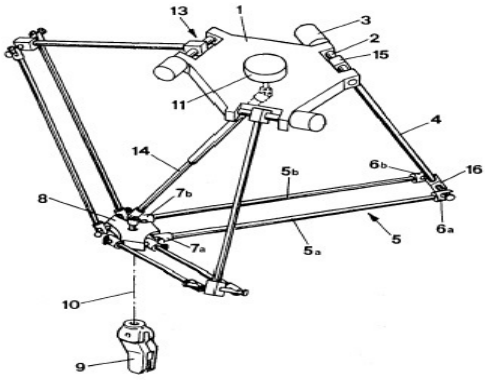

Figure 1.2: Schematic of patent of Delta robot by Reymond Clavel [Clavel 1990]

Figure 1.3: ISIS/SurgiScope system using a Delta robot as microscope stand

Figure 1.4: A Delta type 3D printer

all the actuators are located on it. From the base, three middle jointed arms extend. The ends of these arms are connected to a small triangular platform. Actuation can be done with linear or rotational actuators, with or without reductions (direct drive). Since the actuators are all located in the base, the arms can be made of a light composite material. As a result of this, the moving parts of the delta robot have a small inertia which allows very high speed and high accelerations. Having all the arms connected together to the end-effector increases the robot stiffness, but reduces its working volume.

Industries that take advantage of the high speed of delta robots are the packaging industry, medical and pharmaceutical industry (Fig. 1.3). Other applications include high precision assembly operations in a clean room for electronic components. More recently, the technology has been adapted to 3D printers (Fig. 1.4). From 1980, The Delta robot probably become the most popular robot (more than 4000 robots sold in 20 years).

Another classical spatial parallel robot is the Gough-Stewart platform with 6 DOFs (Fig. 1.5). It consists of six prismatic actuators, commonly hydraulic jacks or electric actuators, attached in pairs to three positions on the platform's baseplate, crossing over to three mounting points on a top plate. Devices placed on the top plate can be moved in a six degrees of freedom (three linear movements $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ (lateral, longitudinal and vertical), and the three rotations (pitch, roll, yaw)). The Gough-Stewart platform is extensively used in flight simulation, particularly in the so-called full flight simulator for which all 6 degrees of freedom are required (Fig. 1.6). The largest driving simulator is the National Advanced Driving Simulator (NADS) at the University of Iowa (Fig. 1.7). For the applications in industry, the first milling machine was presented by the Giddings and Levis company at the IMTS machine-tool exhibition in Chicago in 1994, under the name of *Variax* (see Fig. 1.8). It was based on the principle of the Gough-Stewart platform, which was 5 times stiffer than the classical machine and had much superior advance speed.

1.1.2 Geometric modeling and Kinematic modeling of parallel robots

1.1.2.1 Geometric modeling of parallel robots

In order to represent the pose of a rigid body through a set of parameters $\mathbf{X}$, there are multiple ways. One is to use the coordinates in a reference frame of a given point of the body, and three angles to represent its orientation (such as Euler angles). The geometric

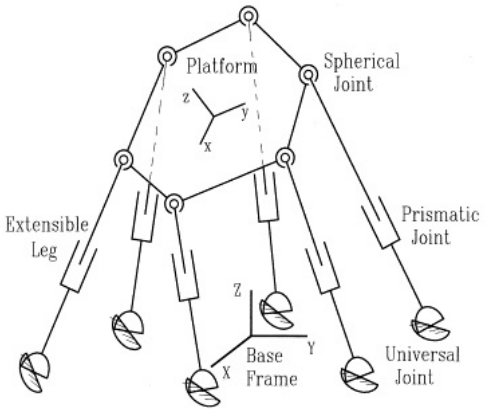

Figure 1.5: 6 UPS Gough Stewart platform [Dasgupta 2000]

Figure 1.6: The Airbus A340 simulator [Merlet 2006]

Figure 1.7: The NADS driving simulator at the University of Iowa

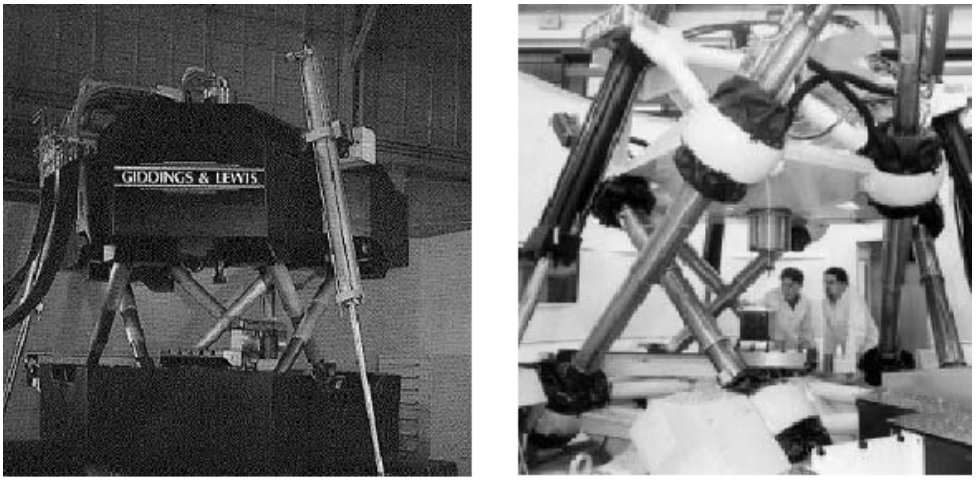

Figure 1.8: The Variax of Gidding and Lewis, the first industrial parallel milling machine presented in 1994 at IMTS [Merlet 2006]

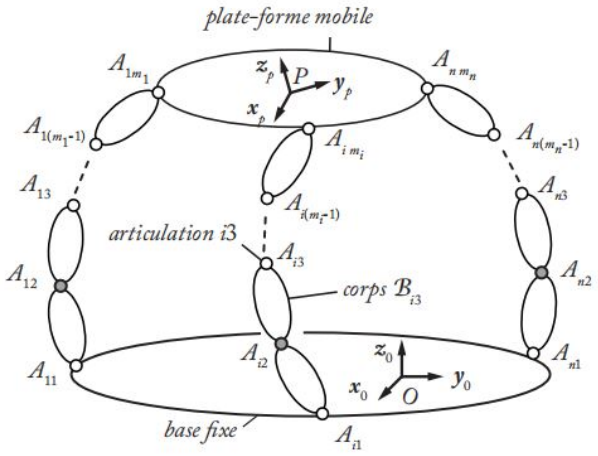

Figure 1.9: General model of a parallel robot

model of the parallel robot gives the relation between the actuated joint coordinates and the pose of the end-effector.

The inverse geometric model of parallel robots gives the value of the joint coordinates corresponding to a given end-effector configuration. For the general analytic method, we consider that each chain links to the moving platform, $\mathbf{A}_{i1}$ is the point which represents the end of the chain linked to the base, and another point $\mathbf{A}_{im_i}$ (Fig. 1.9) which represents the end of the chain linked to the moving platform. $\mathbf{A}_{i1}$ can be expressed in the fixed global frame whose original point is $\mathbf{O}$, at the same time the coordinates of $\mathbf{A}_{im_i}$ can be determined from the moving platform position and orientation. Then we can have

$$\overrightarrow{\mathbf{A_{i1}A_{im_i}}} = \overrightarrow{\mathbf{A_{i1}O}} + \overrightarrow{\mathbf{OA_{im_i}}} = \mathbf{H_1(X)} \tag{1.1}$$

Where $\overrightarrow{\mathbf{A_{i1}A_{im_i}}}$ is the vector and $\mathbf{X}$ represents the generalized coordinates of the moving platform. With the equation (1.1), we can always find the positions of extreme points of any chain for which we want to calculate the joint coordinates (the actuated joint coordinate in most cases).

With the chain joint coordinates $\mathbf{\Theta}$, we can determine the vector $\overrightarrow{\mathbf{A_{i1}A_{im_i}}}$, if needed, with the help of $\mathbf{X}$:

$$\overrightarrow{\mathbf{A_{i1}A_{im_i}}} = \mathbf{H}_2(\mathbf{X}, \mathbf{\Theta}) \tag{1.2}$$

The joint coordinate can thus be calculated by solving the following system of equations:

$$\mathbf{H}_1(\mathbf{X}) = \mathbf{H}_2(\mathbf{X}, \mathbf{\Theta}) \tag{1.3}$$

The direct geometric model of parallel robots address the problem of determining the pose of the end-effector $\mathbf{X}$ of a parallel-robot from its actuated joint coordinates $\mathbf{\Theta}$. In spite of the fact that this problem is very important in the control of parallel robots, there is no unique solution. However, depending on the architecture, there exists specialised methods to solve this problem (see details in [Merlet 2006]).

1.1.2.2 Kinematic modeling of parallel robots

The inverse kinematic model aims at determining the joint velocities from the velocity of the end-effector $\boldsymbol{\tau} = [\mathbf{v}, \boldsymbol{w}]$, $\mathbf{v}$ is the linear velocity vector and $\boldsymbol{w}$ is the rotational velocity vector.

Through the derivation of the equation (1.3), we obtain

$$\mathbf{A}(\mathbf{X}, \mathbf{\Theta})\dot{\mathbf{\Theta}}_a + \mathbf{B}(\mathbf{X}, \mathbf{\Theta})\dot{\mathbf{X}} + \mathbf{C}(\mathbf{X}, \mathbf{\Theta})\dot{\mathbf{\Theta}}_p = 0 \tag{1.4}$$

where $\mathbf{\Theta}_a$ is the vector of active joints and $\mathbf{\Theta}_p$ is the vector of passive joints. In most cases, $\dot{\mathbf{X}} \neq \boldsymbol{\tau}$. Then to simplify it, we can choose to reduce $\mathbf{\Theta}$ to $\mathbf{\Theta}_{\boldsymbol{a}}$, then we have

$$\mathbf{A}\dot{\mathbf{\Theta}}_{\boldsymbol{a}} + \mathbf{B}\boldsymbol{\tau} = 0 \tag{1.5}$$

In this case, both matrices $\mathbf{A}$ and $\mathbf{B}$ are square, and provided that $\mathbf{A}$ is invertible we have

$$\dot{\mathbf{\Theta}}_a = -\mathbf{A}^{-1}\mathbf{B}\boldsymbol{\tau} = \mathbf{J}_{\text{inv}}\boldsymbol{\tau} \tag{1.6}$$

where $\mathbf{J}_{\text{inv}}$ is the inverse Jacobian matrix.

Moreover, there is another way to calculate the inverse Jacobian matrix. Consider the velocity of a point $\mathbf{A}_{in,n}$ which is the end of the ith chain on the end-effector

$$\mathbf{v}_{\mathbf{A}_{in,n}} = \mathbf{v}_C + \overrightarrow{\mathbf{A}_{in,n}\mathbf{C}} \times \boldsymbol{w} \tag{1.7}$$

Then transform this equation into the matrix form as a function of the end-effector velocity:

$$\mathbf{v}_{\mathbf{A}_{in,n}} = \mathbf{J}_{\mathbf{A}_{in,n}C}\boldsymbol{\tau} \tag{1.8}$$

we can also get the velocity of the point $\mathbf{A}_{in,n}$ as a function of joint velocities (active and passive)

$$\mathbf{v}_{\mathbf{A}_{in,n}} = \mathbf{J}_i\dot{\mathbf{\Theta}} \tag{1.9}$$

The above two equations can lead to:

$$\mathbf{J}_{\mathbf{A}_{in,nC}}\boldsymbol{\tau} = \mathbf{J}_i\dot{\boldsymbol{\Theta}} \tag{1.10}$$

Applying the equation (1.10) into all the chains of the parallel robot, it is possible to construct the inverse Jacobian matrix which relates the end-effector velocity $\boldsymbol{\tau}$ to the joint velocities $\dot{\boldsymbol{\Theta}}$.

1.1.3 Singular configurations of parallel robots

As we present above, the main disadvantage of parallel robots is the large number of singular configurations. They are particular poses of the manipulator, for which parallel robots lose their inherent infinite rigidity and in which the end-effector will have uncontrollable degrees of freedom.

Gosselin and Angeles presented a study of singularity which based on the restriction to the active joint $\boldsymbol{\Theta}_\mathbf{a}$ and to the generalized coordinates $\mathbf{X}$ that describes the end-effector motion. With the help of equation (1.5), they distinguish 3 different types of kinematic singularity:

1. $\mathbf{A}$ is singular (called Type 1 or serial singularity): there will be a non-zero velocity vector $\dot{\boldsymbol{\Theta}}_a$ for which the platform does not move.

2. $\mathbf{B}$ is singular (called Type 2 or parallel singularity): there will then be a non-zero end-effector velocity $\boldsymbol{\tau}$ for which the joint velocities are zero. In the neighborhood of such a configuration, the robot will be able to have an infinitesimal motion while the actuators are locked which means that the robot gains some degrees of freedom. Because of this, certain degrees of freedom of the end-effector cannot be controlled. This type of singularity is a major problem in the research of parallel robots [Matone 1999].

3. both $\mathbf{A}$ and $\mathbf{B}$ are singular(Type 3 singularity): the end-effector may be moved while the actuators are locked, and vice versa.

In a serial singularity, a zero twist of the end-effector is obtained for non-zero actuated joint velocities. It has been often claimed that such a singularity corresponds to a "workspace limit" [Merlet 2006]. The parallel singularity affects a lot in control issues

because the pose of the end-effector is no longer controllable. Moreover, the elements of the robot may be submitted to very large forces, causing a breakdown of the robot.

The singularities we present above are kinematic singularities. In [Zlatanov 1994], a more complete analysis of singularities has been performed by using the full range of possible instantaneous motions of a parallel robot with a full-cycle mobility [Hunt 1978] of n and N total joint variables (including passive and active joints). This lead to the analysis of an $(N+n)$-tuple velocity vector $[\boldsymbol{\tau}^T \ \dot{\boldsymbol{\Theta}}_a^T \ \dot{\boldsymbol{\Theta}}_p^T]^T$. where $\dot{\boldsymbol{\Theta}}_a$, $\dot{\boldsymbol{\Theta}}_p$ and $\boldsymbol{\tau}$ represent the active joint velocities, the passive joint velocities and the output velocities, respectively. Under this direction, we identify six different types of singularities:

- redundant input singularity (RI): for the manipulator whose input is non-zero but has a motion of zero output;

- redundant output singularity (RO): for the manipulator whose input is zero but with a motion of non-zero output (the same as Type 2 singularity)

- impossible input singularity (II): when the motion admits n-dimensional vectors that can not be applied as input;

- impossible output singularity (IO): when the motion admits n-dimensional vectors that can't be applied as input;

- increased instantaneous mobility (IIM): the mechanism is in an uncertainty configuration and the instantaneous mobility is greater than the full-cycle mobility;

- redundant passive motion (RPM): the mechanism is in a configuration that admits a non-zero motion with zero input and zero output.

The singularity analysis helps getting a better geometrical understanding of the singularities, which may lead to a systematic discovery of these singularities and to determining what will be the possible motion of the end-effector at a singularity loci. Another contribution of this study is that it helps investigating the relation between a set of poses and singularities. For example we may have to check if the robot workspace or a robot trajectory does not include a singularity, or to find the largest cube in the robot workspace that is singularity-free, which is very important in the optimal design of a parallel robot.

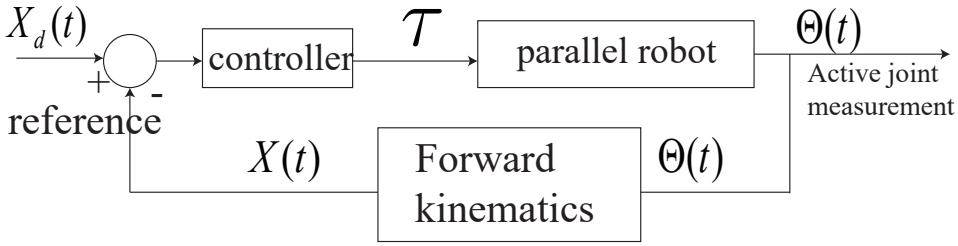

Figure 1.10: The general model-based control when joint variable $\mathbf{\Theta}$ is measured

1.1.4 Control of parallel robots

Parallel robots are designed for the applications that the moving platform of the robot accurately follows a desired position and orientation path in a specific time frame. The generalized coordinate of the moving platform can be defined by considering the motion variable $\mathbf{T}$. Then we have the moving platform velocity $\dot{\mathbf{T}}$ the derivative of the motion variable. Considering the general closed-form dynamic formulation of a parallel robot

$$\mathbf{M}(\mathbf{T})\ddot{\mathbf{T}} + \mathbf{C}(\mathbf{T}, \dot{\mathbf{T}})\dot{\mathbf{T}} + \mathbf{G}(\mathbf{T}) = \mathbf{F} \tag{1.11}$$

where $\mathbf{M}(\mathbf{T})$ is the mass matrix, $\mathbf{C}(\mathbf{T}, \dot{\mathbf{T}})$ is the Coriolis and centrifugal matrix, $\mathbf{G}(\mathbf{T})$ is the gravity vector. $\mathbf{F}$ denotes the generalized forces applied to the end-effector center of mass [Taghirad 2013].

In order to control the motion of a parallel robot, the required actuator forces or torques must be obtained by the controller to drive the robot to a desired position and orientation trajectory. In general, the desired position of the end-effector can be represented by the desired generalized coordinate of the moving platform $\mathbf{X}_d$. As defined in (1.2), $\mathbf{\Theta} = [\Theta_1, \Theta_2 \cdots \Theta_n]$ is the set of joint coordinates. The relationship between the joint variables and the motion variables of the moving platform $\mathbf{X}$ has been detailed studied through the inverse and forward kinematics. With the help of the encoder, the joint variables $\mathbf{\Theta}$ can be measured and $\mathbf{X}$ can be obtained from the forward kinematic analysis. The control topology is defined in Fig. 1.10. The joint variable measurement $\mathbf{\Theta}$ is used as the input of the forward kinematics to generate the motion variable $\mathbf{X}$ which closes the control loop. Then the error of the motion in the task space can be got to generate the actuator forces/torque.

However, as mentioned above, the forward kinematic analysis of the parallel robot is always complex and difficult. In real time control, it is time-consuming to find the

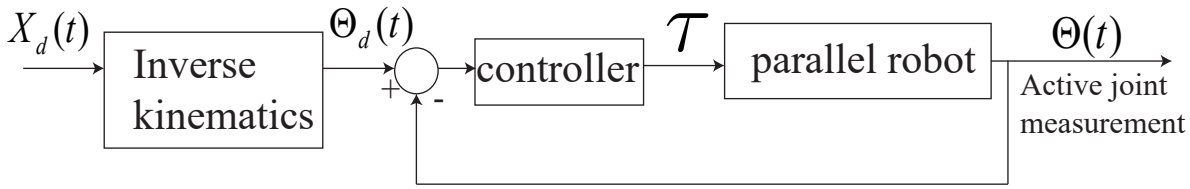

Figure 1.11: The general model-based control when joint variable $\boldsymbol{\Theta}$ is measured and the inverse kinematic analysis is used

solution of the forward kinematic. On the other hand, the inverse kinematic analysis of the parallel robot is much easier. Therefore, another control topology taking the inverse kinematics into consider is proposed. As depicted in Fig. 1.11, through the inverse kinematic analysis, the desired values for joint variable $\boldsymbol{\Theta}_d$ can be calculated from the desired motion trajectory of the platform $\mathbf{X}_d$. As a result, the joint space error $e_{\boldsymbol{\Theta}}$ is used to generate the actuator forces/torque.

Both of the two control topology are based on the fact that the joint variable $\boldsymbol{\Theta}$ can be measured by the encoder and the detailed model of the parallel robot is necessary. It is obvious that the accuracy of the model of the parallel robot will affect directly the final accuracy of the motion of the moving platform [Merlet 2006]. Despite the complex parallel robot model, various factors still affect a lot on the final accuracy of the parallel robot, such as the manufacturing error, clearance, assembling tolerance. In order to bypass the model of the robot and reduce the impacts come from the operations in reality, the direct measurement of the motion variable $\mathbf{X}$ can be set as the feedback in the closed-loop system. The control topology is shown in Fig. 1.12, the real-time position and orientation of the moving platform is measured and is compared to the desired value to generate the motion error vector $\mathbf{e}_{\mathbf{X}}$. Then this error is used to generate commands for the actuator to minimize the tracking error. Since the position and orientation of the moving platform can be measured directly, the detailed robot model is no longer necessary. The positioning error of the parallel robot comes from the measurement error of the motion variable $\mathbf{X}$.

Another problem to be considered is that the appearance of the singularity issues lead to the division of their workspace into several aspects corresponding to different assembly modes [Merlet 2006]. The pose of the end-effector is no longer controllable at the Type 2 singularity locus. [Gogu 2004] shows that the problems of workspace reachability have been overcome by designing robots without singularities. On the other hand, it usually lead to the decrease of the size of the workspace of parallel robots. In [Briot 2008], an approach was presented to pass through Type 2 singularities by planning a trajectory

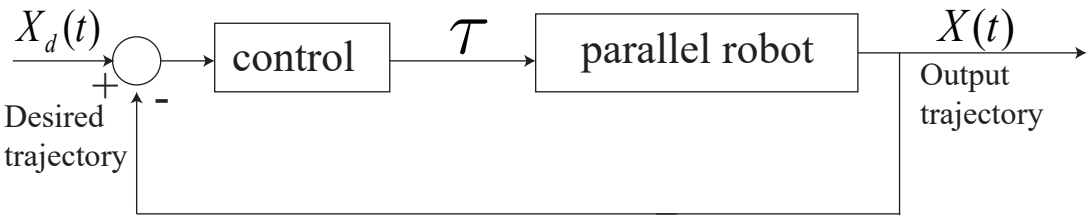

Figure 1.12: The general control topology when motion variable **X** is measured

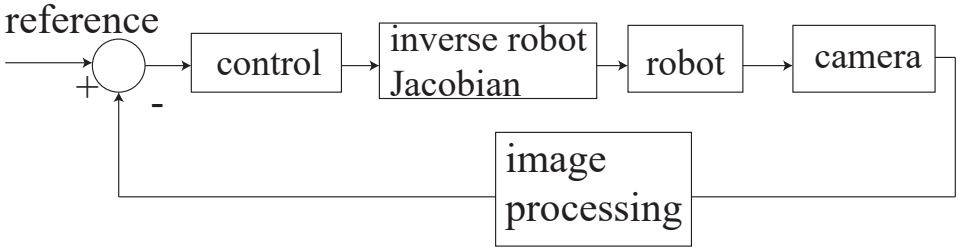

Figure 1.13: Visual servoing control scheme

respecting a physical criterion defined from the analysis of the degeneracy conditions of the dynamic model.

1.2 Visual servoing

1.2.1 Basics of visual servoing

As presented previously, the classical control methods are model based. The robot model which approximates the relationship between the joint input and the end-effector output (Fig. 1.10, Fig. 1.11). It is evident that for improving the control performance of the control scheme, we have no choice but making the model more accurate as possible. Considering the deformations, assembly errors and clearance of the parallel robot, identification is necessary, but it is always costly and time-consuming. Therefore a different control approach should be considered to overcome the complex robot model and inconsistency errors which is to use an external sensor for the control of the robot (Fig. 1.12). It has been proven in [Espiau 1992] that in terms of accuracy, the sensor-based control approaches have a better accuracy compared with the model-based controller.

Visual servoing is such an external sensor-based control method that takes the camera as the external sensor and applies the computer vision data to the servo loop to control

the robot (see Fig. 1.13)[Chaumette 2006]. The visual information makes it possible to know about the position and orientation of the objects presented in the scene and the description of the environment with a relative precision [Traslosheros 2007]. All the vision data can be acquired from a camera which is mounted directly on a robot manipulator or on a fixed base. The robot can induce the camera motion, or the camera can be fixed in the workspace so that it can observe the robot motion from a stationary configuration. The same as other control methods, the purpose of all vision-based control schemes is to minimize an error $\mathbf{e}(t)$, which is defined by [Chaumette 2006]

$$\mathbf{e}(t) = \mathbf{s}(\mathbf{m}(t), \mathbf{a}) - \mathbf{s}^* \tag{1.12}$$

where the vector $\mathbf{m}(t)$ is a set of image measurements (e.g. the image coordinates of interest points, coordinates of the centroid of the object observed, image moments, leg-direction and lines). These image measurements are used to compute the vector of image features $\mathbf{s}$, and $\mathbf{a}$ is the set of parameters that represent potential additional knowledge of the system, such as the intrinsic parameters of the camera used or the models of the objects observed. The vector $\mathbf{s}^*$ is the desired value of the image features.

When the image feature $\mathbf{s}$ is selected , the most straightforward approach is to design a velocity controller. The relationship between the time variation of $\mathbf{s}$ and the camera velocity is required. The spatial relative camera-object kinematics screw is denoted by $\boldsymbol{\tau} = (\mathbf{v}, \boldsymbol{w}) = (v_x, v_y, v_z, w_x, w_y, w_z)$, with $\mathbf{v}$ the instantaneous linear velocity and $\boldsymbol{w}$ the instantaneous angular velocity. The relationship between $\dot{\mathbf{s}}$ and $\boldsymbol{\tau}$ (in the condition of fixing the observed object) is [Tahri 2004]:

$$\dot{\mathbf{s}} = \mathbf{L}_\mathbf{s}\boldsymbol{\tau} \tag{1.13}$$

where $\mathbf{L}_\mathbf{s}$ is called the *interaction matrix* related to $\mathbf{s}$.

Combining equations (1.12) and (1.13), the relationship between the relative camera-object velocity and the time variation of the error can be obtained by

$$\dot{\mathbf{e}} = \mathbf{L}_\mathbf{e}\boldsymbol{\tau} \tag{1.14}$$

where $\mathbf{L}_\mathbf{e} = \mathbf{L}_\mathbf{s}$. We apply an exponential decoupled decrease of the error such as $\dot{\mathbf{e}} = -\lambda\mathbf{e}$, then we can get from 1.14:

$$\boldsymbol{\tau} = -\lambda\mathbf{L}_\mathbf{e}^+\mathbf{e} \tag{1.15}$$

where $\mathbf{L_e^+}$ is the Moore-Penrose pseudo-inverse of $\mathbf{L_e}$. However, in reality, it's impossible to know perfectly $\mathbf{L_e}$ and $\mathbf{L_e^+}$. As a result, an approximation or an estimation of these two matrices can be realized.

There are several servo classifications exist based on the variations in each component of a servo system, for example the location of the camera, the two types are eye-in-hand and eye-to-hand configurations [Huntchinson 1996]. Another classification is based on whether the control is applied to the joints (or DOF) directly or as a position command to a robot controller, the two types are direct servo and dynamic look-and-move [Papanikolopoulos 1993]. In terms of the way that $\mathbf{s}$ is designed, visual servoing can be divided into two approaches. One is the image-based visual servoing (IBVS) in which $\mathbf{s}$ consists of a set of features that are available in the image information. The other one is the position-based visual servoing (PBVS) in which $\mathbf{s}$ consists of a set of 3-D parameters, which must be estimated from image measurements.

Image-based visual servoing (IBVS) uses the image features obtained from the image plane to define the set $\mathbf{s}$. For the image measurements $\mathbf{m}$, they are usually the information of the pixel coordinates of the image points, such as the centroid points of the object, image moments or the intersection points of the lines and the image boundary, the parameters $\mathbf{a}$ in this condition are the camera intrinsic parameters (intrinsic parameter matrix $\mathbf{K}$ of the camera) [Feddema 1989]. Once the image features are selected, we can get the interaction matrix related to the features. With the help of the equation (1.15), we can always make the robot move to the desired position.

Position-based visual servoing (PBVS) [Thuilot 2002] uses the pose of the camera with respect to some reference coordinates frame to define $\mathbf{s}$ ($\mathbf{s} = (\mathbf{t}, \theta\mathbf{u})$), where $\mathbf{t}$ is a translation vector and $\theta\mathbf{u}$ is the angle/axis parameterization for the rotation [Chaumette 2006].

Another class of visual servoing is known as the 2.5D visual servoing, which is first presented in [Malis 1999]. The feature vector used as the input of the control law is selected as $\mathbf{s} = [x, y, z_m, \theta\mathbf{u}^T]^T$, with x and y are the normalized coordinates of an image point, $z_m = \log Z$, Z being the depth of the considered point, θ and $\mathbf{u}$ are respectively the angle and axis of rotation extracted from rotation [Malis 2000]. This 2.5D visual servoing is a method combining image-based visual servoing and position-based visual servoing techniques.

Another advanced visual servoing approach is the spherical Image-based visual servoing [Corke 2010]. For the spherical image-based visual servoing, a set of spherical projected features is selected to design the control for IBVS. The features in image plane

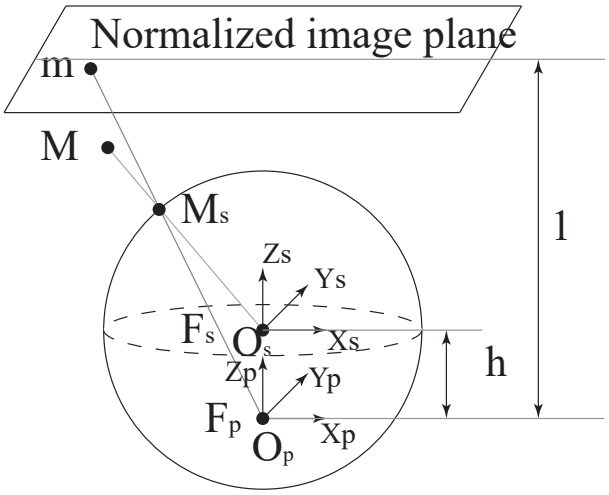

Figure 1.14: Spherical projection model

can be projected onto the surface of a virtual unitary sphere, which helps getting a set of image features with nice decoupling and linear properties [Tahri 2010] [Tahri 2013]. As we see from the figure 1.14, the camera frame is denoted as $\mathbf{F}_p$ and the sphere frame is denoted as $\mathbf{F}_s$. They are related by a simple translation of h along **Z**-axis. The line connecting point m in normalized image plane and the center of the camera frame $\mathbf{O}_p$, intersects with the surface of the unitary sphere at point $\mathbf{M}_s$, which is the spherical projected point. $(\mathbf{M}_s)_{\mathbf{F}_s} = (x_s, y_s, z_s)^T$ denotes the coordinate of M_s w.r.t the sphere frame and $(\mathbf{M}_s)_{\mathbf{F}_p} = (x_s, y_s, z_s + h)^T$ is the coordinate of M_s expressed in the camera frame. Then $(\mathbf{M}_s)_{\mathbf{F}_p}$ can be projected onto the normalized image plane [Wang 2016]:

$$\mathbf{m} = (x, y, 1)^T = (\frac{x_s}{z_s + h}, \frac{y_s}{z_s + h}, 1)^T \tag{1.16}$$

For the unitary sphere: $x_s^2 + y_s^2 + z_s^2 = 1$, when the normalized point feature is given, the spherical projected point feature can be obtained in the form of:

$$(\mathbf{M}_s)_{\mathbf{F}_s} = (x_s, y_s, z_s)^T = \gamma(x, y, 1 - \frac{h}{\gamma})^T \tag{1.17}$$

where $\gamma = \frac{h + \sqrt{1 + (1 - h^2)(x^2 + y^2}}{x^2 + y^2 + 1}$.

With the help of the feature Jacobian $\mathbf{L}_s$ related to the visual servoing features (spherical point features) $\mathbf{s}$, we can get the equation (1.13) for the visual servoing (see more details in [Corke 2010] [Wang 2016]).

All above we present are the classical first-order modeling approaches which give the relationship between the first derivative of the features and the velocity of the end-effector for the Eye-to-hand and the first-order derivative of the features and the velocity of the sensor for the Eye-in-hand [Flandin 2000]. Another class of model which considers the relationship between the second-order time derivative of the features and the acceleration of sensors, in addition to their velocity was proposed in [Fusco 2019]. The derivative of (1.13) gives:

$$\ddot{\mathbf{s}} = \mathbf{L_s}\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\tau} + \dot{\mathbf{L}}_\mathbf{s}\boldsymbol{\tau} \tag{1.18}$$

Assuming an eye-in-hand setup, the acceleration of the sensor can be got by

$$\mathbf{a} = \mathbf{T}_v\mathbf{J}\ddot{\boldsymbol{\Theta}} + \mathbf{T}_v\dot{\mathbf{J}}\dot{\boldsymbol{\Theta}} \tag{1.19}$$

Where $\mathbf{J}$ is the Jacobian matrix of the robot, $\boldsymbol{\Theta}$ and $\dot{\boldsymbol{\Theta}}$ denotes the positions and velocities of the joints, $\mathbf{T}_v$ is a twist-rotation matrix that projects the kinematic screw on the sensor frame.

Then inject the equation (1.19) into the second order model, the kinematic relation is

$$\ddot{\mathbf{s}} = \mathbf{J_s}\ddot{\boldsymbol{\Theta}} + \mathbf{L_s}\mathbf{T}_v\dot{\mathbf{J}}\dot{\boldsymbol{\Theta}} + \mathbf{h_s} = \mathbf{J_s}\ddot{\boldsymbol{\Theta}} + \mathbf{h_q} \tag{1.20}$$

where $\mathbf{J_s} = \mathbf{L_s}\mathbf{T}_v\mathbf{J}$ is the feature Jacobian. The objective of the visual servoing is to regulate the feature error $\mathbf{e}_s = \mathbf{s} - \mathbf{s}^*$ to zero. When the second order model is applied (see more details in [Fusco 2019]:

$$\ddot{\boldsymbol{\Theta}} = \widehat{\mathbf{J}_\mathbf{s}^+}(\ddot{\mathbf{s}} - k_d\dot{\mathbf{e}}_\mathbf{s} - k_p\mathbf{e_s} - \widehat{\mathbf{h}}_\mathbf{q}) \tag{1.21}$$

If the positive gains are selected, the exponential decrease of the error according to the evolution of the autonomous linear system $\ddot{\mathbf{e}}_\mathbf{s} - k_d\dot{\mathbf{e}}_\mathbf{s} - k_p\mathbf{e_s} = 0$.

In [Keshmiri 2014], an augmented version of IBVS (AIBVS) for a six DOFs robot is presented. It produced acceleration as the controlling command. For a point $\mathbf{P} = (X, Y, Z)$, the relationship between the camera motion and the features are [Kraige 2020]

$$\ddot{\mathbf{P}} = \mathbf{L}_a\mathbf{A}_a + \mathbf{L}_v \tag{1.22}$$

where $\mathbf{A}_a$ is the camera acceleration screw and $\mathbf{L}_a$ is the interaction matrix in the con-

ventional IBVS

$$\mathbf{L}_a = \begin{bmatrix} -\lambda/Z & 0 & x/Z & xy/\lambda & -(1+x^2)/\lambda & y \\ 0 & -\lambda/Z & y/Z & 1+y^2/\lambda & -xy/\lambda & -x \end{bmatrix} \tag{1.23}$$

λ being a positive constant. $\mathbf{L}_v$ can be written as

$$\mathbf{L}_v = \begin{bmatrix} \mathbf{V}^T \Omega_x \mathbf{V} \\ \mathbf{V}^T \Omega_y \mathbf{V} \end{bmatrix} \tag{1.24}$$

where $\mathbf{V}$ is the camera velocity screw, and the definition of Ω_x and Ω_y can be found in [Keshmiri 2014].

Based on the equation (1.22), they applied the acceleration of the camera to the PD controller to get the augmented visual servoing controller. [Keshmiri 2014] performed extensive experiments and validated the efficiency of the controller. The experiment results show the advantages of the proposed AIBVS over the classic IBVS in terms of smoother motion in the image space and 3-D space.

A camera with a large field of view is useful in the visual servoing. The last class of cameras with a wide field-of-view are the dioptric systems (fish-eye cameras) [Baker 1998]. A fish-eye camera is composed of a fish-eye lens and a conventional camera. The fish-eye camera models are detailed here: $\mathcal{F}_c$ and $\mathcal{F}_i$ denote the frames attached to the camera with origin C located at the projection center and to the image plane with origin I located at the pricipal point respectively. A 3D point $\mathbf{N}$ with the coordinates $\mathbf{N} = (x, y, z)^T$ with respect to the frame $\mathcal{F}_c$. $\mathbf{n}_p = (x_p \ y_p)^T$ and $\mathbf{n}_f = (x_f \ y_f)^T$ are the perspective image and the fish-eye image coordinates w.r.t. $\mathcal{F}_i$ of the point $\mathbf{N}$. r is the distance between the principal axis and the image point and the angle between the incoming ray and the principal axis is θ. The angle between the X-axis and $\mathbf{n}$ by α (Fig. 1.15).

The way of getting the fish-eye image is detailed presented in [Courbon 2007] [Kannala 2004] and [Kannala 2006]. A novel projection model based on the unified projection model is presented in [Khomutenko 2015]. This model applies to catadioptric systems and wide-angle fish-eye cameras, it takes just two projection parameters more than a simple pinhole model to represent radial distrotion. The application of fish-eye cameras helps getting a large image filed without the dead area in the center of the image, which is the huge drawback of the perspective camera.

[Corke 2001] developed a partitioned visual servoing approach. This method aims finding six features such that each is related to only one degree of freedom (then the

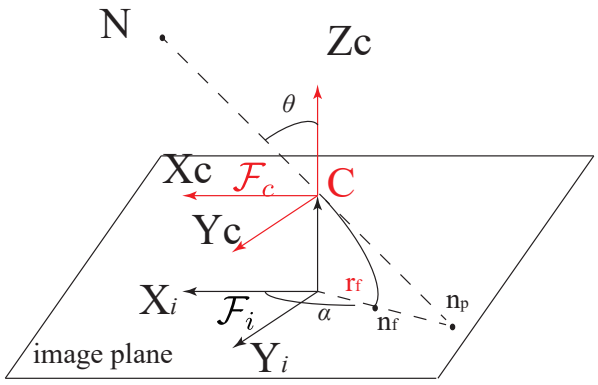

Figure 1.15: Perspective and fish-eye imaging process. The perspective image of a 3D point **M** is $\mathbf{m}_p$ and its fish-eye image is **f** [Courbon 2007].

interaction matrix is a diagonal matrix). The grail is to find a diagonal interaction matrix whose elements are constant, as near as possible to the identity matrix, leading to a pure, direct and simple linear control problem.

In [Bakthavatchalam 2013] and [Bakthavatchalam 2018], photometric image moments which link the image moments and image intensity function were introduced as new image features for IBVS. The interaction matrix has been developed in closed form for the proposed photometric moments. Photometric moments allow us to avoid any spatial segmentation steps and reduce the image processing to a simple and systematic moments computation on all the image plane, while simultaneously leveraging the excellent decoupling properties of binary image moments.

Olivier Kermorgant and François chaumette proposed a method of combining IBVS and PBVS to ensure the visibility constraint in [Kermorgant 2011]. This method consider PBVS as the core of control scheme and IBVS is only applied when the observed object is going to leave the view of the camera. [Cazy 2015] dealt with the possible occlusions or the lose of tracking image features because of the limited camera view by applying the prediction/correction scheme.

[Zake 2019] and [Zake 2019] presented that the PBVS is applied to the control of cable-driven parallel robots (CDPRs) which has a good accuracy and stability (Fig. 1.16). [Dallej 2019] introduced a vision-based control strategy for large-dimension CDPRs displacing heavy payloads in quasi-static operation and it allows the mobile platform to be accurately positioned without having to estimate the cable elastic characteristics. In [Zake 2020], the trajectory planning and tracking with 2.5D visual servoing is considered to control the CDPRs. This control method increases the system robustness and ensures

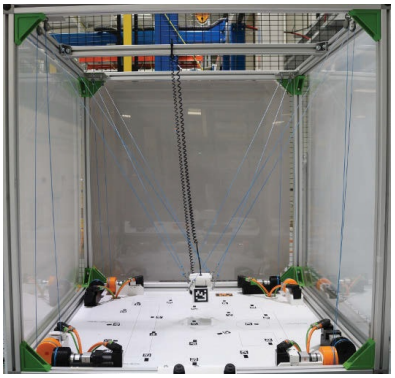

Figure 1.16: Acrobot of the Cable-Driven parallel robot [Zake 2019]

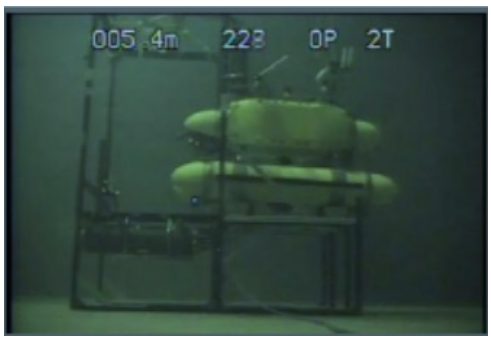

Figure 1.17: Hovering AUV [Maki 2013]

the straight-line motion both in image and the camera in the base frame. PBVS had also been proposed for the docking of Autonomous Underwater Vehicle [Yahya 2016] (Fig. 1.17). Since when considering IBVS, the image features are highly non-linear function of camera pose.

[Fusco 2020] proposed a new prediction models of Visual Predictive Control which leads to both better motions in the feature space and shorter sensor trajectories in 3D. This controller is proposed to integrate acceleration information provided by second-order models, which allows to better estimate the evolution of the image features. In [Jabbari 2014], an adaptive IBVS scheme has been developed for controlling the translational motion and the yaw angle of a quadrotor. The control is designed based on the dynamics of image features in the virtual image plane which makes use of appropriately defined perspective moments. [Marchand 2017] studied the condition of accessing the image features through mirror reflection in visual servoing and the experiments has been achieved with a Gantry robot to show the validity of this approach.

Different from traditional image features, a new feature "trifocal geometry" was taken into account in the design of the visual servoing, which has been proven that trifocal tensor has great potential in addressing visual servoing [Andreff 2016]. The trifocal geometry includes all projective geometric relationships among three-view of the the three cameras. The tensor-based visual servoing serve robot to achieve several tasks, including path following [Sabatta 2013] and trajectory tracking [Chen 2016]. In [Zhang 2018], an 6 DOF trifocal tensor-based visual servoing is presented. The geometric relationship among the current, desired, and initial views is selected to construct the visual feedback and it can avoid explicit camera pose decomposition.

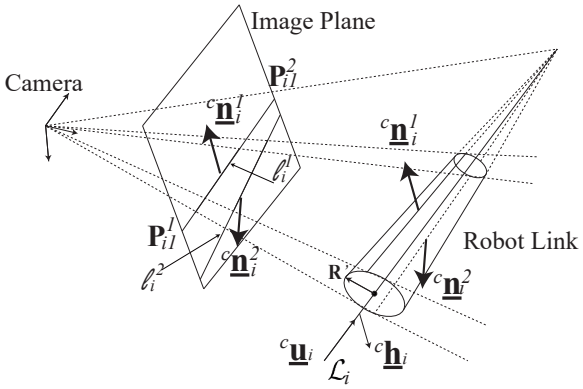

Figure 1.18: Projection of a cylinder in the image

1.2.2 Recalls on Leg-direction based visual servoing

Since visual servoing takes image features to do the control of robot, the most common approach is to observe the end-effector pose directly [Espiau 1992]. However, sometimes it is difficult to observe the end-effector pose directly. For example, in the case of a machine-tool. As a result, some other alternative image features are applied on the visual servoing control. The legs of parallel robots are usually designed with slim and cylindrical rods, then a feature that can be extracted from the observation of the legs is their direction [Andreff 2005]. Robot leg directions can be expressed by a unit vector ${}^c\underline{\mathbf{u}}_i$. In what follows, the superscript "c" denotes the camera frame (equivalent to the world frame in our paper).

Leg observation is necessary for doing the leg-direction based visual servoing. Parallel robots legs have cylindrical cross-sections [Merlet 2006]. The edges of the i-th cylindrical legs are given, in the camera frame, by [Andreff 2007] (Figs. 1.18 and 1.19)

$$ {}^c\underline{\mathbf{n}}_i^1 = -\cos\theta_i \ {}^c\underline{\mathbf{h}}_i - \sin\theta_i {}^c\underline{\mathbf{u}}_i \times^c \underline{\mathbf{h}}_i \tag{1.25} $$

$$ {}^c\underline{\mathbf{n}}_i^2 = +\cos\theta_i \ {}^c\underline{\mathbf{h}}_i - \sin\theta_i {}^c\underline{\mathbf{u}}_i \times^c \underline{\mathbf{h}}_i \tag{1.26} $$

where $\cos\theta_i = \sqrt{{}^ch_i^2 - R_i^2}/{}^ch_i$, $\sin\theta_i = R_i/{}^ch_i$ and $({}^c\underline{\mathbf{u}}_i, {}^c\underline{\mathbf{h}}_i, {}^ch_i)$ [1]are the Binormalized Plücker coordinates of the cylinder axis and R_i is the radius of the cylinder [Andreff 2002].

The leg orientation can also be got, in the camera frame, from the edges of the cylin-

[1]In the paper, $\underline{\mathbf{h}}$ stands for unit vector, while $\mathbf{h}$ stands for non unit vector

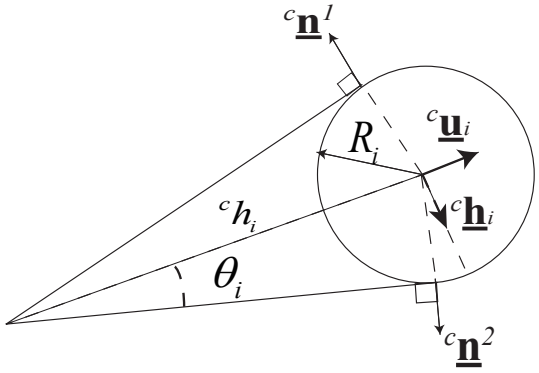

Figure 1.19: Visual edges of a cylinder

drical legs [Andreff 2007]

$$^c\underline{\mathbf{u}}_i = \frac{^c\underline{\mathbf{n}}_i^1 \times^c \underline{\mathbf{n}}_i^2}{\|^c\underline{\mathbf{n}}_i^1 \times^c \underline{\mathbf{n}}_i^2\|} \tag{1.27}$$

The vector $^c\mathbf{h}_i = {}^ch_i^c\underline{\mathbf{h}}_i$ can be computed by using the edges of the i-th cylindrical leg too, which is given by

$$^c\mathbf{h}_i = {}^c\mathbf{D}_i \times {}^c\underline{\mathbf{u}}_i \tag{1.28}$$

where $^c\mathbf{D}_i$ is the position of the point B_i in the camera frame, which is the closest point of the axis of the i-th leg to the camera. It is given by [Andreff 2002]

$$^c\mathbf{D}_i = \frac{R_i}{\sin\theta_i} \cdot \frac{^c\underline{\mathbf{n}}_i^1 + {}^c\underline{\mathbf{n}}_i^2}{\|^c\underline{\mathbf{n}}_i^1 + {}^c\underline{\mathbf{n}}_i^2\|} \tag{1.29}$$

In leg-direction-based visual servoing, as it is shown in [Andreff 2007], by the help of a matrix named "interaction matrix", we can always find the relationship between the twist of robot end-effector $^c\boldsymbol{\tau}_c$ and its leg direction velocity by

$$^c\dot{\underline{\mathbf{u}}}_i = \mathbf{M}_{ui}^T \, {}^c\boldsymbol{\tau}_c \tag{1.30}$$

Where $\mathbf{M}_{ui}^T$ is the interaction matrix for the leg i and this interaction matrix depends on the robot geometry. We also have from [Andreff 2002]

$$^c\dot{\underline{\mathbf{u}}}_i = \mathbf{J}_{ui} \begin{bmatrix} ^c\dot{\underline{\mathbf{n}}}_i^1 \\ ^c\dot{\underline{\mathbf{n}}}_i^2 \end{bmatrix} \tag{1.31}$$

Where $({}^c\underline{\mathbf{n}}_i^1, {}^c\underline{\mathbf{n}}_i^2)$ are the vectors normal to the interpretation planes (passing through the edges of the robot link and the camera center (Fig. 1.18) and $\mathbf{J}_{ui}$ transforms the derivative with respect to time of $({}^c\underline{\mathbf{n}}_i^1, {}^c\underline{\mathbf{n}}_i^2)$ into the leg orientation velocities. As matrix $\mathbf{M}_{ui}^T$ is rank-deficient [Andreff 2005], a set of m links should be observed so that we can obtain the end-effector twist ${}^c\boldsymbol{\tau}_c$ as a function of the vectors $({}^c\dot{\underline{\mathbf{n}}}_i^1, {}^c\dot{\underline{\mathbf{n}}}_i^2)$:

$$\mathbf{M}_u^T \, {}^c\boldsymbol{\tau}_c = \mathbf{J}_u \, {}^c\dot{\underline{\mathbf{n}}} \tag{1.32}$$

Where ${}^c\dot{\underline{\mathbf{n}}} = \left[{}^c\dot{\underline{\mathbf{n}}}_1^{1\,T}, {}^c\dot{\underline{\mathbf{n}}}_1^{2\,T}, \cdots, {}^c\dot{\underline{\mathbf{n}}}_m^{1\;T}, {}^c\dot{\underline{\mathbf{n}}}_m^{2\;T}\right]^T$, $\mathbf{J}_u$ is a block-diagonal matrix containing the matrix $\mathbf{J}_{ui}$ and we can get the interaction matrix $\mathbf{M}_u^T$ by stacking the matrices $\mathbf{M}_{ui}^T$ of m legs. Matrix $\mathbf{M}_u^T$ will be of full rank if enough legs are observed.

Then, by using the pseudo-inverse $\mathbf{M}_u^{T+}$ of the matrix $\mathbf{M}_u^T$, we have:

$${}^c\boldsymbol{\tau}_c = \mathbf{M}_u^{T+}\mathbf{J}_u \, {}^c\dot{\underline{\mathbf{n}}} \tag{1.33}$$

For the i-th leg of the robot, its projection on image plane are two lines l_i^1 and l_i^2 (Fig. 1.19). The coordinates of the intersection points $\mathbf{P}_{i1}^1$, $\mathbf{P}_{i1}^2$ (Fig. 1.18) between the image boundary and the tracked edges of the robot in pixel can be got from the image plane. By applying equations (1.40) and (1.42), we get $({}^c\underline{\mathbf{n}}_i^1, {}^c\underline{\mathbf{n}}_i^2)$. With the help of equation (1.27), the leg direction vector ${}^c\underline{\mathbf{u}}_i$ can be got.

Based on the kinematic relationship, one classical leg-direction-based controller which takes the leg direction vector ${}^c\underline{\mathbf{u}}_i$ as the feedback can be used:

$${}^c\boldsymbol{\tau}_c = -\lambda \mathbf{M}_u^{T+}\mathbf{e} \tag{1.34}$$

In which vector $\mathbf{e}$ stacks the errors $\mathbf{e}_i$ between the desired vector ${}^c\underline{\mathbf{u}}_i$ (denoted as ${}^c\underline{\mathbf{u}}_{di}$) and the current one ($\mathbf{e}_i$ is defined as $\mathbf{e}_i = {}^c\underline{\mathbf{u}}_i - {}^c\underline{\mathbf{u}}_{di}$, $i = 1 \cdots m$), and λ is a positive constant.

This expression can be transformed into a controller for the joint velocities:

$$\dot{\mathbf{q}} = -\lambda \mathbf{J}_{pinv}\mathbf{M}_u^{T+}\mathbf{e} \tag{1.35}$$

Where $\mathbf{J}_{pinv}$ is the inverse Jacobian matrix of the robot linking the end-effector twist to the actuator velocities, i.e. $\mathbf{J}_{pinv} \, {}^c\boldsymbol{\tau}_c = \dot{\mathbf{q}}$.

Another leg-direction-based controller which takes the vector $({}^c\underline{\mathbf{n}}_i^1, {}^c\underline{\mathbf{n}}_i^2)$ as the feedback

to do the visual servoing is defined by [Andreff 2007]:

$$^{c}\boldsymbol{\tau}_{c} = -\lambda \mathbf{M}_{u}^{T+} \mathbf{J}_{u}\, ^{c}\mathbf{e} \tag{1.36}$$

In which vector $\mathbf{e}$ stacks the errors $\mathbf{e}_{i,j}$ between the desired vector $^{c}\underline{\mathbf{n}}_{i}^{j}$ (denoted as $^{c}\underline{\mathbf{n}}_{di}^{j}$) and the current one ($\mathbf{e}_{i,j}$ is defined as $\mathbf{e}_{i,j} = {}^{c}\underline{\mathbf{n}}_{i}^{j} - {}^{c}\underline{\mathbf{n}}_{di}^{j}$, $j = 1, 2, i = 1 \cdots m$).

This expression can be transformed into a controller for the joint velocities:

$$\dot{\mathbf{q}} = -\lambda \mathbf{J}_{pinv} \mathbf{M}_{u}^{T+} \mathbf{J}_{u}\, ^{c}\mathbf{e} \tag{1.37}$$

Where $\mathbf{J}_{pinv}$ is the inverse Jacobian matrix of the robot linking the end-effector twist to the actuator velocities, i.e. $\mathbf{J}_{pinv}\ ^{c}\boldsymbol{\tau}_{c} = \dot{\mathbf{q}}$.

Leg-direction-based visual servoing has been proven to be successfully applied on the control of a PAR4 robot in [Dallej 2006], a GS platform in [Andreff 2007], a IR4 parallel robot in [Dallej 2007] and a Five-bar mechanism in [Kaci 2018].

1.2.3 Recalls on Line-based visual servoing

Another feature that can be extracted from the observation of the robot legs are the lines passing through the leg axes [Andreff 2002]. In the condition of using line-based visual servoing, the line $\mathcal{L}_i$ passing through robot link cylinder axis i is expressed by its Plücker coordinates ($^{c}\underline{\mathbf{u}}_{i}$,$^{c}\,\mathbf{h}_{i}$) (see Fig. 1.18). The vector $^{c}\underline{\mathbf{u}}_{i}$ is the unit vector giving the spatial orientation of the line, $^{c}\mathbf{h}_{i}$ is the vector perpendicular to the so-called interpretation plane of the line $\mathcal{L}_i$ (which is the plane passing through the camera frame origin and the line $\mathcal{L}_i$). The definition of the vector $^{c}\mathbf{h}_{i}$ is $^{c}\mathbf{h}_{i} = {}^{c}\mathbf{P} \times {}^{c}\underline{\mathbf{u}}_{i}$, where $^{c}\mathbf{P}$ is the position of any point P on the line, expressed in the camera frame [Merlet 2006].

Projecting such a line in the image plane, we have the characteristic equation, expressed in the camera frame [Andreff 2002]

$$^{c}\mathbf{h}_{i}^{T}\ ^{c}\mathbf{p} = 0 \tag{1.38}$$

where $^{c}\mathbf{p}$ are the coordinates of a point P in the image plane, lying on the line (expressed in the camera frame).

The matrix $\mathbf{K}$ is formed by the intrinsic parameters of the camera used in visual

servoing

$$\mathbf{K} = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.39}$$

where f_u, f_v are the ratio between the focal length and the size of a pixel and u_0, v_0 are the offset between the camera frame and the pixel frame.

With the help of the matrix $\mathbf{K}$, we can get the line equation in pixel coordinates ${}^{p}\mathbf{h}_i$ from:

$${}^{p}\mathbf{h}_i^T \ {}^{p}\mathbf{p} = 0 \tag{1.40}$$

Replace ${}^{p}\mathbf{p}$ with $\mathbf{K}{}^{c}\mathbf{p}$ in the equation and we have

$${}^{p}\mathbf{h}_i^T \ \mathbf{K}{}^{c}\mathbf{p} = 0 \tag{1.41}$$

By identification of (1.38) and (1.40), we have

$${}^{p}\underline{\mathbf{h}}_i = \frac{\mathbf{K}^{-T} \ {}^{c}\underline{\mathbf{h}}_i}{\|\mathbf{K}^{-T} \ {}^{c}\underline{\mathbf{h}}_i\|}, \ {}^{c}\underline{\mathbf{h}}_i = \frac{\mathbf{K}^{T} \ {}^{p}\underline{\mathbf{h}}_i}{\|\mathbf{K}^{T} \ {}^{p}\underline{\mathbf{h}}_i\|} \tag{1.42}$$

Similar as it was done in [Andreff 2007], the twist of end-effector ${}^{c}\boldsymbol{\tau}_c$ and the time derivation of line's Plücker coordinates can be linked by

$$\begin{bmatrix} {}^{c}\dot{\underline{\mathbf{u}}}_i^T \\ {}^{c}\dot{\mathbf{h}}_i^T \end{bmatrix} = \mathbf{M}_{uhi}^T \, {}^{c}\boldsymbol{\tau}_c \tag{1.43}$$

$\mathbf{M}_{uhi}^T$ is the interaction matrix for the leg i for this type of observation and it depends on the robot geometry. In [Andreff 2007], the vector $({}^{c}\underline{\mathbf{u}}_i, {}^{c}\mathbf{h}_i)$ can be calculated from the knowledge of $({}^{c}\underline{\mathbf{n}}_i^1, {}^{c}\underline{\mathbf{n}}_i^2)$, then we have:

$${}^{c}\dot{\underline{\mathbf{u}}}_i = \mathbf{J}_{ui} \begin{bmatrix} {}^{c}\dot{\underline{\mathbf{n}}}_i^1 \\ {}^{c}\dot{\underline{\mathbf{n}}}_i^2 \end{bmatrix}, \ {}^{c}\dot{\mathbf{h}}_i = \mathbf{J}_{hi} \begin{bmatrix} {}^{c}\dot{\underline{\mathbf{n}}}_i^1 \\ {}^{c}\dot{\underline{\mathbf{n}}}_i^2 \end{bmatrix} \tag{1.44}$$

Where $\mathbf{J}_{ui}$ and $\mathbf{J}_{hi}$ transform the time derivative of $({}^{c}\underline{\mathbf{n}}_i^1, {}^{c}\underline{\mathbf{n}}_i^2)$ into the vector velocities of $({}^{c}\underline{\mathbf{u}}_i, {}^{c}\mathbf{h}_i)$ (see [Vignolo 2014] for more details). Since the matrix $\mathbf{M}_{uhi}^T$ is rank-deficient [Andreff 2005], a set of m links are observed to get the end-effector twist ${}^{c}\boldsymbol{\tau}_c$ as a function

of the vectors $({}^c\dot{\underline{\mathbf{n}}}_i^1, {}^c\dot{\underline{\mathbf{n}}}_i^2)$:

$$\mathbf{M}_{uh}^T \ {}^c\boldsymbol{\tau}_c = \begin{bmatrix} \mathbf{J}_u \\ \mathbf{J}_h \end{bmatrix} \ {}^c\dot{\underline{\mathbf{n}}} \tag{1.45}$$

where ${}^c\dot{\underline{\mathbf{n}}} = \left[{}^c\dot{\underline{\mathbf{n}}}_1^{1\,T}, {}^c\dot{\underline{\mathbf{n}}}_1^{2\,T}, \cdots, {}^c\dot{\underline{\mathbf{n}}}_m^{1\ T}, {}^c\dot{\underline{\mathbf{n}}}_m^{2\ T} \right]^T$, $\mathbf{J}_u$, $\mathbf{J}_h$ are block-diagonal matrices containing the matrices $\mathbf{J}_{ui}$, $\mathbf{J}_{hi}$. The interaction matrix $\mathbf{M}_{uh}^T$ is got by stacking the matrix $\mathbf{M}_{uhi}^T$ and it is of full rank if enough legs are observed.

Then, by using the pseudo-inverse $\mathbf{M}_{uh}^{T+}$ of the matrix $\mathbf{M}_{uh}^T$, we have:

$${}^c\boldsymbol{\tau}_c = \mathbf{M}_{uh}^{T+} \begin{bmatrix} \mathbf{J}_u \\ \mathbf{J}_h \end{bmatrix} {}^c\dot{\underline{\mathbf{n}}} \tag{1.46}$$

Similarly as it was done in Section 1.2.2, we can get the vector $({}^c\underline{\mathbf{n}}_i^1, {}^c\underline{\mathbf{n}}_i^2)$ from the coordinates of the intersection points between the image boundary and the tracked edges of the robot in pixel. Then with the help of equations (1.27) and (1.28), we have the Plücker coordinates $({}^c\underline{\mathbf{u}}_i, {}^c\mathbf{h}_i)$ (see Fig. 1.18).

Based on the kinematic relationship, one classical controller which takes the Plücker coordinates $({}^c\underline{\mathbf{u}}_i, {}^c\mathbf{h}_i)$ as the feedback can be developed [Vignolo 2014]:

$${}^c\boldsymbol{\tau}_c = -\lambda \mathbf{M}_{uh}^{T+} \mathbf{e} \tag{1.47}$$

In which vector $\mathbf{e}$ stacks the errors $\mathbf{e}_i$ between the desired value of ${}^c\mathbf{l}_i = [{}^c\underline{\mathbf{u}}_i, {}^c\mathbf{h}_i]^T$ (denoted as ${}^c\mathbf{l}_{di}$) and the current one ($\mathbf{e}_i$ is defined as $\mathbf{e}_i = {}^c\mathbf{l}_i - {}^c\mathbf{l}_{di}$, $i = 1 \cdots m$), and λ is a positive constant.

This expression can be transformed into a controller for the joint velocities:

$$\dot{\mathbf{q}} = -\lambda \mathbf{J}_{pinv} \mathbf{M}_{uh}^{T+} \mathbf{e} \tag{1.48}$$

Where $\mathbf{J}_{pinv}$ is the inverse Jacobian matrix of the robot linking the end-effector twist to the actuator velocities, i.e. $\mathbf{J}_{pinv} \ {}^c\boldsymbol{\tau}_c = \dot{\mathbf{q}}$.

Another line-based controller which takes the vector $({}^c\underline{\mathbf{n}}_i^1, {}^c\underline{\mathbf{n}}_i^2)$ as the feedback to do the visual servoing is defined as:

$${}^c\boldsymbol{\tau}_c = -\lambda \mathbf{M}_{uh}^{T+} \begin{bmatrix} \mathbf{J}_u \\ \mathbf{J}_h \end{bmatrix} \mathbf{e} \tag{1.49}$$

In which vector $\mathbf{e}$ stacks the errors $\mathbf{e}_{i,j}$ between the desired vector ${}^c\underline{\mathbf{n}}_i^j$ (denoted as ${}^c\underline{\mathbf{n}}_{di}^j$) and the current one ($\mathbf{e}_{i,j}$ is defined as $\mathbf{e}_{i,j} = {}^c\underline{\mathbf{n}}_i^j - {}^c\underline{\mathbf{n}}_{di}^j$, $j = 1, 2, i = 1 \cdots m$), and λ is a positive constant.

This expression can be transformed into a controller for the joint velocities:

$$\dot{\mathbf{q}} = -\lambda \mathbf{J}_{pinv} \mathbf{M}_{uh}^{T+} \begin{bmatrix} \mathbf{J}_u \\ \mathbf{J}_h \end{bmatrix} \mathbf{e} \tag{1.50}$$

Where $\mathbf{J}_{pinv}$ is the inverse Jacobian matrix of the robot linking the end-effector twist to the actuator velocities, i.e. $\mathbf{J}_{pinv}\ {}^c\boldsymbol{\tau}_c = \dot{\mathbf{q}}$.

1.2.4 Recalls on Image moment visual servoing

With the development of computer vision applied on the control of the robot, image moments have been widely used in visual servoing for a very long time, especially for pattern-recognition applications [Mukundan 1998]. Compared to the other features that are used in visual servoing, image moments can be computed from several kinds of objects defined from closed contours (continuous dense object) or a set of points (discrete object) so that they provide a more generic representation of any object.

It is well known that the moments m_{ij} of order $i+j$ related to a set of N image points are defined by:

$$m_{ij} = \sum_{k=1}^{N} x_k^i y_k^j \tag{1.51}$$

Where x_k and y_k are the coordinates of the k-th point belonging to the projection on the image plan of the object observed.

For an object $\mathcal{O}$ defined from closed contours (continuous dense object), we denote that the object $\mathcal{O}$ and $\pi(t)$ is the image acquired by the camera which has a closed contour at time t. We denote $\mathcal{R}(t)$ the part of $\pi(t)$ where the object projects. In that case, the moments m_{ij} of order $i+j$ of the object $\mathcal{O}$ in the image are defined by [Chaumette 2004]

$$m_{ij} = \iint_{\mathcal{R}} x^i y^j dx dy \tag{1.52}$$

Where x and y are the coordinates of any point belonging to the surface $\mathcal{R}$. As shown in [Chaumette 2004], we can always find the analytical form relating the time derivative $\dot{m}_{ij}$ of the moment m_{ij} and the relative kinematic screw $\boldsymbol{\tau} = (\mathbf{v}, \mathbf{w}) = (v_x, v_y, v_z, w_x, w_y, w_z)$

between the camera and the object. The relationship can be expressed in the form [Espiau 1992]

$$\dot{m}_{ij} = \mathbf{L}_{m_{ij}} \boldsymbol{\tau} \tag{1.53}$$

$\mathbf{L}_{m_{ij}}$is the interaction matrix related to moment m_{ij}[Espiau 1992].

By applying the famous Green's theorem, we can have

$$\dot{m}_{ij} = \iint_{\mathcal{R}} \mathbf{div}[f(x,y)\dot{\mathbf{x}})]dxdy \tag{1.54}$$

where $f(x,y) = x^i y^j$, $\dot{\mathbf{x}}$ is the velocity of contour point $\mathbf{x} = (x,y)$. By developing (1.54), we obtain

$$\dot{m}_{ij} = \iint_{\mathcal{R}} \left[\frac{\partial f}{\partial x}\dot{x} + \frac{\partial f}{\partial y}\dot{y} + f(x,y)\left(\frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y}\right)\right] dxdy \tag{1.55}$$

In (1.55), the terms $\dot{x}$, $\dot{y}$, $(\partial\dot{x}/\partial x)$, and $(\partial\dot{y}/\partial y)$ can be linearly expressed to the kinematic screw $\boldsymbol{\tau}$. For any point whose coordinates $\mathbf{x} = (x,y)$ in image plane whose corresponding three-dimensional point has depth Z, we have

$$\dot{\mathbf{x}} = \mathbf{L}_{\mathbf{x}} \boldsymbol{\tau} \tag{1.56}$$

where

$$\mathbf{L}_{\mathbf{x}} = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -1-x^2 & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix} \tag{1.57}$$

In this case, if the object is planar or has a planar limb surface, we can relate linearly the inverse of the depth of three-dimensional point to its image coordinates ($x = X/Z$, $y = Y/Z$) and the plane equation expressed in the camera frame is given by

$$\frac{1}{Z} = A_p x + B_p y + C_p \tag{1.58}$$

Using the equation (1.58) in (1.56) and (1.57), we have (see more details in [Tahri 2004])

$$\mathbf{L}_{m_{ij}} = [m_{vx}\ m_{vy}\ m_{vz}\ m_{wx}\ m_{wy}\ m_{wz}] \tag{1.59}$$

where

$$\begin{cases} m_{vx} = -i(A_p m_{ij} + B_p m_{i-1,j+1} + C_p m_{i-1,j}) - \delta A m_{ij} \\ m_{vy} = -j(A_p m_{i+1,j-1} + B_p m_{ij} + C_p m_{i,j-1}) - \delta B_p m_{ij} \\ m_{vz} = (i+j+3\delta)(A_p m_{i+1,j} + B_p m_{i,j+1} + C_p m_{ij}) - \delta C_p m_{ij} \\ m_{wx} = (i+j+3\delta))m_{i,j+1} + j m_{i,j-1} \\ m_{wy} = -(i+j+3\delta)m_{i+1,j} - i m_{i-1,j} \\ m_{wz} = i m_{i-1,j+1} - j m_{i+1,j-1} \end{cases} \tag{1.60}$$

where $\delta = 1$, for the continuous dense object, and $\delta = 0$ for the discrete object. Consequently, the time variation of a moment of order $i+j$ can be expressed from the moments of order up to $i+j+1$ [Tahri 2005].

We consider the most simple case $i = j = 0$, the image moment $a = m_{00}$ is the area of the object observed for the continuous dense case and the number of the points for the case of discrete points. We can obtain the interaction matrix related to the m_{00} from the equation (1.60) that

$$\mathbf{L}_a = [-aA_p \ -aB_p \ a(3/Z_g - C_p) \ 3ay_g \ -3ax_g \ 0] \tag{1.61}$$

where $1/Z_g = A_p x_g + B_p y_g + C_p$, and x_g, y_g are the coordinates of the center of gravity of the observed object in the image. It is obvious that $x_g (= m_{10}/m_{00})$ and $y_g (= m_{01}/m_{00})$. It is easy to get

$$\begin{aligned} \mathbf{L}_{x_g} &= \begin{bmatrix} -1/Z_g & 0 & x_{g_{vz}} & x_{g_{wx}} & x_{g_{wy}} & y_g \end{bmatrix} \\ \mathbf{L}_{y_g} &= \begin{bmatrix} 0 & -1/Z_g & y_{g_{vz}} & y_{g_{wx}} & y_{g_{wy}} & -x_g \end{bmatrix} \end{aligned} \tag{1.62}$$

where

$$\begin{cases} x_{g_{vz}} = x_g/Z_g + A_p \epsilon n_{20} + B_p \epsilon n_{11} \\ y_{g_{vz}} = y_g/Z_g + A_p \epsilon n_{11} + B_p \epsilon n_{02} \\ x_{g_{wx}} = -y_{g_{wy}} = x_g y_g + \epsilon n_{11} \\ x_{g_{wy}} = -(1 + x_g^2 + \epsilon n_{20}) \\ y_{g_{wx}} = 1 + y_g^2 + \epsilon n_{02} \end{cases} \tag{1.63}$$

where $n_{ij} = \mu_{ij}/m_{00}$, $\epsilon = 4$ for dense objects and $\epsilon = 1$ for discrete objects. μ_{ij} is the

centered moments of order $i+j$, which are given by

$$\mu_{ij} = \sum_{k=1}^{n} (x_k - x_g)^i (y_k - y_g)^j \tag{1.64}$$

for the discrete objects. For the dense objects, we have

$$\mu_{ij} = \iint_R (x - x_g)^i (y - y_g)^i dxdy \tag{1.65}$$

We consider the interaction matrix related to the centered moments and have [Tahri 2005]

$$\mathbf{L}_{\mu_{ij}} = [\mu_{vx}\ \mu_{vy}\ \mu_{vz}\ \mu_{wx}\ \mu_{wy}\ \mu_{wz}] \tag{1.66}$$

with

$$\begin{cases} \mu_{vx} = -(i+\delta)A_p\mu_{ij} - iB_p\mu_{i-1,j+1} \\ \mu_{vy} = -jA_p\mu_{i+1,j-1} - (j+\delta)B_p\mu_{ij} \\ \mu_{vz} = -A_p\mu_{wy} + B_p\mu_{wx} + (i+j+2\delta)C_p\mu_{ij} \\ \mu_{wx} = (i+j+3\delta)\mu_{i,j+1} + (i+2j+3\delta)y_g\mu_{ij} + ix_g\mu_{i-1,j+1} - i\epsilon n_{11}\mu_{i-1,j} - j\epsilon n_{02}\mu_{i.j-1} \\ \mu_{wy} = -(i+j+3\delta)\mu_{i+1,j} - (2i+j+3\delta)x_g\mu_{ij} - jy_g\mu_{i+1,j-1} + i\epsilon n_{20}\mu_{i-1,j} + j\epsilon n_{11}\mu_{i,j-1} \\ \mu_{wz} = i\mu_{i-1,j+1} - j\mu_{i+1,j-1} \end{cases} \tag{1.67}$$

We see that the centered moments of either continue objects or discrete objects are invariant to 2-D translational motion. Many works [Mamistvalov 1998] have presented various methods to derive moment invariants to other transformations, such as scale and 2-D rotation. Several combinations of moments have been proposed in [Mamistvalov 1998]

$$I_s = \frac{m_{pq}}{m_{00}^{\frac{p+q+2}{2}}} \tag{1.68}$$

several combinations of moments that are invariant to 2-D translation, 2-D rotation are presented here and more details can be found in [Mukundan 1998], [Tahri 2004]

$$\begin{aligned} c_1 &= \frac{I_1}{I_2},\ c_2 = \frac{I_3}{I_4},\ c_3 = \frac{I_5}{I_6},\ c_4 = \frac{I_7}{I_6},\ c_5 = \frac{I_8}{I_6} \\ c_6 &= \frac{I_9}{I_6},\ c_7 = \frac{I_{11}}{I_{10}},\ c_8 = \frac{I_{12}}{I_{10}},\ c_9 = \frac{I_{13}}{I_{15}},\ c_{10} = \frac{I_{14}}{I_{15}} \end{aligned} \tag{1.69}$$

Two among these combinations of moments can be selected as visual servoing features to control the rotational velocities w_x and w_y.

To control the six DOFs of the robot, six combinations of image moments should be selected to do the visual servoing. In [Tahri 2004] and [Corke 2001], three image moment features are selected to control the three translational DOFs: x_g, y_g of the center of gravity, and the area $a = m_{00}$ of the object in the image.

From [Corke 2001] and [Tahri 2004], the object orientation α that can be defined from the second-order centered moments: $\alpha = (1/2)\arctan(2\mu_{11}/\mu_{20} - \mu_{02})$ can be used to control the rotation DOF.

Finally, a set of six image moments are combined to be the features do the visual servoing, for example: $\mathbf{s} = [x_g, y_g, m_{00}, \alpha, c_1, c_2]^T$, where c_1 and c_2 can be replaced by other invariant image moments.

Based on the kinematic relationship, one classical image moment visual servoing can be developed [Chaumette 2004]:

$$^c\boldsymbol{\tau}_c = -\lambda \mathbf{L}_s^+ \mathbf{e} \tag{1.70}$$

In which $\mathbf{e}$ stacks the errors between the desired image moment $\mathbf{s}$ (denoted as $\mathbf{s}^*$) and the current one ($\mathbf{e}$ is defined as $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$) and λ is a positive constant.

Then, joint velocity control can be obtained by using the following expression:

$$\dot{\mathbf{q}} = -\lambda \mathbf{J}_{pinv} \mathbf{L}_s^+ \mathbf{e} \tag{1.71}$$

Where $\mathbf{J}_{pinv}$ is the inverse Jacobian matrix of the robot relating the end-effector twist to the actuator velocities, i.e. $\mathbf{J}_{pinv}\ ^c\boldsymbol{\tau}_c = \dot{\mathbf{q}}$.

As image moments can be computed easily from the image in an unknown environment, image moment visual servoing has various application on medical industry. In [Mebarki 2008], a new ulrasound visual servoing based on image moments was proposed and got successful results with observing an ellipsoid object, an egg-shaped object and a real rabbit heart. A further study on visual-servoing method from 2-D ultrasound images by using image moments was presented in [Mebarki 2010] and this time the object observed was not symmetric to be sure it can control the 6 DOF of the system. But it still remains some problems, for example when the object moves with a high velocity, the control system may fail. Also, image moment visual servoing was applied on the control of the position and the orientation of a quadrotor [Ozawa 2011],[Ozawa 2013]. [Zhao 2015] introduced a new scheme that applies two Neural Network-based image features to solve

the problem of decoupling the rotational velocities around x and y axes of camera frame in visual servoing. A new adaptive image moment visual servoing controller was presented in [Shao 2016] which was proven that it can extend the convergence domain of a visual servo system and enhance the performance of the system. In [Xie 2016], an adaptive IBVS that uses the image moment features from projected points in a virtual camera image is used to control an unmanned aerial vehicle (UAV), which simplify the controller's derivation. Different from the traditional image moments, shifted moments were proposed to be the visual servoing features in [Tamtsia 2013], which can efficiently control the rotational DOF.

1.2.5 Controller singularity and hidden robot

It was defined in [Chaumette 1998] that the visual servoing controller singularity appears when the interaction matrix is rank deficient. The singularity problem of the mapping between the space of the observed image features and the Cartesian space has great influence on the accuracy of visual servoing. Therefore, it is important to avoid this kind of singularity problem involved in the visual servoing controller when we optimize a parallel robot with the control-based design methodology.

It was proven in [Andreff 2007] that the GS parallel robot can be controlled by using leg-direction-based visual servoing. The leg directions were chosen as image features and the control was derived based on the reconstruction from the image plane and leg-direction-based visual servoing exhibited better accuracy performance than the previous model-based control approach. During the control process of the Gough-Stewart platform, some surprising results were found without explanation:

- The end-effector does not systematically converge to the desired pose even though the observed leg directions do.

- For a robot composed of n legs, it can be controlled by observing only m leg directions ($m < n$). For the leg directions defined by the 3D unit vectors, the minimal number of observed legs is an integer greater than $n/2$.

- How to be sure that the stacking of the observation matrices cannot lead to local minima (the error in the observation space is non zero when the robot end-effector cannot move [Chaumette 1998]) in the Cartesian space?

- Are we sure that there is no singularity in the mapping between the leg direction space and the Cartesian space?

Thanks to the work of [Briot 2013], with the help of the content of "hidden robot", we can fully explain the problems mentioned above:

- It can be used to explain why observing only m leg directions can fully control a robot composed of n legs ($m < n$).

- It can be used to prove that there does not always exist a global diffeomorphism between the Cartesian space and the leg direction space.

- It simplifies the singularity analysis of the mapping between the Cartesian space and the leg direction space while replacing the study of controller singularity with the study of the singularity of a new robot.

- It can be used to certify that the robot will not converge to local minima.

In the classical control approach, the encoders measure the motion of the actuator. In the leg-based visual servoing, the leg directions or leg edges are observed. The content of "hidden robot" aims at finding the virtual actuators such observations correspond and the virtual architecture hidden inside the controller of a given robot. As we presented in Section 1.2.2, leg-direction-based visual servoing is based on the observation of the leg direction $\underline{\mathbf{u}}_i$ (see Fig. 1.20). In general case, the unit vector $\underline{\mathbf{u}}_i$ is parameterized by two independent coordinates, for example the angels α and β (see Fig. 1.21). $\cos\alpha = \mathbf{x} \cdot \mathbf{v} = \mathbf{y} \cdot \mathbf{w}$ (where $\mathbf{v}$ and $\mathbf{w}$ are defined such that $\mathbf{z} \cdot \mathbf{v} = \mathbf{z} \cdot \mathbf{w} = 0$) and $\cos\beta = \mathbf{u}_i \cdot \mathbf{x}$.

It is obvious that a U joint is able to orientate a link around two orthogonal axes of rotation. Therefore the vector $\underline{\mathbf{u}}_i$ can be parameterized by two angles defined by the U joint. Considering leg-direction-based visual servoing, the U joint can be seen as the virtual actuator when the leg direction is observed. If the value of $\underline{\mathbf{u}}_i$ is fixed, the U joint coordinates α and β must be constant; if the value of $\underline{\mathbf{u}}_i$ is changing, the U joint coordinates α and β must also vary. As a result, observing the direction of the leg is the same as observing the virtual U joint with its two geometric parameters α and β.

In most of the cases, the real leg architecture is not composed of U joints attached on links performing a motion with respect to the base frame. Thus it is necessary to fix the PPP chain on the preceding leg links to rebuild the full platform position and orientation. The PPP joints are only necessary if and only if the point A_{n-1} describes a

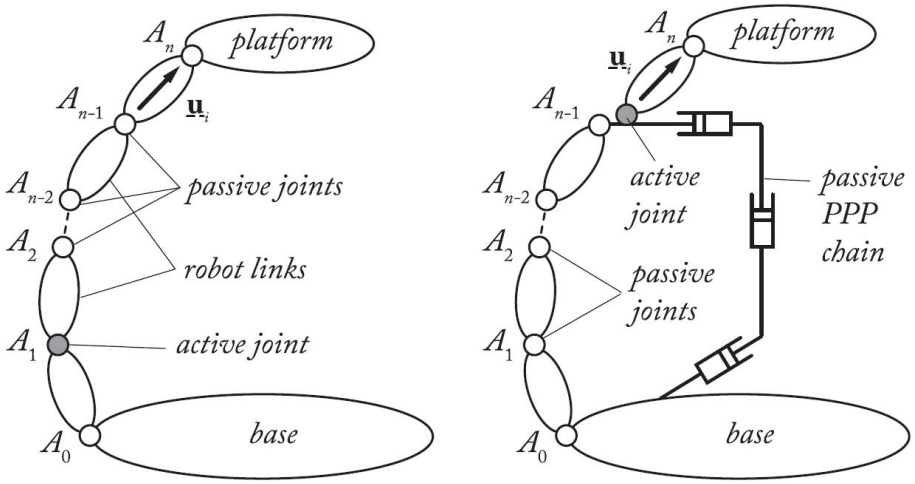

Figure 1.20: A general robot leg and its corresponding hidden robot leg when the vector $\underline{\mathbf{u}}_i$ is observed [Briot 2016]

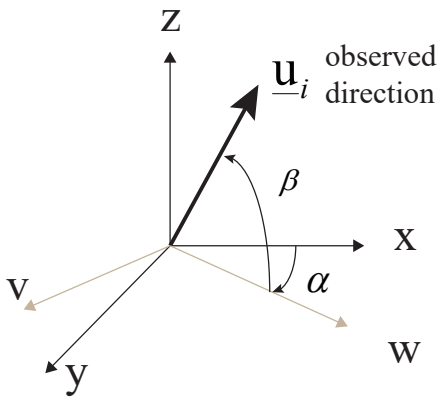

Figure 1.21: Parameterization of a unit vector $\underline{\mathbf{u}}_i$ with respect to a given frame $\mathbf{x}, \mathbf{y}, \mathbf{z}$

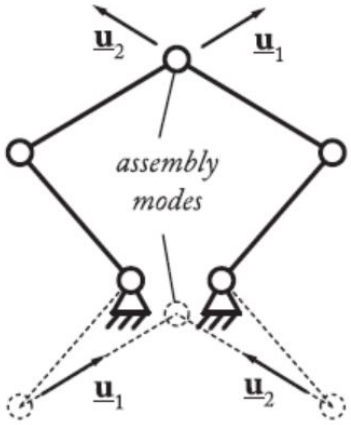

Figure 1.22: Two configurations of a five-bar robot for which the directions $\underline{\mathbf{u}}_i$ are identical (for $i = 1, 2$) [Briot 2016]

motion in the 3D space, if not, the number of P joints can be decreased. When the vector $\underline{\mathbf{u}}_i$ is constrained to move in a plane such as for planar legs, then the virtual actuator U joint becomes an R joint which is mounted on the passive PPP chain.

With the help of the hidden robot, the problems we mentioned above can be answered. The hidden robot can be used to explain why observing only m legs can fully control a robot composed of n legs ($m < n$) and the fact that the minimal number of observed legs should be integer greater than $n/2$. We take the Gough-Stewart platform as an example, the robot composed of 6 individual legs and it has 6 DOFs. In order to control the 6 DOFs, only 6 degrees of actuation are necessary, i.e. three actuated $\underline{U}$ joints are enough. Thus, only observing three legs can fully control the Gough-Stewart platform when applying leg-direction-based visual servoing.

Hidden robot can be used to solve the problem that end-effector not systematically converge to the desired pose when the observed leg directions do. As for a real controlled robot, its hidden robot may has different geometric and kinematics properties, which means that the hidden robot may have assembly modes and singular configurations different from those of the real robot. When the initial and final robot configurations are not included in the same aspect (ie., a workspace area that is singularity-free and bounded by singularities [Merlet 2006]), the end-effector will not be able to converge to the desired pose, but to another pose that has the same leg directions as the desired final pose (see Fig. 1.22).

The "hidden robot" also simplifies the leg-direction-based visual servoing controller singularity analysis by reducing the problem to the singularity analysis of a new robot. Controlling a parallel robot by leg-direction-based visual servoing is the same as controlling a virtual robot hide in the controller, the interaction matrix $\mathbf{M}^T$ involved in the controller gives the relationship between ${}^c\dot{\underline{\mathbf{u}}}_i$ and ${}^c\boldsymbol{\tau}_c$. Thus, $\mathbf{M}^T$ is the inverse Jacobian matrix of the hidden robot ($\mathbf{M}^{T+}$ is the Jacobian matrix of the hidden robot). Then the leg-direction-based visual servoing controller singularity is equivalent to the singularities of the corresponding hidden robot [Briot 2016].

- rank-deficiency of $\mathbf{M}^T$ is equivalent to find the Type 2 singularities of the hidden robot,
- rank-deficiency of $\mathbf{M}^{T+}$ is equivalent to find the Type 1 singularities of the hidden robot.

The hidden robot model can prove that the robot will not converge to local minima.

When using visual servoing, the robot could converge to local minima if the interaction matrix is rank deficient. A necessary and sufficient condition for the rank deficiency of the matrix is that $\mathbf{M}^{T+}$ is rank deficient, for example the hidden robot comes to a Type 1 singularity configuration and many tools have been developed by the mechanical design community for finding the singular configurations of robots and solutions can be provided to ensure that the hidden robot does not meet any Type 1 singularity.

Thanks to the hidden robot model, the study of the singularities of the interaction matrix is reduced to the study of the singularities of the virtual parallel robot hidden in the controller. There are a lot of tools that can help studying the singularities of the parallel robots, such as the Screw theory [Merlet 2006], the Grassman geometry [Merlet 2006] and the Grassman-Cayley algebra [Ben 2006].

The hidden robot models of planar parallel robots with various structure have been presented in [Briot 2016]. The hidden robot of Gough-Stewart platform is studied in [Briot 2013] and its application on the DELTA robot is shown in [Rosenzveig 2013]. In [Briot 2016], the content of hidden robot is applied to determine the singularities for the observation of three image lines in visual servoing. [Briot 2016] presents the study of controller singularity in the visual servoing of image points through the concept of hidden robot.

In addition, the hidden robot concept can also be used to define the control-based performance indices in the design process of robots when applying exteroceptive sensors (e.g. cameras, lasers, etc.). In [Kaci 2017], the hidden robot model is considered in the control-based design of a five-bar mechanism.

1.3 Optimal design of parallel robots

Optimal design methodology of parallel robots aims at finding the geometric parameters of the robot. However, when visual servoing is applied to the control of parallel robots, there are always several problems, such as the singularity problems (depends on the type of the controller and robot structure) and the internal performance (positioning accuracy,especially) (depends on the type of the controller and the geometric parameters of the robot). Therefore, it is necessary to consider an optimal design methodology for the design of the parallel robot so that its singularity problems can be avoided and the positioning accuracy can be certified.

The classical design methodology is proposed in [French 1985], they divided the design

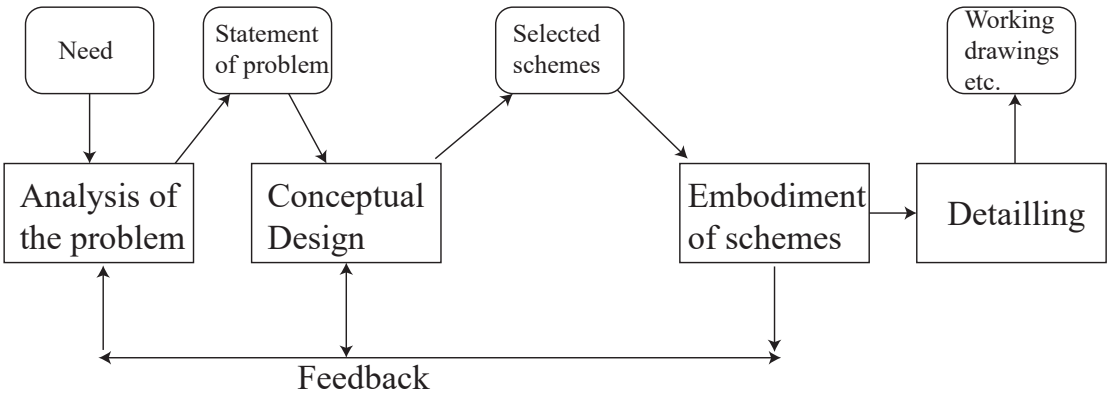

Figure 1.23: Typical French design methodology

into four steps (Fig. 1.23): first of all, be clear of the needs, analysing the requirements; secondly, the conceptual design, during which the broad solutions to the problem are generated and the designers choose the best design approach; thirdly, it's the embodiment of schemes, during which the design approach are developed and analysed; finally, it's the detailing, during which after getting the results of design, doing the drawing work or the producing work.

Based on the classical design methodology, we propose another new design method which takes the controller performance as the indices in the design process to get the optimal geometric parameters of the robot "control-based design methodology" (Fig. 1.24).

Corresponding to the classical design methodology, the control-based design is typically separated into four main phases: (1) be clear of the design specifications coming from the requirements, (2) the conceptual design that we propose and evaluate the concepts, (3) the embodiment of schemes during which the concepts are developed and analyzed, (4) the detailed design which includes the CAD drawing and the prototypes manufacture.

First of all, the performances required for the robot that designed according to the task should be clear above all. Obtaining the best geometric parameters with respect to desired kinematic and dynamic requirements is always the object when we optimize the parallel robots [Briot 2010]. For example, in [Germain 2013], the object is to design a 2 DOF parallel robot for fast pick and place operations, the robot footprint should be as small as possible and with very little vibration. In [Kaci 2018], for environmental friendly, a wooden robot with high accuracy and stiffness was designed. For a given desired workspace, this robot should be as small as and the positioning error should less than the value given throughout the workspace. In [Wu 2010], a parallel robot with

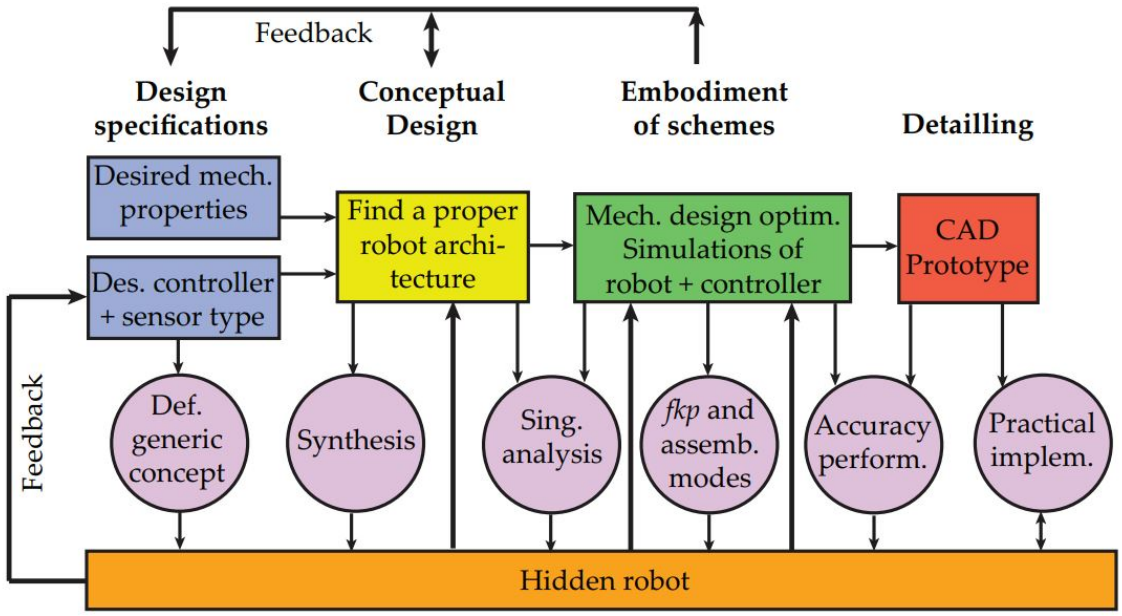

Figure 1.24: Scheme of Optimal design of robot [Briot 2014]

good motion/force transmissibility is dedicated to be optimal designed. This step corresponds to the Analysis of the problem in Fig. 1.23. In addition, for control-based design methodology, the controller which will be applied to control the robot should be decided in advance, including the type of the sensor and the intrinsic parameters of the sensor.

In the next step, a mechanical architecture should be chosen by a synthesis starting from the constraints on the task (in most cases, the required DOF should be satisfied). For example, in [Germain 2011], a novel two-DOF translational robot architecture was developed to overcome its counterparts in terms of mass in motion, stiffness and workspace size. [Chablat 2003] developed a new parallel robot, the "Orthoglide" (Fig. 1.25), which is a new Delta-type parallel robot features three fixed parallel linear joints which are mounted orthogonally, and a mobile platform which moves in the Cartesian space with fixed orientation to meet the advantages of both serial three-axis machines (regular workspace and homogeneous performances) and parallel kinematic architecture (good dynamic performances).

Then it comes to the embodiment of schemes. Once the architecture is selected, the design parameters ought to be determined. The design parameters can be categorized into two types. One is the parameters that determine the physical and geometrical characteristics of the mechanism. Usually, the geometry parameters that define the model of

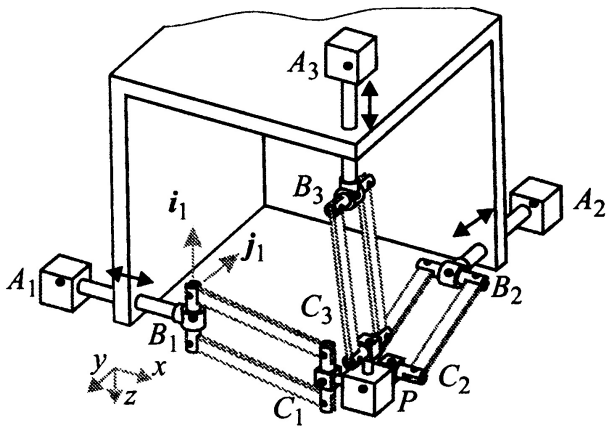

Figure 1.25: Basic kinematic architecture [Chablat 2003]

a parallel robot is huge. Thus we need to simplify the model of the robot and reduce the parameters that specify the robot model. For example, the number of the parameters identified by Masory [Masory 1996] that define the basic geometry of an 6-UPS robot is 132, and an optimization for 132 parameters is difficult and time-consuming. No simple guidelines for reducing the number of design parameters are given in the past. In most case, the symmetrical model of parallel robot is considered to reduce the design parameters [Merlet 2006] [Germain 2013] [Kaci 2018]. In addition, the other type of parameters are those have an influence on specific performances, for example, in [Kaci 2017], the location of the camera will affect the positioning error of the 5-bar robot.

Based on the design parameters and with respect to the specification of the optimal design, the objective function can be formulated. For example, to get a robot whose footprint is as small as possible, the objective function can be the area of the surface area of the bounding rectangle of the robot [Germain 2013], [Kaci 2017]. In order to get a robot with little vibration, the objective function calculates the robot natural frequencies [Germain 2013]. For getting a better dynamic performance, the objective function of the optimization problem is the mass of the moving bodies of the robot [Wu 2014]. [Huang 2013] aims designing a high-speed pick-and-place parallel robot. They proposed two global dynamic performance indices by taking into account the inertial and centrifugal/Coriolis torques of the actuated joints to formulate a weighted cost function for minimization to ensure the rigid body dynamic performance of the system, the goal attainment method [Censor 1977] was used to solve this multi-objective optimization problem. In [Chablat 2003], in order to design of the Orthoglide for a prescribed cartesian workspace, it aimed at defining the position of the base point A_i, the link lengths L and the linear ac-

tuator range $\Delta\rho$ (Fig. 1.26, 1.27) with respect to the limits on the transmission factors and as a function of the size of this prescribed workspace. In most cases, one objective function is not enough, hence the multi-objective optimization problem (MOOP) is needed. All the objective functions are normalized and weighted so that it can convert MOOP into a mono-objective optimization problem [Germain 2013] or applying the Pareto-optimal solutions [Briot 2017] [Deb 2000]. [Bolzon 2017] presented a new approach for the optimal design flexible 5-bar robot which considered the maximization of the workspace size and the maximization of the elastodynamics performance. They took the Maximum Inscribed Circle (MIC) as the index to evaluate the flatness of the usable workspace and the elastodynamic performance is assessed by determining the smallest eigenvalue over the Maximum Inscribed Workspace (MIW). This work also demonstrated that the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [Deb 2002] which permits to find the Pareto front which result a set of optimal solution is an appropriate optimization tool for the multi-objective optimization of the parallel robots. [Kelaiaia 2012] presented a methodology of dimensional design of parallel robots, based on an approach of multiobjective optimization of the different criteria of performance, for example, the stiffness, the kinematic and dynamic performances and the regular dexterous workspace. The multiobjective functions from the criteria of performance was defined and the genetic algorithm Strength Pareto Evolutionary Algorithm (SPEA-II) [Zitzler 2001] was applied to finding the resolution. [Bounab 2016] addressed the dimensional-systhesis-based kineto-elastostatic performance optimization of the DELTA robot. It took the Castigliano's energetic theorem for modeling the elastostatic behavior of the DELTA robot. The global structure's stiffness and global dexterity are considered together for the simultaneous optimization. In other cases, the requirements that the robot should satisfies can be classified as either compulsory or relaxable. In [Hao 2005], a methodology based on interval analysis was proposed for the optimal design of parallel manipulators with multi-criteria requirements. It allows to obtain all the possible solutions that satisfy a set of compulsory requirements and make the best compromise for the relaxable requirements. [Miller 2004] developed a new method of multidimensional kinematic optimization of the geometry of parallel robot. This method get an optimum compromise between manipulability and a new performance index, space utilization, which leads a maximization of the workspace volume and the manipulators with good manipulability.

Afterwards, the constraints related to the performances of the robot ought to be set with respect to the specification to complete the optimization problem formulation. For

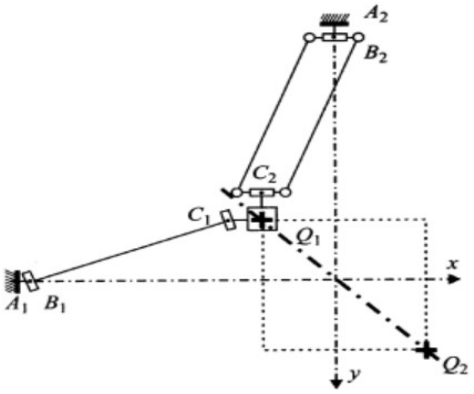

Figure 1.26: Q_1 configuration [Chablat 2003]

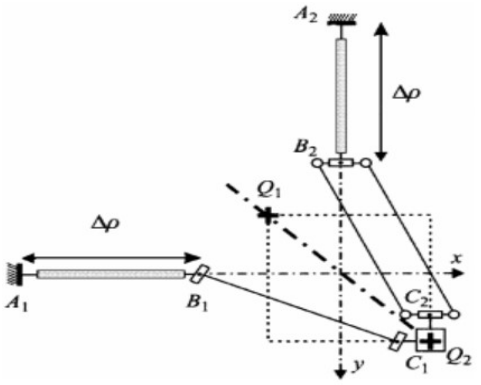

Figure 1.27: Q_2 configuration [Chablat 2003]

example, the regular dexterous workspace should be free of singularity [Germain 2013] and free of controller singularity [Kaci 2017], positioning error calculated by the error model should be less than the desired value [Kaci 2017], avoid the degeneracy of the joint [Briot 2017], no collision throughout the workspace. Finally, we need to find the optimal solution of the problem we formulated. The *ga* MATLAB function is used to find an approximate solution, then the *fmincon* MATLAB function is run to obtain a local optimum by taking the best result of the *ga* function as the starting point. In the end, the optimal designed parameters can be obtained.

For the detailed design, with respect to the optimal design solutions we get from the calculation, the CAD models are created and the prototypes are realized in the next step. This phase is more about the practical implementations.

It is obvious that the phases (1) and (3) are linked in order to modify and improve the design solutions in case of feasibility issues.

The detailed process of formulating the optimal design problem is presented here. Firstly, the mapping g: $\mathbf{\Phi} \rightarrow \mathbf{W}$ should be defined, where $\mathbf{\Phi}$ denotes the configuration space and $\mathbf{W}$ denotes the workspace. For any point P in the workspace of the robot, we can always define a set of matrices $\mathbf{K}_\alpha(\mathbf{P}, \boldsymbol{\pi})$, describing several different mechanical properties α of the robot, where $\boldsymbol{\pi}$ is the set of design parameters. The consistent scalar measures $\sigma_\beta(\mathbf{K})$ are defined physically that may be directly included in the design objectives or constraints. Besides, the indices $\eta_\gamma(g, \boldsymbol{\pi})$ for the valuation of the global performance of the robot are defined that depend on both the adopted structure g and the design parameters $\boldsymbol{\pi}$ [Briot 2010].

Then we can formulate the optimisation problem by achieving the best value for the

indices:

$$\eta_\gamma(g, \pi) \rightarrow \min_{\boldsymbol{\pi}}, \quad \forall\gamma \tag{1.72}$$

subject to the constraints

$$\sigma_\beta(\mathbf{K}(\mathbf{P}, \boldsymbol{\pi})) \in S, \quad \forall\alpha, \beta \tag{1.73}$$

that must be satisfied in all points in the desired workspace $\mathbf{W}_0$, which includes the desired task [Merlet 2015].

Since this methodology can't be solved by direct search methods, a discretization scheme can be utilised. We create the mesh grid throughout the workspace $\mathbf{W}$. At each points of the grid, the local constraints are evaluated, then the largest sub-cube of the grid which satisfies all constraints $\mathbf{W}_L$ is compared to the desired workspace $\mathbf{W}_0$ to be sure that it is larger.

The optimization for a large number of variables is always time-consuming, [Lopez 2019] applied the neural network and genetic algorithm on the optimal design of a parallel robot which decreases computation time and improves computation efficiency.

Apart from the objective-function based optimal design, [Liu 2007] proposed a new optimal kinematic design methodology: Performance-Chart based Design Methodology (PCbDM) was proposed. Its principle is first to reduce the number of parameters, then show the relationship between a performance index and associated design parameters in a limited space by the graphical representation of the performance indices as atlases. In the next step, these atlases can be used to choose the design parameters. One performance criterion corresponds to one chart, which can graphically and globally show the relationship between the criterion and design parameters. For such a reason, the fact that some performance criteria are antagonistic is no longer a problem in the design. The optimal design can consider multi-objective functions or multi-criteria, and also guarantees the optimal result. At the same time, the design method provides all possible optimal solutions to a design problem.

Control-based design is an objective-function based optimal design methodology which takes the controller performance as the indices in the design process. With the content of the visual servoing controller, positioning error models can be created to evaluate the controller accuracy. Based on the hidden robot content, the controller singularity can be analysed. Both the controller accuracy and the content of corresponding hidden robot are all the control-based performance indices to help getting the optimal design results of the robot.

Until now, the controller, especially the vision-based controller, is never taken into account in the optimal design process, which is a big issue due to its singularity problems and of its own internal performance (positioning accuracy, especially). Therefore, the control-based design methodology helps solving this problem and helps getting the optimal designed geometric parameters of the parallel robot when visual servoing controller is the dedicated controller.

1.4 Summary

This chapter introduced the state of the art regarding parallel robots, and visual servoing.

In the first Section of this chapter, a general study of parallel robots was presented. We gave the definition of parallel robots, several classical structures of parallel robots, and their applications in industry. Afterwards we present the general geometric modeling, kinematic modeling and singularities of parallel robots, the dynamic modeling of robot and the control scheme of parallel robot.

Then visual servoing was introduced, We start with the reason of applying visual servoing, how to get the interaction matrix related to the image features and the control scheme. When it is not efficient to observe the end-effector, other alternative methods can be used, such as leg-direction observation or the line observation. In the condition of observing the end-effector directly, image moments can be used as features to do the visual servoing. After that, the controller singularity was proposed and a powerful tool in the study of controller singularity "hidden robot" was presented in details.

In the end, the optimal design process of the parallel robot was presented.

In the following part of this thesis report, this control-based design methodology will be applied to the design of three different types of parallel robots: the Five-bar mechanism, the DELTA robot and the Gough-Stewart platform. At the same time, the co-simulations and the experiment verification will be performed in order to prove that this optimal design methodology really helps get the optimal geometric parameters of the parallel robot when takes the controller performance into account in the design process.

Chapter 2

Optimal design of a Five-bar mechanism

Control-based design methodology is a methodology that takes into account the performance of the controller in the design process to get the optimal geometric parameters of the robot. The optimal control-based design of a Five-bar mechanism has been performed in [Kaci 2017], [Kaci 2018] and the PhD Thesis of L. Kaci on the control-based design of a Five-bar mechanism.

In this chapter, we revisit the previous results and replay the optimization of the Five-bar mechanism for several values of pixel noise to see the impact on the design. Two types of controllers are envisaged for the control of the motions of the Five-bar mechanism: leg-direction-based visual servoing and line-based visual servoing. Based on these two controllers, positioning error models taking into account the error of observation coming from the camera are developed. Then, design optimization problems are formulated in order to find the optimal geometric parameters and camera placement for the Five-bar mechanism for each type of controller. Co-simulations of the robots optimized for the two types of controllers are finally performed in order to check the accuracy performance of the two robots. Finally, the conclusion is drawn.

2.1 Kinematics and design specifications of the Five-bar mechanism

Five-bar mechanism is a two DOF planar parallel robot whose end-effector moves within a plane along two axis. The Five -bar robot geometry is illustrated in Fig. 2.1. The end-effector is located at point C. The parallel mechanism consists of four links: two proximal links are the bodies between A_i and B_i $(i = 1, 2)$ and two distal links are the bodies between B_i and C $(i = 1, 2)$. The proximal and distal links are connected by three passive revolute revolute joints located at B_1, B_2 and C. The two revolute joints located at A_1 and A_2 are active. All revolute joint axes are along Z_0. The end-effector C moves along the axis X_0 and Y_0 and its coordinates are denoted as x and y, respectively. The vector of actuated coordinates is $\mathbf{q}^T = [q_1\ q_2]$.

The position of the point C is given by

$$\mathbf{C} = \mathbf{A}_i + l_{1i}\underline{\mathbf{v}}_i + l_{2i}\underline{\mathbf{u}}_i \tag{2.1}$$

where l_{1i} and l_{2i} are the length of the distal and proximal links A_iB_i, B_iC $(i = 1, 2)$ respectively. $\mathbf{A}_i = [\delta_i\ 0]^T$ ($\delta_1 = -l_{OA_i}$ and $\delta_2 = +l_{OA_i}$) is the position of the point A_i with respect to the base frame (Fig. 2.1). The vectors $\underline{\mathbf{v}}_i$ and $\underline{\mathbf{u}}_i$ are the unit vectors defining the direction of the links A_iB_i and B_iC $(i = 1, 2)$ respectively, $\mathcal{L}_1$ and $\mathcal{L}_2$ are the lines passing through the distal links B_1C and B_2C (Fig. 2.1).

Rearranging equation (2.1), we have

$$\mathbf{C} - \mathbf{A}_i - l_{1i}\underline{\mathbf{v}}_i = l_{2i}\underline{\mathbf{u}}_i \tag{2.2}$$

Then we square both sides of the equation (2.2) and we get

$$(x - \delta_i - l_{1i}\cos q_i)^2 + (y - l_{1i}\sin q_i)^2 = l_{2i}^2 \tag{2.3}$$

After mathematical derivations, we have

$$q_i = 2\tan^{-1}\left(\frac{-b_i \pm \sqrt{b_i^2 - c_i^2 + a_i^2}}{c_i - a_i}\right) \tag{2.4}$$

where $a_i = -2l_{1i}(x - \delta_i)$, $b_i = -2l_{1i}y$, $c_i = (x - \delta_i)^2 + y^2 + l_{1i}^2 - l_{2i}^2$. The equation (2.4) is the inverse geometric model of the Five-bar mechanism.

Table 2.1: Requirements of the Five-bar mechanism

Regular dexterous workspace size	800 mm × 100 mm
Positioning accuracy wherever in regular dexterous workspace	⩽0.5 mm

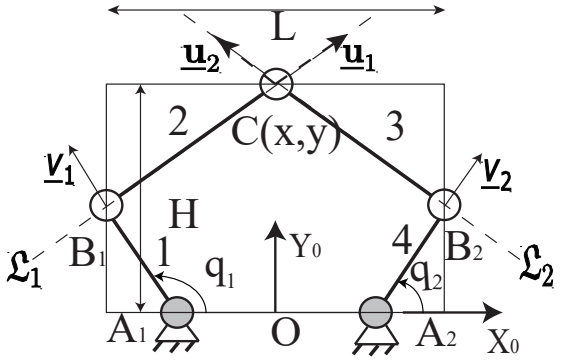

Figure 2.1: Five-bar mechanism

By the time derivative of (2.3), we can obtain the first-order kinematic equation that relates the platform translational velocity $\boldsymbol{\tau_p} = [v_x, v_y]^T$ ($v_x$and v_y are the velocities of the platform along the axis X_0 and Y_0) to the actuator velocities as:

$$\mathbf{A}\boldsymbol{\tau_p} + \mathbf{B}\dot{\mathbf{q}} = 0 \tag{2.5}$$

where

$$\mathbf{A} = \begin{bmatrix} l_{21}\underline{\mathbf{u}}_1^T \\ l_{22}\underline{\mathbf{u}}_2^T \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} l_{11}l_{21}\underline{\mathbf{u}}_1^T\underline{\mathbf{v}}_1^{\perp} & 0 \\ 0 & l_{12}l_{22}\underline{\mathbf{u}}_2^T\underline{\mathbf{v}}_2^{\perp} \end{bmatrix} \tag{2.6}$$

with $\underline{\mathbf{v}}_i^{\perp} = [-\sin q_i \;\; \cos q_i]^T$. Thus we have

$$\boldsymbol{\tau_p} = -\mathbf{A}^{-1}\mathbf{B}\dot{\mathbf{q}} = \mathbf{J}\dot{\mathbf{q}} \tag{2.7}$$

where $\mathbf{J}$ is the Jacobian matrix of the Five-bar mechanism. We also have

$$\dot{\mathbf{q}} = -\mathbf{B}^{-1}\mathbf{A}\boldsymbol{\tau} = \mathbf{J}_{inv}\boldsymbol{\tau_p} \tag{2.8}$$

where $\mathbf{J}_{inv}$ is the inverse Jacobian matrix.

As introduced in Section. 1.1.3, based on the kinematic model of the parallel robot, the

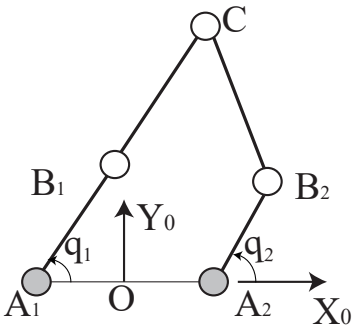

Figure 2.2: Type 1 singularity configuration of Five-bar mechanism

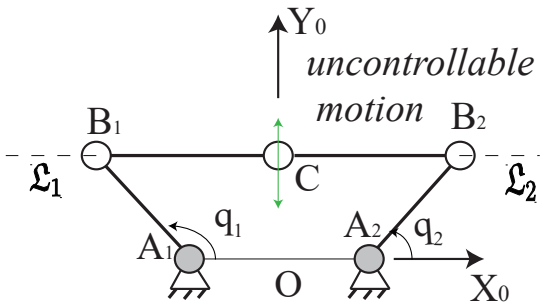

Figure 2.3: Type 2 singularity configuration of Five-bar mechanism

singularities of Five-bar mechanism can be classified into three different types [Merlet 2006]. The Type 1 singularities appear when the kinematic matrix **B** is rank deficient. The Type 2 singularities appear when the kinematic matrix **A** becomes rank deficient. The Type 3 singularities are the configurations that both matrix **B** and **A** are rank deficient at the same time.

From equation (2.6), we see that the Type 1 singularity of Five-bar mechanism occurs when $\underline{\mathbf{u}}_i$ and $\underline{\mathbf{v}}_i$ are parallel, which is the situation that one leg is fully stretched or folded (Fig 2.2) [Park 1999]. For the Type 2 singularity of Five-bar mechanism, **A** is rank deficient when $\underline{\mathbf{u}}_1$ and $\underline{\mathbf{u}}_2$ are parallel, which means that the points B_1,B_2 and C are aligned (Fig. 2.3) [Park 1999].

The requirements that the Five-bar mechanism should satisfy is shown in Tab. 2.1. The shape of desired regular dexterous workspace [Merlet 2006] is a rectangle whose length and width are given in Tab. 2.1. For practical industrial reasons, in the condition of getting a regular dexterous workspace larger than the size given in Tab. 2.1, the robot must be as compact as possible.

To avoid this kind of singularity, in the optimal design process of the Five-bar mechanism, the angle between B_1C and B_2C are set to be in the range from 15° to 165°. In addition, another range from 30° to 150° is added to test the impacts of the range of this angle on the final optimization results.

In order to get the desired 0.5 mm of positioning accuracy specified in Tab. 2.1 which is the most important criterion in the design of a robot, we propose to apply visual servoing approaches. A single camera (2336 × 1728 pixels of resolution and a focal length of 10 mm) fixed in the space is used to get the vision information. Two types of standard visual servoing approaches will be applied here: leg-based visual servoing and line-based visual

servoing [Vignolo 2014].

2.2 Vision-based kinematic of Five-bar mechanism

Vision-based kinematic model of parallel robot aims at finding the relationship between the twist of the moving platform and the time variation of the image features, which is the interaction matrix in the visual servoing. Four main approaches for visual servoing of the Five-bar mechanism will be applied.

2.2.1 Vision-based kinematics of a Five-bar mechanism using the leg-direction-based visual servoing

Leg-direction-based visual servoing developed in [Andreff 2005] is applied on the control of a Five-bar mechanism. The leg direction $\underline{\mathbf{u}}_i$ extracted from the observation of the robot leg is selected as the feature to do the visual servoing. For a Five-bar mechanism, the vector $\underline{\mathbf{u}}_i$ (Fig. 3.1) can be obtained directly from

$$\underline{\mathbf{u}}_i = (\mathbf{C} - \mathbf{A}_i - l_{1i}\underline{\mathbf{v}}_i)/l_{2i} \tag{2.9}$$

Differentiating the equation (2.9) with respect to time, then we have

$$\dot{\underline{\mathbf{u}}}_i = (\boldsymbol{\tau}_p - l_{1i}\underline{\mathbf{v}}_i^{\perp}\dot{q}_i)/l_{2i} \tag{2.10}$$

where $\boldsymbol{\tau_p}$ is the twist of the point C. Then, by the help of (2.8), (2.6) it comes that

$$\dot{\underline{\mathbf{u}}}_i = (\mathbf{I}_3 + l_1\underline{\mathbf{v}}_i^{\perp}a_i/b_{ii})/l_{2i}\ \boldsymbol{\tau}_p = \mathbf{M}_{ui}^T\boldsymbol{\tau}_p \tag{2.11}$$

where $\mathbf{I}_3$ is the (3×3) identity matrix and $\mathbf{M}_{ui}^T$ is the interaction matrix related to the leg-directions. It can be proven that the matrix $\mathbf{M}_{ui}^T$ is of rank 1. Then to fully control the location of the end-effector, it is better to observe all the two legs. We can get $\mathbf{M}_u^T$ by stacking the matrix $\mathbf{M}_{ui}^T$ of the two legs $(i = 1, 2)$. The two leg-direction-based visual servoing control schemes applied in this case are presented in detail in Section 1.2.2. One is based on the control of the projection of the edges of the legs, the other one is directly based on the control of the 3D estimation of the leg directions.

2.2.2 Vision-based kinematics of a Five-bar mechanism using the line-based visual servoing

Line-based visual servoing takes the lines passing through the leg axes as features to control the movement of the end-effector [Andreff 2002]. In the condition of using line-based visual servoing, the line $\mathcal{L}_i$ passing through robot link cylinder axis i is expressed by its Plücker coordinates $({}^c\underline{\mathbf{u}}_i, {}^c\mathbf{h}_i)$ (see definition in [Andreff 2002] and Fig. 1.18). Based on the fact that the point C is the intersection point of the lines of the two observed cylindrical legs, the homogeneous coordinates of point C can be got by applying the formula of the intersection point between two lines expressed in Plücker coordinates, for $i = 1, 2$ [Selig 2004]:

$$\mathbf{C}_w = (-(\mathbf{h}_1 \cdot \mathbf{N}) \cdot \underline{\mathbf{u}}_2 + (\mathbf{h}_2 \cdot \mathbf{N}) \cdot \underline{\mathbf{u}}_1 + (\mathbf{h}_1 \cdot \underline{\mathbf{u}}_2) \cdot \mathbf{N} : (\underline{\mathbf{u}}_1 \times \underline{\mathbf{u}}_2) \cdot \mathbf{N}) \tag{2.12}$$

where $(\underline{\mathbf{u}}_1, \mathbf{h}_1)$ and $(\underline{\mathbf{u}}_2, \mathbf{h}_2)$ are the Plücker coordinates of the two lines passing through the two robot legs respectively, $\mathbf{N}$ is a unit vector along a coordinates axis, with $(\underline{\mathbf{u}}_1 \times \underline{\mathbf{u}}_2) \cdot \mathbf{N}$ non-zero.

Moving the right terms of (2.12) to the left-hand side and extend it, naming the equations with f_i leads to

$$f_1 = x + h_{1z}u_{2x} - h_{2z}u_{1x} = 0 \tag{2.13}$$

$$f_2 = y + h_{1z}u_{2y} - h_{2z}u_{1y} = 0 \tag{2.14}$$

$$f_3 = z - h_{1x}u_{2x} - h_{1y}u_{2y} = 0 \tag{2.15}$$

$$f_4 = w - u_{1x}u_{2y} + u_{2x}u_{1y} = 0 \tag{2.16}$$

where (x, y, z, w) are the homogenous coordinates of $\mathbf{C}_w$, (u_{ix}, u_{iy}, u_{iz}) are the cartesian components of the vector $\underline{\mathbf{u}}_i$, (h_{ix}, h_{iy}, h_{iz}) are the Cartesian components of the vector $\mathbf{h}_i$.

Differentiating (2.13), (2.14), (2.15) and (2.16) with respect to time, we have

$$\dot{x} - h_{2z}\dot{u}_{1x} + h_{1z}\dot{u}_{2x} + u_{2x}\dot{h}_{1z} - u_{1x}\dot{h}_{2z} = 0 \tag{2.17}$$

$$\dot{y} - h_{2z}\dot{u}_{1y} + h_{1z}\dot{u}_{2y} + u_{2y}\dot{h}_{1z} - u_{1y}\dot{h}_{2z} = 0 \tag{2.18}$$

$$\dot{z} - h_{1x}\dot{u}_{2x} - h_{1y}\dot{u}_{2y} - u_{2y}\dot{h}_{1x} - u_{2y}\dot{h}_{1y} = 0 \tag{2.19}$$

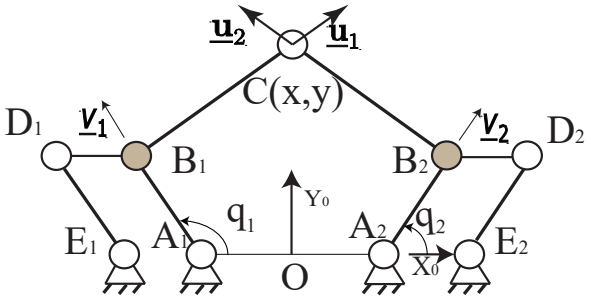

Figure 2.4: The hidden robot involved into the leg-direction-based visual servoing of a Five-bar mechanism (the gray pairs denote the actuated joints)

$$\dot{w} - u_{2y}\dot{u}_{1x} + u_{2x}\dot{u}_{1y} + u_{1y}\dot{u}_{2x} - u_{1x}\dot{u}_{2y} = 0 \tag{2.20}$$

In the end, writing (2.17), (2.18), (2.19) and (2.20) in matrix form, then we have

$$\dot{\mathbf{l}} = \mathbf{M}_{uh}^{T}\boldsymbol{\tau}_p \tag{2.21}$$

where $\mathbf{l} = \begin{bmatrix} \underline{\mathbf{u}} \\ \mathbf{h} \end{bmatrix}$.

The two line-based visual servoing control schemes applied in this case are detailed introduced in Section 1.2.3. One is based on the control of the projection of the edges of the legs, the other one is directly based on the control of the 3D estimation of the Plücker coordinates of the lines passing through the legs.

2.3 Controller singularity and hidden robot of Five-bar mechanism

As presented in the Section 1.2.5, the controller singularity appears when the interaction matrix is rank deficient. However the singularities of the interaction matrix are difficult to find, but in [Briot 2013], a tool named "hidden robot" was proposed to simplify the analysis of the singularity of the mapping involved into the controller (see more details in Section 1.2.5).

The hidden robot involved into the leg-direction-based visual servoing for a Five-bar mechanism is presented in [Briot 2016] (see Fig. 2.4). As is shown in the figure, the virtual hidden robot is made of two passive parallelogram (Π) joints $A_iB_iD_iE_i$ fixed on the ground. The joint at point B_i is actuated to control the direction of the link B_iC

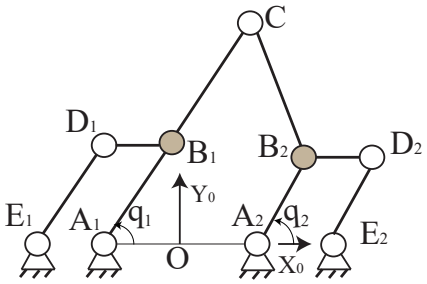

Figure 2.5: Example of Type 1 singularity configuration for a hidden robot of Five-bar mechanism

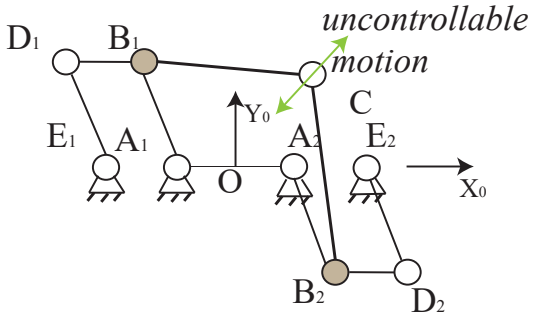

Figure 2.6: Example of Type 2 singularity configuration for a hidden robot of Five-bar mechanism

$(i = 1, 2)$.

After a kinematic analysis of the hidden robot, we have

- Type 1 singularities appear when one leg is fully stretched or folded (see Fig. 2.5).

- Type 2 singularities appear when $\underline{\mathbf{v}}_1$ and $\underline{\mathbf{v}}_2$ are parallel (see Fig. 2.6). This kind of singularity affects a lot the performance in terms of both accuracy and stability of controller and need to be avoided in the optimal design of parallel robots.

The hidden robot involved into the line-based visual servoing for a Five-bar mechanism is presented in [Kaci 2017] (see Fig. 2.7), which is a $\underline{RR}PRP\underline{RR}$ mechanism. For the line-based controller, the only measurements are the Plücker coordinates $({}^c\underline{\mathbf{u}}_i, {}^c\mathbf{h}_i)$. These coordinates can be fixed thanks to the virtual actuators at points A_i and B_i in Fig. 2.7 while all other joints are passive [Kaci 2017]. Type 2 singularities appear when the vector ${}^c\underline{\mathbf{u}}_1$ and ${}^c\underline{\mathbf{u}}_2$ are parallel, which means that the two lines B_1C and B_2C are parallel or coincide (Fig. 2.8). Such singularity conditions appear in the same configuration loci as Type 2 singularity configurations of a five bar mechanism (Fig. 2.3).

Based on these considerations, in order to have good performance in terms of accuracy and stability when using leg-direction based visual servoing and line-based visual servoing, singularities of the corresponding hidden robot must be avoided in the robot operational workspace, as well as the singularities of the real robots.

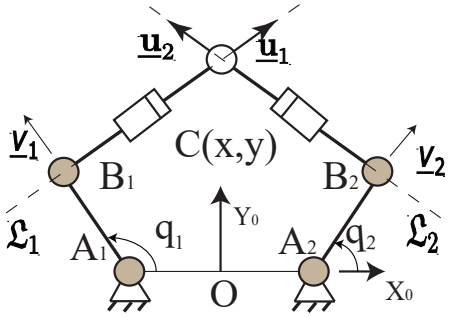

Figure 2.7: The hidden robot involved into the line-based visual servoing of a Five-bar mechanism

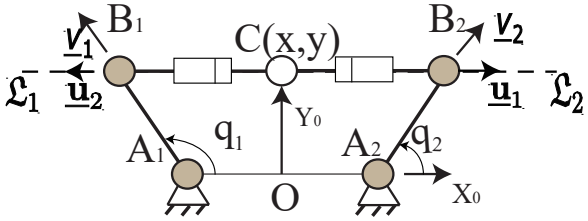

Figure 2.8: Example of singularity configuration for the hidden robot involved into the line-based visual servoing of a Five-bar mechanism

2.4 Positioning error models of Five-bar mechanism

When leg-direction-based visual servoing and line-based visual servoing are considered, the positioning errors come from the camera observation error of the legs directions and the leg Plücker coordinates [Kaci 2018]. With respect to the accuracy requirements for the robot design, the positioning errors of visual servoing which come from camera observation and interaction model should be considered. In this section, the positioning error models of leg-direction visual servoing and line-based visual servoing are described.

When the camera observe the robot links, the link edges are projected into the image plane. So we have ℓ_i^1 and ℓ_i^2 in image plane of the robot link i, thus we can get the intersection points of ℓ_i^j and the image plane boundary $\mathbf{P}_{ij}^1, \mathbf{P}_{ij}^2$ (see Fig. 1.18 and 2.9). By applying equations (1.40) and (1.42), we get ${}^c\underline{\mathbf{n}}_i^j$ $(j = 1, 2)$ from the coordinates in pixel of the point $\mathbf{P}_{ij}^1, \mathbf{P}_{ij}^2$. By taking the derivative with respect to time, we get the relationship between the derivative with respect to time of the vector ${}^c\underline{\mathbf{n}}_i^j$ and the time derivative of $\mathbf{P}_{ij}^1, \mathbf{P}_{ij}^2$. It can be written in the form:

$$ {}^c\dot{\underline{\mathbf{n}}}_i^j = \mathbf{J}_{ni} \begin{bmatrix} \dot{\mathbf{P}}_{ij}^1 \\ \dot{\mathbf{P}}_{ij}^2 \end{bmatrix} \tag{2.22} $$

Where $\mathbf{J}_{ni}$ is the matrix which transforms the time derivatives of $\mathbf{P}_{ij}^1$ and ‘$\mathbf{P}_{ij}^2$ to time derivatives of ${}^c\underline{\mathbf{n}}_i^j$. By applying (2.22) to (1.33) and (1.46), we have:

$$ {}^c\boldsymbol{\tau}_c = \mathbf{M}_u^{T+} \mathbf{J}_u \, \mathbf{J}_n \dot{\mathbf{P}} \tag{2.23} $$

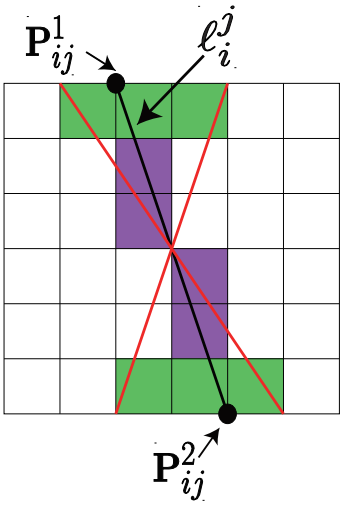

Figure 2.9: One-pixel error on the intersection of the image boundary and the observed line[Kaci 2018]

and

$$^{c}\boldsymbol{\tau}_c = \mathbf{M}_{uh}^{T+} \begin{bmatrix} \mathbf{J}_u \\ \mathbf{J}_h \end{bmatrix} \mathbf{J}_n \dot{\mathbf{P}} \tag{2.24}$$

Where the definitions of the matrices $\mathbf{J}_u$ and $\mathbf{J}_h$ are given in the Section 1.2.2 and 1.2.3. $\mathbf{J}_n$ is block diagonal matrix containing the matrices $\mathbf{J}_{ni}$, $\dot{\mathbf{P}} = \left[\dot{\mathbf{P}}_{11}^{1\,T}, \dot{\mathbf{P}}_{11}^{2\,T}, \cdots, \dot{\mathbf{P}}_{m2}^{1\,T}, \dot{\mathbf{P}}_{m2}^{2\,T}\right]^T$ when a set of m legs are observed. In reality, the observation errors come from the noise in the image plane. We thus model this noise by adding a random shift in pixels. This random shift is added on the intersection points of image projection of the leg edges and the image plane boundary (Fig. 2.9). Since the error of observation is very small, the error model relating the movement of end-effector $\boldsymbol{\delta}\mathbf{x}$ to the variation $\boldsymbol{\delta}\mathbf{P}$ of the intersection points can be approximated by using the first-order geometric models based on the use of the transformations given at (2.23) and (2.24) [Kaci 2018]:

$$\boldsymbol{\delta}\mathbf{x} = \mathbf{M}_{u}^{T+} \mathbf{J}_u \, \mathbf{J}_n \boldsymbol{\delta}\mathbf{P} \tag{2.25}$$

for leg-direction-based visual servoing control, and

$$\boldsymbol{\delta}\mathbf{x} = \mathbf{M}_{uh}^{T+} \begin{bmatrix} \mathbf{J}_u \\ \mathbf{J}_h \end{bmatrix} \mathbf{J}_n \boldsymbol{\delta}\mathbf{P} \tag{2.26}$$

for line-based visual servoing, where $\boldsymbol{\delta}\mathbf{x}$ is the robot platform positioning error due to the

pixellic observation error $\boldsymbol{\delta}\mathbf{P}$.

For the points in image plane, the resolution of the camera is taken as ± 1 pixel. Thus every component of vector $\boldsymbol{\delta}\mathbf{P}$ can take the values ± 1 [Kaci 2017], In this case, every component of vector $\boldsymbol{\delta}\mathbf{P}$ will take the values ± 0.5 and ± 0.1 in the optimal design process to be the comparison to see the impacts on the final results. These two error models will be used to characterize the robot accuracy in optimal design process when controlled using strategies of Sections 1.2.2 and 1.2.3.

2.5 Optimal design process of Five-bar mechanism

We assume that the two kinematic chains of the Five-bar mechanism are identical. Then the Five-bar mechanism can be defined by the following geometric parameters: l_1, l_2, l_3 with $l_1 = l_{A_1A_2}, l_2 = l_{A_1B_1} = l_{A_2B_2}$ and $l_3 = l_{B_1C} = l_{B_2C}$ (see Fig. 2.1). When leg-direction-based visual servoing and line-based visual servoing are applied, the radius of the cylindrical distal links of robot also affects the positioning error [Kaci 2018]. To simplify the calculation, the radius of the observed cylinders B_1C and B_2C is set to be 4 cm [Kaci 2017]. (x_c, z_c, z_c) define the position of the camera with respect to the robot frame. In order to reduce the number of decision variables in the optimization problem, the orientation of the camera plane is fixed to be parallel to the plane $(\mathbf{X_0OY_0})$. To observe the robot legs in a symmetrical way, the coordinate x_c of the camera along the axis $\mathbf{X}_0$ is fixed at $x_c = 0$ m. l_{W_x} and l_{W_y} are the dimensions along $\mathbf{X_0}$ and $\mathbf{Y_0}$ of the rectangular dexterous workspace in which all the performance should be satisfied. The method of getting the regular dexterous workspace is introduced in [Germain 2013] and following performances must be satisfied throughout the dexterous workspace:

- Type 2 and constraint singularity-free: ensuring the Five-bar mechanism will not meet any Type 2 or serial singularities.
- controller singularity-free: no singularities of the controllers
- end-effector in image: ensuring that all the robot distal legs can be observed when using leg-direction-based visual servoing and line-based visual servoing.
- required positioning accuracy: Maximal positioning error computed with the positioning error models should be lower than 0.5 mm.

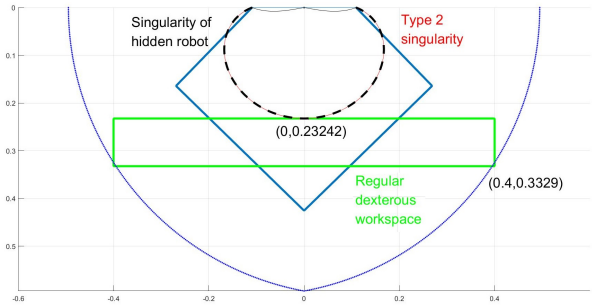

Figure 2.10: Regular dexterous workspace of the optimized Five-bar mechanism using line-based visual servoing ($15° \sim 165°$, $\pm$ 1 pixel)

Since the robot footprint should be minimal, then the footprint of Five-bar mechanism is characterized by the rectangular area $A = LH$ in which the robot is included when the links A_iB_i and B_iC are perpendicular (See Fig. 2.1).

In order to create a compact Five-bar mechanism which has the specifications detailed in Tab. 2.1, the following optimization problem is formulated:

$$\begin{aligned} \text{minimize} \quad & A = LH \\ \text{over} \quad & \mathbf{x} = [l_1, l_2, l_3, y_c, z_c] \\ \text{subject to} \quad & l_{W_X} \geqslant 800 \text{ mm}, \; l_{W_y} \geqslant 100 \text{ mm} \end{aligned} \tag{2.27}$$

The algorithm of computing the size of the maximal dexterous workspace is presented in [Briot 2010]. The previous optimization algorithm is applied for the design of the mentioned Five-bar mechanism. Two comparison optimal designs are performed at the same time, one comparison is setting the angle between B_1C and B_2C (Fig. 2.1) to be in the range from 30° to 150°, the other comparison is setting the camera observation error to be $\pm$0.5 pixel and $\pm$0.1 pixel. The optimal design results are given in Tab. 2.2.

The results in Tab. 2.2 show that, in terms of the robot size, the Five-bar mechanism designed based on the line-based visual servoing is smaller than the robot designed based on the leg-direction based visual servoing, but the difference is almost negligible. The robot size grows with the decrease of the range of the angle between B_1C and B_2C (Fig. 2.1). The robot size decreases with the decrease of the camera observation error.

Based on the optimal design results, the position of the largest regular dexterous workspace can be found (see from Fig. 2.10 to 2.17).

For each controller, based on the positioning error models, the values of the positioning errors in the regular dexterous workspace can be computed. All these results are

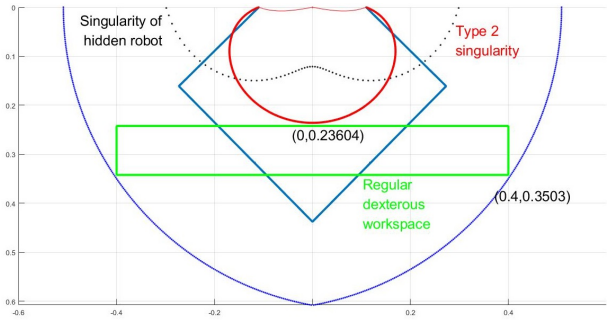

Figure 2.11: Regular dexterous workspace of the optimized Five-bar mechanism using leg-direction-based visual servoing ($15° \sim 165°$, $\pm$ 1 pixel)

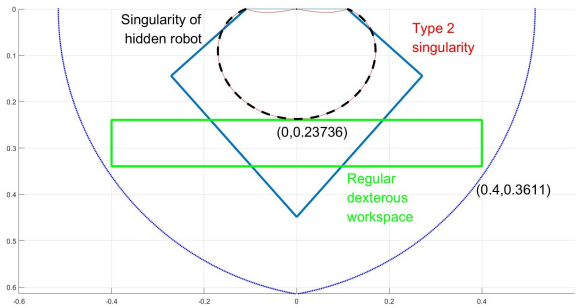

Figure 2.12: Regular dexterous workspace of the optimized Five-bar mechanism using line-based visual servoing ($30° \sim 150°$, $\pm$ 1 pixel)

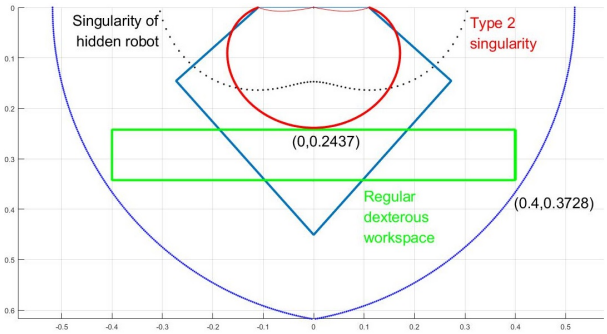

Figure 2.13: Regular dexterous workspace of the optimized Five-bar mechanism using leg-direction-based visual servoing ($30° \sim 150°$, $\pm$ 1 pixel)

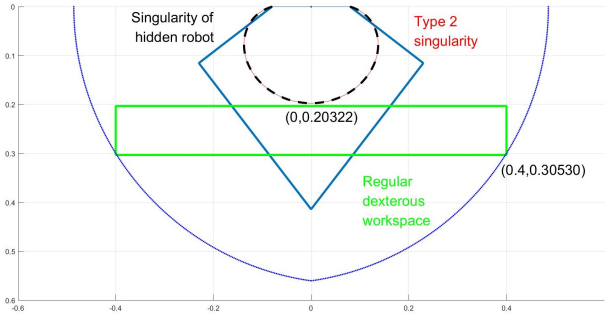

Figure 2.14: Regular dexterous workspace of the optimized Five-bar mechanism using line-based visual servoing ($15° \sim 165°$, $\pm$ 0.5 pixel)

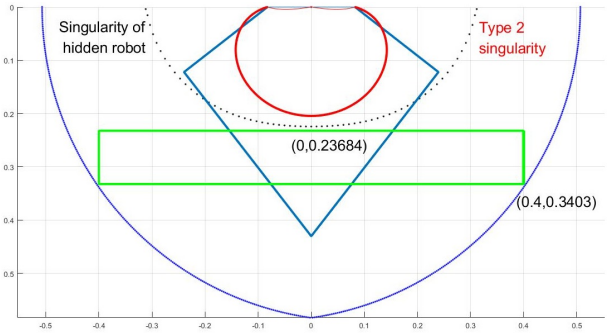

Figure 2.15: Regular dexterous workspace of the optimized Five-bar mechanism using leg-direction-based visual servoing ($15° \sim 165°$, $\pm$ 0.5 pixel)

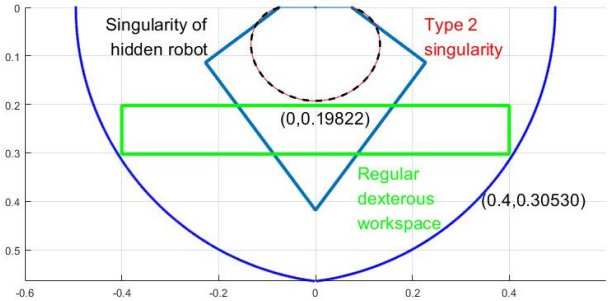

Figure 2.16: Regular dexterous workspace of the optimized Five-bar mechanism using line-based visual servoing ($15° \sim 165°$, $\pm$ 0.1 pixel)

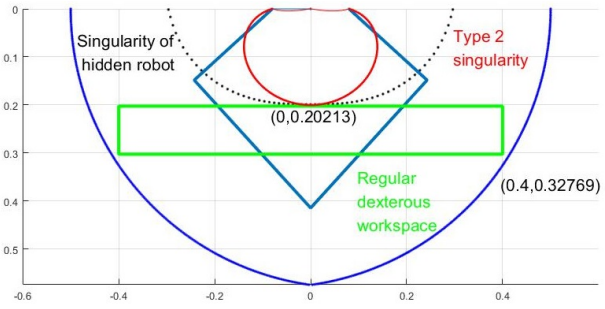

Figure 2.17: Regular dexterous workspace of the optimized Five-bar mechanism using leg-direction-based visual servoing ($15° \sim 165°$, $\pm$ 0.1 pixel)

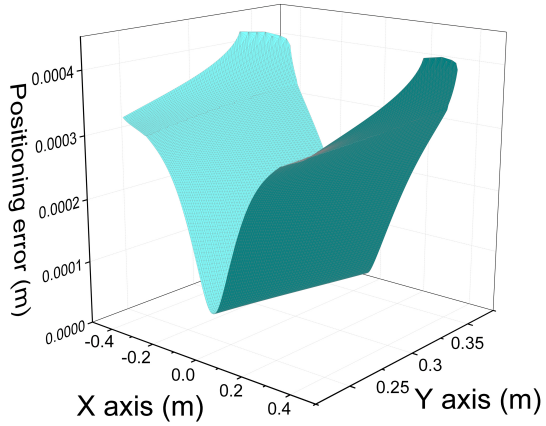

Figure 2.18: Positioning error in the regular dexterous workspace when using line-based visual servoing ($15° \sim 165°$, $\pm$ 1 pixel)

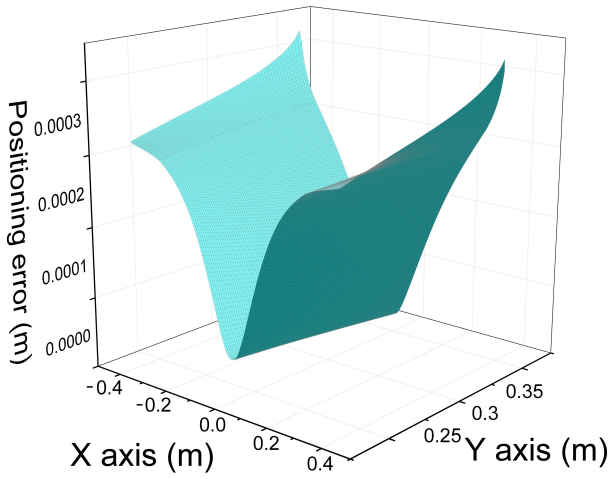

Figure 2.19: Positioning error in the regular dexterous workspace when using leg-direction-based visual servoing ($15° \sim 165°$, $\pm$ 1 pixel)

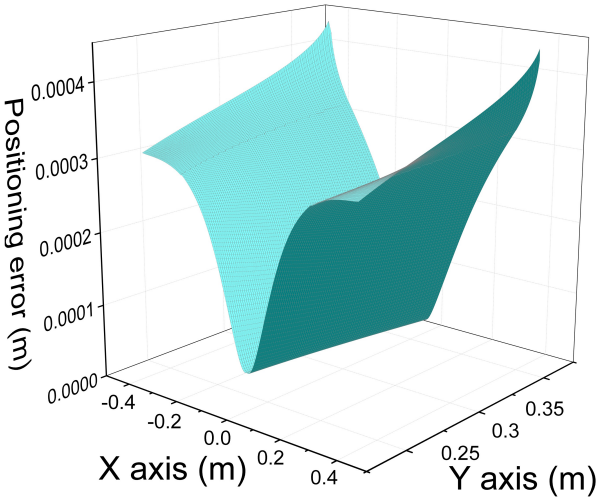

Figure 2.20: Positioning error in the regular dexterous workspace when using line-based visual servoing ($30° \sim 150°$, $\pm$ 1 pixel)

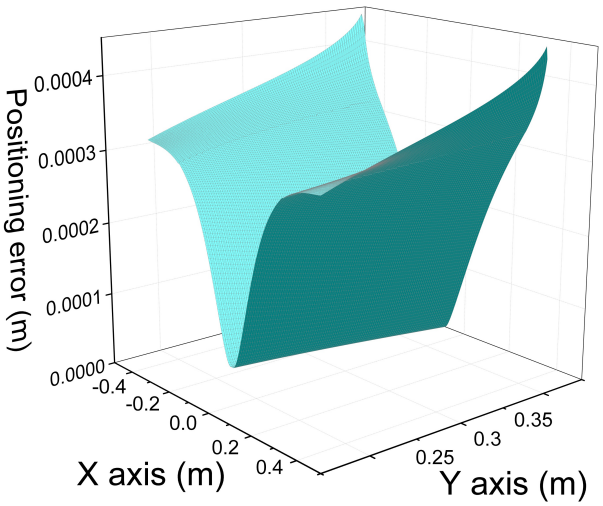

Figure 2.21: Positioning error in the regular dexterous workspace when using leg-direction-based visual servoing ($30° \sim 150°$, $\pm$ 1 pixel)

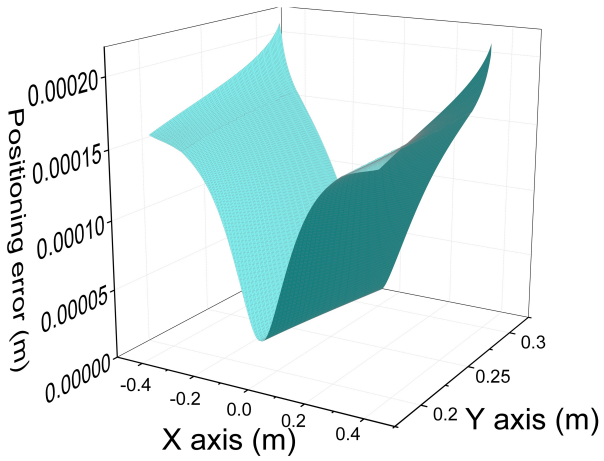

Figure 2.22: Positioning error in the regular dexterous workspace when using line-based visual servoing ($15° \sim 165°$, $\pm$ 0.5 pixel)

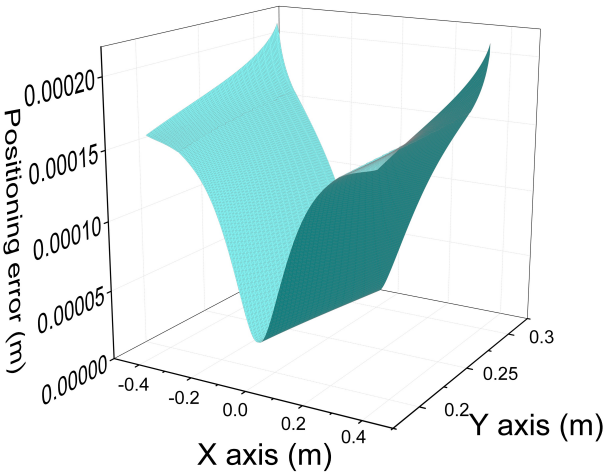

Figure 2.23: Positioning error in the regular dexterous workspace when using leg-direction-based visual servoing ($15° \sim 165°$, $\pm$ 0.5 pixel)

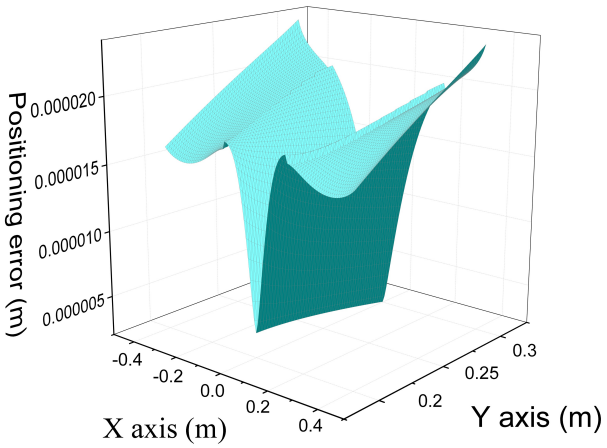

Figure 2.24: Positioning error in the regular dexterous workspace when using line-based visual servoing ($15° \sim 165°$, $\pm$ 0.1 pixel)

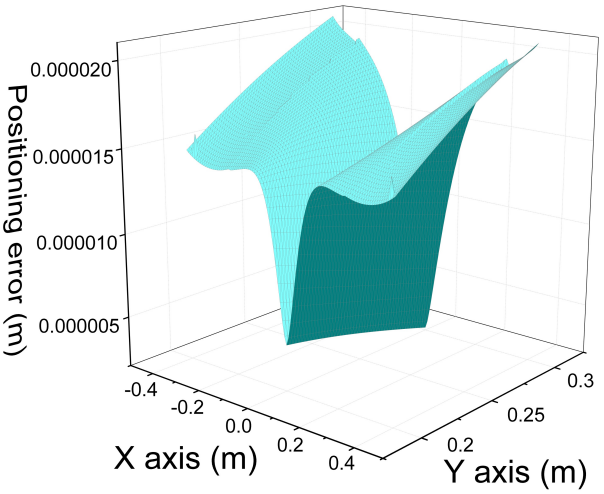

Figure 2.25: Positioning error in the regular dexterous workspace when using leg-direction-based visual servoing ($15° \sim 165°$, $\pm$ 0.1 pixel)

Table 2.2: Optimal design parameters and value of the objective function as a function of the chosen controller

	l_1[m]	l_2[m]	l_3[m]	y_c[m]	z_c[m]	A[m^2]
Line-based visual servoing ($15° \sim 165°$, $\pm$ 1 pixel)	0.2184	0.2290	0.3749	0.4340	0.5908	0.1144
Leg-direction-based visual servoing ($15° \sim 165°$, $\pm$ 1 pixel)	0.2190	0.2295	0.3888	0.4396	0.5889	0.1208
Line-based visual servoing ($30° \sim 150°$, $\pm$ 1 pixel)	0.2188	0.2168	0.4076	0.4340	0.5908	0.1216
Leg-direction-based visual servoing ($30° \sim 150°$, $\pm$ 1 pixel)	0.2201	0.2184	0.4096	0.4432	0.5953	0.1230
Line-based visual servoing ($15° \sim 165°$, $\pm$ 0.5 pixel)	0.1620	0.1988	0.3768	0.3746	0.6090	0.0951
Leg-direction-based visual servoing ($15° \sim 165°$, $\pm$ 0.5 pixel)	0.1650	0.1990	0.3904	0.4083	0.5880	0.1031
Line-based visual servoing ($15° \sim 165°$, $\pm$ 0.1 pixel)	0.1500	0.1902	0.3822	0.3523	0.5774	0.0950
Leg-direction-based visual servoing ($15° \sim 165°$, $\pm$ 0.1 pixel)	0.1600	0.2202	0.3684	0.3683	0.5780	0.1004

illustrated from Fig. 2.18 to 2.25. As we see from the figures, in the dexterous workspace, all the maximal positioning errors calculated are lower than 0.5 mm, which means that all the Five-bar mechanism optimal designs meet the accuracy constraint which is the most important criteria in our optimal design process.

2.6 Co-simulation of Five-bar mechanism

In terms of the robot size, the Five-bar mechanism designed based on the line-based visual servoing and leg-direction-based visual servoing are very close from each other, the difference is almost negligible. However, the positioning error model is very simplified and the validity of the optimisation need to be checked. The co-simulations with ADAMS and Simulink are needed to test the robot accuracy performance.

Simulations are performed in a connected ADAMS-Simulink environment, the scheme of the co-simulation is shown in Fig. 2.26. We created two Five-bar mechanism mechanical models using the results got from the line-based visual servoing ($15° \sim 165°$, $\pm$1 pixel)

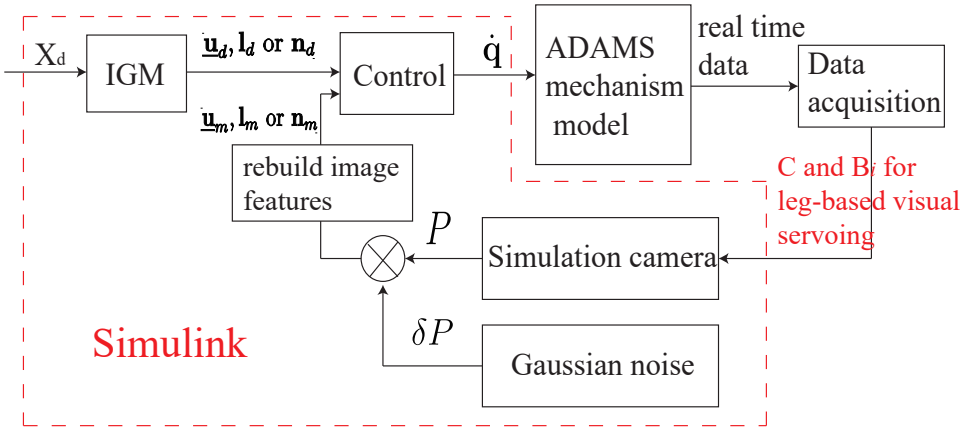

Figure 2.26: Co-simulation scheme between ADAMS and Simulink

optimal design and leg-direction-based visual servoing ($15° \sim 165°$ ± 1 pixel) optimal design in ADAMS. Real time data are the output of ADAMS and are sent to Simulink. In Simulink, the real time data are used to build the image features: leg direction ${}^c\underline{\mathbf{u}}_i$ in leg-direction-based visual servoing and Plücker coordinates $({}^c\underline{\mathbf{u}}_i, {}^c\mathbf{h}_i)$ in line-based visual servoing of the mechanical models. Then, the camera and the controller are simulated. The camera model in the simulation is a pinhole camera since it is easy to implement and is a good approximation of real cameras. As we see from the Fig. 2.27, the point O is the center of projection and the principal axis parallel to Z axis is shown. The distance between the center of projection and the image plane is the focal length f. For a 3D point P in space whose coordinate is (X, Y, Z), its projection on the image plane is $P_c = (u, v)$ in pixel coordinates. The parameters of the camera in the simulations are set to be:

- focal length with respect to $u = 10^3$;
- focal length with respect to $v = 10^3$;
- principle point in image along $u = 1024$ pixels;
- principle point in image along $v = 768$ pixels;

One-pixel noise (Gaussian noise in the simulation) is added on the intersection points of image boundary and the lines which are the projections of robot leg edges (Fig. 2.9), so that the controller accuracy performance can be checked. In the next step, the intersection points of the leg edges and image boundary are applied to rebuilding the image features $({}^c\underline{\mathbf{u}}_i$, $({}^c\underline{\mathbf{u}}_i, {}^c\mathbf{h}_i)$, $({}^c\underline{\mathbf{n}}_i^1, {}^c\underline{\mathbf{n}}_i^2))$ which will be used in the visual servoing. Each robot

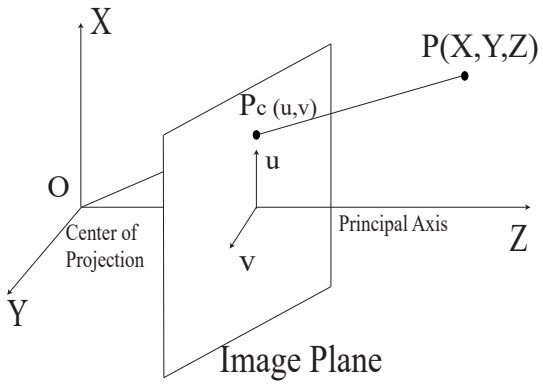

Figure 2.27: A pinhole camera model

is controlled with its dedicated visual servoing approach, the actuator velocities $\dot{\mathbf{q}}$ are the output of Simulink and are sent to ADAMS to drive the robot. In all these simulations, A home position and several desired positions within the dexterous workspace are set. Then the Five-bar mechanisms performs motions from their home positions to the desired positions and the positioning errors can be checked.

In the co-simulation, four controllers are applied on the control of the Five-bar mechanism.

- Controller 1 is a leg-direction-based controller which takes the vector ${}^c\underline{\mathbf{u}}_i$ as the feedback to do the visual servoing. Its definition is given in Section 1.2.2, equation (1.34).

- Controller 2 is a line-based controller which takes the Plücker coordinates $({}^c\underline{\mathbf{u}}_i, {}^c\,\mathbf{h}_i)$ as the feedback to do the visual servoing. Its definition is given in Section 1.2.3, equation (1.47).

- Controller 3 is a leg-direction-based controller which takes the vectors $({}^c\underline{\mathbf{n}}_i^1, {}^c\,\underline{\mathbf{n}}_i^2)$ as the feedback to do the visual servoing. Its definition is given in Section 1.2.2, equation (1.36).

- Controller 4 is a line-based controller which takes the vectors $({}^c\underline{\mathbf{n}}_i^1, {}^c\,\underline{\mathbf{n}}_i^2)$ as the feedback to do the visual servoing. Its definition is given in Section 1.2.3, equation (1.49).

Controller 1 and Controller 3 are applied on the control of the Five-bar mechanism mechanical model using the results got from the leg-direction-based visual servoing ($15^\circ \sim$

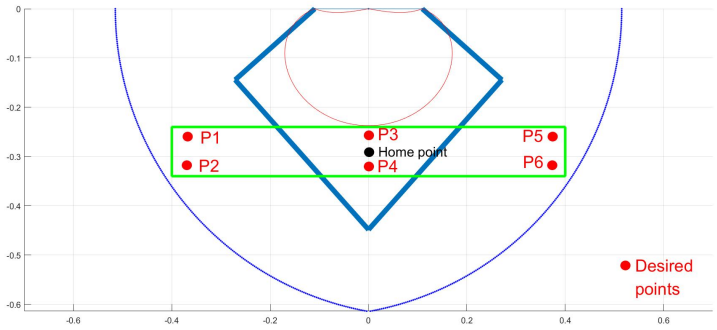

Figure 2.28: Desired points in the regular dexterous workspace

165°, ±1 pixel) optimal design. Controller 2 and Controller 4 are applied on the control of the Five-bar mechanism mechanical model using the results got from the line-based visual servoing (15° ∼ 165°, ±1 pixel) optimal design.

The regular dexterous workspace of the Five-bar mechanism is a rectangle whose length is 0.8 m and width is 0.1 m. The home point is set to be the center of the rectangular workspace and six points in the regular dexterous workspace are chosen to be the desired points (Fig. 2.28). We set the home point to be the original point of the workspace frame whose coordinate is $(0, 0)$ m (all the coordinates below in the co-simulation are with respect to the workspace frame). Then the coordinates of the desired points are:

- $\mathbf{P}_1 = (-0.35, -0.04)$ m
- $\mathbf{P}_2 = (-0.35, 0.04)$ m
- $\mathbf{P}_3 = (0, -0.04)$ m
- $\mathbf{P}_4 = (0, 0.04)$ m
- $\mathbf{P}_5 = (0.35, -0.04)$ m
- $\mathbf{P}_6 = (0.35, 0.04)$ m

The co-simulation results are given in Tab. 2.3. The simulation results of points $\mathbf{P}_1$, $\mathbf{P}_3$ and $\mathbf{P}4$ are illustrated from Fig. 2.29 to Fig. 2.40.

The simulation results in the Tab. 2.3 show that the maximal positioning errors for the leg-direction-based visual servoing Controller 1 are between 0.148 mm to 0.314 mm, the maximal positioning errors for the line-based visual servoing Controller 2 are between

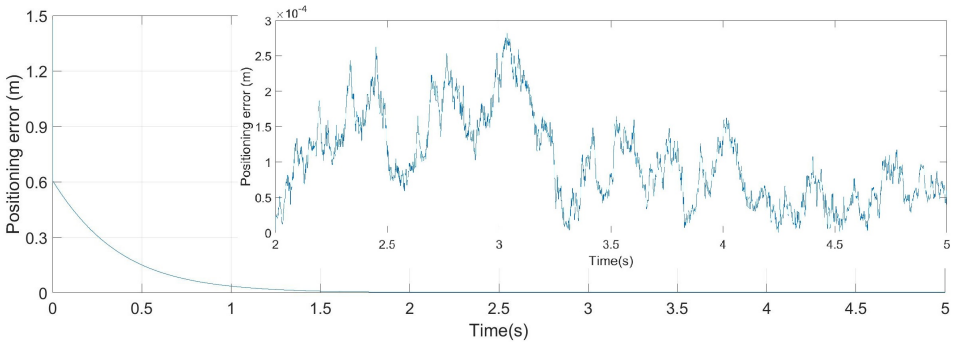

Figure 2.29: Positioning error at point $\mathbf{P}_1$ of the leg-direction-based design when applying Controller 1

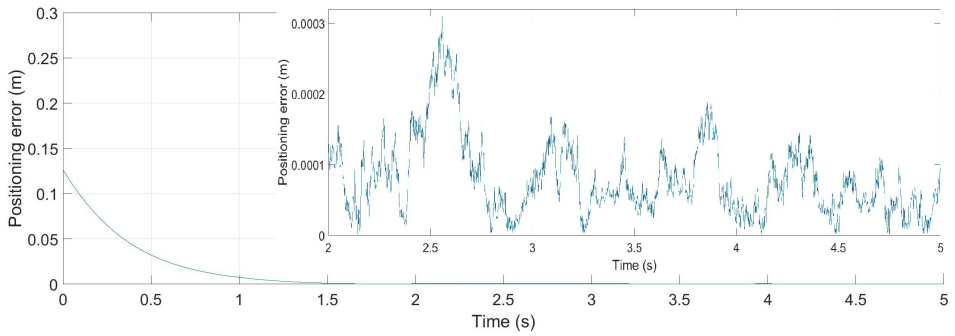

Figure 2.30: Positioning error at point $\mathbf{P}_3$ of the leg-direction-based design when applying Controller 1

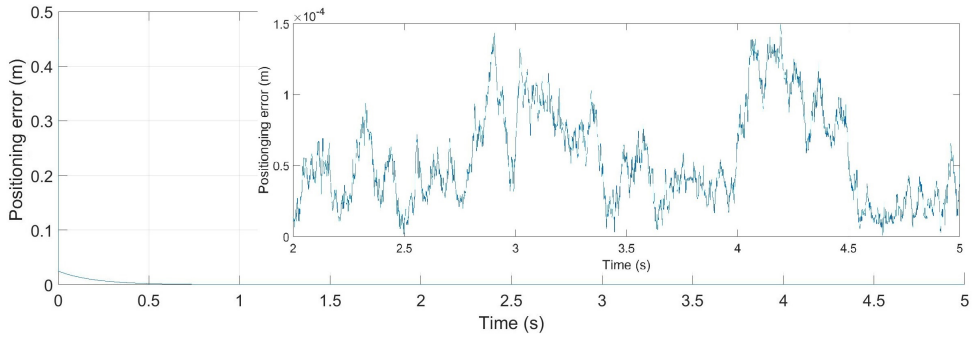

Figure 2.31: Positioning error at point $\mathbf{P}_4$ of the leg-direction-based design when applying Controller 1

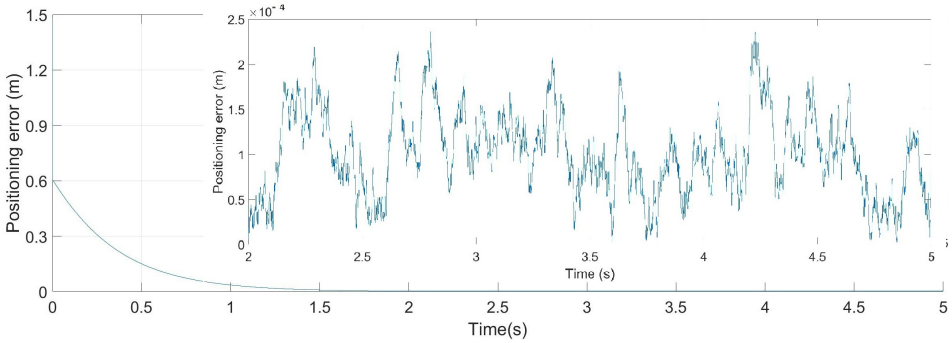

Figure 2.32: Positioning error at point $\mathbf{P}_1$ of the line-based design when applying Controller 2

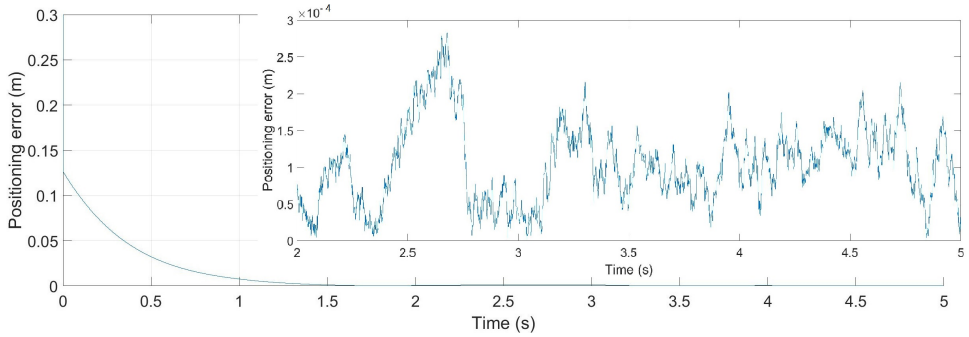

Figure 2.33: Positioning error at point $\mathbf{P}_3$ of the line-based design when applying Controller 2

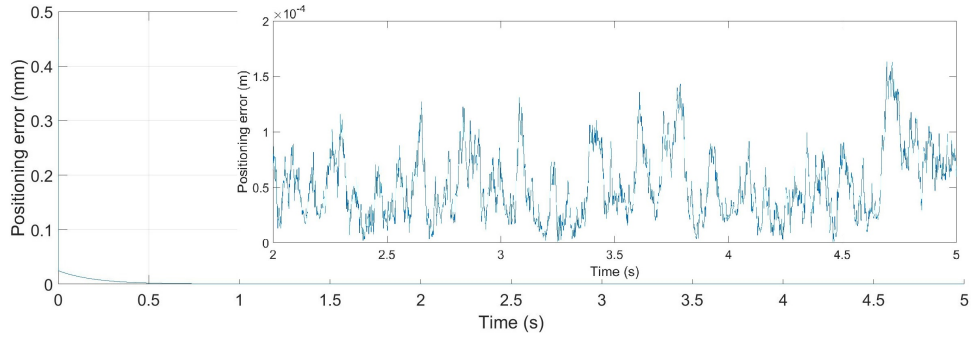

Figure 2.34: Positioning error at point $\mathbf{P}_4$ of the line-based design when applying Controller 2

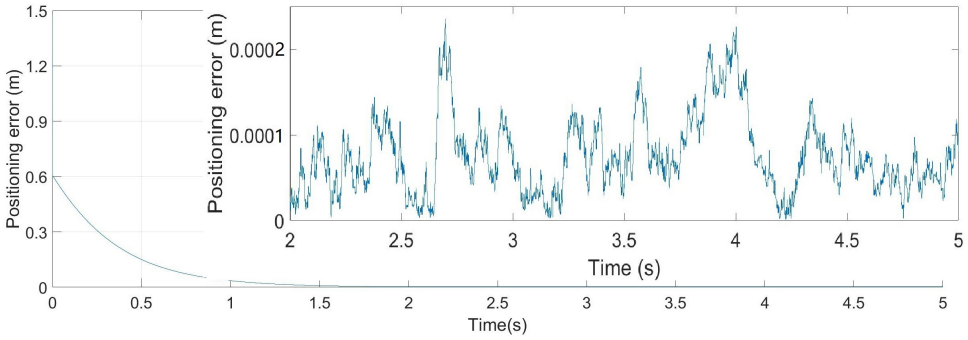

Figure 2.35: Positioning error at point $\mathbf{P}_1$ of the leg-direction-based design when applying Controller 3

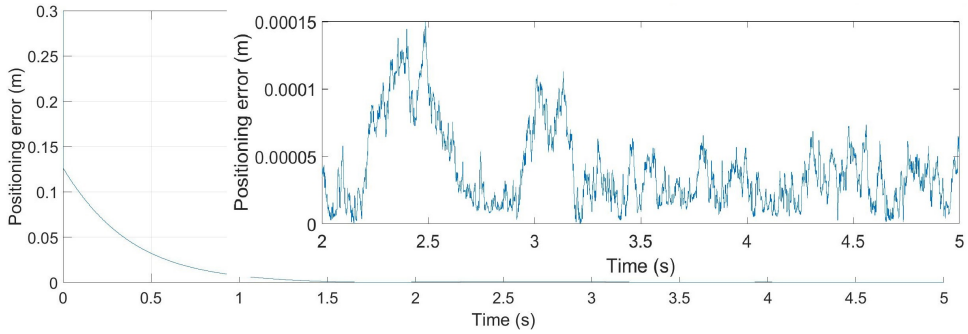

Figure 2.36: Positioning error at point $\mathbf{P}_3$ of the leg-direction-based design when applying Controller 3

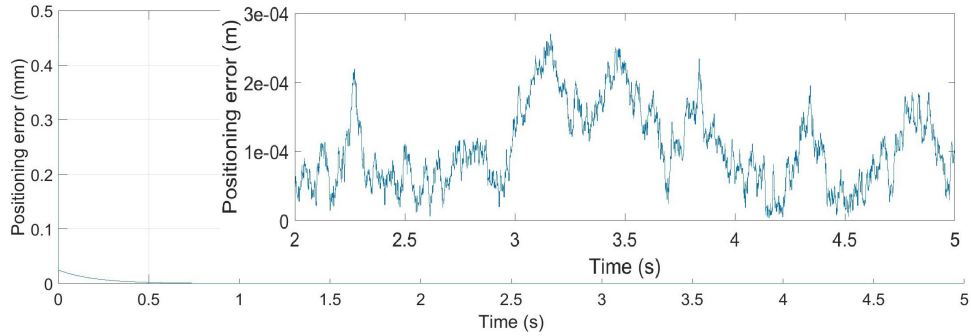

Figure 2.37: Positioning error at point $\mathbf{P}_4$ of the leg-direction-based design when applying Controller 3

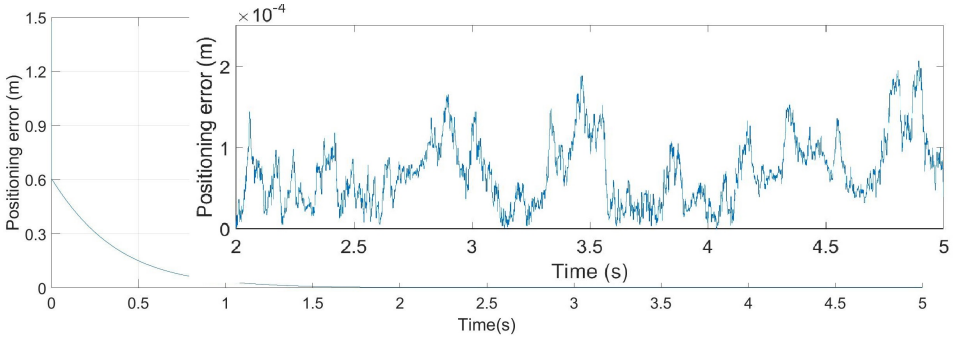

Figure 2.38: Positioning error at point $\mathbf{P}_1$ of the line-based design when applying Controller 4

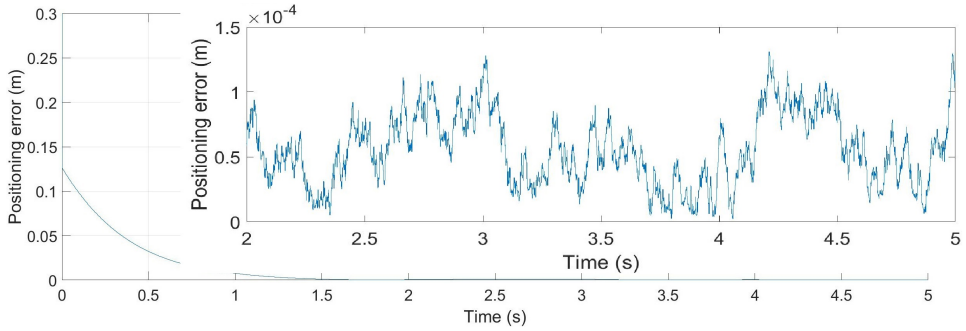

Figure 2.39: Positioning error at point $\mathbf{P}_3$ of the line-based design when applying Controller 4

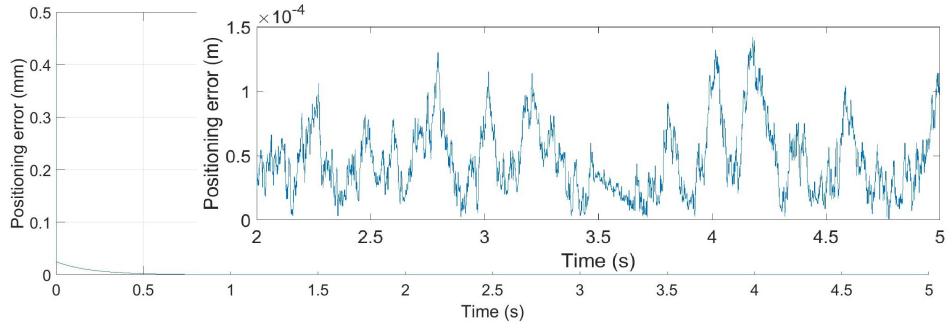

Figure 2.40: Positioning error at point $\mathbf{P}_4$ of the line-based design when applying Controller 4

Table 2.3: Simulation results of Leg-direction-based visual servoing and line-based visual servoing

Desired point	Max error of controller 1 [mm]	Max error of controller 2 [mm]	Max error of controller 3 [mm]	Max error of controller 4 [mm]
$\mathbf{P}_1$	0.271	0.238	0.231	0.211
$\mathbf{P}_2$	0.153	0.132	0.113	0.121
$\mathbf{P}_3$	0.314	0.275	0.149	0.132
$\mathbf{P}_4$	0.148	0.161	0.238	0.145
$\mathbf{P}_5$	0.299	0.256	0.237	0.153
$\mathbf{P}_6$	0.188	0.183	0.202	0.189

0.132 mm to 0.275 mm, the maximal positioning errors for the leg-direction-based visual servoing Controller 3 are between 0.113 mm to 0.238 mm, the maximal positioning errors for the line-based visual servoing Controller 4 are between 0.121 mm to 0.211 mm. We see from these simulation results that the optimized robots using the classical Controller 1 and Controller 2 have a maximal accuracy of lower than 0.5 mm for all points tested in their regular dexterous workspace. When applying the Controller 3 and Controller 4, the maximal accuracy is lower than 0.238 mm for all points tested in their regular dexterous workspace. Controller 3 and Controller 4 have better control accuracy compared with Controller 1 and Controller 2, which means that controlling directly the leg-direction vector $\underline{\mathbf{u}}$ and the Plücker coordinates $({}^c\underline{\mathbf{u}}_i, {}^c\mathbf{h}_i)$ have worse accuracy than controlling the vector $({}^c\underline{\mathbf{n}}_i^1, {}^c\underline{\mathbf{n}}_i^2)$. When using line-based visual servoing, the controller accuracy is better than leg-direction-based controller, but the difference of accuracy is not significant. Moreover, the accuracy performance is worse when the desired points are located near the singularities of the hidden robot.

Based on the results above, for Controller 3 and Controller 4, the difference of accuracy is not significant, we cant draw the conclusion that which controller is better for the control of the Five-bar mechanism. However, in reality, the points near the controller singularity may be included in the workspace of the Five-bar mechanism. Then four points near both the singularity of hidden robot of Controller 3 and Controller 4 are chosen to be the desired points in the co-simulation (Fig. 2.41). The coordinates of the desired points (with respect to the base frame) are:

- $\mathbf{P}_7$=(-0.18,0.14) m

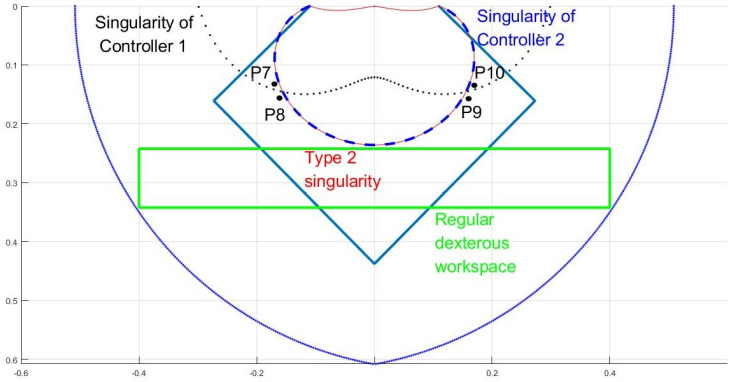

Figure 2.41: Four desired points near both the singularity of hidden robot of Controller 3 and Controller 4

Table 2.4: Simulation results of Leg-direction-based visual servoing and line-based visual servoing

Desired point	Max error of controller 3 [mm]	Max error of controller 4 [mm]
$\mathbf{P}_7$	11.1	0.238
$\mathbf{P}_8$	6.15	0.289
$\mathbf{P}_9$	10.0	0.375
$\mathbf{P}_{10}$	14.8	0.442

- $\mathbf{P}_8$=(-0.17,0.16) m
- $\mathbf{P}_9$=(0.17,0.16) m
- $\mathbf{P}_{10}$=(0.18,0.14) m

The results of the co-simulation are given in Tab. 2.4 and Fig. 2.42 to 2.45. From the results, we see that the positioning errors of Controller 3 are much bigger than the positioning errors of Controller 4, which means that the singularity of the hidden robot of the Controller 4 is much more robust to measurement noise.

2.7 Conclusion

In this chapter, control-based design of a Five-bar mechanism was performed in order to obtain the best accuracy performance of the robot with its controller. The proposed

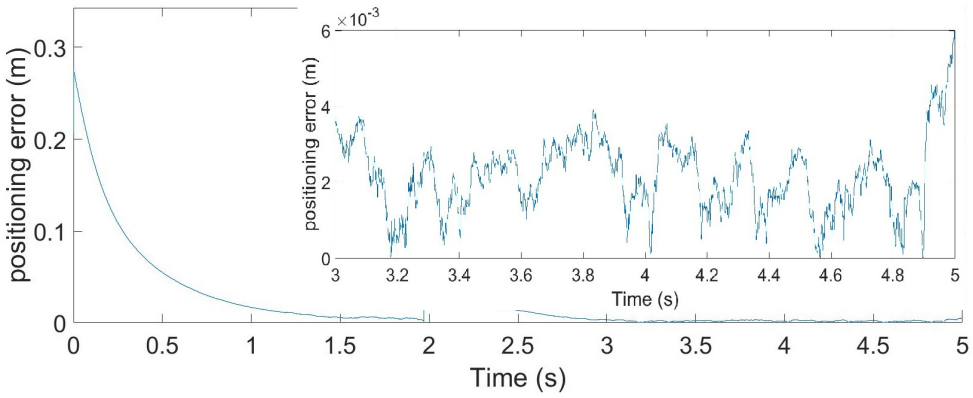

Figure 2.42: Positioning error at point $\mathbf{P}_8$ of the leg-direction-based design when applying Controller 3

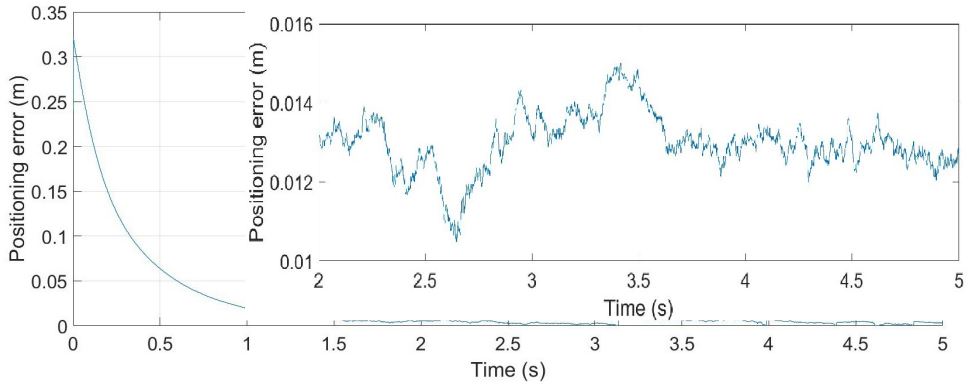

Figure 2.43: Positioning error at point $\mathbf{P}_{10}$ of the leg-direction-based design when applying Controller 3

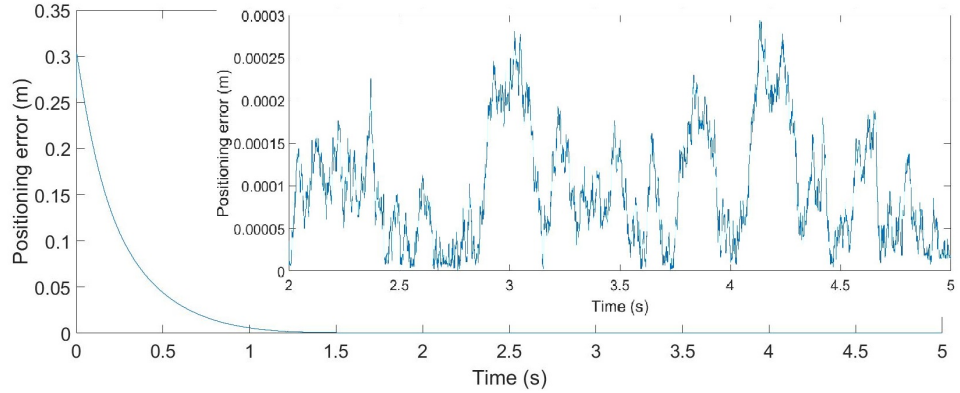

Figure 2.44: Positioning error at point $\mathbf{P}_8$ of the line-based design when applying Controller 4

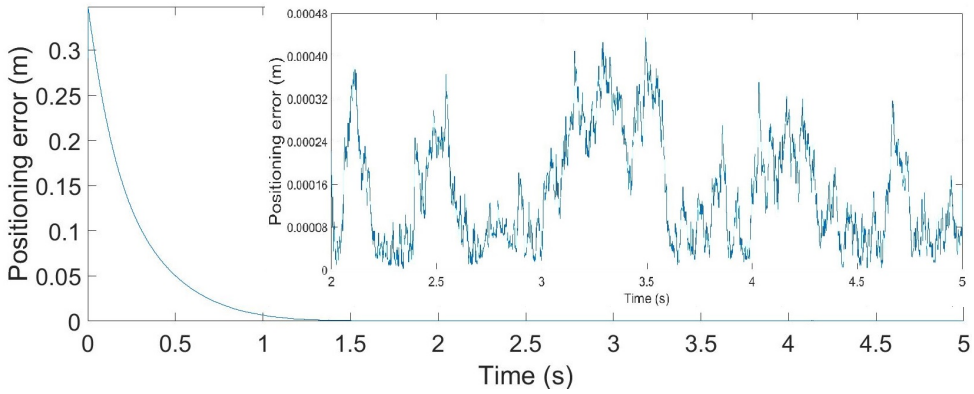

Figure 2.45: Positioning error at point $\mathbf{P}_{10}$ of the line-based design when applying Controller 4

control-based design methodology took into account the performance of the controller in the design process to get the optimal geometric parameters of robot. We optimized the design of Five-bar mechanism for two different types of controllers: leg-direction-based visual servoing and line-based visual servoing. Based on these two controllers, we developed positioning error models taking into account the error of observation coming from the camera. We also analyzed the singularities of these controllers to be sure that no singularity of the controller appeared in the final design of the robot, thus avoiding instability issues. Then, design optimization problems have been formulated in order to find the optimal geometric parameters and camera placement for the Five-bar mechanism for each type of controller. Co-simulations between ADAMS and Simulink were finished to test the accuracy performance of four controllers. The line-based visual servoing controller which controls the leg edges shows a better control performance with respect to the accuracy based on the co-simulation, experimental works on real prototypes should be done to verify this simulation results in the future.

Chapter 3

Optimal design of a DELTA robot

In this chapter, a control-based design methodology is applied in order to create a DELTA robot with the best accuracy performance for a dedicated controller. Three types of controllers are envisaged for the control of the motions of the DELTA robot: leg-direction-based visual servoing, line-based visual servoing and image moment visual servoing. Firstly, the design specifications, the singularities and the vision-based kinematic of the DELTA robot are presented. Then based on these three controllers proposed, positioning error models taking into account the error of observation coming from the camera and the controller singularities are detailed. In the next step, design optimization problems are formulated in order to find the optimal geometric parameters and camera placement for the DELTA robot for each type of controller. Co-simulations of the robots optimized for the three types of controllers are performed in order to check the accuracy performance of the three robots. Finally, the conclusions are drawn.

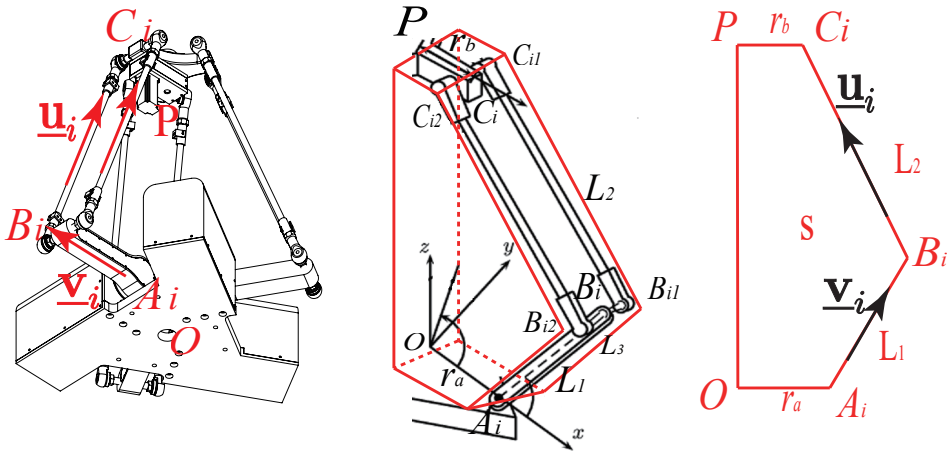

Figure 3.1: DELTA robot mechanism

3.1 Kinematics and design specifications of the DELTA robot

The DELTA robot is a parallel manipulator with three translational degrees of freedom (DOF). In this case, the 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 parallel kinematic chains. Its architecture is illustrated in Fig. 3.1 (In the left is the CAD model of DELTA robot, in the middle is the architecture of one of its three chains, in the right is the left view of the chain). Each chain contains a rotational joint actuated by a motor located at A_i $(i = 1, 2, 3)$ in the base platform. Then the motion is transmitted to the mobile platform through spatial parallelograms $B_{i1}B_{i2}C_{i2}C_{i1}$ $(i = 1, 2, 3)$.

P is the center of the moving platform of the DELTA robot, then the geometric model can be written as [Laribi 2007]

$$X_P = \cos\theta_i(r_a + L_1\cos\varphi_{1i} + L_2\cos\varphi_{3i}\cos(\varphi_{1i} + \varphi_{2i}) - r_b) - L_2\sin\theta_i\sin\varphi_{3i} \tag{3.1}$$

$$Y_P = \sin\theta_i(r_a + L_1\cos\varphi_{1i} + L_2\cos\varphi_{3i}\cos(\varphi_{1i} + \varphi_{2i}) - r_b) - L_2\cos\theta_i\sin\varphi_{3i} \tag{3.2}$$

$$Z_P = L_1\sin\varphi_{1i} + L_2\cos\varphi_{3i}\sin(\varphi_{1i} + \varphi_{2i}) \tag{3.3}$$

Where $i = 1, 2, 3$ and $[X_P, Y_P, Z_P]$ are the coordinates of the point P with respect to the global frame as is shown in Fig. 3.1. The definitions of θ_i, φ_{1i}, φ_{2i}, φ_{3i} are given in Fig. 3.2.

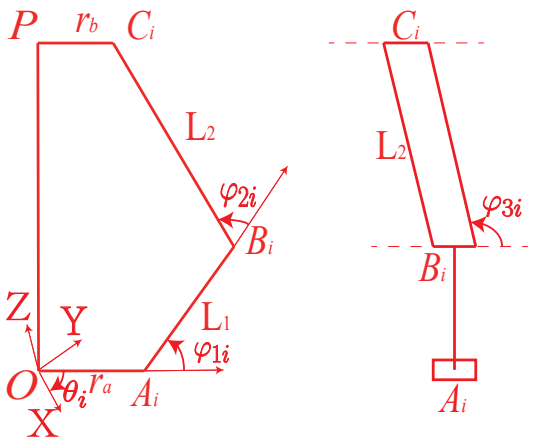

Figure 3.2: DELTA robot parameters

The direct geometric model aims at determining the location of the point P= $[X_P, Y_P, Z_P]$ for given joint angles $\varphi_{1i}, \varphi_{2i}, \varphi_{3i}$ $(i = 1, 2, 3)$. Then the coordinates of P is given by the simultaneous solution of the three equations $(i = 1, 2, 3)$

$$(X_P - X_i)^2 + (Y_P - Y_i)^2 + (Z_P - Z_i)^2 = L_2^2 \tag{3.4}$$

where

$$\begin{cases} X_i = r + L_1 \cos \varphi_{1i} \\ Y_i = r + L_1 \cos \varphi_{1i} \\ Z_i = -L_1 \sin \varphi_{1i} \end{cases} \tag{3.5}$$

with $r = r_a - r_b$.

Each individual equation in (3.4) defines a sphere whose center is $[X_i, Y_i, Z_i]$ and with radius L_2. Then for given joint angles, there are two possible solutions of this system of equations, which means that the moving platform have two possible configurations with respect to the base, as is shown in Fig. 3.3 [Pierrot 1990].

Details of the inverse geometric model of the DELTA robot are presented in [Laribi 2007] and [Goudali 1995].

The workspace of the DELTA robot is a volume in space defined by $n_i^2 - (l_i^2 + m_i^2) \leqslant 0$ whose boundary is $n_i^2 - (l_i^2 + m_i^2) = 0$.

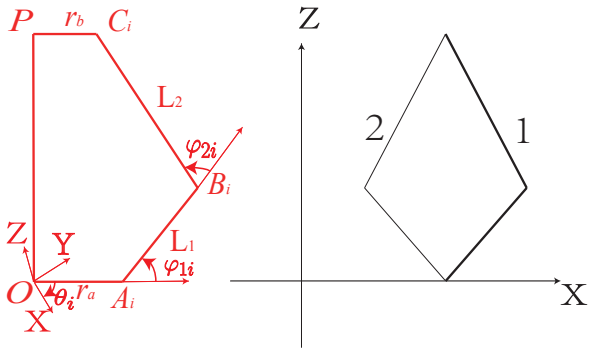

Figure 3.3: Two possible configurations of the DELTA robot for given joint angles

where

$$\begin{cases} l_i = 2rL_1 - 2L_1X_P\cos_{\theta_i} - 2L_1Y_P\sin\theta_i \\ m_i = 2L_1Z_P \\ n_i = -2rX_P\cos\theta_i - 2rY_P\sin\theta_i + X_P^2 + r^2 + L_1^2 + Z_P^2 + Y_P^2 - L_1^2 \end{cases} \tag{3.6}$$

To get the inverse kinematics of the DELTA robot, we can compute from the following loop-closure equations (Fig. 3.1) [Briot 2016]

$$\mathbf{C}_i - \mathbf{B}_i = L_2\underline{\mathbf{u}_i} \tag{3.7}$$

where

$$\mathbf{B}_i = \mathbf{A}_i + L_1[\cos\varphi_{1i}\ 0\ \ \sin\varphi_{1i}]^T = \mathbf{A}_i + L_1\underline{\mathbf{v}_i} \tag{3.8}$$

$$\mathbf{C}_i = \mathbf{P} + \overrightarrow{PC_i} \tag{3.9}$$

Squaring both sides of (3.7) and introducing (3.8), then we have

$$(X_{A_iC_i} - L_1\cos\varphi_{1i})^2 + Y_{A_iC_i}^2 + (Z_{A_iC_i} - L_1\sin\varphi_{1i})^2 - L_2^2 = 0 \tag{3.10}$$

where $C_i - A_i = [X_{A_iC_i}, Y_{A_iC_i}, Z_{A_iC_i}]^T$. Replacing $\cos\varphi_{1i}$ by $(1-t_i^2)/(1+t_i^2)$ and $\sin\varphi_{1i}$ by $2t_i/(1+t_i^2)$, where $t_i = \tan(\varphi_{1i}/2)$. Then equation (3.10) becomes

$$\varphi_{1i} = 2\tan^{-1}\left(\frac{-\beta_i \pm \sqrt{\alpha_i^2 + \beta_i^2 - \gamma_i^2}}{\gamma_i - \alpha_i}\right) \tag{3.11}$$

where

$$\begin{cases} \alpha_i = -2L_1 X_{A_iC_i} \\ \beta_i = -2L_1 Z_{A_iC_i} \\ \gamma_i = X_{A_iC_i}^2 + Y_{A_iC_i}^2 + Z_{A_iC_i}^2 + L_1^2 - L_2^2 \end{cases} \tag{3.12}$$

we define the platform translational velocity $\boldsymbol{\tau}_p = [v_x, v_y, v_z]^T$ and actuator velocities $\dot{\mathbf{q}} = [\dot{\varphi_{1i}}, \dot{\varphi_{2i}}, \dot{\varphi_{3i}}]^T$. Differentiating (3.10) with respect to time and we have

$$\mathbf{A}\boldsymbol{\tau}_p + \mathbf{B}\dot{\mathbf{q}} = 0 \tag{3.13}$$

where

$$\mathbf{A} = L_2 \begin{bmatrix} \underline{\mathbf{u}}_1^T \\ \underline{\mathbf{u}}_2^T \\ \underline{\mathbf{u}}_3^T \end{bmatrix} \tag{3.14}$$

$$\mathbf{B} = L_1 L_2 \begin{bmatrix} \underline{\mathbf{u}}_1^T \underline{\mathbf{v}}_1^{\perp} & 0 & 0 \\ 0 & \underline{\mathbf{u}}_2^T \underline{\mathbf{v}}_2^{\perp} & 0 \\ 0 & 0 & \underline{\mathbf{u}}_3^T \underline{\mathbf{v}}_3^{\perp} \end{bmatrix}, \quad \underline{\mathbf{v}}_i^{\perp} = [-\sin\varphi_{1i} \ 0 \ \cos\varphi_{1i}]^T \quad (i = 1, 2, 3) \tag{3.15}$$

As a result, we have

$$\dot{\mathbf{q}} = -\mathbf{B}^{-1}\mathbf{A}\boldsymbol{\tau} = \mathbf{J}_{inv}\boldsymbol{\tau}_p \tag{3.16}$$

where $\mathbf{J}_{inv}$ is the inverse Jacobian matrix of the DELTA robot. or

$$\boldsymbol{\tau}_p = -\mathbf{A}^{-1}\mathbf{B}\dot{\mathbf{q}} = \mathbf{J}\dot{\mathbf{q}} \tag{3.17}$$

where $\mathbf{J}$ is the Jacobian matrix of the DELTA robot.

From equation (3.15), we see that the Type 1 singularity of DELTA robot occurs when $\underline{\mathbf{u}}_i$ and $\underline{\mathbf{v}}_i$ are parallel, which is the situation that one chain of the DELTA robot is fully stretched or folded (the same as the reachable workspace of DELTA robot). For the Type 2 singularity of DELTA robot, $\mathbf{A}$ is rank deficient when [Romdhane 2002]

$$\det(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3) = 0 \Rightarrow \underline{\mathbf{u}}_1 \cdot (\underline{\mathbf{u}}_2 \times \underline{\mathbf{u}}_3) = 0 \tag{3.18}$$

which means that the DELTA robot comes to a singularity configuration when the set of the vectors $(\underline{\mathbf{u}}_1, \underline{\mathbf{u}}_2, \underline{\mathbf{u}}_3)$ is linearly dependent. All the possible cases of singularities of DELTA robot are presented here:

- two of the three unit vectors are equal, for example $\underline{\mathbf{u}}_1 = \underline{\mathbf{u}}_2$, hence the set $(\underline{\mathbf{u}}_1, \underline{\mathbf{u}}_2, \underline{\mathbf{u}}_3)$

Table 3.1: Requirements of the DELTA robot

Regular dexterous workspace size (radius of cylinder) r_0	$\geqslant$ 200 mm
Regular dexterous workspace size (height of cylinder) h_0	$\geqslant$100 mm
Positioning accuracy wherever in regular dexterous workspace	$\leqslant$0.5 mm

is of rank 2.

- all the three vectors are equal, $\underline{\mathbf{u}}_1 = \underline{\mathbf{u}}_2 = \underline{\mathbf{u}}_3$, hence the set $(\underline{\mathbf{u}}_1, \underline{\mathbf{u}}_2, \underline{\mathbf{u}}_3)$ is of rank 1.

- the vector $\underline{\mathbf{u}}_1, \underline{\mathbf{u}}_2, \underline{\mathbf{u}}_3$ are coplanar, hence the set $(\underline{\mathbf{u}}_1, \underline{\mathbf{u}}_2, \underline{\mathbf{u}}_3)$ is of rank 2.

All these singularity cases should be avoided in the dexterous workspace of the DELTA robot in the optimal design process.

The requirements that must be achieved by the DELTA robot in this work are given in Tab. 3.1. The DELTA robot should cover a regular workspace. Some geometric and kinematic constraints should be satisfied throughout the regular workspace, such as within the reachable workspace of DELTA robot, be free of singularity, has a better positioning accuracy compared to the requirements given in Tab. 3.1, thus obtaining a regular dexterous workspace (Fig. 3.4) [Germain 2013]. The shape of regular dexterous workspace is a cylinder in this work and its desired radius and height are given in Tab. 3.1. For practical industrial reasons, in the condition of getting a regular dexterous workspace larger than the size given in Tab. 3.1, the robot must be as compact as possible.

In order to get the desired 0.5 mm of positioning accuracy specified in Tab. 3.1, we propose to apply visual servoing approaches. We plan to use a single camera (2336×1728 pixels of resolution and a focal length of 10 mm) mounted onto the base platform in order to control the DELTA robot. Three types of standard visual servoing approaches will be tested: leg-direction-based visual servoing, line-based visual servoing [Vignolo 2014] and image moment visual servoing [Chaumette 2004]. All these three visual servoing controllers will be applied to the control of a DELTA robot in order to determine which control method is the best adapted to its accuracy performance.

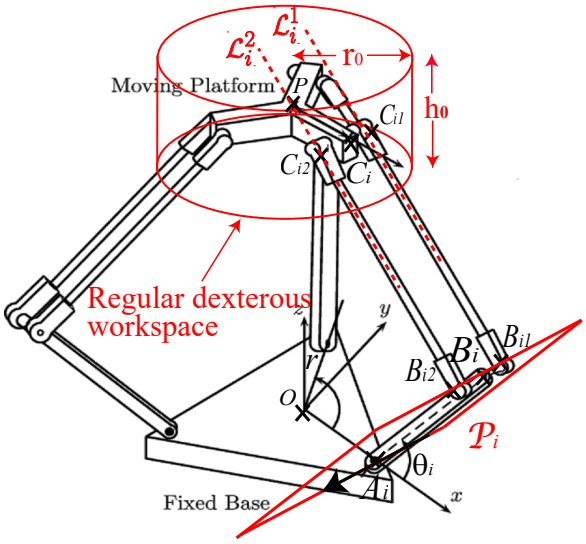

Figure 3.4: Desired workspace of the DELTA robot

3.2 Vision-based kinematic of DELTA robot

Vision-based kinematic model of DELTA robot aims at finding the relationship between the twist of the moving platform and the time variation of the image features, which is the interaction matrix in the visual servoing.

3.2.1 Vision-based kinematics of a DELTA robot using the leg-direction-based visual servoing

Leg-direction-based visual servoing developed in [Andreff 2005] is applied on the control of a DELTA robot. The leg direction $\underline{\mathbf{u}}_i$ extracted from the observation of the robot leg (Fig. 3.1) is selected as the feature to do the visual servoing. For a DELTA robot, the vector $\underline{\mathbf{u}}_i$ can be calculated directly from

$$\underline{\mathbf{u}}_i = (\mathbf{C}_i - \mathbf{B}_i)/L_2 \tag{3.19}$$

Introducing (3.8) into (3.19) and differentiating (3.19) with respect to time, we have

$$\dot{\underline{\mathbf{u}}}_i = (\boldsymbol{\tau_p} - L_1 \underline{\mathbf{v}}_i^{\perp} \dot{\mathbf{q}}_i)/L_2 \tag{3.20}$$

where $\boldsymbol{\tau}_p = [v_x, v_y, v_z]^T$ is the twist of the moving platform and $\dot{\mathbf{q}} = [\dot{\varphi_{1i}}, \dot{\varphi_{2i}}, \dot{\varphi_{3i}}]^T$ are the actuator velocities.

Finally, with the help of (3.16), we have

$$\dot{\underline{\mathbf{u}}}_i = ((\mathbf{I}_3 + L_1 \underline{\mathbf{v}}_i^{\perp} \mathbf{a}_i / b_i)/L_2)\boldsymbol{\tau}_p = \mathbf{M}_{ui}^T \boldsymbol{\tau}_p \tag{3.21}$$

with $\mathbf{a}_i = L_2 \underline{\mathbf{u}}_i^T$ and $b_i = L_1 L_2 \underline{\mathbf{u}}_i^T \underline{\mathbf{v}}_i^{\perp}$, $\mathbf{I}_3$ is the 3×3 identity matrix and matrix $\mathbf{M}_{ui}^T$ is the interaction matrix related to the leg direction $\underline{\mathbf{u}}_i$.

The matrix $\mathbf{M}_{ui}^T$ is of rank 2 [Andreff 2005]. Therefore, to fully control the end-effector pose of the DELTA robot, observing a minimum of two independent legs is necessary. Then we have the interaction matrix $\mathbf{M}_u^T$ by stacking the matrices $\mathbf{M}_{ui}^T$ of k legs ($k = 2 \cdots 6$). As the choice of robot legs will affect the positioning accuracy [Rosenzveig 2013], to make sure to get the best control accuracy, it is better to use observation redundancy. In this case, we observe all the six distal links of DELTA robot.

The leg-direction-based visual servoing controller applied in this case is defined in Section 1.2.2, equation (1.34).

3.2.2 Vision-based kinematics of a DELTA robot using the line-based visual servoing

Line-based visual servoing kinematic aims finding the relationship between the time variation of the Plücker coordinates $(\underline{\mathbf{u}}_i, \mathbf{h}_i)$ of the robot legs and the twist of its platform.

From the definition of the Plücker coordinates, we have $\mathbf{h}_i = \mathbf{D} \times \underline{\mathbf{u}}_i$ where $\mathbf{D}$ is the position of any point P on the line. Then we have

$$\dot{\mathbf{h}}_i = \dot{\mathbf{D}} \times \underline{\mathbf{u}}_i + \mathbf{D} \times \dot{\underline{\mathbf{u}}}_i \tag{3.22}$$

See Fig. 3.4, take the line $\mathcal{L}_i^1$ as an example, the equation (3.22) becomes

$$\dot{\mathbf{h}}_i^1 = \dot{\mathbf{C}}_{i1} \times \underline{\mathbf{u}}_i^1 + \mathbf{C}_{i1} \times \dot{\underline{\mathbf{u}}}_i^1 = \boldsymbol{\tau}_p \times \underline{\mathbf{u}}_i^1 + \mathbf{C}_{i1} \times \dot{\underline{\mathbf{u}}}_i^1 = (\left[-\underline{\mathbf{u}}_i^1\right]_\times + \left[\mathbf{C}_{i1}\right]_\times \mathbf{M}_{ui1}^T)\boldsymbol{\tau}_p \tag{3.23}$$

where $[\cdots]_\times$ is the antisymetric matrix associated to a 3D vector [Martinet 1996], $\mathbf{M}_{ui1}^T$ is the interaction matrix related to the vector $\underline{\mathbf{u}}_i^1$. Writing Eq. (3.23) in the matrix form, we have

$$\dot{\mathbf{h}}_i^1 = \mathbf{M}_{hi1}^T \boldsymbol{\tau}_p \tag{3.24}$$

Therefore, for a line $\mathcal{L}_i^1$, we have

$$\begin{bmatrix} \dot{\underline{\mathbf{u}}}_i^1 \\ \dot{\mathbf{h}}_i^1 \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{ui1}^T \\ \mathbf{M}_{hi1}^T \end{bmatrix} \boldsymbol{\tau}_p = \mathbf{M}_{uhi1}^T \boldsymbol{\tau}_p \tag{3.25}$$

For line-based visual servoing, to fully control the end-effector pose of the DELTA robot, observing a minimum of two independent legs is necessary. Then we have the interaction matrix $\mathbf{M}_{uh}^T$ by stacking the matrices $\mathbf{M}_{uhij}^T$ of k legs $((i = 1, 2, 3),\ (j = 1, 2),\ (k = 2 \cdots 6))$.

The line-based visual servoing controller applied in this case is defined in Section 1.2.3, equation (1.47).

3.2.3 Vision-based kinematics of a DELTA robot using the image moment visual servoing

As it was presented in Section 1.2.4, for image moment visual servoing, a picture E fixed on the end-effector of the parallel robot is the target to be observed and the image moment is got from the image R acquired from the camera.

The end-effector of DELTA robot only translates along the three axes of the space with respect to the base. Similarly as it was done in [Chaumette 2004], the shape of the picture observed is chosen to be an ellipse, three visual features used to control the translational DOFs have been selected: the coordinates x_g, y_g of the center of gravity of R $(x_g(= m_{10}/m_{00}),\ y_g(= m_{01}/m_{00}))$, and the area a of the picture in image plane $(a = m_{00})$. Since we fix the picture on the end-effector of DELTA robot which has no rotation, so we only consider the translation velocities of the object E denoted as $\boldsymbol{\tau}_p = [v_x, v_y, v_z]^T$. The image plane is set to be parallel to the planar object similarly as it was done in [Chaumette 2004], so that the picture R is centered and horizontal in the image and the interaction matrix has a nice decoupling form, then we obtain

$$\boldsymbol{\tau}_p = \mathbf{L}_m^+ \dot{\mathbf{m}} \tag{3.26}$$

Where $\mathbf{L}_m$ is got by stacking the matrix $\mathbf{L}_{m_{ij}}$ and $\dot{\mathbf{m}} = \left[\dot{x}_g, \dot{y}_g, \dot{a}\right]^T$ are the time derivatives

of three features observed. Thus, as shown in [Chaumette 2004]

$$\mathbf{L}_m = \begin{bmatrix} -C & 0 & Cx_g \\ 0 & -C & Cy_g \\ 0 & 0 & 2aC \end{bmatrix} \tag{3.27}$$

Where $C = \frac{1}{Z}$, Z being the depth between the camera and the object, and a is the area of the object in the image.

The image moment visual servoing controller applied in this case is defined in Section 1.2.4, equation (1.70).

3.3 Positioning error models of DELTA robot

When leg-direction-based visual servoing and line-based visual servoing are considered, the positioning error models is the same as the models presented in Section 2.4. However, when image moment visual servoing is considered, the positioning error model is different.

Similarly as it was done in [Chaumette 2004], the shape of the picture observed is chosen to be an ellipse (Fig. 3.5). For image moment visual servoing, the positioning error comes from the observation error. As it is presented in Section 3.2, the observation features are the coordinates of the center of gravity and the area of the object in the image plane. So we add a noise which is a random shift in the pixels on the coordinates of object center of gravity in image plane and on the coordinates of the top points of major and minor axes of the ellipse (see Fig. 3.5). Since the error of observation is very small, the error model relating the variation of end-effector $\boldsymbol{\delta}\mathbf{x}$ to the variation $\boldsymbol{\delta}\mathbf{m}$ of the moments can be approximated by using the first-order geometric model based on the use of the interaction matrix given in formula (3.26):

$$\boldsymbol{\delta}\mathbf{x} = \mathbf{L}_m^+ \boldsymbol{\delta}\mathbf{m} \tag{3.28}$$

In this case, $\boldsymbol{\delta}\mathbf{m} = [\delta x_g, \delta y_g, \delta a]^T$. For a point projected in the image plane, knowing the camera resolution, taken as ± 1 pixel, so $\delta x_g = \pm 1$, $\delta y_g = \pm 1$ and $\delta a = \pi[({}^p r_1 \pm 1)({}^p r_2 \pm 1) - {}^p r_1 {}^p r_2]$ (${}^p r_1$,${}^p r_2$ are radii of the major and minor axes of ellipse projected to image plane in pixel as is shown in Fig. 3.5).

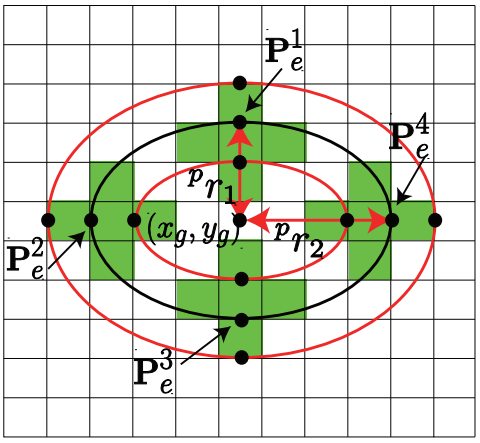

Figure 3.5: One-pixel error on the intersection top points of the ellipse

3.4 Controller singularity and hidden robot of DELTA robot

As presented in the Section 1.2.5, the controller singularity appears when the interaction matrix is rank deficient. In [Briot 2013], a tool named "hidden robot" was proposed to simplify the complex analysis of the singularity of the mapping involved into the controller (see more details in Section 1.2.5).

Singularities of leg-direction-based visual servoing applied to the control of the DELTA robot have been studied in [Rosenzveig 2013] and [Briot 2016]. The virtual equivalent leg involved in the visual servoing of the DELTA robot using leg-direction-based visual servoing has a $\Pi(2-\underline{U}U)$ architecture, (where Π stands for a planar parallelogram linkage joint, see Fig. 3.6). This Π joint keeps constant the orientation of the rod $B_{i3}B_{i4}$ (Fig. 3.6) and guarantees that, the first U joint (the link the planar and spatial parallelograms) are not moving when the vector $\underline{\mathbf{u}}_i$ is set at constant value. Then, observing the direction of the DELTA robot remains not to control the displacement of a $\underline{R}(2-SS)$ leg but of a virtual $\Pi(2-\underline{U}U)$ leg with the same geometric properties as the real leg ($\left\|\overrightarrow{A_iB_i}\right\| = L_1$, $\left\|\overrightarrow{B_iC_i}\right\| = L_2$ in Fig. 3.6). It can be demonstrated that a $2-\Pi(2-\underline{U}U)$ robot (Fig. 3.7) is fully-actuated. Therefore, it can be explained that it is possible to control the DELTA robot by observing the displacements of two of its three legs.

For the $2-\Pi(2-\underline{U}U)$ robot, Type 1 singularities when $\overrightarrow{A_iB_i}$ and $\overrightarrow{B_iC_i}$ are colinear. In this configuration, the robot reaches its workspace boundary. Types 2 singularities appear when the planes $\mathcal{P}_i$ (containing points A_i and B_i and the axis of the active revolute joint

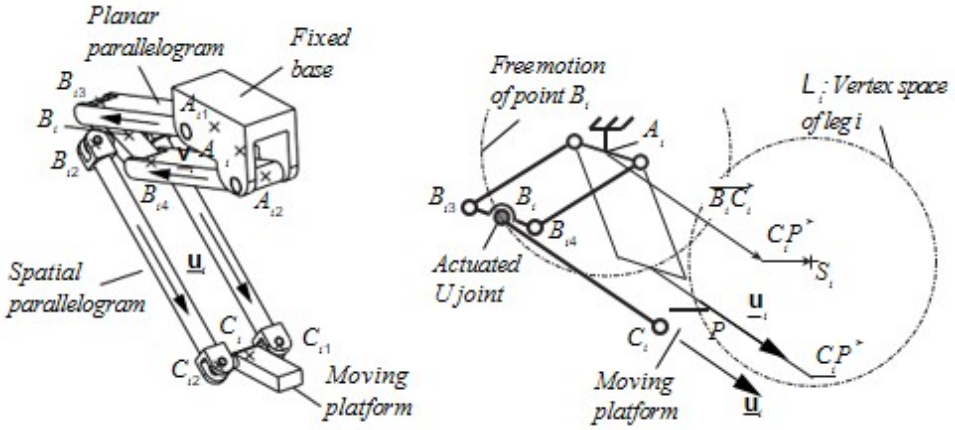

Figure 3.6: A 2-Π $(2 - \underline{U}U)$ leg kinematic description and the vertex space for a given vector $\underline{\mathbf{u}}_i$ [Rosenzveig 2013]

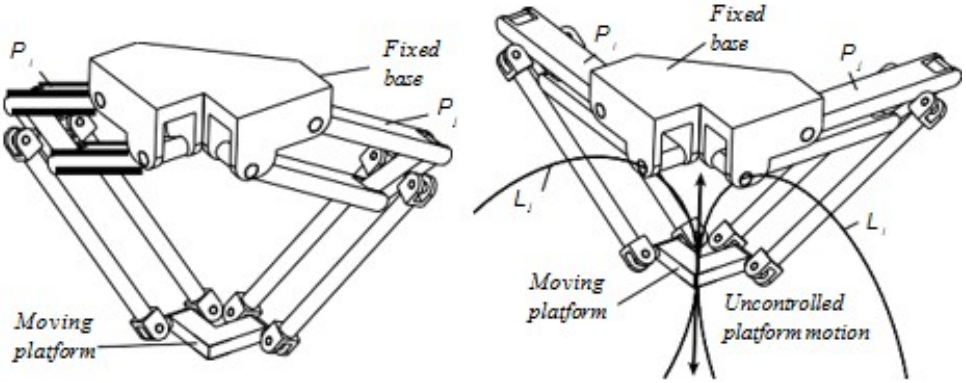

Figure 3.7: A $2 - \Pi(2 - \underline{U}U)$ robot and its singularity configuration when $\mathcal{P}_i \parallel \mathcal{P}_j$ [Rosenzveig 2013]

located at point A_i) and $\mathcal{P}_j$ (see Fig. 3.7) are parallel.

Singularities of the line-based visual servoing are different. As known from [Merlet 2006], singularities of kinematic models are also singularities of the pose estimation models. When observing the six lines $\mathcal{L}_{ij}$ passing through the links $B_{ij}C_{ij}$ $(i = 1, 2, 3,\ j = 1, 2,$ Fig. 3.4) 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 find the common intersection point of the six lines $\mathcal{L}'_{ij}$ obtained from lines $\mathcal{L}_{ij}$ by a translation of vectors $\overrightarrow{C_{ij}P}$. The only condition of degeneracy of this pose estimation model is when all lines $\mathcal{L}'_{ij}$ are parallel, which means that the controller singularity of line-based visual servoing for the DELTA robot will appear when all its distal links will be parallel.

From (3.27), we see that the interaction matrix $\mathbf{L}_m$ 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 feature E is lying on the camera center (thus meaning that the feature E cannot be observed anymore). Thus, these singularity cases will never happen in the image moment visual servoing of the DELTA robot.

3.5 Optimal design process of DELTA robot

We assume that all the three kinematic chains of DELTA robot are identical in length, then the DELTA robot can be defined by the following geometric parameters: L_1, L_2, L_3, r_a, r_b as we see from Fig. 3.1 ($L_1 = L_{A_iB_i}$, $L_2 = L_{B_iC_i}$, $L_3 = L_{B_{i1}B_{i2}} = L_{C_{i1}C_{i2}}$, $r_a = L_{OA_i}$, $r_b = L_{PC_i}$). When leg-direction-based visual servoing and line-based visual servoing are applied, the radius of the cylindrical distal links of robot also affects the positioning error [Kaci 2018], contrary to the optimization of the Five-bar mechanism, here the radius of the cylindrical distal links $B_{ij}C_{ij}$ ($i = 1, 2, 3, j = 1, 2$, Fig. 3.1), denoted as R' (see Fig. 1.18), is a decision variable of the optimization process. (x_c, y_c, z_c) define the position of the camera with respect to the robot frame. The camera image plane is set to be parallel to the end-effector of the DELTA robot, and the coordinates (x_c, y_c) of the camera frame origin at set at $(0, 0)$ so that all legs can be observed in a symmetrical way. r_1 and r_2 are the radii (in world frame) defining the ellipse in image moment visual servoing. r_0 and h_0 are the dimensions of radius and height of the cylindrical dexterous workspace (Fig. 3.4) in which the following performance must be satisfied:

- Type 2 singularity-free: ensuring the DELTA robot will not meet any Type 2 singularities.
- controller singularity-free: no singularities of the controllers
- end-effector in image: ensuring that all the robot distal legs can be observed when using leg-based visual servoing, as well as the ellipse shape picture on the end-effector when using image moment visual servoing.
- required positioning accuracy: Maximal positioning error computed with the models of Section 3.3 should be lower than 0.5 mm.
- no link collision in the workspace.

Since the three chains of DELTA robot are identical in length, we study the bounding box of a single chain of the DELTA robot in order to study its footprint. This bounding box is shown in Fig. 3.1 (the red part). The objective function amounts to the volume $V = S \cdot L_3$ of the bounding box in Fig. 3.1 when the link A_iB_i and B_iC_i are perpendicular, where S is the lateral surface of the box. The objective is to minimize the volume V as this will allow the definition of a compact robot.

In order to create a compact DELTA robot which has the specifications detailed in Tab. 3.1, the following optimization problem is formulated:

$$\begin{aligned}
\text{minimize} \quad & V = S \cdot L_3 \\
& (S = ((r_a + r_b) \cdot \sqrt{L_1^2 + L_2^2 - (r_a - r_b)^2} + L_1 \cdot L_2)/2) \\
\text{over} \quad & \mathbf{x} = [r_a, r_b, L_1, L_2, L_3, z_c, R']^T \\
& \text{(leg-direction-based and line-based visual servoing)} \\
& \mathbf{x} = [r_a, r_b, L_1, L_2, L_3, z_c, r_1, r_2]^T \\
& \text{(image moment visual servoing)} \\
\text{subject to} \quad & r_0 \geqslant 200 \text{ mm}, \ h_0 \geqslant 100 \text{ mm} \\
& 0 \leqslant r_b \leqslant 0.1 \text{ m} \leqslant r_a \leqslant 0.4 \text{ m}, \ 0.4 \text{ m} \leqslant L_1 \leqslant 0.8 \text{ m}, \\
& 0.2 \text{ m} \leqslant L_2 \leqslant 0.6 \text{ m}, \ 0.08 \text{ m} \leqslant L_3 \leqslant 0.16 \text{ m}, \\
& 0 \leqslant z_c \leqslant 0.15 \text{ m}, 0 \leqslant R' \leqslant 0.04 \text{ m}, \\
& 0 \leqslant r_2 \leqslant r_1 \leqslant 0.04 \text{ m}
\end{aligned} \tag{3.29}$$

In order to compute the size of the maximal dexterous workspace [Merlet 2006], we use the algorithm presented in [Germain 2013]. The previous optimization algorithm is applied for the design of the mentioned DELTA robot and the optimal design results are given in Tab. 3.2. For each controller, based on the error models given by equations (2.23), (2.24) and (3.28), the positioning error in the dexterous cylindrical workspace can be computed. Then the results are illustrated in Figs. 3.8 to 3.10, for example in Fig. 3.9, $z = 0.4$ m and $z = 0.5$ m correspond to the two bottoms of the workspace with respect to the robot frame in the case of line-based visual servoing.

As we see from the figures, in the dexterous workspace of each DELTA robot, all the maximal positioning errors calculated are less than 0.5 mm, which means that all DELTA robot optimal designs meet the accuracy constraint which is the most important criteria in our optimal design process.

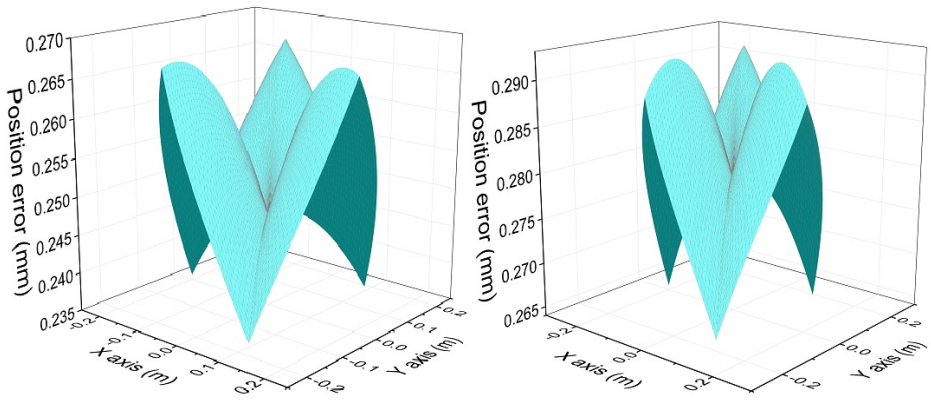

Figure 3.8: Positioning error in (mm) for $z = 0.78$ m (left) and $z = 0.88$ m (right) (with respect to robot frame) of DELTA robot designed by leg-direction-based visual servoing

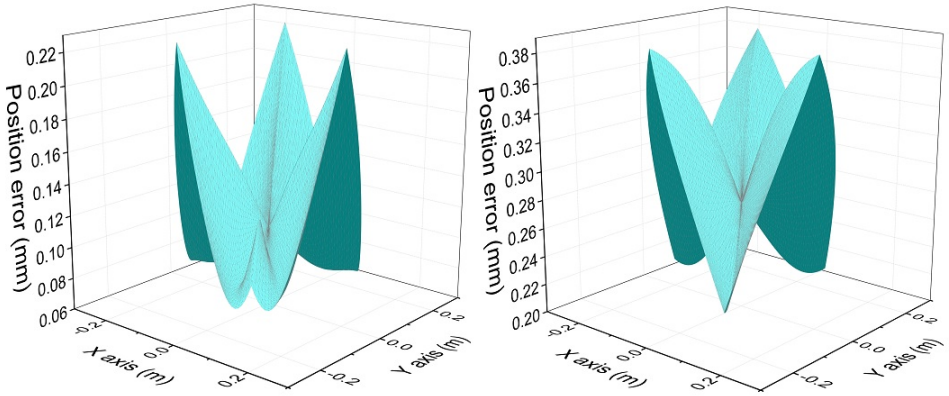

Figure 3.9: Positioning error in (mm) for $z = 0.4$ m (left) and $z = 0.5$ m (right) (with respect to robot frame)of DELTA robot designed by line-based visual servoing

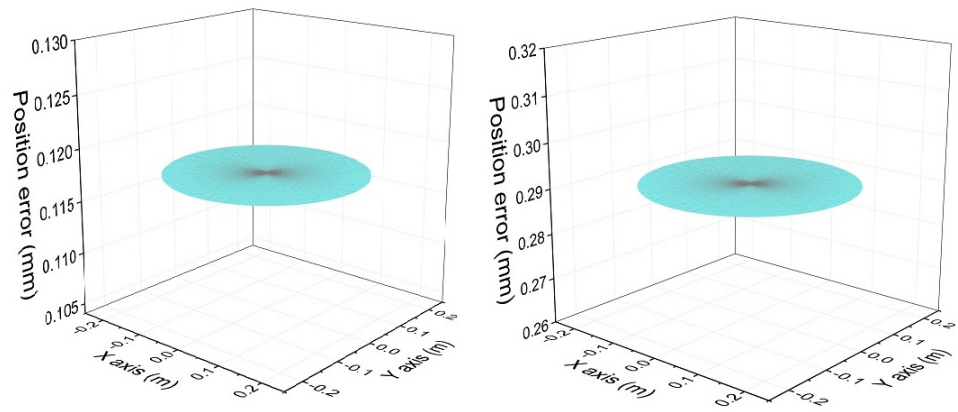

Figure 3.10: Positioning error in (mm) for $z = 0.4$ m (left) and $z = 0.5$ m (right) (with respect to robot frame) of DELTA robot designed by image moment visual servoing

Table 3.2: 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

In terms of the robot size, the DELTA robot designed based on the line-based visual servoing and image moment visual servoing are very close from each other, the difference is almost negligible while the size of DELTA robot designed by leg-direction-based visual servoing is more than twice bigger. In the following section, we perform co-simulations with ADAMS and Simulink to test the robot accuracy performance.

3.6 Co-simulation of DELTA robot

Co-simulations are performed in a connected ADAMS-Simulink environment. The control scheme is similar with the scheme presented in Section 2.6. We created three DELTA robot mechanical models using the results got from the optimal design (one model per controller). Real time data (for leg-direction-based visual servoing simulation and line-based visual servoing simulation, the real-time data are the coordinates of the points $\mathbf{C}_i$ and $\mathbf{B}_i$ (Fig. 3.1). For image moment visual servoing simulation, the real-time data are the coordinates of centroid point of the ellipse shape picture E and two top points on the major and minor axes of the ellipse) of the mechanical models are the output of ADAMS and are sent to Simulink to rebuild the image features (leg-directions, Plücker coordinates and image moments) used in visual servoing. In Simulink, the pinhole camera is created (see more details in Section 2.6). For leg-direction-based visual servoing, the controller

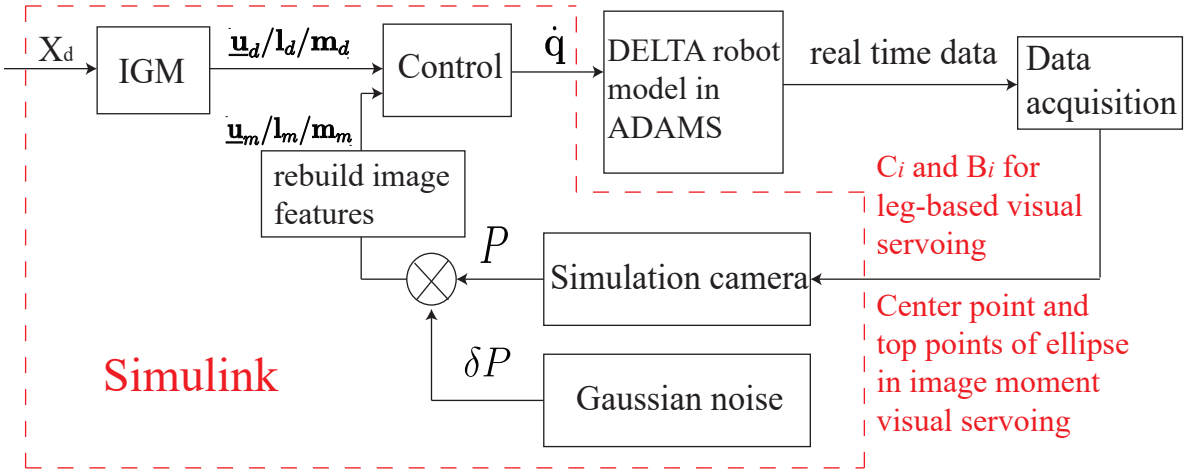

Figure 3.11: Co-simulation control scheme of DELTA robot

is defined in Section 1.2.2, equation (1.34). For line-based visual servoing, the controller is defined in Section 1.2.3, equation (1.47). For image moment visual servoing, the controller is defined in Section 1.2.4, equation (1.70). Then one-pixel noise is added into the camera simulation: for the case of leg-direction-based visual servoing and line-based visual servoing, the noise is added on the intersection points of image boundary and the lines which are the projections of robot leg edges (see Fig. 2.9), for the case of image moment visual servoing, the noise is added on the centroid point and two top points of the ellipse (see Fig. 3.5), so that controller accuracy performance can be checked. Each robot is controlled with its dedicated visual servoing approach.

In all these simulations, we set a home position and several desired positions within the dexterous workspace. Then DELTA robots perform motions from their home position to the desired positions and we check their positioning error. The regular dexterous workspace of the DELTA robot is a cylinder whose radius is 0.2 m and height is 0.1 m. The home point is set to be the center of the cylindrical workspace. The results from Figs. 3.8 to 3.10 show that the positioning errors of DELTA robot are symmetrical because of the symmetrical structure of DELTA robot and the errors grow at the edge of the workspace. twenty one points from the edge of the cylinder are chosen to be the desired points and the simulation results of these points are enough to see the controller performances (Fig. 3.12). We set the home point to be the original point of the workspace frame whose coordinate is $(0, 0, 0)$ m (all the coordinates below in the co-simulation are with respect to the workspace frame). Then the coordinates of the desired points are given in Tab. 3.3.

We start with the DELTA robot designed for line-based visual servoing. When the

Table 3.3: Desired points in the co-simulation of DELTA robot

Point	Coordinate [m]	Point	Coordinate [m]	Point	Coordinate [m]
$\mathbf{P}_1$	(0.2,0,-0.05)	$\mathbf{P}_9$	(0.2,0,0)	$\mathbf{P}_{17}$	(0.2,0,0.05)
$\mathbf{P}_2$	(0.14,0.14,-0.05)	$\mathbf{P}_{10}$	(0.14,0.14,0)	$\mathbf{P}_{18}$	(0.14,0.14,0.05)
$\mathbf{P}_3$	(0,0.2,-0.05)	$\mathbf{P}_{11}$	(0,0.2,0)	$\mathbf{P}_{19}$	(0,0.2,0.05)
$\mathbf{P}_4$	(-0.14,0.14,-0.05)	$\mathbf{P}_{12}$	(-0.14,0.14,0)	$\mathbf{P}_{20}$	(-0.14,0.14,0.05)
$\mathbf{P}_5$	(-0.2,0,-0.05)	$\mathbf{P}_{13}$	(-0.2,0,0)	$\mathbf{P}_{21}$	(-0.2,0,0.05)
$\mathbf{P}_6$	(-0.14,-0.14,-0.05)	$\mathbf{P}_{14}$	(-0.14,-0.14,0)	$\mathbf{P}_{22}$	(-0.14,-0.14,0.05)
$\mathbf{P}_7$	(0,-0.2,-0.05)	$\mathbf{P}_{15}$	(0,-0.2,0)	$\mathbf{P}_{23}$	(0,-0.2,0.05)
$\mathbf{P}_8$	(0.14,-0.14,-0.05)	$\mathbf{P}_{16}$	(0.14,-0.14,0)	$\mathbf{P}_{24}$	(0.14,-0.14,0.05)

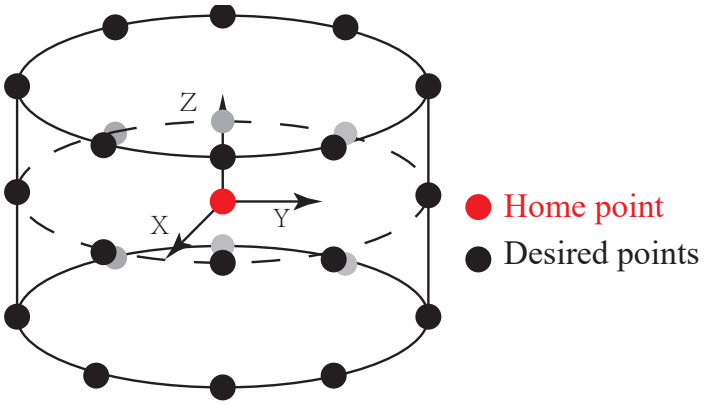

Figure 3.12: Desired points in the regular dexterous workspace for DELTA robot co-simulation

desired point to attain is $\mathbf{P}_{10}$, the simulation results are presented in Fig. 3.13. When desired point is $\mathbf{P}_{18}$, the simulation result is shown in Fig. 3.14. As is shown in Figs. 3.13 and 3.14, final positioning error is less than 0.5 mm. The same simulation experiments are done for the other desired points and the results are shown in Tab. 3.4. The simulation results in Tab. 3.4 show that the maximal positioning errors for the line-based controller are between 0.361 mm to 0.558 mm.

For DELTA robot optimized for image moment visual servoing, we do the same operations as what we did in the line-based visual servoing simulation experiments. When the desired point is $\mathbf{P}_{10}$, the simulation results are presented in Fig. 3.15. When the desired point is moved to $\mathbf{P}_{18}$, the simulation results are shown in Fig. 3.16. Maximal positioning errors for these two desired points are lower than 0.3 mm. For the other desired points, the results of image moment visual servoing simulation as shown in Tab. 3.5. From the

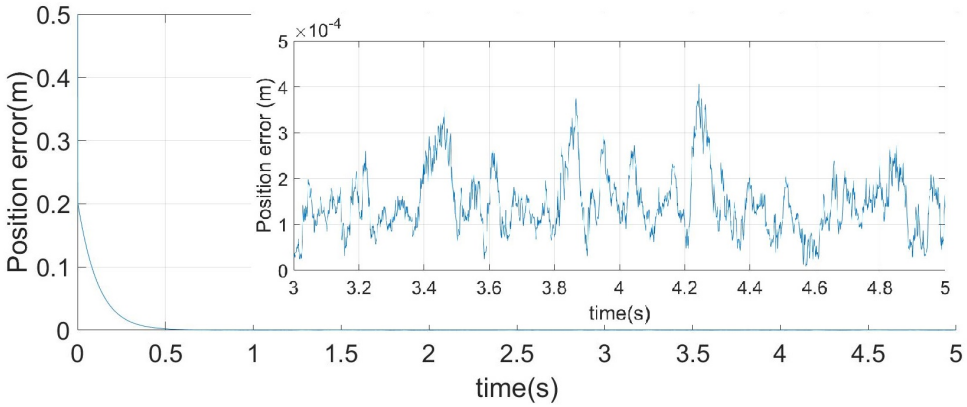

Figure 3.13: Positioning error at point $\mathbf{P}_{10}$ of the line-based optimal design

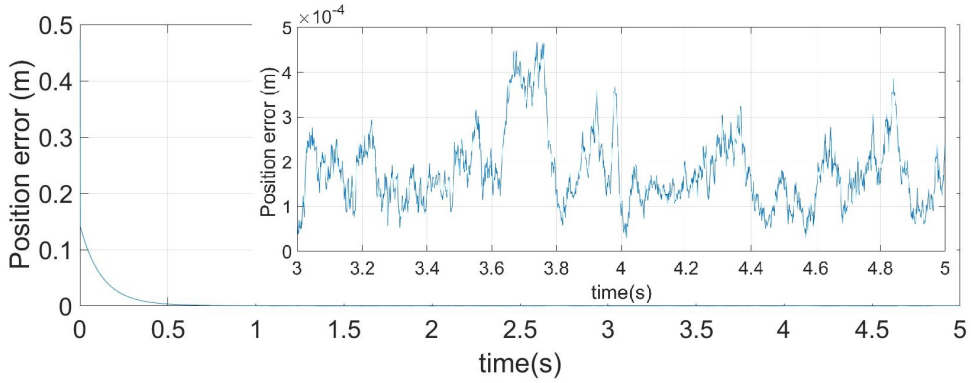

Figure 3.14: Positioning error at point $\mathbf{P}_{18}$ of the line-based optimized design

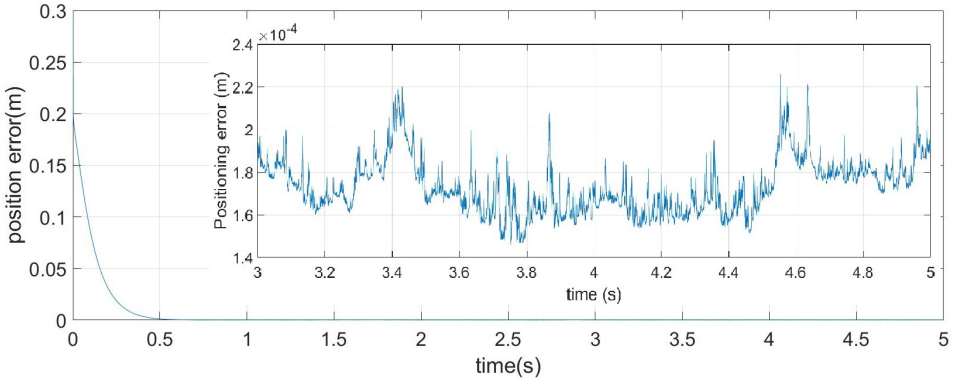

Figure 3.15: Positioning error at point $\mathbf{P}_{10}$ of the image moment optimized design

Table 3.4: DELTA robot simulation result of line-based visual servoing

Point	Max error[mm]	Point	Max error[mm]	Point	Max error[mm]
$\mathbf{P}_1$	0.361	$\mathbf{P}_9$	0.379	$\mathbf{P}_{17}$	0.431
$\mathbf{P}_2$	0.463	$\mathbf{P}_{10}$	0.401	$\mathbf{P}_{18}$	0.467
$\mathbf{P}_3$	0.442	$\mathbf{P}_{11}$	0.388	$\mathbf{P}_{19}$	0.558
$\mathbf{P}_4$	0.465	$\mathbf{P}_{12}$	0.478	$\mathbf{P}_{20}$	0.533
$\mathbf{P}_5$	0.429	$\mathbf{P}_{13}$	0.465	$\mathbf{P}_{21}$	0.487
$\mathbf{P}_6$	0.411	$\mathbf{P}_{14}$	0.392	$\mathbf{P}_{22}$	0.511
$\mathbf{P}_7$	0.389	$\mathbf{P}_{15}$	0.471	$\mathbf{P}_{23}$	0.472
$\mathbf{P}_8$	0.423	$\mathbf{P}_{16}$	0.476	$\mathbf{P}_{24}$	0.527

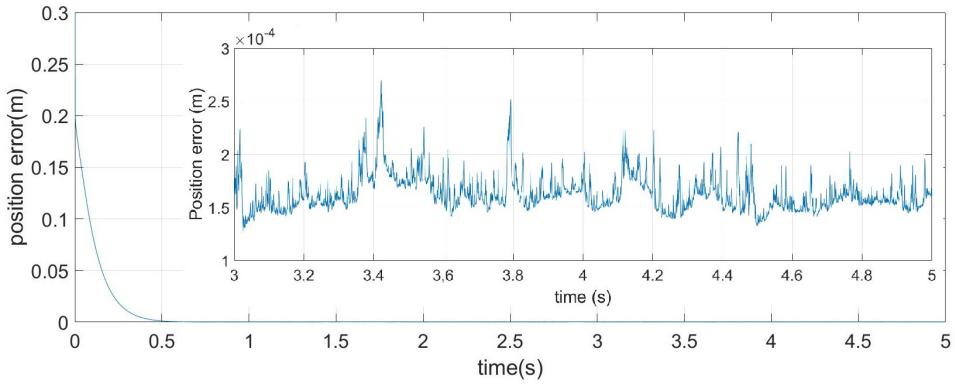

Figure 3.16: Positioning error at point $\mathbf{P}_{18}$ of the image moment optimized design

Figs. 3.15, 3.16 and Tab. 3.5, we see that when using image moment visual servoing, the maximal positioning errors are lower than 0.3 mm.

For the DELTA robot designed by using leg-direction-based visual servoing, when the desired point to attain is $\mathbf{P}_{10}$, the simulation results are presented in Fig. 3.17. When the desired point is moved to $\mathbf{P}_{18}$, the simulation results are illustrated in Fig. 3.18. For the other desired points, the maximal and average positioning errors are presented in Tab. 3.6.

From the Figs. 3.17, 3.18 and Tab 3.6, we see that when using leg-direction-based visual servoing, the maximal positioning errors are lower than 0.563 mm.

We see from these simulation results that, even if the error models defined in Section 3.3 were simplified, resulting optimized robots have a maximal accuracy of around 0.5 mm for all points tested in their workspace.

Then, considering the size of optimized DELTA robots and their positioning errors,

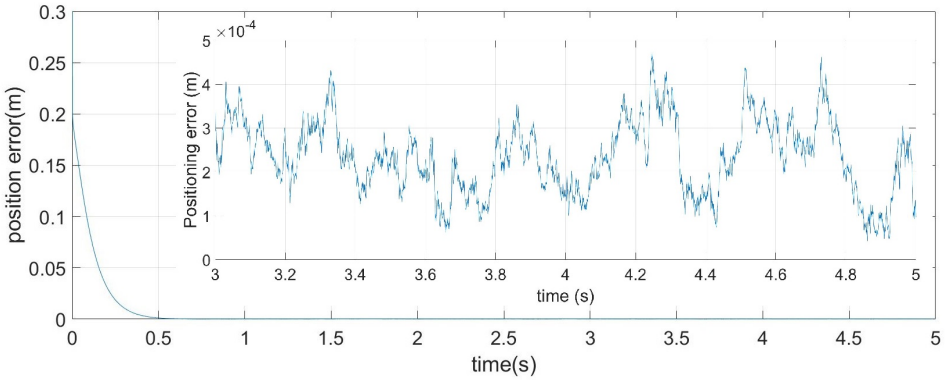

Figure 3.17: Positioning error at point $\mathbf{P}_{10}$ of the leg-direction-based optimized design

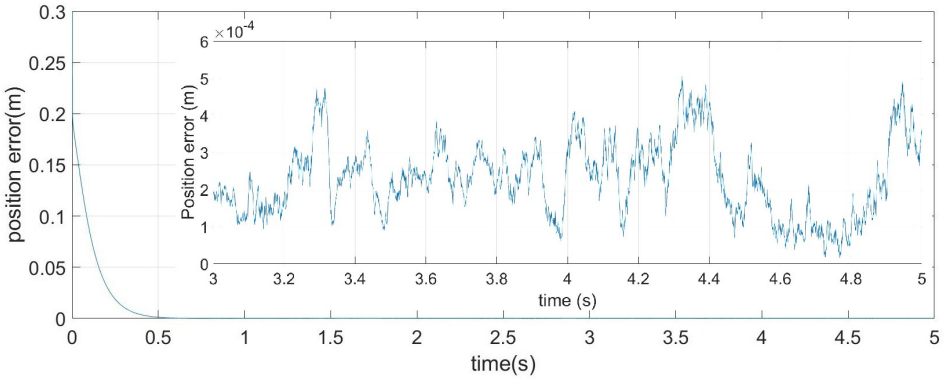

Figure 3.18: Positioning error at point $\mathbf{P}_{18}$ of the leg-direction-based optimized design

Table 3.5: DELTA robot simulation result of image moment visual servoing

Point	Max error[mm]	Point	Max error[mm]	Point	Max error[mm]
$\mathbf{P}_1$	0.228	$\mathbf{P}_9$	0.249	$\mathbf{P}_{17}$	0.253
$\mathbf{P}_2$	0.234	$\mathbf{P}_{10}$	0.225	$\mathbf{P}_{18}$	0.269
$\mathbf{P}_3$	0.202	$\mathbf{P}_{11}$	0.227	$\mathbf{P}_{19}$	0.256
$\mathbf{P}_4$	0.213	$\mathbf{P}_{12}$	0.268	$\mathbf{P}_{20}$	0.254
$\mathbf{P}_5$	0.210	$\mathbf{P}_{13}$	0.252	$\mathbf{P}_{21}$	0.259
$\mathbf{P}_6$	0.221	$\mathbf{P}_{14}$	0.230	$\mathbf{P}_{22}$	0.271
$\mathbf{P}_7$	0.213	$\mathbf{P}_{15}$	0.231	$\mathbf{P}_{23}$	0.267
$\mathbf{P}_8$	0.218	$\mathbf{P}_{16}$	0.250	$\mathbf{P}_{24}$	0.247

Table 3.6: DELTA robot simulation result of leg-direction-based visual servoing

Point	Max error[mm]	Point	Max error[mm]	Point	Max error[mm]
$\mathbf{P}_1$	0.417	$\mathbf{P}_9$	0.492	$\mathbf{P}_{17}$	0.563
$\mathbf{P}_2$	0.473	$\mathbf{P}_{10}$	0.471	$\mathbf{P}_{18}$	0.506
$\mathbf{P}_3$	0.414	$\mathbf{P}_{11}$	0.477	$\mathbf{P}_{19}$	0.486
$\mathbf{P}_4$	0.437	$\mathbf{P}_{12}$	0.487	$\mathbf{P}_{20}$	0.461
$\mathbf{P}_5$	0.422	$\mathbf{P}_{13}$	0.469	$\mathbf{P}_{21}$	0.559
$\mathbf{P}_6$	0.455	$\mathbf{P}_{14}$	0.493	$\mathbf{P}_{22}$	0.534
$\mathbf{P}_7$	0.421	$\mathbf{P}_{15}$	0.468	$\mathbf{P}_{23}$	0.497
$\mathbf{P}_8$	0.433	$\mathbf{P}_{16}$	0.470	$\mathbf{P}_{24}$	0.479

we see that when using leg-direction-based visual servoing, the robot size is twice the size of the line-based designed robot and image moment designed robot. Thus, leg-direction-based visual servoing for the DELTA robot leads to the biggest robot size, so it is not a good choice. For the robot optimized for line-based visual servoing, the robot size is almost the same as for the robot designed for image moment visual servoing. Additionally, when using image moments, the controller accuracy is better than line-based controller.

Among the three tested controllers, image moment visual servoing may be the best controller with respect to the accuracy performance, but the difference of accuracy between image moment and line-based visual servoing is not significant. Thus only experiments may provide the final conclusions. So we still need to do experiments on real prototypes to see which controller can lead to the best accuracy performance of the DELTA robot. This is the next step of our work.

3.7 Conclusion

In the work presented above, we performed "control-based design" of a DELTA robot in order to obtain the best accuracy performance of the robot with its controller. The proposed control-based design methodology took into account the performance of the controller in the design process to get the optimal geometric parameters of robot. We optimized the design of DELTA robot for three different types of controllers: leg-direction-based visual servoing, line-based visual servoing and image moment visual servoing. Based on these three controllers, we developed positioning error models taking into account the error of observation coming from the camera. We also analyzed the singularities of these controllers to be sure that no singularity of the controller appeared in the final design of the robot, thus avoiding instability issues. Then, design optimization problems have been formulated in order to find the optimal geometric parameters and camera placement for the DELTA robot for each type of controller. The simulation results showed that the robot designed for image moment visual servoing was more compact and had better accuracy performance than the other two robots optimized for other control techniques. However the differences of robot size and accuracy between the image moments controller and line-based controller were not significant enough in order to draw general conclusions. Therefore, experimental works on real prototypes should be done in the next step in order to verify the simulation results.

Chapter 4

Prototype and experimental validations

This chapter is devoted to presenting the experimental results for the visual servoing of DELTA robot prototypes. First of all, the mechanical design of the prototypes and the instrumentation of the prototypes are explained. Then the camera identification is performed to get the intrinsic and extrinsic parameters of the camera, the hypothesis over the noise introduced by the camera and the image processing is checked. In the next step, the image moment visual servoing controller and line-based visual servoing controller applied in the experiments are presented. Finally, the experiments are performed and the experiment results obtained demonstrate that image moment visual servoing controller is the best controller for the control of DELTA robot prototypes compared with line-based visual servoing.

4.1 Description of the prototypes

In this section, the DELTA prototypes as well as their hardware and controller implementations that I did by myself under the supervision of the LS2N staff technical will be presented. The geometric parameters of the DELTA prototypes are got from the optimization results from Section 3.5. Two prototypes with different links, base and platform sizes are produced, one is controlled by the line-based visual servoing controller (We call it "Prototype 1"), the other one is controlled by image moment visual servoing controller (We call it "Prototype 2"). The two prototypes share the same mechanical design concepts and they are only different in geometric parameters. In this section, we take the prototype designed for image moment visual servoing (Prototype 2) as an example and we present it in details.

4.1.1 CAD modeling and prototyping of robot

Here, the CAD model of the DELTA robot Prototype 2 is shown in Fig. 4.1. The robot is composed as follows:

- One base which is fixed in space.
- Three motors for the actuation of the active joints which are linked to the fixed base by a part called "support of the motors" (Fig. 4.2).
- Three kinematic chains, each kinematic chain consists of one proximal link and one parallelogram. Each parallelogram is composed of four ball joints, two axis links and two distal links.
- One end-effector connected with the three parallelograms (Fig. 4.3).

Fig. 4.4 shows the front view of the real Prototype 1 and Prototype 2. Based on the results got from the optimization, the location of the camera is shown in Fig. 4.5. In order to keep the structure of the parallelogram, two springs were fixed on each parallelogram architecture (Fig. 4.6).

Considering reducing the overall weight of the prototype while keeping the structure stiff, the proximal links and the rectified square were made of plastic by the 3D printer; the support of motors, the end-effector and the distal links were make of aluminum; the axis links were made of steel.

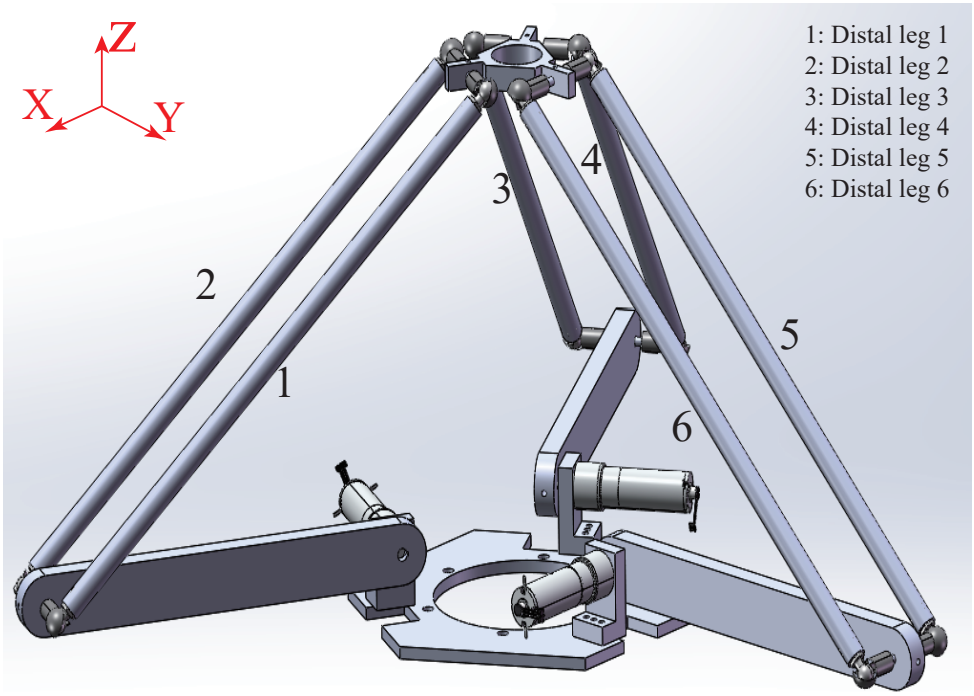

Figure 4.1: CAD of the prototype designed for image moment visual servoing

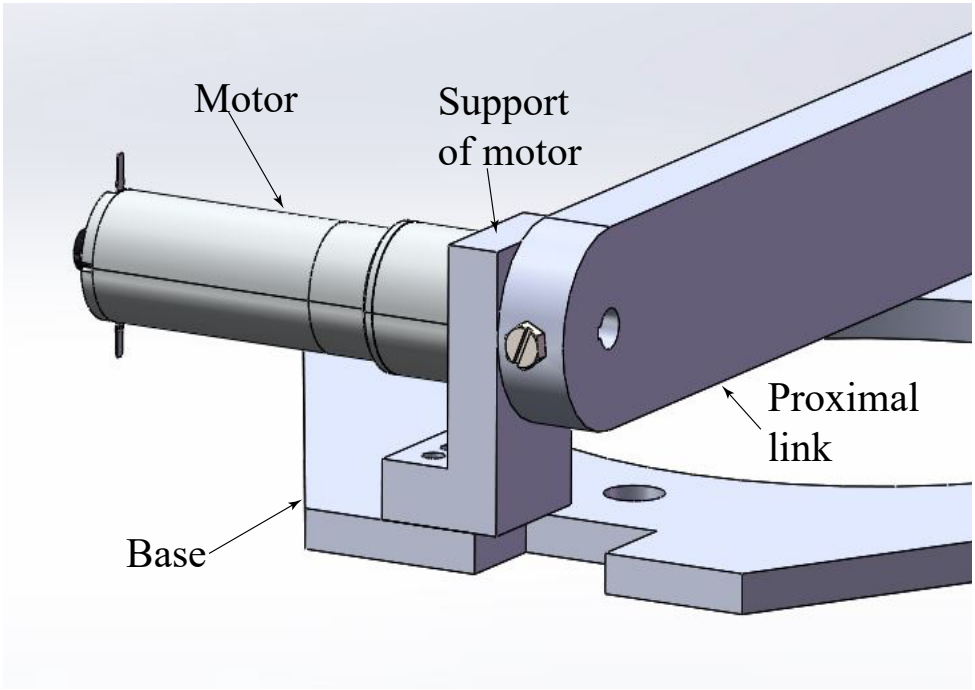

Figure 4.2: Detailed description of the link between the base, motor and proximal link

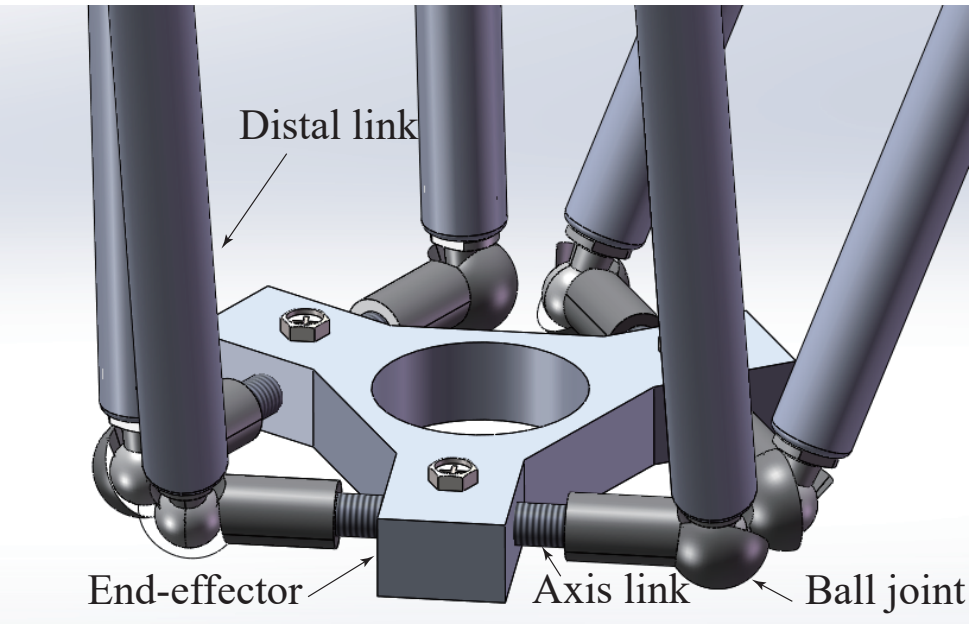

Figure 4.3: Detailed description of the link between the end-effector and the parallelogram

Figure 4.4: Real prototype in front view; left: Prototype 1 for line-based visual servoing, right: prototype 2 for image moment visual servoing

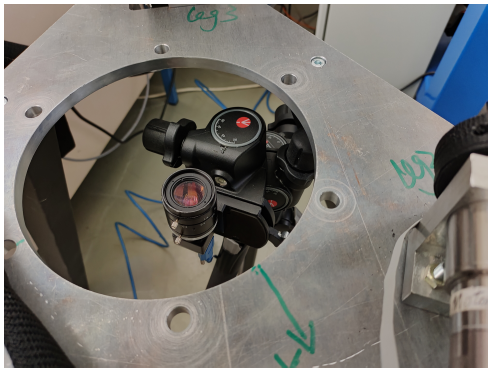

Figure 4.5: Camera location of the prototype

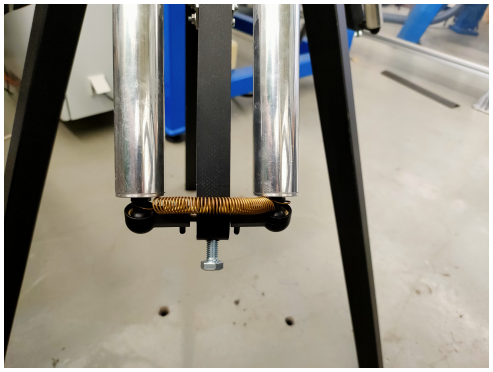

Figure 4.6: Spring fixed on the parallelogram structure of the prototype

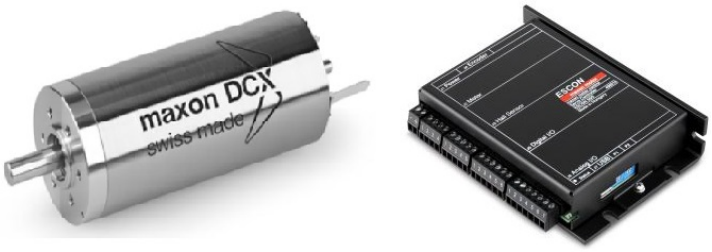

Figure 4.7: Maxon motor drive system

4.1.2 Instrumentation and communication

Maxon motors were used for actuating the three active joints of the prototype. The three motor drive systems were selected identical and the motors are Maxon DCX 32L motors and ESCON 50/5 for the drivers (Fig. 4.7). The motor was equipped with the ECX EASY Encoder and GPX 37A gear. The general characteristics of the motors are:

- Rated torque: 128 mNm
- Rated speed: 11300 rpm
- reduction ratio of the gear: 150:1
- resolution of the encoders: 1024 point/revolution

With the gear connected to the motor, the total number of encoder points for one turn of the proximal link is $1024 * 150 = 153600$.

In order to control the motors in the experiments, a dSPACE DS1103 PPC controller board which allows to implement real-time control in combination with Matlab/Simulink

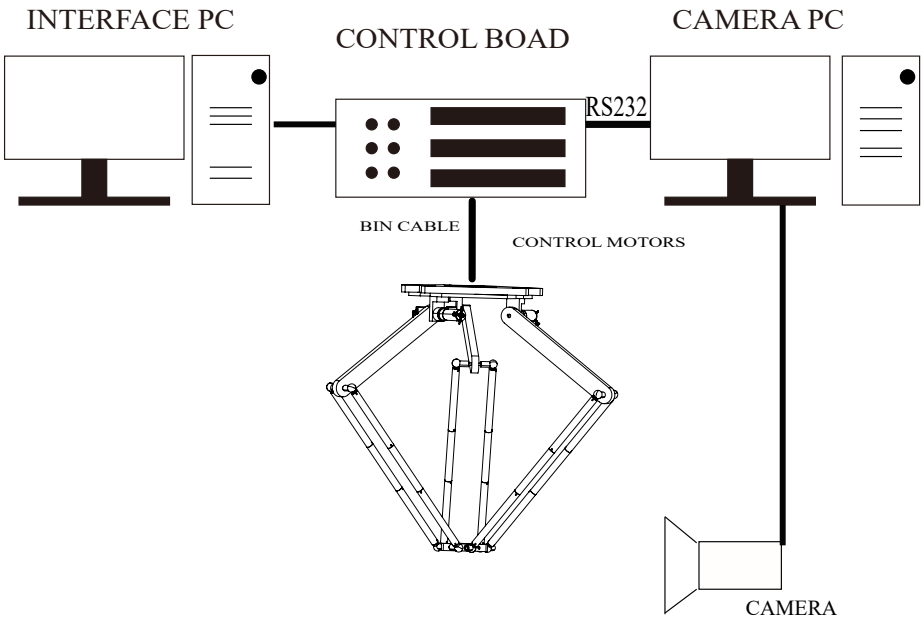

Figure 4.8: Interaction layout for DELTA robot prototype

was used. The dSPACE experiment software ControlDesk 5.6 was also used to design the graphical interface for measuring and monitoring data from the experiments. The dSPACE controller board is a single-board system with real-time processor. The possibility of using Matlab/Simulink to control the dSPACE processor card is an advantage for rapid-prototype testing.

Then the layout to control and interact with the robot is shown in Fig. 4.8. The motors of the prototypes are controlled by the dSPACE control board (and through the converters). This control board is interfaced with the INTERFACE PC through optics fiber and the graphical interface designed with the software ControlDesk 5.6 is built in the INTERFACE PC. Additionally, the control board also takes care to communicate and process the information extracted and sent from the cameras by the CAMERA PC. Between the CAMERA PC and the control board is done through the RS232 cable, which highly limits the amount of information that can be transmitted.

4.1.3 Prototype system safety design

4.1.3.1 Hardware safety design

In order to ensure the safety of the operator and avoid damage to the motors, the motor drivers and the prototypes, the system safety design should be considered. In terms of hardware design, two breakers were set in this system. One is the circuit breaker which connects the power supply and the motor drivers. It is the emergency button of the system. We can press the button to cut off the power supply of the whole system when there is an emergency. The other one is the motor button that controls the enable state of the motor (one motor button for each of the three motors). Once the motor is working in a dangerous situation and we do not want to cut off the power supply of the whole system, the motor button can be pressed to individually disable the motor.

4.1.3.2 Software safety design

In addition to the hardware safety design, we also add multiple software safety designs to avoid any risks. Before the experiment, all possible emergency or unsafe cases ought to be considered. For example, the camera loses the track of image features, the serial communication between the CAMERA PC and the dSPACE control board fails, the robot comes to a singularity configuration or the boundary of the reachable workspace etc.

The software safety designs to avoid above unsafe cases were illustrated in Fig. 4.9. Two test values were set as the feedback: serial data feedback and visual servoing feedback. Serial feedback is assigned with a Boolean value to reflect the working status, when the serial communication works well, the camera continues to track image features and the image data can be translated to the CAMERA PC well, the Boolean value is true; otherwise, this Boolean value is false. The visual servoing feedback is assigned with a Boolean value too. It is true only when all the output joint velocities are scalar. When both the serial feedback and the visual servoing feedback are true, the visual servoing controller will be active and the joint velocities obtained from the controller will be sent to the DAC of the dSPACE control board to drive the motor. Once either of the two values is false, the PID position controller will be activated and drive the prototype to the home position which is the center of the dexterous workspace.

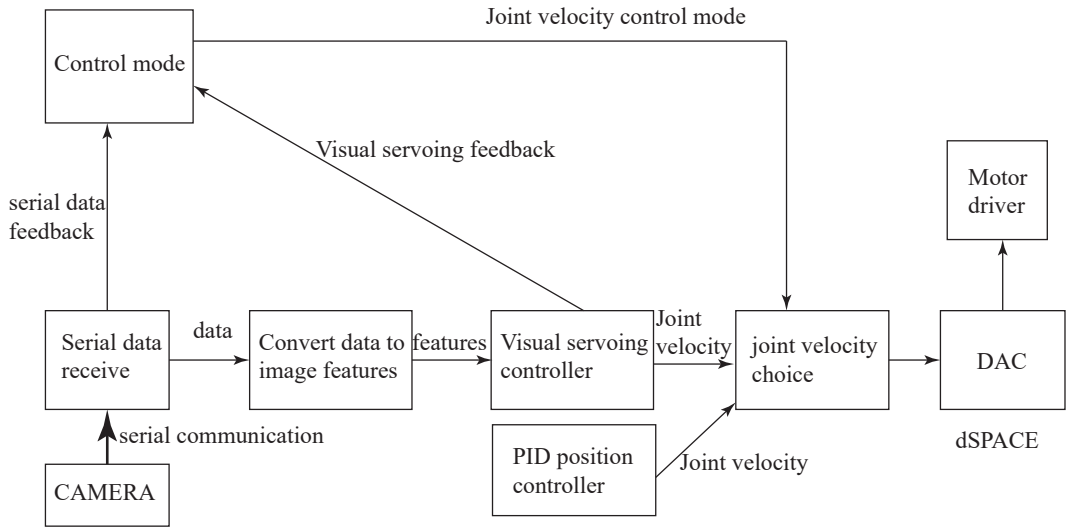

Figure 4.9: Software safety design

4.2 Controller design

4.2.1 Camera identification

In order to perform the experiments with the visual servoing controllers, the camera's intrinsic and extrinsic parameters should be properly identified. Then, confirm the noise introduced by the camera observation.

4.2.1.1 Camera parameters

The camera model we studied in this section is a pinhole model and the camera calibration is the process that estimates the parameters of the pinhole model. The pinhole camera model expresses how a 3D point is projected onto the image plane of a camera and the following equation defines this model (see Fig. 4.10):

$$ {}^{p}\mathbf{A} = \frac{1}{{}^{c}z}\mathbf{K}\ {}^{c}\mathbf{T}_{o}\ {}^{o}\mathbf{A} \tag{4.1} $$

with

$$ {}^{c}\mathbf{A} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad {}^{p}\mathbf{A} = \begin{bmatrix} {}^{c}x \\ {}^{c}y \\ {}^{c}z \\ 1 \end{bmatrix}, \quad {}^{o}\mathbf{A} = \begin{bmatrix} {}^{o}x \\ {}^{o}y \\ {}^{o}z \\ 1 \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} f_u & 0 & u_0 & 0 \\ 0 & f_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad {}^{c}\mathbf{T}_{o} = \begin{bmatrix} {}^{c}\mathbf{R}_{o} & {}^{c}\mathbf{t}_{o} \\ 0_{1\times 3} & 1 \end{bmatrix} \tag{4.2} $$

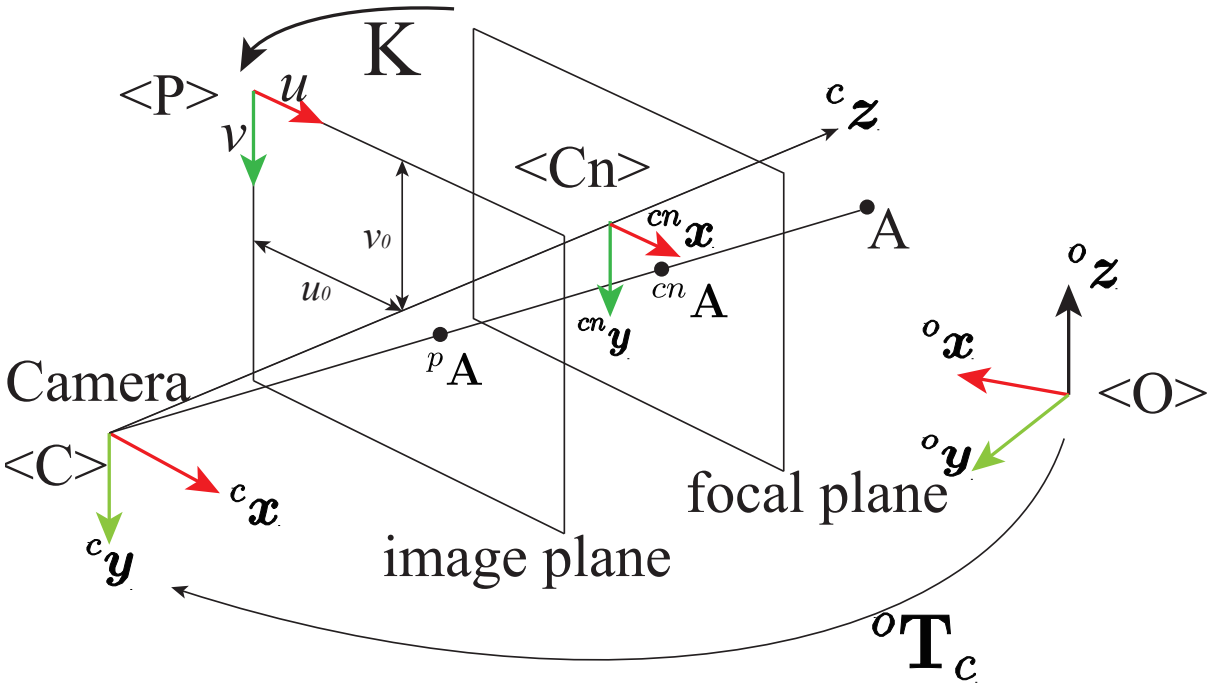

Figure 4.10: Global diagram of the camera frames and parameters

where the subscripts o, c, p represent the origin frame **O** (which is the global frame), camera frame **C** (origin point is the center of the camera) and **P** pixel frame (origin point is the center of the pixel plane of the camera) respectively. For a point **A** in space, it is expressed in homogeneous coordinates, x, y, z are the coordinates this point in any frame and u, v are the pixel coordinates in pixel frame. The matrix ${}^{c}\mathbf{T}_{o}$ is the homogeneous matrix that transforms the pose of the origin frame with respect to the camera frame, where ${}^{c}\mathbf{R}_{o}$ is the rotation matrix and ${}^{c}\mathbf{t}_{o}$ is the translation vector. The definition of the components of matrix **K** are given in section 1.2.3.

In the Fig. 4.10, the subscript cn represents the normalized camera frame $< Cn >$, ${}^{cn}\mathbf{A} = [{}^{cn}x,\ {}^{cn}y, 1, 1]^{T}$, ${}^{cn}x = \frac{{}^{c}x}{{}^{c}z}$, ${}^{cn}y = \frac{{}^{c}y}{{}^{c}z}$, ${}^{cn}z = \frac{{}^{c}z}{{}^{c}z}$. In this experiment, the distortion of the camera is not taken into consideration.

Therefore, there are ten parameters to calibrate before the experiment, four belong to **K** and six to ${}^{c}\mathbf{T}_{o}$. Taking into consideration the sources of noise, the calibration process is done in multiple images. The transformation matrix ${}^{c}\mathbf{T}_{o}$ is different for each frame and the matrix **K** is constant. The use of several images makes a better estimation of the parameters of **K**. All the calibration process was based in the available VISP library which implements a camera calibration based in virtual visual servoing, presented by

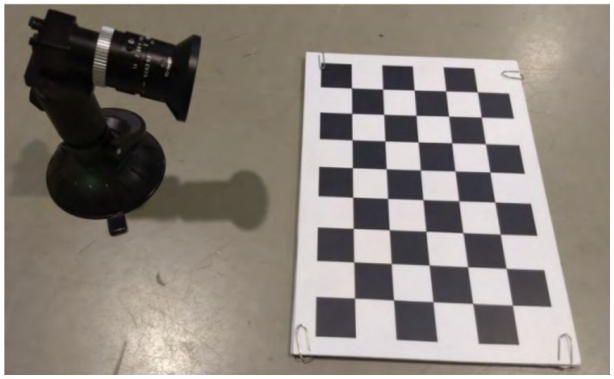

Figure 4.11: Toshiba Teli BU238M and chessboard

[Marchand 2001]. The camera we used in this project was Toshiba Teli BU238M whose resolution is 1920×1200 pixels and its maximum frequency is 165 Hz. The objectives used along with the camera had 8 mm of focal length with a field of view of $79.3° \times 62.8°$. A classical chessboard pattern of 8×5 squares is applied as a calibration marker (see Fig. 4.11).

The identification results are given below

- $f_u = 1395$;
- $f_v = 1396$;
- $u_0 = 939$;
- $v_0 = 523$;

4.2.1.2 Estimation of the measurement noise

As we see from the figure 4.8, the CAMERA PC is used to drive the camera to observe the image features and send the data to the dSPACE controller board by the serial cable RS232. This function is realized in the C++ environment with the help of VISP library. When using line-based visual servoing, a dedicated tracker was implemented, which tracks the links blobs and fits in the border subsets of pixels, the desired lines (Fig. 4.12). When using image moment visual servoing, an ellipse tracker is provided by the VISP library, the coordinate of the centroid point of the ellipse and its area can be got directly from this library (Fig. 4.13).

In order to measure the real noise introduced by the camera system, the camera and the robot data were recorded while placing the later on different points of the workspace

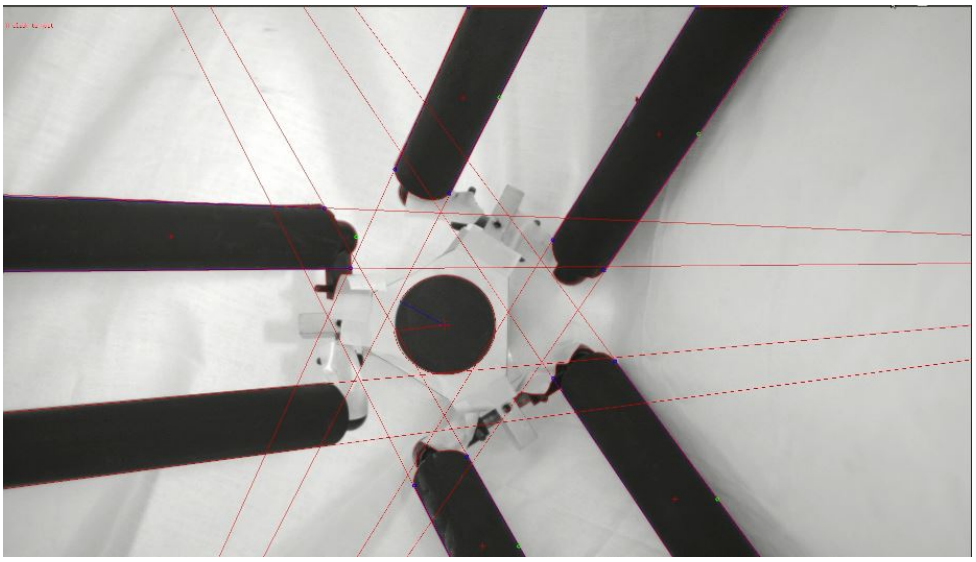

Figure 4.12: Blob track when using line-based visual servoing (tracking 6 legs)

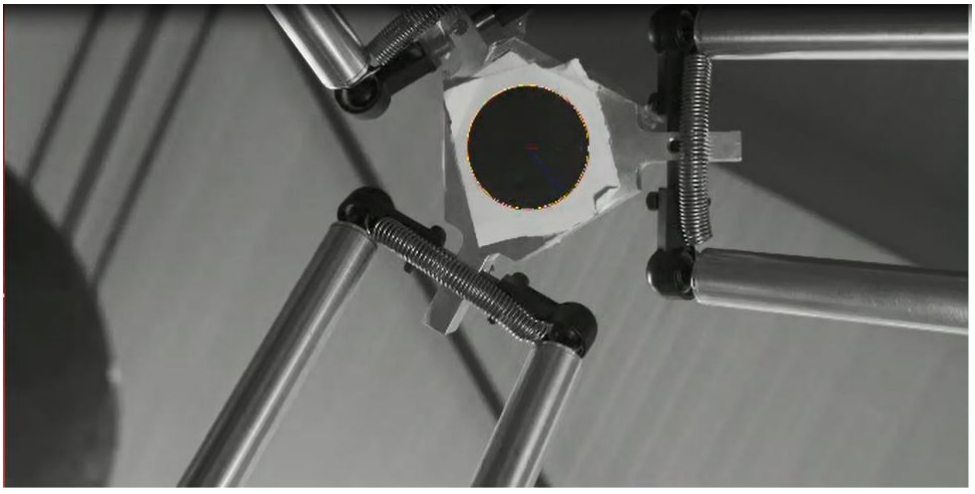

Figure 4.13: Ellipse track when using image moment visual servoing

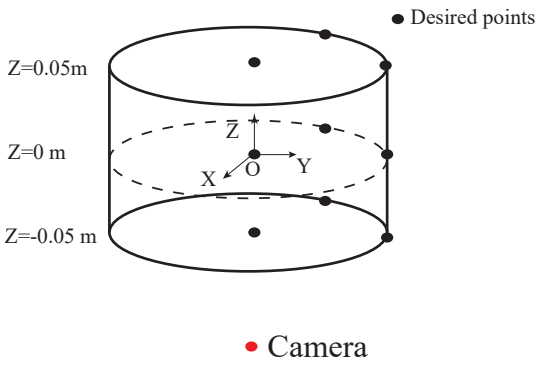

Figure 4.14: DELTA robot prototype noise analysis (line-based visual servoing) test points in the workspace

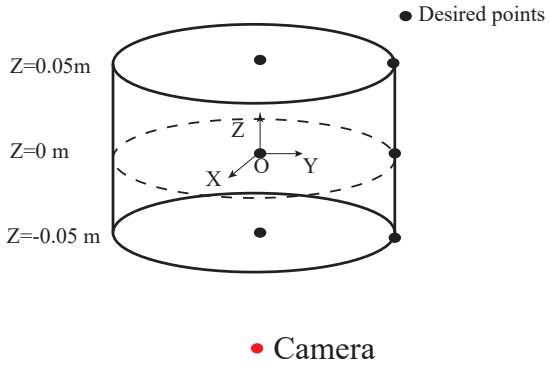

Figure 4.15: DELTA robot prototype noise analysis (image moment visual servoing) test points in the workspace

Table 4.1: Standard deviation of the noise of each feature introduced by the camera. Measures are in pixel

		Std of noise [pixel]
Line-based visual servoing	**Intersect points**	0.2073
Image moment visual servoing	$\mathbf{x}_g, \mathbf{y}_g$	0.1699
Image moment visual servoing	**Radius**	0.1854

of interest (see Fig. 4.14 and 4.15). The methodology to measure the noise was given here:

1. Move the prototype to the desired point.

2. Deactivate the controller

3. Measure the data for 10 s (record the measurements)

4. Repeat 1 to 3 step for all the test points within the workspace.

For the Prototype 1 controlled by line-based visual servoing, the noises of the intersection points of the projection of the robot leg edges and the image plane boundary were measured. For the Prototype 2 controlled by image moment servoing, three noises were recorded: the coordinates of the centroid point (x_g, y_g) in pixel and the radius ($r = \sqrt{\frac{m_{00}}{\pi}}$, m_{00} is the area of the image) of the image in pixel. The measurements we recorded from the step 3 are illustrated in Tab. 4.1 and Figs. 4.16, 4.17 and 4.18.

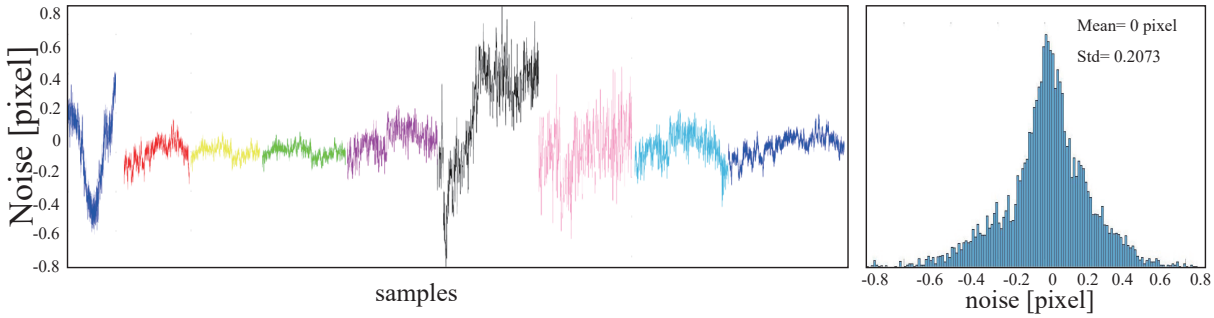

Figure 4.16: Noise measures of the edges features in format: border pixels (each color is a point) in line-based visual servoing and the histogram of the noise

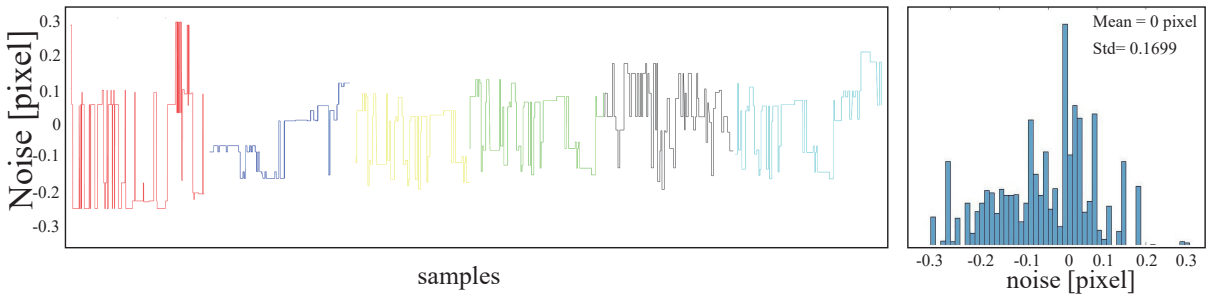

Figure 4.17: Noise measures of the centroid point of the image in pixels (each color is a point) in image moment visual servoing and the histogram of the noise

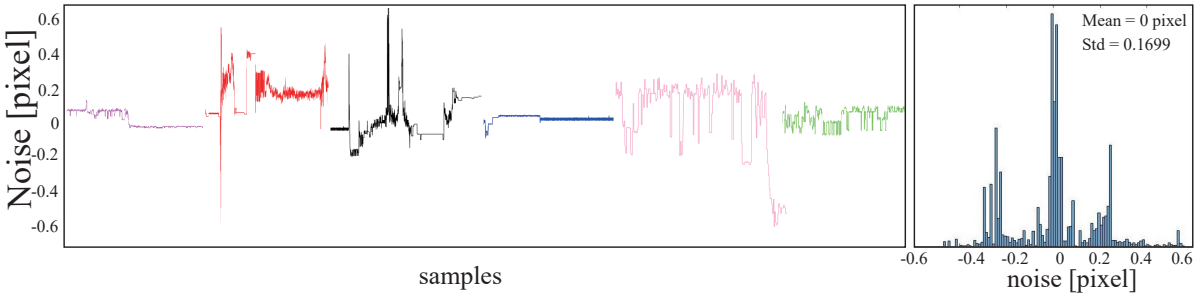

Figure 4.18: Noise measures of the radius of the image in pixels (each color is a point) in image moment visual servoing and the histogram of the noise

From the results in Tab. 4.1, we see that the hypothesis of a noise of 1 pixel for the intersection points between the edges and the image plane boundary in line-based visual servoing was largely overestimated, since 99% of the noise for those points is less than 0.6 pixels. The hypothesis of a noise of 1 pixel for the centroid point of the image and its radius in image moment visual servoing were largely overestimated too, since 99% of the noise of the centroid point is less than 0.5 pixels and 99% of the noise of the radius is less than 0.54 pixels. Figs. 4.16 4.17 4.18 show the plotted noise for each of the cases.

Based on the results of the noise measurements, in the process of creating the positioning error model, the noise of 1 pixel is overestimated.

4.2.2 Visual servoing controller design

As explained in the Section 3.7, line-based visual servoing is applied on the control of the DELTA Prototype 1 and image moment visual servoing is applied on the control of the DELTA Prototype 2. Line-based visual servoing took the cylinder edge of the robot's distal links as the visual features. Image moment visual servoing took the coordinated of the centroid point and area of the ellipse fixed on the end-effector as the image features.

As presented in Section 1.2.3, the vector $({}^{c}\underline{\mathbf{n}}_i^1, {}^{c}\underline{\mathbf{n}}_i^2)$ can be obtained from the line equation in pixel [Vignolo 2014]. Line-based visual servoing controller applied in the experiment was the controller based on the control of the projection of the edges of legs and is defined by the equation (1.49). As we studied in Section 3.2.2, observing two of its six distal legs were enough to fully control the DELTA robot when using line-based visual servoing and in [Briot 2016], the selection of the legs to observe is crucial for the final pose accuracy. As a result, in this experiment, three different sets of observed legs were used: (i) legs {1,3}; (ii) legs{1,3,5}; (iii) legs{1,2,3,4,5,6}. The final results will show us which choice is better in terms of pose accuracy in experiments.

For the image moment visual servoing controller, the coordinates of the centroid point and the area of the ellipse in pixel can be got from the camera. The image moment visual servoing controller applied in the experiment was the controller defined by the equation (1.71) in Section 1.2.4.

However, since the gain of the controller is constant, the joint velocity calculated was big when the error of the feedback is huge. The camera will lose the track of the image features if the robot moves too quickly. A limit of the joint velocity should be applied in the visual servoing controller. As a result, the gain adaptive was applied in the controller

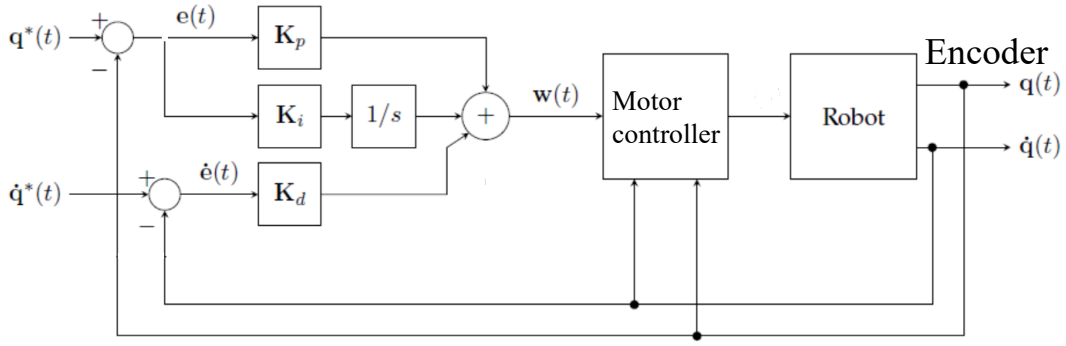

Figure 4.19: PID control scheme

[VISP]:

$$\lambda(x) = (\lambda_0 - \lambda_\infty) exp(\frac{-\mu}{\lambda_0 - \lambda_\infty} x) + \lambda_\infty \tag{4.3}$$

with $x = \|e\|_\infty$. We have $\lambda_0 > \lambda_\infty > 0$, $\mu > 0$. When the error $\mathbf{e}$ is big, the gain $\lambda(x)$ tend towards λ_∞, when the error $\mathbf{e}$ is small, the gain tend towards λ_0.

In addition, a PID position controller is needed to drive the robot move from the home position to the desired position to note the image features. In general, the output joint velocity can be defined as a PID controller:

$$\mathbf{w}(t) = \mathbf{K}_d(\dot{\mathbf{q}}^*(t) - \dot{\mathbf{q}}(t)) + \mathbf{K}_p(\mathbf{q}^*(t) - \mathbf{q}(t)) + \mathbf{K}_i \int_{t_0}^{t} (\mathbf{q}^*(t) - \mathbf{q}(t))dt \tag{4.4}$$

where $\mathbf{q}^*$ is the (3×1) vector of joint positions and $\mathbf{K}_d$, $\mathbf{K}_p$, $\mathbf{K}_i$ are the positive diagonal matrices, $\mathbf{w}$ is the (3×1) vector of joint velocity. The control scheme is shown in Fig. 4.19.

4.3 Experiment process

In order to validate the precision hypothesis, the repeatability of each robot was measured. Twenty one points throughout the workspace were chosen to be the desired points in which the repeatability was measured (Fig. 4.21 and Tab. 4.2). A rectified square was attached to the end-effector so that the repeatability of the prototype can be measured by the micrometers (Fig. 4.20). Three micrometers were used to measure the displacement of the end-effector along X, Y, Z axes. The rectified square ought to be attached to the end-effectors with its faces parallel to the main axes, then the three micrometers positioned

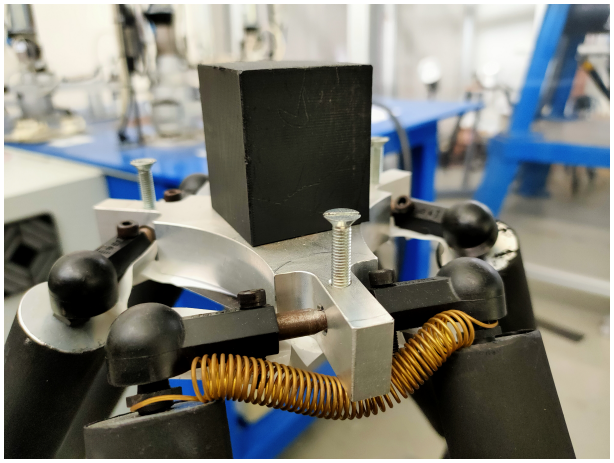

Figure 4.20: Rectified square of the prototype

against it, each one in one face, so that the displacement along X Y and Z axes can be measured.

The processes of the experiment are given here:

1. Place the DELTA robot prototype in the desired final position using the position controller

2. Record the comparators values and the desired visual features.

3. Move the robot to the initial position with the position controller

4. Return to the desired position using the visual servoing controller

5. Read the measures in the comparator
   Repeat 9 times the steps from 3 to 5 (3 times for approaching the point along X direction, 3 times along Y direction and 3 times along Z axis) for one desired final position.

6. Repeat all the steps above for all the 21 points in the workspace.

For the experiments operated on the prototype 1 when applying line-based visual servoing, as presented above, three different sets of observed legs were used: (i) legs {1,3}; (ii) legs{1,3,5}; (iii) legs{1,2,3,4,5,6} (Fig. 4.1).

The results of the experiments are presented in Tab. 4.3 to 4.6 and Fig. 4.22 to 4.25. The standard deviations of the visual servoing repeatability results are presented in Tab. 4.7 to 4.10 and Fig. 4.26 to 4.29.

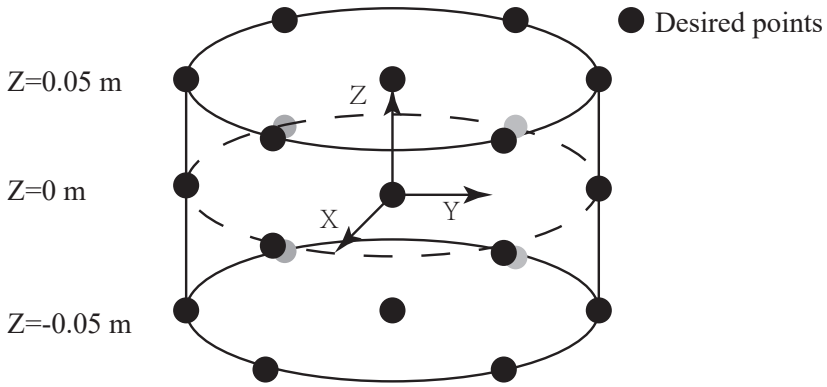

Figure 4.21: Desired test points with respect to the workspace frame in the experiment

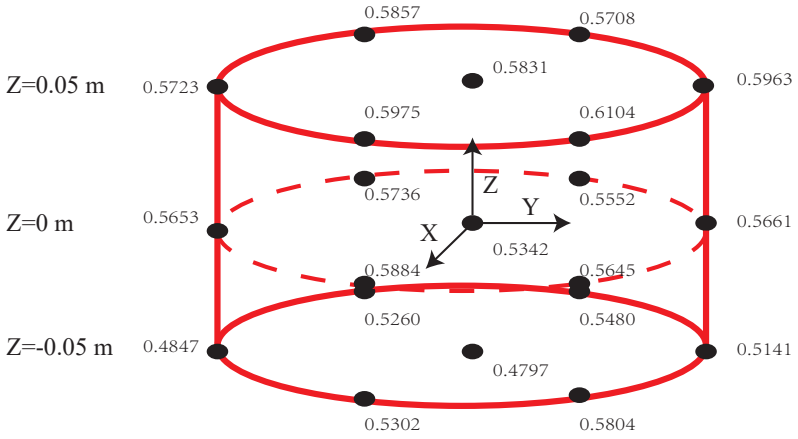

Figure 4.22: Visual servoing repeatability results of the test points using line-based visual servoing (observing 2 legs {1,3}). Measures are in mm

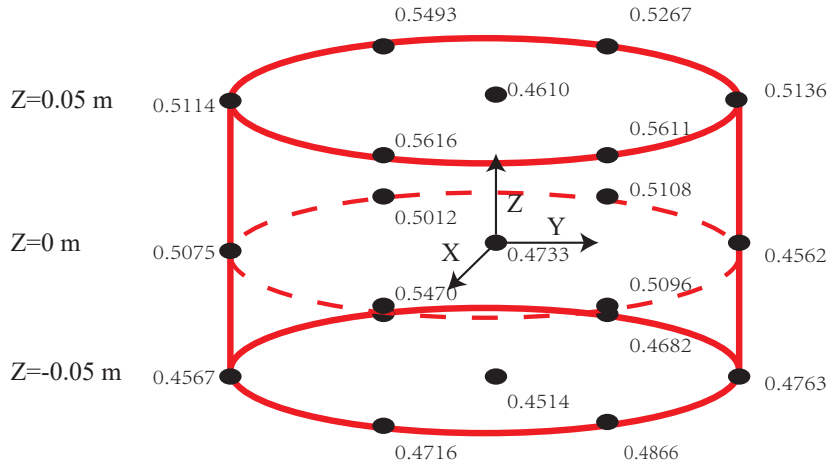

Figure 4.23: Visual servoing repeatability results of the test points using line-based visual servoing (observing 3 legs {1,3,5}). Measures are in mm

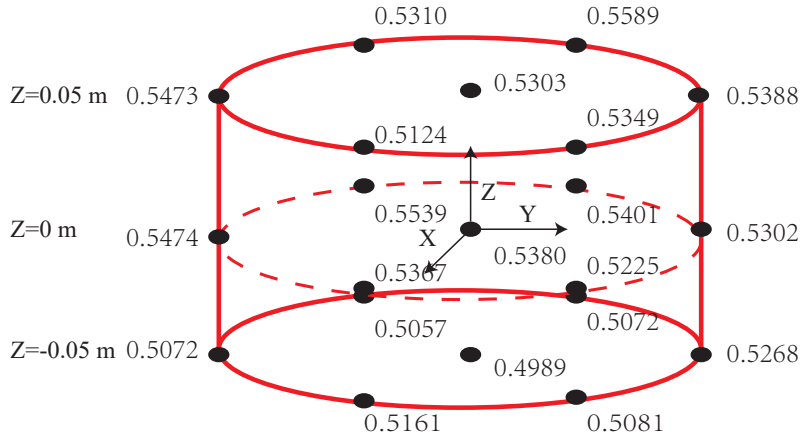

Figure 4.24: Visual servoing repeatability results of the test points using line-based visual servoing (observing 6 legs {1,2,3,4,5,6 }). Measures are in mm

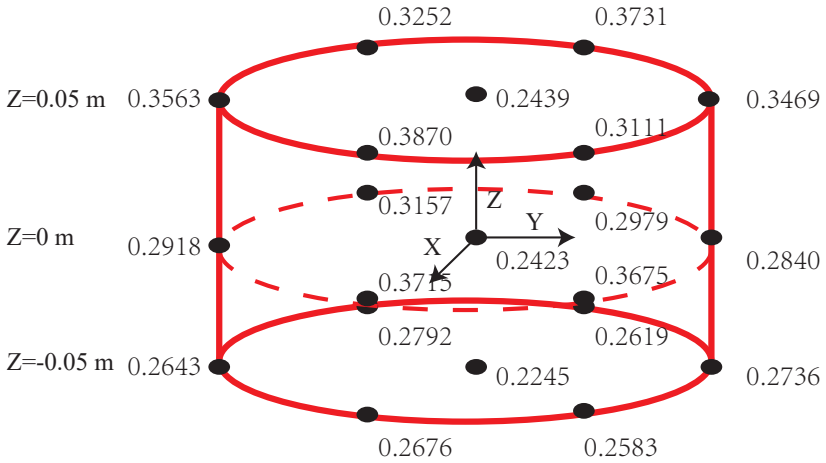

Figure 4.25: Visual servoing repeatability results of the test points using image moment visual servoing. Measures are in mm

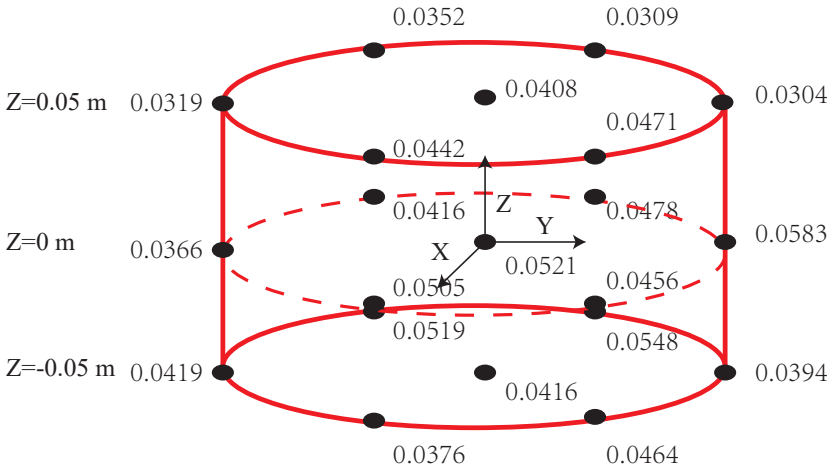

Figure 4.26: Standard deviations of visual servoing repeatability results of the test points using line-based visual servoing (observing 2 legs {1,3}). Measures are in mm

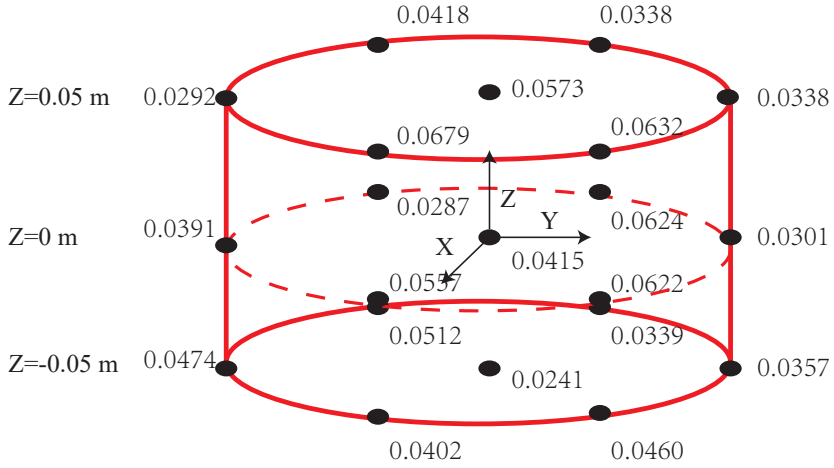

Figure 4.27: Standard deviations of visual servoing repeatability results of the test points using line-based visual servoing (observing 3 legs {1,3,5 }). Measures are in mm

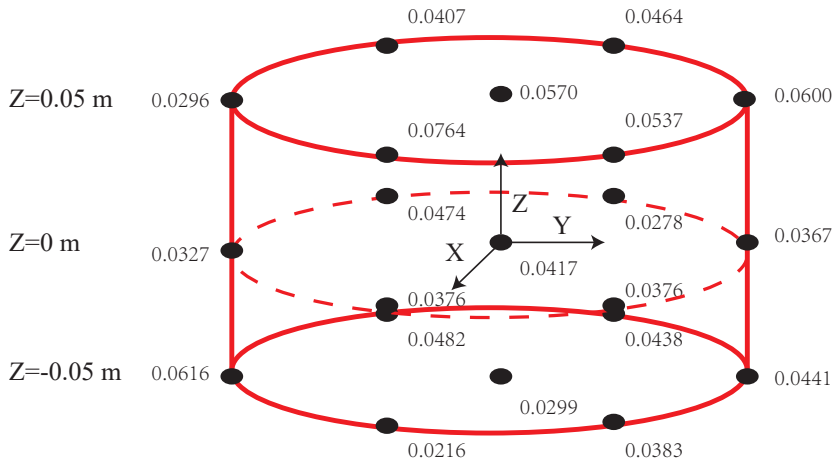

Figure 4.28: Standard deviations of visual servoing repeatability results of the test points using line-based visual servoing (observing 6 legs {1,2,3,4,5,6 }). Measures are in mm

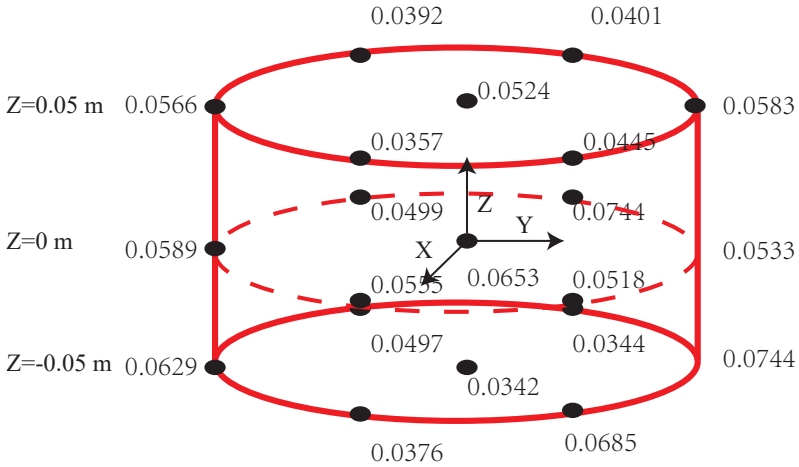

Figure 4.29: Standard deviations of visual servoing repeatability results of the test points using image moment visual servoing. Measures are in mm

Table 4.2: Desired test points with respect to the workspace frame in the experiment

Point	Coordinate [m]	Point	Coordinate [m]	Point	Coordinate [m]
$\mathbf{P}_1$	(0,0,-0.05)	$\mathbf{P}_8$	(0,0,0)	$\mathbf{P}_{15}$	(0,0,0.05)
$\mathbf{P}_2$	(0.17,0.1,-0.05)	$\mathbf{P}_9$	(0.17,0.1,0)	$\mathbf{P}_{16}$	(0.17,0.1,0.05)
$\mathbf{P}_3$	(0.17,-0.1,-0.05)	$\mathbf{P}_{10}$	(0.17,-0.1,0)	$\mathbf{P}_{17}$	(0.17,-0.1,0.05)
$\mathbf{P}_4$	(-0.17,-0.1,-0.05)	$\mathbf{P}_{11}$	(-0.17,-0.1,0)	$\mathbf{P}_{18}$	(-0.17,-0.1,0.05)
$\mathbf{P}_5$	(-0.17,0.1,-0.05)	$\mathbf{P}_{12}$	(-0.17,0.1,0)	$\mathbf{P}_{19}$	(-0.17,0.1,0.05)
$\mathbf{P}_6$	(0,0.2,-0.05)	$\mathbf{P}_{13}$	(0,0.2,0)	$\mathbf{P}_{20}$	(0,0.2,0.05)
$\mathbf{P}_7$	(0,-0.2,-0.05)	$\mathbf{P}_{14}$	(0,-0.2,0)	$\mathbf{P}_{21}$	(0,-0.2,0.05)

Table 4.3: Visual servoing repeatability results for in the test points using line-based visual servoing (observing 2 legs {1,3}). Measures are in mm

Point	Error [mm]	Point	Error [mm]	Point	Error [mm]
$\mathbf{P}_1$	0.4797	$\mathbf{P}_8$	0.5342	$\mathbf{P}_{15}$	0.5831
$\mathbf{P}_2$	0.5521	$\mathbf{P}_9$	0.5645	$\mathbf{P}_{16}$	0.6104
$\mathbf{P}_3$	0.5480	$\mathbf{P}_{10}$	0.5552	$\mathbf{P}_{17}$	0.5708
$\mathbf{P}_4$	0.5260	$\mathbf{P}_{11}$	0.5736	$\mathbf{P}_{18}$	0.5857
$\mathbf{P}_5$	0.5302	$\mathbf{P}_{12}$	0.5883	$\mathbf{P}_{19}$	0.5975
$\mathbf{P}_6$	0.5141	$\mathbf{P}_{13}$	0.5661	$\mathbf{P}_{20}$	0.5963
$\mathbf{P}_7$	0.4847	$\mathbf{P}_{14}$	0.5653	$\mathbf{P}_{21}$	0.5723

As we see from the results, visual servoing repeatability when using line-based visual servoing (observing 2 legs {1, 3}) are between 0.4797 mm and 0.6104 mm; visual servoing repeatability when using line-based visual servoing (observing 3 legs {1, 3, 5}) are between 0.4408 mm and 0.5611 mm; visual servoing repeatability when using line-based visual servoing (observing 6 legs {1,2,3,4,5,6 }) are between 0.4989 mm and 0.5589 mm; visual servoing repeatability when using image moment visual servoing are between 0.2245 mm and 0.3870 mm. All the standard deviations of the repeatability results are less than 0.08, which proves the stability of the experimental results.

It is clear that image moment visual servoing has a better control performance compared with line-based visual servoing in terms of the accuracy. The edges of robot legs had a far lower performance compared to the image moment. In addition, the experiment results prove that the the selection of the legs to observe affects the final pose accuracy. Observing 3 legs has a better accuracy performance compared with observing 2 legs and

Table 4.4: Visual servoing repeatability results for in the test points using line-based visual servoing (observing 3 legs {1,3,5}). Measures are in mm

Point	Error [mm]	Point	Error [mm]	Point	Error [mm]
$\mathbf{P}_1$	0.4514	$\mathbf{P}_8$	0.4733	$\mathbf{P}_{15}$	0.4610
$\mathbf{P}_2$	0.5017	$\mathbf{P}_9$	0.5096	$\mathbf{P}_{16}$	0.5611
$\mathbf{P}_3$	0.4682	$\mathbf{P}_{10}$	0.5108	$\mathbf{P}_{17}$	0.5267
$\mathbf{P}_4$	0.4736	$\mathbf{P}_{11}$	0.5012	$\mathbf{P}_{18}$	0.5493
$\mathbf{P}_5$	0.4716	$\mathbf{P}_{12}$	0.5470	$\mathbf{P}_{19}$	0.5561
$\mathbf{P}_6$	0.4408	$\mathbf{P}_{13}$	0.4562	$\mathbf{P}_{20}$	0.4865
$\mathbf{P}_7$	0.4567	$\mathbf{P}_{14}$	0.5075	$\mathbf{P}_{21}$	0.4813

Table 4.5: Visual servoing repeatability results of the test points using line-based visual servoing (observing 6 legs {1,2,3,4,5,6 }). Measures are in mm

Point	Error [mm]	Point	Error [mm]	Point	Error [mm]
$\mathbf{P}_1$	0.4989	$\mathbf{P}_8$	0.5380	$\mathbf{P}_{15}$	0.5303
$\mathbf{P}_2$	0.5080	$\mathbf{P}_9$	0.5302	$\mathbf{P}_{16}$	0.5349
$\mathbf{P}_3$	0.5071	$\mathbf{P}_{10}$	0.5401	$\mathbf{P}_{17}$	0.5589
$\mathbf{P}_4$	0.5057	$\mathbf{P}_{11}$	0.5539	$\mathbf{P}_{18}$	0.5309
$\mathbf{P}_5$	0.5161	$\mathbf{P}_{12}$	0.5367	$\mathbf{P}_{19}$	0.5124
$\mathbf{P}_6$	0.5268	$\mathbf{P}_{13}$	0.5302	$\mathbf{P}_{20}$	0.5388
$\mathbf{P}_7$	0.5072	$\mathbf{P}_{14}$	0.5473	$\mathbf{P}_{21}$	0.5473

6 legs.

Since the hypothesis of a noise of 1 pixel in the co-simulation in Section 3.6 was largely overestimated, we reperformed the co-simulations of DELTA robot with a noise of 0.6 pixels for line-based visual servoing and a noise of 0.5 pixels for image moment visual servoing. In addition, the parameters of the camera in the co-simulation were changed to be the same as the camera we used in the experiment. Desired points in the co-simulation are the same as in the experiment. The co-simulation results are shown in Tabs . 4.11 to 4.14 and Figs. 4.30 to 4.33.

From the co-simulation results, we see that the positioning errors when using image moment visual servoing are from 0.1712 mm to 0.3396mm. When using line-based visual servoing, the positioning errors are from 0.5532 mm to 0.6903 mm (observing 2 legs 1,3), from 0.5007 mm to 0.6531 mm (observign 3 legs 1,3,5), from 0.4507 mm to 0.6600 mm

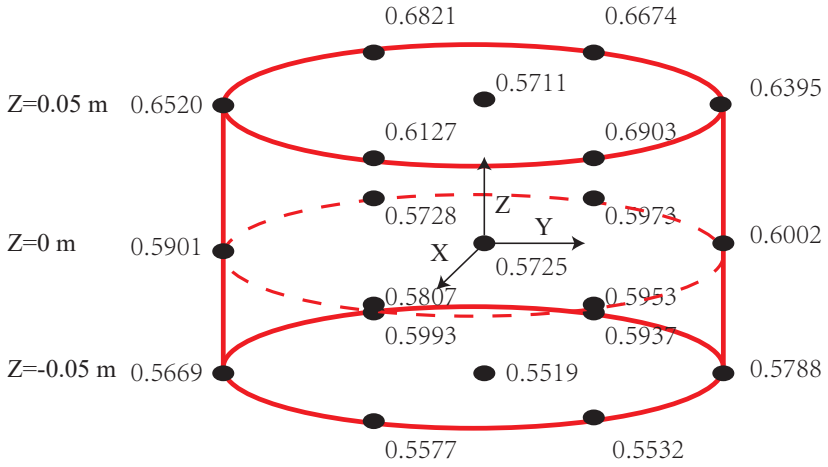

Figure 4.30: Co-simulation positioning errors of the test points using line-based visual servoing (observing 2 legs {1,3}). Measures are in mm

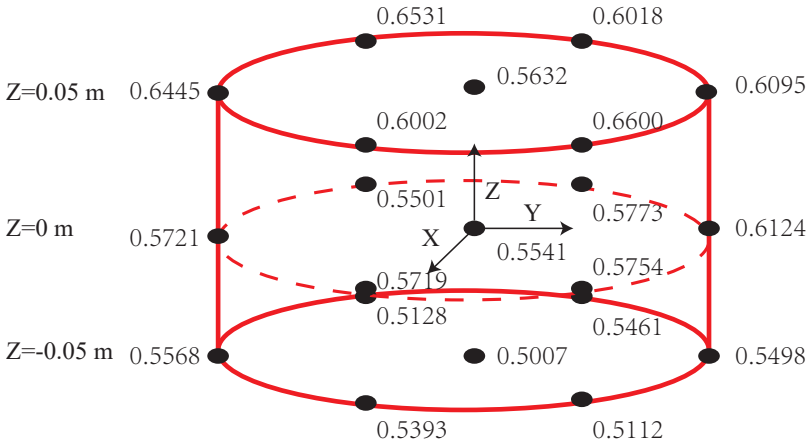

Figure 4.31: Co-simulation positioning errors of the test points using line-based visual servoing (observing 3 legs {1,3,5}). Measures are in mm

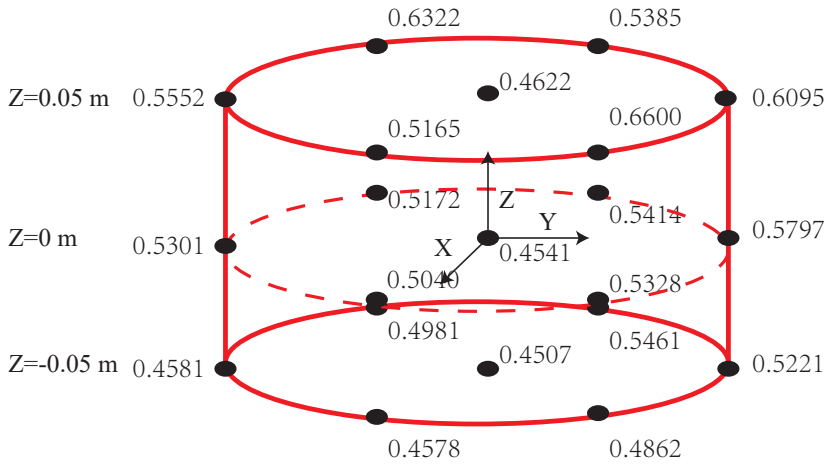

Figure 4.32: Co-simulation positioning errors of the test points using line-based visual servoing (observing 6 legs {1,2,3,4,5,6}). Measures are in mm

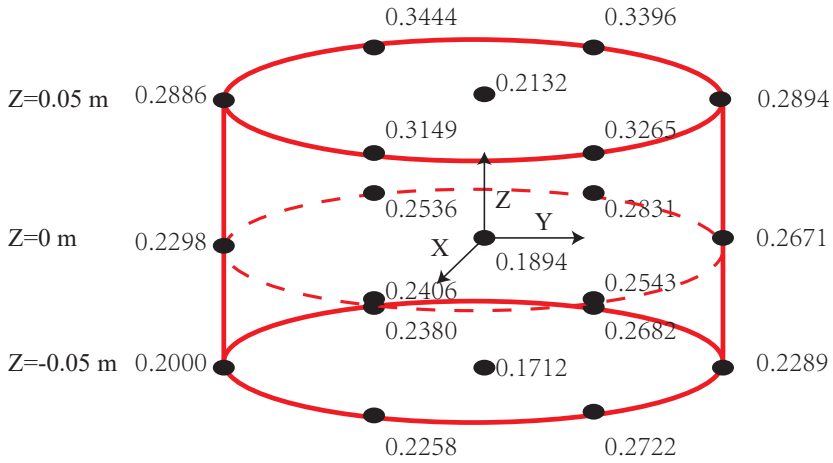

Figure 4.33: Co-simulation positioning errors of the test points using image moment visual servoing. Measures are in mm

Table 4.6: Visual servoing repeatability results for in the test points using image moment visual servoing. Measures are in mm

Point	Error [mm]	Point	Error [mm]	Point	Error [mm]
$\mathbf{P}_1$	0.2245	$\mathbf{P}_8$	0.2423	$\mathbf{P}_{15}$	0.2439
$\mathbf{P}_2$	0.2583	$\mathbf{P}_9$	0.3675	$\mathbf{P}_{16}$	0.3111
$\mathbf{P}_3$	0.2619	$\mathbf{P}_{10}$	0.2979	$\mathbf{P}_{17}$	0.3731
$\mathbf{P}_4$	0.2792	$\mathbf{P}_{11}$	0.3157	$\mathbf{P}_{18}$	0.3252
$\mathbf{P}_5$	0.2676	$\mathbf{P}_{12}$	0.3714	$\mathbf{P}_{19}$	0.3870
$\mathbf{P}_6$	0.2736	$\mathbf{P}_{13}$	0.2840	$\mathbf{P}_{20}$	0.3469
$\mathbf{P}_7$	0.2643	$\mathbf{P}_{14}$	0.2918	$\mathbf{P}_{21}$	0.3563

Table 4.7: Standard deviations (Std) of the visual servoing repeatability results of the test points using line-based visual servoing (observing 2 legs {1,3}). Measures are in mm

Point	Std [mm]	Point	Std [mm]	Point	Std [mm]
$\mathbf{P}_1$	0.0416	$\mathbf{P}_8$	0.0521	$\mathbf{P}_{15}$	0.0408
$\mathbf{P}_2$	0.0775	$\mathbf{P}_9$	0.0456	$\mathbf{P}_{16}$	0.0471
$\mathbf{P}_3$	0.0548	$\mathbf{P}_{10}$	0.0478	$\mathbf{P}_{17}$	0.0309
$\mathbf{P}_4$	0.0519	$\mathbf{P}_{11}$	0.0416	$\mathbf{P}_{18}$	0.0352
$\mathbf{P}_5$	0.0376	$\mathbf{P}_{12}$	0.0505	$\mathbf{P}_{19}$	0.0442
$\mathbf{P}_6$	0.0394	$\mathbf{P}_{13}$	0.0583	$\mathbf{P}_{20}$	0.0304
$\mathbf{P}_7$	0.0419	$\mathbf{P}_{14}$	0.0366	$\mathbf{P}_{21}$	0.0319

(observing 6 legs 1,2,3,4,5,6). The positioning accuracy when applying the image moment visual servoing is far below the desired objective of 0.5 mm while the positioning accuracy when using line-based visual servoing is around 0.5 mm. This results confirm the results obtained from the prototype's experiments that the positioning accuracy is around 0.38 mm for image moment visual servoing and 0.5 mm for line-based visual servoing. In the co-simulation, the increment of the number of the legs observed improves the accuracy, which is different from the experiment results.

The experiments above provide the conclusion that image moment visual servoing is the better controller for the control of DELTA robot compared with line-based visual servoing while it has the better accuracy performance.

Table 4.8: Standard deviations (Std) of the visual servoing repeatability results of the test points using line-based visual servoing (observing 3 legs {1,3,5}). Measures are in mm

Point	Std [mm]	Point	Std [mm]	Point	Std [mm]
$\mathbf{P}_1$	0.0241	$\mathbf{P}_8$	0.0415	$\mathbf{P}_{15}$	0.0573
$\mathbf{P}_2$	0.0869	$\mathbf{P}_9$	0.0622	$\mathbf{P}_{16}$	0.0632
$\mathbf{P}_3$	0.0339	$\mathbf{P}_{10}$	0.0624	$\mathbf{P}_{17}$	0.0338
$\mathbf{P}_4$	0.0512	$\mathbf{P}_{11}$	0.0287	$\mathbf{P}_{18}$	0.0418
$\mathbf{P}_5$	0.0402	$\mathbf{P}_{12}$	0.0557	$\mathbf{P}_{19}$	0.0732
$\mathbf{P}_6$	0.0802	$\mathbf{P}_{13}$	0.0301	$\mathbf{P}_{20}$	0.0460
$\mathbf{P}_7$	0.0474	$\mathbf{P}_{14}$	0.0391	$\mathbf{P}_{21}$	0.0405

Table 4.9: Standard deviations (Std) of the visual servoing repeatability results of the test points using line-based visual servoing (observing 6 legs {1,2,3,4,5,6}). Measures are in mm

Point	Std [mm]	Point	Std [mm]	Point	Std [mm]
$\mathbf{P}_1$	0.0299	$\mathbf{P}_8$	0.0417	$\mathbf{P}_{15}$	0.0570
$\mathbf{P}_2$	0.0383	$\mathbf{P}_9$	0.0376	$\mathbf{P}_{16}$	0.0537
$\mathbf{P}_3$	0.0438	$\mathbf{P}_{10}$	0.0278	$\mathbf{P}_{17}$	0.0464
$\mathbf{P}_4$	0.0482	$\mathbf{P}_{11}$	0.0474	$\mathbf{P}_{18}$	0.0407
$\mathbf{P}_5$	0.0216	$\mathbf{P}_{12}$	0.0376	$\mathbf{P}_{19}$	0.0764
$\mathbf{P}_6$	0.0441	$\mathbf{P}_{13}$	0.0367	$\mathbf{P}_{20}$	0.0600
$\mathbf{P}_7$	0.0616	$\mathbf{P}_{14}$	0.0327	$\mathbf{P}_{21}$	0.0296

4.4 Conclusion

In this chapter, the experimental results obtained in this thesis were presented. In the first section of this chapter, the mechanical design of the prototypes was explained in detail and the hardware and software safety designs were given. Then the instrumentation and communication network was presented. In the second section, the camera identification was done, it was performed in two different steps: firstly the calibration of the intrinsic and extrinsic parameters for the camera were introduced; Secondly, estimation of the measurement noise was performed. As the results show, the hypothesis of a noise of 1 pixel in the co-simulation in Section 3.6 was largely overestimated while the noise introduced by the edges feature in the image plane was quantified as 0.6 pixels, the noise introduced by the image moment feature was quantified as 0.5 pixel. The visual servoing controller

Table 4.10: Standard deviations (Std) of the visual servoing repeatability results of the test points using image moment visual servoing. Measures are in mm

Point	Std [mm]	Point	Std [mm]	Point	Std [mm]
$\mathbf{P}_1$	0.0342	$\mathbf{P}_8$	0.0653	$\mathbf{P}_{15}$	0.0524
$\mathbf{P}_2$	0.0685	$\mathbf{P}_9$	0.0518	$\mathbf{P}_{16}$	0.0445
$\mathbf{P}_3$	0.0344	$\mathbf{P}_{10}$	0.0744	$\mathbf{P}_{17}$	0.0401
$\mathbf{P}_4$	0.0497	$\mathbf{P}_{11}$	0.0499	$\mathbf{P}_{18}$	0.0392
$\mathbf{P}_5$	0.0376	$\mathbf{P}_{12}$	0.0555	$\mathbf{P}_{19}$	0.0357
$\mathbf{P}_6$	0.0744	$\mathbf{P}_{13}$	0.0533	$\mathbf{P}_{20}$	0.0583
$\mathbf{P}_7$	0.0629	$\mathbf{P}_{14}$	0.0589	$\mathbf{P}_{21}$	0.0566

Table 4.11: Co-simulation results of the positioning error for desired points using line-based visual servoing (observing 2 legs {1,3}). Measure in mm

Point	Error [mm]	Point	Error [mm]	Point	Error [mm]
$\mathbf{P}_1$	0.5519	$\mathbf{P}_8$	0.5725	$\mathbf{P}_{15}$	0.5711
$\mathbf{P}_2$	0.5532	$\mathbf{P}_9$	0.5953	$\mathbf{P}_{16}$	0.6903
$\mathbf{P}_3$	0.5937	$\mathbf{P}_{10}$	0.5973	$\mathbf{P}_{17}$	0.6674
$\mathbf{P}_4$	0.5993	$\mathbf{P}_{11}$	0.5728	$\mathbf{P}_{18}$	0.6821
$\mathbf{P}_5$	0.5577	$\mathbf{P}_{12}$	0.5807	$\mathbf{P}_{19}$	0.6127
$\mathbf{P}_6$	0.5788	$\mathbf{P}_{13}$	0.6002	$\mathbf{P}_{20}$	0.6395
$\mathbf{P}_7$	0.5669	$\mathbf{P}_{14}$	0.5901	$\mathbf{P}_{21}$	0.6520

applied in the experiment was defined afterwards. Then, the experiment process were given and we carried out the tests on the visual servoing controllers, the repeatability of the robot was measured. In order to have a wide overview of the prototype's performance, four sets of image features were tested: observing 2 legs ({1,3}), observing 3 legs ({1,3,5}), observing 6 legs ({1,2,3,4,5,6}) when using line-based visual servoing for Prototype 1; coordinates of the centroid point and the area of the ellipse fixed on the end-effector when using image moment visual servoing for Prototype 2. At the same time, co-simulations for DELTA robot were reperformed with the noise recalibrated to be the comparison of the experiment results. The experiment results showed that line-based visual servoing can achieve a precision around 0.5 mm, image moment visual servoing can achieve a precision around 0.38 mm. The co-simulation results confirmed the results regarding the prototype's performance. As a conclusion, image moment visual servoing is the better

Table 4.12: Co-simulation results of the positioning error for desired points using line-based visual servoing (observing 3 legs {1,3,5}). Measure in mm

Point	Error [mm]	Point	Error [mm]	Point	Error [mm]
$\mathbf{P}_1$	0.5007	$\mathbf{P}_8$	0.5541	$\mathbf{P}_{15}$	0.5632
$\mathbf{P}_2$	0.5112	$\mathbf{P}_9$	0.5754	$\mathbf{P}_{16}$	0.6600
$\mathbf{P}_3$	0.5461	$\mathbf{P}_{10}$	0.5773	$\mathbf{P}_{17}$	0.6018
$\mathbf{P}_4$	0.5128	$\mathbf{P}_{11}$	0.5501	$\mathbf{P}_{18}$	0.6531
$\mathbf{P}_5$	0.5393	$\mathbf{P}_{12}$	0.5719	$\mathbf{P}_{19}$	0.6002
$\mathbf{P}_6$	0.5498	$\mathbf{P}_{13}$	0.6124	$\mathbf{P}_{20}$	0.6095
$\mathbf{P}_7$	0.5568	$\mathbf{P}_{14}$	0.5721	$\mathbf{P}_{21}$	0.6445

Table 4.13: Co-simulation results of the positioning error for desired points using line-based visual servoing (observing 6 legs {1,2,3,4,5,6}). Measure in mm

Point	Error [mm]	Point	Error [mm]	Point	Error [mm]
$\mathbf{P}_1$	0.4507	$\mathbf{P}_8$	0.4541	$\mathbf{P}_{15}$	0.4622
$\mathbf{P}_2$	0.4862	$\mathbf{P}_9$	0.5328	$\mathbf{P}_{16}$	0.6600
$\mathbf{P}_3$	0.5461	$\mathbf{P}_{10}$	0.5414	$\mathbf{P}_{17}$	0.5385
$\mathbf{P}_4$	0.4981	$\mathbf{P}_{11}$	0.5172	$\mathbf{P}_{18}$	0.6322
$\mathbf{P}_5$	0.4578	$\mathbf{P}_{12}$	0.5040	$\mathbf{P}_{19}$	0.5165
$\mathbf{P}_6$	0.5221	$\mathbf{P}_{13}$	0.5797	$\mathbf{P}_{20}$	0.6095
$\mathbf{P}_7$	0.4581	$\mathbf{P}_{14}$	0.5301	$\mathbf{P}_{21}$	0.5552

controller for the control of DELTA robot compared with line-based visual servoing.

Table 4.14: Co-simulation results of the positioning error for desired points using image moment visual servoing. Measure in mm

Point	Error [mm]	Point	Error [mm]	Point	Error [mm]
$\mathbf{P}_1$	0.1712	$\mathbf{P}_8$	0.1894	$\mathbf{P}_{15}$	0.2132
$\mathbf{P}_2$	0.2722	$\mathbf{P}_9$	0.2543	$\mathbf{P}_{16}$	0.3265
$\mathbf{P}_3$	0.2682	$\mathbf{P}_{10}$	0.2831	$\mathbf{P}_{17}$	0.3396
$\mathbf{P}_4$	0.2380	$\mathbf{P}_{11}$	0.2536	$\mathbf{P}_{18}$	0.3444
$\mathbf{P}_5$	0.2258	$\mathbf{P}_{12}$	0.2406	$\mathbf{P}_{19}$	0.3149
$\mathbf{P}_6$	0.2289	$\mathbf{P}_{13}$	0.2671	$\mathbf{P}_{20}$	0.2894
$\mathbf{P}_7$	0.2000	$\mathbf{P}_{14}$	0.2298	$\mathbf{P}_{21}$	0.2886

Chapter 5

Optimal design of a Gough-Stewart platform

In this chapter, a control-based design methodology is applied in order to create a Gough-Stewart platform which is a 6 DOF paralllel robot with the best accuracy performance for a dedicated controller. To control the motions of the Gough-Stewart platform, three visual servoing controllers are envisaged : leg-direction-based visual servoing, line-based visual servoing and image moment visual servoing. Firstly, the design specifications, the singularities and the vision-based kinematic of the Gough-Stewart platform are presented. Then, based on these three controllers, positioning error models taking into account the error of observation coming from the camera and the controller singularities are detailed. In the next step, design optimization problems are formulated in order to find the optimal geometric parameters and camera placement for the Gough-Stewart platform for each type of controller. Co-simulations of the robots optimized for the three types of controllers are performed in order to check the accuracy performance of the three robots and the robustness with respect to the manufacturing errors. In the end, the conclusions are drawn.

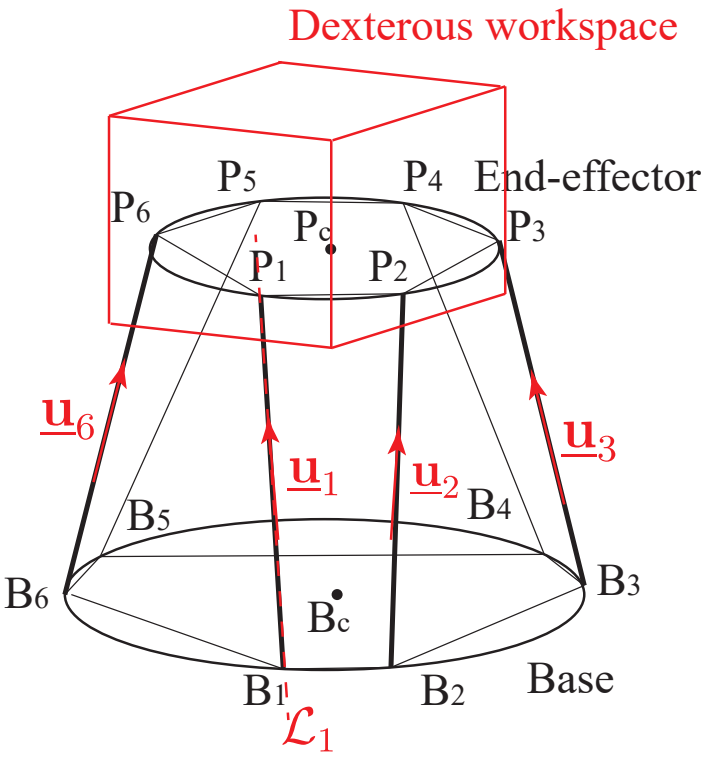

Figure 5.1: Model of a Gough-Stewart platform

5.1 Kinematics and design specifications of the Gough-Stewart platform

The Gough-Stewart platform, also called hexapod, is one of the best-known parallel robot in industry, mainly for motion simulation (e.g. flight simulators). It has 6 degrees of freedom: The end-effector of the Gough-Stewart platform translates along the three axes of the space and rotates around the three axes of the space with respect to the fixed base. The Gough-Stewart platform designed in this chapter is a 6-UPS robot (Fig. 5.1). The platform is linked to the fixed based by 6 chains B_iP_i $(i = 1 \cdots 6)$. The connection of the chains with the base is a U joint located at B_i $(i = 1 \cdots 6)$, the chains are attached to the end-effector by a S joint located at P_i $(i = 1 \cdots 6)$ and the prismatic actuator allows the change of the lengths of the links B_iP_i $(i = 1 \cdots 6)$.

The base and the end-effector of the considered Gough-Stewart platform are symmetric hexagons (See Fig. 5.2). The radius of the circumcircle of the base is r_b, and the radius of the circumcircle of the end-effector is r_a. The angle $\angle B_1B_cB_2 = 2\alpha_1$, the angle $\angle P_1P_cP_2 =$

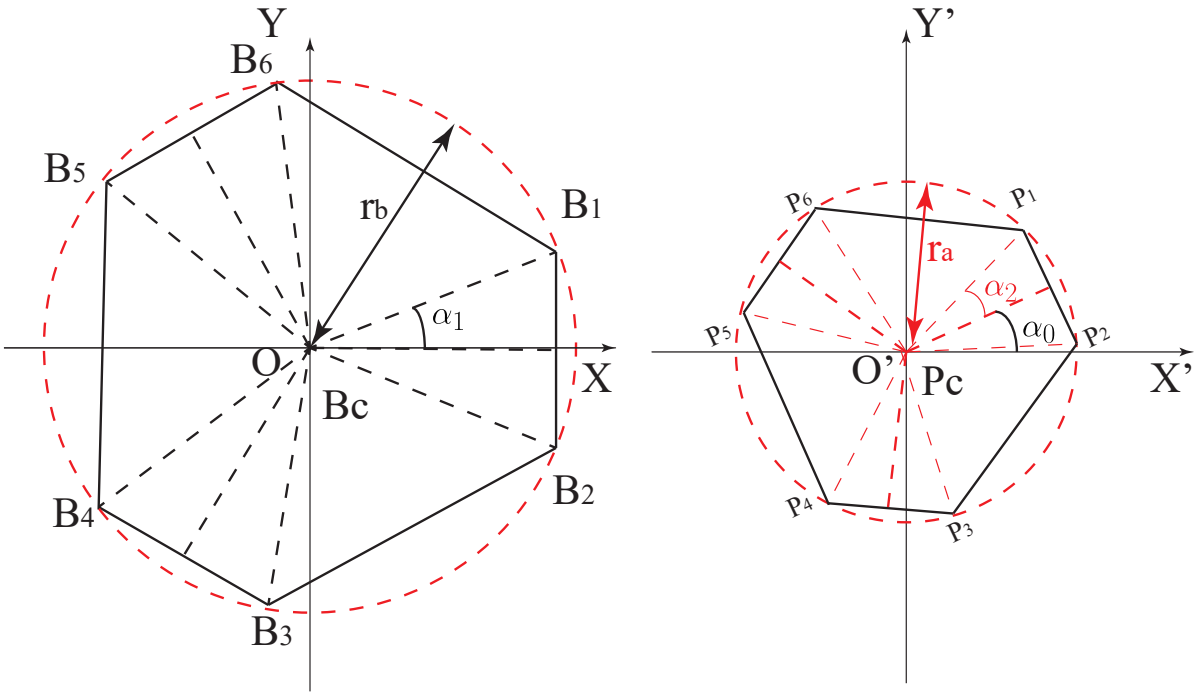

Figure 5.2: Gough-Stewart platform parameters

$2\alpha_2$ and the angle $\angle XP_cP = \alpha_0$ (See Fig. 5.2).

A three-angle orientation called Tilt-and Torsion (T&T) angles was proposed in 1999 [Bonev 1999] to represent the orientation workspace of the Gough-Stewart platform. It was proven that the T&T angles take full advantage of a mechanism's symmetry. The T&T angles are defined in two stages: a tilt and a torsion. In the first stage, as illustrated in Fig. 5.3, the frame i first rotates about the base z_i-axis by an angle ϕ, then about the y_i-axis by an angle θ, then about the z_j-axis by an angle $-\phi$, finally about the new z_k-axis by an angle σ. In T&T angles, ϕ is called azimuth, θ is tilt and σ is torsion. The orientation matrix can be easily obtained as follows:

$$\mathbf{R}(\phi,\theta,\sigma) = \mathbf{R}_z(\phi)\mathbf{R}_y(\theta)\mathbf{R}_z(-\phi)\mathbf{R}_z(\sigma) = \begin{bmatrix} c\phi c\theta c\varphi - s\phi s\varphi & -c\phi c\theta s\varphi - s\phi c\varphi & c\phi s\theta \\ s\phi c\theta c\varphi + c\phi s\varphi & -s\phi c\theta s\varphi + c\phi c\varphi & s\phi s\theta \\ -s\theta c\phi & s\theta s\varphi & c\theta \end{bmatrix} \tag{5.1}$$

where $c\phi = \cos\phi$, $s\phi = \sin\phi$,$c\theta = \cos\theta$, $s\theta = \sin\theta$, $c\varphi = \cos\varphi$, $s\varphi = \sin\varphi$ and $\varphi = \sigma - \phi$.

For any three-angle orientation representation, there are always two triplets of angles that represent a given orientation. In order to avoid this and representational singularity

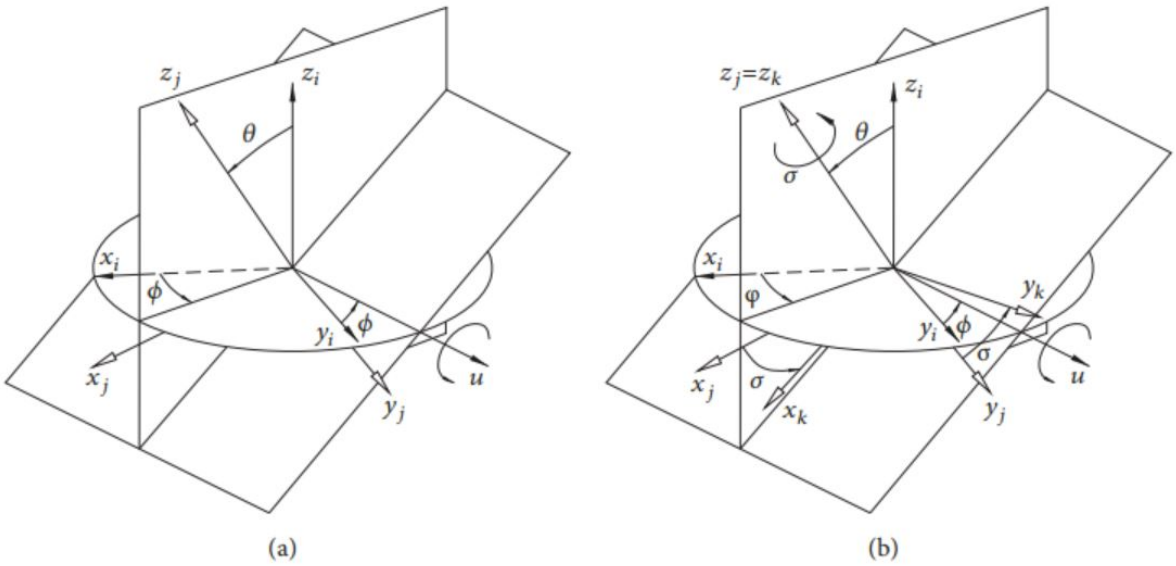

Figure 5.3: Two successive rotations that define the T&T angles: (a)tilt; (b)torsion [Bonev 1999]

at $\theta = \pi$ (it is hardly achieved by any parallel robot) [Blaise 2010], at the same time to keep the orientation workspace of the Gough-Stewart platform to be symmetrical; the ranges of the azimuth, tilt and torsion are thus defined respectively as, $\phi \in (-\pi, \pi]$, $\theta \in [0, \pi/12]$, and $\sigma \in [0, \pi/12]$.

The fixed global frame OXYZ is attached to the base of the Gough-Stewart platform and the origin O is located at the center of the base. Similarly, the mobile frame O'X'Y'Z' is attached to the moving platform by selecting the center of the end-effector as the origin O' (Fig. 5.2). $\mathbf{B}_i^G = [x_{bi}, y_{bi}, z_{bi}]^T$ $(i = 1 \cdots 6)$ denotes the position of the point $\mathbf{B}_i$ with respect to the global frame. $\mathbf{P}_i^L = [x_{pi}, y_{pi}, z_{pi}]^T$ $(i = 1 \cdots 6)$ denotes the position of the point $\mathbf{P}_i$ $(i = 1 \cdots 6)$ with respect to the mobile frame. Let vector $\mathbf{T} = [x_t, y_t, z_t]^T$ denotes the position of the origin O' of the mobile frame in the global fixed frame. Then, we have

$$\mathbf{P}_i^G = \mathbf{T} + \mathbf{R}(\phi, \theta, \sigma)\mathbf{P}_i^L \qquad (i = 1 \cdots 6) \tag{5.2}$$

where $\mathbf{P}_i^G$ is the position of the point $\mathbf{P}_i$ $(i = 1 \cdots 6)$ with respect to the global frame.

The length L_i of leg is the distance between points $\mathbf{B}_i$ and $\mathbf{P}_i$ and we have

$$L_i^2 = (\mathbf{P}_i^G - \mathbf{B}_i^G)^T(\mathbf{P}_i^G - \mathbf{B}_i^G) \tag{5.3}$$

Thus for a given pose of the Gough-Stewart platform $[x_t, y_t, z_t, \phi, \theta, \sigma]$, by the help of (5.3), we can always get the leg length $L_i (i = 1, 2, \cdots, 6)$.

Differentiating the equation (5.3) with respect to time, we obtain [Liu 1993]

$$L_i \dot{L}_i = (\mathbf{P}_i^G - \mathbf{B}_i^G)^T \dot{\mathbf{P}_i^G} + [\mathbf{R}(\phi, \theta, \sigma)(\mathbf{P}_i^G - \mathbf{T}) \times (\mathbf{P}_i^G - \mathbf{B}_i^G)]^T \boldsymbol{\omega} \tag{5.4}$$

where $\boldsymbol{\omega}$ is the angular velocity of the platform. For all the six legs, the equation (5.4) can be also written in the form:

$$\mathbf{A}\boldsymbol{\tau} = \mathbf{B}\dot{\mathbf{L}} \tag{5.5}$$

where $\dot{\mathbf{L}} = [\dot{L_1}, \dot{L_2}, \cdots, \dot{L_6}]^T$ is the actuator velocities and $\boldsymbol{\tau} = [\dot{\mathbf{T}}^T, \boldsymbol{\omega}^T]^T$ is the Cartesian velocity vector of the platform. $\mathbf{A}$ and $\mathbf{B}$ are two Jacobian matrices. $\mathbf{A} = [\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_6]^T$, with $\mathbf{c}_i = [\overrightarrow{\mathbf{B}_i\mathbf{P}_i}^T \ \ (\mathbf{P}_i^L \times \overrightarrow{\mathbf{B}_i\mathbf{P}_i})^T]$. $\mathbf{B} = \mathbf{diag}(\mathbf{L}_1, \mathbf{L}_2, \cdots, \mathbf{L}_6)$.

The Type 2 singularities of Gough-Stewart platform have been studied decades ago. Hunt found that a singularity occurs when all the lines B_iP_i intersect a common line [Hunt 1978]. Fichter found that a singular configuration is attained when the platform is rotated around an axis orthogonal to the plane of the base by an angle of ± 90 deg [Fichter 1986]. Merlet proposed a more general approach using Grassmann geometry to find the singularity configuration of the Gough-Stewart platform [Merlet 2006]. [Ben 2009] applied the Grassmann-Cayley algebra on studying the singularity of a general class of Gough-Stewart platform to translate the algebraic expressions of the singularities into geometrically meaningful statements. Another way to express the singularity configuration mathematically is to find the conditions that the Jacobian matrix $\mathbf{A}$ is singular, i.e. when $\det(\mathbf{A}) = 0$. In [St 2000], an algorithm based on analytical expression of the determinant of the Jacobian matrix using linear decomposition and cofactor expansion is applied on the study of the singularity of the general Gough-Stewart Platform and a graphical representation of the singularity loci can be obtained.

In [Arakelian 2007] and [Arakelian 2008], a solution was proposed to determine the singularity-free zones of a parallel robot. It is the kinetostatic approach taking account of the force transmission. It is known that when the parallel robot is close to a singular configuration, the stiffness and the quality of motion transmission are lost. Thus a indicator of the quality of motion transmission close to the singular configurations of parallel manipulators should be defined. The pressure angle can be used to evaluate the quality of motion transmission and the method of calculating the pressure angle is presented in detail in [Arakelian 2007] and [Arakelian 2008]. When the pressure angle is close to 90

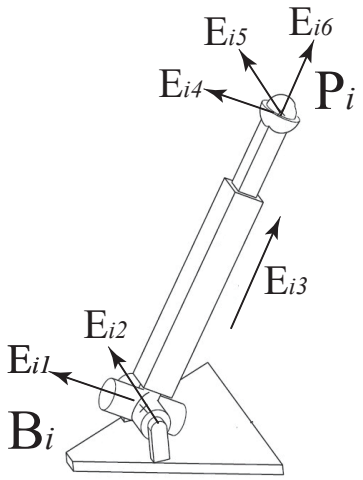

Figure 5.4: UPS robot leg (i-th chain of the Gough-Stewart platform)

deg, the parallel robot is close to a singular configuration.

The way of calculating the pressure angle is detailed presented [Arakelian 2007]. For the Gough-Stewart platform (Fig. 5.1), the actuated prismatic joints $\mathbf{B}_i\mathbf{P}_i$ and passive joints located at $\mathbf{B}_i$ and $\mathbf{P}_i$. The wrench acting to the output link is reciprocal to the unit vectors situated along the axes of non-actuated pairs. Let $\mathbf{E}_{i1}$, $\mathbf{E}_{i2}$, $\mathbf{E}_{i3}$, $\mathbf{E}_{i4}$, $\mathbf{E}_{i5}$, $\mathbf{E}_{i6}$, be the unit vectors of the axes of kinematic pairs, where $i(i = 1, 2, \cdots, 6)$ denotes the number of the chain (Fig. 5.4). Here $\mathbf{E}_{i1}$ and $\mathbf{E}_{i2}$ correspond to two revolute passive pairs of Universal joint, $\mathbf{E}_{i3}$ corresponds to sliding actuated pair, $\mathbf{E}_{i4}$, $\mathbf{E}_{i5}$ and $\mathbf{E}_{i6}$ correspond to the spherical passive pair. Then the Plücker coordinates of these unit screws can be described in matrix $(\mathbf{E})_i$.

$$(\mathbf{E})_i = \begin{bmatrix} e_{i1x} & e_{i1y} & e_{i1z} & e^0_{i1x} & e^0_{i1y} & e^0_{i1z} \\ e_{i2x} & e_{i2y} & e_{i2z} & e^0_{i2x} & e^0_{i2y} & e^0_{i2z} \\ 0 & 0 & 0 & e^0_{i3x} & e^0_{i3y} & e^0_{i3z} \\ e_{i4x} & e_{i4y} & e_{i4z} & e^0_{i4x} & e^0_{i4y} & e^0_{i4z} \\ e_{i5x} & e_{i5y} & e_{i5z} & e^0_{i5x} & e^0_{i5y} & e^0_{i5z} \\ e_{i6x} & e_{i6y} & e_{i6z} & e^0_{i6x} & e^0_{i6y} & e^0_{i6z} \end{bmatrix} \tag{5.6}$$

Here $\mathbf{E}_{i1}$, $\mathbf{E}_{i2}$, $\mathbf{E}_{i4}$, $\mathbf{E}_{i5}$, $\mathbf{E}_{i6}$ are the unit screws of zero pitch, i.e. $e_{i1x} \cdot e^0_{i1x} + e_{i1y} \cdot e^0_{i1y} + e_{i1z} \cdot e^0_{i1z} = 0$, $\mathbf{E}_{i3}$ is the unit screw of infinite pitch, $e^0_{i3x} = (x_{P_i} - x_{B_i})/L_i$, $e^0_{i3y} = (y_{P_i} - y_{B_i})/L_i$, $e^0_{i3z} = (z_{P_i} - z_{B_i})/L_i$, x_{B_i}, y_{B_i}, z_{B_i}, x_{P_i}, y_{P_i}, z_{P_i} are the coordinates of the points $\mathbf{B}_i$ and $\mathbf{P}_i$, L_i is the distance between the points $\mathbf{B}_i$ and $\mathbf{P}_i$, $(i = 1, 2, \cdots, 6)$.

Then we can obtain the wrenches which are reciprocal to the unit vectors of the axes of the passive kinematic pairs. The coordinates of wrenches in the form of the matrix $\mathbf{R}$ can be obtained:

$$(\mathbf{R}) = \begin{bmatrix} r_{1x} & r_{1y} & r_{1z} & r^0_{1x} & r^0_{1y} & r^0_{1z} \\ r_{2x} & r_{2y} & r_{2z} & r^0_{2x} & r^0_{2y} & r^0_{2z} \\ r_{3x} & r_{3x} & r_{3x} & r^0_{3x} & r^0_{3y} & r^0_{3z} \\ r_{4x} & r_{4y} & r_{4z} & r^0_{4x} & r^0_{4y} & r^0_{4z} \\ r_{5x} & r_{5y} & r_{5z} & r^0_{5x} & r^0_{5y} & r^0_{5z} \\ r_{6x} & r_{6y} & r_{6z} & r^0_{6x} & r^0_{6y} & r^0_{6z} \end{bmatrix} \tag{5.7}$$

To find the pressure angles we consider the wrenches $\mathbf{R}_i$ and the directions of the velocities of the points $\mathbf{P}_i$. The velocity of the point is determined by the equations expressing the Plücker coordinates $(w_{1x}, w_{1y}, w_{1z}, v_{1x}, v_{1y}, v_{1z})$ of the twist by fixed the other actuated joints $\mathbf{B}_i\mathbf{P}_i$ $(i = 2, 3, \cdots, 6)$. After a lengthy calculation (see more details in [Arakelian 2007]), we have the V_{P1x}, V_{P1y}, V_{P1z} the coordinates of the velocity $\mathbf{V}_{P1}$ of the point $\mathbf{P}_1$:

$$V_{P1x} = V_{1x} + w_{1y}z_{P1} - w_{1z}y_{P1}, V_{P1y} = V_{1y} + w_{1z}x_{P1} - w_{1x}z_{P1}, V_{P1z} = V_{1z} + w_{1x}y_{P1} - w_{1y}x_{P1}$$

Finally, the pressure angle of leg 1 can be written as:

$$\alpha_1 = \left| \arccos \left[\frac{V_{P1x}r_{11x} + V_{P1y}r_{11y} + V_{P1z}r_{11z}}{\sqrt{V^2_{P1x} + V^2_{P1y} + V^2_{P1z}}\sqrt{r^2_{11x} + r^2_{11y} + r^2_{11z}}} \right] \right| \tag{5.8}$$

We can find five other pressure angles by a similar way. Thus in the design process, we should make sure that the pressure angle of the Gough-Stewart platform for every pose in the dexterous workspace is lower than 80 deg to avoid the singularity configurations.

The requirements that must be achieved by the Gough-Stewart platform in this work are given in Tab. 5.1. The Gough-Stewart platform should cover a regular 3D dexterous workspace and the orientation space. Some geometric and kinematic constraints should be satisfied throughout the regular workspace, such as within the reachable workspace of Gough-Stewart platform, be free of singularity, has a better positioning accuracy compared to the requirements given in Tab. 5.1, thus obtaining a regular dexterous workspace. The shape of regular dexterous workspace is a cube in this work and its desired side length is given in Tab. For practical industrial reasons, in the condition of getting a regular dexterous workspace larger than the size given in Tab. 5.1, the robot must be as compact as

Table 5.1: Requirements of the Gough-Stewart platform

Regular dexterous workspace size (side length of the cube) l_0	$\geqslant$ 100 mm
Orientation workspace defined with the Tilt and Torsion angles	$\phi \in (-\pi,\ \pi]$, $\theta \in [0,\ \pi/12]$, $\sigma \in [0,\ \pi/12]$
Positioning accuracy wherever in regular dexterous workspace	$\leqslant$1 mm
Orientation accuracy wherever in regular dexterous workspace	$\leqslant$0.02 rad

possible.

In order to get the desired 1 mm of positioning accuracy and 0.02 rad angular accuracy specified in Tab. 5.1, we propose to apply visual servoing approaches. We plan to use a single camera (2336 $\times$ 1728 pixels of resolution and a focal length of 10 mm) mounted onto the base platform and the camera image plane is set to be parallel to the base platform in order to control the Gough-Stewart platform. Three types of standard visual servoing approaches will be tested: leg-direction-based visual servoing, line-based visual servoing [Vignolo 2014] and image moment visual servoing [Chaumette 2004]. All these three visual servoing controllers will be applied to the control of a Gough-Stewart platform in order to determine which control method is the best adapted to its accuracy performance.

5.2 Vision-based kinematic of Gough-Stewart platform

Vision-based kinematic model of Gough-Stewart platform aims at finding the relationship between the velocity of the moving platform and the time variation of the image features.

5.2.1 Vision-based kinematics of a Gough-Stewart platform using the leg-direction-based visual servoing

Leg-direction-based visual servoing developed in [Andreff 2005] is applied on the control of a Gough-Stewart platform. The leg direction $\underline{\mathbf{u}}_i$ extracted from the observation of the

robot leg $\mathbf{B}_i\mathbf{P}_i$ is selected as the feature to do the visual servoing. We have

$$\underline{\mathbf{u}}_i = (\mathbf{P}_i^G - \mathbf{B}_i^G)/L_i \tag{5.9}$$

The inverse Jacobian matrix is

$$\mathbf{J}_{inv} = \begin{bmatrix} \underline{\mathbf{u}}_1^T & (\overrightarrow{\mathbf{P_c}\mathbf{P_1}} \times \underline{\mathbf{u}}_1)^T \\ \vdots & \vdots \\ \underline{\mathbf{u}}_6^T & (\overrightarrow{\mathbf{P_c}\mathbf{P_6}} \times \underline{\mathbf{u}}_6)^T \end{bmatrix} \tag{5.10}$$

$\mathbf{P}_c$ is the center of the moving platform (Fig. 5.2).

From equation (5.2), we have the vision-based kinematics of the Gough-Stewart platform expressed in the global frame

$$L_i\underline{\mathbf{u}}_i = \mathbf{T} + \mathbf{R}(\phi, \theta, \sigma)\mathbf{P}_i^L - \mathbf{B}_i^G \tag{5.11}$$

$$\dot{\underline{\mathbf{u}}}_i = \frac{1}{L_i}\frac{1}{\mathrm{d}t}\overrightarrow{\mathbf{P}_i\mathbf{B}_i} - \frac{\dot{L}_i}{L_i}\underline{\mathbf{u}}_i \tag{5.12}$$

Inserting the interaction matix associated to a 3D point [Martinet 1996], we have

$$\dot{\underline{\mathbf{u}}}_i = -\frac{1}{L_i}[\mathbf{I}_3 - [\mathbf{B}_i]_\times]\boldsymbol{\tau} - \frac{\dot{L}_i}{L_i}\underline{\mathbf{u}}_i \tag{5.13}$$

where $[\cdots]_\times$ is the antisymmetric matrix associated to the cross product [Martinet 1996]. With the help of the inverse Jacobian matrix, we can obtain the relationship between each $\dot{\underline{\mathbf{u}}}_i$ and $\boldsymbol{\tau}$.

$$\dot{\underline{\mathbf{u}}}_i = \mathbf{M}_i^T\boldsymbol{\tau} \tag{5.14}$$

$$\mathbf{M}_i^T = -\frac{1}{L_i}(\mathbf{I}_3 - \underline{\mathbf{u}}_i\underline{\mathbf{u}}_i^T)[\mathbf{I}_3 - [\mathbf{B}_i]_\times] \tag{5.15}$$

where $\mathbf{M}_i^T$ is of rank 2 [Andreff 2005]. To fully control the six DOFs of the Gough-Stewart platform, a minimum of three independent legs should be observed and the interaction matrix $\mathbf{M}^T$ is obtained by stacking $\mathbf{M}_i^T, (i = 1, 2, \cdots, 6)$ of k legs $(k = 3, 4, 5, 6)$.

5.2.2 Vision-based kinematics of a Gough-Stewart platform using the line-based visual servoing

Line-based visual servoing kinematic aims finding the relationship between the time variation of the Plücker coordinates $(\underline{\mathbf{u}}_i, \mathbf{h}_i)$ of the robot legs and the twist of its platform.

From the definition of the Plücker coordinates, we have $\mathbf{h}_i = \mathbf{D} \times \underline{\mathbf{u}}_i$ where $\mathbf{D}$ is the position of any point on the line passing through the center of the cylindrical leg. Then we have

$$\dot{\mathbf{h}}_i = \dot{\mathbf{D}} \times \underline{\mathbf{u}}_i + \mathbf{D} \times \dot{\underline{\mathbf{u}}}_i \tag{5.16}$$

For the Gough-Stewart platform, the U joints $\mathbf{B}_i$ $(i = 1, 2 \cdots 6)$ (Fig. 5.1) are all fixed on the base platform, then the equation (5.16) can be written in the form

$$\dot{\mathbf{h}}_i = \dot{\mathbf{B}}_i \times \underline{\mathbf{u}}_i + \mathbf{B}_i \times \dot{\underline{\mathbf{u}}}_i = \mathbf{B}_i \times \dot{\underline{\mathbf{u}}}_i \tag{5.17}$$

With the help of equation (5.14), the equation (5.17) can be written in the matrix form

$$\dot{\mathbf{h}}_i = [\mathbf{B}_i]_\times \times \mathbf{M}_i^T \boldsymbol{\tau} = \mathbf{M}_{hi}^T \boldsymbol{\tau} \tag{5.18}$$

$\mathbf{M}_i$ is the interaction matrix related to the vector $\underline{\mathbf{u}}_i$.

Therefore, for a line $\mathcal{L}_i$, we have

$$\begin{bmatrix} \dot{\underline{\mathbf{u}}}_i \\ \dot{\mathbf{h}}_i \end{bmatrix} = \begin{bmatrix} \mathbf{M}_i^T \\ \mathbf{M}_{hi}^T \end{bmatrix} \boldsymbol{\tau} = \mathbf{M}_{uhi}^T \boldsymbol{\tau} \tag{5.19}$$

Where the matrix $\mathbf{M}_{uhi}$ is of rank 2. To fully control the 6 DOFs of the Gough-Stewart platform, observing a minimum of three independent legs is necessary. Then we have the interaction matrix $\mathbf{M}_{uh}^T$ by stacking the matrices $\mathbf{M}_{uhi}^T$ of k legs $(k = 3, 4, 5, 6)$.

The line-based visual servoing controller applied in this case is defined in Section 1.2.3, equation (1.47).

By studying the equation (5.18), Since the location of $\mathbf{B}_i$ $(i = 1, 2, \cdots, 6)$ are fixed, the components of matrices $[\mathbf{B}_i]_\times$ are constant. Therefore, the velocities of $\mathbf{h}_i$ and the velocities of $\underline{\mathbf{u}}_i$ are linearly dependent. This would mean that for the Gough-Stewart platform, leg-direction-based visual servoing and line-based visual servoing may lead to the same controller performance, at least in terms of sinularity.

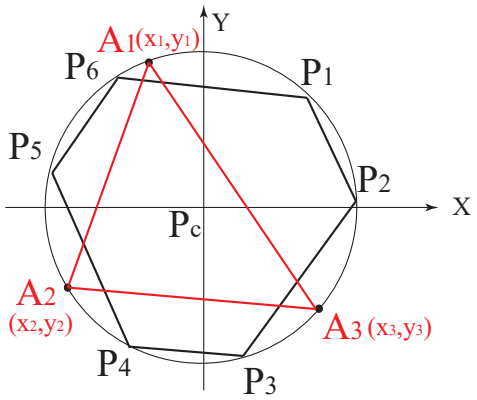

Figure 5.5: Discrete model composed of three points for image moment visual servoing

5.2.3 Vision-based kinematics of a Gough-Stewart platform using the image moment visual servoing

For image moment visual servoing, a picture fixed on the end-effector of the parallel robot is the target to be observed and the image moment is got from the image acquired from the camera. In this work, a discrete model composed of three points ($\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{A}_3$) is chosen to be the object to be observed (Fig. 5.5).

The Gough-Stewart platform has 6 DOFs, the end-effector moves along the three axes of the space and rotates around the three axes of the space with respect to the fixed base. To fully control this parallel robot, a (6×6) full rank interaction matrix should be used in the visual servoing, thus six image moment visual features should be selected[Tahri 2005]: the coordinates x_g, y_g of the center of gravity of the discrete model, the area a of the triangle composed of the three points in image plane, α the orientation of the discrete model in the image, c_1, c_2 two combinations of moments invariant to scale (see definitions in Sec. 1.2.4). Then we have

$$\boldsymbol{\tau} = \mathbf{L}_m^{+}\dot{\mathbf{m}} \tag{5.20}$$

Where $\boldsymbol{\tau}$ is the twist of the end-effector of the Gough-Stewart platform, $\dot{\mathbf{m}} = [\dot{x}_g, \dot{y}_g, \dot{a}, \dot{\alpha}, \dot{c}_1, \dot{c}_2]^T$ are the time derivatives of six image features observed. $\mathbf{L}_m = [L_{x_g}, L_{y_g}, L_a, L_\alpha, L_{c_1}, L_{c_2}]^T$ (see definitions in Sec. 1.2.4) [Chaumette 2004].

The image moment visual servoing controller applied in this case is defined in Section 1.2.4, equation (1.70).

5.3 Positioning error models of the Gough-Stewart platform

When leg-direction-based visual servoing and line-based visual servoing are considered, the positioning error models are similar to the models presented in Section 2.4, the errors added on the intersection points of image boundary and the observed lines are ± 0.5 pixel. However, when image moment visual servoing is considered, the positioning error model is different.

As it is presented in Section 1.2.4, the image moment is calculated based on the coordinates of the points belonging to the projection on the image plane of the object observed. We set $(x_{1p}, y_{1p}), (x_{2p}, y_{2p}), (x_{3p}, y_{3p})$ to be the coordinates of the projection of the three points $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3$ (Fig. 5.5) in pixel. Then we have

$$\frac{\partial \mathbf{m}}{\partial t} = \frac{\partial \mathbf{m}}{\partial \mathbf{P}} \frac{\partial \mathbf{P}}{\partial t} = \mathbf{M} \frac{\partial \mathbf{P}}{\partial t} \tag{5.21}$$

where $\mathbf{P} = [x_{1p}, x_{2p}, x_{3p}, y_{1p}, y_{2p}, y_{3p}]^T$, the matrix $\mathbf{M}$ is the matrix which transforms the time derivatives of the image moments $\mathbf{m}$ to the time derivatives of the coordinates of the points projected to the pixel plane.

Then, the equation (5.20) can be written as the form

$$\boldsymbol{\tau} = \mathbf{L}_m^+ \dot{\mathbf{m}} = \mathbf{L}_m^+ \mathbf{M} \dot{\mathbf{P}} \tag{5.22}$$

We add a noise which is a random shift in the pixels on the coordinates of points projected to the pixel plane (see Fig. 5.6). Since the error of observation is very small, the error model relating the variation of end-effector $\boldsymbol{\delta}\mathbf{x}$ to the variation $\boldsymbol{\delta}\mathbf{P}$ of the points can be approximated by using the first-order geometric model based on the use of the interaction matrix given in formula (5.22):

$$\boldsymbol{\delta}\mathbf{x} = \mathbf{L}_m^+ \mathbf{M} \boldsymbol{\delta}\mathbf{P} \tag{5.23}$$

Since the Gough-Stewart platform has six DOFs, thus $\boldsymbol{\delta}\mathbf{x} = [\delta t_x, \delta t_y, \delta t_z, \delta w_x, \delta w_y, \delta w_z]$. $[\delta t_x, \delta t_y, \delta t_z]$ are the translation errors along the three axes and $[\delta w_x, \delta w_y, \delta w_z]$ are the rotation errors around the three axes. Then the positioning error is defined as

$$E_t = \sqrt{\delta t_x^2 + \delta t_y^2 + \delta t_z^2} \tag{5.24}$$

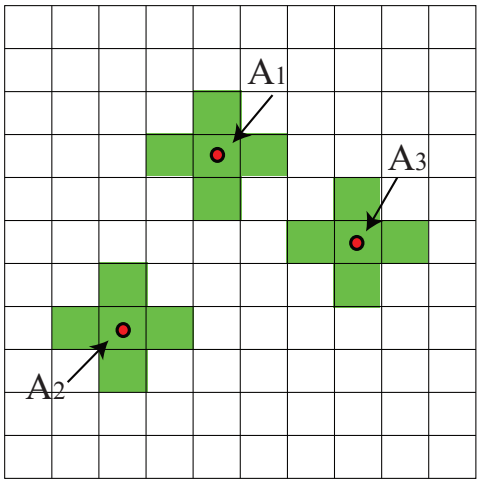

Figure 5.6: Error on the three points discrete model

and the orientation error is defined as

$$E_w = \sqrt{\delta w_x^2 + \delta w_y^2 + \delta w_z^2} \tag{5.25}$$

Every component of vector $\boldsymbol{\delta}\mathbf{P}$ takes the values ± 0.5 in the optimal design process.

5.4 Controller singularity and hidden robot of the Gough-Stewart platform

Singularities of leg-direction-based visual servoing applied to the control of the Gough-Stewart platform have been studied in [Briot 2016]. The Gough-Stewart platform is made of six $U\underline{P}S$ legs and its equivalent hidden robot is made of $\underline{U}PS$ legs (Fig. 5.7). $\underline{U}PS$ legs have 2 degree of actuation, only three legs to be observed are enough to fully control the Gough-Stewart platform when using leg direction observation [Briot 2013].

The singular configurations of $3-\underline{U}PS$-like robots have been deeply studied in [Ben 2006] and [Caro 2010]. Type 2 singularities appear when the planes $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$ (whose normal directions are defined by the vectors $\underline{\mathbf{u}}_1$, $\underline{\mathbf{u}}_2$, $\underline{\mathbf{u}}_3$ and the plane $\mathcal{P}_4$ (passing through the points B_1, B_2, B_3 in Fig. 5.8) intersect in one point (that can be at infinity) (Fig. 5.8).

The concept of the hidden robot is to find what kind of virtual actuators correspond to the features of observation applied in visual servoing. For line-based visual servoing, we take the Plücker coordinates of a line $\mathcal{L}$ as the image feature to be observed and it can

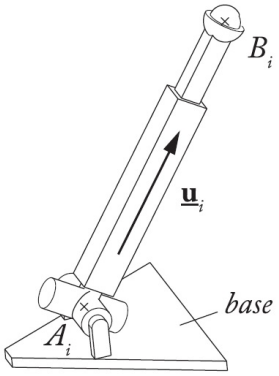

Figure 5.7: UPS leg [Briot 2016]

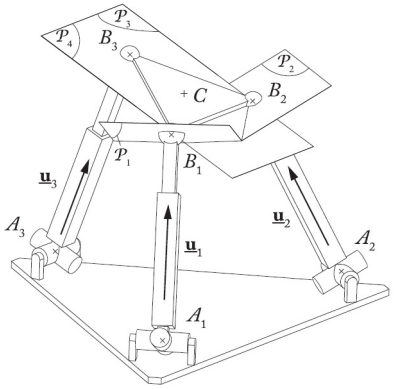

Figure 5.8: Example of a Type 2 singularity for a 3-$\underline{U}PS$ robot: the platform gets an uncontrollable rotation around B_1B_2 [Briot 2016]

be defined from the fact that a 3D point and a 3D orientation define a unique 3D line. Thus we should find the virtual actuators corresponding to the 3D line $\mathcal{L}$.

As we see from Fig. 5.9, let $\mathbf{B}_i$ be the 3D point and $\underline{\mathbf{u}}_i$ the unit vector expressed in the camera frame and $\mathcal{L}_i$ $(i = 1, 2, \cdots, 6)$ is the 3D line they define. The active $\underline{U}$ joint in space is the virtual actuator that makes the vector $\underline{\mathbf{u}}$ move. In general, it is necessary to add the actuated PPP chain on the preceding leg links so that the point $\mathbf{B}_i$ can move in space. Therefore for a $U\underline{P}S$ leg, its corresponding hidden robot when using line-based visual servoing is a $\underline{PPPU}PS$ leg (Fig. 5.9). However, in the case of a Gough-Stewart platform, all the U joints are fixed on the base which means that the points $\mathbf{B}_i$ are fixed in space. Then the actuated PPP chain is no longer needed and the 3D lines $\mathcal{L}_i$ passing through the robot legs can be defined only by the vectors $\underline{\mathbf{u}}_i$. Therefore, the corresponding hidden robot of Gough-Stewart platform is the same as the hidden robot when applying leg-direction-based visual servoing. The controller singularities of line-based visual servoing are the same as the conditions of leg-direction-based visual servoing.

For image moment visual servoing, the controller singularity appears when the matrix $\mathbf{L}_m$ is rank deficient. The expression of the matrix $\mathbf{L}_m$ is rather complex and it is difficult to find the condition of rank deficient analytically. Now let us consider the linear system

$$\mathbf{L}_m \delta \boldsymbol{x} = \delta \mathbf{m} \tag{5.26}$$

where $\mathbf{L}_m$ is a (6×6) interaction matrix. A possible error amplification factor for this

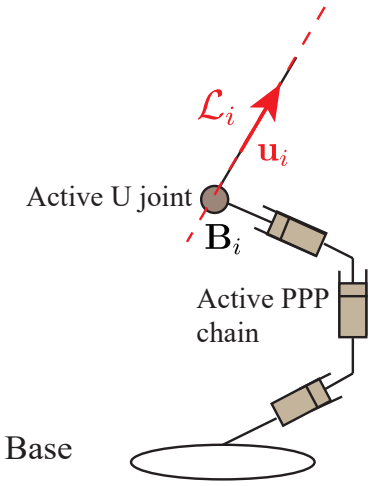

Figure 5.9: Hidden robot leg when the line $\mathcal{L}$ is observed

system expresses how a relative error in image moments $\mathbf{m}$ gets multiplied and leads to a relative error in the movement of the end-effector of the Gough-Stewart platform $\mathbf{x}$. We use a norm such that

$$\|\mathbf{L}_m \delta\mathbf{x}\| \leqslant \|\mathbf{L}_m\| \, \|\delta\mathbf{x}\| \tag{5.27}$$

and obtain

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \leqslant \|\mathbf{L}_m\| \left\|\mathbf{L}_m^{-1}\right\| \frac{\|\delta\boldsymbol{\tau}\|}{\|\boldsymbol{\tau}\|} \tag{5.28}$$

$\|\mathbf{L}_m\| \, \|\mathbf{L}_m^{-1}\|$ is the definition of the condition number of the matrix $\mathbf{L}_m$. The condition number of the interaction matrix is an index to describe the closeness of a pose to a singularity [Voglewede 2004] [Downing 2002] [Merlet 2006]. For the inverse of the condition number, a value of 0 indicates that the matrix is singular. By the standard of the IEEE Standard for Floating-Point Arithmetic (IEEE 754) [Kahan 1996], a matrix is ill-conditioned when its condition number is greater than 10^4 (in this case, it is close to a singularity condition). Thus, we make sure that the value of the inverse of the condition number is always greater than 0.0001 to avoid the singularities.

5.5 Optimal design process of Gough-Stewart platform

The Gough-Stewart platform can be defined by the following geometric parameters: $r_a, r_b, \alpha_0, \alpha_1, \alpha_2$ as we see from Fig. 5.2. When leg-direction-based visual servoing and line-based visual servoing are applied, the radius of the cylindrical distal links of robot also affects the positioning error [Kaci 2018], thus the radius of the cylindrical distal links P_iB_i $(i = 1, 2, \cdots, 6)$, denoted as R (see Fig. 5.1), is a decision variable of the optimization process. The camera image plane is set to be parallel to the base of the Gough-Stewart platform, and the coordinates (x_c, y_c) of the camera frame origin at set at $(0, 0)$ so that all legs can be observed in a symmetrical way. The coordinates of the discrete three points model applied in image moment visual servoing $[x_1, y_1, x_2, y_2, x_3, y_3]$ (in moving platform frame $\mathbf{X'O'Y'}$) defining the configuration of the model (Fig. 5.5). L is the length of the prismatic actuator B_iP_i $(i = 1, 2, \cdots, 6)$ (Fig. 5.1). l_0 is the dimensions of the side length of the cube dexterous workspace [Merlet 2006] (Fig. 5.1) in which the following performance must be satisfied:

- Type 2 singularity-free: Make sure that the pressure angle of each chain of the Gough-Stewart platform is lower than 80 degree throughout the dexterous workspace.

- controller singularity-free: no singularities of the controllers, for leg-based visual servoing, avoid the Type 2 singularities of the corresponding hidden robot; for image moment visual servoing, make sure the inverse of the condition number of the interaction matrix is greater than 0.0001.

- end-effector in image: ensuring that all the robot distal legs can be observed when using leg-based visual servoing, as well as the discrete three points on the end-effector when using image moment visual servoing.

- the discrete three points $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3$ should be within the moving platform of the Gough-Stewart platform.

- required positioning accuracy and angular accuracy: Maximal positioning error computed with the models of Section 5.3 should be lower than 1 mm and the orientation error lower than 0.02 rad (see the definition of the positioning error and orientation error in Sec. 5.3).

The footprint of the Gough-Stewart platform is evaluated by the radius r_b of its base.

As introduced in Sec. 5.4, observing three legs is enough to fully control the Gough-Stewart platform when using leg-direction-based visual servoing and line-based visual servoing. In this work, as a matter of comparison, we will optimize the geometric parameters of the Gough-Stewart platform when observing only three legs (observing legs B_1P_1, B_3P_3, B_5P_5) and observing all the six legs (observing legs B_1P_1, B_2P_2, B_3P_3, B_4P_4, B_5P_5, B_6P_6) for leg-direction-based visual servoing and line-based visual servoing.

In order to create a compact Gough-Stewart platform which has the specifications detailed in Tab. 5.1, the following optimization problem is formulated:

$$
\begin{aligned}
\text{minimize} \quad & r_b \\
\text{over} \quad & \mathbf{x} = [r_a, r_b, \alpha_0, \alpha_1, \alpha_2, z_c, R]^T \\
& \text{(leg-direction-based and line-based visual servoing)} \\
& \mathbf{x} = [r_a, r_b, \alpha_0, \alpha_1, \alpha_2, z_c, x_1, x_2, x_3, y_1, y_2, y_3]^T \\
& \text{(image moment visual servoing)} \\
\text{subject to} \quad & l_0 \geqslant 100 \text{ mm},\ 0.4 \text{ m} \leqslant L \leqslant 0.76 \text{ m} \\
& 0.1 \text{ m} \leqslant r_a \leqslant 0.3 \text{ m} \leqslant r_b \leqslant 0.5 \text{ m},\ r_a < 0.9 \times r_b,\ -\frac{\pi}{6} \leqslant \alpha_0 \leqslant -\frac{\pi}{6}, \\
& 0 \leqslant \alpha_1 \leqslant -\frac{\pi}{9},\ 0 \leqslant \alpha_2 \leqslant -\frac{\pi}{4},\ -0.2 \text{ m} \leqslant z_c \leqslant 0.3 \text{ m},\ 0.01 \text{ m} \leqslant R \leqslant 0.03 \text{ m},
\end{aligned}
\tag{5.29}
$$

The algorithm of calculating the size of the maximal dexterous workspace [Merlet 2006] is presented in [Germain 2013]. The previous optimization algorithm is applied for the design of the mentioned Gough-Stewart platform and the optimal design results are given in Tab. 5.2 and the configuration of the Gough-Stewart platforms are illustrated in Fig. 5.10 to 5.14.

As we see from the results of optimization, in terms of the footprint of the robot, the Gough-Stewart platform designed based on the leg-direction-based visual servoing, line-based visual servoing and image moment visual servoing are close from each other and the differences are almost negligible. For leg-direction-based visual servoing and line-based visual servoing, there is no huge difference of robot size between observing three independent legs (observing legs [1,3,5]) and all the six legs. In the next section, we will perform co-simulations with ADAMS and Simulink to test the robot accuracy performance.

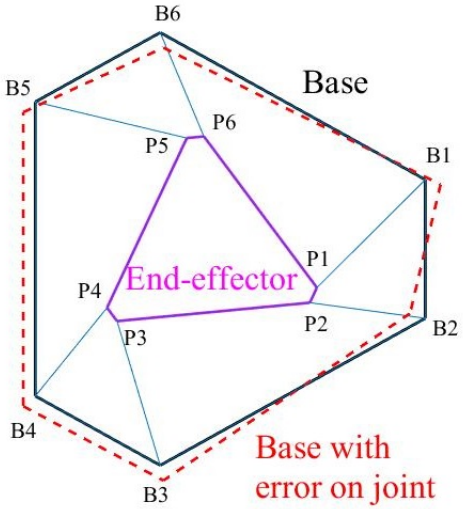

Figure 5.10: Gough-Stewart platform optimized using leng-direction-based visual servoing (observing 3 legs)

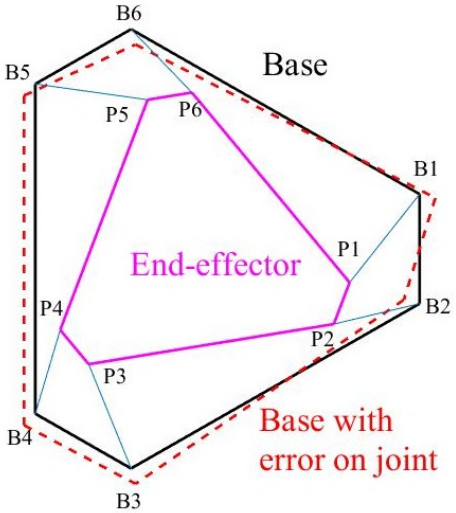

Figure 5.11: Gough-Stewart platform optimized using line-based visual servoing (observing 3 lines)

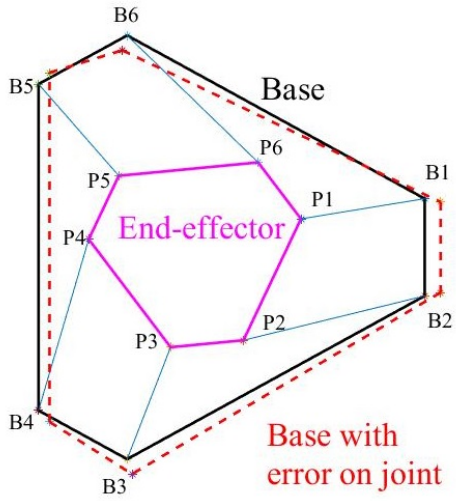

Figure 5.12: Gough-Stewart platform optimized using leg-direction-based visual servoing (observing 6 legs)

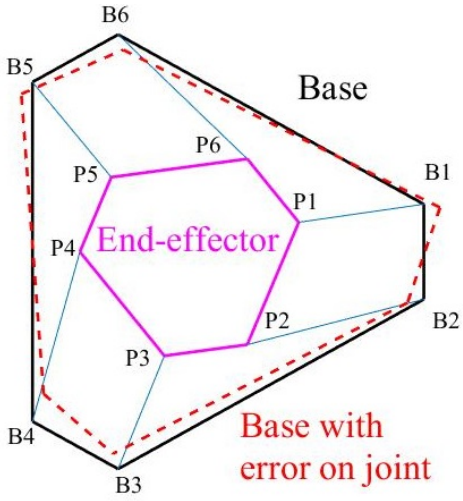

Figure 5.13: Gough-Stewart platform optimized using line-based visual servoing (observing 6 lines)

Table 5.2: Design parameters and value of the objective function as a function of the chosen controller

	leg-direction-based visual servoing (leg [1,3,5])	line-based visual servoing (leg [1,3,5])	leg-direction-based visual servoing (leg [1,2,3,4,5,6])	line-based visual servoing (leg [1,2,3,4,5,6])	image moment visual servoing
r_a[m]	0.1528	0.2058	0.1392	0.1409	0.16
r_b[m]	0.30	0.30	0.30	0.30	0.30
α_0[rad]	-0.4244	-0.3596	-0.4229	-0.3843	0.2668
α_1[rad]	0.3190	0.2418	0.2209	0.2098	0.1986
α_2[rad]	0.0733	0.1424	0.7026	0.6627	0.2406
z_c[m]	-0.0954	-0.0523	-0.0439	-0.0291	0.1204
R[m]	0.0285	0.0197	0.029	0.0199	N/A
x_1[m]	N/A	N/A	N/A	N/A	-0.1311
x_2[m]	N/A	N/A	N/A	N/A	0.1303
x_3[m]	N/A	N/A	N/A	N/A	-0.0933
y_1[m]	N/A	N/A	N/A	N/A	-0.0870
y_2[m]	N/A	N/A	N/A	N/A	-0.0839
y_3[m]	N/A	N/A	N/A	N/A	0.0872
r_b[m]	0.30	0.30	0.30	0.30	0.30

5.6 Co-simulation of Gough-Stewart platform

Co-simulations are performed in a connected ADAMS-Simulink environment. The control scheme is similar with the scheme presented in Section 3.6. We created ten Gough-Stewart platform mechanical models using the results got from the optimal design (Fig. 5.10 to 5.14). Since in reality, there are always the manufacturing and assembly errors on the platform. In order to test the robustness to the manufacturing error, all the co-simulations are performed with both the accurate model and the error-added model.

Real time data (for leg-direction-based visual servoing simulation and line-based visual servoing simulation, the real-time data are the coordinates of the points $\mathbf{P}_i$ and $\mathbf{B}_i$ (Fig. 5.1). For image moment visual servoing simulation, the real-time data are the coordinates of the three points $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{A}_3$ (Fig. 5.5) of the mechanical models are the output of ADAMS and are sent to Simulink to rebuild the image features (leg-directions, Plücker coordinates and image moments) used in visual servoing. In Simulink, the pinhole camera is created (see more details in Section 2.6). For leg-direction-based visual servoing, the controller is defined in Section 1.2.2, equation (1.34). For line-based visual servoing, the controller is defined in Section 1.2.3, equation (1.47). For image moment visual servoing, the controller is defined in Section 1.2.4, equation (1.70). Then 0.5 pixel noise is added

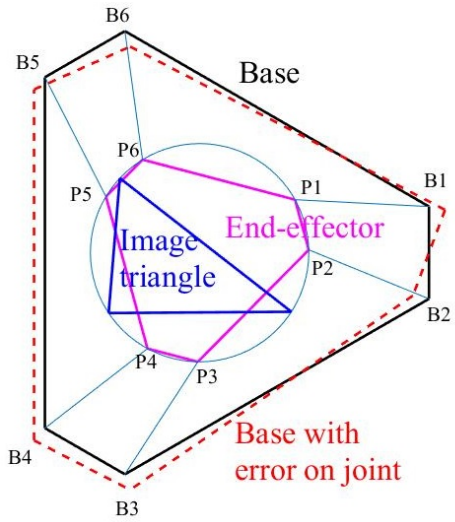

Figure 5.14: Gough-Stewart platform optimized using image moment visual servoing

into the camera simulation: for the case of leg-direction-based visual servoing and line-based visual servoing, the noise is added on the intersection points of image boundary and the lines which are the projections of robot leg edges (see Fig. 2.9), for the case of image moment visual servoing, the noise is added on the three points of the discrete model on the pixel plane, so that controller accuracy performance can be checked. Each robot is controlled with its dedicated visual servoing approach.

In all these simulations, we set a home pose and several desired poses within the dexterous workspace. Then Gough-Stewart platforms perform motions from their home pose to the desired poses and we check their positioning error. The regular dexterous workspace of the Gough-Stewart platform is a cube whose side length is 0.1 m, and the orientation workspace is set based on the (T&T) angles $\phi \in (-\pi\ \pi]$, $\theta \in [0\ \pi/12]$, $\sigma \in [0\ \pi/12]$. We set the home point to be the original point (center of the cube) of the workspace frame whose coordinate is (0,0,0) m (all the coordinates below in the co-simulation are with respect to the workspace frame) (Fig. 5.16). Nine points are chosen to be the desired points, the coordinates of the desired points are given in Tab. 5.3. For each points, three orientation are selected: for $[\phi, \theta, \sigma]^T$, the three orientation poses are : Pose 1 $[0, 0, 0]^T$, Pose 2 $[\pi/2, \pi/12, \pi/12]^T$, Pose 3 $[-\pi/2, \pi/12, \pi/12]^T$ (Fig. 5.16).

For the Gough-Stewart platform optimized for leg-direction-based visual servoing, when observing three legs [1,3,5] and the co-simulations are operated with the accurate mechanical model created in ADAMS, the results are given in Tab. 5.4 and the Fig. 5.17 shows the result when the desired point is P2 and desired pose is Pose2. The positioning

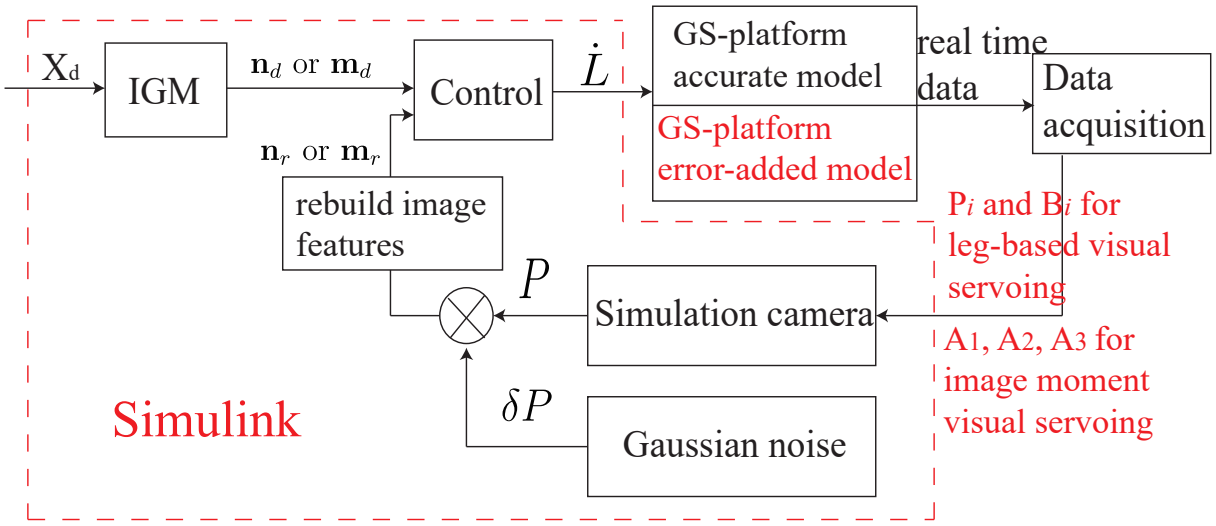

Figure 5.15: Co-simulation control scheme of Gough-Stewart platform

errors are between 0.963 mm and 1.348 mm and the maximal orientation error is 4.7 E-4 rad.

For the Gough-Stewart platform optimized for line-based visual servoing, when observing three legs [1,3,5] and the co-simulations are operated with the accurate mechanical model created in ADAMS, the results are given in Table. 5.6 and the Fig. 5.19 shows the result when the desired point is P2 and desired pose is Pose 2. The positioning errors are between 0.938 mm and 1.358 mm and the maximal orientation error is 4.5 E-4 rad.

For the Gough-Stewart platform optimized for leg-direction-based visual servoing, when observing six legs [1,2,3,4,5,6] and the co-simulations are operated with the accurate mechanical model created in ADAMS, the results are given in Table. 5.8 and the Fig. 5.21 shows the result when the desired point is P2 and desired pose is Pose 2. The positioning errors are between 0.976 mm and 1.432 mm and the maximal orientation error is 4.7 E-4 rad.

For the Gough-Stewart platform optimized for line-based visual servoing, when observing six legs [1,2,3,4,5,6] and the co-simulations are operated with the accurate mechanical model created in ADAMS, the results are given in Table. 5.10 and the Fig. 5.23 shows the result when the desired point is P2 and desired pose is Pose 2. The positioning errors are between 0.912 mm and 1.242 mm and the maximal orientation error is 4.0 E-4 rad.

For the Gough-Stewart platform optimized for image moment visual servoing and the co-simulations are operated with the accurate mechanical model created in ADAMS, the results are given in Table. 5.12 and the Fig. 5.25 shows the result when the desired point

is P2 and desired pose is Pose 2. The positioning errors are between 0.278 mm and 0.627 mm and the maximal orientation error is 4.3 E-4 rad.

In order to test the robustness of the accuracy of model with geometry errors, the same co-simulations were operated with the error added in model. The models we added errors on joints are defined as blow: we add a random error on the location of the joint B_i on the base of the robot, the distance between the accurate joint B_i and the joint with error B_i', denoted as $l_{B_iB_i'}$, ($l_{B_iB_i'} = 0.1 \times r_b$) (see red parts of Fig. 5.10 to 5.14).

For the Gough-Stewart platform optimized for leg-direction-based visual servoing, when observing three legs [1,3,5] and the co-simulations are operated with the error-added mechanical model created in ADAMS, the results are given in Tab. 5.5 and the Fig. 5.18 shows the result when the desired point is P2 and desired pose is Pose 2. The positioning errors are between 0.977 mm and 1.340 mm and the maximal orientation error is 4.7 E-4 rad.

For the Gough-Stewart platform optimized for line-based visual servoing, when observing three legs [1,3,5] and the co-simulations are operated with the error-added mechanical model created in ADAMS, the results are given in Table. 5.7 and the Fig. 5.20 shows the result when the desired point is P2 and desired pose is Pose 2. The positioning errors are between 0.963 mm and 1.331 mm and the maximal orientation error is 4.5 E-4 rad.

For the Gough-Stewart platform optimized for leg-direction-based visual servoing, when observing six legs [1,2,3,4,5,6] and the co-simulations are operated with the error-added mechanical model created in ADAMS, the results are given in Table. 5.9 and the Fig. 5.22 shows the result when the desired point is P2 and desired pose is Pose 2. The positioning errors are between 0.961 mm and 1.345 mm and the maximal orientation error is 4.6 E-4 rad.

For the Gough-Stewart platform optimized for line-based visual servoing, when observing six legs [1,2,3,4,5,6] and the co-simulations are operated with the error-added mechanical model created in ADAMS, the results are given in Table. 5.11 and the Fig. 5.24 shows the result when the desired point is P2 and desired pose is Pose 2. The positioning errors are between 0.998 mm and 1.342 mm and the maximal orientation error is 4.5 E-4 rad.

For the Gough-Stewart platform optimized for image moment visual servoing and the co-simulations are operated with the error-added mechanical model created in ADAMS, the results are given in Table. 5.13 and the Fig. 5.26 shows the result when the desired point is P2 and desired pose is Pose 2. The positioning errors are between 0.297 mm and

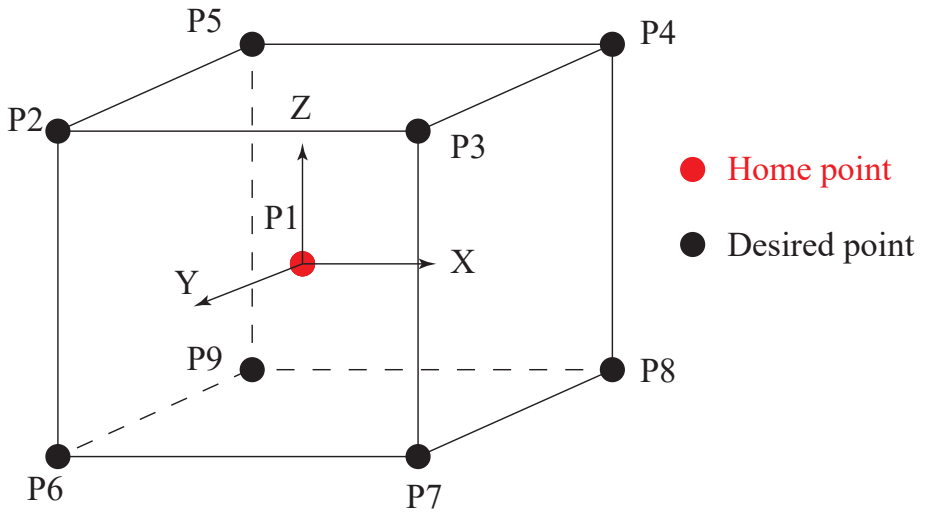

Figure 5.16: Desired points in the regular dexterous workspace for Gough-Stewart platform co-simulation

Table 5.3: Desired points in the co-simulation of Gough-Stewart platform

Point	Coordinate [m]	Point	Coordinate [m]	Point	Coordinate [m]
$\mathbf{P}_1$	(0,0,0)	$\mathbf{P}_4$	(0.05,-0.05,0.05)	$\mathbf{P}_7$	(0.05,0.05,-0.05)
$\mathbf{P}_2$	(-0.05,0.05,0.05)	$\mathbf{P}_5$	(-0.05,-0.05,0.05)	$\mathbf{P}_8$	(0.05,-0.05,-0.05)
$\mathbf{P}_3$	(0.05,0.05,0.05)	$\mathbf{P}_6$	(-0.05,0.05,-0.05)	$\mathbf{P}_9$	(-0.05,-0.05,-0.05)

0.664 mm and the maximal orientation error is 4.6 E-4 rad.

Considering the size of the Gough-Stewart platform and their positioning errors, we see that the robot sizes of the three robots are almost the same. When using the image moments, the controller accuracy is better than leg-direction-based controller and line-based controller. Additionally, the differences between the results operated with the accurate models and error-added models are not significant. When applying leg-direction-based controller and line-based controller, the results of observing three legs [1,3,5] are close to the results of observing six legs [1,2,3,4,5,6].

As we see from the optimization results when using image moment visual servoing, the discrete three points model form a triangle which is not a regular triangle. In order to study why it is such a configuration, we create a discrete three points whose configuration is a regular triangle. The coordinates of the three points (with respect to the moving platform frame $\mathbf{X'O'Y'}$) $\mathbf{A}_{1r}, \mathbf{A}_{2r}, \mathbf{A}_{3r}$ are (0,0.222) m, (0.192,-0.111) m, (-0.192,0.111)

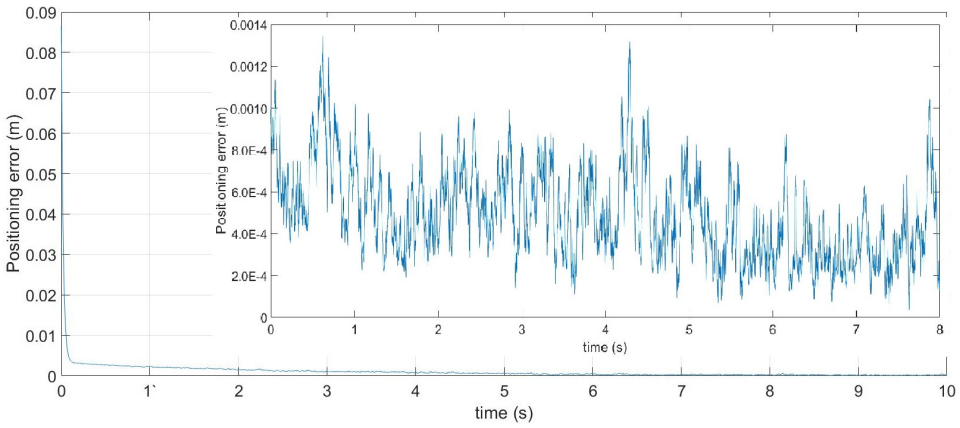

Figure 5.17: Positioning error at point $\mathbf{P}_2$, Pose 2 of the leg-direction-based optimized (observing legs [1,3,5]) design with the accurate model

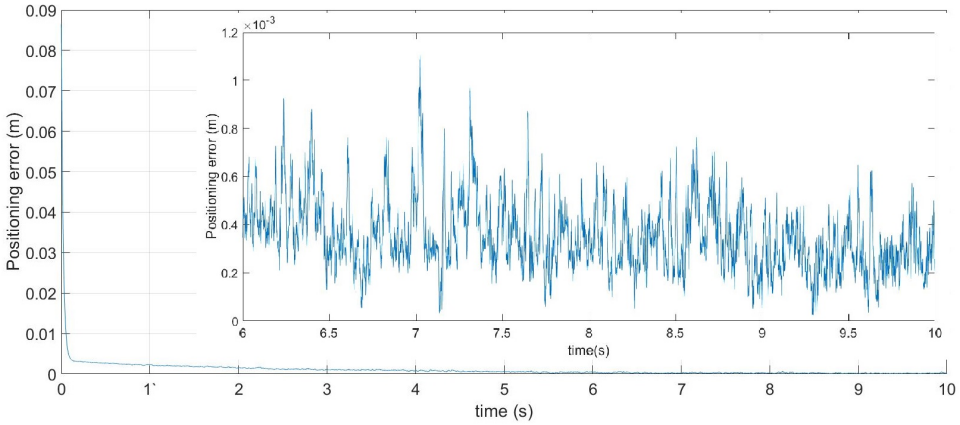

Figure 5.18: Positioning error at point $\mathbf{P}_2$, Pose 2 of the leg-direction-based optimized (observing legs [1,3,5]) design with the error added in model

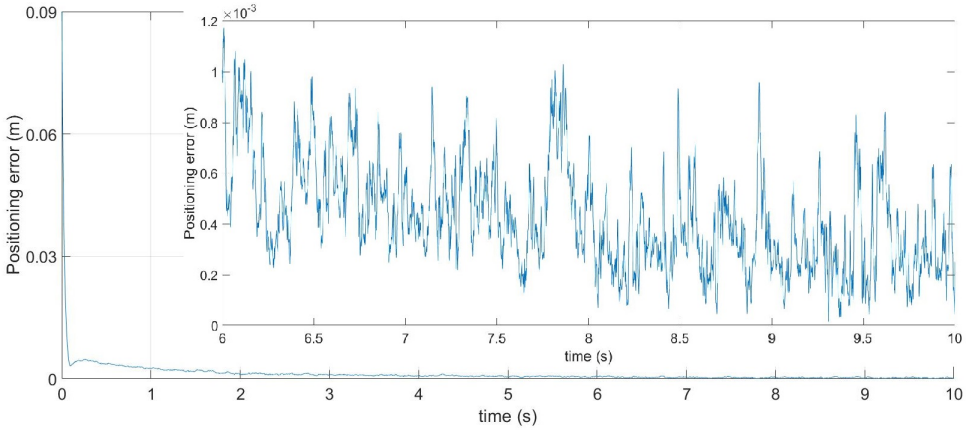

Figure 5.19: Positioning error at point $\mathbf{P}_2$, Pose 2 of the line-based optimized (observing legs [1,3,5]) design with the accurate model

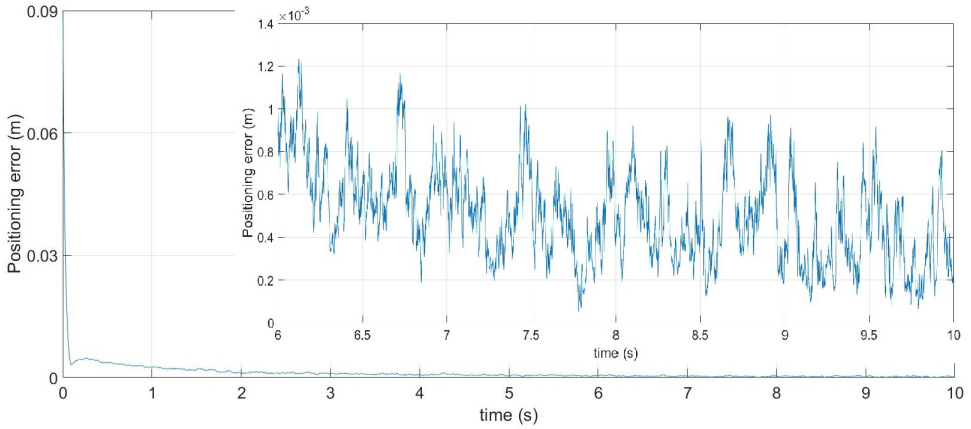

Figure 5.20: Positioning error at point $\mathbf{P}_2$, Pose 2 of the line-based optimized (observing legs [1,3,5]) design with the error added in model

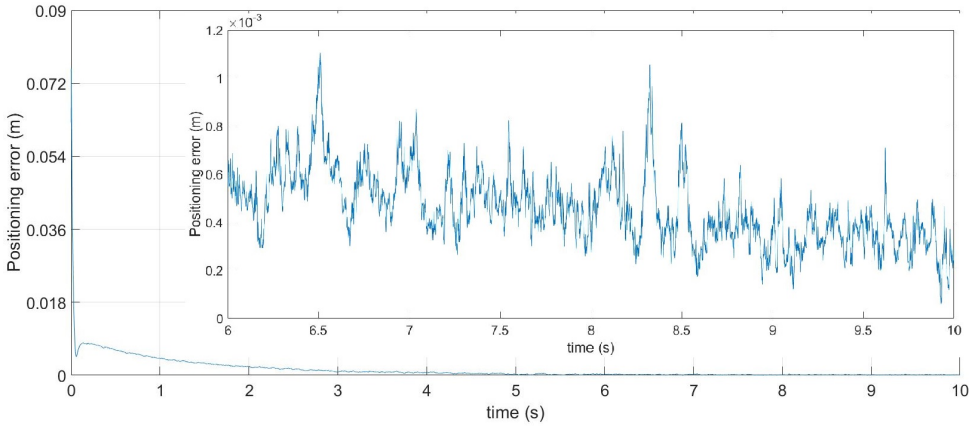

Figure 5.21: Positioning error at point $\mathbf{P}_2$, Pose 2 of the leg-direction-based optimized (observing legs [1,2,3,4,5,6]) design with the accurate model

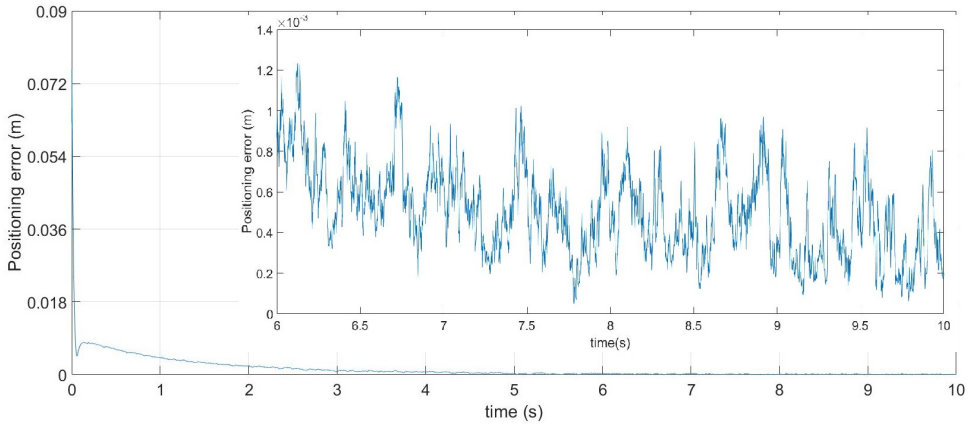

Figure 5.22: Positioning error at point $\mathbf{P}_2$, Pose 2 of the leg-direction-based optimized (observing legs [1,2,3,4,5,6]) design with the error added in model

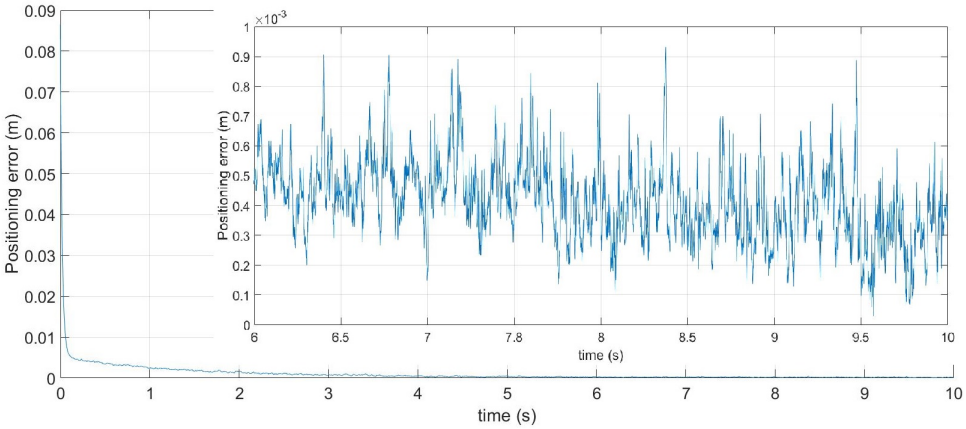

Figure 5.23: Positioning error at point $\mathbf{P}_2$, Pose 2 of the line-based optimized (observing legs [1,2,3,4,5,6]) design with the accurate model

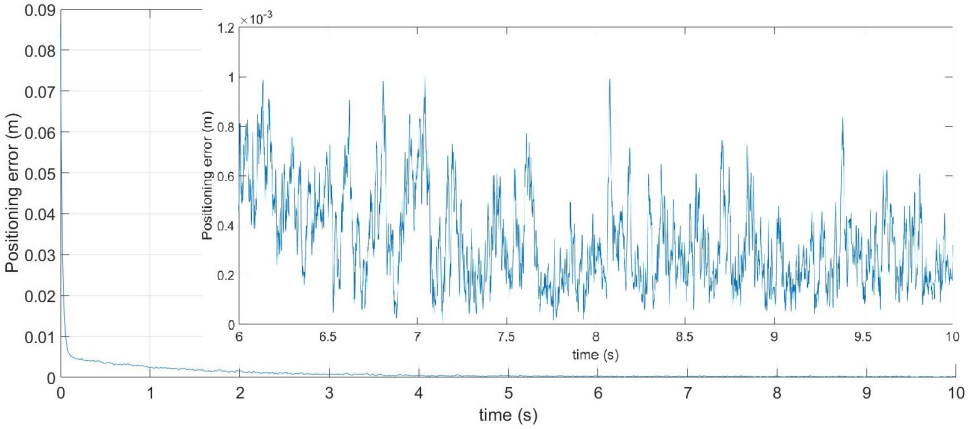

Figure 5.24: Positioning error at point $\mathbf{P}_2$, Pose 2 of the line-based optimized (observing legs [1,2,3,4,5,6]) design with the error added in model

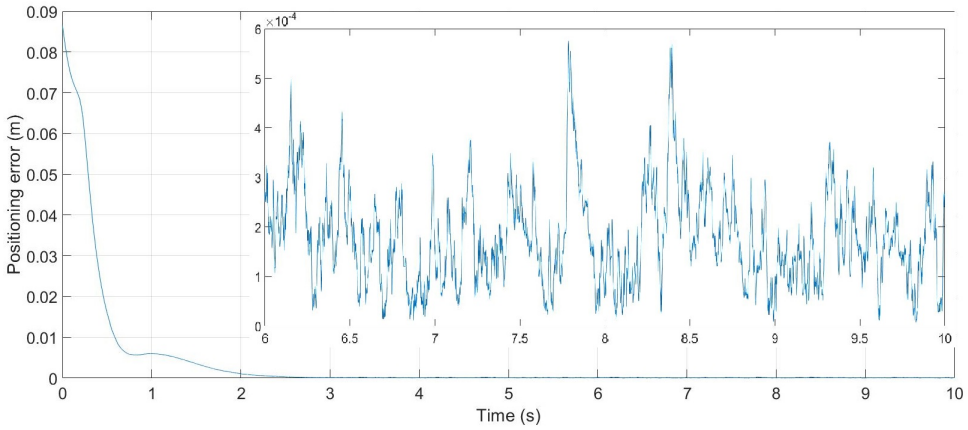

Figure 5.25: Positioning error at point $\mathbf{P}_2$, Pose 2 of the image moment optimized design with the accurate model

Table 5.4: Gough-Stewart platform simulation result of leg-direction-based visual servoing (observing legs [1,3,5]) operated with the accurate model)

Point	Pose 1 Max error[mm]	Pose 1 orientation error [rad]	Pose 2 Max error [mm]	Pose 2 orientation error [rad]	Pose 3 Max error [mm]	Pose 3 orientation error [rad]
$\mathbf{P}_1$	1.012	3.1E-4	1.128	3.4E-4	1.137	2.7E-4
$\mathbf{P}_2$	0.963	4.3E-4	1.348	2.8E-4	0.982	3.1E-4
$\mathbf{P}_3$	0.991	2.7E-4	1.179	4.1E-4	1.220	2.2E-4
$\mathbf{P}_4$	1.123	3.9E-4	1.245	3.0E-4	1.131	2.5E-4
$\mathbf{P}_5$	1.184	3.8E-4	1.251	4.5E-4	1.196	2.0E-4
$\mathbf{P}_6$	1.009	3.4E-4	1.193	4.7E-4	1.224	3.9E-4
$\mathbf{P}_7$	0.994	2.9E-4	1.216	4.3E-4	1.172	3.6E-4
$\mathbf{P}_8$	1.152	2.8E-4	1.123	2.8E-4	1.028	2.8E-4
$\mathbf{P}_9$	1.234	4.5E-4	1.022	3.3E-4	1.153	4.1E-4

m. We add the same noise on the projection of the points to see the variation of the image moments we applied in the visual servoing (see the scheme Fig. 5.27). For the image moments $[x_g, y_g, a]$, the variation of the regular triangle and the triangle we got from the optimization are almost the same. However, for the image moments $[\alpha, c_1, c_2]$, the differences of the variations are huge (Fig. 5.28 to Fig. 5.33). For the same noise, the variations of the image moments $[\alpha, c_1, c_2]$ for the triangle got from the optimization are [0.01,0.08,0.08], while the variations of the image moments $[\alpha, c_1, c_2]$ for regular triangle are [1.6,20,400]. The variations of the image moments $[\alpha, c_1, c_2]$ for the regular triangle are bigger than the variations of the image moments $[\alpha, c_1, c_2]$ for the triangle got from the optimization.

Then we created the mechanical model of Gough-Stewart platform using the results got from the optimization when using the image moment visual servoing, changed the coordinates of the discrete three points into the coordinates of $\mathbf{A}_{1r}, \mathbf{A}_{2r}, \mathbf{A}_{3r}$. Operated the same co-simulations (applying the image moment visual servoing) and the results are shown in Tab. 5.14 and Fig. 5.34). We see that the positioning errors are between 1.031 mm and 1.666 mm and the orientation errors are between 3.7E-4 rad and 5.7E-4 rad.

The positioning errors of image moment visual servoing operated with the regular triangle $\mathbf{A}_{1r}, \mathbf{A}_{2r}, \mathbf{A}_{3r}$ are almost three times the positioning errors of image moment visual servoing operated with the three points we got from the optimization, which proves that the optimization for the coordinates of the three points we used in the image moment helps improve the control accuracy.

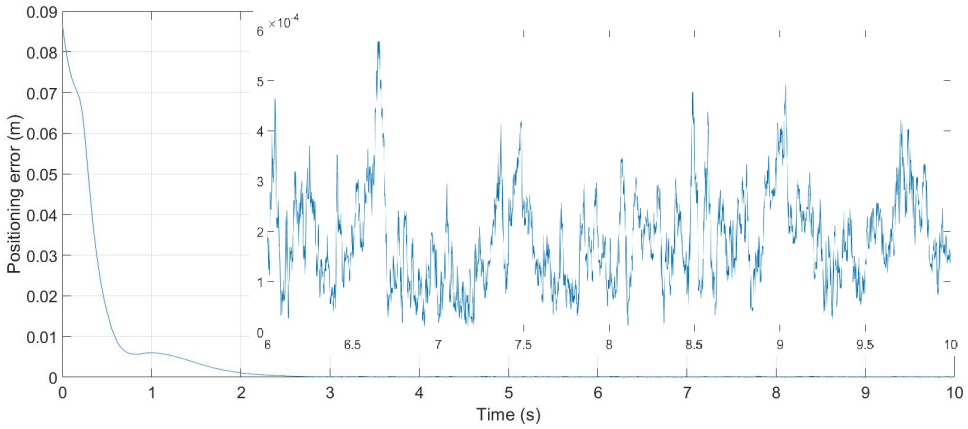

Figure 5.26: Positioning error at point $\mathbf{P}_2$, Pose 2 of the image moment optimized design with the error added in model

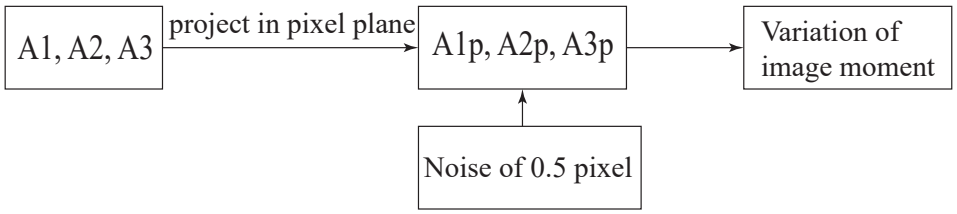

Figure 5.27: Scheme of testing the variation of image moments

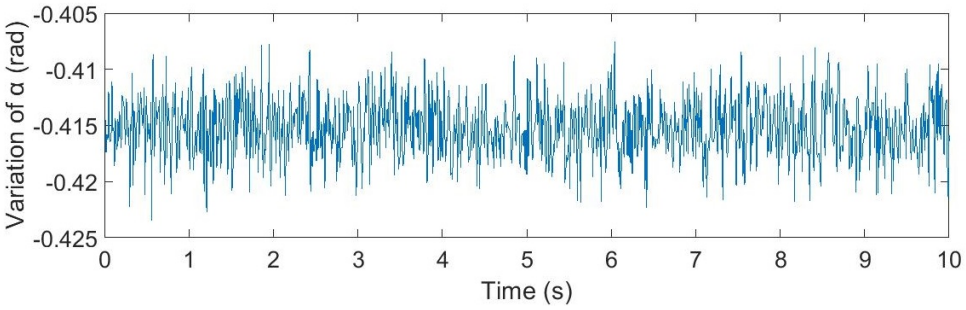

Figure 5.28: Variation of the image moment α for the triangle got from the optimization

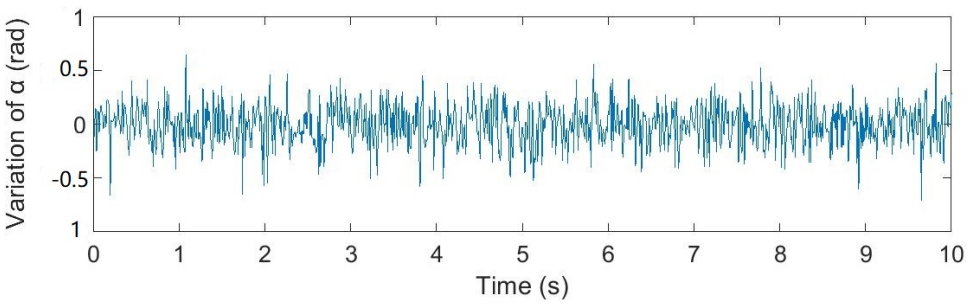

Figure 5.29: Variation of the image moment α for the regular triangle

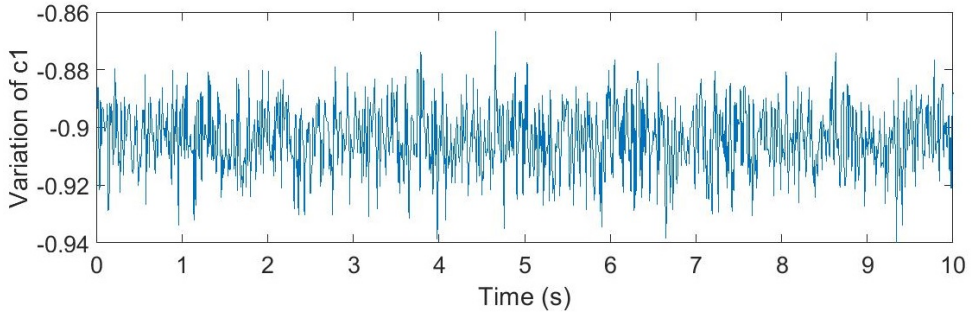

Figure 5.30: Variation of the image moment c_1 for the triangle got from the optimization

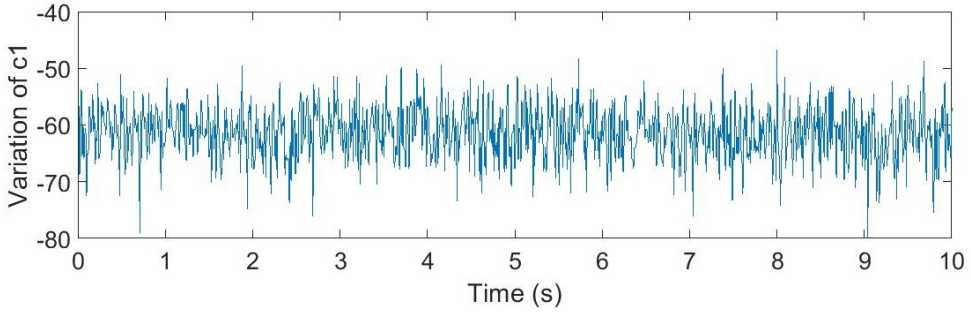

Figure 5.31: Variation of the image moment c_1 for the regular triangle

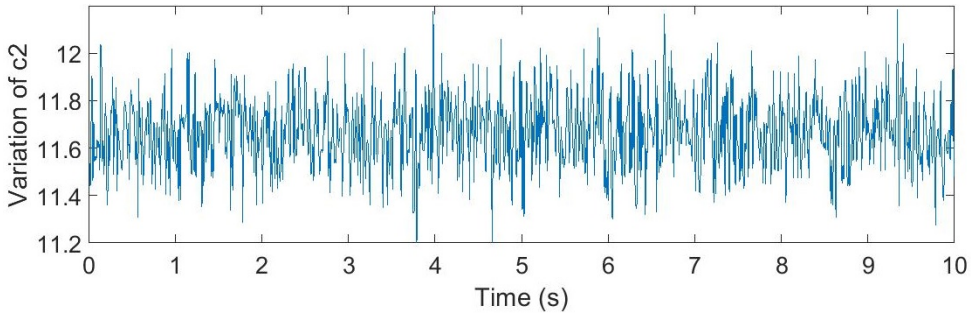

Figure 5.32: Variation of the image moment c_2 for the triangle got from the optimization

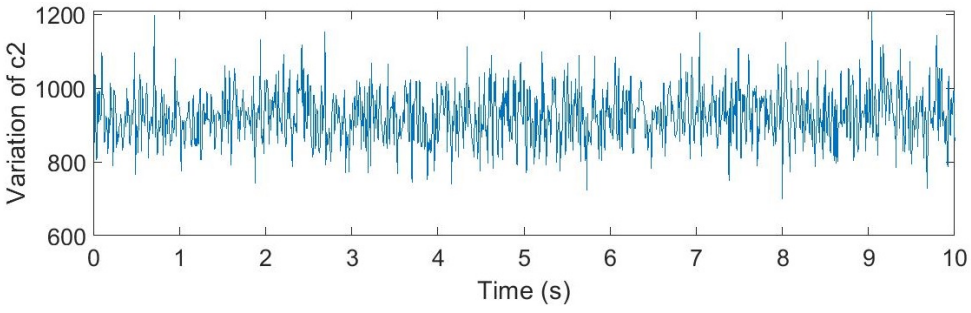

Figure 5.33: Variation of the image moment c_2 for the regular triangle

Table 5.5: Gough-Stewart platform simulation result (positioning error) of leg-direction-based visual servoing (observing legs [1,3,5]) operated with the error-added model)

Point	Pose 1 Max error[mm]	Pose 1 orientation error [rad]	Pose 2 Max error [mm]	Pose 2 orientation error [rad]	Pose 3 Max error [mm]	Pose 3 orientation error [rad]
$\mathbf{P}_1$	1.193	2.7E-4	1.091	3.0E-4	1.137	4.5E-4
$\mathbf{P}_2$	1.031	3.1E-4	1.117	2.1E-4	0.998	3.1E-4
$\mathbf{P}_3$	0.977	3.6E-4	1.325	4.4E-4	1.221	4.6E-4
$\mathbf{P}_4$	1.250	3.0E-4	1.114	4.2E-4	1.202	2.9E-4
$\mathbf{P}_5$	1.195	2.9E-4	1.260	2.6E-4	1.340	2.2E-4
$\mathbf{P}_6$	1.308	4.3E-4	1.317	3.8E-4	1.292	3.1E-4
$\mathbf{P}_7$	1.232	2.9E-4	1.188	3.7E-4	1.168	4.0E-4
$\mathbf{P}_8$	1.251	3.8E-4	1.226	3.9E-4	1.326	4.7E-4
$\mathbf{P}_9$	1.119	3.5E-4	1.181	4.3E-4	1.214	3.8E-4

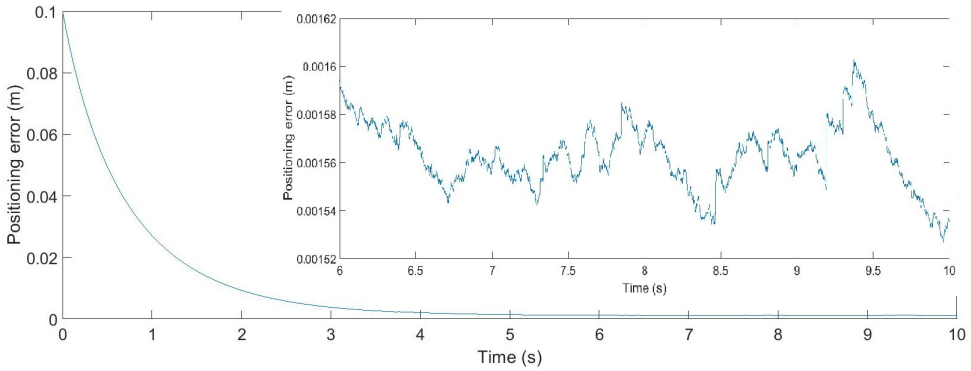

Figure 5.34: Positioning error at point $\mathbf{P}_2$, Pose 2 of the image moment optimized design with the regular triangle $\mathbf{A}_{1r}, \mathbf{A}_{2r}, \mathbf{A}_{3r}$

Based on all the results above, among the three tested controllers, image moment visual servoing may be the best controller with respect to the accuracy performance, but the difference of accuracy between image moment and the other two visual servoing controllers is not significant. Thus only experiments may provide the final conclusion.

5.7 Conclusion

In the work presented above, we performed "control-based design" of a Gough-Stewart platform in order to obtain the best accuracy performance of the robot with its controller. The proposed control-based design methodology took into account the performance of the

Table 5.6: Gough-Stewart platform simulation result (positioning error) of line-based visual servoing (observing legs [1,3,5]) operated with the accurate model)

Point	Pose 1 Max error[mm]	Pose 1 orientation error [rad]	Pose 2 Max error [mm]	Pose 2 orientation error [rad]	Pose 3 Max error [mm]	Pose 3 orientation error [rad]
$\mathbf{P}_1$	1.006	3.1E-4	1.143	3.2E-4	1.034	4.2E-4
$\mathbf{P}_2$	1.122	2.9E-4	1.193	3.7E-4	0.938	3.3E-4
$\mathbf{P}_3$	0.958	2.2E-4	1.217	3.4E-4	1.242	4.5E-4
$\mathbf{P}_4$	1.180	2.5E-4	1.290	4.1E-4	1.295	2.8E-4
$\mathbf{P}_5$	1.095	2.9E-4	1.225	4.0E-4	1.358	2.2E-4
$\mathbf{P}_6$	1.225	3.5E-4	1.146	2.9E-4	1.162	3.8E-4
$\mathbf{P}_7$	1.073	3.7E-4	1.081	3.2E-4	1.204	3.0E-4
$\mathbf{P}_8$	1.159	2.9E-4	1.192	4.5E-4	1.145	4.0E-4
$\mathbf{P}_9$	1.314	4.0E-4	1.137	2.6E-4	1.229	4.1E-4

controller in the design process to get the optimal geometric parameters of robot. We optimized the design of Gough-Stewart platform for three different types of controllers: leg-direction-based visual servoing, line-based visual servoing and image moment visual servoing. Based on these three controllers, we developed positioning error models taking into account the error of observation coming from the camera. We also analyzed the singularities of these controllers to be sure that no controller singularities appeared in the final design of the robot, thus avoiding instability issues. Then, design optimization problems have been formulated in order to find the optimal geometric parameters and camera placement for the Gough-Stewart platform for each type of controller. The simulation results showed that the robots designed for these three visual servoing controllers had the similar size and the robot designed for image moment visual servoing had better accuracy performance than the other two robots optimized for other control techniques. However the differences of robot size and accuracy between the image moments controller and the other two controllers were not significant enough in order to draw general conclusions. Therefore, experimental works on real prototypes may be done in the future in order to verify the simulation results.

Table 5.7: Gough-Stewart platform simulation result of line-based visual servoing (observing legs [1,3,5]) operated with the error added in model)

Point	Pose 1 Max error[mm]	Pose 1 orientation error [rad]	Pose 2 Max error [mm]	Pose 2 orientation error [rad]	Pose 3 Max error [mm]	Pose 3 orientation error [rad]
$\mathbf{P}_1$	1.017	3.7E-4	1.127	2.8E-4	1.230	3.3E-4
$\mathbf{P}_2$	0.961	3.8E-4	1.122	3.5E-4	0.968	3.8E-4
$\mathbf{P}_3$	0.993	4.5E-4	1.271	2.9E-4	1.122	2.8E-4
$\mathbf{P}_4$	1.126	2.9E-4	1.248	2.7E-4	1.213	3.7E-4
$\mathbf{P}_5$	1.282	2.0E-4	1.225	3.4E-4	1.119	4.4E-4
$\mathbf{P}_6$	1.005	3.6E-4	1.190	4.0E-4	1.132	4.0E-4
$\mathbf{P}_7$	0.998	4.5E-4	1.313	4.1E-4	1.247	2.9E-4
$\mathbf{P}_8$	1.150	3.8E-4	1.224	2.9E-4	1.052	3.1E-4
$\mathbf{P}_9$	1.331	3.3E-4	1.028	3.0E-4	1.152	4.0E-4

Table 5.8: Gough-Stewart platform simulation result of leg-direction-based visual servoing (observing legs [1,2,3,4,5,6]) operated with the accurate model)

Point	Pose 1 Max error[mm]	Pose 1 orientation error [rad]	Pose 2 Max error [mm]	Pose 2 orientation error [rad]	Pose 3 Max error [mm]	Pose 3 orientation error [rad]
$\mathbf{P}_1$	1.108	3.4E-4	1.193	3.0E-4	1.303	2.5E-4
$\mathbf{P}_2$	1.232	2.8E-4	1.114	4.0E-4	0.976	3.3E-4
$\mathbf{P}_3$	1.151	2.5E-4	1.422	2.8E-4	1.208	4.7E-4
$\mathbf{P}_4$	1.122	3.1E-4	0.987	2.2E-4	1.117	2.3E-4
$\mathbf{P}_5$	1.208	4.4E-4	1.128	4.1E-4	1.204	4.1E-4
$\mathbf{P}_6$	1.193	4.5E-4	1.219	4.7E-4	1.115	4.0E-4
$\mathbf{P}_7$	1.280	3.6E-4	1.221	2.9E-4	1.017	3.0E-4
$\mathbf{P}_8$	1.432	3.8E-4	1.017	3.5E-4	1.222	3.3E-4
$\mathbf{P}_9$	1.216	2.9E-4	1.120	3.3E-4	1.312	2.9E-4

Table 5.9: Gough-Stewart platform simulation result of leg-direction-based visual servoing (observing legs [1,2,3,4,5,6]) operated with the error added in model)

Point	Pose 1 Max error[mm]	Pose 1 orientation error [rad]	Pose 2 Max error [mm]	Pose 2 orientation error [rad]	Pose 3 Max error [mm]	Pose 3 orientation error [rad]
$\mathbf{P}_1$	0.961	3.5E-4	1.304	2.9E-4	1.120	4.6E-4
$\mathbf{P}_2$	0.992	2.9E-4	1.231	2.5E-4	1.205	4.1E-4
$\mathbf{P}_3$	1.145	2.4E-4	1.137	3.8E-4	1.235	2.5E-4
$\mathbf{P}_4$	1.208	4.5E-4	1.314	3.5E-4	1.272	2.9E-4
$\mathbf{P}_5$	1.321	3.9E-4	1.290	4.1E-4	1.134	3.8E-4
$\mathbf{P}_6$	1.199	3.3E-4	1.120	2.9E-4	1.345	4.1E-4
$\mathbf{P}_7$	1.228	3.0E-4	1.315	2.2E-4	1.219	2.8E-4
$\mathbf{P}_8$	1.174	4.2E-4	1.218	3.7E-4	1.133	3.3E-4
$\mathbf{P}_9$	1.322	3.5E-4	1.307	3.6E-4	1.333	3.6E-4

Table 5.10: Gough-Stewart platform simulation result of line-based visual servoing (observing legs [1,2,3,4,5,6]) operated with the accurate model)

Point	Pose 1 Max error[mm]	Pose 1 orientation error [rad]	Pose 2 Max error [mm]	Pose 2 orientation error [rad]	Pose 3 Max error [mm]	Pose 3 orientation error [rad]
$\mathbf{P}_1$	0.992	2.6E-4	1.093	2.5E-4	1.125	3.7E-4
$\mathbf{P}_2$	0.957	3.1E-4	0.912	3.1E-4	0.990	2.8E-4
$\mathbf{P}_3$	0.971	2.7E-4	1.135	3.3E-4	1.022	2.3E-4
$\mathbf{P}_4$	1.051	4.0E-4	1.011	2.2E-4	0.998	2.5E-4
$\mathbf{P}_5$	1.109	3.7E-4	1.021	2.5E-4	1.034	3.8E-4
$\mathbf{P}_6$	1.040	2.7E-4	1.144	3.4E-4	1.091	3.3E-4
$\mathbf{P}_7$	1.123	2.0E-4	1.218	3.0E-4	1.116	2.8E-4
$\mathbf{P}_8$	1.115	3.1E-4	1.132	2.9E-4	1.142	2.5E-4
$\mathbf{P}_9$	1.191	1.9E-4	1.035	3.6E-4	1.121	2.2E-4

Table 5.11: Gough-Stewart platform simulation result of line-based visual servoing (observing legs [1,2,3,4,5,6]) operated with the error added in model)

Point	Pose 1 Max error[mm]	Pose 1 orientation error [rad]	Pose 2 Max error [mm]	Pose 2 orientation error [rad]	Pose 3 Max error [mm]	Pose 3 orientation error [rad]
$\mathbf{P}_1$	1.021	3.1E-4	1.092	2.9E-4	1.313	3.4E-4
$\mathbf{P}_2$	1.013	2.2E-4	1.012	2.4E-4	0.999	3.5E-4
$\mathbf{P}_3$	0.998	2.1E-4	1.053	3.0E-4	1.222	2.9E-4
$\mathbf{P}_4$	1.125	3.5E-4	1.311	2.5E-4	1.220	3.7E-4
$\mathbf{P}_5$	1.087	3.7E-4	1.136	2.9E-4	1.139	4.0E-4
$\mathbf{P}_6$	1.194	4.0E-4	1.037	3.1E-4	1.225	2.6E-4
$\mathbf{P}_7$	1.123	2.9E-4	1.218	3.0E-4	1.163	3.5E-4
$\mathbf{P}_8$	1.225	3.4E-4	1.320	2.7E-4	1.342	2.8E-4
$\mathbf{P}_9$	1.180	3.8E-4	1.128	2.8E-4	1.226	2.9E-4

Table 5.12: Gough-Stewart platform simulation result of image moment visual servoing operated with the accurate model)

Point	Pose 1 Max error[mm]	Pose 1 orientation error [rad]	Pose 2 Max error [mm]	Pose 2 orientation error [rad]	Pose 3 Max error [mm]	Pose 3 orientation error [rad]
$\mathbf{P}_1$	0.278	2.9E-4	0.579	2.9E-4	0.574	3.1E-4
$\mathbf{P}_2$	0.299	4.3E-4	0.576	2.8E-4	0.601	2.9E-4
$\mathbf{P}_3$	0.319	4.1E-4	0.558	2.9E-4	0.582	2.5E-4
$\mathbf{P}_4$	0.320	3.3E-4	0.574	3.5E-4	0.593	2.2E-4
$\mathbf{P}_5$	0.314	2.9E-4	0.612	3.3E-4	0.598	3.7E-4
$\mathbf{P}_6$	0.345	2.5E-4	0.557	2.7E-4	0.618	3.8E-4
$\mathbf{P}_7$	0.328	3.0E-4	0.595	4.1E-4	0.627	2.7E-4
$\mathbf{P}_8$	0.303	3.1E-4	0.607	3.0E-4	0.580	2.2E-4
$\mathbf{P}_9$	0.341	3.3E-4	0.599	2.9E-4	0.621	3.9E-4

Table 5.13: Gough-Stewart platform simulation result of image moment visual servoing operated with the error added in model)

Point	Pose 1 Max error[mm]	Pose 1 orientation error [rad]	Pose 2 Max error [mm]	Pose 2 orientation error [rad]	Pose 3 Max error [mm]	Pose 3 orientation error [rad]
$\mathbf{P}_1$	0.312	2.9E-4	0.629	3.7E-4	0.613	2.9E-4
$\mathbf{P}_2$	0.331	3.5E-4	0.591	3.4E-4	0.595	4.6E-4
$\mathbf{P}_3$	0.297	4.0E-4	0.662	2.9E-4	0.622	4.5E-4
$\mathbf{P}_4$	0.325	4.4E-4	0.651	4.4E-4	0.648	2.8E-4
$\mathbf{P}_5$	0.348	4.1E-4	0.628	4.0E-4	0.633	3.3E-4
$\mathbf{P}_6$	0.330	3.8E-4	0.664	4.5E-4	0.641	3.5E-4
$\mathbf{P}_7$	0.373	3.1E-4	0.598	4.3E-4	0.623	2.8E-4
$\mathbf{P}_8$	0.328	2.6E-4	0.614	3.8E-4	0.619	2.9E-4
$\mathbf{P}_9$	0.317	2.9E-4	0.663	3.9E-4	0.607	3.1E-4

Table 5.14: Gough-Stewart platform simulation result of image moment visual servoing operated with the regular triangle $\mathbf{A}_{1r}, \mathbf{A}_{2r}, \mathbf{A}_{3r}$)

Point	Pose 1 Max error[mm]	Pose 1 orientation error [rad]	Pose 2 Max error [mm]	Pose 2 orientation error [rad]	Pose 3 Max error [mm]	Pose 3 orientation error [rad]
$\mathbf{P}_1$	1.141	4.1E-4	1.611	5.4E-4	1.437	4.7E-4
$\mathbf{P}_2$	1.293	4.5E-4	1.603	4.8E-4	1.564	3.8E-4
$\mathbf{P}_3$	1.182	3.7E-4	1.628	4.9E-4	1.593	4.2E-4
$\mathbf{P}_4$	1.037	4.9E-4	1.666	5.2E-4	1.431	5.5E-4
$\mathbf{P}_5$	1.184	5.8E-4	1.578	5.6E-4	1.629	5.3E-4
$\mathbf{P}_6$	1.020	5.4E-4	1.592	5.7E-4	1.551	3.9E-4
$\mathbf{P}_7$	1.192	5.0E-4	1.516	5.1E-4	1.478	5.6E-4
$\mathbf{P}_8$	1.052	4.8E-4	1.575	4.8E-4	1.527	4.8E-4
$\mathbf{P}_9$	1.031	4.3E-4	1.542	5.3E-4	1.553	4.8E-4

Chapter 6

Conclusions and perspectives for future works

6.1 Summary and contribution of the thesis

This subject of this thesis was the control-based design of robot.

The first Chapter of our manuscript was devoted to detail the concept of parallel robots, including its definition, geometric model, inverse and direct kinematic model, singularities configurations and control scheme. However, the classical model-based control schemes are limited because of the complex structure of parallel robots. The manufacturing and assembly errors affect the final control accuracy. Then the external sensor-based control was proposed to bypass the model of the robot entirely. Visual servoing improves the control performance of the parallel robot compared with the classical control method. Different kinds of image features have been applied in the visual servoing: leg-directions, lines and image moments. However, the use of visual servoing leads to controller singularity and a tool hidden robot can help with the analysis of the controller singularities. In order to get the best performance of the robot with its controller (in terms of singularity avoidance and accuracy), we then proposed to use a control-based design methodology. Control-based design methodology is an objective-function-based optimal design method which takes the controller performance into account in the design process to get the optimal robot geometric parameters.

The present work applied the control-based design methodology on the design of different kinds of parallel robots: Five-bar mechanisms, DELTA robots and Gough-Stewart platforms.

Five-bar mechanism is a planar parallel robot with 2 DOFs. We performed the control-based design of Five-bar mechanisms to get the best accuracy performance of the robot with its controller. Two different types of controllers were tested in the design of Five-bar mechanism: leg-direction-based visual servoing and line-based visual servoing. We devel-

oped positioning error models taking into account the error of observation coming from the camera and analyzed the controller singularities with the help of the hidden robot. Then, design optimizations problems were formulated and co-simulations between ADAMS and Simulink were performed to test the accuracy performance of the controllers. The robot size designed for leg-direction-based visual servoing and line-based visual servoing were similar. The line-based visual servoing controller which controls the leg edges shows a better control performance with respect to the accuracy estimated on the co-simulation.

Then the control-based design methodology was applied to the optimal design of DELTA robots in order to obtain the best accuracy performance of the robots with their controllers. We optimized the design of DELTA robot for three different types of controllers: leg-direction-based visual servoing, line-based visual servoing and image moment visual servoing. The corresponding positioning error models taking into account the error of observation coming from the camera based on these controllers were developed. In order to avoid the instability issues, we also analyzed the singularities of these controllers to be sure that no singularity of the controller appeared in the final design of the robot. Then, design optimization problems have been formulated in order to find the optimal geometric parameters and camera placement for the DELTA robot for each type of controller. The simulation results showed that the robot designed for image moment visual servoing was more compact and had better accuracy performance than the other two robots optimized for other control techniques. However the differences of robot size and accuracy between the image moments controller and line-based controller were not significant enough in order to draw general conclusions. Therefore, the experimental works on real prototypes were needed in order to verify the simulation results.

To verify the results from the optimization and simulation, we performed the experiments on the prototypes. The two DELTA prototypes (Prototype 1 for line-based visual servoing, Prototype 2 for image moment visual servoing) as well as their hardware and controller implementations were designed by myself under the supervision of the LS2N staff technical. Then, the camera identification was done: the calibration of the intrinsic and extrinsic parameters of the camera and the measurement of noise. The calibration results showed that the real noise is about 0.5 pixel to 0.6 pixel, and the hypothesis of a noise of 1 pixel in the co-simulation performed for DELTA robot was overestimated. In the next step, we carried out the tests on the visual servoing controllers, the repeatability of the robot was measured. The experiments results showed that line-based visual servoing can achieve a precision around 0.5mm, image moment visual servoing can achieve a

precision around 0.38mm. At the same time, we reperformed the co-simulations for the DELTA robot with the noise calibrated to be the comparison of the experiments results. The co-simulation results confirmed the results regarding the prototype's performance. As a conclusion, image moment visual servoing is the best controller for the control of DELTA robot compared with line-based visual servoing.

In the next step, the control-based design of the Gough-Stewart platform was performed in order to obtain the best accuracy performance of the robot with its controller. We optimized the design of Gough-Stewart platform for three different types of controllers: leg-direction-based visual servoing, line-based visual servoing and image moment visual servoing. Based on these three controllers, we developed positioning error models taking into account the error of observation coming from the camera and analyzed the singularities of these controllers to be sure that no controller singularities appeared in the final design of the robot, thus avoiding instability issues. Then, design optimization problems have been formulated in order to find the optimal geometric parameters and camera placement for the Gough-Stewart platform for each type of controller. The simulation results showed that the robots designed for these three visual servoing controllers had the similar size and the robot designed for image moment visual servoing had better accuracy performance than the other two robots optimized for other control techniques. Nevertheless, the differences of robot size and accuracy between the image moments controller and the other two controllers were not significant enough in order to draw general conclusions. Therefore, experimental works on real prototypes may be done in the future in order to verify the simulation results.

Thus, all the research results presented above showed that the control-based design methodology was a powerful tool for the optimal design of the parallel robot. It helped getting the optimal geometric parameters of parallel robots which considers the controller performance in the design process. The positioning error models we created could be used to evaluate the controller accuracy and the positioning errors obtained from the models were confirmed by the experiment results. The study of hidden robot simplified the analysis of controller singularity and helped avoiding the instability issues. Both positioning error models and hidden robot were all the important indices in the control-based design process.

Compared to the previous work, this was the first time that control-based design approaches were applied to the optimal design of a three DOFs spatial parallel robot, DELTA robot, and six DOFs spatial parallel robot, Gough-Stewart platform. Further-

more, this was the first time that image moment visual servoing was taken into account as one of the controllers in control-based design.

6.2 Direction for future works

At first, experimental validation of different designs for the Five-bar mechanism and the Gough-Stewart platform could be envisioned. The experiments performed on the real prototypes could allow the comparison of many different Five-bar mechanism designs and Gough-Stewart platform designs, using different visual servoings,

In the optimal design of Gough-Stewart platform, the analysis of the controller singularity when using image moment visual servoing was based on the study of the condition number of the interaction matrix since its algebraic expression is complex. In the future, we can try to find other image moment features that each is related to only one DOF (interaction matrix is a diagonal matrix) or try to apply the hidden robot on the image moment visual servoing of parallel robots to simplify the analysis of the controller singularity.

In this work, all the study of the hidden robot concept of parallel robots are all based on the cylindrical legs. Then, another possible research is the study of the hidden robot of the parallel robots whose legs are not cylindrical. It is possible to find the way of estimating the pose of an object with an arbitrary shape through the classical vision-based techniques. Therefore, the main challenge is to assimilate these arbitrary shapes into an equivalent cylindrical leg robot and apply the hidden robot concept. Another solution is to develop a new strategy for finding the corresponding hidden robots for these non-cylindrical legs.

In the optimal design of DELTA robot and Gough-Stewart platform, when image moment visual servoing is considered, the objects we observed are regular graphics and discrete models. However, the continuous irregular graphics have never been considered in the control-based design of parallel robots. The challenge is the creation of the positioning error model when observing the irregular graphic object and the topological optimization of the its shape. Thus, considering the application of continuous irregular graphics in the control-based design of robots is a direction of study in the next step.

In the experiments of the DELTA robot, the camera observation noise has a big influence in the final control accuracy. Thus a more robust approach, a hybrid controller, which takes into account both the image moment data and the Plücker coordinates of

the robot link can be used. For this new hybrid controller, the weights representing each image features' contribution to the final control law can be a variable in the optimal design problem in order to get a good control performance.

In addition, throughout this work, the control law applied on the control of the motion of the robots was a simple point-to-point motion. An online trajectory generator could be used in order to achieve the best possible performance in accordance with a given index.

Appendix A

List of publications of presented works

Publications in international congress

- Minglei Zhu, A. Chriette and S. Briot. *Control-based design of a DELTA robot.* Proceedings of the 23rd CISM IFToMM Symposium on Robot Design, Dynamics and Control (RoManSy 2020), Sept. 20-24, 2020, Sapporo, Japan.

Poster presentations

- Minglei Zhu, A. Chriette and S. Briot. *Optimal design of a DELTA robot.* Journé des Jeunes Chercheurs en Robotique Edition 2019 (JJCR 2019), Vittle, France.

Other publications

- Minglei Zhu. *Control-based design of a DELTA robot.* Journé des doctorants Edition 2019 (JDOC 2019), Doctoral school MathSTIC, Nantes, France.

References

[Allibert 2010] Guillaume Allibert, Estelle Courtial, and Francçois Chaumette. Visual servoing via nonlinear predictive control. In *Visual Servoing via Advanced Numerical Methods*, pages 375–393. Springer, 2010.

[Allibert 2010] Guillaume Allibert, Estelle Courtial, and François Chaumette. Predictive control for constrained image-based visual servoing. *IEEE Transactions on Robotics*, 26(5):933–939, 2010.

[Andreff 2002] Nicolas Andreff, Bernard Espiau, and Radu Horaud. Visual servoing from lines. *The International Journal of Robotics Research*, 21(8):679–699, 2002.

[Andreff 2005] Nicolas Andreff, Arnaud Marchadier, and Philippe Martinet. Vision-based control of a gough-stewart parallel mechanism using legs observation. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2535–2540. IEEE, 2005.

[Andreff 2007] Nicolas Andreff, Tej Dallej, and Philippe Martinet. Image-based visual servoing of a gough—stewart parallel manipulator using leg observations. *The International Journal of Robotics Research*, 26(7):677–687, 2007.

[Andreff 2016] Nicolas Andreff and Brahim Tamadazte. Laser steering using virtual trifocal visual servoing. *The International Journal of Robotics Research*, 35(6):672–694, 2016.

[Arakelian 2007] Vigen Arakelian, Sébastien Briot, and Victor Glazunov. Improvement of functional performance of spatial parallel manipulators using mechanisms of variable structure. 2007.

[Arakelian 2008] Vigen Arakelian, Sébastien Briot, and Victor Glazunov. Increase of singularity-free zones in the workspace of parallel manipulators using mechanisms of variable structure. *Mechanism and Machine Theory*, 43(9):1129–1140, 2008.

[Baker 1998] Simon Baker and Shree K Nayar. A theory of catadioptric image formation. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, pages 35–42. IEEE, 1998.

[Bakthavatchalam 2013] Manikandan Bakthavatchalam, François Chaumette, and Eric Marchand. Photometric moments: New promising candidates for visual servoing. In *2013 IEEE International Conference on Robotics and Automation*, pages 5241–5246. IEEE, 2013.

[Bakthavatchalam 2014] Manikandan Bakthavatchalam, Omar Tahri, and François Chaumette. Improving moments-based visual servoing with tunable visual features. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6186–6191. IEEE, 2014.

[Bakthavatchalam 2018] Manikandan Bakthavatchalam, Omar Tahri, and François Chaumette. A direct dense visual servoing approach using photometric moments. *IEEE Transactions on Robotics*, 34(5):1226–1239, 2018.

[Ben 2006] Patricia Ben-Horin and Moshe Shoham. Singularity analysis of a class of parallel robots based on grassmann–cayley algebra. *Mechanism and Machine Theory*, 41(8):958–970, 2006.

[Ben 2009] Patricia Ben-Horin and Moshe Shoham. Application of grassmann—cayley algebra to geometrical interpretation of parallel robot singularities. *The International Journal of Robotics Research*, 28(1):127–141, 2009.

[Blaise 2010] Julien Blaise, Ilian Bonev, Bruno Monsarrat, Sébastien Briot, Jason Michel Lambert, and Claude Perron. Kinematic characterisation of hexapods for industry. *Industrial Robot: An International Journal*, 2010.

[Bolzon 2017] Victor R Bolzon and Fabian Andres Lara-Molina. Optimal design of flexible parallel robot based on kinematic and dynamic criteria. In *2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)*, pages 1–6. IEEE, 2017.

[Bonev 1999] Ilian A Bonev and Jeha Ryu. Orientation workspace analysis of 6-dof parallel manipulators. In *Proceedings of the ASME*, pages 1–8, 1999.

[Bounab 2016] Belkacem Bounab. Multi-objective optimal design based kineto-elastostatic performance for the delta parallel mechanism. *Robotica*, 34(2):258–273, 2016.

[Bourquardez 2009] Odile Bourquardez, Robert Mahony, Nicolas Guenard, François Chaumette, Tarek Hamel, and Laurent Eck. Image-based visual servo control of the

translation kinematics of a quadrotor aerial vehicle. *IEEE Transactions on Robotics*, 25(3):743–749, 2009.

[Briot 2008] Sébastien Briot and Vigen Arakelian. Optimal force generation in parallel manipulators for passing through the singular positions. *The International Journal of Robotics Research*, 27(8):967–983, 2008.

[Briot 2010] Sébastien Briot, Anatol Pashkevich, and Damien Chablat. Optimal technology-oriented design of parallel robots for high-speed machining applications. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1155–1161. IEEE, 2010.

[Briot 2013] Sébastien Briot and Philippe Martinet. Minimal representation for the control of gough-stewart platforms via leg observation considering a hidden robot model. In *2013 IEEE International Conference on Robotics and Automation*, pages 4653–4658. IEEE, 2013.

[Briot 2014] Sébastien Briot and Wisama Khalil. Recursive and symbolic calculation of the elastodynamic model of flexible parallel robots. *The International Journal of Robotics Research*, 33(3):469–483, 2014.

[Briot 2014] Sébastien Briot, Victor Rosenzveig, and Philippe Martinet. The hidden robot concept: a tool for control analysis and robot control-based design. In *Advances in Robot Kinematics*, pages 31–39. Springer, 2014.

[Briot 2016] Sébastien Briot, François Chaumette, and Philippe Martinet. Revisiting the determination of the singularity cases in the visual servoing of image points through the concept of hidden robot. *IEEE Transactions on Robotics*, 33(3):536–546, 2016.

[Briot 2016] Sébastien Briot, Philippe Martinet, and François Chaumette. Determining the singularities for the observation of three image lines. *IEEE Robotics and Automation Letters*, 2(2):412–419, 2016.

[Briot 2016] Sébastien Briot, Victor Rosenzveig, Philippe Martinet, Erol Özgür, and Nicolas Bouton. Minimal representation for the control of parallel robots via leg observation considering a hidden robot model. *Mechanism and Machine Theory*, 106:115–147, 2016.

[Briot 2017] Sébastien Briot, Stéphane Caro, and Coralie Germain. Design procedure for a fast and accurate parallel manipulator. *Journal of Mechanisms and Robotics*, 9(6), 2017.

[Caro 2010] Stéphane Caro, Guillaume Moroz, Thibault Gayral, Damien Chablat, and Chao Chen. Singularity analysis of a six-dof parallel manipulator using grassmann-cayley algebra and groebner bases. In *Brain, Body and Machine*, pages 341–352. Springer, 2010.

[Cazy 2015] Nicolas Cazy, Pierre-Brice Wieber, Paolo Robuffo Giordano, and François Chaumette. Visual servoing when visual information is missing: Experimental comparison of visual feature prediction schemes. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6031–6036. IEEE, 2015.

[Censor 1977] Yair Censor. Pareto optimality in multiobjective problems. *Applied Mathematics and Optimization*, 4(1):41–59, 1977.

[Chablat 2003] Damien Chablat and Philippe Wenger. Architecture optimization of a 3-dof translational parallel mechanism for machining applications, the orthoglide. *IEEE Transactions on Robotics and Automation*, 19(3):403–410, 2003.

[Chaumette 1998] Francois Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In *The confluence of vision and control*, pages 66–78. Springer, 1998.

[Chaumette 2004] François Chaumette. Image moments: a general and useful set of features for visual servoing. *IEEE Transactions on Robotics*, 20(4):713–723, 2004.

[Chaumette 2006] François Chaumette and Seth Hutchinson. Visual servo control. i. basic approaches. *IEEE Robotics & Automation Magazine*, 13(4):82–90, 2006.

[Chaumette 2016] François Chaumette, Seth Hutchinson, and Peter Corke. Visual servoing. In *Springer Handbook of Robotics*, pages 841–866. Springer, 2016.

[Chen 2016] Jian Chen, Bingxi Jia, and Kaixiang Zhang. Trifocal tensor-based adaptive visual trajectory tracking control of mobile robots. *IEEE Transactions on Cybernetics*, 47(11):3784–3798, 2016.

[Clavel 1990] Reymond Clavel. Device for the movement and positioning of an element in space, December 11 1990. US Patent 4,976,582.

[Coello 2002] CA Coello Coello and M Salazar Lechuga. Mopso: A proposal for multiple objective particle swarm optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, volume 2, pages 1051–1056. IEEE, 2002.

[Corke 2001] Peter I Corke and Seth A Hutchinson. A new partitioned approach to image-based visual servo control. *IEEE Transactions on Robotics and Automation*, 17(4):507–515, 2001.

[Corke 2010] Peter I Corke. Spherical image-based visual servo and structure estimation. In *2010 IEEE international conference on Robotics and Automation*, pages 5550–5555. IEEE, 2010.

[Courbon 2007] Jonathan Courbon, Youcef Mezouar, Laurent Eckt, and Philippe Martinet. A generic fisheye camera model for robotic applications. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1683–1688. IEEE, 2007.

[Dallej 2006] Tej Dallej, Nicolas Andreff, and Philippe Martinet. Visual servoing of par4 using leg observation. In *IECON 2006-32nd Annual Conference on IEEE Industrial Electronics*, pages 3782–3787. IEEE, 2006.

[Dallej 2007] Tej Dallej, Nicolas Andreff, and Philippe Martinet. Image-based visual servoing of the i4r parallel robot without proprioceptive sensors. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 1709–1714. IEEE, 2007.

[Dallej 2019] Tej Dallej, Marc Gouttefarde, Nicolas Andreff, Pierre-Elie Hervé, and Philippe Martinet. Modeling and vision-based control of large-dimension cable-driven parallel robots using a multiple-camera setup. *Mechatronics*, 61:20–36, 2019.

[Dasgupta 2000] Bhaskar Dasgupta and TS Mruthyunjaya. The stewart platform manipulator: a review. *Mechanism and machine theory*, 35(1):15–40, 2000.

[Deb 2000] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and Tanaka Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *International conference on parallel problem solving from nature*, pages 849–858. Springer, 2000.

[Deb 2002] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[Downing 2002] DM Downing, Andrew E Samuel, and Kenneth H Hunt. Identification of the special configurations of the octahedral manipulator using the pure condition. *The International Journal of Robotics Research*, 21(2):147–159, 2002.

[Espiau 1992] Bernard Espiau, François Chaumette, and Patrick Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326, 1992.

[Feddema 1989] John T Feddema and Owen Robert Mitchell. Vision-guided servoing with feature-based trajectory generation (for robots). *IEEE Transactions on Robotics and Automation*, 5(5):691–700, 1989.

[Fichter 1986] Eugene F Fichter. A stewart platform-based manipulator: general theory and practical construction. *The International Journal of Robotics Research*, 5(2):157–182, 1986.

[Flandin 2000] Grégory Flandin, François Chaumette, and Eric Marchand. Eye-in-hand/eye-to-hand cooperation for visual servoing. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 3, pages 2741–2746. IEEE, 2000.

[French 1985] Michael J French, JT Gravdahl, and MJ French. *Conceptual design for engineers*. Springer, 1985.

[Fusco 2019] Franco Fusco, Olivier Kermorgant, and Philippe Martinet. A comparison of visual servoing from features velocity and acceleration interaction models. 2019.

[Fusco 2020] Franco Fusco, Olivier Kermorgant, and Philippe Martinet. Integrating features acceleration in visual predictive control. *IEEE Robotics and Automation Letters*, 5(4):5197–5204, 2020.

[Germain 2011] Coralie Germain, Se´ bastien Briot, Victor Glazunov, Ste´ phane Caro, and Philippe Wenger. Irsbot-2: A novel two-dof parallel robot for high-speed operations. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 54839, pages 899–909, 2011.

[Germain 2013] Coralie Germain, Stéphane Caro, Sébastien Briot, and Philippe Wenger. Singularity-free design of the translational parallel manipulator irsbot-2. *Mechanism and Machine Theory*, 64:262–285, 2013.

[Germain 2013] Coralie Germain, Stéphane Caro, Sébastien Briot, and Philippe Wenger. Optimal design of the irsbot-2 based on an optimized test trajectory. In *ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2013*, 2013.

[Gogu 2004] Grigore Gogu. Structural synthesis of fully-isotropic translational parallel robots via theory of linear transformations. *European Journal of Mechanics-A/Solids*, 23(6):1021–1039, 2004.

[Gosselin 1988] Clément Gosselin. *Kinematic analysis, optimization and programming of parallel robotic manipulators*. McGill University Montréal, Canada, 1988.

[Goudali 1995] Ahmed Goudali. *Contribution à l'étude d'un nouveau robot Parallèle 2-Delta à six degrés de liberté avec découplage*. PhD thesis, Poitiers, 1995.

[Hao 2005] Fang Hao and J-P Merlet. Multi-criteria optimal design of parallel manipulators based on interval analysis. *Mechanism and machine theory*, 40(2):157–171, 2005.

[Hill 2017] Rafael Balderas Hill, Damien Six, Abdelhamid Chriette, Sébastien Briot, and Philippe Martinet. Crossing type 2 singularities of parallel robots without pre-planned trajectory with a virtual-constraint-based controller. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6080–6085. IEEE, 2017.

[Huang 2013] Tian Huang, Songtao Liu, Jiangping Mei, and Derek G Chetwynd. Optimal design of a 2-dof pick-and-place parallel robot using dynamic performance indices and angular constraints. *Mechanism and Machine Theory*, 70:246–253, 2013.

[Hunt 1978] Kenneth Henderson Hunt. *Kinematic geometry of mechanisms*, volume 7. Oxford University Press, USA, 1978.

[Huntchinson 1996] Seth Hutchinson, Gregory D Hager, and Peter I Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, 1996.

[Jabbari 2014] Hamed Jabbari, Giuseppe Oriolo, and Hossein Bolandi. An adaptive scheme for image-based visual servoing of an underactuated uav. *International Journal of Robotics and Automation*, 29(1):92–104, 2014.

[Kaci 2017] Lila Kaci, Sébastien Briot, Clément Boudaud, and Philippe Martinet. Control-based design of a five-bar mechanism. In *New Trends in Mechanism and Machine Science*, pages 303–311. Springer, 2017.

[Kaci 2018] Lila Kaci, Clement Boudaud, Sebastien Briot, and Philippe Martinet. Elastostatic modelling of a wooden parallel robot. In *Computational Kinematics*, pages 53–61. Springer, 2018.

[Kaci 2018] Lila Kaci, Sébastien Briot, Clément Boudaud, and Philippe Martinet. Robecolo: Optimal design of a wooden five-bar mechanism. In *ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers Digital Collection, 2018.

[Kahan 1996] William Kahan. Ieee standard 754 for binary floating-point arithmetic. *Lecture Notes on the Status of IEEE*, 754(94720-1776):11, 1996.

[Kannala 2004] Juho Kannala and Sami Brandt. A generic camera calibration method for fish-eye lenses. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 1, pages 10–13. IEEE, 2004.

[Kannala 2006] Juho Kannala and Sami S Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE transactions on pattern analysis and machine intelligence*, 28(8):1335–1340, 2006.

[Kelaiaia 2012] Ridha Kelaiaia, Olivier Company, and Abdelouahab Zaatri. Multiobjective optimization of a linear delta parallel robot. *Mechanism and Machine Theory*, 50:159–178, 2012.

[Kermorgant 2011] Olivier Kermorgant and François Chaumette. Combining ibvs and pbvs to ensure the visibility constraint. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2849–2854. IEEE, 2011.

[Keshmiri 2014] Mohammad Keshmiri, Wen-Fang Xie, and Abolfazl Mohebbi. Augmented image-based visual servoing of a manipulator using acceleration command. *IEEE Transactions on Industrial Electronics*, 61(10):5444–5452, 2014.

[Khomutenko 2015] Bogdan Khomutenko, Gaëtan Garcia, and Philippe Martinet. An enhanced unified camera model. *IEEE Robotics and Automation Letters*, 1(1):137–144, 2015.

[Kraige 2020] L Glenn Kraige, JN Bolton, et al. *Engineering Mechanics: Dynamics*. John Wiley & Sons, 2020.

[Laribi 2007] MA Laribi, L Romdhane, and S Zeghloul. Analysis and dimensional synthesis of the delta robot for a prescribed workspace. *Mechanism and machine theory*, 42(7):859–870, 2007.

[Liu 1993] Kai Liu, John M Fitzgerald, and Frank L Lewis. Kinematic analysis of a stewart platform manipulator. *IEEE Transactions on industrial electronics*, 40(2):282–293, 1993.

[Liu 2007] Xin-Jun Liu and Jinsong Wang. A new methodology for optimal kinematic design of parallel mechanisms. *Mechanism and machine theory*, 42(9):1210–1224, 2007.

[Lopez 2019] Erick García López, Wen Yu, and Xiaoou Li. Optimal design of a parallel robot using neural network and genetic algorithm. In *2019 Tenth International Conference on Intelligent Control and Information Processing (ICICIP)*, pages 64–69. IEEE, 2019.

[Maki 2013] Toshihiro Maki, ReyesTatsuru Shiroku, Yoshiki Sato, Takumi Matsuda, Takashi Sakamaki, and Tamaki Ura. Docking method for hovering type auvs by acoustic and visual positioning. In *2013 IEEE international underwater technology symposium (UT)*, pages 1–6. IEEE, 2013.

[Malis 2000] Ezio Malis and François Chaumette. 2 1/2 d visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. *International Journal of Computer Vision*, 37(1):79–97, 2000.

[Malis 2004] Ezio Malis. Improving vision-based control using efficient second-order minimization techniques. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 2, pages 1843–1848. IEEE, 2004.

[Malis 1999] Ezio Malis, François Chaumette, and Sylvie Boudet. 2d 1/2 visual servoing. In *IEEE TRANS. ON ROBOTICS AND AUTOMATION*. Citeseer, 1999.

[Mamistvalov 1998] Alexander G Mamistvalov. N-dimensional moment invariants and conceptual mathematical theory of recognition n-dimensional solids. *IEEE Transactions on pattern analysis and machine intelligence*, 20(8):819–831, 1998.

[Marchand 2001] Eric Marchand and François Chaumette. A new formulation for non-linear camera calibration using virtual visual servoing. 2001.

[Marchand 2017] Eric Marchand and François Chaumette. Visual servoing through mirror reflection. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3798–3804. IEEE, 2017.

[Martinet 1996] P Martinet, J Gallice, and D Khadraoui. Vision based control law using 3d visual features. In *Proc. WAC*, volume 96, pages 497–502, 1996.

[Masory 1996] Oren Masory, Jian Wang, and Hanqi Zhuang. Kinematic modeling and calibration of a stewart platform. *Advanced Robotics*, 11(5):519–539, 1996.

[Matone 1999] R Matone and B Roth. In-parallel manipulators: a framework on how to model actuation schemes and a study of their effects on singular postures. *Journal of Mechanical Design*, 121(1):2–8, 1999.

[Mebarki 2008] Rafik Mebarki, Alexandre Krupa, and François Chaumette. Image moments-based ultrasound visual servoing. In *2008 IEEE International Conference on Robotics and Automation*, pages 113–119. IEEE, 2008.

[Mebarki 2010] Rafik Mebarki, Alexandre Krupa, and François Chaumette. 2-d ultrasound probe complete guidance by visual servoing using image moments. *IEEE Transactions on Robotics*, 26(2):296–306, 2010.

[Merlet 2006] Jean-Pierre Merlet. Jacobian, manipulability, condition number, and accuracy of parallel robots. 2006.

[Merlet 2006] Jean-Pierre Merlet. *Parallel robots*, volume 128. Springer Science & Business Media, 2006.

[Merlet 2015] M Jean-Pierre MERLET. *Sensor-Based Design and Control of High-Speed Manipulators*. PhD thesis, École Centrale de Nantes, 2015.

[Michel 1993] Henri Michel and Patrick Rives. Singularities in the determination of the situation of a robot effector from the perspective view of 3 points. 1993.

[Miller 2004] Karol Miller. Optimal design and modeling of spatial parallel manipulators. *The International Journal of Robotics Research*, 23(2):127–140, 2004.

[Mukundan 1998] Ramakrishnan Mukundan and KR Ramakrishnan. *Moment functions in image analysis-theory and applications*. World Scientific, 1998.

[Ozawa 2011] Ryuta Ozawa and François Chaumette. Dynamic visual servoing with image moments for a quadrotor using a virtual spring approach. In *2011 IEEE international conference on robotics and automation*, pages 5670–5676. IEEE, 2011.

[Ozawa 2013] Ryuta Ozawa and François Chaumette. Dynamic visual servoing with image moments for an unmanned aerial vehicle using a virtual spring approach. *Advanced Robotics*, 27(9):683–696, 2013.

[Papanikolopoulos 1993] Nikolaos P Papanikolopoulos and Pradeep K Khosla. Adaptive robotic visual tracking: Theory and experiments. *IEEE Transactions on Automatic Control*, 38(3):429–445, 1993.

[Park 1999] FC Park and Jin Wook Kim. Singularity analysis of closed kinematic chains. 1999.

[Pierrot 1990] François Pierrot, C Reynaud, and Alain Fournier. Delta: a simple and efficient parallel robot. *Robotica*, 8(2):105–109, 1990.

[Romdhane 2002] Lotfi Romdhane, Z Affi, and M Fayet. Design and singularity analysis of a 3-translational-dof in-parallel manipulator. *J. Mech. Des.*, 124(3):419–426, 2002.

[Rosenzveig 2013] Victor Rosenzveig, Sébastien Briot, and Philippe Martinet. 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*, 2013.

[Rosenzveig 2014] Victor Rosenzveig, Sébastien Briot, Philippe Martinet, Erol Özgür, and Nicolas Bouton. A method for simplifying the analysis of leg-based visual servoing of parallel robots. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5720–5727. IEEE, 2014.

[Sabatta 2013] Deon Sabatta and Roland Siegwart. Vision-based path following using the 1d trifocal tensor. In *2013 IEEE International Conference on Robotics and Automation*, pages 3095–3102. IEEE, 2013.

[Selig 2004] Jon M Selig. *Geometric fundamentals of robotics*. Springer Science & Business Media, 2004.

[Shao 2016] Quanquan Shao, Jie Hu, Yi Fang, Wenhai Liu, Jin Qi, and Guo-Niu Zhu. Image moments based visual servoing of robot using an adaptive controller. In *2016 Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, pages 57–61. IEEE, 2016.

[St 2000] Boris Mayer St-Onge and Clément M Gosselin. Singularity analysis and representation of the general gough-stewart platform. *The International Journal of Robotics Research*, 19(3):271–288, 2000.

[Taghirad 2013] Hamid D Taghirad. *Parallel robots: mechanics and control*. CRC press, 2013.

[Tahri 2004] O Tahri. Application des momentsa l'asservissement visuel et au calcul de pose. *These de doctorat, Université de Rennes*, 1, 2004.

[Tahri 2004] Omar Tahri and François Chaumette. Image moments: Generic descriptors for decoupled image-based visual servo. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 2, pages 1185–1190. IEEE, 2004.

[Tahri 2005] Omar Tahri and Francois Chaumette. Point-based and region-based image moments for visual servoing of planar objects. *IEEE Transactions on Robotics*, 21(6):1116–1127, 2005.

[Tahri 2010] Omar Tahri, Youcef Mezouar, François Chaumette, and Peter Corke. Decoupled image-based visual servoing for cameras obeying the unified projection model. *IEEE Transactions on Robotics*, 26(4):684–697, 2010.

[Tahri 2013] Omar Tahri, Helder Araujo, François Chaumette, and Youcef Mezouar. Robust image-based visual servoing using invariant visual information. *Robotics and Autonomous Systems*, 61(12):1588–1600, 2013.

[Tamtsia 2013] A Yeremou Tamtsia, Omar Tahri, Youcef Mezouar, Haman Djalo, and Emmanuel Tonye. New results in images moments-based visual servoing. In *2013 IEEE International Conference on Robotics and Automation*, pages 5271–5276. IEEE, 2013.

[Thuilot 2002] Benoit Thuilot, Philippe Martinet, Lionel Cordesses, and Jean Gallice. Position based visual servoing: keeping the object in the field of vision. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 2, pages 1624–1629. IEEE, 2002.

[Traslosheros 2007] A Traslosheros, JM Sebastian, L Angel, F Roberti, and R Carelli. Visual servoing of a parallel robot system. In *2007 IEEE International Symposium on Intelligent Signal Processing*, pages 1–6. IEEE, 2007.

[Vignolo 2014] Alessia Vignolo, Sébastien Briot, Philippe Martinet, and Chao Chen. Comparative analysis of two types of leg-observation-based visual servoing approaches for the control of the five-bar mechanism. In *Proceedings of the 2014 Australasian Conference on Robotics and Automation (ACRA 2014). University of Melbourne, Australia*, 2014.

[VISP] Visual Servoing Plateform (ViSP). Tutorial : How to boost your visual servo control law. version 3.1.0. URL:http ://visp-doc.inria.fr/doxygen/visp-daily/tutorial-boostvs.html.

[Voglewede 2004] Philip A Voglewede and Imme Ebert-Uphoff. Measuring" closeness" to singularities for parallel manipulators. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 5, pages 4539–4544. IEEE, 2004.

[Wang 2016] Geng Wang and Guoqiang Ye. Spherical image based visual servoing via nonlinear model predictive control. In *2016 Seventh International Conference on Intelligent Control and Information Processing (ICICIP)*, pages 223–228. IEEE, 2016.

[Wang 2017] Ruizhou Wang and Xianmin Zhang. Optimal design of a planar parallel 3-dof nanopositioner with multi-objective. *Mechanism and Machine Theory*, 112:61–83, 2017.

[Weiss 1987] LEEE Weiss, ARTHURC Sanderson, and CHARLESP Neuman. Dynamic sensor-based control of robots with visual feedback. *IEEE Journal on Robotics and Automation*, 3(5):404–417, 1987.

[Wu 2010] Chao Wu, Xin-Jun Liu, Liping Wang, and Jinsong Wang. Optimal design of spherical 5r parallel manipulators considering the motion/force transmissibility. *Journal of Mechanical Design*, 132(3), 2010.

[Wu 2014] Guanglei Wu, Stéphane Caro, Shaoping Bai, and Jørgen Kepler. Dynamic modeling and design optimization of a 3-dof spherical parallel manipulator. *Robotics and Autonomous Systems*, 62(10):1377–1386, 2014.

[Xie 2016] Hui Xie, Geoff Fink, Alan F Lynch, and Martin Jagersand. Adaptive visual servoing of uavs using a virtual camera. *IEEE Transactions on Aerospace and Electronic Systems*, 52(5):2529–2538, 2016.

[Yahya 2016] MF Yahya and MR Arshad. Position-based visual servoing for underwater docking of an autonomous underwater vehicle. In *2016 IEEE International Conference on Underwater System Technology: Theory and Applications (USYS)*, pages 121–126. IEEE, 2016.

[Zake 2019] Zane Zake, Stéphane Caro, Adolfo Suarez Roos, François Chaumette, and Nicolò Pedemonte. Stability analysis of pose-based visual servoing control of cable-driven parallel robots. In *International Conference on Cable-Driven Parallel Robots*, pages 73–84. Springer, 2019.

[Zake 2019] Zane Zake, François Chaumette, Nicolò Pedemonte, and Stéphane Caro. Vision-based control and stability analysis of a cable-driven parallel robot. *IEEE Robotics and Automation Letters*, 4(2):1029–1036, 2019.

[Zake 2020] Zane Zake, François Chaumette, Nicolò Pedemonte, and Stéphane Caro. Robust 2 1/2d visual servoing of a cable-driven parallel robot thanks to trajectory tracking. *IEEE Robotics and Automation Letters*, 2020.

[Zhang 1999] Hong Zhang and James P Ostrowski. Visual servoing with dynamics: Control of an unmanned blimp. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, volume 1, pages 618–623. IEEE, 1999.

[Zhang 2018] Kaixiang Zhang, Jian Chen, and François Chaumette. Visual servoing with trifocal tensor. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 2334–2340. IEEE, 2018.

[Zhao 2015] Yi-Min Zhao, Wen-Fang Xie, Sining Liu, and Tingting Wang. Neural network-based image moments for robotic visual servoing. *Journal of Intelligent & Robotic Systems*, 78(2):239–256, 2015.

[Zitzler 2001] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. Spea2: Improving the strength pareto evolutionary algorithm. *TIK-report*, 103, 2001.

[Zlatanov 1994] Dimiter Zlatanov, Robert G Fenton, and Beno Benhabib. Singularity analysis of mechanisms and robots via a motion-space model of the instantaneous kinematics. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 980–985. IEEE, 1994.

Titre : Conception de robot orientée commande

Mots clés : robots parallèles , asservissement visuel , conception orientée commande , robot caché, moment dans l'image

Résumé : Il est bien connu que les robots parallèles ont de nombreuses applications dans l 'industrie. Cependant, en raison de leur structure complexe, leur contrôle peut être difficile. Lorsqu' une précision élevée est nécessaire , un modèle complet du robot détaillé est nécessaire . Les approches de contrôle référencées capteurs se sont avérées plus efficaces , en termes de précision que les contrôleurs basés modèles puisqu 'elles s'affranchissent des modèles de robots complexes et des erreurs de modélisation associées. Néanmoins, lors de l'application de d' un asservissement visuel , il y a toujours des problèmes dans le processus de contrôle , tels que les singularités du contrôleur . Cette thèse propose une méthodologie de conception orientée commande qui prend en compte les performances de précision du contrôleur dans le processus de conception du robot pour obtenir les paramètres géométriques optimaux de ce dernier

Dans le cadre de ce travail de thèse , il a été question d'appliquer la méthodologie de conception orientée commande à la conception optimale de trois types de robots parallèles : le mécanisme cinq barres , le robot DELTA et, enfin , la plate-forme de Gough-Stewart . Trois contrôleurs ont été sélectionnés dans le processus de conception du robot : les commandes basées sur l'observation des directions des jambes, les commandes basées sur l'observation des lignes et les commandes basées sur des moments dans l'image .Pour vérifier les performances en terme de précision des robots optimisés, nous avons effectué des co-simulations des robots optimisés avec les contrôleurs correspondants .

En terme d'expérimentation, deux prototypes de robots DELTA ont été conçus et expérimentés afin de valider la précision du contrôleur.

Title : Control-based design of robots

Keywords : parallel robots, visual servoing, control-based design, hidden robot, image moment.

Abstract : It is well-known that parallel robots have a lot of applications in industry for their high stiffness , high payload , can reach higher acceleration and speed . However , because of their complex structure , their control may be troublesome. When high accuracy is needed, the detailed robot model is necessary . However , even detailed models still suffer from the problem of inaccuracy in reality because of robot assembly and manufacturing errors . Sensor-based control approaches have been proven to be more efficient than model-based controllers in terms of accuracy since they overcome the complex robot models and inconsistency errors. Nevertheless, when applying the visual servoing, there are always some problems in the control process , such as the controller singularities . Thus , this thesis proposes proposes a control-based design metodology which takes into account the accuracy performance of the controller in the design process to get the geometric parameters of the robot.

This thesis applied the control-based design methodology to the optimal design of three types of parallel robots: Five-bar mechanisms , DELTA robots , Gough-Stewart platforms . Three types of controllers are selected in the design process : leg-direction-based visual servoing, line-baesd visual servoing and image moment visual servoing . Design optimization problems are formulated to find the geometric parameters of the robot . Co-simulations are performed to check the accuracy performance of the robots obtained from the optimization.

Experiments are performed with two DELTA robot prototypes in order to validate the controller accuracy.