Stereo visual servoing from straight lines

Fadi Alkhalil

UNIVERSITÉ DE STRASBOURG

École Doctorale [MSII]

[LABORATOIRE : LSIIT]

[ÉQUIPE de RECHERCHE : AVR]

THÈSE présenté par :

[Fadi ALKHALIL]

soutenue le : 24 Septembre 2012

pour obtenir le grade de : Docteur de l'université de Strasbourg

Discipline/ Spécialité : Signal, Image, Automatique et Robotique

TITRE de la thèse

[Stereo visual servoing from straight lines]

THÈSE DIRIGÉE PAR :

DOIGNON Christophe	Professeur, université de Strasbourg

RAPPORTEURS :

ANDREFF Nicolas	Professeur, FEMTO-ST/AS2M, Besançon
LABBANI-IGBIDA Ouiddad	MCF HDR, MIS-UPJV, Amiens

JURY :

PERDEREAU Véronique	Professeur, ISIR-UPMC, Paris Président du jury
LABBANI-IGBIDA Ouiddad	MCF HDR, MIS-UPJV, Amiens
ANDREFF Nicolas	Professeur, FEMTO-ST/AS2M, Besançon
LAROCHE Edouard	Professeur, Université de Strasbourg
DOIGNON Christophe	Professeur, Université de Strasbourg

Remerciements

Un travail de trois ans ne pourrait pas se concrétiser sans un ensemble d'aides souvent nécessaires mais toujours précieuses. C'est pourquoi je voudrais remercier ici tout ceux qui ont contribué au bon déroulement de ce travail aussi bien par leur encadrement, leurs compétences que par leurs soutiens.
Les travaux de recherche présentés dans ce mémoire ont été effectués au **L**aboratoire des **S**ciences de l'**I**mage, de l'**I**nformatique, et de la **T**élédétection (**LSIIT**), à l'université de Strasbourg (UMR 7005) au sein de l'équipe du Automatique, Vision, et Robotique. Aussi, je remercie en premier lieu monsieur Michel de Mathelin, Professeur, pour m'avoir accueilli dans son équipe de recherche ainsi que l'IRCAD et le professeur Jacques Marescaux.

Je suis très reconnaissant envers madame Ouiddad Labbani-igbida, maître de Conférences à l'université de Picardie Jules Verne (UPJV), Amiens, et envers monsieur Nicolas Andreff, Professeur à L'institut FEMTO-ST/AS2M, de Besançon, d'avoir acceptés d'être les rapporteurs de ma thèse.

Je remercie aussi particulièrement madame Véronique Perdereau, Professeur à l' université Pierre et Marie Curie, Paris, et monsieur Edouard Laroche, Professeur à l'université de Strasbourg, pour l'honneur qu'ils me font d'avoir accepté d'examiner mon travail et de faire partie de mon jury de thèse.

Je tiens à remercier très chaleureusement mon directeur de thèse, Christophe Doignon, pour le temps immense qu'il a consacré à l'encadrement de ma thèse, une disponibilité d'esprit considérable qui m'a fait découvrir la recherche tout au long de ces dernières années. Son rôle, ses critiques constructives, ses précieux conseils et ses propositions m'ont permis de mener cette thèse jusqu'à la fin.

Je voudrais aussi remercier tous les doctorants et les collègues de l'équipe AVR pour leur soutien durant ces années passées avec eux. Mes remerciements vont également envers tous les chercheurs de l'IRCAD que j'ai rencontré. J'adresse mes chaleureux remerciements à tous mes amis, doctorants, pour les jours que nous avons partagé ensemble et aussi pour leur sympathie.

Enfin, je ne remercierai sans doute jamais assez à ma chère famille qui m'a accompagné et supporté sans trêve depuis toujours.

Abstract

Closing the control loop of a manipulator robot with vision feedback is widely known. It concerns nowadays all areas of robotics. Such a return can make a comparison between a desired state and current state, using visual measurements.

The main objective of this doctoral thesis is to design several types of kinematic control laws for stereo visual servoing. It strongly involves the formalism of the task function which is a well-known and useful mathematical tool to express the visual error as a function of state vectors. Then the proofs of the stability of the closed loop system and the convergence study of the task functions associated with these laws through the Lyapunov theory are provided.

We have investigated the decoupling between the rotational and translational velocities control laws together with the epipolar constraint with a stereo visual feedback. That is why, the visual measurements and features used in this thesis are the 3D straight lines. The interests of this type of visual features rely on the robustness against the noise, and the possibility to represent straight lines or other features like points or planes pairs by the Plücker coordinates, as a 3D straight line can be represented as well by two points or the intersection of two planes. This makes all the control laws designed in this thesis valid for another visual features like points pair or plane pairs in any relative configuration.
These control laws have been applied to the motion control of a 6 DOF manipulator robot, with respect to a rigid or articulated object of interest.

Keywords: Decoupling control laws, Stereo visual servoing, Straight lines.

Résumé

L'emploi d'un retour visuel dans le but d'effectuer une commande en boucle fermée de robot s'est largement répandu et concerne de nos jours tous les domaines de la robotique. Un tel retour permet d'effectuer une comparaison entre un état désiré et l'état actuel, à l'aide de mesures visuelles. A cette fin, le formalisme de la fonction de tâche est un outil mathématique bien connu pour exprimer le retour et établir l'erreur d'asservissement en fonction des vecteurs d'état.

L'objectif principal de cette thèse consiste à concevoir plusieurs types de lois de commande cinématiques par vision stéréo. Ceci concerne aussi l'étude de la stabilité du système en boucle fermée et la convergence des fonctions de tâches associées à ces lois grâce à la théorie de Lyapunov.

C'est essentiellement le découplage des lois de commandes cinématiques en rotation et en translation qui est recherché ici, selon le nombre d'indices visuels considérés. Les simulations ont confirmé les résultats théoriques, avant que ces lois de commande soient appliquées au déplacement d'un robot manipulateur vis-à-vis d'un objet d'intérêt rigide ou articulé.
Les mesures visuelles utilisées dans cette thèse sont les lignes droites 3D. Les intérêts apportés à ce type de mesures visuelles sont la robustesse contre le bruit, et la possibilité de représenter d'autres primitives comme des couples de points ou de plans par la modélisation de Plücker. Comme une droite 3D peut être représentée par deux points ou l'intersection des deux plans, toutes les lois de commande conçues dans cette thèse sont valables pour d'autres mesures visuelles, comme des couples de points ou de plans.

Mots clés : Découplage des lois de commande, Asservissement visuel stéréo, Droites.

Publications

Communications lors de conférences internationales

1. **IROS 2012**, Fadi ALKHALIL and Christophe DOIGNON, Stereo visual servoing with decoupling control, IEEE/RSJ Int'l Conference on Intelligent Robots and Systems, 7-11 Octobre 2012, Vilamoura-Algarve, Portugal.

2. **CIFA 2012**, Fadi ALKHALIL, Laurent BARBE et Christophe DOIGNON, Commandes cinématiques découplées pour un asservissement visuel stéréo, 7ème Conférence Internationale Francophone d'Automatique, 4-6 juillet 2012, Grenoble, France.

3. **IEEE Rose 2011**, Fadi ALKHALIL, Christophe DOIGNON and Laurent BARBE, Visual servoing of an articulated object based on stereo-vision, IEEE International Symposium on Robotic and Sensors Environments (ROSE), 17-18 Sept. 2011, Montreal, QC, Canada.

Journées nationales et doctorales

4. **ORASIS 2011**, Fadi ALKHALIL, Christophe DOIGNON et Pierre GRAEBLING, Asservissement visuel de droites par commande découplée et stéréo-vision, ORASIS - Congrès des jeunes chercheurs en vision par ordinateur, mai 2011, Praz-sur-Arly, France.

5. **JJCR 2009**, Fadi ALKHALIL, Christophe DOIGNON, et Pierre GRAEBLING, Poster : Asservissement visuel sur des objets articulés, JNNR 7ème Journées Nationales de la Recherche en Robotique, novembre 2009, Neuvy-sur-Barangeon, France.

Contents

Chapter 1

Introduction

1.1 Motivation

Visual servoing is considered as one of the most important field of artificial and robot vision. It aims at providing a mobile system with visual control laws and the control of the entire closed loop, to give the system more autonomous capabilities. The visual servoing methods are usually classified in two main categories. The Position Based Visual Servoing (PBVS), and the Image Based Visual Servoing (IBVS)[1]. In the first kind of servoing (PBVS), the goal of the vision system is to estimate the relative pose (position and orientation) between the camera frame and the object frame, then compute the error of servoing in the Euclidean space.
While in IBVS, the servoing error is directly defined in the image plane. It usually avoids the system a static error in the final pose, because with this kind of servoing, one does not need the Euclidean recovery nor perspective model of the camera in the feedback chain.

In this PhD thesis, we hope to make a step forward in the visual servoing field, by designing a novel kind of visual servoing, called "Stereo Visual Servoing" (SVS), especially dedicated to stereovision systems.
Using a stereo vision system gives many advantages, like the additional information supplied by the second camera, the depth estimation, without forgetting the possibility of 3D reconstruction for the most types of simple geometrical features (points, lines, circles, cylinders, ...).
We will go further to the classical visual servoing, by defining a stereo servoing space, which is not an Euclidean space, nor the image plane. This space makes the features in the two images available to be used by new state vectors.

[1] but other hybrid and other partitioned approaches exist and may be equally used for classification purpose.

1.2 Problem formulation

The visual servoing aims at closing the control loop with vision feedback. A simple block diagram of the closed control loop is shown in figure (1.1)

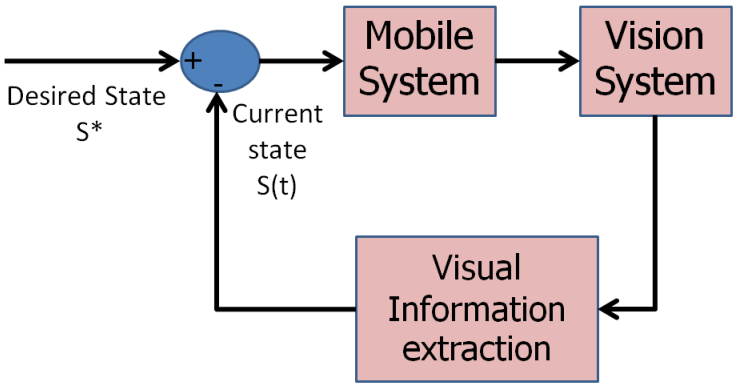


Figure 1.1: A Simple visual servoing block diagram.

The desired state S^* in the above block diagram represents the reference vector. $S(t)$ represents the current state vector, which evolves along the time and is of the same type of S^*. The goal of the servoing is then to drive $S(t)$ towards S^*. Usually, at the end of servoing, one would get $S = S^*$.
The mobile system can be the robot end-effector which holds the vision system, in the "eye-in-hand" configuration, it can be a mobile object in the field of view of a motionless vision system in the world space - also called the "eye-to-hand" configuration -, or both.

The nature of the state vector defines the kind of the visual servoing. If the state vector is a 3D parameter, then, the visual servoing is based on the position (PBVS). While if the state vector is an image feature, then the visual servoing is expressed in the image plane (IBVS).
The difference between the desired and current state vectors defines the error of servoing. This error is then the output of the task function $e(t)$. Sometimes a matrix $\mathtt{C}$ relates the task function with the servoing error $e(t) = \mathtt{C}(S(t) - S^*)$. $\mathtt{C}$ is then called the *combinaison matrix* and it may account for the redundancy or robustness (statistics) in the error definition.

The time derivative of the error gives $\dot{e} = \dot{S}$, when S^* is assumed to be constant (it can be a time variant in a spatial task [MDGD97]).
The relation between the time derivative of state vector and the camera motion is

defined by a non linear matrix equation

$$\dot{S} = \mathtt{L_s}\tau \ .$$

$\tau = (\mathbf{V}, \mathbf{\Omega})^\mathsf{T}$ is the kinematic screw of the camera velocity, $\mathtt{L_s}$ is called the interaction matrix related to S, it is often a function of the state (sometimes, the subscript notation relies on the velocity screw, or part of it). The term feature Jacobian is also used somewhat interchangeably in the visual servoing literature.
If we wish an exponential decrease of the error (i.e., $\dot{e} = -\lambda\ e$) we obtain

$$\begin{aligned} \dot{S} &= \mathtt{L_s}\tau \\ \dot{S} &= -\lambda\ e \\ \tau &= -\lambda\ \mathtt{L_s}^{+}\mathtt{e} \end{aligned}$$

$\mathtt{L_s}^{+}$ is the pseudo inverse of $\mathtt{L_s}$ in general cases, and can be the direct inverse of $\mathtt{L_s}$ if it is invertible. In real visual servoing systems, it is impossible to perfectly know in practice either $\mathtt{L_s}$ or $\mathtt{L_s}^{+}$. So an approximation or an estimation of one of these two matrices must be realized.
We note $\widehat{\mathtt{L_s}^{+}}$ for the estimation of the pseudo inverse $\mathtt{L_s}^{+}$ and $\widehat{\mathtt{L_s}}^{+}$ for the pseudo inverse of the estimation of $\mathtt{L_s}$.
However, the goal of the visual servoing is to recover the kinematic screw τ which serves as control input vector. So, one needs to find the inverse of the interaction matrix $\mathtt{L_s}$. As the interaction matrix $\mathtt{L_s}$ is of dimensions $n \times m$, and generally $n \neq m, m \leq 6$, this matrix does not have a direct inverse, but a pseudo inverse $\mathtt{L_s}^{+}$. If this pseudo inverse exists, the kinematic screw is given by

$$\tau = -\lambda\widehat{\mathtt{L_s}^{+}}\ e \ .$$

The principal problem we focus in this thesis is to define $\mathtt{L_s}^{+}$ to control the motion by means of stereo images, and first of all, its existence. The pseudo inverse is given by $\mathtt{L_s}^{+} = (\mathtt{L_s}^\mathsf{T}\mathtt{L_s})^{-1}\mathtt{L_s}^\mathsf{T}$, so, the matrix $\mathtt{L_s}^{+}$ exists when $\mathtt{L_s}^\mathsf{T}\mathtt{L_s}$ is invertible.
The dimensions of $\mathtt{L_s}^\mathsf{T}\mathtt{L_s}$ are $m \times m$. With $m = 6$, finding an analytical inverse may be impossible, and proof of its existence is generally very difficult. This is mainly due to the high dimensionality of the system.
One suitable way to reduce the dimensions of $\mathtt{L_s}$ is to separate the rotational velocity control law $\mathbf{\Omega}$ and the transitional velocity control law $\mathbf{V}$. This separation makes the interaction matrix divided in two interaction matrices $\mathtt{L}_\Omega$ and $\mathtt{L}_V$, each of them is of dimensions up to $n \times 3$ with rigid object of interest.

For visual servoing purpose, not only the existence of the inverse of the interaction matrix is important, but, the behaviour of the system in closed loop is also very

important. We mean by the system behaviour here, the stability of the system in the closed loop, and the asymptotic convergence of the task functions. To prove the asymptotic stability of $\mathbf{V}, \boldsymbol{\Omega}$ all over the servoing, the separation is not enough. The control laws must be decoupled, therefore independent task functions for translation and rotation motion are targeted. This means to define two kinds of state vectors, one is related to the translation motion, and the other one is related to the rotation motion.

The changes in the features of the object of interest like points, lines, circles, ... are related to both motions (rotation and translation), and these features should allow to define independent task functions, so to achieve the decoupling of the control laws.
To do this, the state vectors will be a combination of these features. Then, the objective is to find a time-invariant rotational state vector S_Ω relative to the translation motion, which means

$$\frac{d}{dV} S_\Omega = 0$$

and a time-invariant translational state vector S_V relative to the rotation motion

$$\frac{d}{d\Omega} S_V = 0 \ .$$

If such state vectors may be defined from the visual mesurements, one get two independent task functions $e_\Omega(t) = S_\Omega(t) - S_\Omega^*$ for the rotation, and $e_V(t) = S_V(t) - S_V^*$ for the translation. These two independent task functions will contribute to design two decoupled velocities control laws

$$\begin{aligned} \mathbf{V(t)} &= -\lambda_V \ \mathtt{L}_V^+ \ e_V(t) \\ \boldsymbol{\Omega}(t) &= -\lambda_\Omega \ \mathtt{L}_\Omega^+ \ e_\Omega(t) \ . \end{aligned}$$

With this full decoupling between the rotational and translational control laws, it is easier to get an analytical study of stability and convergence. It is worth pointing out that other separation techniques exist (other than the rotational and the translational control laws), but the Plücker line modeling is promoting this kind of decoupling.

1.3 Proposed solution

As already explained, to reduce the dimensions of the interaction matrix $\mathtt{L_s}$, one needs to separate the kinematic screw τ in two velocities control laws $\mathbf{V}, \boldsymbol{\Omega}$. To go towards the proof of the asymptotic stability of these control laws, they must be

decoupled. This decoupling can be ensured with independent task functions. To this purpose, one should build state vectors S_V for the translation motion and S_Ω for the rotation motion. These state vectors are depending on the vision system and on the salient and visible features of the object of interest.
The selected features in this thesis are the 3D straight lines. The vision system used is a stereo system, because it is useful in the 3D reconstruction of the features, and in the construction of the state vectors.

For PBVS, we succeeded in designing two pairs of rotational and translational velocities control laws. The first pair depends on the estimation of the relative rotation matrix $\mathtt{R}$, and the relative translation vector t between the current position frame and the desired one. In this case, the rotational state vector is selected as one of the rotation matrix $\mathtt{R}$ representation, and the translational state vector represents the translation vector between the two frames origins.
The second pair of velocities control laws is defined using the bi-normalized Plücker coordinates. The rotational state vector S_Ω is the direction vector v of the 3D straight line while the orthogonal distance d is used in the translational state vector S_V.

For the classical IBVS, the state vectors are defined in the image plane and we did not succeed in finding independent state vectors for rotational and translational task functions. Then, we define a "stereo space" for expressing the servoing. It is built from the projective and affine properties of the two views. In this space, the state vectors are a combination of the features in the stereo images pair. Two independent task functions for rotation and translation motion in the "stereo space" have been presented. With the given epipoles and the homography of the plane at infinity, $\mathtt{H}_\infty$, one can transfer the feature in one image plane to the other image plane or the stereo space, to build the state vectors.

1.4 Organization of the manuscript

The manuscript is organized in 3 core chapters, enclosed by an introductory - the current chapter - and a concluding chapter. Below, a chapter-by-chapter overview on contributions and impact is given. The organization of the chapters themselves follows the general guideline: State-of-the-art - Stereo position based visual servoing from lines - Stereo image based visual servoing from lines.

- **State-of-the-art** In this chapter, we review some previous works in the literature concerning some artificial vision applications. Considering the visual

servoing is one of the most important field of the artificial vision and robotics, we present several works interested of the position and image servoing specially the use of multi-cameras. Then we look for the modeling of a 3D straight line, as it is the visual feature used for all the visual servoings presented in this thesis. At the end of the state-of-the-art, we present the most close works of our interest, the decoupling between the velocities control laws.

- **Stereo position based visual servoing from lines** In this chapter, the velocities control laws are designed using 3D state vectors. We work on the decoupling between the rotational and the translational velocities control laws. To this end, a new state vector is proposed based on the Plücker coordinates of a 3D straight line. The stability of the system in closed loop and asymptotic convergence of the task functions are proved with the help of Lyapunov functions.

  The velocities control laws designed in this chapter are also used to control the pose of a non-rigid object, in particular, an articulated one. First they are developed to control the 4 DOF of an object represented by only one 3D straight line. Then the angular velocity control laws are applied to control the joints angles of an articulated object, which is assumed to be composed of several straight lines linked with revolute joints only.

- **Stereo image based visual servoing from lines** The interaction matrix proposed in the literature and the interaction matrix designed in our work are not satisfying for ensuring the global asymptotic stability of the servoing system.

  Then, a decoupling between the translational and the rotational velocities control laws is proposed so as trying to reduce the local minima. To do that, we define the stereo space. In this virtual space the two images planes are considered. That makes the features in the two images available to build the state vectors.

  With the new independent state vectors for translation and the rotation motion, we design decoupled velocities control laws. Then we prove the global asymptotic stability (GAS) of the system in closed loop, together with the improvement in the control behaviour.

Chapter 2

State of the art

The following synopsis on the state-of-the-art aims at situating this work with respect to the main approaches and the key-papers in the field. We concentrate on the visual servoing applications first, as it is the domain of the work in this thesis.

2.1 Vision applications

The artificial vision science is very large and impossible to cover entirely in this introduction, but some of its real-time applications which concern the visual servoing attract our attention. Task requirements in artificial vision applications have become increasingly complicated with advanced sciences, especially for robotics.

Navigation and exploration of unknown areas is a fundamental task for a mobile robot. Avoiding the obstacles without any human intervention makes this task more complicated. Omnidirectional vision offers a large field of view and a complete information about the world surrounding the robot that could allow an autonomous and safe navigation of the robot only using one catadioptric sensor [Mou05].
With only omnidirectional vision, Merveilleux *et al.* [MLIM11] realized this task in real-time. They use the active contours to realize a fast free space extraction, defined as homogeneous area, designed from the elevated parts around the robot in the image.
The authors have improved energy approximations and weighting parameters definition, in order to make a fast and adapted contour expansion. Moreover, the active contour is initialized from the omnidirectional projection of the scene surrounding the robot. These energy definitions and approximations lead to a fast and adequate evolution of the initial contour to the free space boundaries in real omnidirectional images.

The space applications are one of important tasks of the artificial vision. Dionnet and Marchand have proposed a real-time, robust and efficient 3D model-based tracking algorithm for visual servoing in [DM07]. The goal was to develop a robot demonstrator able to grasp complex objects by visual servoing in space environment.
The authors used Virtual Visual Servoing (VVS) for the pose computation. The pose computation is considered as an optimization problem similar to image-based visual servoing.
They developed a robust and fast 3D tracking algorithms for visual servoing applications in a space context. Indeed, the algorithm has to tackle several space specific problems, like the lack of computing power, since resource (energy, volume, mass) and environmental (thermal dissipation, radiation compatibility) constraints limit all the performances of computers that may be used in space.

Lu and Manduch in [LM05] present new, robust algorithms to detect and precisely localize curbs and stairways for autonomous navigation. Detection and localization of border lines are important for two reasons. First, a robot may not be able to negotiate a curb or, if possible, may need to precisely align with the edge of pavement. Second, monitoring and following a line represents a simple and effective pavement percept-referenced behavior for urban navigation.
These algorithms combine brightness edge information with 3D data from a stereo system. The strategy is to use information from both images and geometry, which, in a stereo system, are perfectly registered together.
In fact, even a single image conveys a good amount of information about the presence of a curb, since a curb line normally looks like a brightness edge. However, several other edges, usually co–exist in an image, and thus brightness alone would not be sufficient for reliable detection.
Their system finds application in indoor and urban autonomous navigation, where curbs and stairways are important landmarks for world modeling.

The interest in using cameras in simultaneous localization and mapping (SLAM) has grown tremendously in recent years. Current visual SLAM research has been focused on the use of either monocular or stereo vision to obtain 3D information from the environment.
Paz *et al.* [PPT08] used the stereo rig as the only sensor for 6 DOF visual mapping of large environments to be efficiently and accurately carried out. They show the advantages of being able to accommodate both monocular and stereo information in carrying out a 6 DOF SLAM with a hand-held camera. The ability of a stereo assembly to directly and immediately provide 3D landmark estimates allows to use

the best available SLAM algorithms and rapidly obtain good results with little effort in the conceptual parts.
Solà *et al.* [SMDVC08] used multi camera in SLAM too. They consider each camera as an independent sensor rather than the entire set as a monolithic super sensor. The visual data are treated by monocular methods and fused by the SLAM filter. Several advantages naturally arise as interesting possibilities, such as the desynchronized firing of the sensors, the use of several unequal cameras, self-calibration, and cooperative SLAM with several independently moving cameras.

2.2 Visual servoing

Using visual feedback to control a robot is commonly termed "visual servoing" [HHC96]. The first visual servoing systems was reported in the 1970's with Shirai and Inoue. They developed a method for entering a cubic object using a robot and a static camera [SI73]. They have shown the value of a feedback loop both to correct the robot position but also to improve accuracy. As the camera provides the system with an enormeous quantity of information extracted from a sequence of images, a two-step iterative scheme had appeared: "static look and move".
With the works of Agin [Agi77], [Agi79], it appears that there, for the first time, the term visual servoing on the assembly of mechanical parts. This is actually what we call the "dynamic look and move" that allows us to consider objects in the scene in motion.
In the work of Hill and Park in 1979 [HP79], the position of the end-effector of a robot is controlled through the computation of the relative pose between the visual sensor and the target: it is the visual servoing based on the 3D position (Position-Based Visual servoing). Sanderson and Weiss in [SW80] introduce an important classification of visual servoing structures based on two criteria, space of control and presence of joint feedback. So in this classification one distinguishes two main approaches:

- Position based control: the image features are extracted from the image of the target, and with a modelling, it is used to determine the pose of the target with respect to the camera frame.
- Image based control: in this kind of servoing, the pose estimation is omitted (even if the depth appears in the image Jacobian and needs the partial pose computation), and the control laws are directly expressed in the image plane [Cha90].

With the work due to Chaumette in [Cha90], both the task function formalism is applied to the image-based visual servoing, but also a virtual link is establish be-

tween the camera and the object of interest. Many interaction matrices are then reported and visual seroing are provided for the related visual features used.
A mixed 2D–3D approach can be used in order to take advantage of both approaches. This is done in the so-called 2D$\frac{1}{2}$D visual servoing (Malis, Chaumette, and Boudet [MCB99], Chaumette and Malis [CM00]). This approach consists of combining visual features obtained from the image and several features expressed in the Euclidean space. For more details, please consult ([MC00],[CH06],[CH07]) for instance.
The notion of task function has been introduced by Samson *et al.* [SBE91] to define the error of convergence. The task function measures the difference between the current state vector (or features) and the desired one. The goal of servoing is to annul the task function value. The desired decrease of the task function is usually to be at exponential form, primarily to the design of a more efficient controller.

2.2.1 Position Based Visual Servoing

Doignon *et al.* in [DAO94a] presented the first real-time positioning of a 6 DOF robot manipulator with a monocular vision, in Strasbourg university. They worked with the dynamic "look and move" configuration for real-time low frame rate visual feedback ($T_e = 80$ ms) by means of points and lines and binary image vision.
The classical visual servoing techniques have been extended in [MCB00], to the use of several cameras observing different parts of an object. The multi-camera visual servoing has been designed as a part of the task function approach. The particular choice of the task function simplifies the design of the control law and the stability analysis.
Ly *et al.* in [LZG10], propose a simple and accurate method for needle localization. Using stereo visual servoing techniques based on projective geometry and perspective invariants. Their method requires no additional sensors (infrared, laser, ultrasound, MRI, and etc), according to the parameters (intrinsic and extrinsic matrices of cameras, etc), a parameterized experimental model is established. The only human interaction required by the system is the choice of the needle entry point on the patient.
The eye-to-hand and eye-in-hand configurations are foregathered in [QSL+06], to realize the visual servoing on Puma 562 manipulator robot with a camera attached on the end-effector and an active stereo rig over the head of the robot manipulator. Under this configuration, the target pose can be computed by using the stereo images, and then the robot can be controlled to track and align with the target.
While the first related work is using monocular vision, the two other ones involves sterovision systems. All of these works highlight a necessary accurate calibration previously to the pose computation.

2.2.2 Image Based Visual Servoing

A visual servoing approach to the problem of object grasping and more generally, to the problem of aligning an end-effector with an object is presented by Horaud *et al.* in [HDE98]. The authors show how to represent a grasp or more generally, an alignment between two solids in three-dimensional (3D) projective space using an uncalibrated stereo system. Such a 3D projective representation is view-invariant in the sense that it can be easily mapped into an image set-point without any knowledge about the camera parameters.
Zhang and Ostrowski [ZO02] have presented the motion planning of a wheel robot directly in the image, rather than transfer the image feature back to the robot pose. They developed nominal paths that are based in the same space as the sensor measurements. With this idea, they eliminate unnecessary modeling errors due to extrinsic camera calibration parameters. They also enlarged the feature space to demonstrate the feasibility of tracking two targets simultaneously, which helps to remove the ambiguity in the configuration space that results from tracking a single object.
Beside the camera, Krupa *et al.* [KGdM+03] used laser spot for distance estimation. They control a 4 DOF medical robot for minimally invasive surgery, that is by means of a 6 DOF robot with only 4 actuated joints. 1 DOF for rotation controlled with image servoing of three marks on the surgical instrument tips, while 1 DOF and 2 DOF for the distance and rotation respectively are controlled by the estimation of the distance between the end of the robot and the projected laser spot on the target. The surgical instrument is mounted on the end-effector of a surgical robot (AESOP 3000). The goal of the automated task is to bring the instrument at a desired location from an unknown or hidden pose. This approach is successfully validated in a real surgical environment by performing experiments on living animals in the surgical training room of IRCAD.
In the first paper, only 2 DOF have be controlled by vision feedback while in the second one the robot has 4 DOF but an external laser source is used beside the camera to estimate the relative distance (robot end-effector - target), like a very simple structured light system. These examples validate our choice of using stereo rig to control an object with 6 DOF.

2.2.3 Multi-cameras visual servoing

The most part of visual servoing systems use only one camera for several reasons. The simplicity of the hardware and software system design (including the mysyn-

chrinzation of the acquisition), to reduce the necessary time for image processing, and to reduce the cost of the system. However, the loss of the 3D information can complicate the design of the control if this information has to be estimated on-line. Using multi-cameras configuration is a solution for estimating 3D information. The target must stay in the field of view of all the cameras. The most well-known configuration is the stereo system (i.e. two cameras rigidly linked) since it allows, if the system is fully calibrated, to get the complete Euclidean reconstruction of the observed objects.

With more than one camera, 3D reconstruction is possible in general case. Even in IBVS, the 3D information is useful in visual servoing like in [CBM02] where the authors show the effectiveness of the servoing task can be improved if estimated 3D features are used instead of raw image data.
Theoretical developments show how 3D control features are extracted from stereo images, and the interaction matrix is computed for raw pixels, estimated 3D coordinates, and a new feature vector which uses stereo disparity.
Andersson [And89] has probably made one of the first robot control system with stereo vision. The application is dedicated to a robot playing table tennis. He takes advantage of redundant degrees of freedom in the task, as well as in the robot. New sensor data, proprioception, and internal feedback are fused to produce a system that continually optimizes its task plan.
In other application, the 3D reconstruction or 3D informations are not the goal of a stereo rig, only the redundancy of the image features in different cameras are useful to build an interaction matrix by stacking the two camera Jacobians [MC01]. The null space of this interaction matrix is empty, which ensures the convergence toward the desired images.

Ruf and Horaud [RH99], avoid the very difficult metric calibration for some robotic applications. They presented an approach to model the geometry of a pan-tilt mechanism relative to a weakly calibrated stereo-rig, mounted on top of it. This active sensor is useful for several applications like the tracking of moving objects, 3D reconstruction, and visual servoing. The authors introduce a new approach and a new mathematical framework to represent both the camera-pair geometry and the pan-tilt mechanism in projective space. The proposed method is highly accurate. The proposed mathematical formalism for the modeling of a kinematic chain is sound and complete.

Chandraker *et al.* [CLK09] present a fast and robust algorithms for estimating the structure and motion using a stereo vision, with straight lines as features. The visual input for these algorithms stems from a calibrated stereo rig.

These algorithms use only two or three lines for motion estimation. Rather than resorting to a non-linear minimization, they reduce the problem to a small polynomial system of low degree, which can be solved faster.
The use of calibrated stereo cameras also makes it possible to come up with simpler solutions that firstly reconstruct the 3D lines for each stereo pair and then compute a rotation and translation to align them.

2.2.4 The interaction matrix

The interaction matrix is very important in the behaviour of the control law. Here some works discussed the importance of this matrix and proposed some solutions for the convergence of the system in the closed loop.
Marey and Chaumette [MC08] analyze and compare five image based visual servoing control laws. Three of them are classical ones, while two new ones are proposed. They found that all control laws used in image-based visual servoing have their respective drawbacks and strengths. In some cases, a control law is not able to converge while the others succeed. In other cases, all classical control laws may fail. They propose a new configuration, this configuration has been found by studying a new control scheme built to try to achieve its global asymptotic stability.
Thanks to the interaction matrix, image features are predicted in [AC09], which helps in IBVS for large displacements or rotations around the optical axis. The difference between the reference features and the predicted features is to be minimized over the predicted horizon in regard to the camera velocity screw inputs. Another research work did not use the interaction matrix, and that because the difficulty of defined this matrix, they replace it by an image Jacobian estimated on line during the servoing.
Miura *et al.* [MHGdM05] used the modified simplex optimization techniques for a positioning task by visual servoing. This method does not need a model of the robot and does not require the estimation of Jacobian matrices. Since the simplex method took many iterations near the minimum, a Nelder and Mead-like method with a recursive least squares algorithm for Jacobian matrix estimation was also introduced in order to have quicker convergence near the minimum. This approach is adaptive, and stable near the target if the motions of the robot guarantee sufficient excitation. The Jacobian estimation was carried out using obtained images at the vertices during the simplex iterative search. Once the simplex had roughly converged, the process switched to the Newton-like optimization method. They successfully demonstrated the proposed scheme with simulations and experiments.

In [RMLH99], in contrast to classical approaches, the entire system is modeled by

means of projective geometry. The authors have shown how to projectively represent a kinematic model and how to recover from image measurements only, with no need of metric information.
This representation of robot kinematics is predestined for visual control of a robot in order to perform basic reaching, grasping, or manipulation tasks. In comparison to other works, this approach has a several advantages. Visual servoing can be done along trajectories covering the whole visible configuration space. This formalism opens perspectives for the design of "visual mechanisms".
Pari and Sebastiàn [PS09] developed two methods for estimating the image Jacobian by using line features. The first one takes into account the orientation of normal vector to the interpretation plane of the line, and the second one utilizes the normal representation of the line in the image, but taking into account the Mahalanobis distance to compensate the large difference in magnitude of features. Both methods estimate the image Jacobian with the last N realized moves method. In another work, Pari *et al.* [PSSA08], describes a comparative study of performance between the estimated image Jacobian that comes from taking into account the epipolar geometry of a system of two cameras and the well-known analytic image Jacobian that is utilized for most applications in visual servoing.
Experiments in static and dynamic cases were carried out and showed that the performance of estimated Jacobian by using the properties of the epipolar geometry is such as good and robust against noise as the analytic Jacobian. This fact is considered as an advantage because the estimated Jacobian does not need laborious previous work prior to the control task in contrast to the analytic Jacobian does.
With estimated image Jacobian to relate the kinematic screw, the convergence is obtained in the simulations and experiments, but not proved in theoretical studies. This evaluates our efforts for decoupling the control laws and reducing the interaction matrix dimensions, for simplifying the analytical solutions of proving the stability and convergence of the system in closed loop.

2.2.5 Visual servoing of an articulated object

Previously, non-rigid motion has been classified into three categories describing different levels of constraints on the movement of a body: articulated, elastic and fluid [ACLS98]. In this work the first class of non-rigid motion is considered.
An "articulated" object is defined as a multi-body system composed of at least two rigid components and at most six independent degrees of freedom between any two components [CMC05].
Modeling humans from images involves recovering both body-shape and motion. Most existing approaches can be classified as addressing one or the other of these

two problems, Pan and Liu [PL08] have presented a model of elastic articulated objects based on revolving conic surface and a method of model-based motion estimation. The 3D deformation parameters are determined by corresponding 2D image points and contours with volume invariable constraint from a sequence of stereo images.
Ruff and Horaud [RH99] give a kinematic-chain method for the estimation of articulated motion with an uncalibrated stereo rig. They introduce the notion of projective kinematics which allows rigid and articulated motions to be represented within the transformation group of projective space. The authors link the inherent projective motions to the Lie-group structure of the displacement group. The minimization is determined in projective space and is therefore invariant to camera calibration parameters.
A novel kinematic set approach has been proposed by Comport *et al.* [CMC07]. Their approach consists in symmetrically modeling the motion and the velocity of an articulated object. The advantages of this new model include improved precision, robustness and efficiency, leading to real-time performance. The author uses the Pfaffian velocity constraints for modeling the joint configuration. Then they build a general Jacobian Matrix using the configuration and location of a joint. This matrix relates individual rigid body velocities to an underlying minimal subspace.

2.2.6 Stereo system calibration

With one camera, it is not so easy to obtain the pose without extra information from another sensor, or without a model of the object of interest.
A stereo rig helps to deal with unknown object or environment like in Darius Burschka and Gregory D. Hager [BH05], Guo *et al.* [GMM09]. They present a vision-based system for traffic sign detection and road boundary detection method by combining homography estimation and MRF-based belief propagation to cope with challenging scenarios such as unstructured roads with unhomogeneous surfaces.
Annett Chilian and Heiko Hirschmüller [CH09] have presented a navigation algorithm for mobile robots in unknown rough terrain. The algorithm is solely based on stereo images and is suitable for wheeled and legged robots. The navigation system is able to guide the robot along a short and safe path to a goal specified by the operator and given in coordinates relative to the starting point of the robot.
Using two cameras or a stereo rig is very useful for 3D reconstruction and depth estimation, that can be used for the visual servoing purposes. But, if one needs precession in the 3D estimation, the stereo rig must be calibrated, to estimate the

intrinsic and extrinsic parameters.

In the literature, many works are concerned with camera calibration ([Tsa87], [Zha99], [DA99]) and stereo calibration ([ZBR95], [HC98], [KR00]). Hartley [Har97b] proposed a simplified derivation for Kruppa's equations. This derivation is based on the Singular Value Decomposition of the Fundamental matrix. In [LD00], additional details about this derivation were presented.
Other self-calibration methods rely on retrieving the homography of the plane at infinity [PG99], the absolute quadric ([Har97a], [HA96]) or constraining the Essential matrix [Fus01]. All the above cited methods require at least three images of the same scene. In order to simplify the self-calibration process and improve its accuracy, more recent methods introduced additional constraints from the scene and/or the camera thus reducing the minimum number of images to only two.
Mouaddib *et al.* [MSEY05], define a stereo rig using one camera and multiple mirrors. This way is interesting to create catadioptric omni-directional stereo vision using several mirrors and a single camera.

2.3 Straight line as visual feature

Straight lines are not as widely used as geometrical points in visual servoing. Points are easy to represent in 3D space and in 2D image plane. In spatial servoing tasks, one can add a distinguishable mark to the object of interest to create useful features in the object image. The problem of using points in visual servoing is the precision, the detection of the points in the image is not robust against noise. Morover, adding several marks on the object of interest is not always possible.

Straight lines are used for tracking ([HT98], [ZF90]), for visual servoing [ECR92], [HCM95], [Hag97], or for position estimation ([LHF90],[DAO94b]) and their reconstruction has been well studied (see e.g. [Can86] for image detection, [SZ97] for matching and [SA90], [TK95], [Zha94] for structure and motion).
There are Two difficulties for using 3D straight line in the visual servoing. Firstly, there is no parameterization for straight lines representing their 4 DOF by 4 global parameters. Secondly, there is no universally agreed distance between straight lines. Two principals advantages for using lines in visual servoing, the first one is the robustness against the noise. The second advantage is the possibility to represent points (in couples) by straight lines, like what one can see in the work of Solá *et al.* in [SVCD09]. With recent real-time algorithms for extracting and tracking straight lines ([DC00], [TN00]), the lines are more used as visual features, in visual servoing, and real-time applications.

The natural existence of the 3D straight lines in the robot environment and the high precision of their detection in the image make them salient feature in visual servoing. The other advantage of the straight lines is the possibility to equally represent some other visual features like points pairs and planes pairs.
In mobile robotic, lines are useful because of their natural existence in the motion field, Li *et al.* [LXXX03] used white lines on the ground to detect the position and the orientation of a mobile robot.

The estimation of 3D motion and structure from sequences of images is a difficult problem, because of its instability. Many algorithms have been proposed to solve the problem of motion recovery. Some of these algorithms use straight line correspondences.
Yen and Huang [LH88] iteratively solve a set of nonlinear equations for the motion parameters. Navab and Faugeras in [NF97], are concerned with the problem of recovering the relative displacements of a camera by using line matches in three views. They have provided a clear description of the geometric and algebraic connections between the basic equations that constrain the camera displacements and the critical sets of lines.

Bartoli and Sturm [BS01] proposed several methods for estimating 3D motion from line correspondence, based on Plücker coordinates. The Plücker matrix representation is quadratic, and that provides some tricky computations in the transformation, which therefore can not be estimated linearly from line matches.
To overcome this problem, they derive a motion representation to transfer the Plücker coordinates linearly between two bases. The transformation is represented by a 6×6 matrix called "3D line motion matrix". Using this representation, several estimators are derived for 3D motion from line reconstructions. The motion representation allows straight lines to be transferred and reprojected from the first reconstruction onto the images of the second one. Optimization criteria can therefore be expressed in image-related quantities.

Ly *et al.* [LDV10], use lines to estimate the translation of moving vision system, after recovering the camera rotation using vanishing points of parallel lines sets. Their algorithm can be applied in navigation of autonomous robots besides the conventional devices such as global position system (GPS) and international navigation system (INS). They use straight lines as features for several reasons: such features are typically more stable than points and are less likely to be produced by clutter or noise, especially in man-made environment. Compared to point features, lines are

less numerous but more informative, they have geometrical and topological characteristics which are useful for matching.

A sequential algorithm for the straight lines recovery in projective space is proposed and implemented by Seo and Hong in [SH96]. They represent the 3D straight lines by Plücker coordinates. A constraint among three views are developed and used for refining the camera matrices through a nonlinear minimization method.

Hage Abdelkader *et al.* [AMAM06] presented a new approach for homography-based visual servoing using 3D straight lines. A partial Euclidean reconstruction is obtained from a generic homography matrix linearly estimated from the geometrical relationship between polar lines and conic curves. The polar lines and the information extracted from the homography are then used to design a control law which allows to fully decouple rotational motions from translational motions.

Wang *et al.* [WLW08] proposed a new controller for dynamic image based visual servoing of robot manipulator using lines. When the intrinsic and extrinsic parameters of the camera are not calibrated, they proposed a new method similar to the Plücker coordinates to represent projections of the lines features. They prove the asymptotic convergence of the image error using the Lyapunov theory.

2.3.1 Geometric modeling of a 3D straight line

Because of its 4 DOF, there is no simple representation found for the 3D straight line. Notice that a 3D point has 3 DOF, and represented by its Cartesian coordinate (x, y, z) or homogeneous coordinates $(x, y, z, 1)$. A plane Π has 3 DOF too, and can be represented by a four components vector $(a, b, c, d)^\mathsf{T}$ defined up to a scale.
The most known representations of a 3D straight line [AEH02] are:

1. A 3D straight line is defined by a point and a direction (or two points): In this case, the 3D straight line L is represented by a point, $P0 = (x0, y0, z0)^T \in L$, and a direction $u = (a, b, c)^T$.
   Every point from the straight line $P \in L$ is given by $P = P0 + \alpha u \ \ (\alpha \in \mathbb{R})$.
   In the case of two points $P0, P1$, the 3d straight line is defined by $P0$ and $u = P0 - P1$.
   This representation requires to find a distinguished point (or two) on the 3D straight line. Indeed, this representation uses 5 parameters (the vector u is defined up to a scale), to define a line which has 4 DOF, see figure (2.1 -a).

2. A 3D straight line is the intersection of two planes: This representation defines

a 3d straight line as a set of points satisfying two plane equations:

$$\Pi_1 : \quad a_1x + b_1y + c_1z + d_1 = 0$$
$$\Pi_2 : \quad a_2x + b_2y + c_2z + d_2 = 0$$

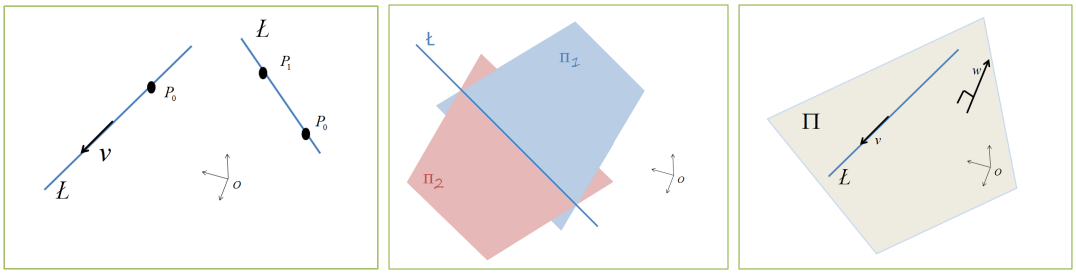
The plane can be represented by a vector $\Pi = (u, d)^T$ with u is a normalized vector ($||u|| = 1$) normal to the plane Π and d is the orthogonal distance between the origin of the frame and the plane. This representation uses 6 parameters to define the 3D straight line, see figure (2.1 -b).

3. The $(a; b; p; q)$ representation: This representation is a minimal variant of the others previous:

$$x = az + p$$
$$y = bz + q$$

It represents a 3D straight line that is neither parallel to the optical axis, nor parallel to the image plane. To take into account all the 3D straight lines in the 3D space, one has to make permutations over x, y and z.

4. The Plücker representation: Given a 3D straight line L, its Plücker coordinates are two algebraically dependent vectors (v, w) such that $v^{\mathsf{T}}w = 0$ [Plu65]. v is the line direction vector and w is a vector with a direction perpendicular to the interpretation plane (Π) contains the straight line and the origin of the frame, figure (2.1 -c).
Adding a constraint like $||v|| = 1$ to this representation, with the relation $v^{\mathsf{T}}w = 0$, make this representation minimal, as we get two vectors with two constraints, reduce the representation parameters to 4 free parameters.



(a) (b) (c)

Figure 2.1: Line representations. a) Two Points or one point with a direction vector. b) Two planes. c) Plücker coordinates.

2.3.2 Plücker coordinates and other representations

One of the most important property of Plücker coordinates is the possibility to find these coordinates from other representations.
Given two 3D points $\tilde{M}$ and $\tilde{N}$ on a 3D line L represented by their homogeneous coordinates

$$\tilde{M} = \begin{pmatrix} x_m \\ y_m \\ z_m \\ 1 \end{pmatrix} \qquad \tilde{N} = \begin{pmatrix} x_n \\ y_n \\ z_n \\ 1 \end{pmatrix}$$

a (4×4) matrix $\mathtt{L}$, named the Plücker matrix, is built as

$$\mathtt{L} \equiv \tilde{M}\tilde{N}^{\mathsf{T}} \; - \; \tilde{N}\tilde{M}^{\mathsf{T}} \equiv \begin{bmatrix} [w]_\times & v \\ -v^{\mathsf{T}} & 0 \end{bmatrix}$$

This matrix is independent of the position of the two points on the line. With $[w]_\times$ is the anti-symmetric matrix generated by vector w, represented as

$$[w]_\times = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & w_x \\ -w_y & w_x & 0 \end{bmatrix}$$

In the same way, if the straight line is represented by a point and a vector, $(P0, u)$, the Plücker coordinates (v, w) are expressed by $v = u$ and $w = P0 \times u$.

Now, given a 3D straight line L defining by the intersection of two planes Π_1 and Π_2, the dual of the Plücker matrix, $\mathtt{L}^\star$, also represents the 3D line, and is expressed by

$$\mathtt{L}^\star \equiv \Pi_1 \; {\Pi_2}^{\mathsf{T}} \; - \; \Pi_2 \; {\Pi_1}^{\mathsf{T}} \equiv \begin{bmatrix} [v]_\times & w \\ -w^{\mathsf{T}} & 0 \end{bmatrix}$$

$\mathtt{L}$ and $\mathtt{L}^\star$ are related with a simple rule ($\mathtt{L}\;\mathtt{L}^\star = \mathtt{0}$). It is a suitable representation since one may easily deal with any geometrical transformations $\mathtt{H}$, as the transformed Plücker matrix $\mathtt{L}'$ is simply expressed by $\mathtt{L}' \equiv \mathtt{H}\;\mathtt{L}\;\mathtt{H}^{\mathsf{T}}$ and v', w', v, w are linearly linked.
The two matrices are anti-symmetric, homogeneous and singular ($rank(\mathtt{L}) = 2$ and $rank(\mathtt{L}^\star) = 2$).
With the derivation of the characteristic polynomial of $\mathtt{L}$, one can easily show that the eigenvalues are $\{\pm i\ \mu,\ 0,\ 0\}$ with $\mu = \sqrt{v^{\mathsf{T}}v \; + \; w^{\mathsf{T}}w}$ and μ can be arbitrarily set to any non-zero value in order to normalize $\mathtt{L}$.
The relation between the representation of the same 3D straight line in two frames

is simple due to Plücker matrix $\mathtt{L}$, it is given by the matrix equation $\mathtt{L}' \equiv \mathtt{H}\ \mathtt{L}\ \mathtt{H}^{\mathsf{T}}$, $\mathtt{L}'$ is the Plücker matrix in the new frame.
Finally, by selecting the Plücker coordinates to represent the 3D straight line, we get two advantages. The first advantage is the total independent of the points, which keeps the high precession of this feature in the image. The second is the easy transformation between frames.

2.3.3 Normalized and bi-normalized Plücker coordinates

The vectors (v, w) are defined up to a scale factor (same for the two vectors). In normalized Plücker coordinates the vector v is normalized by $(||v|| = 1)$, which makes the norm of the vector w equal to the orthogonal distance between the frame origin and the line $(||w|| = d)$.
In bi-normalized Plücker coordinates [AEH02], the two vectors are normalized $(||v|| = 1, ||w|| = 1)$, and the distance d is an independent parameter now, this means the 3D straight line can be also represented by (v, w, d).
To illustrate the geometrical meaning of every parameter in the bi-normalized Plücker coordinates, figure (2.2 -a) shows an ensemble of lines with the same direction v. One can see that if only the direction of the line is known, an infinite number of parallel straight lines can be represented.
The vector w represent the ensemble of coplanar straight lines in the plane defined by w, figure (2.2 -b) shows ensemble of straight lines represented by w.
Finally, if the orthogonal distance d between the straight line and the frame origin is the only known parameter, the ensemble of straight lines represented by this distance are the lines tangent of a ball with radius d, see figure (2.2 -c).

Figure (2.3) shows a 3D straight line L, define in Cartesian frame (origin O), in this figure, the line and the origin O define the plane Π, the direction vector v ported on the line L, the vector w orthogonal on the plane Π, and the distance d between O and L.

2.3.4 Perspective projection of 3D straight line

The image l of a 3D straight line L in a perspective camera (figure 2.4) is defined (see [HZ00] for example), by the equation

$$[l]_{\times} \equiv \mathtt{P}_c\ \mathtt{L}\ \mathtt{P}_c^{\mathsf{T}}$$

with $\mathtt{L}$: is the Plücker matrix and $\mathtt{P}_c$: is the (3×4) camera matrix, defined by $\mathtt{P}_c = \mathtt{K}\ \mathtt{P}\ \mathtt{T}_{co}$

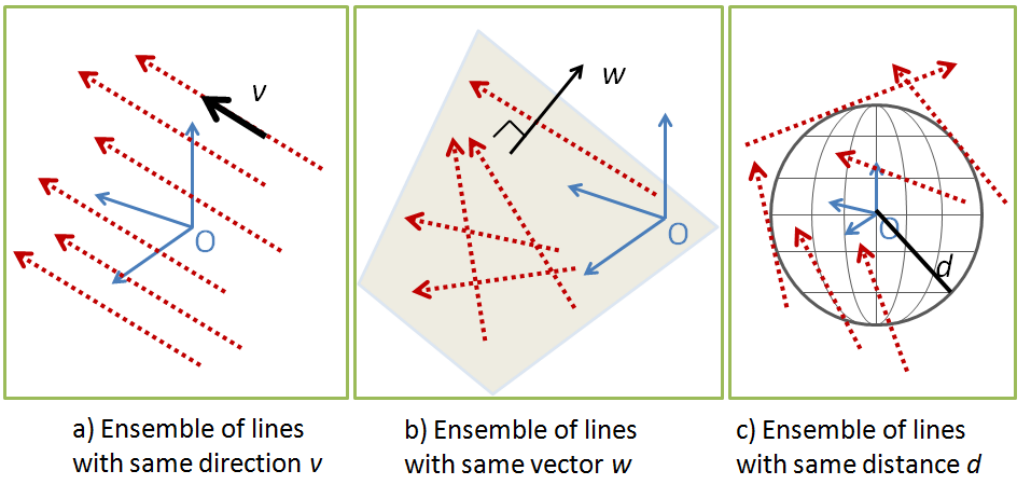


Figure 2.2: a) v is known. b) w is known. c) d is known.

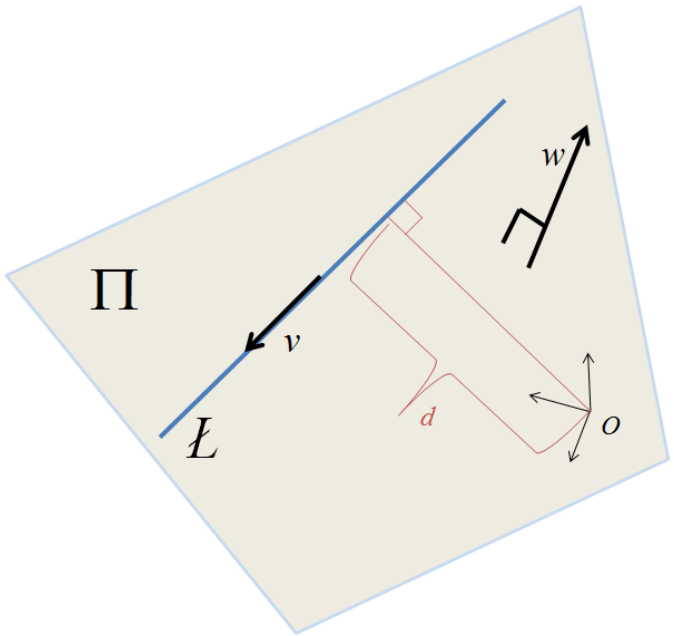


Figure 2.3: The Plücker coordinates (v, w) of a 3D straight line L.

$$\mathtt{L} \equiv \begin{bmatrix} [w]_\times & v \\ -v^\mathsf{T} & 0 \end{bmatrix} \qquad \mathtt{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad \mathtt{T}_{co} = \begin{bmatrix} \mathtt{R} & t \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$$

K: holds the intrinsic parameters of the camera.

$\mathtt{T}_{co}$: is the homogeneous transformation matrix from the line frame to the camera frame.

$\mathtt{R}, t$: are respectively, the rotation matrix, and the translation vector, between the line frame and the camera frame origins.

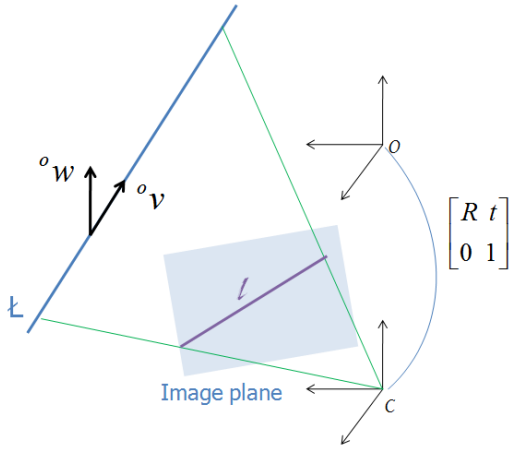


Figure 2.4: The image l of a 3D straight line L, and the transformation matrix $\mathtt{T}_{co}$.

The stereo system consists of two cameras called, right and left cameras. For the right camera, the index (r) will be used, and the index (l) used for the left camera. Figure (2.5) shows the right and left images of a 3D straight line.

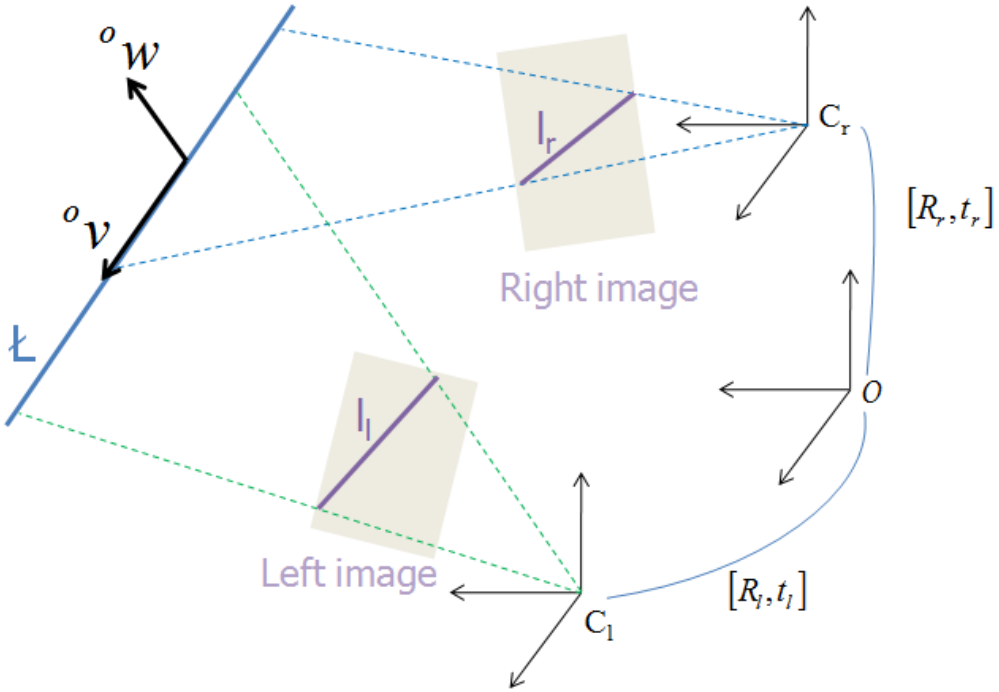


Figure 2.5: Stereo images (l_r, l_l), of a 3D straight line L.

In this figure, three frames are defined, one for the 3D straight line L (the object),

and two for the cameras. $\mathtt{R}_r, t_r$ are the rotation and translation from the right camera to the object, $\mathtt{R}_l, t_l$ are the rotation and translation from for the left camera.
The image equations are given by:

$$\begin{aligned} [l_r]_\times &\equiv \mathtt{P}_r\ \mathtt{L}\ \mathtt{P}_r^\mathsf{T} \\ [l_l]_\times &\equiv \mathtt{P}_l\ \mathtt{L}\ \mathtt{P}_l^\mathsf{T} \end{aligned}$$

By developing these equations, we obtain the relation between line image and Plücker coordinates:

$$\begin{aligned} [l_r]_\times &\equiv \mathtt{P}_r\ \mathtt{L}\ \mathtt{P}_r^\mathsf{T} \equiv \mathtt{K}_r\ \mathtt{P}\ \mathtt{T}_r\ \mathtt{L}\ \mathtt{T}_r^\mathsf{T}\ \mathtt{P}^\mathsf{T}\ \mathtt{K}_r^\mathsf{T} \\ [l_r]_\times &\equiv [\mathtt{K}_r\ \mathtt{R}_r \mid \mathtt{K}_r\ t_r] \begin{bmatrix} [w]_\times & v \\ -v^\mathsf{T} & 0 \end{bmatrix} \begin{bmatrix} \mathtt{R}_r^\mathsf{T}\ \mathtt{K}_r^\mathsf{T} \\ t_r^\mathsf{T}\ \mathtt{K}_r^\mathsf{T} \end{bmatrix} \\ [l_r]_\times &\equiv \left[\mathtt{K}_r\ \mathtt{R}_r\ [w]_\times - \mathtt{K}_r\ t_r\ v^\mathsf{T} | \mathtt{K}_r\ \mathtt{R}_r\ v\right] \begin{bmatrix} \mathtt{R}_r^\mathsf{T}\ \mathtt{K}_r^\mathsf{T} \\ t_r^\mathsf{T}\ \mathtt{K}_r^\mathsf{T} \end{bmatrix} \\ [l_r]_\times &\equiv \mathtt{K}_r\ \mathtt{R}_r\ [w]_\times\ \mathtt{R}_r^\mathsf{T}\ \mathtt{K}_r^\mathsf{T} - \mathtt{K}_r\ t_r\ v^\mathsf{T}\ \mathtt{R}_r^\mathsf{T}\ \mathtt{K}_r^\mathsf{T} + \mathtt{K}_r\ \mathtt{R}_r\ v\ t_r^\mathsf{T}\ \mathtt{K}_r^\mathsf{T} \end{aligned}$$

Since the matrix $\mathtt{K}$ is invertible, we get

$$\begin{aligned} \mathtt{K}_r^{-1}\ [l_r]_\times\ \mathtt{K}_r^{-\mathsf{T}} &\equiv \mathtt{R}_r\ [w]_\times\ \mathtt{R}_r^\mathsf{T} + t_r\ v^\mathsf{T}\ \mathtt{R}_r^\mathsf{T} - \mathtt{R}_r\ v\ t_r^\mathsf{T} \\ \frac{1}{\det(\mathtt{K}_r)}[\mathtt{K}_r^\mathsf{T}\ l_r]_\times &\equiv [\mathtt{R}_r\ w]_\times - t_r\ (\mathtt{R}_r\ v)^\mathsf{T} + (\mathtt{R}_r\ v)\ t_r^\mathsf{T} \end{aligned}$$

using $ba^\mathsf{T} - ab^\mathsf{T} = [a \times b]_\times$, we have after omit $\det(\mathtt{K}_r)$ as we have an equivalence relation

$$\begin{aligned} [\mathtt{K}_r^\mathsf{T}\ l_r]_\times &\equiv [\mathtt{R}_r\ w]_\times + [t_r\ \times\ \mathtt{R}_r\ v]_\times \\ \mathtt{K}_r^\mathsf{T}\ l_r &\equiv \mathtt{R}_r\ w + t_r \times \mathtt{R}_r\ v \end{aligned}$$

finally

$$l_r \equiv \mathtt{K}_r^{-\mathsf{T}}\ [\mathtt{R}_r\ w + t_r \times \mathtt{R}_r\ v] \tag{2.1}$$

same for the left camera

$$l_l \equiv \mathtt{K}_l^{-\mathsf{T}}\ [\mathtt{R}_l\ w + t_l \times \mathtt{R}_l\ v] \tag{2.2}$$

The equations (2.1, 2.2) show the relationship between the Plücker coordinates (v, w) of a 3D straight line, and its stereo $2D$ image lines, through the stereo rig intrinsic and extrinsic parameters.
Hager in [Hag97] represent the 3d straight line by a fixed point P and a direction vector v, with this representation, the stereo images of a 3d line are given by

$$\begin{aligned} l_r &\equiv \mathtt{R}_r \ (v \times (P - C_r)) \\ l_l &\equiv \mathtt{R}_l \ (v \times (P - C_l)) \end{aligned}$$

which are the same equations (2.1, 2.2) as $w = P \times v$.

2.3.5 3D estimation of the Plücker coordinates from stereo

With a pair of images, the 3D estimation of the Plücker coordinates of the 3D straight line is possible. The equations (2.1, 2.2) define the line images as a function of the Plücker coordinates (v, w). These equations are defined up to scale. Trying to inverse these equations to retrieve (v, w) as functions of (l_r, l_l) will define (v, w) with two different factors.
So it is necessary to use the geometry of the stereo images of a 3D straight line, in order to estimate its Plücker coordinates.

Figure (2.6) shows two planes (Π_r, Π_l) generated by the two cameras centers and the 3D line. The intersection of these two planes defines the 3D straight line, so according to [HZ00] the planes equations are

$$\begin{aligned} \Pi_r &\equiv \mathtt{P}_r^{\mathsf{T}} \ l_r \\ \Pi_l &\equiv \mathtt{P}_l^{\mathsf{T}} \ l_l \end{aligned}$$

and $\mathtt{P}_r, \mathtt{P}_l$ are given by

$$\begin{aligned} \mathtt{P}_r &\equiv \mathtt{K}_r \ \mathtt{P} \ \mathtt{T}_r \equiv [\mathtt{K}_r \ \mathtt{R}_r \mid \mathtt{K}_r \ t_r] \\ \mathtt{P}_l &\equiv \mathtt{K}_l \ \mathtt{P} \ \mathtt{T}_l \equiv [\mathtt{K}_l \ \mathtt{R}_l \mid \mathtt{K}_l \ t_l] \end{aligned}$$

Now, recall the definition of the dual Plücker matrix $\mathtt{L}^\star$

$$\mathtt{L}^\star \equiv \begin{bmatrix} [v]_\times & w \\ -w^{\mathsf{T}} & 0 \end{bmatrix} \equiv \Pi_r \ \Pi_l{}^{\mathsf{T}} \ - \ \Pi_l \ \Pi_r{}^{\mathsf{T}}$$

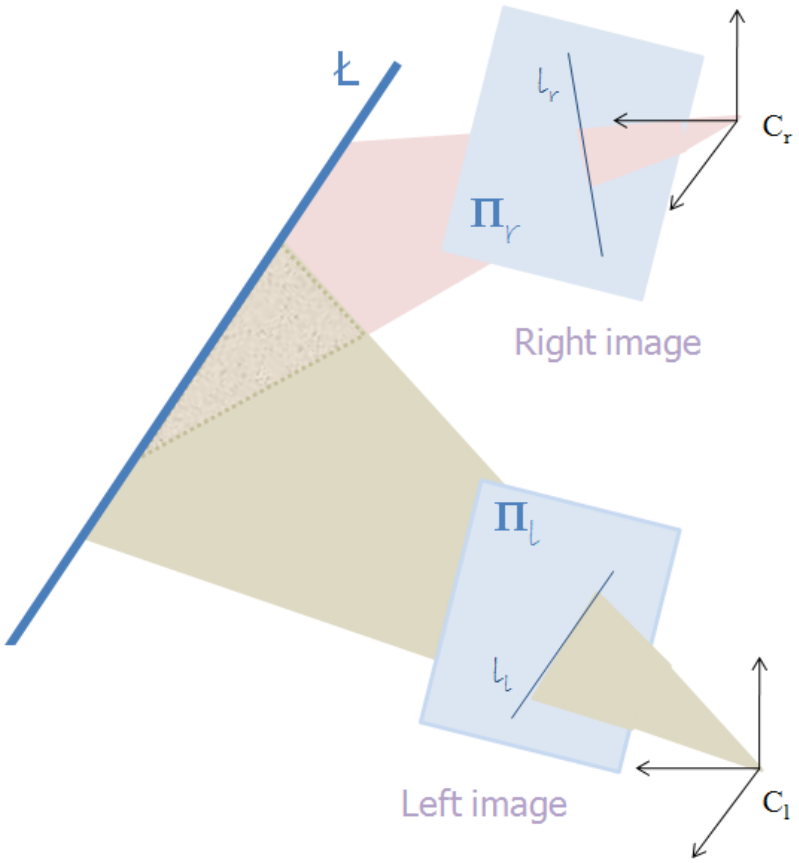


Figure 2.6: Intersection of two planes generated by the cameras centers and the 3D straight line.

Suppose $\mathtt{A} = \Pi_r \ \Pi_l{}^{\mathsf{T}}$, so by developing this equation we get

$$
\begin{aligned}
\mathtt{A} &= [\mathtt{K}_r \ \mathtt{R}_r \mid \mathtt{K}_r \ t_r]^{\mathsf{T}} \ l_r \ l_l^{\mathsf{T}} \ [\mathtt{K}_l \ \mathtt{R}_l \mid \mathtt{K}_l \ t_l] \\
&= \begin{bmatrix} \mathtt{R}_r^{\mathsf{T}} \ \mathtt{K}_r^{\mathsf{T}} \\ t_r^{\mathsf{T}} \ \mathtt{K}_r^{\mathsf{T}} \end{bmatrix} \ l_r \ l_l^{\mathsf{T}} \ [\mathtt{K}_l \ \mathtt{R}_l \mid \mathtt{K}_l \ t_l] \\
&= \left[\begin{array}{c|c} \mathtt{R}_r^{\mathsf{T}} \ \mathtt{K}_r^{\mathsf{T}} \ l_r \ l_l^{\mathsf{T}} \ \mathtt{K}_l \ \mathtt{R}_l & \mathtt{R}_r^{\mathsf{T}} \ \mathtt{K}_r^{\mathsf{T}} \ l_r \ l_l^{\mathsf{T}} \ \mathtt{K}_l \ t_l \\ \hline t_r^{\mathsf{T}} \ \mathtt{K}_r^{\mathsf{T}} \ l_r \ l_l^{\mathsf{T}} \ \mathtt{K}_l \ \mathtt{R}_l & t_r^{\mathsf{T}} \ \mathtt{K}_r^{\mathsf{T}} \ l_r \ l_l^{\mathsf{T}} \ \mathtt{K}_l \ t_l \end{array}\right] \\
&= \left[\begin{array}{c|c} \mathtt{A}_{11} & \mathtt{A}_{12} \\ \hline \mathtt{A}_{21} & \mathtt{A}_{22} \end{array}\right]
\end{aligned}
$$

which means

$$
\begin{bmatrix} [v]_\times & w \\ -w^{\mathsf{T}} & 0 \end{bmatrix} \equiv \left[\begin{array}{c|c} \mathtt{A}_{11} - \mathtt{A}_{11}^{\mathsf{T}} & \mathtt{A}_{12} - \mathtt{A}_{21}^{\mathsf{T}} \\ \hline \mathtt{A}_{21} - \mathtt{A}_{12}^{\mathsf{T}} & \mathtt{A}_{22} - \mathtt{A}_{22}^{\mathsf{T}} \end{array}\right]
$$

So, the vector v is expressed as

$$\begin{aligned}
[v]_\times &\equiv \mathtt{A}_{11} - \mathtt{A}_{11}^\mathsf{T} \\
&\equiv \mathtt{R}_r^\mathsf{T}\ \mathtt{K}_r^\mathsf{T}\ l_r\ l_l^\mathsf{T}\ \mathtt{K}_l\ \mathtt{R}_l\ -\ \mathtt{R}_l^\mathsf{T}\ \mathtt{K}_t^\mathsf{T}\ l_l\ l_r^\mathsf{T}\ \mathtt{K}_r\ \mathtt{R}_r \\
&\equiv \mathtt{R}_r^\mathsf{T}\ \mathtt{K}_r^\mathsf{T}\ l_r\ (\mathtt{R}_l^\mathsf{T}\ \mathtt{K}_t^\mathsf{T}\ l)^\mathsf{T}\ -\ \mathtt{R}_l^\mathsf{T}\ \mathtt{K}_t^\mathsf{T}\ l_l\ (\mathtt{R}_r^\mathsf{T}\ \mathtt{K}_r^\mathsf{T}\ l_r)^\mathsf{T} \\
&\equiv [\mathtt{R}_l^\mathsf{T}\ \mathtt{K}_l^\mathsf{T}\ l_l \times \mathtt{R}_r^\mathsf{T}\ \mathtt{K}_r^\mathsf{T}\ l_r]_\times
\end{aligned}$$

and finally

$$v \equiv \mathtt{R}_l^\mathsf{T}\ \mathtt{K}_l^\mathsf{T}\ l_l \times \mathtt{R}_r^\mathsf{T}\ \mathtt{K}_r^\mathsf{T}\ l_r \tag{2.3}$$

For the vector w, we have

$$w \equiv \mathtt{A}_{12} - \mathtt{A}_{21}^\mathsf{T}$$

which gives

$$w \equiv \mathtt{R}_r^\mathsf{T}\ \mathtt{K}_r^\mathsf{T}\ l_r\ l_l^\mathsf{T}\ \mathtt{K}_l\ t_l\ -\ \mathtt{R}_l^\mathsf{T}\ \mathtt{K}_l^\mathsf{T}\ l_l\ l_r^\mathsf{T}\ \mathtt{K}_r\ t_r \tag{2.4}$$

Equations (2.3) and (2.4) provides a solution for the Plücker coordinates v, w from the stereo images l_r, l_l. They are defined up to a common scale.
The equation (2.3) is found in [Hag97] with the same form.

2.3.6 Singularity case

Recall the formula of the direction vector (2.3),

$$v \equiv \mathtt{R}_l^\mathsf{T}\ \mathtt{K}_l^\mathsf{T}\ l_l \times \mathtt{R}_r^\mathsf{T}\ \mathtt{K}_r^\mathsf{T}\ l_r\ .$$

This equation represents in the general case a vector, but if the two terms of the cross product $\mathtt{R}_r^\mathsf{T}\ \mathtt{K}_r^\mathsf{T}\ l_r$ and $\mathtt{R}_l^\mathsf{T}\ \mathtt{K}_l^\mathsf{T}\ l_l$ are equivalent, this equation is equal to zero, and the direction of the 3D line is unknown.
This case means that the two orthogonal vectors of the two planes (Π_r, Π_l) in figure (2.6) are parallel, which means the two planes are parallel too (identical because they share a common line).
These two planes defined by the line and the two cameras centers, if they are identical, means the two cameras centers are coplanar with the 3D line L. This is a singularity for the 3D reconstruction of the line from its stereo images.
For the second vector in Plücker coordinates

$$w \equiv \mathtt{R}_r^\mathsf{T}\ \mathtt{K}_r^\mathsf{T}\ l_r\ l_l^\mathsf{T}\ \mathtt{K}_l\ t_l\ -\ \mathtt{R}_l^\mathsf{T}\ \mathtt{K}_l^\mathsf{T}\ l_l\ l_r^\mathsf{T}\ \mathtt{K}_r\ t_r$$

it can be written as $w \equiv d_1 * \mathtt{R}_l^\mathsf{T}\ \mathtt{K}_l^\mathsf{T}\ l_l\ -\ d_2 * \mathtt{R}_r^\mathsf{T}\ \mathtt{K}_r^\mathsf{T}\ l_r$ with $d_1 = l_r^\mathsf{T}\ \mathtt{K}_r\ t_r$, $d_2 = l_l^\mathsf{T}\ \mathtt{K}_l\ t_l$. This vector is unknown, if the 3D line is coplanar with the two cameras centers, because $d_1 = d_2 = 0$. It is unknown if the 3D line goes through the reference frame origin, but this case is included in the first case.
To resume, the singularity of recovering a 3D straight line from its stereo images will be found if the 3D straight line is coplanar with the two camera centers.

2.3.7 Kinematic modeling

The motion analysis of straight lines has been studied by many authors. For instance Mitiche [Mit94] provides relationships between the apparent motion and the 3D camera displacement with straight lines and single camera. Rives in [RE87] and Navab in [NFV93] described the time derivation of a 3D straight line represented by Plücker coordinates in relation with the kinematic screw $\tau = (\mathbf{\Omega}, \mathbf{V})$ by the equations:

$$\begin{aligned} \dot{v} &= \mathbf{\Omega} \times v + \alpha\, v \\ \dot{w} &= \mathbf{\Omega} \times w + \mathbf{V} \times v + \alpha\, w \end{aligned}$$

The parameter α can be determined if a constraint was added like $||v|| = 1$ (so $\dot{v}^{\mathsf{T}} v = 0$) in normalized Plücker coordinates. With this constraint we obtain $\alpha = 0$.

$$\dot{v} = \mathbf{\Omega} \times v \tag{2.5}$$

$$\dot{w} = \mathbf{\Omega} \times w + \mathbf{V} \times v \tag{2.6}$$

These equations show that the direction of the 3D line is independent to the translation velocity $\mathbf{V}$, that helps in constructing a decoupled control law for rotation or fully cascaded control laws for rotation and translation velocities.
With the bi-normalized Plücker coordinates $((v, w, d), ||v|| = 1, ||w|| = 1)$, Andreff in [AEH02] isolates the variation of the direction of the planar vector (w) from the variation of its norm (the distance d), so the kinematic of the 3D straight line represented by bi-normalized Plücker coordinates is given by [AEH02]

$$\begin{pmatrix} \dot{w} \\ \dot{v} \\ \dot{d} \end{pmatrix} = \begin{pmatrix} -\frac{1}{d}(v \times w)\; w^{\mathsf{T}} & -[w]_{\times} \\ \mathbf{0}_{3\times 3} & -[v]_{\times} \\ (v \times w)^{\mathsf{T}} & \mathbf{0}^{\mathsf{T}} \end{pmatrix} \boldsymbol{\tau} \tag{2.7}$$

where $\boldsymbol{\tau}^{\mathsf{T}} = (\mathbf{V}^{\mathsf{T}}, \mathbf{\Omega}^{\mathsf{T}})$.

2.4 Pose estimation

The estimation of the pose (position and orientation) and the motion is considered as important application of the artificial vision. The image information is used for "positioning" an object with respect to the camera frame.
The recovery of the 3D geometric information from 2D images is a fundamental problem in computer vision. When only one view is available, the appearance or the relative arrangement of the object features of interest should be modelled in a symbolic description so as to be compared with the image descriptors thanks to a similarity criterion [Doi07]. Geometric-based approaches restrict the search for

correspondence to a sparse set of geometrical features. They use numerical and symbolic properties of entities available. To automatically compute a rigid-body transformation (the pose), it is necessary to match a 3D model features with part of the visible 2D image features, a process referred to as the correspondence problem, and for the past four decades, the model-based pose estimation of objects with a simple geometry has been intensively studied. In the following, we report only few works about the pose estimations, as there is a vaste literature on this subject (see [Doi07] for a review).

Doignon and de Mathelin in [DdM07] have addressed the problem of estimating the pose of a straight homogeneous circular cylinder (SHCC) from a single perspective view of its surface outlines. The cylinder is modelled by a singular matrix composed with the Plücker coordinates of the symmetry axis and the cylinder radius. The apparent contour of the cylinder serves as input data. The perspective projection of a SHCC is related to the pose parameters. The authors have presented a degenerated conic-based fitting approach and they have provided both a closed-form solution and a numerical minimization.

In [DAAAM08] the authors present novel methods for high speed pose and velocity computation from visual sensor. Usually, camera video rate is limited by the transmission interface bandwidth. Reducing the image resolution to decrease the video flow tightens a lot the field of view of the camera for a given accuracy of the end-effector pose estimation. The proposed approach solves the problem of the video rate transmission by increasing the information density instead of the data rate transmission. Instead of transmitting the whole image and then selecting a region of interest (ROI), it is more interesting to inverse the process by first selecting the ROI position, and then to transmit it. The idea of the authors is to use the image artifacts for real-time high speed object pose and velocity computation based on non-linear least squares minimization.

To estimate the motion of the robot, several methods have been proposed using traditional navigation equipments such as Global Positioning System (GPS) or/and Inertial Navigation System (INS). However, these sensors suffer from many limitations. In order to overcome those limits, a vision based approach of the navigation problem has been proposed. The goal is to estimate location and/or orientation of the robot when GPS or inertial guidance is not available. Using conventional cameras, that have a relative small field of view, may lead to important difficulties. Therefore, Bazin *et al.* [BDVK10] estimate the motion by decoupling rotation and translation in catadioptic vision. The proposed system provides two strong advantages over the existing methods: first, it can implicitly handle the difficulty of

planar/non-planar scenes, and second, it is computationally much less expensive. The rotation is computed by an efficient algorithm based on the detection of dominant bundles of parallel catadioptric lines and the translation is calculated from a robust 2-point algorithm.

Determining the pose of a mobile robot from the image invariant is the method used by Charron *et al.* in [CLIM05] with omnidirectional images. Their approach is based on the recognition of panoramic images to recover the pose of a mobile robot. The goal is to make the robot recognizing places by the help of an omnidirectional camera. The idea is to determine a mapping to extract intrinsic features on the local environment of the robot, i.e. features that remain unchanged if the robot position and/or orientation locally changes in the scene. The approach consists in using integral invariant features computed on omnidirectional images and showing their interest in a context of mobile robots localization.
The integral method used to build the invariant has the advantage of being more straightforward than differential or geometrical methods. The integral method requires neither image segmentation as in geometrical methods nor derivative computation as in differential methods. Moreover, the invariant is also robust to rotations although they are not considered in the initial transformation group.
The main advantage of this method is that it allows to take into account the real movements of the robot to build the invariants.

In 1989, Dhome *et al.* in [DRLR89] define an analytical solution to the problem of the determination of the 3D object attitude in space from a single perspective image. Its principle is based on the interpretation of a triplet of any image lines as the perspective projection of triplet of linear ridges of the object model, and on the search of the model attitude consistent with these projections. The corresponding solutions are obtained by the resolution of an four-degree equation (a polynomial equation of degree eight with only even terms) and some simple but efficient rules are given to reduce the number of solutions.
With n known points in two frames, Umeyama in [Ume91] presented a method to estimate the rotation matrix between the two frames. This method based on a least square minimization.
Daniilidis [Dan96],[Dan99] presented a unified and fast way of formulating with dual-quaternions – algebraic counterparts of screws – for both the rotational and the translational relative between the desired and the current poses.
In ([WHB96],[HEK96]), features are detected in an image and used to estimate the current camera pose. A pose error is then computed in the Cartesian task space, and this error is used by the control system. With the technique of Lowe in [Low87],

an efficient and fast minimization algorithm provides the 6 pose parameters with a gradient descent. The camera pose was obtained by Rivera-Rios *et al.* in [RRSM05] via a nonlinear program that minimizes the total mean square error of the length measurements while satisfying the sensor constraints. They present a probabilistic analysis of the effect of the localization errors on the dimensional measurements of the line entities for a parallel stereo setup.
In PBVS, it is well known that the visibility of the object of interest is not guaranteed during the servoing time. A modified PBVS has been proposed in [TMCG02] which allows to keep the object frame origin in the field of view (FOV). However, in most cases it is not guaranteed either that all points remain in the FOV or the camera motion converges to the desired configuration from any feasible initial one.

2.5 Decoupling the control laws

Decoupling or non-interactive control has attracted considerable research attention since the 1960's when control engineers started to deal with multivariable systems. The theory and design techniques for decoupling control is now, matured for linear time-invariant systems.
In many complex industrial processes, the coupling among control loops often invalidates conventional single-loop controllers. Decoupling controls was initially developed for deterministic linear systems. Typical approaches included design of pre-compensator that transforms the controlled transfer function matrix into a diagonal matrix or diagonal dominance [Kav57], or the design of state feedback to reach decoupling of state equations [FW67].
The goal of decoupling control is to eliminate complicated loop interactions, so that a change in one process variable will not interact with changes in other process variables. This makes an advantage to the control loop, to be easily implemented and tuned. Note that in a system heavily dependent on the coordination of all loops, the failure of one component may lead to the failure of the overall structure.
In 1999, Andreff *et al.* ([And99],[AEH00],[AEH02]) used 3D straight lines to design a partial decoupled velocities control laws for a monocular PBVS. This work's application was concerning the improvement the automatic welding of ship parts, whose relative position is not exactly known in advance.
By representing the 3D straight line using bi-normalized Plücker coordinates (v, w, d), the authors obtain a naturally partial decoupling between the rotation velocity control and the translation velocity control from the kinematics of a 3D straight line. The rotational task function is defined using the direction vector $e_\Omega = [v_1 \times v_1^*, ..., v_n \times v_n^*]^\mathsf{T}$. This task function is independent to the translation motion. Then, the rotational velocity control law is given by $\mathbf{\Omega} = -\lambda_\Omega \sum_i v_i \times v_i^*$. After the end of the

rotational servoing, the direction vectors v_i are converging toward their desired values v_i^*. Then, the translation task function $e_V = v \times (w - w^*)$ is a pure translation error. The translational control law $\mathbf{V} = -\lambda_V \sum_i \frac{1}{d_i} (v_i \times w_i)^\mathsf{T} w_i^* w_i$ moves the system to the desired position.
In the practical application of welding the orthogonal trihedral, a commercial laser device is used as measurements for the depth estimation.

Abdelkader *et al.* [AMAM06] present a new approach for $2\frac{1}{2}$D homography-based visual servoing using 3D straight lines imaged with central catadioptric cameras. Their work concerned the design of a control law which allows to fully decoupled rotational motions from translational motions.
To this end, they designed two independent state vectors for rotational and translational motions. The rotational state vector is $s_\Omega = \theta u$, that is the angle/axis representation of the rotation matrix $\mathtt{R}$ between the current and desired frames, while the translational state vector is $s_V = [\log(d_1), \log(d_2), \log(d_3)]^\mathsf{T}$. In this vector, d_i is the orthogonal distance between the 3D straight line L_i and the frame origin.
The rotation matrix $\mathtt{R}$ and the ratios $\frac{d_i}{d_i^*}$ have been estimated from the correspondence between four couples of polar lines $(l_i, l_i^*, \quad i = 1..4)$. The task function e to regulate to 0 is then given by

$$e = \begin{pmatrix} s_V - s_V^* \\ s_\Omega - s_\Omega^* \end{pmatrix} = \begin{pmatrix} \log(\frac{d_1}{d_1^*}) \\ \log(\frac{d_2}{d_2^*}) \\ \log(\frac{d_3}{d_3^*}) \\ \theta u \end{pmatrix} .$$

This servoing is an hybrid-based one because the rotational state vector s_Ω is a 3D vector, while the translational state vector s_V is estimated from visual measurements, every depth ratio is estimated from the current and desired images, as it is with oriented projective space: $\frac{d}{d^*} = r' \frac{n^* \times \mathtt{K}^\mathsf{T} l_i^*}{\mathtt{R} n^* \times \mathtt{K}^\mathsf{T} l_i}$ (d^* is the normal vector of the interpretation plane π^*, $r' = 1 + t_d^{*\mathsf{T}} \mathtt{R}^\mathsf{T} n^*$, and t_d^* is the desired translation).

These two works concerned the monocular visual servoing. The first one was a partial decoupled PBVS, and the authors used a laser pointer for estimating the depth, while the object of interest consists of three orthogonal straight lines. The second was a full decoupled hybrid visual servoing. The authors need four 3D straight lines to estimate the matrix of rotation and the ratio of depth.
In our work, we will use the stereo vision for visual servoing. This allows to reduce the number of 3D straight lines needed to control the camera motion with respect to the object of interest, and to achieve the full decoupling between rotational and

translational velocities control laws in both PBVS and IBVS. There are other benefits from the stereo vision measurements, they will be pointed out through out the thesis report.

Chapter 3

Stereo position based visual servoing from lines

This section addresses the position based visual servoing (PBVS). The vision system used to close the control loop is a stereo rig. The object of interest is modeled by a set of 3D straight lines, represented by the Plücker coordinates.
Figure (3.1) shows a simple block diagram for the closed loop of (PBVS).

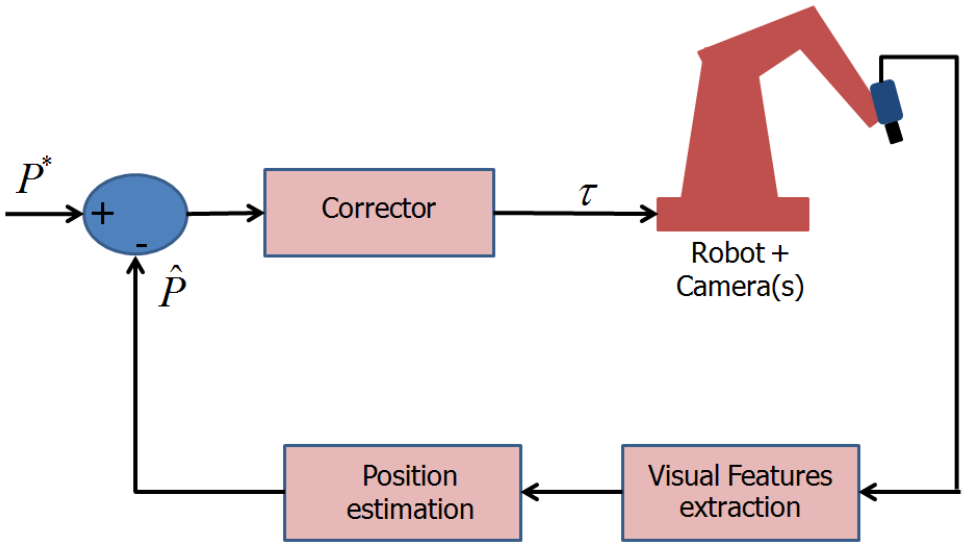


Figure 3.1: Position Based Visual Servoing (PBVS) block diagram.

The notation P^* denotes the desired pose, and $\hat{P}$ is the estimation of the current one. P^* and $\hat{P}$ are relative poses (position and orientation) between the vision system and the object in the vision field. P^* is assumed to be constant in most applications of PBVS while $\hat{P}$ is a time variant. The challenge of the vision system is to estimate $\hat{P}$ during the servoing time.

3.1 Position based visual servoing using $[\mathtt{R}, t]$

In this kind of servoing the state vectors are, the relative rotation and the relative translation, between the current stereo rig frame F_c and the desired stereo rig frame F_{c^*}.
Malis in [Mal98], has proposed the fully decoupling control laws for rotational and translational velocities by

$$\mathbf{V} = -\lambda_V \, \mathtt{R}^\top t \tag{3.1}$$

$$\mathbf{\Omega} = -\lambda_\Omega \, \theta u \tag{3.2}$$

with $\lambda_V, \lambda_\Omega \in \mathbb{R}^+$ are the gains of the closed loop. $\mathtt{R}$ is the relative matrix of rotation between F_c and F_{c^*}.
t is the relative translation vector between the origin c of F_c and the origin c^* of F_{c^*}.
θu gives the angle/axis parameterization for the rotation:

$$\theta = \arccos((tr(\mathtt{R}) - 1)/2)$$

$$u = \begin{pmatrix} \mathtt{R}(3,2) - \mathtt{R}(2,3) \\ \mathtt{R}(1,3) - \mathtt{R}(3,1) \\ \mathtt{R}(2,1) - \mathtt{R}(1,2) \end{pmatrix} /(2 \, \sin(\theta)).$$

$\mathbf{V}, \mathbf{\Omega}$ are the translational and rotational velocities defined in the stereo rig frame F_c.
The camera trajectory is a pure straight line in the Euclidean space, because the state vectors represent Cartesian coordinates, as the translational state vector is $s_V = \mathtt{R}^\top t$ and the rotational state vector is $s_\Omega = \theta u$, when the desired state vectors are null $s_V^* = 0, s_\Omega^* = 0$. So the task functions are:

$$e_V = \mathtt{R}^\top t$$

$$e_\Omega = \theta u$$

The global asymptotic stability GAS of these control laws are proved if $\theta \neq k\pi, k \in \mathbb{Z}$ in [Mal98]. The Euclidean relative pose $[\mathtt{R}, t]$ between the current pose and the desired pose, is estimated using Plücker coordinates of the 3D straight lines, as we will see in the next section.
The control laws used in this kind of servoing will be called " Euclidean control laws".

3.1.1 Pose estimation from lines

In the previous section, the " Euclidean control laws" are defined from the relative rotation and the relative translation $[\mathtt{R}, t]$.

Figure (3.2) shows an object that consists of 3D straight lines $L_1, ..., L_n$, represented by a set of Plücker coordinates $\left({}^{o}v_1, {}^{o}w_1), ..., ({}^{o}v_n, {}^{o}w_n)\right)$ at the object frame F_o which is the stereo rig frame F_{c^*} at the desired pose. This object is represented in the current stereo rig frame F_c, by $\left(({}^{c}v_1, {}^{c}w_1), ..., ({}^{c}v_n, {}^{c}w_n)\right)$.

The goal of the PBVS is to guide the stereo rig from its current pose F_c, to the

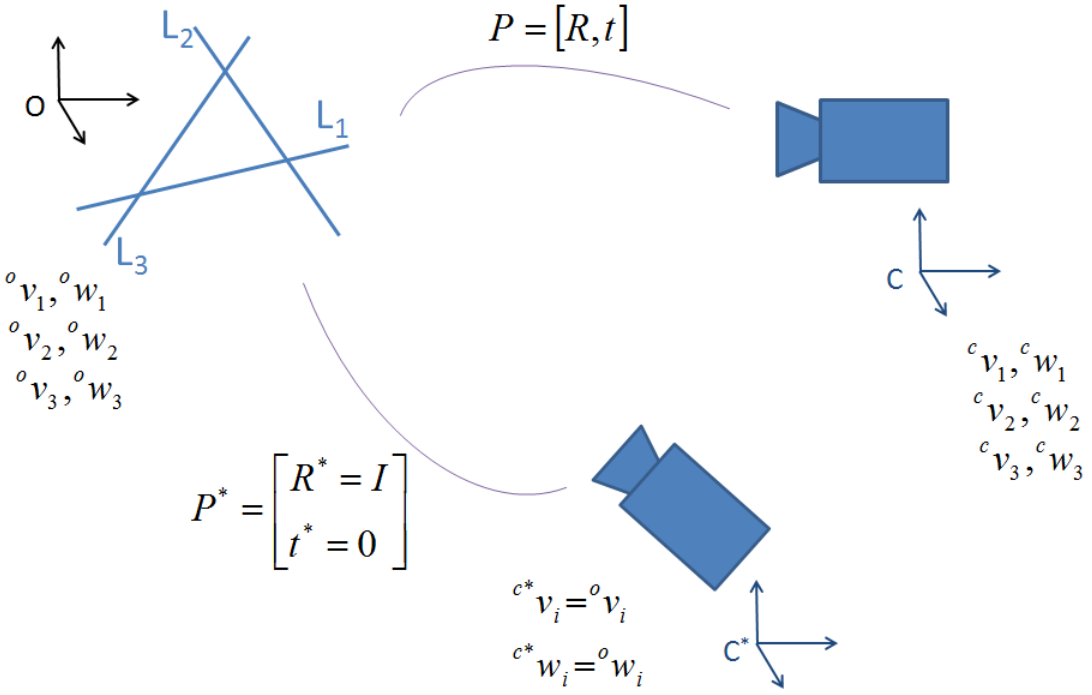


Figure 3.2: Initial and desired relative Object-Camera pose.

desired pose F_{c^*}. The relative current Object-Stereo rig pose is given by $P = [\mathtt{R}, t]$ and the desired one is $P^* = [\mathtt{R}^* = \mathtt{I}, t^* = 0]$. That means $F_{c^*} = F_o$.

We need to find the rigid transformation defined with the homogeneous matrix $\mathtt{T} = \begin{bmatrix} \mathtt{R} & t \\ \mathbf{0} & 1 \end{bmatrix}$ between the current and the desired stereo rig frames.

The relation between Plücker matrix ${}^{c}\mathtt{L}_j$ for the straight line L_j in the current stereo rig frame F_c, and the Plücker matrix ${}^{o}\mathtt{L}_j$ of the same straight line L_j in the object frame F_o is given by ${}^{o}\mathtt{L}_j \equiv \mathtt{T}\ {}^{c}\mathtt{L}_j\ \mathtt{T}^{\mathsf{T}}$ [HZ00].

To transfer the equivalent sign into equal sign, the Plücker matrices ${}^{c}\mathtt{L}_j, {}^{o}\mathtt{L}_j$ are normalized to have $||{}^{c}v_j|| = 1, ||{}^{o}v_j|| = 1$.

So the equation becomes ${}^{o}\mathtt{L}_j = \mathtt{T}\ {}^{c}\mathtt{L}_j\ \mathtt{T}^{\mathsf{T}}$. By developing this equation (we will remove

the stereo rig frame index ($^c.$) for simplifying the text)

$$\begin{bmatrix} [{}^o w_j]_\times & {}^o v_j \\ -{}^o v_j^\mathsf{T} & 0 \end{bmatrix} = \begin{bmatrix} \mathtt{R} & t \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} [w_j]_\times & v_j \\ -{v_j}^\mathsf{T} & 0 \end{bmatrix} \begin{bmatrix} \mathtt{R}^\mathsf{T} & \mathbf{0} \\ t^\mathsf{T} & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \mathtt{R}[w_j]_\times - t{v_j}^\mathsf{T} & \mathtt{R}v_j \\ -{v_j}^\mathsf{T} & 0 \end{bmatrix} \begin{bmatrix} \mathtt{R}^\mathsf{T} & \mathbf{0} \\ t^\mathsf{T} & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \mathtt{R}[w_j]_\times \mathtt{R}^\mathsf{T} - t({v_j}^\mathsf{T}\mathtt{R}^\mathsf{T}) + (\mathtt{R}v_j)t^\mathsf{T} & \mathtt{R}v_j \\ -{v_j}^\mathsf{T}\mathtt{R}^\mathsf{T} & 0 \end{bmatrix}$$

so,

$$^o v_j = \mathtt{R}v_j \tag{3.3}$$

and,

$$[{}^o w_j]_\times = \mathtt{R}[w_j]_\times \mathtt{R}^\mathsf{T} - t({v_j}^\mathsf{T}\mathtt{R}^\mathsf{T}) + (\mathtt{R}v_j)t^\mathsf{T}$$

$$[{}^o w_j]_\times = [\mathtt{R}w_j]_\times + [t \times \mathtt{R}v_j]_\times$$

$$^o w_j = \mathtt{R}w_j + t \times \mathtt{R}v_j \tag{3.4}$$

The equation (3.3) is showed in [LHF90], but the authors use the three Euler angles to represent the rotation matrix $\mathtt{R}$, for that they need three 2d to 3d line correspondences to find the angles.

We now recall the important lemma of Umeyama [Ume91], which gives a solution by the least squares for the estimation of the rotation matrix between two groups of points in a space of dimension m.

Suppose $\mathtt{A}$ and $\mathtt{B}$ are two $(m \times n)$ matrices, $\mathtt{R}$ is the rotational $(m \times m)$ matrix and $\mathtt{UDV}^\mathsf{T}$ is the decomposition in singular values of $\mathtt{AB}^\mathsf{T}$ ($\mathtt{UU}^\mathsf{T} = \mathtt{VV}^\mathsf{T} = \mathtt{I}$, $\mathtt{D} = \mathrm{diag}(d_i)$, $d_1 \geq d_2 \geq \cdots \geq d_m \geq 0$), so the minimum of $||\mathtt{A} - \mathtt{RB}||^2$ in function of $\mathtt{R}$ is expressed by

$$\min_{\mathtt{R}} ||\mathtt{A} - \mathtt{RB}||^2 = ||\mathtt{A}||^2 + ||\mathtt{B}||^2 - 2\ tr(\mathtt{DS})$$

with

$$\mathtt{S} = \begin{cases} \mathtt{I} & \text{if } \det(\mathtt{U})\ \det(\mathtt{V}) = 1, \\ \mathrm{diag}(1, \cdots, 1, -1) & \text{if } \det(\mathtt{U})\ \det(\mathtt{V}) = -1 \end{cases}$$

The minimum of the above criterion is given by $\mathtt{R} = \mathtt{USV}^\mathsf{T}$. Proof and details are in [Ume91].

Considering equation (3.3), the two matrices $\mathtt{A}$ and $\mathtt{B}$ can be expressed with the Plücker coordinates v, therefore with $\mathtt{A} = [{}^o v_1, \cdots, {}^o v_n]$, and $\mathtt{B} = [v_1, \cdots, v_n]$. Of course, this solution is not possible if there is only one line ($n = 1$), or if all the lines are parallel ($v_1 = v_2 = \cdots = v_n$).

After the estimation of the rotation matrix $\mathtt{R}$, we now look for the estimation of the translation vector t between the current and desired stereo rig frames. This can be done using equation (3.4):

$$\begin{aligned} {}^o w_j &= \mathtt{R} w_j + t \times \mathtt{R} v_j \\ {}^o w_j - \mathtt{R} w_j &= -{}^o v_j \times t \\ {}^o w_j - \mathtt{R} w_j &= -[{}^o v_j]_\times t \end{aligned}$$

Define $q_j = \mathtt{R} w_j - {}^o w_j$ as the matrix $\mathtt{R}$ is known

$$\mathtt{Q} = \begin{pmatrix} q_1 \\ \vdots \\ q_n \end{pmatrix} \quad \text{and} \quad \mathtt{H} = \begin{bmatrix} [{}^o v_1]_\times \\ \vdots \\ [{}^o v_n]_\times \end{bmatrix}$$

we get a linear system of n equations of the form

$$\mathtt{H} t = \mathtt{Q} \ .$$

A solution for this system (with t as unknown) is given by (with the least squares)

$$t = \mathtt{H}^+ \mathtt{Q} \ .$$

This solution is defined if and only if $\mathtt{H}^+$ exists, for that, and as $\mathtt{H}^+ = (\mathtt{H}^\mathsf{T}\mathtt{H})^{-1}\mathtt{H}^\mathsf{T}$ then $\mathtt{H}^\mathsf{T}\mathtt{H}$ must be invertible.

$$\begin{aligned} \mathtt{H}^\mathsf{T}\mathtt{H} &= \begin{bmatrix} [{}^o v_1]_\times \\ \vdots \\ [{}^o v_n]_\times \end{bmatrix}^\mathsf{T} \begin{bmatrix} [{}^o v_1]_\times \\ \vdots \\ [{}^o v_n]_\times \end{bmatrix} \\ &= -[[{}^o v_1]_\times, \ldots, [{}^o v_n]_\times] \begin{bmatrix} [{}^o v_1]_\times \\ \vdots \\ [{}^o v_n]_\times \end{bmatrix} \\ &= -[{}^o v_1]_\times^2 - \ldots - [{}^o v_n]_\times^2 \end{aligned}$$

It is evident that $\mathtt{H}^\mathsf{T}\mathtt{H}$ is not invertible in the case of one line ($n = 1$), since $[{}^o v]_\times^2$ is of rank 2. In the case of two lines ($n = 2$), we get

$$det(\mathtt{H}^\mathsf{T}\mathtt{H}) = 2 \, || {}^o v_1 \times {}^o v_2 ||^2$$

so, if the two lines are not parallel, the matrix $\mathtt{H}^\mathsf{T}\mathtt{H}$ is invertible.
In the general case (where $n > 1$), the matrix $\mathtt{H}^\mathsf{T}\mathtt{H}$ is invertible if and only if there are at least two non parallel lines in the object.

To resume, the relative rotation matrix $\mathtt{R}$ between the current and final stereo rig frames is given by $\mathtt{R} = \mathtt{U}\mathtt{S}\mathtt{V}^\mathsf{T}$, and the relative translation vector t between the frames origins is given by $t = \mathtt{H}^{+}\mathtt{Q}$.

In the next section, what is the maximum rotation and translation between the initial and final frame, that can be estimated using the Plücker coordinates, and stereo vision.

3.1.2 Domain of the state vectors

In the previous section, the rotation matrix $\mathtt{R}$ and the translation vector t had been estimated from the knowledge of the Plücker coordinates of the straight lines in the initial and final pose.
The estimation of the rotation matrix is defined by the equation (3.3) ${}^o v = \mathtt{R} v$, with $||{}^o v|| = ||v|| = 1$.
Now, recall the equations (2.3, 2.4), which define the Plücker coordinates v, w from a pair of stereo image.

$$\begin{aligned} v &\equiv \mathtt{R}_l^\mathsf{T}\, \mathtt{K}_l^\mathsf{T}\, l_l \times \mathtt{R}_r^\mathsf{T}\, \mathtt{K}_r^\mathsf{T}\, l_r \\ w &\equiv \mathtt{R}_r^\mathsf{T}\, \mathtt{K}_r^\mathsf{T}\, l_r\, l_l^\mathsf{T}\, \mathtt{K}_l\, t_l \;-\; \mathtt{R}_l^\mathsf{T}\, \mathtt{K}_l^\mathsf{T}\, l_l\, l_r^\mathsf{T}\, \mathtt{K}_r\, t_r \end{aligned}$$

In these equations, (v, w) are defined up to common scale. The constraint $||v|| = 1$ place the scale, but not the sign, which means (v, w) and $(-v, -w)$ are corresponding to the same line and the same image lines l_r, l_l. So the equation (3.3) ${}^o v = \mathtt{R} v$, is not valid until the sign of v is well defined.
Given an object represented by n 3D straight lines, defined in the final pose by ${}^o v_j, {}^o w_j, j = 1..n$. Then, by adding the constraint ${}^o v_j{}^\mathsf{T} v_j > 0$, $j = 1..n$, one can define the sign of the direction vector v_j. But this constraint reduce the domain of rotation angle of each line to be less than $\pi/2$ (Acute angle).
After selecting the sign of v_j. The sign of w_j is defined because it is the same sign of v_j.
No such problem for the translational estimation, but it suppose that the rotation matrix $\mathtt{R}$ is known, so, the translation estimation is not valid if the rotational estimation is wrong.
In the next section, the estimated rotation and translation $\mathtt{R}, t$, will be used to define the state vectors.

3.1.3 Velocity control laws

After the estimation of the rotation matrix $\mathtt{R}$ and the translation vector t, the velocity control laws presented in section (3.1) are ready to conduit the system towards the desired pose.

The error of servoing (the task function) is defined by the difference between the current state vector and the desired one. The rotational state vector is $s_\Omega = \theta u$ and the translational state vector is $s_V = \mathtt{R}^\mathsf{T} t$.

So the rotational task function is $e_\Omega = \theta u$ as the desired state vector is null $s^*_\Omega = 0$ and the translational task function is $e_V = \mathtt{R}^\mathsf{T} t$ as the desired state vector is null $s^*_V = 0$.

Figure (3.3) shows the block diagram of the PBVS using $\theta u, \mathtt{R}^\mathsf{T} t$ as state vectors.

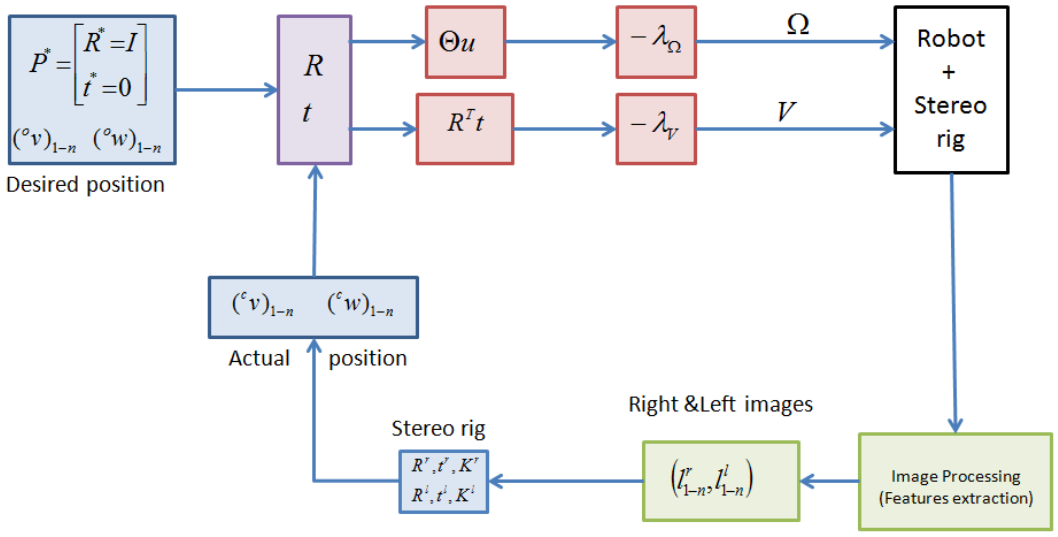


Figure 3.3: PBVS using $[\mathtt{R}, t]$ for pose representation.

In this figure, the desired pose P^* is represent by the desired state vectors $[{}^O\mathtt{R}_{c^*} = \mathtt{I}, {}^O t_{c^*} = 0]$.

The normalized Plücker coordinates of the straight lines at the object frame $\left(({}^o v_j, {}^o w_j)\ j = 1..n\right)$ are compared with those in the current pose $\left(({}^c v_j, {}^c w_j)\ \ j = 1..n\right)$, so as to calculate the relative rotation and the translation $([\mathtt{R}, t])$ between the current and desired state vectors.

$[\mathtt{R}, t]$ define the rotational and translational task functions, then the velocity control laws $\mathbf{V}, \mathbf{\Omega}$.

3.1.4 Simulations with Euclidean control laws

The goal of the simulations is to ensure the decoupling and the global asymptotic stability (GAS) convergence of the servoing using the "Euclidean control laws" defined in section (3.1.3).

$$\begin{aligned} \mathbf{\Omega} &= -\lambda_\Omega \theta u \\ \mathbf{V} &= -\lambda_V \mathtt{R}^\mathsf{T} t \end{aligned}$$

Many simulations have been done, with different configurations for the stereo rig, and different configurations of the object of interest in the initial and desired position. But as the results are always the same, we present here one simulation for configuration similar to the practical tests, and repeat it with noise or with error of calibration.

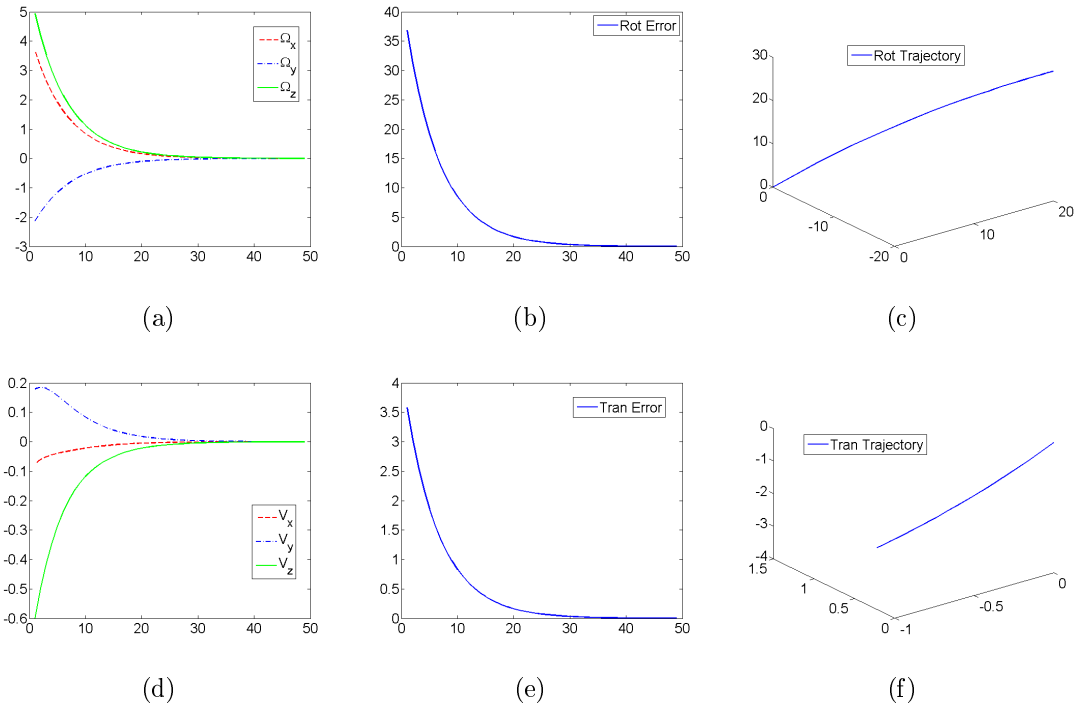
As the simulation does not interested of the robot dynamic, the robot is considered as a perfect integrator.
The stereo system consists of two parallel identical cameras. The homogeneous matrices of transformation between the two cameras and the stereo rig frame F_c are $\mathtt{R}_r = \mathtt{R}_l = \mathtt{I}, t_r = -t_l = (0.1, 0, 0)^\mathsf{T}$.
The intrinsic parameters of the two cameras used in these simulations are

$$\mathtt{K_r} = \mathtt{K_l} = \begin{bmatrix} 1200 & 0 & 300 \\ 0 & 1200 & 200 \\ 0 & 0 & 1 \end{bmatrix}$$

These matrices supposed to be exact (the stereo system is perfectly calibrated) except for the simulations with errors of calibration.

The object of interest consists of three non coplanar straight lines. The current relative pose between the object and the stereo rig is represented by the normalized Plücker coordinates of the 3 straight lines in the object v_1, w_1, v_2, w_2, v_3, w_3. While the desired pose is defined by the desired rotation matrix $\mathtt{R}^* = \mathtt{I}$ and the desired translation vector $t^* = 0$.
The final relative pose between the object and the stereo rig is represented by the normalized Plücker coordinates of the 3 straight lines of the object $(v_1(\infty), v_2(\infty), v_3(\infty)$, $w_1(\infty), w_2(\infty), w_3(\infty))$. The gains of the close loop are $\lambda_\Omega = \lambda_V = 0.15$.
The servoings of the rotation and translation are done simultaneously and that due to the full decoupling between them.

Figure (3.4) shows the first series of simulations for the control laws defined in section (3.1.3).
In this simulation, the rotation between the initial and final pose for each line in the object of interest is less than $\pi/2$.



(a) (b) (c)

(d) (e) (f)

Figure 3.4: *Simulation results for the PBVS using the "Euclidean control laws". The object of interest consists of three 3D straight lines. The horizontal axis of the figures is the iteration number. a) Rotation velocity (deg/s). b) Rotation error (rotational task function e_Ω) (deg). c) Rotation trajectory. d) Translation velocity (m/s). e) Translation error (translational task function e_V) (m). f) Translation trajectory. The errors (the task functions) are null at the end of servoing, which show the asymptotic convergence of state vectors.*

Figures (3.4 - a, d) show an exponential decrease of the rotational and the translational velocities. This is the desired behaviour of the system in the closed loop.
The asymptotic convergence of the task function is illustrated in figures (3.4 - b, e). Finally, as the state vectors are defined in the Euclidean space, the trajectory of the stereo rig frame origin is linear, as one can see in figure (3.4 - f), also the trajectory of the rotation angle θ (fig. 3.4 - c).

Let us add white noise for the image, which provides an error in the estimation of the 2D line parameters in right and left image.

To do this, we generate 100 points for each 2D line, then add a random value in the domain $[-1, 1]$ for the points coordinates in pixel, after that we estimate the 2D line parameters using the least squares method.
Figures (3.5 - a, b), show the convergence of rotational and translational velocities with a small oscillation magnitude near the equilibrium point.
Figures (3.5 - c, d), show the asymptotic convergence of the rotational and translational task functions, but with non null error at the end of servoing, about (0.2 deg) in rotation and (2 cm) in translation.

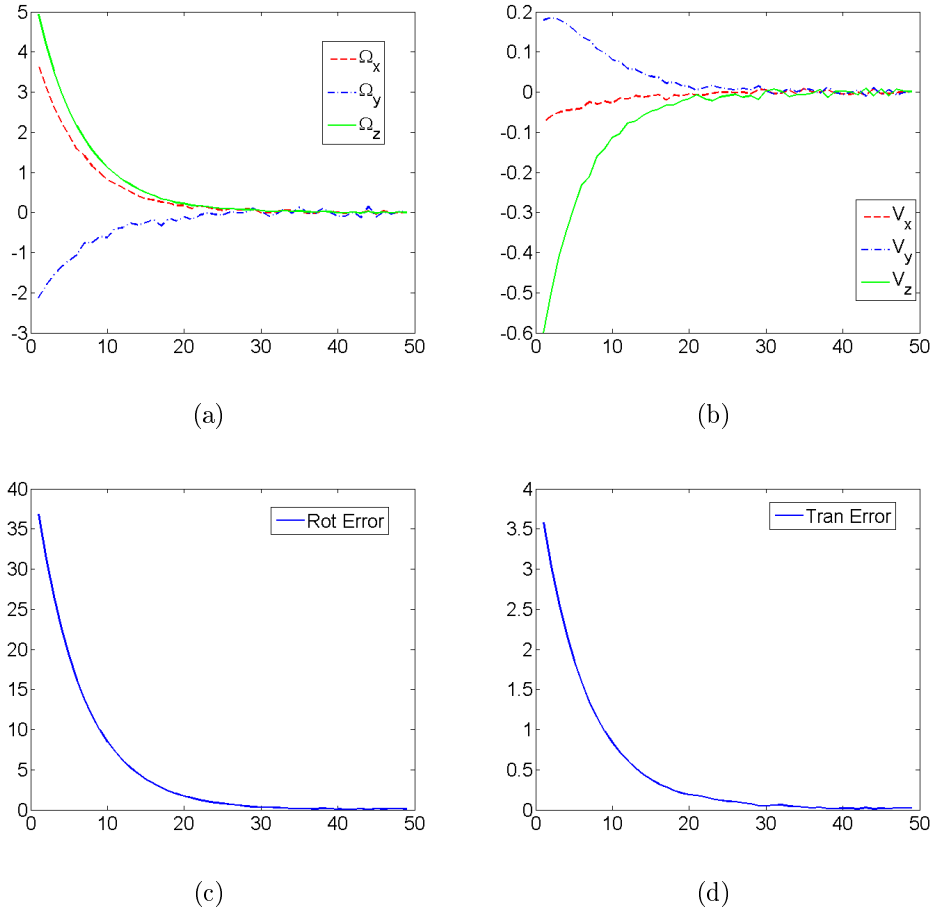


Figure 3.5: *Simulation results for the PBVS using the "Euclidean control laws" in the presence of white noise. The object of interest consists of three 3D straight lines. The horizontal axis of the figures is the iteration number. a) Rotation velocity (deg/s). b) Translation velocity (m/s). c) Rotation error (rotational task function e_Ω) (deg). d) Translation error (translational task function e_V) (m). The errors (the task functions) are almost null with small perturbations at the end of servoing, which show the asymptotic convergence of state vectors.*

In the next simulations, we propose to include some calibration errors, in the rotation between the two camera frames about ± 1 deg, in translation between the

camera centers about ±2 mm, about ±0.2 mm in the focal length, and ±2 pixels in the image centers.

As one can see in figures (3.6 - a, b), the control laws are not affected by the errors of calibration, and they decrease to zero. While the task functions (the error of servoing) are very affected, one can see a static error about 12 cm of translation and 2deg of rotation (fig. 3.6 - c, d).

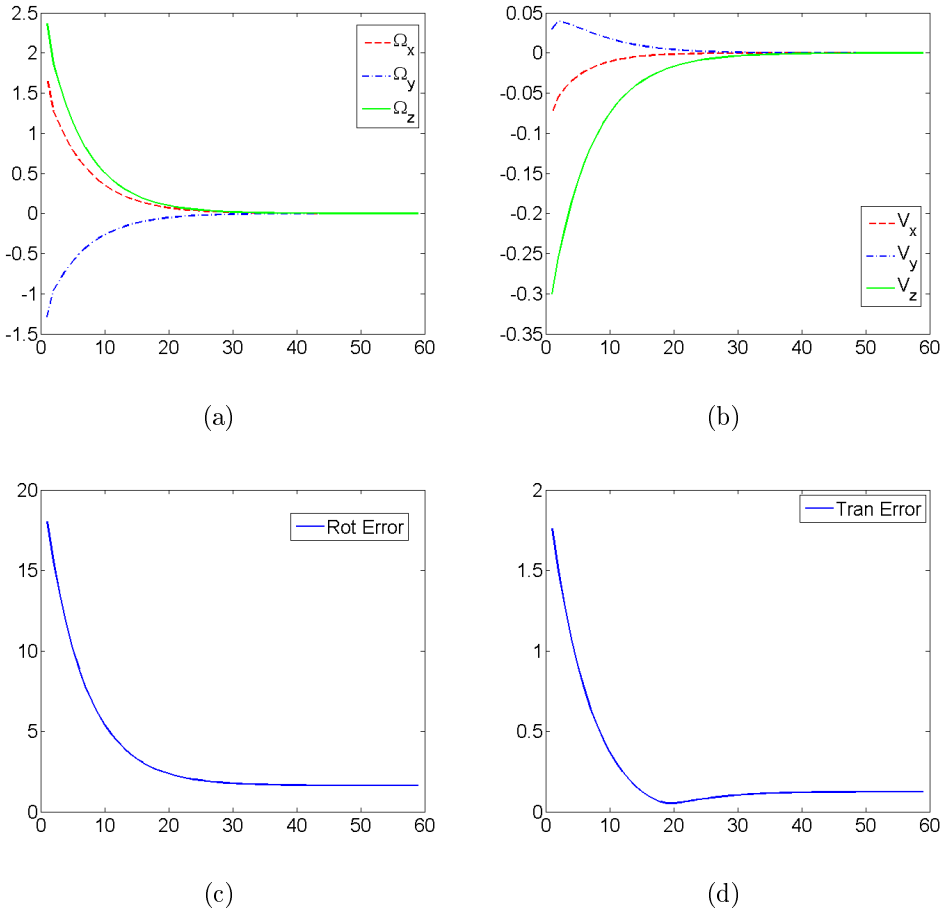


Figure 3.6: *Simulation results for the PBVS using the "Euclidean control laws" with calibration errors. The object of interest consists of three 3D straight lines. The horizontal axis of the figures is the iteration number. a) Rotation velocity (deg/s). b) Translation velocity (m/s). c) Rotation error (rotational task function e_Ω) (deg). d) Translation error (translational task function e_V) (m). The errors (the task functions) are not null at the end of servoing, because of the calibration errors. While the control laws are decreasing to zero with no effect of the calibration errors.*

3.1.5 Experiments on robot platform for the Euclidean control laws

The Euclidean control laws are applied to a 6 DOF robot to control the relative pose between the end-effector of the robot and the object of interest in the view field of the stereo rig.
The Adept Viper S650 robot has 6 axes and 6 revolute joints, which gives 6 DOF in the Euclidean space (see figure 3.7).

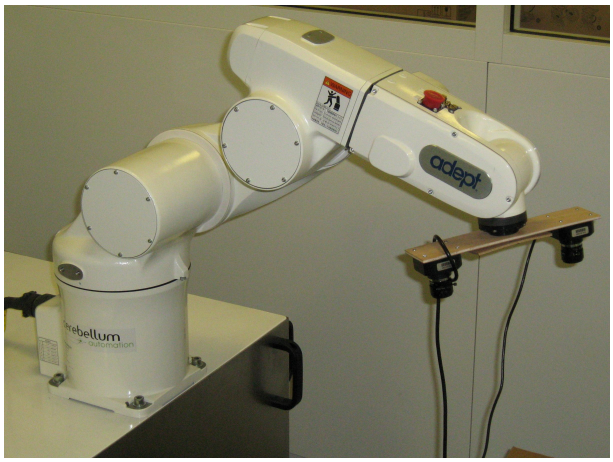


Figure 3.7: The Adept Viper S650 robot.

Providing the robot with the "Cerebellum Automation" hardware package of amplifiers and controllers, without forgetting the software library "CIDE" make the velocity control of the robot available through a fire-wire 1394 connection with high speed command transformation.
The stereo rig consists of two USB2 identical cameras "SUMIX SMX-150M", provided with a 8mm objective, they can acquire images of size 1280×1024 pixel at $27.5 fps$.
The two cameras are fixed in parallel configuration at the robot end-effector (see figure 3.7). The acquired image size is 600×400 pixel at $75 fps$.
The object of interest is a white triangle painted on a black background, the three edges of the triangle are considered as the three 3D straight lines of the object.

Figures (3.8 - a, b) show the right and left images of the object of interest in the initial pose, and figures (3.8 - c, d) show the right and left images of the object of interest in the desired pose.

To segment the image, we use a Canny detector to extract the pixels of every line, then we use a robust line estimation to estimate the parameters of the line. This robust estimation randomly chose two points of the line, estimate the line pa-

(a) (b) (c) (d)

Figure 3.8: *The object of interest. a) Right image in the initial pose. b) Let image in the initial pose. c) Right image in the final pose. d) Left image in the final pose.*

rameters using these two points, and test the maximum number of other points that belong to this line with a distance less than one pixel. The pair of points who verify the maximum number of in-lines serves to compute the line parameters.

To maintain a high frame rate, a Canny-based fast edges detector (within a tracked area of interest) and a RANSAC-based line fitting have been implemented in C++ with Linux and multi-threading instructions (one thread per robust line fitting) and executed with a Core 2 Duo 2.67 Ghz PC.

For practical reasons, and as the goal of the experience is to validate the global asymptotic stability of the system in the closed loop using the Euclidean control laws, the initial and desired poses are defined using the 3D estimation of the lines Plücker coordinates in the stereo images.

The stereo rig captures the image lines at the desired pose. The normalized Plücker coordinates $v(\infty), w(\infty)$ are estimated from the stereo image for each line. Then the stereo rig is moved to the initial pose before the starting of the servoing. The reference frame origin is defined at the center between the two cameras, so the extrinsic stereo matrices are given by

$$\mathtt{R}_r = \mathtt{R}_l = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad , \quad t_r = -t_l = \begin{pmatrix} 0.1 \\ 0 \\ 0 \end{pmatrix}$$

The intrinsic stereo matrices are defined using the cameras technical specifications (focal length, sensor length, and image size), we have

$$\mathtt{K}_r = \mathtt{K}_l = \begin{bmatrix} 1200 & 0 & 300 \\ 0 & 1200 & 200 \\ 0 & 0 & 1 \end{bmatrix}$$

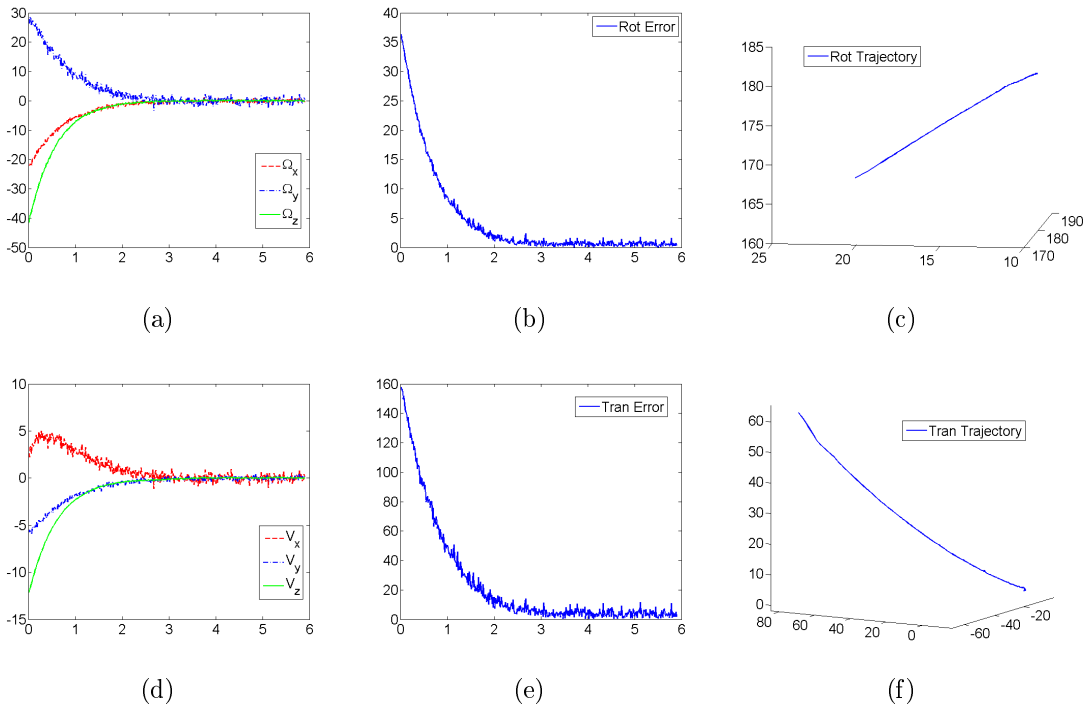


Figure 3.9: *Experimental results for the Euclidean control laws. The object of interest consists of three 3D straight lines. The horizontal axis of the figures is the time in second. a) Rotational velocity (deg/s). b) Rotation error (deg). c) Rotation trajectory (the angle θ trajectory). d) Translational velocity (mm/s). e) Translation error(mm). f) Translation trajectory (motion of the origin of the stereo frame). The errors (the task functions) are almost null at the end of servoing, which shows the asymptotic convergence of state vectors.*

Figure (3.9) shows the experiment results. The stability of rotational and translational velocities is shown in figures (3.9 - a, d), with a small perturbation during the servoing and especially near the equilibrium point. One can see the similarity to the simulation results in the case of adding white noise.
Figures (3.9 - b, e), show the asymptotic convergence of the rotational and translational task functions, with a small error at the end of servoing, about (0.5deg) in rotation and (4mm) in translation.
Finally, the trajectory of the robot end effector is shown in figures (3.9 - c, f). One can see a line trajectory for the translation and for the rotation in the Euclidean space.
There is no static errors in the final pose of the stereo rig has been done because the estimation of the desired pose using the stereo rig, in the same way of the estimation of the current pose during the servoing.

3.2 Position based visual servoing using Plücker coordinates

In the previous section, the Plücker coordinates are used to estimate the relative rotation matrix $\mathtt{R}$ and the relative translation vector t between the current and the desired poses. The goal of this section is to design decoupled controls laws for rotation and translation, without pose estimation.
The object of interest consists of n straight lines $(L_1, ..., L_n)$ represented by the Plücker coordinates $\big((v_j, w_j) \;\; j = 1..n\big)$ at the current pose, and $\big((v_j^*, w_j^*) \;\; j = 1..n\big)$ at the final or desired pose.
The control laws designed in this section will be called " Plücker coordinates control laws".

3.2.1 Rotational velocity control law $\boldsymbol{\Omega}$

To design a state vector for rotation control law, we recall the equation (2.5) which defines a direct relation between the rotation velocity $\boldsymbol{\Omega}$ and the kinematic of the direction vector v, $\dot{v} = \boldsymbol{\Omega} \times v$.
By extending this equation for n lines we get:

$$\begin{aligned} \dot{v}_1 &= \boldsymbol{\Omega} \times v_1 = -[v_1]_\times \boldsymbol{\Omega} \\ &\vdots \\ \dot{v}_n &= \boldsymbol{\Omega} \times v_n = -[v_n]_\times \boldsymbol{\Omega} \end{aligned}$$

For translating these n equations in one matrix equation, we define

$$q = \begin{pmatrix} \dot{v}_1 \\ \vdots \\ \dot{v}_n \end{pmatrix} \quad \text{and} \quad \mathtt{H}_\Omega = \begin{bmatrix} -[v_1]_\times \\ \vdots \\ -[v_n]_\times \end{bmatrix}$$

which makes the equation form like $q = \mathtt{H}_\Omega \, \boldsymbol{\Omega}$. The solution of this equation is given by $\boldsymbol{\Omega} = \mathtt{H}_\Omega^+ \, q$. In section (3.1) we saw that the only condition for the existence of $\mathtt{H}_\Omega^+$ is to have at least two non parallel lines in the object.
To carry out that control law, we define a rotational task function

$$e_\Omega = \begin{pmatrix} v_1 - v_1^* \\ \vdots \\ v_n - v_n^* \end{pmatrix} \tag{3.5}$$

and we suppose an exponential decrease of the form $\dot{e}_\Omega = -\lambda_\Omega \, e_\Omega$. As the desired pose is invariable (constant during the servoing), then, the time derivation of the

task function is $\dot{e}_\Omega = (\dot{v}_1^\mathsf{T}, \ldots, \dot{v}_n^\mathsf{T})^\mathsf{T}$ and the rotation control law is

$$\boldsymbol{\Omega} = -\lambda_\Omega \, \mathtt{H}_\Omega^+ \, e_\Omega \tag{3.6}$$

Discussion

This selection of the task function is classical. It defines the error as the difference between the state vectors (the direction vectors v_j in this case), and the desired state vectors (v_j^*).

Like in the previous section (3.1.2), the sign of the vectors v_j is not defined, so, the rotation for each line of the object, between the initial and the final poses is assumed to be less than $\pi/2$.

3.2.2 Stability and convergence of the rotational control law

To prove the stability of the closed control loop with the rotation control law defined in (3.6), and the asymptotic convergence of the task function (3.5), we use the Lyapunov function $\mathcal{L}(t) = 1/2||e_\Omega(t)||^2$ and study the sign of its time derivative $\dot{\mathcal{L}}(t)$

$$\begin{aligned}
\dot{\mathcal{L}}(t) &= \dot{e}_\Omega^\mathsf{T} e_\Omega \\
&= \begin{pmatrix} \dot{v}_1 \\ \vdots \\ \dot{v}_n \end{pmatrix}^\mathsf{T} \begin{pmatrix} v_1 - v_1^* \\ \vdots \\ v_n - v_n^* \end{pmatrix} \\
&= \begin{pmatrix} \boldsymbol{\Omega} \times v_1 \\ \vdots \\ \boldsymbol{\Omega} \times v_n \end{pmatrix}^\mathsf{T} \begin{pmatrix} v_1 - v_1^* \\ \vdots \\ v_n - v_n^* \end{pmatrix} \\
&= \begin{pmatrix} -[v_1]_\times \boldsymbol{\Omega} \\ \vdots \\ -[v_n]_\times \boldsymbol{\Omega} \end{pmatrix}^\mathsf{T} \begin{pmatrix} v_1 - v_1^* \\ \vdots \\ v_n - v_n^* \end{pmatrix} \\
&= (\mathtt{H}_\Omega \boldsymbol{\Omega})^\mathsf{T} e_\Omega = \boldsymbol{\Omega}^\mathsf{T} \mathtt{H}_\Omega^\mathsf{T} e_\Omega
\end{aligned}$$

Given $\boldsymbol{\Omega} = -\lambda_\Omega \, \mathtt{H}_\Omega^+ \, e_\Omega$ we find $\boldsymbol{\Omega}^\mathsf{T} = -\lambda_\Omega \, e_\Omega^\mathsf{T} \, \mathtt{H}_\Omega \, (\mathtt{H}_\Omega^\mathsf{T} \mathtt{H}_\Omega)^{-1}$ so

$$\dot{\mathcal{L}}(t) = -\lambda_\Omega \, e_\Omega^\mathsf{T} \, \mathtt{H}_\Omega \, (\mathtt{H}_\Omega^\mathsf{T} \mathtt{H}_\Omega)^{-1} \mathtt{H}_\Omega^\mathsf{T} \, e_\Omega$$

To determine the sign of the Lyapunov function derivation, we start with $(\mathtt{H}_{\boldsymbol{\Omega}}^\mathsf{T} \mathtt{H}_{\boldsymbol{\Omega}})^{-1}$, and we prove that $\mathtt{H}_\Omega^\mathsf{T} \mathtt{H}_\Omega$ is positive definite, so its invert is positive definite too.

$$\forall\ a \in \mathbb{R}^{*3}$$

$$\begin{aligned} a^{\mathsf{T}}\mathtt{H}_{\Omega}^{\mathsf{T}}\mathtt{H}_{\Omega}a &= a^{\mathsf{T}}\left[[v_1]_\times,\ldots,[v_n]_\times\right]\begin{bmatrix} -[v_1]_\times \\ \vdots \\ -[v_n]_\times \end{bmatrix} a \\ &= -a^{\mathsf{T}}\sum_i [v_i]_\times[v_i]_\times a \\ &= -\sum_i a^{\mathsf{T}}[v_i]_\times[v_i]_\times a \\ &= \sum_i ||v_i \times a||^2 \ > \ 0 \end{aligned}$$

If there are two distinct vectors (v_i, v_j) that is, two non parallel lines in the object, the matrix $\mathtt{H}_{\Omega}^{\mathsf{T}}\mathtt{H}_{\Omega}$ will be positive definite.

Now the part $e_{\Omega}^{\mathsf{T}}\ \mathtt{H}_{\Omega}\mathtt{H}_{\Omega}^{\mathsf{T}}\ e_{\Omega}$

$$\begin{aligned} e_{\Omega}^{\mathsf{T}}\mathtt{H}_{\Omega}\mathtt{H}_{\Omega}^{\mathsf{T}}e_{\Omega} &= \begin{pmatrix} v_1 - v_1^* \\ \vdots \\ v_n - v_n^* \end{pmatrix}^{\mathsf{T}}\begin{bmatrix} -[v_1]_\times \\ \vdots \\ -[v_n]_\times \end{bmatrix}\begin{bmatrix} -[v_1]_\times \\ \vdots \\ -[v_n]_\times \end{bmatrix}^{\mathsf{T}}\begin{pmatrix} v_1 - v_1^* \\ \vdots \\ v_n - v_n^* \end{pmatrix} \\ &= \left(\begin{bmatrix} -[v_1]_\times \\ \vdots \\ -[v_n]_\times \end{bmatrix}^{\mathsf{T}}\begin{pmatrix} v_1 - v_1^* \\ \vdots \\ v_n - v_n^* \end{pmatrix}\right)^{\mathsf{T}}\begin{bmatrix} -[v_1]_\times \\ \vdots \\ -[v_n]_\times \end{bmatrix}^{\mathsf{T}}\begin{pmatrix} v_1 - v_1^* \\ \vdots \\ v_n - v_n^* \end{pmatrix} \\ &= \left|\left|\begin{bmatrix} -[v_1]_\times \\ \vdots \\ -[v_n]_\times \end{bmatrix}^{\mathsf{T}}\begin{pmatrix} v_1 - v_1^* \\ \vdots \\ v_n - v_n^* \end{pmatrix}\right|\right|^2 \\ &= \left|\left|\left[[v_1]_\times,\ldots,[v_n]_\times\right]\begin{pmatrix} v_1 - v_1^* \\ \vdots \\ v_n - v_n^* \end{pmatrix}\right|\right|^2 \\ &= \left|\left|\sum_i v_i \times v_i^*\right|\right|^2 \end{aligned}$$

We will prove that $\sum_i v_i \times v_i^* = 0$ if and only if $v_i = v_i^*$ (proof and details are in [AEH02]).

Note that $v_i{}^* = \mathtt{R}v_i$, using the representation axis/angle θu of the rotation matrix $\mathtt{R}$

and Rodrigue's formula, we can write

$$\begin{aligned} v_i^* &= v_i\cos(\theta) + (u \times v_i)\sin(\theta) + (u^\mathsf{T} v_i)(1-\cos(\theta))u \\ \text{so}\quad v_i \times v_i^* &= v_i \times (u \times v_i)\sin(\theta) + (v_i \times u)(u^\mathsf{T} v_i)(1-\cos(\theta)) \\ v_i \times v_i^* &= \big(u - (u^\mathsf{T} v_i)v_i\big)\sin(\theta) + (v_i \times u)(u^\mathsf{T} v_i)(1-\cos(\theta)) \end{aligned}$$

Now, taken $(u, u \times v_i, u \times (u \times v_i))$ as an orthogonal basis, and projecting the last equation to this basis, we get

$$\begin{aligned} v_i \times v_i^* &= \Big[\big(1-(u^\mathsf{T} v_j)^2\big)\sin(\theta)\Big]\ u \\ &- \Big[\big(u^\mathsf{T} v_i)(1-\cos(\theta)\big)\Big]\ u \times v_i \\ &+ \Big[\big(1-(u^\mathsf{T} v_j)^3\big)\sin(\theta)\Big]\ u \times (u \times v_i)) \end{aligned}$$

Then, if $\sum_i v_i \times {v_i}^* = 0$ we get $\sum_i \big(1-(u^\mathsf{T} v_j)^2\big)\sin(\theta) = 0$.

If there are two non parallel lines in the object of interest, then there exists i such that $(u^\mathsf{T} v_i)^2 < 1$. So the only solution is $\sin(\theta) = 0$, that means $\theta = 0$ as the angles are acutes, then $v_i = {v_i}^*$.
That proves the global asymptotic stability of the rotation control law $\boldsymbol{\Omega} = -\lambda_\Omega \, \mathtt{H}_\Omega^+ \, e_\Omega$ for any positive real number $\lambda_\Omega > 0$.

3.2.3 Translational velocity control law V

As the vector w is depending on the orientation between the initial and the desired poses, it can not be directly used as a state vector for translation, if we look for a full decoupled control laws.
In [AEH02], Andreff *et al.* use the bi-normalized Plücker coordinates to verify a cascaded control. They used the vector $v \times (w - w^*), ||w|| = 1$ so as to define the translation error after the end of the rotation servoing ($\boldsymbol{\Omega} = 0, v = v^*$).
To define a task function for the translation, which is independent of rotation task function e_Ω, for that we calculate, $\dot{w}^\mathsf{T} w = u^\mathsf{T}\mathbf{V}$.
Note that $\dot{w}^\mathsf{T} w = \frac{1}{2}\frac{d}{dt}||w||^2 = \frac{1}{2}\frac{d}{dt} w^\mathsf{T} w$. Or recall the equation in [AEH02] which defines the time derivation of the distance d of a line: $\dot{d} = u^\mathsf{T}\mathbf{V}$ with $||u|| = 1$, as in normalized Plücker coordinates $d = ||w||$.
Let us define $d^2 = w^\mathsf{T} w$ as a state vector for the translation, and the desired one will be d^{2*}, now for n lines,

$$\begin{aligned} \dot{w}_1^\mathsf{T} w_1 &= u_1^\mathsf{T}\mathbf{V} \\ &\vdots \\ \dot{w}_n^\mathsf{T} w_n &= u_n^\mathsf{T}\mathbf{V} \end{aligned}$$

Suppose $\mathtt{H}_V = [u_1, ..., u_n]^\mathsf{T}$, the task function is then chosen as

$$e_V = \frac{1}{2} \begin{pmatrix} d_1^2 - d_1^{2*} \\ \vdots \\ d_n^2 - d_n^{2*} \end{pmatrix} \tag{3.7}$$

for an exponential decrease of the task function given by $\dot{e}_V = -\lambda_V e_V$ we get the translation velocity control law

$$\mathbf{V} \;=\; -\lambda_V \mathtt{H}_V^+ e_V \tag{3.8}$$

Note: We can use the time derivation of the distance $\dot{d}_i = u_i^\mathsf{T}\mathbf{V}$ with $||u_i|| = 1$, in this case, the task function e_V and the interaction matrix $\mathtt{H}_V$ are defined by

$$e_V = \begin{pmatrix} d_1 - d_1^* \\ \vdots \\ d_n - d_n^* \end{pmatrix} \qquad \mathtt{H}_V = \begin{bmatrix} u_1^\mathsf{T} \\ \vdots \\ u_n^\mathsf{T} \end{bmatrix} \qquad ||u_i|| \;=\; 1$$

3.2.4 Stability and convergence of the translational control law

To prove the stability of the servoing closed loop with the translational control law given in (3.8), and the asymptotic convergence of the task function (3.7), we use the Lyapunov function $\mathcal{L}(t) = ||e_V(t)||^2$ and study the sign of its time derivative $\dot{\mathcal{L}}(t)$

$$\begin{aligned}
\dot{\mathcal{L}}(t) \;&=\; 2\, \dot{e}_V^\mathsf{T} e_V \\
&=\; \begin{pmatrix} \dot{w}_1^\mathsf{T} w_1 \\ \vdots \\ \dot{w}_n^\mathsf{T} w_n \end{pmatrix}^\mathsf{T} \begin{pmatrix} d_1^2 - d_1^{2*} \\ \vdots \\ d_n^2 - d_n^{2*} \end{pmatrix} \\
&=\; \begin{pmatrix} u_1^\mathsf{T}\mathbf{V} \\ \vdots \\ u_n^\mathsf{T}\mathbf{V} \end{pmatrix}^\mathsf{T} \begin{pmatrix} d_1^2 - d_1^{2*} \\ \vdots \\ d_n^2 - d_n^{2*} \end{pmatrix} \\
&=\; (\mathtt{H}_V\mathbf{V})^\mathsf{T} e_V \;=\; \mathbf{V}^\mathsf{T}\mathtt{H}_V^\mathsf{T} e_V
\end{aligned}$$

Using the control law defined in (3.8) for the translational velocity, we obtain

$$\begin{aligned}
\mathbf{V} \;&=\; -\lambda_V \mathtt{H}_V^+ e_V \\
\mathbf{V}^\mathsf{T} \;&=\; -\lambda_V e_V^\mathsf{T} \mathtt{H}_V^\mathsf{T} (\mathtt{H}_V^\mathsf{T}\mathtt{H}_V)^{-1}
\end{aligned}$$

$$\dot{\mathcal{L}}(t) \;=\; -\lambda_V e_V^\mathsf{T} \mathtt{H}_V^\mathsf{T} (\mathtt{H}_V^\mathsf{T}\mathtt{H}_V)^{-1} \mathtt{H}_V^\mathsf{T} e_V$$

First, we show that, the matrix $(\mathtt{H}_V^\mathsf{T}\mathtt{H}_V)^{-1}$ is positive definite,

$$\begin{aligned}
&\forall\ a \in \mathbb{R}^{*3} \\
a^\mathsf{T}\mathtt{H}_V^\mathsf{T}\mathtt{H}_V a &= a^\mathsf{T}[u_1,\ldots,u_n]\begin{bmatrix} u_1^\mathsf{T} \\ \vdots \\ u_n^\mathsf{T} \end{bmatrix} a \\
&= a^\mathsf{T}[u_1u_1^\mathsf{T} + \ldots + u_nu_n^\mathsf{T}]a \\
&= \sum_i a^\mathsf{T}u_iu_i^\mathsf{T}a \ = \ \sum_i ||a^\mathsf{T}u_i||^2
\end{aligned}$$

So, if there are three non coplanar vectors u_i, the matrix $(\mathtt{H}_V^\mathsf{T}\mathtt{H}_V)$ is definite positive, so the matrix $(\mathtt{H}_V^\mathsf{T}\mathtt{H}_V)^{-1}$ exist and definite positive.
Now, the part $e_V^\mathsf{T}\mathtt{H}_V\mathtt{H}_V^\mathsf{T}e_V$ will be developed to show if it is definite positive,

$$\begin{aligned}
e_V^\mathsf{T}\mathtt{H}_V &= (e_1,\ldots,e_n)\begin{bmatrix} u_1^\mathsf{T} \\ \vdots \\ u_n^\mathsf{T} \end{bmatrix} = \sum_i e_iu_i^\mathsf{T} \\
\mathtt{H}_V^\mathsf{T}e_V &= [u_1,\ldots,u_n]\begin{pmatrix} e_1 \\ \vdots \\ e_n \end{pmatrix} = \sum_i e_iu_i \\
\text{so}\quad e_V^\mathsf{T}\mathtt{H}_V\mathtt{H}_V^\mathsf{T}e_V &= \Big|\Big|\sum_i e_iu_i\Big|\Big|^2
\end{aligned}$$

One can see that $e_V^\mathsf{T}\mathtt{H}_V\mathtt{H}_V^\mathsf{T}e_V$ can be null even if $e_V \neq 0$, because the dimension of the input vector ($\dim(s_V^*) = n$). The dimension of the system (translation system) is 3, so if $n > 3$ we will have a controllability problem.
To guarantee the asymptotic convergence, the input vector dimension must be equal to the system dimension ($n = 3$).
Notice that if $n = 3$, the matrix $\mathtt{H}_V$ is of dimensions 3×3, so $\mathtt{H}_V^+ = \mathtt{H}_V^{-1}$ if the vectors (u_1, u_2, u_3) are not coplanar. the time derivation of the Lyapunov function is $\dot{\mathcal{L}} = -\lambda_V e_V^\mathsf{T}e_V < 0$. That prove the asymptotic convergence of the translational task function.

Figure (3.10) shows the block diagram of the servoing closed loop using Plücker coordinates as 3D state vectors.
In this figure, the full decoupling of the velocities control laws is clear, using the direction vectors v_i for the rotational task function, and the norm of the vectors w_i (the distance d) for the translational task function. The desired pose is defined

by the bi-normalized Plücker coordinates for each line $(v_i^*, d_i^*, \quad i = 1..n)$, and the current pose is estimated from the image $(l_i^r, l_i^l, i = 1..n)$ and the stereo rig matrices $(\mathtt{R}_r, \mathtt{R}_l, t_r, t_l, \mathtt{K}_r, \mathtt{K}_l)$.

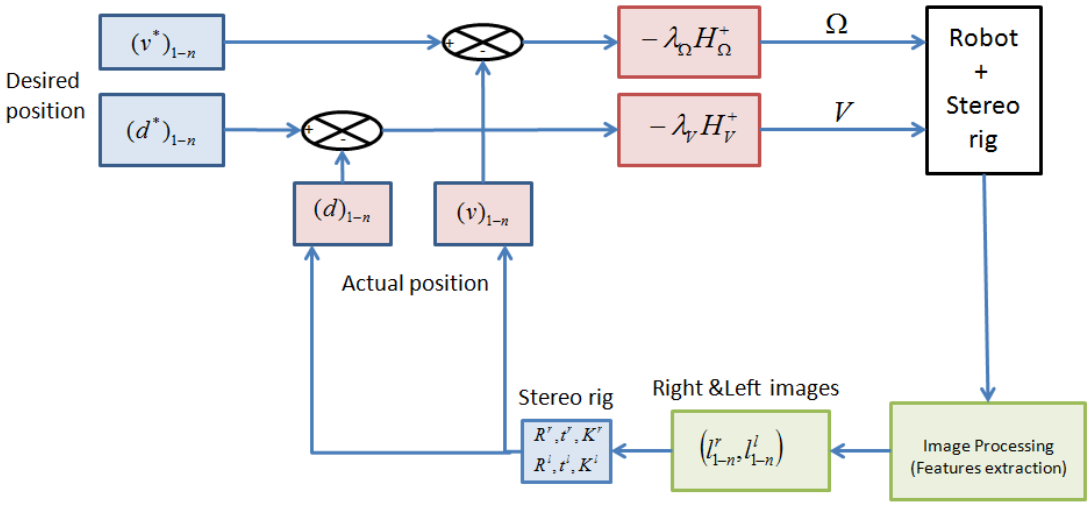


Figure 3.10: Block diagram of the servoing closed loop using Plücker coordinates as state vectors.

3.2.5 Simulations with "Plücker coordinates control laws"

The goal of these simulations is to show the global asymptotic stability GAS of the system in closed loop, using "Plücker coordinates control laws"

$$\begin{aligned} \mathbf{\Omega} &= -\lambda_\Omega \mathtt{H}_\Omega^+ e_\Omega \\ \mathbf{V} &= -\lambda_V \mathtt{H}_V^+ e_V \end{aligned}$$

Many simulations have been done, with different configurations for the stereo rig, and different configurations of the object of interest in the initial and desired position. But as the results are always the same, we present here one simulation for configuration similar to the practical tests, and repeat it with noise or with error of calibration.

Like what we proposed in the simulations of the "Euclidean control laws", the robot is considered as a perfect integrator. The stereo system consists of two parallel identical cameras. The homogeneous matrices of transformation between the two cameras and the stereo rig frame are $\mathtt{R}_r = \mathtt{R}_l = \mathtt{I}, t_r = -t_l = (0.1, 0, 0)^\mathsf{T}$.

The intrinsic parameters of the two cameras used in these simulations are

$$\mathtt{K_r} = \mathtt{K_l} = \begin{bmatrix} 1200 & 0 & 300 \\ 0 & 1200 & 200 \\ 0 & 0 & 1 \end{bmatrix}$$

These matrices supposed to be exact (the stereo system is perfectly calibrated) except for the simulations with errors of calibration.
The object of interest consists of three non coplanar straight lines. The current relative pose between the object and the stereo rig is represented by the bi-normalized Plücker coordinates of the $3D$ straight lines in the object $v_1, d_1,\ v_2, d_2,\ v_3, d_3$. The desired pose is defined by the desired bi-normalized Plücker coordinates of the $3D$ straight lines in the object $v_1^*, d_1^*,\ v_2^*, d_2^*,\ v_3^*, d_3^*$.
The closed loop gain are set to $\lambda_\Omega = \lambda_V = 0.15$.
As the rotational task function and the translational task function do not have units, we will illustrate the angle θ in the axis/angle representation of the rotation matrix to display the rotation error (just like the task function of the "Euclidean control laws"). The error of translation is represented by the Euclidean distance between the current frame origin and the final one.

Figure (3.11) shows the simulation results of the proposed position based visual servoing using "Plücker coordinates control laws".
Figures (3.11 - a, d) show the rotational and the translational velocities, their components decrease to zero, and make the system look like a first order system.
The asymptotic convergence of the servoing error is shown in figures (3.11 - b, e). The task functions decrease in exponential way as desired.
As the task functions (rotation and translation) are defined in Plücker coordinates space, the trajectory of the stereo rig (end effector of the robot) in the Euclidean space is not a straight line, can be seen in figures (3.11 - c, f).

Adding a white noise to the lines image (less than 1pixel), does not destruct the stability of the closed loop as one can see in figures (3.12 - a, b), but the errors in the final pose may be significant for the translational control law, while the rotational control law better resists (figures 3.12 - c, d).
One can measure an error about 5cm in the translation, this error is between the final poses with and without noise. An error about 0.5deg in the rotation between the final poses with and without noise.

Now, we test the effect of the calibration errors, for that we perform the same simulations in the case of "Euclidean control laws", and the results are reported in

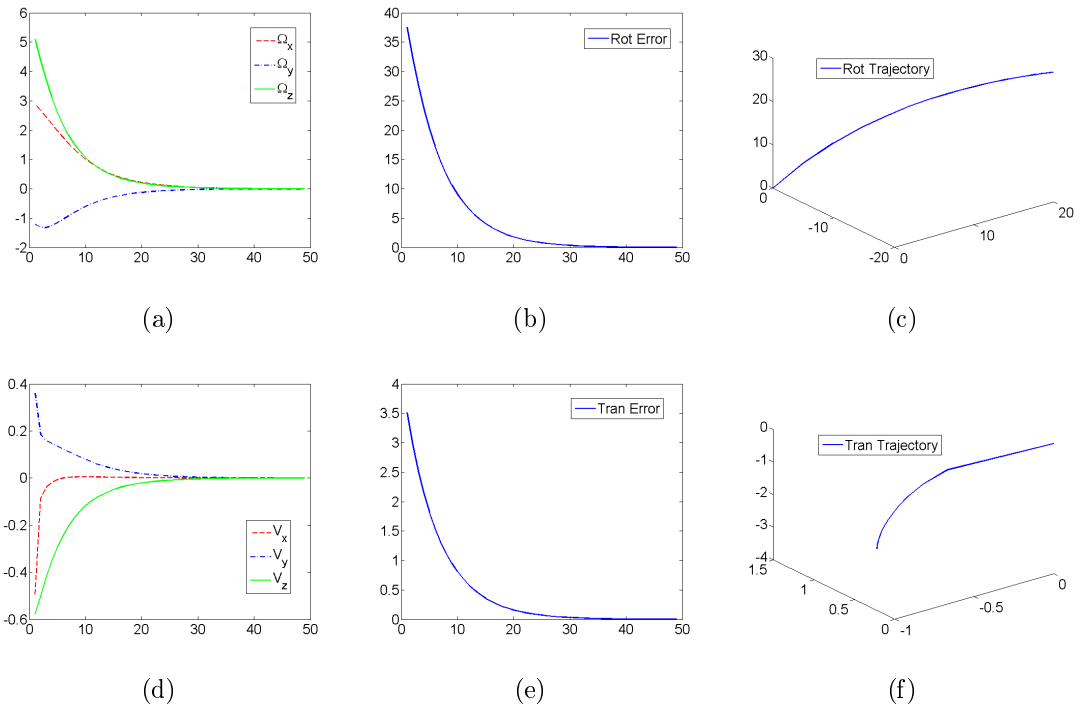


Figure 3.11: *Simulation results for the PBVS using the "Plücker coordinates control laws". The object of interest consists of three 3D straight lines. The horizontal axis of the figures is the iteration number. a) Rotation velocity (deg/s). b) Rotation error (represented by θ) (deg). c) Rotation trajectory. d) Translation velocity (m/s). e) Translation error (Euclidean distance) (m). f) Translation trajectory. The errors of convergence are null at the end of servoing, which show the asymptotic convergence of state vectors.*

figure (3.13).

As one can see in figure (3.13 - a, b), the control laws are not greatly affected by the errors of calibration, and they still decreased to zero. The task functions (the error of servoing) are very affected, one can see a static error of about 40 cm in translation and 2 deg in rotation (fig. 3.13 - c, d).

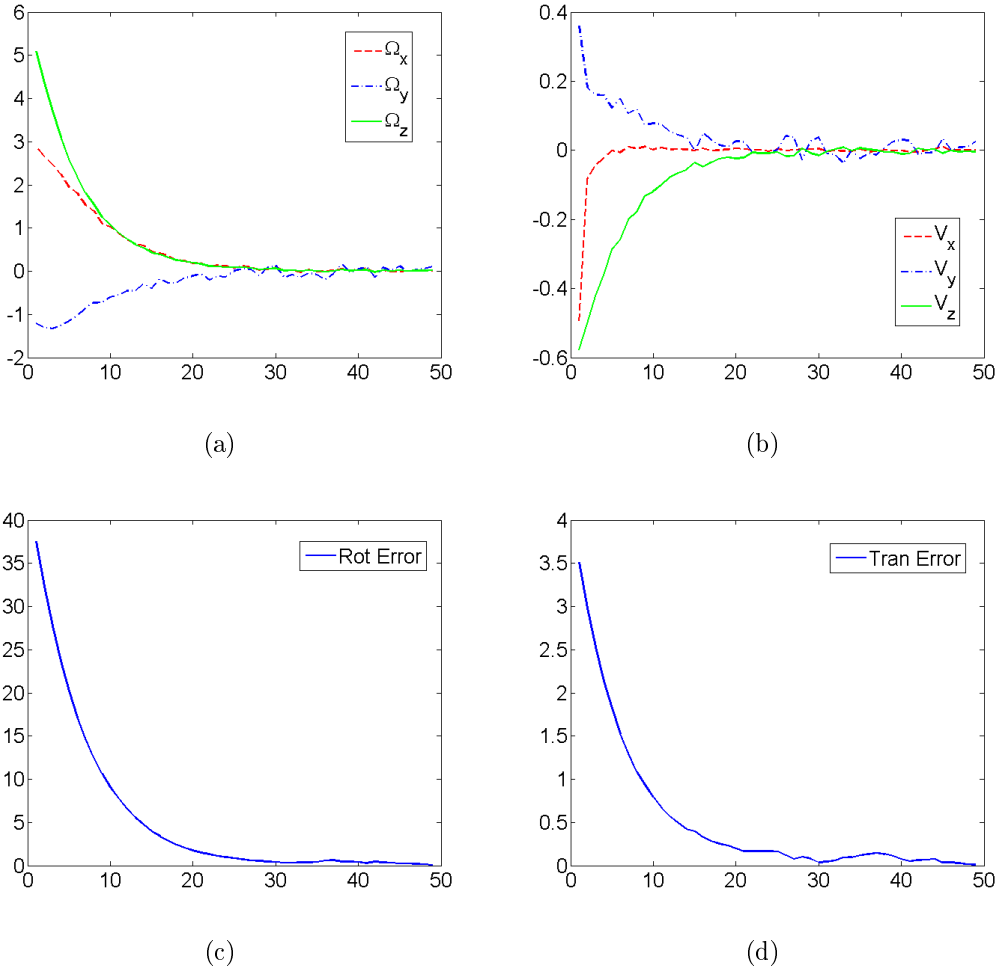


Figure 3.12: *Simulation results for the PBVS using the "Plücker coordinates control laws" after adding white noise to the image. The object of interest consists of three 3D straight lines. The horizontal axis of the figures is the iteration number. a) Rotation velocity (deg/s). b) Translation velocity (m/s). c) Rotation error (θ) (deg). d) Translation error (distance between the current and final frames origins) (m). The errors of convergence are almost null at the end of servoing, the velocity control laws are decreasing to zero, the system is stable.*

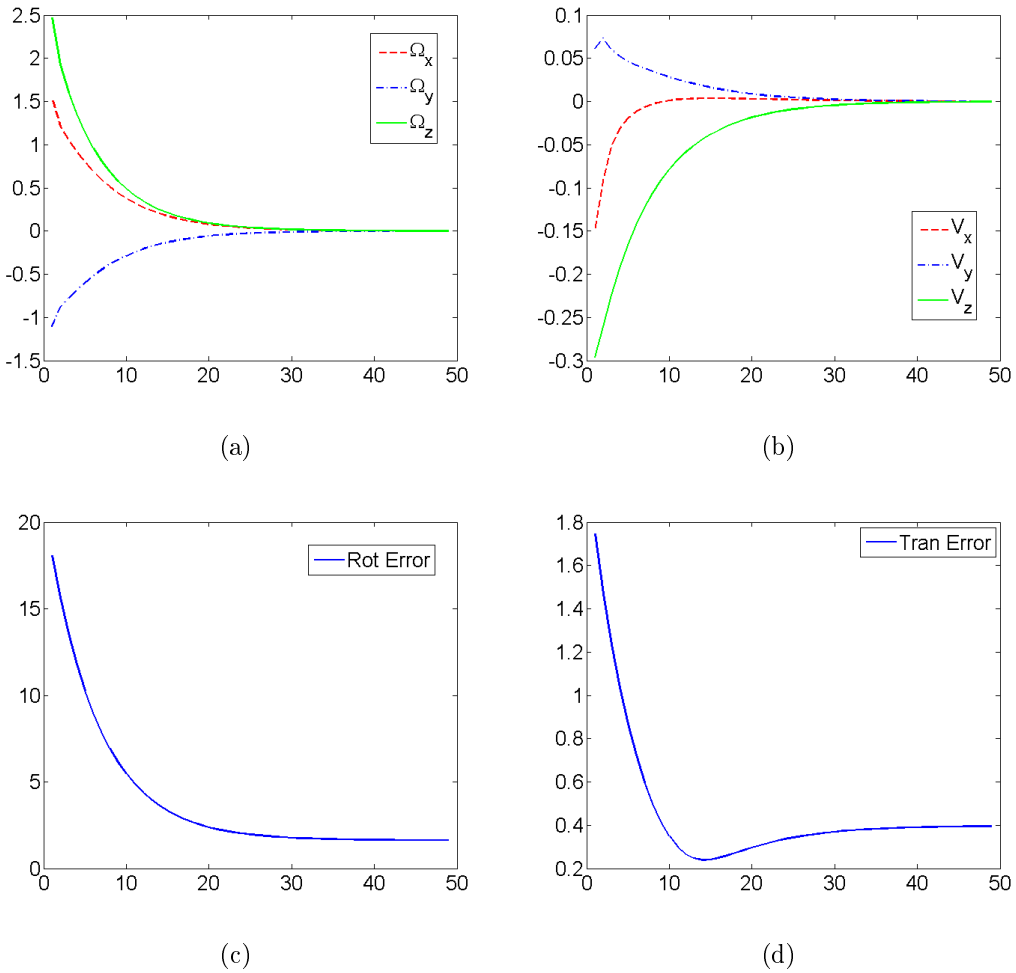


Figure 3.13: *Simulation results for the PBVS using the "Plücker coordinates control laws" with uncalibrated stereo system. The object of interest consists of three 3D straight lines. The horizontal axis of the figures is the iteration number. a) Rotation velocity (deg/s). b) Translation velocity (m/s). c) Rotation error (θ) (deg). d) Translation error (Euclidean distance between the current and final frames origins) (m). The translation error is very large, and the rotation error is important, these errors are not the convergence error, we always have an asymptotic convergence, but this is static error, due to the bad calibration of the stereo system.*

3.2.6 Experiments on platform robot using the "Plücker coordinates control laws"

Similarly to the "Euclidean control laws", we experiment the "Plücker coordinates control laws" with the help of the Viper S650 robot, using the same stereo rig.
The desired and the current poses are defined by the 3D reconstruction from the stereo image for the straight lines of the object of interest.
The goal is to move the embedded stereo rig at the robot end-effector, from the initial pose to the desired pose using the "Plücker coordinates control laws".
Figure (3.14) displays the experimental results.

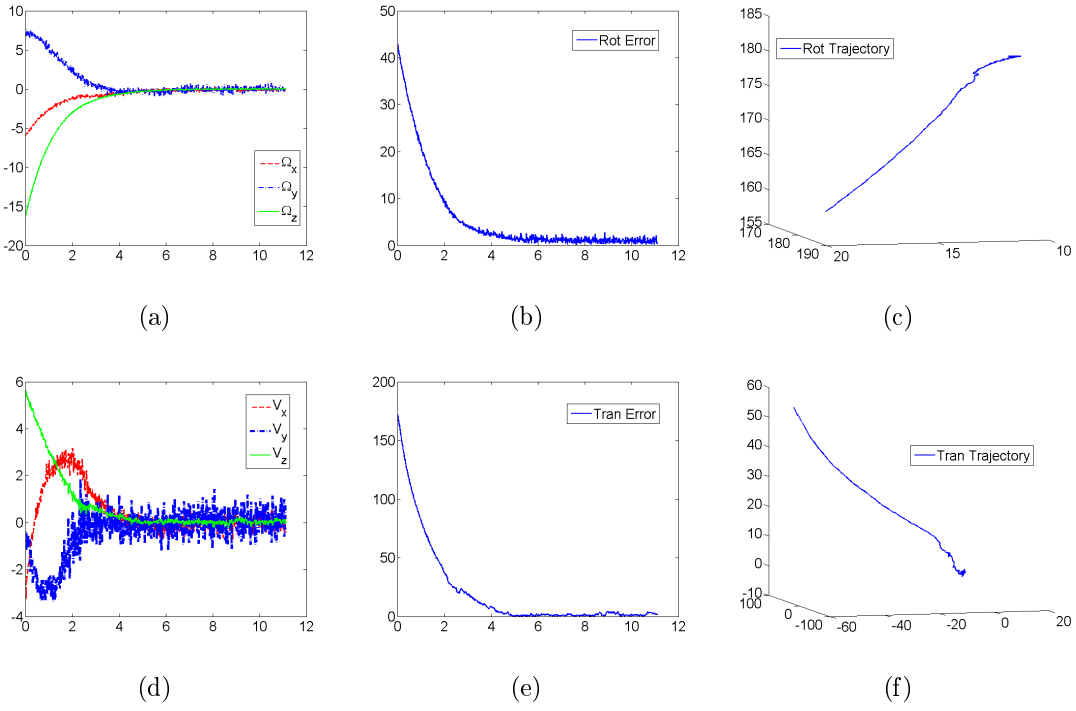


Figure 3.14: *Experimental results for the "Plücker coordinates control laws". The object of interest consists of three 3D straight lines. The horizontal axis of the figures is the time in second. a) Rotational velocity (deg/s). b) Rotation error (deg). c) Rotation trajectory (the angle θ trajectory). d) Translational velocity (mm/s). e) Translation error(mm). f) Translation trajectory (motion of the origin of the stereo frame). The errors (the task functions) are almost null at the end of servoing, which show the asymptotic convergence of state vectors.*

These results are similar to the previous results using "Euclidean control laws". Figures (3.14 - a, d) show the stability of the rotational and the translational velocities. When figures (3.14 - b, e) show the asymptotic convergence of the rotational and translational task functions.

The trajectory of the robot end-effector in the Euclidean space is displayed in figure (3.14 - f). This trajectory is not a straight line, and that is expected as the translational task function is defined in Plücker coordinates space.
Same to the angle of rotation θ, its trajectory is not linear, see figure (3.14 - c).
The rotation velocity (fig. 3.14 - a) is similar to the rotation velocity with the "Euclidean control laws" experiment (fig. 3.9 - a), and that is expected as in the two cases the direction vectors v are used to calculate the task function.

At the end of the servoing, we have about (0.5 deg) in rotation and (5 mm) in translation which is similar to the results in Euclidean control laws experiment.
The perturbation at the translation velocity (fig. 3.14 - b) is more significant than that in the rotation velocity or that in the translation velocity using Euclidean control laws (fig. 3.14 - b), that is expected too, as the distance is the most disturbed for stereo rig.

3.3 The Plücker coordinates and Euclidean control laws

3.3.1 Introduction

In this chapter, two kinds of control laws are designed, they are two pairs of decoupled control laws, for controlling the translation velocity and the rotation velocity. The first is called "Euclidean control laws" given by the equations (3.1, 3.3)

$$\begin{aligned} \mathbf{V} &= -\lambda_V \, \mathtt{R}^{\mathsf{T}} t \\ \mathbf{\Omega} &= -\lambda_\Omega \, \theta u \end{aligned}$$

and the second is called "Plücker coordinates control laws" given by the equations (3.8, 3.6)

$$\begin{aligned} \mathbf{V} &= -\lambda_V \, \mathtt{H}_V^+ e_V \\ \mathbf{\Omega} &= -\lambda_\Omega \, \mathtt{H}_\Omega^+ e_\Omega \end{aligned}$$

They use 3D state vectors in the input, and require a 3D estimation of the current state vectors, so they carry out a static error at the end of servoing.
Every pair of these controls laws have its own drawbacks and advantages, we will explain them in this section.

3.3.2 The final pose

Given an object represented by ${}^{o}v_i, {}^{o}w_i, i = 1..n$ at the desired pose. The question is, does one have the same final pose for the stereo rig what ever the kind of control laws used in servoing?
We saw that, in the case of using "Euclidean control laws", at the end of servoing one has ${}^{o}v_i = {}^{c^*}v_i, {}^{o}w_i = {}^{c^*}w_i, i = 1..n$.
But, in the case of using "Plücker coordinates control laws", at the end of servoing one has ${}^{o}v_i = {}^{c^*}v_i, ||{}^{o}w_i||^2 = d_i^{c^*2}, i = 1..n$. Let's use the notation w_i^∞ for the second vector of Plücker coordinates of the object straight lines in the final pose.
To prove the equivalence between the two final poses, we need to prove that the vectors ${}^{o}w_i$ are equal to $w_i(\infty)$ for $i = 1..n$.

Proof
We have $w_i^{\infty\mathsf{T}} w_i^\infty = {}^{o}w_i^{\mathsf{T}} {}^{o}w_i$ and ${}^{o}v_i = v_i^* = v_i$. We need to prove $w_i^\infty = {}^{o}w_i$.
Suppose that the two final poses are different, which mean the "Plücker coordinates control laws" lead the stereo rig to different pose $c^p \neq c^*$ from that one reached by the "Euclidean control laws". So one has a rotation matrix $\mathtt{R}$ and translation vector t between the two poses.
Recall the equation (3.4) ${}^{o}v_i = \mathtt{R}v_i^*$, but ${}^{o}v_i = v_i^*$, so if the object has two non parallel lines, one gets $\mathtt{R} = \mathtt{I}$.
Now, recall the equation (3.4)

$$\begin{aligned} {}^{o}w_i &= \mathtt{R}w_i^\infty + t \times \mathtt{R}v_i^* \\ {}^{o}w_i &= w_i^\infty + t \times v_i \\ {}^{o}w_i - w_i^\infty &= t \times v_i \end{aligned}$$

we will prove that $t = 0$ so $w_i^\infty = {}^{o}w_i$

$$\begin{aligned} ({}^{o}w_i - w_i^\infty)^{\mathsf{T}}({}^{o}w_i - w_i^\infty) &= ||t \times v_i||^2 \\ {}^{o}w_i^{\mathsf{T}} {}^{o}w_i + w_i^{\infty\mathsf{T}} w_i^\infty - 2{}^{o}w_i^{\mathsf{T}} w_i^\infty &= ||t \times v_i||^2 \end{aligned}$$

writing this equation in two ways:

$$\begin{aligned} 2{}^{o}w_i^{\mathsf{T}} {}^{o}w_i - 2{}^{o}w_i^{\mathsf{T}} w_i^\infty &= ||t \times v_i||^2 \\ 2{}^{o}w_i^{\mathsf{T}}({}^{o}w_i - w_i^\infty) &= ||t \times v_i||^2 \\ 2{}^{o}w_i^{\mathsf{T}}(t \times v_i) &= ||t \times v_i||^2 \\ 2t^{\mathsf{T}}(v_i \times {}^{o}w_i) &= ||t \times v_i||^2 \qquad (1) \end{aligned}$$

and

$$\begin{aligned} 2w_i^{\infty\mathsf{T}}w_i^{\infty} - 2w_i^{\infty\mathsf{T}}{}^{o}w_i &= ||t \times v_i||^2 \\ 2w_i^{\infty\mathsf{T}}(w_i^{\infty} - {}^{o}w_i) &= ||t \times v_i||^2 \\ -2w_i^{\infty\mathsf{T}}(t \times v_i) &= ||t \times v_i||^2 \\ -2t^{\mathsf{T}}(v_i \times w_i^{\infty}) &= ||t \times v_i||^2 \qquad (2) \end{aligned}$$

(1) − (2) $\Longrightarrow$

$$\begin{aligned} 2t^{\mathsf{T}}(v_i \times {}^{o}w_i) + 2t^{\mathsf{T}}(v_i \times w_i^{\infty}) &= 0 \\ t^{\mathsf{T}}(v_i \times {}^{o}w_i + v_i \times w_i^{\infty}) &= 0 \\ t^{\mathsf{T}}(v_i \times ({}^{o}w_i + w_i^{\infty}) &= 0 \end{aligned}$$

there are 4 solutions for the last equation

- ${}^{o}w_i = -w_i^{\infty}$, this is the mirror pose, which means all the lines are behind the vision system, this is impossible practically.
- $v_i \equiv ({}^{o}w_i + w_i^{\infty})$ which is impossible because $v_i \perp {}^{o}w_i$
- $t \perp (v_i \times ({}^{o}w_i + w_i^{\infty}) \ \ \forall i$ that is impossible, if there are three non coplanar vectors ${}^{o}u_i = v_i \times {}^{o}w_i$
- $t = 0$ which means $w_i^{\infty} = {}^{o}w_i$.

So, the two pairs of control laws, guide the stereo rig to the same final pose ($c^p = c^*$).

3.3.3 The advantages and the drawbacks

The two pairs of control laws ("Euclidean control laws" and "Plücker coordinates control laws") are fully decoupled, and verify the global asymptotic convergence of the state vectors in the closed loop.
The Euclidean control laws are general kinematic pose control laws. They do not interest the type of the state vectors, they only need the estimation of the relative rotation matrix $\mathtt{R}$ and the relative translation vector t between the current and the desired poses.
With "Euclidean control laws", the trajectory of the stereo rig frame is linear in the Euclidean space.
The main drawbacks of the Euclidean control laws can be seen in the translation task function $e_V = \mathtt{R}^{\mathsf{T}}t$. The estimation of the translation error is depending on the estimation of the rotation error, which makes the rotation error is accumulated in the translation error.

The "Plücker coordinates control laws" are specified for the Plücker coordinates state vectors. The task functions defined the difference between the current and the desired state vectors.
There are two main drawbacks with the Plücker coordinates control laws. First, using the distance as state vector for translation, causes an important perturbation in the control, because of the big effect of the noise at the distance estimation. Second, the trajectory of the frame origin in the Euclidean space is not a straight line, because the task functions are defined in the Plücker coordinates space.
Finally, the "Plücker coordinates control laws" need only the knowledge of the direction vectors v_i and the distance d_i of the object in the desired pose, while the "Euclidean control laws" need both vectors v_i and w_i.

3.4 Object with two straight lines

The object of interest in this case consists of only two straight lines L_1, L_2, and the goal is to servo the relative pose (object - stereo rig).
As we know (section 3.1.1), two non parallel lines are enough to define the 6 DOF in the Euclidean space, so, the "Euclidean control laws" defined in (3.1, 3.3), are still valid to perform the visual servoing with an object consisting of only two straight lines.
The rotation control law in The "Plücker coordinates control laws" given by the equation (3.6) is defined with two non parallel straight lines, so it is valid in this task too.
The translation control law given by the equation (3.8), needs at least three distances d_i to be defined, but, two non parallel lines are also enough to define the 3 DOF of translation, so they must be enough to define the translation control law.
In this section, we are interested to find the coordinates of the closest two points of these lines. These points can be useful for defining additional state vector for translational visual servoing.
Given two 3D straight lines (L_1, L_2), represented by their normalized Plücker coordinates, we look for the coordinates of the most close points of each line on the other line, see figure (3.15).

Suppose $m_1 \in L_1$ is the closest point of the straight line L_1 to the straight line L_2, and $m_2 \in L_2$ is the closest point of the straight line L_2 to the straight line L_1.
One can define the two points m_1, m_2 using the normalized Plücker coordinates of

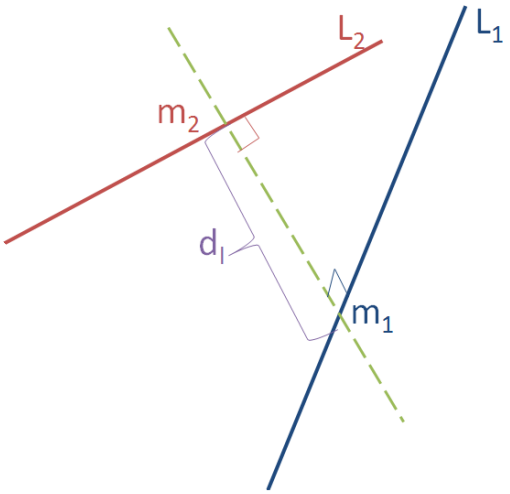


Figure 3.15: The minimal distance d_l between two straight lines L_1, L_2. $m_1 \in L_1, m_2 \in L_2$ are the two closest points.

these two lines as:

$$m_1 \in L_1 \Longrightarrow m_1 = u_1 + \alpha v_1 \quad \text{with} \quad \alpha \in \mathbb{R} \quad , \quad u_1 = v_1 \times w_1$$
$$m_2 \in L_2 \Longrightarrow m_2 = u_2 + \beta v_2 \quad \text{with} \quad \beta \in \mathbb{R} \quad , \quad u_2 = v_2 \times w_2$$

with $u_1 = v_1 \times w_1$ is a known point on L_1, and v_1 is the direction on this line, the same for u_2, v_2.
The parameters (α, β), define a minimal distance d_l between the two straight lines. To compute them, we look for $\min_{\alpha,\beta} ||m_1 - m_2||^2$.
Let $f(\alpha, \beta) = \frac{1}{2}||m_1 - m_2||^2$. Then $\frac{\partial f}{\partial \alpha} = 0$, and $\frac{\partial f}{\partial \beta} = 0$.

$$\begin{aligned} f(\alpha, \beta) &= \frac{1}{2}||u_1 - u_2 + \alpha v_1 - \beta v_2||^2 \\ \frac{\partial f}{\partial \alpha} &= v_1^{\mathsf{T}}(u_1 - u_2 + \alpha v_1 - \beta v_2) \\ \frac{\partial f}{\partial \beta} &= -v_2^{\mathsf{T}}(u_1 - u_2 + \alpha v_1 - \beta v_2) \end{aligned}$$

with the notation $\gamma = v_1^\top v_2$

$$\begin{aligned} & v_1^\top u_1 - v_1^\top u_2 + \alpha - \beta v_1^\top v_2 = 0 \quad \text{that is} \\ & \alpha - \gamma\beta = v_1^\top u_2 \end{aligned} \tag{1}$$

$$\begin{aligned} & -v_2^\top u_1 + v_2^\top u_2 - \alpha v_2^\top v_1 + \beta = 0 \quad \text{that is} \\ & \beta - \gamma\alpha = v_1^\top u_2 \end{aligned} \tag{2}$$

$$\begin{aligned} (1) + \gamma * (2) \Longrightarrow \alpha(1-\gamma^2) &= v_1^\top u_2 + \gamma v_2^\top u_1 \\ (2) + \gamma * (1) \Longrightarrow \beta(1-\gamma^2) &= v_2^\top u_1 + \gamma v_1^\top u_2 \end{aligned}$$

$$\Longrightarrow \alpha = \frac{v_1^\top u_2 + \gamma v_2^\top u_1}{1-\gamma^2} \quad \text{and} \quad \beta = \frac{v_2^\top u_1 + \gamma v_1^\top u_2}{1-\gamma^2}$$

It is easy to verify that ($\frac{\partial^2 f}{\partial \alpha^2} = 1 > 0$) and ($\frac{\partial^2 f}{\partial \beta^2} = 1 > 0$), then this extrema is a local minimum.
Then, the two closest points of the two straight lines are

$$m_1 = u_1 + \frac{1}{1-\gamma^2}(v_1^\top u_2 v_1 + \gamma v_2^\top u_1 v_1) \tag{3.9}$$

$$m_2 = u_2 + \frac{1}{1-\gamma^2}(v_2^\top u_1 v_2 + \gamma v_1^\top u_2 v_2) \tag{3.10}$$

If $\gamma = 1$, then $v_1^\top v_2 = 1$, which means ($v_1 = v_2$) the two straight lines are parallel. In this case we get $m_1 = u_1$ and $m_2 = u_2$.
Finally, we can note that the Euclidean distance between L_1, L_2 is given by

$$d_l = ||m_1 - m_2|. \tag{3.11}$$

Now, a third (virtual) line L_3 is computed using the points m_1, m_2. Even if these points are not real in the scene, and that there is no way to detect them in the images, they can be computed thanks to the 3D reconstruction of the normalized Plücker coordinates of the two first lines.
Then, the virtual third line L_3 can be defined as follows:

$$v_3 = \frac{m_1 - m_2}{||(m_1 - m_2)||}$$

$$w_3 = m_1 \times v_3$$

If $m_1 = m_2$ then L_3 is defined by $v_3 = v_1 \times v_2$ and $w_3 = m_1 \times v_3$.
Finally, this third line can be included in the task function e_V and in the translation control law $\mathbf{V}$.

3.5 One-line servoing

Considering the 3D straight line as the primitive unit of the articulated object, we are interested of the servoing of the relative pose between the stereo rig and the object represented by one straight line.
Since the geometry of a straight line is defined with 4 free parameters (4 DOF in the 3D Euclidean space). Then, the control laws defined above are not valid to serve the pose of this straight line.
The pose servoing of one 3D straight line is important too, because many objects (or object axis) can be represented by one straight line like pens, and the right circular cylinders [DdM07].
To look for translational and rotational control laws, by projecting the equations (2.5, 2.6) to the basis $(v, w, u = v \times w, ||v|| = 1)$ one obtains:

$$\dot{v}^{\mathsf{T}} v = 0 \tag{3.12}$$
$$\dot{v}^{\mathsf{T}} w = u^{\mathsf{T}} \mathbf{\Omega} \tag{3.13}$$
$$\dot{v}^{\mathsf{T}} u = -w^{\mathsf{T}} \mathbf{\Omega} \tag{3.14}$$
$$\dot{w}^{\mathsf{T}} v = -u^{\mathsf{T}} \mathbf{\Omega} \tag{3.15}$$
$$\dot{w}^{\mathsf{T}} w = u^{\mathsf{T}} \mathbf{V} \tag{3.16}$$
$$\dot{w}^{\mathsf{T}} u = ||w||^2 v^{\mathsf{T}} \mathbf{\Omega} - w^{\mathsf{T}} \mathbf{V} \tag{3.17}$$

Equation (3.12) is useless and it considers the constraint $||v|| = 1$. The equations (3.13), (3.15) are equivalent as they consider the constraint $v^{\mathsf{T}} w = 0$.
So we have four equations which represent the kinematic of the 4 DOF of a 3D straight line in the normalized Plücker coordinates basis (v, w, u).
In these equations one notes that the rotation velocity $\mathbf{\Omega}$ can be observed on the axis v by the equation (3.17), the axis w by the equation (3.14), and the axis u by the equation (3.15). However the translation velocity $\mathbf{V}$ can be observed on the axis w, u only by the equations (3.17, 3.16), and it is impossible to observe the translation motion on the axis v.
That is why the projection of the translation velocity on the direction axis will consider null $\mathbf{V}^{\mathsf{T}} v = 0$, and it may be controlled by additional translation control law if necessary.
Now, still having 5 components of the kinematic screw and 4 DOF for the motion of a straight line. We have the possibility to add one constraint to design the velocity control law.
This constraint can be useful to eliminate the coupling between the rotation velocity and the translation velocity in the equation (3.17), for example

- $w^{\mathsf{T}} \mathbf{V} = 0$: with this constraint, the 4 DOF of the straight line are divided in

3 DOF for the rotation motion given by $\dot{v}^\mathsf{T} u = -w^\mathsf{T}\mathbf{\Omega}$, $\dot{w}^\mathsf{T} v = u^\mathsf{T}\mathbf{\Omega}$, $\dot{w}^\mathsf{T} u = ||w||^2 v^\mathsf{T}\mathbf{\Omega}$, and 1 DOF for the translation given in the equation $\dot{w}^\mathsf{T} w = u^\mathsf{T}\mathbf{V}$, which is related to the orthogonal distance d of the 3D straight line with respect to the frame origin.

- $v^\mathsf{T}\mathbf{\Omega} = 0$: the equation (3.17) becomes $\dot{w}^\mathsf{T} u = -w^\mathsf{T}\mathbf{V}$. The 4 DOF of the straight line are divided in 2 DOF for the rotation motion and 2 DOF for the translation motion.

3.5.1 Control laws with 3 DOF for rotation and 1 DOF for translation

In this case, we directly refereed to use the bi-normalized Plücker coordinates [AEH02], (v, w_n, d) for the line in the current pose, and v^*, w_n^*, d^* in the desired pose.
To servo the distance d, we define the task function $e_V = (d - d^*)$. Then the translation velocity control law is

$$\mathbf{V} = -\lambda_V (d - d^*) u. \tag{3.18}$$

Note: $u = v \times w_n$, $||u|| = 1$.
To prove the stability of this control law and the asymptotic convergence of the task function, let's use the Lyapunov function $\mathcal{L}(t) = ||e_V(t)||^2$

$$\begin{aligned}
\dot{\mathcal{L}}(t) &= \dot{e}_V^\mathsf{T} e_V = \dot{d}(d - d^*) \\
&= \mathbf{V}^\mathsf{T} u (d - d^*) \qquad \text{as} \qquad \dot{d} = \mathbf{V}^\mathsf{T} u \ \text{(equation (2.7))} \\
&= -\lambda_V (d - d^*) u^\mathsf{T} u (d - d^*) \\
&= -\lambda_V (d - d^*)^2 \qquad u^\mathsf{T} u = 1 \\
\dot{\mathcal{L}}(t) &< 0.
\end{aligned}$$

So, the translation velocity control law $\mathbf{V} = -\lambda_V (d - d^*) u$ globally stabilizes the system in the closed loop, and the distance d converges asymptotically towards d^*.
Back to the equation (2.7) [AEH02], the part related to w_n has its kinematic given by

$$\begin{aligned}
\dot{w}_n &= -\frac{1}{d}(v \times w_n) w_n^\mathsf{T} \mathbf{V} - w_n \times \mathbf{\Omega} \\
\dot{w}_n &= \mathbf{\Omega} \times w_n - \frac{1}{d} u (w_n^\mathsf{T} \mathbf{V}) \\
\dot{w}_n &= \mathbf{\Omega} \times w_n + \frac{\lambda_V (d - d^*)}{d} u (w_n^\mathsf{T} u) \qquad w_n^\mathsf{T} u = 0 \\
\dot{w}_n &= \mathbf{\Omega} \times w_n.
\end{aligned}$$

This means that the kinematic of the vector w_n is like the kinematic of the direction vector v.
These two vectors (v, w_n) are enough to define the rotational control law designed in section (3.2.1), $\mathbf{\Omega} = -\lambda_\Omega \mathtt{H}_\Omega^+ e_\Omega$ which is asymptotically stable. The rotational task function e_Ω and the interaction matrix $\mathtt{H}_\Omega$ are given by

$$e_\Omega = \begin{pmatrix} v - v^* \\ w_n - w_n^* \end{pmatrix} \qquad \mathtt{H}_\Omega = \begin{bmatrix} -[v]_\times \\ -[w_n]_\times \end{bmatrix}$$

With this proposition of the state vectors, the full decoupling is performed between $\mathbf{\Omega}$ and $\mathbf{V}$. At the end of the servoing we get $d = d^*$ using the translational control law, and $v = v^*, w_n = w_n^*$ using the rotational control law.
This pair of control laws will be named "3R1t".

3.5.2 Simulation results

The goal of these simulations is to verify the asymptotic convergence of the state vectors (v, w, d), towards their desired values (v^*, w^*, d^*). The translation and rotation velocities used in the servoing are:

$$\begin{aligned} \mathbf{\Omega} &= -\lambda_\Omega \mathtt{H}_\Omega^+ e_\Omega \\ \mathbf{V} &= -\lambda_V (d - d^*) u \end{aligned}$$

Many simulations have been done, with different configurations for the stereo rig, and different configurations of the object of interest in the initial and desired position. But as the results are always the same, we present here one simulation for configuration similar to the practical tests, and repeat it with noise.

Like what we proposed in the simulations in the previous chapter (3), the robot is considered as a perfect integrator.
The stereo system consists of two parallel identical cameras. The homogeneous matrices of transformation between the two cameras and the stereo rig frame are $\mathtt{R}_r = \mathtt{R}_l = \mathtt{I}, t_r = -t_l = (0.1, 0, 0)^\mathsf{T}$.
The intrinsic parameters of the two cameras used in these simulations are

$$\mathtt{K_r} = \mathtt{K_l} = \begin{bmatrix} 1200 & 0 & 300 \\ 0 & 1200 & 200 \\ 0 & 0 & 1 \end{bmatrix}$$

These matrices are supposed to be exact if nothing else proposed (the stereo system is perfect calibrated).

The object of interest consists of a 3D straight line. The current relative pose between the object and the stereo rig is represented by the bi-normalized Plücker coordinates of the 3D straight line (v, w, d). The desired pose is defined by the desired bi-normalized Plücker coordinates of the 3D straight line (v^*, w^*, d^*).
The closed loop gains are set to $\lambda_\Omega = \lambda_V = 0.15$.
As the rotational task function does not have unit, we will illustrate the angle θ in the axis/angle representation of the rotation matrix to display the rotation error (just like the task function of the "Euclidean control laws").

Figure (3.16) shows the simulation results of the visual servoing for one-line object.

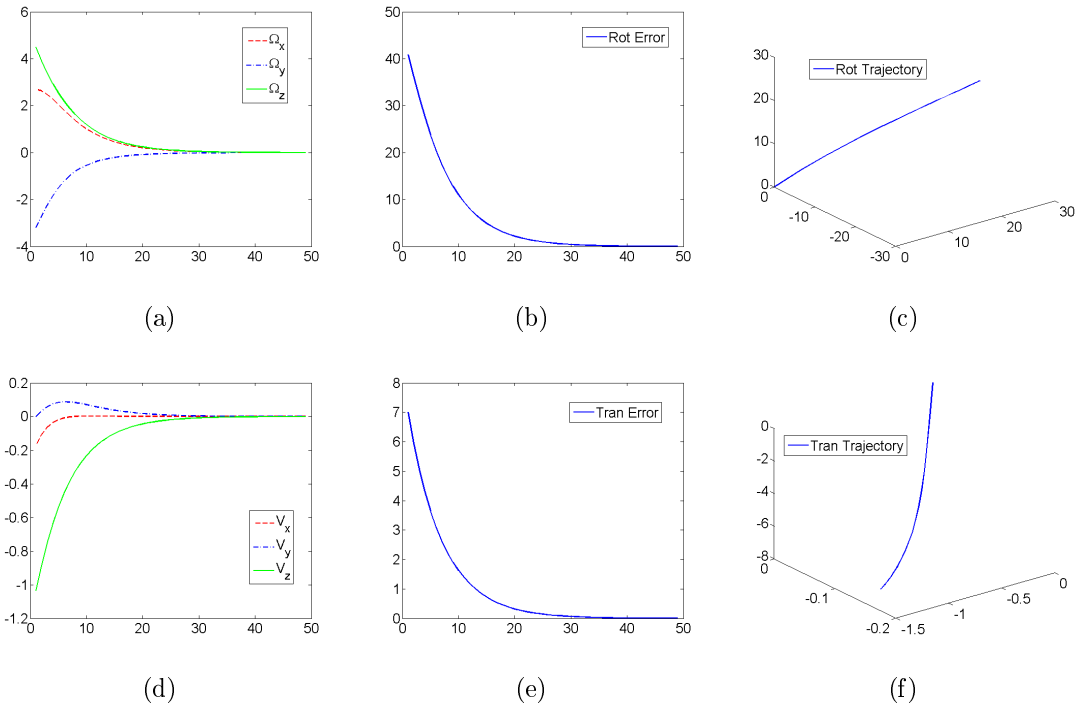


Figure 3.16: *Simulation results for the PBVS of one-line object using* 3 *DOF for rotation and* 1 *DOF for translation. The object of interest consists of a 3D straight line. The horizontal axis of the figures is the iteration number. a) Rotation velocity (deg/s). b) Rotation error θ (deg). c) Rotation trajectory. d) Translation velocity (m/s). e) Translation error (translational task function e_V) (m). f) Translation trajectory. The errors (the task functions) are null at the end of servoing, which show the asymptotic convergence of state vectors.*

The simulation results in this case are similar to those in the general case (with 6 DOF and object with more than one line).

One can see the stability of the velocities control laws in the closed loop (fig. 3.16- a, d), and the asymptotic convergence of the task functions (fig. 3.16- b, e).
The trajectory of the stereo rig frame is shown in (fig. 3.16- c, f).

This simulation is repeated after adding a white noise to the image lines. Figure (3.17) shows the results.

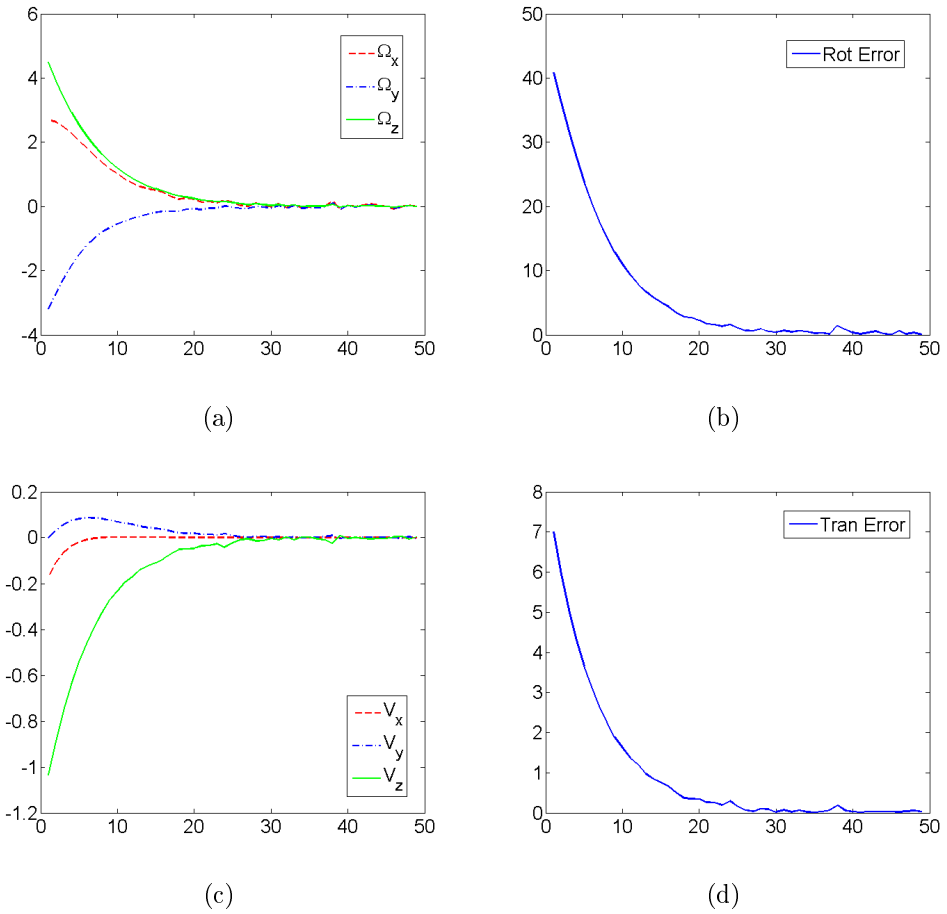


Figure 3.17: *Simulation results for the PBVS of one-line object using* 3 *DOF for rotation and* 1 *DOF for translation, after adding a white noise to the image lines. The object of interest consists of a 3D straight line. The horizontal axis of the figures is the iteration number. a) Rotation velocity (deg/s). b) Rotation error* θ *(deg). c) Translation velocity (m/s). d) Translation error (translational task function* e_V*) (m). The errors (the task functions) are almost null at the end of servoing, which show the asymptotic convergence of state vectors.*

The effect of the noise on the stability of the system in the closed loop, and the asymptotic convergence of the task functions is very small. Figures (3.17 - a, c) show the velocities in rotation and translation. They are decreasing to zero although the presence of the noise in the images.

One can note an error of about 0.5 deg in rotational task function and an error of about 1 cm in the translation task function, see figures (3.17 - b, d).

3.5.3 Experimental results

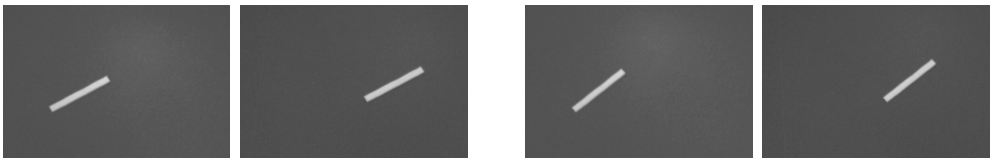
The difference between this case and the other experiments in the equipment is only the object of interest, as the same robot and the same stereo rig are used.
The extrinsic stereo matrices are given by

$$\mathtt{R}_r = \mathtt{R}_l = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad , \quad t_r = -t_l = \begin{pmatrix} 0.1 \\ 0 \\ 0 \end{pmatrix}$$

The intrinsic matrices are defined using the cameras technical specifications (focal length, sensor length, and image size), we have

$$\mathtt{K}_r = \mathtt{K}_l = \begin{bmatrix} 1200 & 0 & 300 \\ 0 & 1200 & 200 \\ 0 & 0 & 1 \end{bmatrix}$$

The current and the desired poses are defined by the bi-normalized Plücker coordinates of the 3D straight line (the object of interest), using the stereo images of this object in the current and desired poses, see figure (3.18).



(a) (b) (c) (d)

Figure 3.18: The stereo image of the object of interest (which consists of one 3D straight line). a) Right image in the initial pose. b) Let image in the initial pose. c) Right image in the final pose. d) Left image in the final pose.

The object of interest is a white rectangle, and the background is black. We use a threshold filter to separate the object pixels in the image and estimate the image line parameters (image binarization).
Figure (3.19) shows the stability of the control closed loop, and the asymptotic convergence of the task functions.

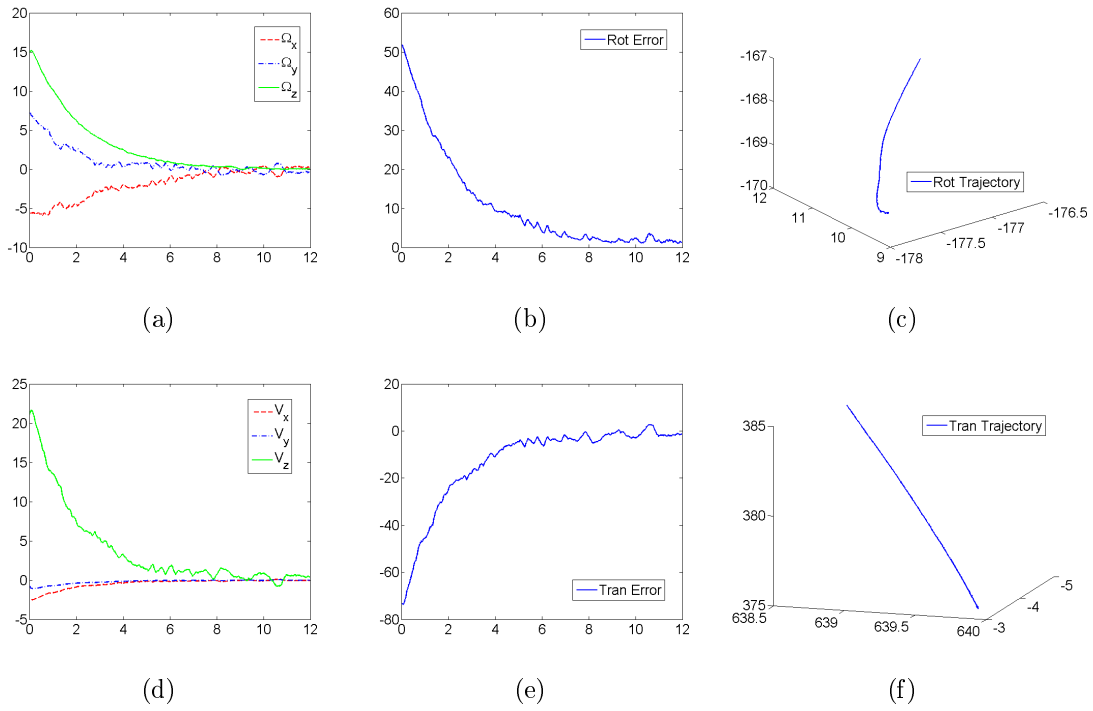


Figure 3.19: *Experiment results for the PBVS of one-line object using* 3 *DOF for rotation and* 1 *DOF for translation. The object of interest consists of a 3D straight line. The horizontal axis of the figures is the time in second. a) Rotation velocity (deg/s). b) Rotation error θ (deg). c) Rotation trajectory. d) Translation velocity (mm/s). e) Translation error (translational task function e_V) (mm). f) Translation trajectory. The errors (the task functions) are almost null at the end of servoing, which show the asymptotic convergence of state vectors.*

The disturbance is very significant in the behaviour of the velocity control laws (fig. 3.19 - a, b). The task functions errors (fig. 3.19 - c, d) are not negligible (about 1 deg in rotation and 2 mm in translation).
This is due to the absence of redundancy in state vectors, as the object of interest consists of only one 3D straight line, which makes the signal-to-noise ratio (S/N) low. Nevertheless, we get the convergence of the state vectors toward their desired values.

3.5.4 Control laws with 2 DOF for rotation and 2 DOF for translation

In this case, the proposed constraints are $v^{\mathsf{T}}\mathbf{\Omega} = 0$, and $v^{\mathsf{T}}\mathbf{V} = 0$. The 3D straight line is represented by normalized Plücker coordinates (v, w) in the current pose, and (v^*, w^*) in the final pose.
Starting with the equation (2.5) $\dot{v} = \mathbf{\Omega} \times v$, so

$$\begin{aligned} v \times \dot{v} &= v \times (\mathbf{\Omega} \times v) \\ v \times \dot{v} &= \mathbf{\Omega} - (v^{\mathsf{T}}\mathbf{\Omega})v \qquad \text{but} \qquad v^{\mathsf{T}}\mathbf{\Omega} = 0 \\ \mathbf{\Omega} &= v \times \dot{v} \end{aligned}$$

With a rotational task function $e_\Omega = v - v^*$ and an exponential decrease suppose of the form $\dot{e}_\Omega = -\lambda_\Omega e_\Omega$ we get the rotation velocity control law

$$\mathbf{\Omega} = \lambda_\Omega \; v \times v^* \tag{3.19}$$

For the translational velocity control law, recall the equation (2.6)

$$\begin{aligned} \dot{w} &= \mathbf{\Omega} \times w + \mathbf{V} \times v \\ v \times \dot{w} &= v \times (\mathbf{\Omega} \times w) + v \times (\mathbf{V} \times v) \\ v \times \dot{w} &= (v^{\mathsf{T}}w)\mathbf{\Omega} - (v^{\mathsf{T}}\mathbf{\Omega})w + \mathbf{V} - v^{\mathsf{T}}\mathbf{V} \\ \mathbf{V} &= v \times \dot{w} \end{aligned}$$

For the task function, chosen $e_V = w - w^*$ and exponential decrease ($\dot{e}_V = -\lambda_V e_V$), the translational velocity control law is designed by

$$\begin{aligned} \dot{e_V} &= \dot{w} = -\lambda_V e_V \\ \dot{w} &= -\lambda_V (w - w^*) \\ v \times \dot{w} &= -\lambda_V v \times (w - w^*) \end{aligned}$$

Then,

$$\mathbf{V} = -\lambda_V \; v \times (w - w^*) \tag{3.20}$$

These control laws are defined in [AEH02] using the bi-normalized Plücker coordinates and monocular vision.
$\mathbf{\Omega}$ and $\mathbf{V}$ verify the global asymptotic stability of the system in closed loop, but they are not fully decoupled, so, one can use them in a sequential control.
This pair of control laws will be named "2R2t".

3.5.5 Simulation results

In this simulation, the visual servoing is divided in two phases. In the first phase, only the rotational control law $\mathbf{\Omega} = \lambda_\Omega v \times v^*$ is used in the closed loop.

The second phase starts after we have the convergence of the vector v toward v^*. In the second phase, the translational control law $\mathbf{V} = -\lambda_V\ v \times (w - w^*)$ is applied to the closed loop to verify the convergence of the vector w toward w^*. We will repeat the previous simulation using the "2R2t" control laws to compare the results with the simulation using "3R1t" control laws.

The object of interest consists of a 3D straight line. The current relative pose between the object and the stereo rig is represented by the normalized Plücker coordinates of the 3D straight line (v, w). The desired pose is defined by the desired normalized Plücker coordinates of the 3D straight line (v^*, w^*).
The closed loop gains are set to $\lambda_\Omega = \lambda_V = 0.15$.

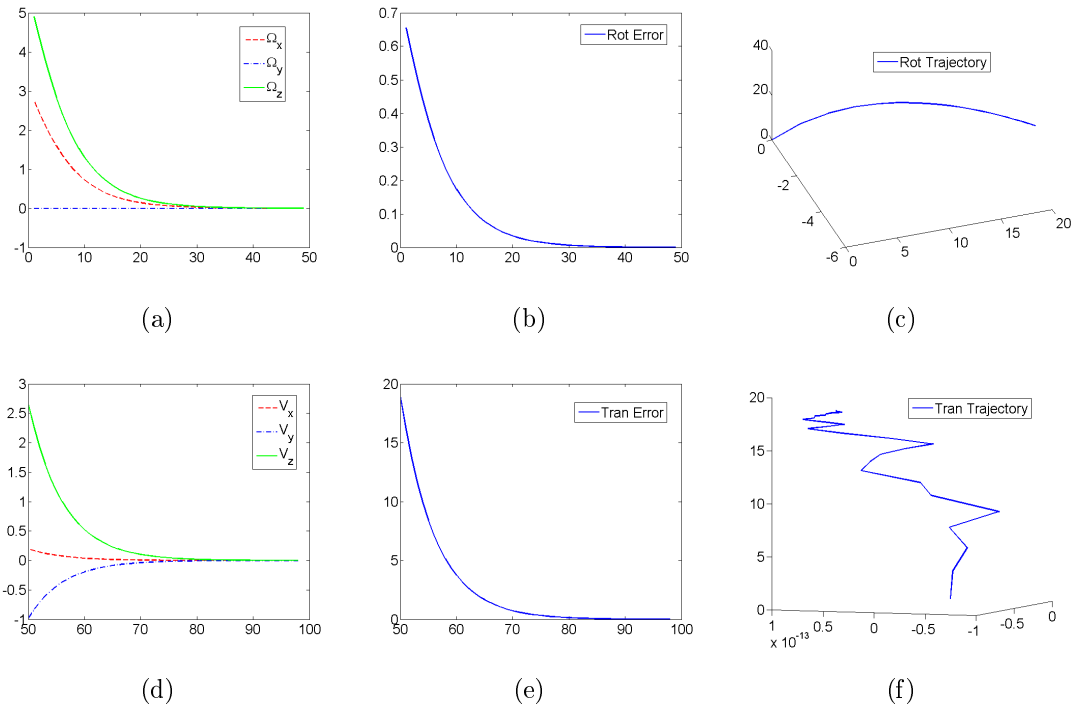
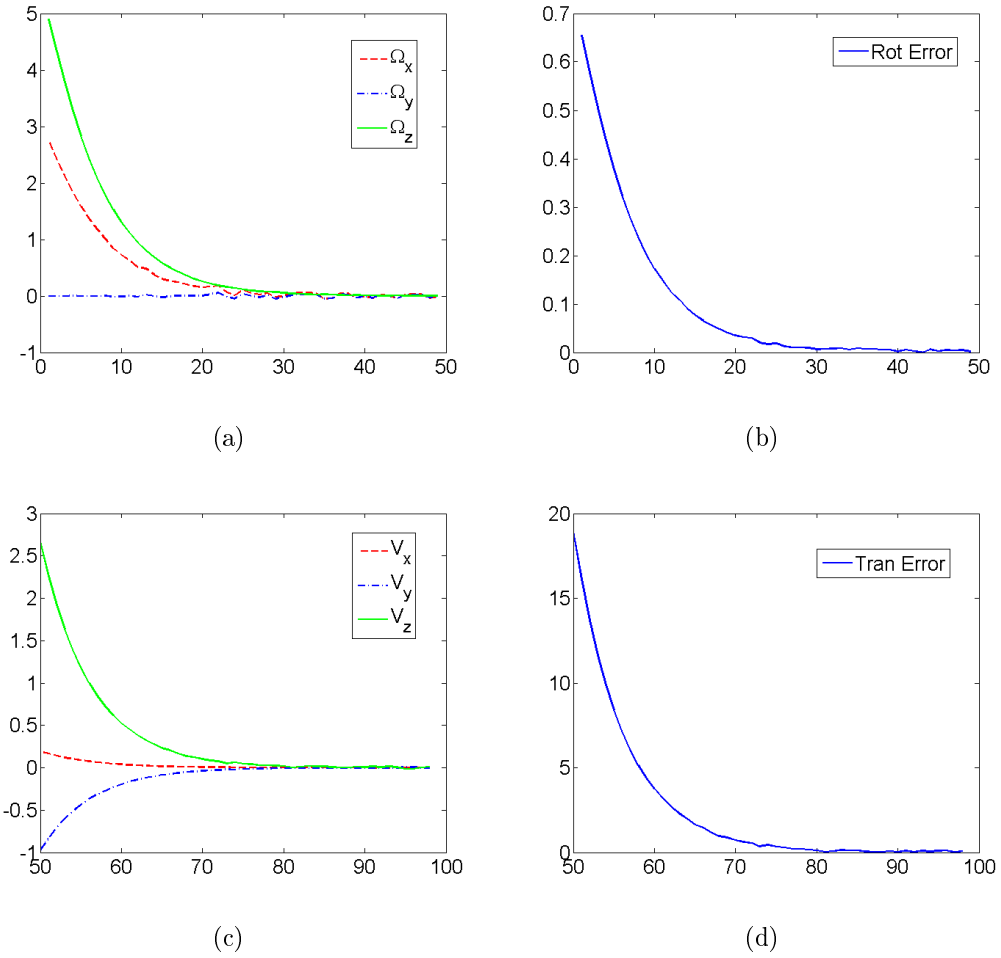


Figure 3.20: *Simulation results for the PBVS of one-line object using 2 DOF for rotation and 2 DOF for translation. The object of interest consists of a 3D straight line. The horizontal axis of the figures is the iteration number. a) Rotation velocity (deg/s). b) Rotation error (rotational task functione_Ω) (no unit). c) Rotation trajectory. d) Translation velocity (m/s). e) Translation error (translational task function e_V) (no unit). f) Translation trajectory. The errors (the task functions) are null at the end of servoing, which show the asymptotic convergence of state vectors.*

Figure (3.20) shows the simulation results of the visual servoing for one-line object. The simulation results in this case are similar to those in the general case (with 6

DOF and object with more than one line).
One can see the stability of the velocities control laws in the closed loop (fig. 3.20- a, d), and the asymptotic convergence of the task functions (fig. 3.20- b, e), which means, at the end of servoing we have $v = v^*$ and $w = w^*$. The trajectory of the stereo rig frame is shown in (fig. 3.20- c, f).

As the servoing of one 3D straight line controls only 4 DOF in the Euclidean space, it is expected that the two pairs of control laws ("3R1t", "2R2t") define two different trajectory in the Euclidean space, which can be seen in (3.20- c, f) and (3.16- c, f).
This simulation is repeated after adding a white noise to the image lines, the noise is generated by adding a random value in the range $[-1, +1]$ pixel to every point in the line image, then estimate the line parameter with the least square minimization. Figure (3.21) shows the results.
The effect of the noise on the stability of the system in the closed loop, and the asymptotic convergence of the task functions is very small. Figures (3.21 - a, c) show the velocities control laws. They are decreasing to zero although the presence of the noise in the images.



(a) (b)

(c) (d)

Figure 3.21: *Simulation results for the PBVS of one-line object using 2 DOF for rotation and 2 DOF for translation, after adding a white noise to the image lines. The object of interest consists of a 3D straight line. The horizontal axis of the figures is the iteration number. a) Rotation velocity (deg/s). b) Rotation error (rotational task functione_Ω) (no unit). c) Translation velocity (m/s). d) Translation error (translational task function e_V) (no unit). The errors (the task functions) are almost null at the end of servoing, which show the asymptotic convergence of state vectors.*

3.6 Articulated object

In this section, we address the servoing from an articulated object which consists of rigid parts and motorized revolute joints only. As many kinds of serial robot manipulator consists of rigid parts and revolute joints.
The goal of the PBVS in this section is to control the relative pose between the stereo rig and the articulated object, and to control the joints in the object simultaneously.
The articulated object is represented by a sequence of n 3D straight lines $L_1, ..., L_n$, between every two successive parts L_i, L_{i+1}, there is a revolute joint represented by its angle ψ_i.
The relative current and desired poses (Articulated object - Stereo rig) are represented by a set of normalized Plücker coordinates (v_i, w_i) for $i = 1..n$.
Figure (3.22) shows a simple representation of an articulated object with n parts and $n-1$ joints.

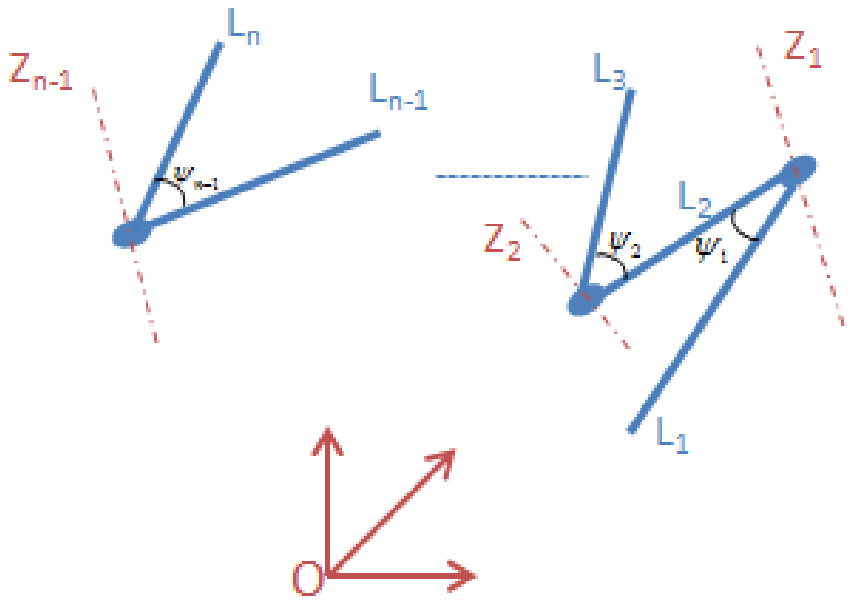


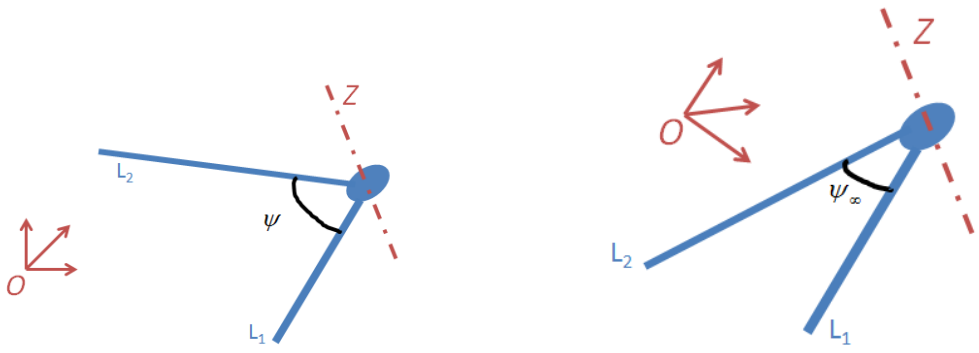
Figure 3.22: An articulated object which consists n 3D straight lines $L_1, ..., L_n$, and $n-1$ revolute joints ψ_i. The axes Z_i represents the rotation axes.

3.6.1 Object with one revolute joint

With two non parallel straight lines, the 6 DOF in the Euclidean space can be constrained. Every straight line is defined with 4 DOF in the Euclidean space, so if the two straight lines are non rigid (form an articulated object), more than 6 DOF can be controlled.
The object of interest is represented by two 3D straight lines L_1, L_2, and a revolute joint ψ between them. We suppose that this joint is motorized. The angular velocity of the joint is $\mathbf{\Omega}_\psi$.

The goal is to control the 6 DOF of the relative pose (Articulated object - Stereo rig), and the 1 DOF of the joint ψ simultaneously, so 7 DOF in total.
We suppose the stereo rig is embedded on a 6 DOF robot end-effector. The part L_1 of the articulated object is fix in the object frame, and the part L_2 is provided with a rotational velocity around the axis Z, which is perpendicular to the two 3D straight lines L_1, L_2.

Figure (3.23) shows such an articulated object, which consists of two parts L_1, L_2, and a revolute joint between them. Figure (3.23-a) shows the initial pose of the object in the stereo rig frame, and the initial value of the joint ψ. Figure (3.23-b) shows the same object at the desired pose, one can see the change in the angle ψ at the end of servoing.



(a) (b)

Figure 3.23: The initial and the final relative pose between the object and the stereo rig, with the angle ψ between the two parts. a) The initial pose and initial value of the angle. b) The final pose and final value of the angle.

Let v_1, w_1, v_2, w_2 be the normalized Plücker coordinates of L_1, L_2 in the stereo rig frame, then

$$Z = \frac{v_1 \times v_2}{||v_1 \times v_2||} \qquad \text{rotation axis between } L_1 \text{ and } L_2$$

$$\psi = \arccos(v_1^{\mathsf{T}} v_2) \qquad \psi \in]-\pi/2 \,..\, \pi/2[\qquad \text{the angle between } L_1 \text{ and } L_2$$

The Plücker coordinates of L_1, L_2 in the desired pose are ${}^o v_1, {}^o w_1, {}^o v_2, {}^o w_2$, in the object frame. This immediately defines the final value of ψ_∞ by $\psi_\infty = \arccos({}^o v_1{}^{\mathsf{T}} {}^o v_2)$. The value of the angle ψ is independent of the reference frame used to represent v, w. The angular velocity control law $\mathbf{\Omega}_\psi$ of the joint angle ψ is a scalar (1 DOF), so it is independent from the reference frame too.
The possibility of measuring the angle ψ during the servoing, defines the joint task function by:

$$e_\psi = \psi - \psi_\infty. \tag{3.21}$$

Then, with an exponential decrease of the task function($\dot{e}_\psi = -\lambda_\psi e_\psi, \ \lambda_\psi \in \mathbb{R}^+$), the angular velocity of the angle ψ is $\mathbf{\Omega}_\psi = \dot{\psi}$, given by:

$$\mathbf{\Omega}_\psi = -\lambda_\psi e_\psi \tag{3.22}$$

The proof of the stability and the asymptotic convergence of the joint angle ψ toward the final value ψ_∞ is very simple:

$$\begin{aligned} \mathcal{L}(t) &= 1/2||e_\psi(t)||^2 \\ \dot{\mathcal{L}}(t) &= \dot{e}_\psi e_\psi \\ &= \mathbf{\Omega}_\psi e_\psi \\ &= -\lambda_\psi e_\psi e_\psi \\ &< 0. \end{aligned}$$

Now, for controlling the relative pose (Articulated object - Stereo rig). The vectors ${}^o v_1, {}^o w_1$ considered the direct images of v_1, w_1 by Euclidean transformation $[\mathtt{R}, t]$ between the current and the desired pose, while the vectors ${}^o v_2, {}^o w_2$ are not the direct images of v_2, w_2 with the same Euclidean transformation $[\mathtt{R}, t]$ until $\psi = \psi_\infty$.
As the angle ψ between v_1 and v_2 is observable, one does not need to wait for the end of the angular servoing to start the pose servoing (cascaded control). Considering figure (3.22) we know that ${}^o v_2, {}^o w_2$ are the images of v_2', w_2' by the Euclidean transformation $[\mathtt{R}, t]$, which in turn are the images of v_2, w_2 by a simple rotation $\varphi = \psi_\infty - \psi$ around the axis Z.
The vectors v_2', w_2' can be estimated using Rodrigues' rotation formula as follows:

$$\begin{aligned} v_2' &= v_2 \cos(\varphi) + (Z \times v_2)\sin(\varphi) + (Z^{\mathsf{T}} v_2) Z (1 - \cos(\varphi)) \\ w_2' &= w_2 \cos(\varphi) + (Z \times w_2)\sin(\varphi) + (Z^{\mathsf{T}} w_2) Z (1 - \cos(\varphi)) \end{aligned}$$

then the two vectors sets ${}^{o}v_1, {}^{o}w_1, {}^{o}v_2, {}^{o}w_2$ and v_1, w_1, v_2', w_2' define two rigid non parallel 3D straight lines in the current and desired pose, which allows to involve, either the "Plücker coordinates control laws"

$$\begin{aligned} \mathbf{\Omega} &= -\lambda_\Omega \mathtt{H}_\Omega^+ e_\Omega \\ \mathbf{V} &= -\lambda_V \mathtt{H}_V^+ e_V \end{aligned}$$

or the "Euclidean control laws"

$$\begin{aligned} \mathbf{\Omega} &= -\lambda_\Omega \theta u \\ \mathbf{V} &= -\lambda_V \mathtt{R}^\mathsf{T} t \end{aligned}$$

Figure (3.24) shows a block diagram of the closed loop for the articulated object control.

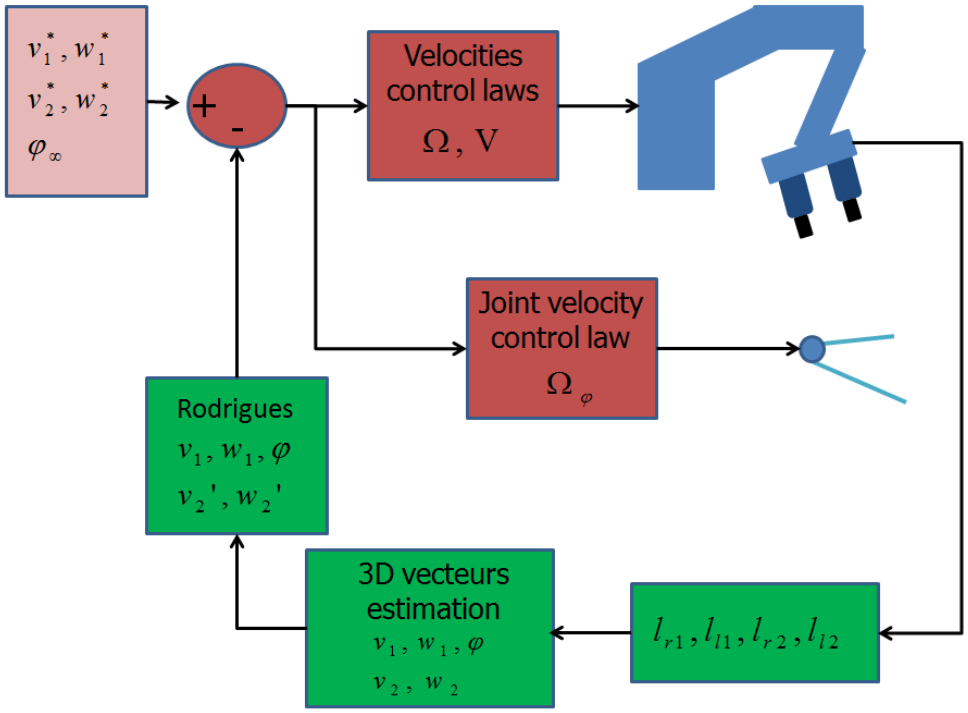


Figure 3.24: The block diagram of the closed loop for the articulated object control

3.6.2 Simulation results of PBVS for an articulated object

Like the previous simulations, the robot is considered as a perfect integrator. The stereo system consists of two parallel identical cameras embedded on the robot end-effector.

We consider that the object of interest consists of two 3D straight lines linked with a revolute joint between them. The current pose is represented by (v_1, w_1, v_2, w_2), so the current angle ψ is given by, $\psi = \arccos(v_1^{\mathsf{T}} v_2)$. The desired pose is represented by $({}^o v_1, {}^o w_1, {}^o v_2, {}^o w_2)$, the final angle is given by $\psi_\infty = \arccos({}^o v_1{}^{\mathsf{T}} {}^o v_2)$.

The goal of this simulation is to show the stability and the asymptotic convergence of all the velocities control laws designed in the previous section.

In this simulation, the "Euclidean control laws" are controlling the velocities of the 6 DOF of the robot end-effector (so the stereo rig). The angular velocity $\mathbf{\Omega}_\psi$ defined in (3.22) is controlling the joint angle ψ.

Figures (3.25 - a, b) show the behaviour of the rotational and translational velocities control laws, one can see that the system (robot end-effector) has the same behaviour in the case of articulated object and the case of rigid object.

The asymptotic convergence of the pose task functions is shown in figures (3.25 - c, d).

Figures (3.25 - e, f), show the angular velocity and the joint angle value ψ during the servoing.

With this simulation, one can see that the possibility of the 3D reconstruction provided by the stereo vision, allows the servoing of a non rigid object to be similar to that of a rigid object, because it is possible all the time to evaluate the changes in the object.

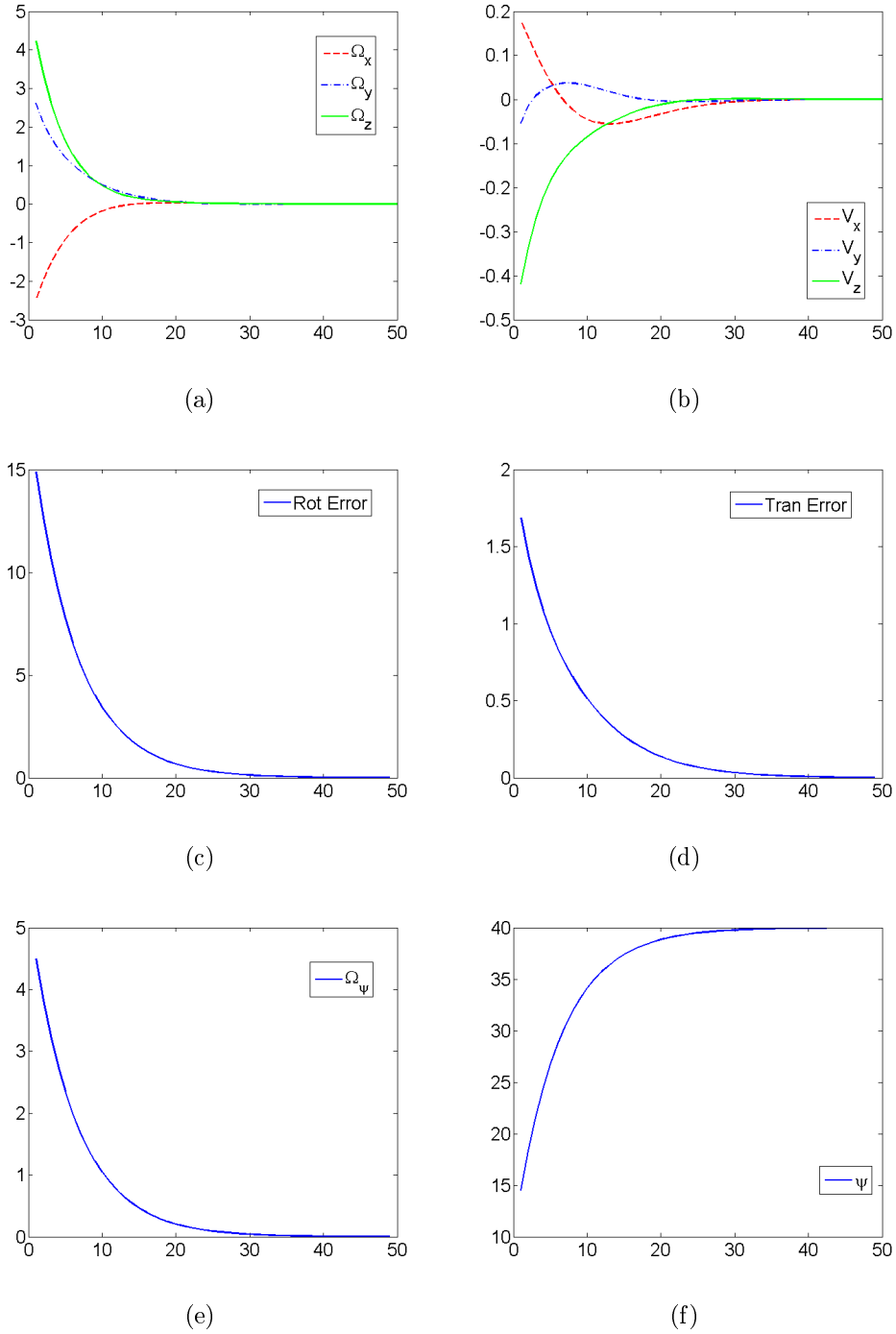


Figure 3.25: *Simulation results for the PBVS of an articulated object. The object of interest consists of two 3D straight lines linked with a revolute joint. The horizontal axis of the figures is the iteration number. a) Rotation velocity (deg/s). b) Translation velocity (m/s). c) Rotation error (rotation task function) (deg) . d) Translation error (translation task function) (m). e) Angular velocity (deg/s). f) Joint angle value ψ (deg). The joint angle ψ converges toward the final value ψ_∞. The pose task functions converge to zero (asymptotic convergence).*

3.6.3 Experiments on articulated object

The articulated object is made with two rigid segment of different colors, linked together by a step motor, as seen in figure (3.26).

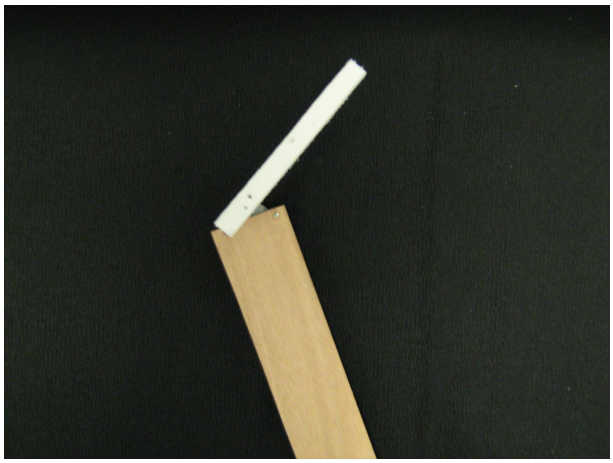


Figure 3.26: The articulated object consists of two rigid parts linked with a step motor.

The two parts axes are represented by two 3D straight lines. The goal is to control the relative pose (Stereo rig - Articulated object) and the joint angle of the object simultaneously.
The Euclidean control laws were used to control the relative pose (Stereo rig - Articulated object), and the joint angle velocity is controlled by the angular velocity $\boldsymbol{\Omega}_\psi = -\lambda_\psi e_\psi$.

Figures (3.27 - a, b, c, d), show the velocities and the convergence errors of the relative pose. The curve are very perturbed, that because, the object consists of only two straight lines (rather than three lines in the previous experiments) which makes the signal-to-noise ratio low, and, the parts of the object are thick which makes their representation by a straight line not perfect.
Figure (3.27 - f) shows the asymptotic convergence of the joint angle ψ toward its final value $\psi_\infty = 15$deg. The stability of the angular velocity control law is shown in figure (3.27 - e).

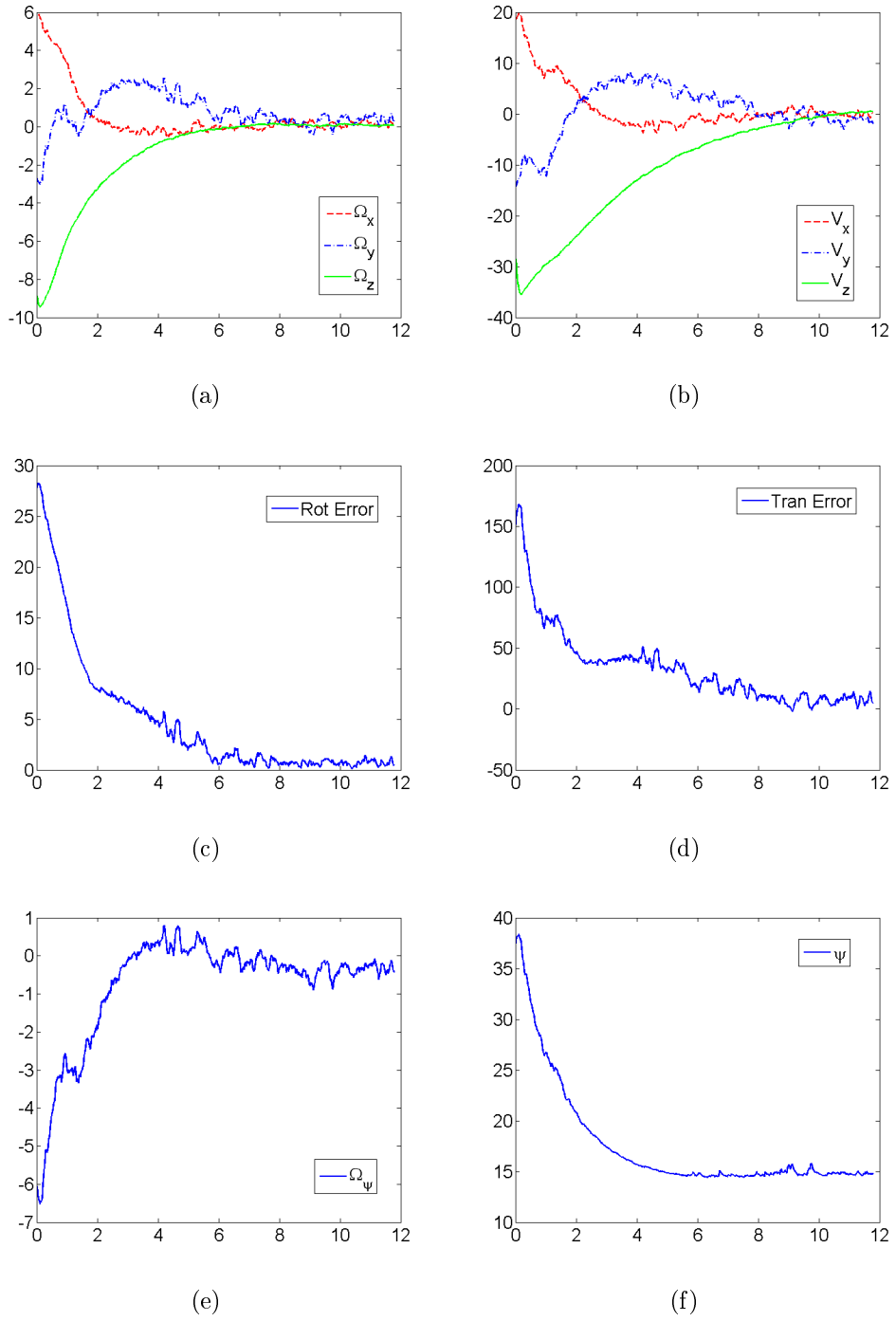


Figure 3.27: *Experimental results for the PBVS of an articulated object. The object of interest consists of two parts linked with a step motor. The horizontal axis of the figures is the time in second. a) Rotation velocity (deg/s). b) Translation velocity (mm/s). c) Rotation error (deg) . d) Translation error (mm). e) Angular velocity (deg/s). f) Joint angle value ψ (deg). The joint angle ψ converges toward the final value $\psi_\infty = 15deg$. The pose task functions converge to zero (asymptotic convergence). The disturbance is very important and that because of the number of the state vectors and the coarse model of the articulated object.*

3.6.4 Object with several revolute joints

We address here the manipulator robots which consists of several axes linked by revolute joints.
The velocities control laws defined in the previous section (3.6) can be extended to an articulated object modelled with n straight lines and $n-1$ revolute joints, if every rotation axis Z_i is perpendicular to the two straight lines L_i, L_{i+1} (see figure 3.22).
The 3D straight lines are represented in the object frame by a set of Plücker coordinates ${}^{o}v_1, {}^{o}w_1, ..., {}^{o}v_n, {}^{o}w_n$, and in the stereo rig frame by $v_1, w_1, ..., v_n, w_n$.
The joints angles $\psi_1, .., \psi_{n-1}$ can be estimated by

$$\cos(\psi_i) \;=\; v_i^{\mathsf{T}} v_{i+1} \qquad\qquad i = 1 \;..\; n-1$$

as the final angles $\psi_{\infty i}$ can be estimated too in the same way, then the $(n-1)$ task functions and control laws can be defined by

$$e_{\psi i} = \psi_i - \psi_{\infty i} \tag{3.23}$$

$$\mathbf{\Omega}_{\psi i} = -\lambda_{\psi i} e_{\psi i} \tag{3.24}$$

The velocity control laws $(\mathbf{\Omega}, \mathbf{V})$, which serve the relative pose (Articulated object - stereo rig) will be computed using the first and second straight lines L_1, L_2 only, exactly like in the case of an articulated object with one joint. That because, the vectors ${}^{o}v_3, {}^{o}w_3$ are the images of $v"_3, w"_3$ by the Euclidean transformation $[\mathtt{R}, t]$, $v"_3, w"_3$ are the images of v'_3, w'_3 by a rotation $\varphi_2 = \psi_3 - \psi_{\infty 3}$, and v'_3, w'_3 are the images of v_3, w_3 by a transformation depending on $\varphi_1 = \psi_2 - \psi_{\infty 2}$. The relation between ${}^{o}v_4, {}^{o}w_4$ and v_4, w_4 is more complicated.
So it will be not recommended to use $L_3, ..., L_n$ to enhance the estimation of $[\mathtt{R}, t]$. That because the errors in the estimation of all the joints angles will accumulate in the estimation of $[\mathtt{R}, t]$.
So, as two straight lines are enough to define the velocity control laws $(\mathbf{\Omega}, \mathbf{V})$, we will use the two straight lines L_1, L_2 only, for control the relative pose between the articulated object and the stereo rig.

3.6.5 Simulation results for an articulated object with two joints

Like the previous simulations, the robot is considered as a perfect integrator. The stereo system consists of two parallel identical cameras embedded on the robot end-effector.
The articulated object is modelled with three 3D straight lines, and two revolute

joints. The objective of the simulation is to control the two joints angles ψ_1, ψ_2, and the relative pose (Articulated object - Stereo rig) simultaneously.
For the pose servoing, only the first two straight lines are used, so no difference are expected from the case of object with one joint. That what figures (3.28 - a, b, c, d) show for the rotational and translational task functions and velocities.
The angles (ψ_1, ψ_2) converge towards their final values (fig. 3.28 - f, h), and their angular velocities $\mathbf{\Omega}_{\psi 1}, \mathbf{\Omega}_{\psi 2}$ are decreasing in an exponential way towards zero (fig. 3.28 - e, g).

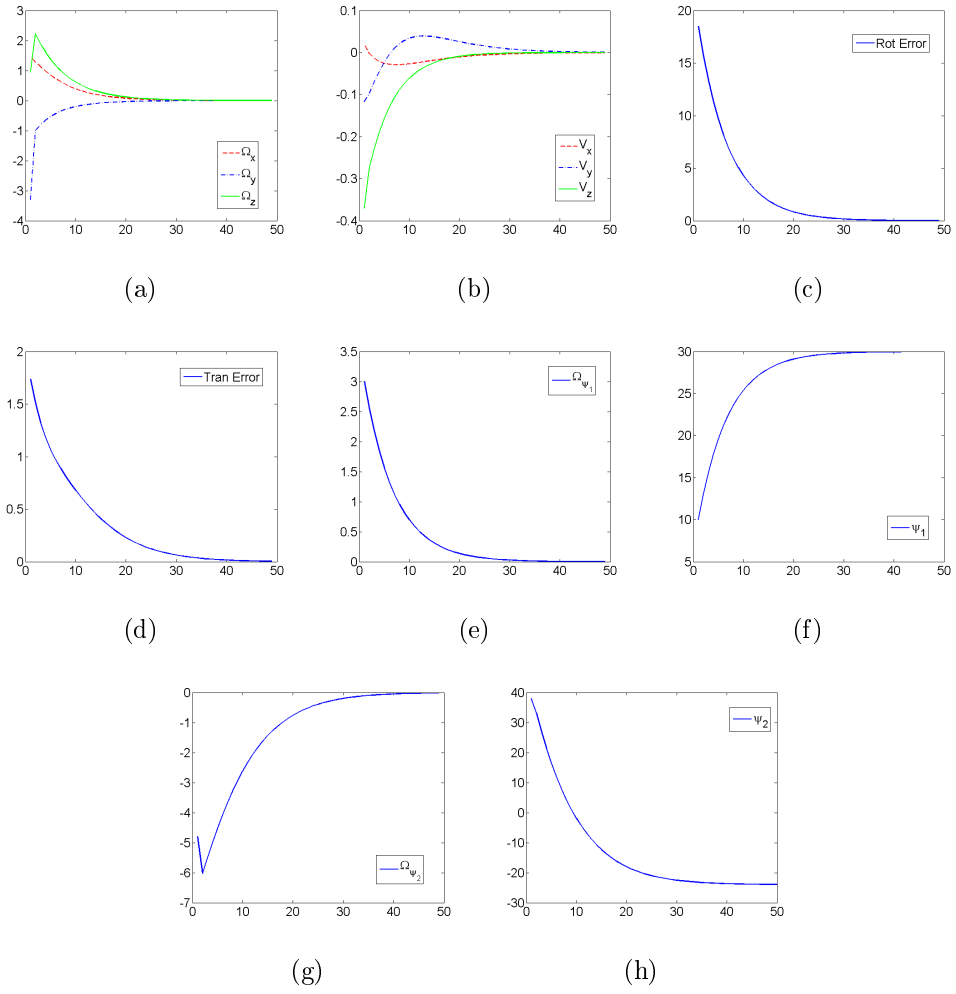


Figure 3.28: *Simulation results for the PBVS of an articulated object with 2 joints. The object of interest consists of three 3D straight lines linked with two revolute joints. The horizontal axis of the figures is the iteration number. a) Rotation velocity (deg/s). b) Translation velocity (m/s). c) Rotation error (rotation task function) (deg) . d) Translation error (translation task function) (m). e) Angular velocity* $\mathbf{\Omega}_{\psi 1}$ *(deg/s). f) Angle value* ψ_1 *(deg). g) Angular velocity* $\mathbf{\Omega}_{\psi 2}$ *(deg/s). h) Angle value* ψ_2 *(deg). The joints angles* ψ_1, ψ_2 *converge toward their final values. The pose task functions converge to zero (asymptotic convergence).*

3.7 Conclusion

In this chapter, an overview on the pose based visual servoing (PBVS) from straight lines is presented. By the estimation of the relative pose between the stereo rig and the object of interest, we could use the "Euclidean control laws" in the visual closed loop.

$$\begin{aligned} \mathbf{V} &= -\lambda_V \, \mathtt{R}^\mathsf{T} \, t \\ \mathbf{\Omega} &= -\lambda_\Omega \, \theta \, u \end{aligned}$$

The "Euclidean control laws" are fully decoupled, and verify the global asymptotic stability of the system in closed loop.
These velocity control laws are simulated and applied to a robotic platform. The results are illustrated and analyzed in this chapter.

Without estimation for the rotational matrix $\mathtt{R}$ or the translational vector t, the "Plücker coordinates control laws" are valid to control the mobile object or the robot end-effector in the vision closed loop.
The direction vector v represents the rotation state vector, and the orthogonal distance d or the norm of the vector w represents the translation state vector. These state vectors are independent and verify the decoupling between the rotation and the translation velocities.

$$\begin{aligned} \mathbf{V} &= -\lambda_V \, \mathtt{H}_V^+ \, e_V \\ \mathbf{\Omega} &= -\lambda_\Omega \, \mathtt{H}_\Omega^+ \, e_\Omega \end{aligned}$$

The global asymptotic stability (GAS) of the system with "Plücker coordinates control laws" are proved. Then many simulations have been carried out to check the theoretical studies, and show the GAS of the system in closed loop, before applying the "Plücker coordinates control laws" to a robotic platform.

The "Plücker coordinates control laws" are modified to be suitable for the special objects which consist of one 3d straight line. Two pairs of velocity control laws are designed, one of them is fully decoupled, and the other is partially decoupled.
We present the simulation for the PBVS on one 3D straight line with and without noise for both pair of velocities.
Experiments on a robot manipulator are presented. The goal was to control the relative pose (3D line - stereo rig). The velocity control laws are

$$\begin{aligned} \mathbf{\Omega} &= -\lambda_\Omega \, \mathtt{H}_\Omega^+ e_\Omega \\ \mathbf{V} &= -\lambda_V \, (d - d^*) u \end{aligned}$$

and guide the robot end-effector to the desired pose.

Then, a new application for the stereo PBVS is presented, the visual servoing of an articulated object. The goal was to control the joints of the articulated object and the relative pose (Articulated object - stereo rig) simultaneously.
We start with a simple articulated object, which consists of two 3D straight lines and one revolute joint. With this kind of object, there are 7 DOF to be controlled. 6 DOF of them are concerning the relative pose (Articulate object - Stereo rig), and the seventh to control the revolute joint of the object.
A new task function is designed to define the error of angular servoing $e_\psi = \psi - \psi_\infty$. Then the angular velocity control law is given by $\boldsymbol{\Omega}_\psi = -\lambda_\psi e_\psi$.
The possibility of estimating the joint angle during the servoing, makes the articulated object as a rigid object by computing the virtual pose of the 3D straight lines using Rodrigues' rotation formula.
The simulations and the experiments on a robot manipulator show the global asymptotic stability of the visual closed loop and the angular closed loop.
These task functions and velocity control laws are generated for servoing an articulated object with several revolute joints. The first two 3D straight lines are used to estimate the Euclidean transformation $[\mathtt{R}, t]$ just like in the case of a simple articulated object.
The simulations show the convergence of all the joint angles ψ_i towards their final values in the desired pose.

Chapter 4

Stereo image based visual servoing from lines

In PBVS, the vision system was considered as a pose sensor. It needs to estimation the Euclidean relative pose between the object of interest and the vision system.
To do that, one must use the metric camera model, represented by the intrinsic and extrinsic parameters, which requires the vision system calibration.
Although the new algorithms for calibration, and the height precession of the metric camera model estimation, this model is not perfect, and there is no way to get an exact measurement of the object pose.
The estimated pose is the feedback of the vision closed loop (fig. 3.1). So the error in this pose causes an error in the final pose of the system. This error is called "static error", because it is constant and does not related to the difference with initial and final poses.
The Image based visual servoing (IBVS) does not need pose estimation (but some components may need to partially retrieve the pose), then does not require the metric camera model. So, the static error due to the pose estimation will be null.
In this kind of servoing, the state vector is defined in the image plane. The error is the difference between the state vector in the current image and the desired one.
Figure (4.1) shows the block diagram of classical IBVS. The symbol F^* denotes the desired features, and $\hat{F}$ is the actual image features.
The camera model does not appear in the feedback chain, it may be rather part of the direct chain, but this does not cause a static error in the system convergence.
The error of servoing in the closed loop of an IBVS is given by

$$e = S(t) - S^*$$

S is the state vector, it can be a feature $S = F$ or a combination of features $S = f(F)$.
The goal of this chapter is to design decoupled control laws for translation velocity

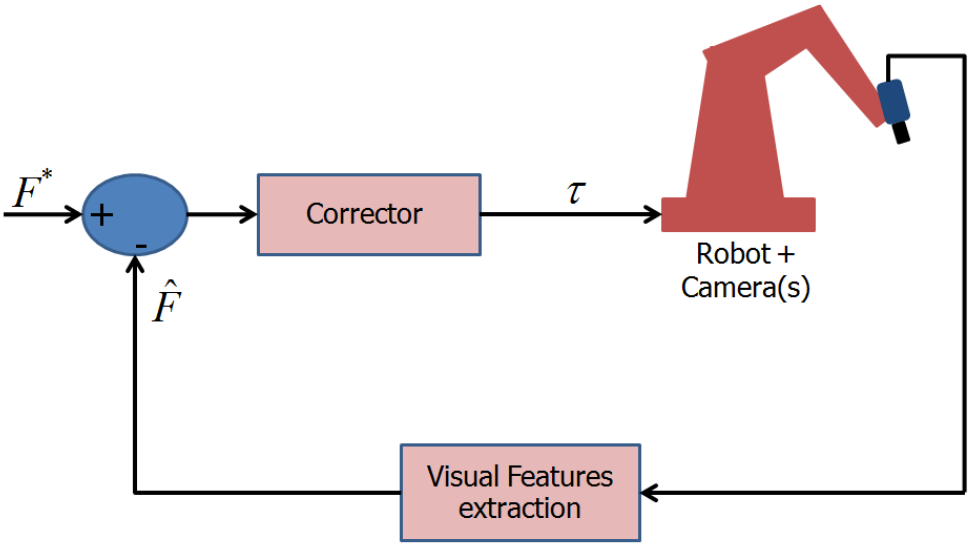


Figure 4.1: *Image based visual servoing (IBVS) block diagram. The difference between the current and desired features generate the control laws, and drives the robot which holds the vision system toward the desired image.*

and rotation velocity, by the help of the stereo vision and Plücker coordinates for straight lines.

4.1 Interaction matrices for lines

In this section, we present some interaction matrices defined in the literature for the line state vector, and a new interaction matrix generated by the lines equations (2.1, 2.2)

$$\begin{aligned} l_r &\equiv \mathtt{K}_r^{-\mathsf{T}} \left[\mathtt{R}_r \; w + t_r \times \mathtt{R}_r \; v\right] \\ l_l &\equiv \mathtt{K}_l^{-\mathsf{T}} \left[\mathtt{R}_l \; w + t_l \times \mathtt{R}_l \; v\right] \end{aligned}$$

Then several simulations will be presented to evaluate the behaviour of the system, for every interaction matrix.

4.1.1 Others interaction matrices

Chaumette in his PhD thesis [Cha90] defined three interaction matrices according to the straight line representation:

1. The 3D straight line is represented by normalized Plücker coordinates v, w and the 2D image line represented by the normalized Plücker coordinates l_v, l_w too.

With $l_v = (w_2/\triangle, -w_1/\triangle, 0)^\mathsf{T} = (L_2, -L_1, 0)^\mathsf{T}$ and $l_w = (w_1/\triangle, w_2/\triangle, w_3/\triangle)^\mathsf{T}$ $= (L_1, L_2, L_3)^\mathsf{T}$, $\triangle = \sqrt{w_1^2 + w_2^2}$.
The interaction matrix for this representation is of dimensions 3×6 for each 3D straight line, and is given by

$$\left\{ \begin{array}{l} \mathtt{L_I(1)} = \left[\begin{array}{cccccc} L_1L_2v_3/\triangle & L_2^2v_3/\triangle & -(L_1L_2v_1 + L_2^2v_2)/\triangle & & & \\ L_1L_2L_3 & L_2^2L_3 & -L_2 & & & \end{array} \right] \\ \mathtt{L_I(2)} = \left[\begin{array}{cccccc} -L_1^2v_3/\triangle & -L_1L_2v_3/\triangle & (L_1L_2v_2 + L_1^2v_1)/\triangle & & & \\ L_1^2L_3 & -L_1L_2L_3 & L_1 & & & \end{array} \right] \\ \mathtt{L_I(3)} = \left[\begin{array}{cccccc} (v_2 + L_2L_3v_3)/\triangle & -(v_1 + L_1L_3v_3)/\triangle & (L_1L_3v_2 - L_2L_3v_1)/\triangle & & & \\ L_2(1+L_3^2) & -L_1(1+L_3^2) & 0 & & & \end{array} \right] \end{array} \right\} \quad (4.1)$$

The matrix lines are dependent because of the quadratic constraint $L_1^2 + L_2^2 = 1$.

2. The 3D straight line is represented by the intersection of two planes, these two planes are defined by two 4 components vectors $(a_1, b_1, c_1, d_1)^\mathsf{T}$ and $(a_2, b_2, c_2, d_2)^\mathsf{T}$.
The 2D image line l is represented by an angle θ and a distance ρ, $l = \big(\cos(\theta), \sin(\theta), -\rho\big)^\mathsf{T}$.
The interaction matrix is of dimensions 2×6 for each line and is given by

$$\left\{ \begin{array}{l} \mathtt{L_I}(\rho) = \left[\begin{array}{cccccc} \lambda_\rho \cos(\theta) & \lambda_\rho \sin(\theta) & -\lambda_\rho \rho & (1+\rho^2)\sin(\theta) & -(1+\rho^2)\cos(\theta) & 0 \end{array} \right] \\ \mathtt{L_I}(\theta) = \left[\begin{array}{cccccc} \lambda_\theta \cos(\theta) & \lambda_\theta \cos(\theta) & -\lambda_\theta \rho & -\rho\cos(\theta) & -\rho\sin(\theta) & -1 \end{array} \right] \end{array} \right\} \quad (4.2)$$

with

$$\begin{array}{rcl} \lambda_\rho & = & \big[(c_2a_1 - c_1a_2)\cos(\theta) + (c_2b_1 - c_1b_2)\sin(\theta)\big]/\sqrt{A^2 + B^2} \\ \lambda_\theta & = & (a_2b_1 - a_1b_2)/\sqrt{A^2 + B^2} \\ A & = & a_1d_2 - a_2d_1 \quad , \quad B = b_1d_2 - b_2d_1 \quad , \quad C = c_1d_2 - c_2d_1 \end{array}$$

3. The 3D straight line is represented by the intersection of two planes. These two planes are defined by two 4 component vectors $(a_1, b_1, c_1, d_1)^\mathsf{T}$ and $(a_2, b_2, c_2, d_2)^\mathsf{T}$.
The 2D image line l is represented by (a,b). This representation is minimal but needs to use two equations according to the pose of the line in the image:
line most vertical: $Y = aX + b$ $(a = -A/B, b = -C/B)$ $\rightarrow$ $l = (a, -1, b)^\mathsf{T}$.
line most horizontal: $X = aY + b$ $(a = -B/A, b = -C/A)$ $\rightarrow$ $l = (-1, a, b)^\mathsf{T}$.
The interaction matrix is then given by

$$\left\{ \begin{array}{l} \mathtt{L_I(a)} = \left[\begin{array}{cccccc} \lambda_1 a & -\lambda_1 & \lambda_1 b & 1+b^2 & a & -ab \end{array} \right] \\ \mathtt{L_I(b)} = \left[\begin{array}{cccccc} \lambda_2 a & -\lambda_2 & \lambda_2 b & ab & -b & -1-a^2 \end{array} \right] \end{array} \right\} \quad (4.3)$$

with $\lambda_1 = -(b_1 b + c1)/d_1$ and $\lambda_2 = -(b_1 a + a1)/d_1$ if $d_1 \neq 0$,
or $\lambda_1 = -(b_2 b + c2)/d_2$ and $\lambda_2 = -(b_2 a + a2)/d_2$ if $d_1 = 0$.

The second and third interaction matrices (4.2, 4.3) are identical, because the 3D straight line is represented in the same way, and so the 2D image line, the only difference is the norm of the 2D image line.
The first matrix is closer to our work. It involves the Plücker coordinates for a 3D straight line. We will use this matrix for simulation experiments and to compare the results with our work.

4.1.2 The stereo interaction matrix

Here, a new interaction matrix will be designed using the normalized Plücker coordinates to represent a 3D straight line. The image lines are represented by two 3-component vectors $l_r = (a_r, b_r, c_r)^\mathsf{T}, l_l = (a_l, b_l, c_l)^\mathsf{T}$.
The relations between the image lines and the Plücker coordinates of the 3D straight line are given in the equations (2.1, 2.2)

$$\begin{aligned} l_r &\equiv \mathtt{K}_r^{-\mathsf{T}} \ [\mathtt{R}_r \ w + t_r \times \mathtt{R}_r \ v] \\ l_l &\equiv \mathtt{K}_l^{-\mathsf{T}} \ [\mathtt{R}_l \ w + t_l \times \mathtt{R}_l \ v] \ . \end{aligned}$$

These two equations are defined with two different and arbitrary scale factors. To find the time derivative of these equations and define the scale factors, we will estimate the Plücker coordinates $\hat{v}, \hat{w}$ of the 3D straight line using the equations (2.3, 2.4)

$$\begin{aligned} \hat{v} &\equiv \mathtt{R}_l^\mathsf{T} \ \mathtt{K}_l^\mathsf{T} \ l_l \times \mathtt{R}_r^\mathsf{T} \ \mathtt{K}_r^\mathsf{T} \ l_r \\ \hat{w} &\equiv \mathtt{R}_r^\mathsf{T} \ \mathtt{K}_r^\mathsf{T} \ l_r \ l_l^\mathsf{T} \ \mathtt{K}_l \ t_l \ - \ \mathtt{R}_l^\mathsf{T} \ \mathtt{K}_l^\mathsf{T} \ l_l \ l_r^\mathsf{T} \ \mathtt{K}_r \ t_r \end{aligned}$$

The notation $(\hat{.})$ means that the Plücker coordinates $\hat{v}, \hat{w}$ should be seen either as estimations of the real v, w or internal observers, then with some nominal values for intrinsic parameters. Moreover, $\hat{v}, \hat{w}$ have only ONE common scale factor, and to benefit to the straight line kinematic defined in (2.5, 2.6)

$$\begin{aligned} \dot{\hat{v}} &= \boldsymbol{\Omega} \times \hat{v} \\ \dot{\hat{w}} &= \boldsymbol{\Omega} \times \hat{w} + \mathbf{V} \times \hat{v} \ , \end{aligned}$$

the scale factor will be selected to verify $||\hat{v}|| = 1$.
Now, our state vectors are designed from the equations (2.1, 2.2) after defining the scale factors verifying $||\hat{v}|| = 1$.

$$\begin{aligned} s_r &= \mathtt{K}_r^{-\mathsf{T}} \ [\mathtt{R}_r \ \hat{w} + t_r \times \mathtt{R}_r \ \hat{v}] \\ s_l &= \mathtt{K}_l^{-\mathsf{T}} \ [\mathtt{R}_l \ \hat{w} + t_l \times \mathtt{R}_l \ \hat{v}] \end{aligned}$$

These state vectors (s_r, s_l) are equivalent to the image lines (l_r, l_l), but with a precise scale factor. The desired state vector (s_r^*, s_l^*) are selected in the same way as with the desired image.
The interaction matrix can be defined from the lattest equations, as follows:

$$\begin{aligned}
s_r &= \mathtt{K}_r^{-\mathsf{T}} \left[\mathtt{R}_r\ \hat{w} + t_r \times \mathtt{R}_r\ \hat{v}\right] \\
s_r &= \mathtt{K}_r^{-\mathsf{T}}\mathtt{R}_r \left[\hat{w} + \mathtt{R}_r^{\mathsf{T}} t_r \times \hat{v}\right] \\
\dot{s}_r &= \mathtt{K}_r^{-\mathsf{T}}\mathtt{R}_r \left[\dot{\hat{w}} - c_r \times \dot{\hat{v}}\right] \quad c_r \text{ is the camera right center} \\
\dot{s}_r &= \mathtt{K}_r^{-\mathsf{T}}\mathtt{R}_r \left[\boldsymbol{\Omega} \times \hat{w} + \mathbf{V} \times \hat{v} - c_r \times (\boldsymbol{\Omega} \times \hat{v})\right] \\
\dot{s}_r &= \mathtt{K}_r^{-\mathsf{T}}\mathtt{R}_r \left[-[\hat{w}]_\times \boldsymbol{\Omega} - [\hat{v}]_\times \mathbf{V} + [c_r]_\times [\hat{v}]_\times \boldsymbol{\Omega}\right] \\
\dot{s}_r &= \mathtt{K}_r^{-\mathsf{T}}\mathtt{R}_r \left[-[\hat{v}]_\times \mathbf{V} - ([\hat{w}]_\times - [c_r]_\times [\hat{v}]_\times)\boldsymbol{\Omega}\right]
\end{aligned}$$

So, supposing $\mathtt{M}_r = \mathtt{K}_r\mathtt{R}_r, \mathtt{M}_l = \mathtt{K}_l\mathtt{R}_l$, the interaction matrix for a stereo images of a 3D line is given by

$$\mathtt{L}_I = \begin{bmatrix} -\mathtt{M}_r^{-\mathsf{T}}[\hat{v}]_\times & \mathtt{M}_r^{-\mathsf{T}}[c_r]_\times[\hat{v}]_\times - \mathtt{M}_r^{-\mathsf{T}}[\hat{w}]_\times \\ -\mathtt{M}_l^{-\mathsf{T}}[\hat{v}]_\times & \mathtt{M}_l^{-\mathsf{T}}[c_l]_\times[\hat{v}]_\times - \mathtt{M}_l^{-\mathsf{T}}[\hat{w}]_\times \end{bmatrix} \tag{4.4}$$

This is a singular interaction matrix for the 3D line of dimensions 6×6, as it is a rank-4 matrix. The time derivative of the state vectors $S = (s_r^{\mathsf{T}}, s_l^{\mathsf{T}})^{\mathsf{T}}$ can be related with the kinematic screw $\tau = (\mathbf{V}^{\mathsf{T}}, \boldsymbol{\Omega}^{\mathsf{T}})^{\mathsf{T}}$ by $\dot{S} = \mathtt{L}_\mathtt{I}\tau$

$$\begin{pmatrix} \dot{s}_r \\ s_l \end{pmatrix} = \left[\begin{array}{c|c} -\mathtt{M}_r^{-\mathsf{T}}\ [\hat{v}]_\times & \mathtt{M}_r^{-\mathsf{T}}\ [c_r]_\times[\hat{v}]_\times - \mathtt{M}_r^{-\mathsf{T}}\ [\hat{w}]_\times \\ \hline -\mathtt{M}_l^{-\mathsf{T}}\ [\hat{v}]_\times & \mathtt{M}_l^{-\mathsf{T}}\ [c_l]_\times[\hat{v}]_\times - \mathtt{M}_l^{-\mathsf{T}}\ [\hat{w}]_\times \end{array}\right] \begin{pmatrix} \mathbf{V} \\ \boldsymbol{\Omega} \end{pmatrix}$$

With two 3D lines, one can obtain an interaction matrix $\mathtt{L}_\mathtt{I}$ of dimensions 12×6 which may has a left pseudo inverse if the two lines are not parallel (cf. chapter 3).

In the case of n straight lines in the scene, the interaction matrix $\mathtt{L}_\mathtt{I}$ will be of dimensions $6n \times 6$, and the task function e will be of dimensions $6n$. The expression for $\mathtt{L}_\mathtt{I}$ and e are

$$\mathtt{L}_\mathtt{I} = \begin{bmatrix} -\mathtt{M}_r^{-\mathsf{T}}\ [\hat{v}1]_\times & \mathtt{M}_r^{-\mathsf{T}}\ [c_r]_\times[\hat{v}1]_\times - \mathtt{M}_r^{-\mathsf{T}}\ [\hat{w}1]_\times \\ -\mathtt{M}_l^{-\mathsf{T}}\ [\hat{v}1]_\times & \mathtt{M}_l^{-\mathsf{T}}\ [c_l]_\times[\hat{v}1]_\times - \mathtt{M}_l^{-\mathsf{T}}\ [\hat{w}1]_\times \\ \vdots & \vdots \\ -\mathtt{M}_r^{-\mathsf{T}}\ [\hat{v}n]_\times & \mathtt{M}_r^{-\mathsf{T}}\ [c_r]_\times[\hat{v}n]_\times - \mathtt{M}_r^{-\mathsf{T}}\ [\hat{w}n]_\times \\ -\mathtt{M}_l^{-\mathsf{T}}\ [\hat{v}n]_\times & \mathtt{M}_l^{-\mathsf{T}}\ [c_l]_\times[\hat{v}n]_\times - \mathtt{M}_l^{-\mathsf{T}}\ [\hat{w}n]_\times \end{bmatrix} \qquad \mathtt{e} = \begin{pmatrix} s_{r1} - s_{r1}^* \\ s_{l1} - s_{l1}^* \\ \vdots \\ s_{rn} - s_{rn}^* \\ s_{ln} - s_{ln}^* \end{pmatrix}$$

The next section presents the simulation results and comparisons of these two stereo IBVS, using the interaction matrix defined by Chaumette (Eq. 4.1) and this new interaction matrix (Eq. 4.4).

4.2 Simulation results

This section presents the simulations results of the system behaviour in the closed loop, after applied the control law

$$\tau = -\lambda \widehat{\mathtt{L_I}^+} e.$$

with the interaction matrix defined by Chaumette (Eq. 4.1), and the new interaction matrix defined in the previous section (Eq. 4.4).
As the simulation does not interested of the robot dynamics, the robot is considered as a perfect integrator. The stereo system consists of two parallel identical cameras. The object of interest consists of three non coplanar straight lines.
The intrinsic parameters of the two cameras are

$$\mathtt{K_r} = \mathtt{K_l} = \begin{bmatrix} 1200 & 0 & 300 \\ 0 & 1200 & 200 \\ 0 & 0 & 1 \end{bmatrix}$$

The goal of the simulations is to verify the stability and the convergence of the closed loop system in the perfect case, and the global behaviour with the presence of a white noise.
As known, the IBVS is robust against calibration errors, so the simulations will not interested in the servoing behaviour with calibration errors.

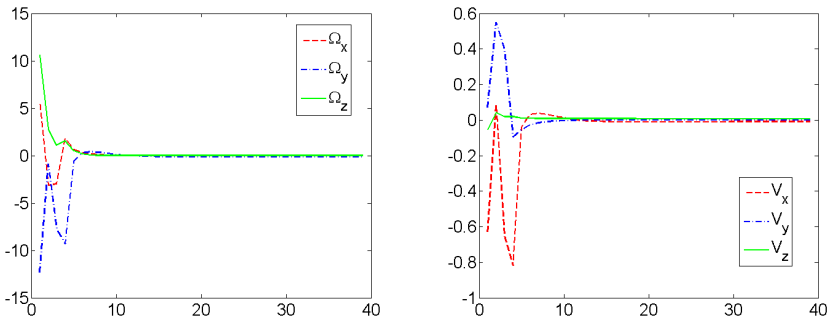
4.2.1 Simulation for the interaction matrix defined by Chaumette

For this simulation, the interaction matrix used is the one defined by Chaumette for 3D straight lines represented by Plücker coordinates (Eq. 4.1).
The gain of the control law is set to $\lambda = 0.45$. No white noise is added to the image lines.

The simulation results show the existence of local minima, like the example shown in figure (4.2).
To be familiar with the state vectors, we present the initial and desired state vectors in the right image:

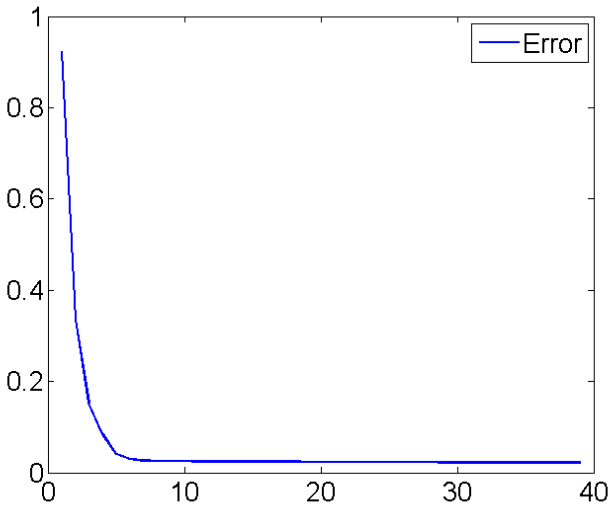
$$l_1 = \begin{pmatrix} 0.9959 \\ -0.0905 \\ -0.1516 \end{pmatrix} \quad l_2 = \begin{pmatrix} 0.3363 \\ -0.9417 \\ 0.2892 \end{pmatrix} \quad l_3 = \begin{pmatrix} 0.8944 \\ 0.4472 \\ -0.0958 \end{pmatrix}$$

$$l_1^* = \begin{pmatrix} 0.9995 \\ 0.0320 \\ -0.0623 \end{pmatrix} \quad l_2^* = \begin{pmatrix} 0.2853 \\ -0.9584 \\ 0.1707 \end{pmatrix} \quad l_3^* = \begin{pmatrix} 0.8391 \\ 0.543 \\ 0.0436 \end{pmatrix}$$



(a)

(b)



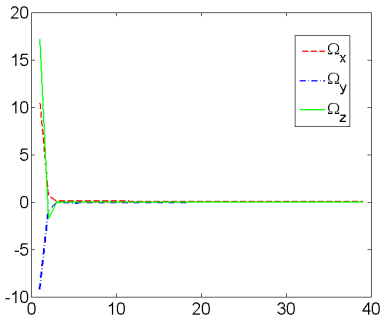
(c)

Figure 4.2: *Simulation results for the IBVS using the interaction matrix defined by Chaumette (Eq. 4.1). Case of non convergence: the object of interest consists of three 3D straight lines. The horizontal axis of the figures is the iteration number. a) Rotational velocity (deg/s). b) Translational velocity (m/s). c) Convergence error (task function e). The error (the task function) is not null at the end of servoing, which means that the system converges toward a local minimum.*

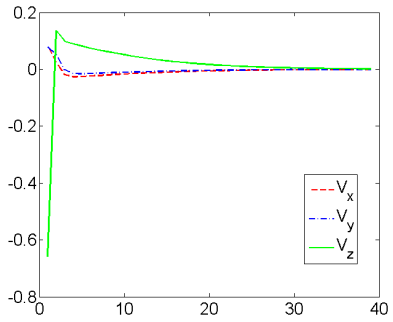
Although the decreasing of rotational and translational velocities to zero (fig. 4.2 - a, b), the state vectors do not reach their desired values, and that is shown in (fig. 4.2 - c).

This figure (fig. 4.2 - c) represents the norm of the error of convergence (the task function e).

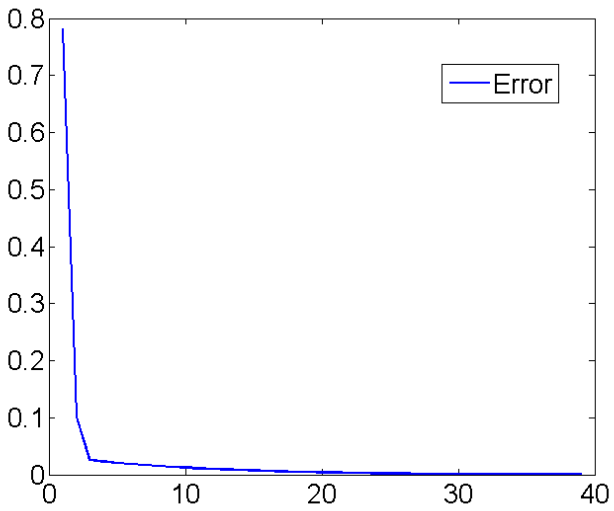
Now we change the desired state vectors, to reduce the distance between the initial and the final poses. The asymptotic convergence is shown in figure (4.3). The real



(a)



(b)



(c)

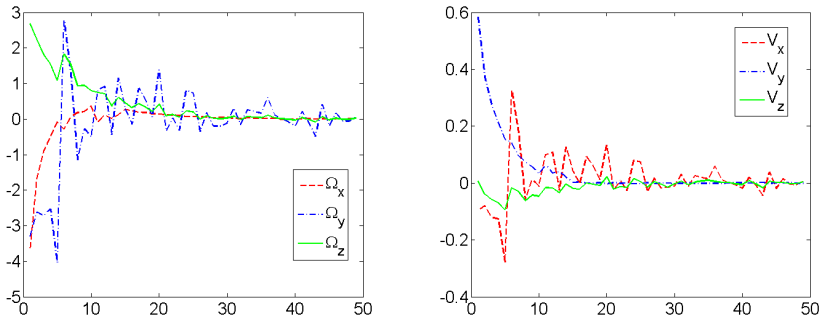
Figure 4.3: *Simulation results for the IBVS using the interaction matrix defined by Chaumette (Eq. 4.1). Case of convergence: the object of interest consists of three 3D straight lines. The horizontal axis of the figures is the iteration number. a) Rotational velocity (deg/s). b) Translational velocity (m/s). c) Convergence error (task function e). The error (the task function) is asymptotic convergence.*

difference in this case is the asymptotic convergence of the task function e shown in (fig. 4.3 - c). This difference is due to the small difference between the initial and the desired state vectors.

As one can see, the norm of the task function in this case in the first iteration is about 0.45 (fig. 4.3 - c) but in the last case it is about 0.9 (fig. 4.2 - c). Figures (4.3 - a, b) show the velocity control laws. There are no difference in the behavior of the velocities in the two cases (asymptotic convergence or convergence toward a local minimum).

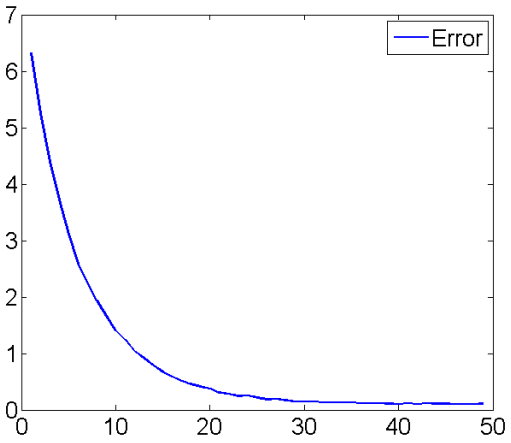
4.2.2 Simulation for the stereo interaction matrix

The stereo interaction matrix $\mathtt{L_I}$ defined in (4.4) considered as a big matrix (6×6 for each 3D straight line). It is so difficult to go on with a theoretical analyses, for that, the simulation results will be the assistant to evaluate the behaviour of the velocities designed using this stereo interaction matrix.
The gain of the control law is set to $\lambda = 0.25$, no white noise has been added to the image lines.
The simulation results show the existence of local minima, like in the simulation results in the previous section. An example of convergence toward a local minimum is shown in figure (4.4).



(a)

(b)



(c)

Figure 4.4: *Simulation results for the IBVS using the stereo interaction matrix (Eq. 4.4). Case of non convergence: the object of interest consists of three 3D straight lines. The horizontal axis of the figures is the iteration number. a) Rotational velocity (deg/s). b) Translational velocity (m/s). c) Convergence error (task function e). The error (the task function) is not null at the end of servoing, which means that the system converge towards a local minimum.*

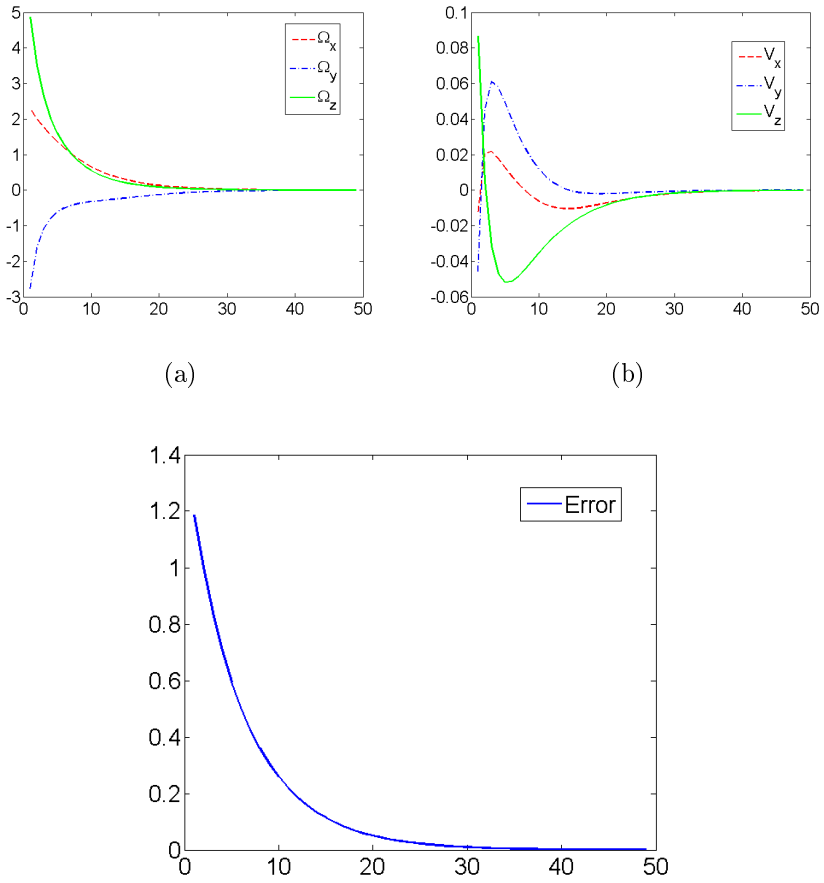
Figures (4.4 - a, b) show the rotational and the translational velocities. We separate these two figures for better display, not for the decoupling (they are not decoupled). All the velocities components (except $\mathbf{\Omega}_z$) increase at the start of servoing simulation and reach their maximum before they decrease to zero.
Figure (4.4 - c) shows the task function error. One can observe the convergence of the task function toward a non null value, that means a local minimum has been reached.

Figure (4.5), shows an example of a stereo IBVS with asymptotic convergence. The current state vectors converges toward their desired values.
Figures (4.5 - a, b) show the rotational and translational velocities. They reveal a pike near the 25^{th} iteration because of a singularity in one 3D straight line of the object of interest, but the closed loop system is still stable. In figure (4.5 - c), one can observe the asymptotic convergence of the task function.

This simulation is repeated with the presence of white noise in the images. The noise is modeled by adding a randoms value (less than one pixel) to the coordinates of each pixel in the image line, then reestimate the lines coordinates with the least square method. The results are displayed in fig. 4.6.
Even if the error at the end of servoing is very small (fig. 4.6 -c), the observed disturbances in the velocities (fig. 4.6 -a, b) are very significant, and especially the translational velocity control (b).
The white noise added to the image is not very important (less than one pixel for each point coordinates in the image line), so one can consider this control laws are not very robust against the noise.

Other simulations have been performed with different objects (2, 3, 4, ... straight lines) and different initial and final state vectors, sometimes we get the asymptotic convergence, and sometimes a local minimum is reached.
In both cases, the simulations results show an undesired behaviour of the system in closed loop. That is clear in the velocities curves (fig. 4.5 - a, b) and (fig. 4.4 - a, b), beside the big effect of the noise (fig. 4.6 - a, b). As one wish a first degree system behaviour, and that means an exponential decrease of the velocity components.

The difficulty in developing the interaction matrix together with the bad behaviour of the system in the servoing closed loop, make a serious need of decoupling the control laws for rotation and translation, which makes the theoretical analyses of the interaction matrix relevant and easier.

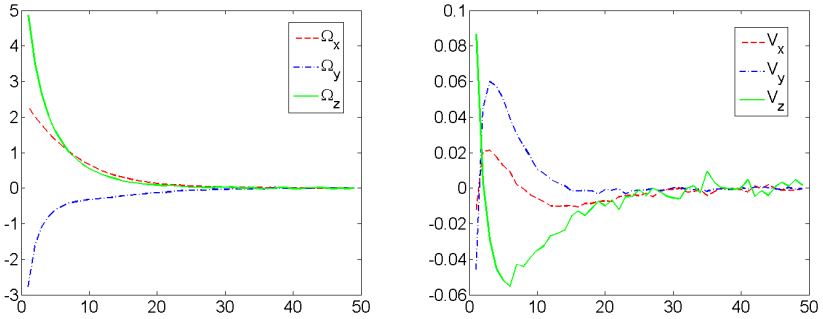


(a) (b)

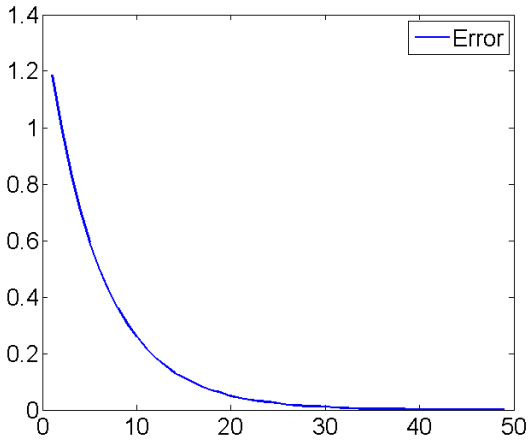
(c)

Figure 4.5: *Simulation results for the IBVS using the stereo interaction matrix (Eq. 4.4). Case of convergence: the object of interest consists of three 3D straight lines. The horizontal axis of the figures is the iteration number. a) Rotational velocity (deg/s). b) Translational velocity (m/s). c) Convergence error (task function e). The error (the task function) is almost null at the end of servoing (there is a small disturbance due to the white noise), which means that the system is asymptotically convergent.*

In the next section, we will look for a method for decoupling the two control laws (rotation, translation), and this will involve the definition of new state vectors.



(a) (b)



(c)

Figure 4.6: *Simulation results for the IBVS using the stereo interaction matrix (Eq. 4.4). Case of convergence in the presence of white noise in the image: the object of interest consists of three 3D straight lines. The horizontal axis of the figures is the iteration number. a) Rotational velocity (deg/s). b) Translational velocity (m/s). c) Convergence error (task function e). The error (the task function) is null at the end of servoing, which means that the system is asymptotic convergence.*

4.3 Stereo Visual Servoing (SVS)

In the previous section, the stereo image lines (l_r, l_l) are considered as state vectors and used directly in the task function. This makes the stereo system no more than a vision system with two cameras, and the interaction matrix is a stacking of two Jacobian of the image pair. This visual servoing diagram is a classical IBVS.

In this section, we rather try here to transfer this classical IBVS into the Stereo Visual Servoing SVS. This can be done by defining a new state vectors S in a new space called stereo servoing space, a kind of virtual space which is not the Euclidean space nor image planes.

In the stereo servoing space, the stereo image lines (l_r, l_l) are the visual measurements. The state vectors must be separated to define independent task functions for rotation and translation. The independent task functions define decoupled control laws for rotation and translation velocities.

In the Euclidean space, the stereo rig is characterized by the rotation matrix ($\mathtt{R}$) between the right and left camera frames, and the translation vector (t) between their origins. Beside this extrinsic parameters, there are the intrinsic parameters, represented by the camera matrices $\mathtt{K}_r$ and $\mathtt{K}_l$, which transfer the measurements in the plan image to measurements in the camera frame.
Calibration of a stereo rig in the Euclidean space means the estimation of these matrices. Whatever the method used and the precision obtained for the estimation of these parameters, the perfect values will never reach, and that because, the metric model of the camera is an approximate model. This approximation and errors usually yield a static error with a PBVS.

The stereo image can be related by a projective matrix called the fundamental matrix $\mathtt{F}$, while the infinite homography $\mathtt{H}_\infty$ bring the affine property of the stereo image.
The fundamental matrix $\mathtt{F}$ is independent of scene structure. It can be computed from correspondences of imaged scene points alone, without requiring knowledge of the camera internal parameters or relative pose.
The infinite homography $\mathtt{H}_\infty$ is the affine transformation from the plane at infinity π_∞ to one of the image planes of the stereo system. It can be computed from the correspondence of at least three vanishing points and the epipoles [BZR95].
Then, to reduce the static error, we propose in the next paragraph to define new state vectors to cope with the calibration trouble holdings and the control decoupling.

4.3.1 New State Vectors

The new state vectors will be a combination of the visual measurements using $\mathtt{F}$ and $\mathtt{H}_\infty$. Recall the kinematic of a 3D line defined in (2.5) and (2.6)

$$\begin{aligned} \dot{v} &= \boldsymbol{\Omega} \times v \\ \dot{w} &= \boldsymbol{\Omega} \times w + \mathbf{V} \times v \end{aligned}$$

These equations show a partial decoupling between the rotational velocity and the translational velocity. The time derivation of the direction vector v is depending on the rotational velocity only. The goal is to define state vectors with kinematics equivalent to the 3D straight line kinematics. For that, recall the Plücker coordinates v, w given in the equations (2.3) and (2.4)

$$\begin{aligned} v &\equiv \mathtt{R}_l^\mathsf{T}\ \mathtt{K}_l^\mathsf{T}\ l_l \times \mathtt{R}_r^\mathsf{T}\ \mathtt{K}_r^\mathsf{T}\ l_r \\ w &\equiv \mathtt{R}_r^\mathsf{T}\ \mathtt{K}_r^\mathsf{T}\ l_r\ l_l^\mathsf{T}\ \mathtt{K}_l\ t_l\ -\ \mathtt{R}_l^\mathsf{T}\ \mathtt{K}_l^\mathsf{T}\ l_l\ l_r^\mathsf{T}\ \mathtt{K}_r\ t_r \end{aligned}$$

The image lines pair l_r, l_l are vectors of three components with 2 DOF. One can add a constraint on the norm of each line. These constraints are chosen so as to simplify the equation (2.4) and to help to define a state vector equivalent to w. The constraints we propose are:

$$\begin{aligned} l_r^\mathsf{T}\ \mathtt{K}_r\ t_r &= -1 \\ l_l^\mathsf{T}\ \mathtt{K}_l\ t_l &= 1 \end{aligned}$$

With the notations $\mathtt{M}_r = \mathtt{K}_r\mathtt{R}_r$, $\mathtt{M}_l = \mathtt{K}_l\mathtt{R}_l$, $\mathtt{H}_\infty = \mathtt{M}_l\ \mathtt{M}_r^{-1}$, and $m = det(\mathtt{M}_r)$ the equation (2.3) becomes

$$\begin{aligned} v &\equiv \mathtt{M}_l^\mathsf{T}\ l_l \times \mathtt{M}_r^\mathsf{T}\ l_r \\ v &\equiv m\mathtt{M}_r^{-1}(\mathtt{M}_r^{-\mathsf{T}}\ \mathtt{M}_l^\mathsf{T}\ l_l \times l_r) \\ v &\equiv m\mathtt{M}_r^{-1}(\mathtt{H}_\infty^\mathsf{T}\ l_l \times l_r) \end{aligned}$$

Take $S_1 = \mathtt{H}_\infty^\mathsf{T}\ l_l \times l_r$

$$v \equiv m\ \mathtt{M}_r^{-1}\ S_1 \tag{4.5}$$

and for the equation (2.4)

$$\begin{aligned} w &\equiv \mathtt{M}_r^\mathsf{T}\ l_r\ +\ \mathtt{M}_l^\mathsf{T}\ l_l \\ w &\equiv \mathtt{M}_r^\mathsf{T}(l_r\ +\ \mathtt{M}_r^{-\mathsf{T}}\ \mathtt{M}_l^\mathsf{T}\ l_l) \\ w &\equiv \mathtt{M}_r^\mathsf{T}(l_r\ +\ \mathtt{H}_\infty^\mathsf{T}\ l_l) \end{aligned}$$

Take $S_2 = l_r \ + \ \mathtt{H}_\infty^\mathsf{T} \ l_l$

$$w \equiv \mathtt{M}_r^\mathsf{T} S_2 \tag{4.6}$$

To normalize the vector v with $||v|| = 1$ the scale factor is chosen as $q = ||m\mathtt{M}_r^{-1}(\mathtt{H}_\infty^\mathsf{T} \ l_l \times l_r)||$. The new state vectors are

$$s_\Omega = S_1/q \tag{4.7}$$
$$s_V = S_2/q \tag{4.8}$$

Then, the normalized Plücker coordinates v, w are given by

$$v = m\mathtt{M}_r^{-1} s_\Omega \tag{4.9}$$
$$w = \mathtt{M}_r^\mathsf{T} s_V \tag{4.10}$$

The equations (4.7, 4.8) show new stereo state vectors. These state vectors are not defined in the right image plane, nor in the left image plane. They are defined in the stereo space.
The equations (4.9, 4.10) show an equality between the 3D vectors v, w and the stereo state vectors s_Ω, s_V. That is important to define the kinematic of the stereo state vectors, which will be equivalent to the kinematic of v and w.

In the next section, we will discuss the constraints proposed to design these state vectors.

4.3.2 Discussion about the constraints

In the previous section, we proposed constraints on the norm of the image lines l_r, l_l. Here, one wants to know if these constraints are possible during all the servoing. The constraints are $l_r^\mathsf{T} \ \mathtt{K}_r \ t_r = -1$ and $l_l^\mathsf{T} \ \mathtt{K}_l \ t_l = 1$, what are satisfied if $l_r^\mathsf{T} \ \mathtt{K}_r \ t_r = 0$ or $l_l^\mathsf{T} \ \mathtt{K}_l \ t_l = 0$?

If $l_r^\mathsf{T} \ \mathtt{K}_r \ t_r = 0$, the equation (2.4) is $w \equiv \mathtt{R}_r^\mathsf{T} \ \mathtt{K}_r^\mathsf{T} \ l_r \ l_l^\mathsf{T} \ \mathtt{K}_l \ t_l$ but $l_l^\mathsf{T} \ \mathtt{K}_l \ t_l$ is a scalar, then $w \equiv \mathtt{R}_r^\mathsf{T} \ \mathtt{K}_r^\mathsf{T} \ l_r$.
We know that the vector normal of the plane generated by the right camera center and the image line l_r so the 3D straight line is given by $w_r \equiv \mathtt{K}_r^\mathsf{T} \ l_r$ in the right camera frame.
Transferring this vector to the stereo frame gives $w_r \equiv \mathtt{R}_r^\mathsf{T} \ \mathtt{K}_r^\mathsf{T} \ l_r = w$!!!
This means that this plane, generated by the right camera center and the 3D straight line, contains the stereo frame origin too.

The stereo frame is a virtual frame, so if we chose the stereo frame origin on the line between the two cameras, which means $C_r = -\alpha C_l$, the two cameras centers will be in the same plane with the 3d straight line in this case, and this is a singularity case excepted in this work.
With the same analysis for the second constraint $l_l^{\mathsf{T}}\ \mathtt{K}_l\ t_l = 0$, we conclude that these constraints are always valid.

4.3.3 The state vectors kinematics

From the equation (4.9) we can write

$$\begin{aligned} v &= m\mathtt{M}_r^{-1} s_\Omega \\ s_\Omega &= \frac{1}{m}\mathtt{M}_r v \\ \dot{s}_\Omega &= \frac{1}{m}\mathtt{M}_r \dot{v} \\ \dot{s}_\Omega &= \frac{1}{m}\mathtt{M}_r(\boldsymbol{\Omega} \times v) \ = \ -\frac{1}{m}\mathtt{M}_r[v]_\times \boldsymbol{\Omega} \end{aligned}$$

but, $[v]_\times \ = \ [m\mathtt{M}_r^{-1} s_\Omega]_\times \ = \ \mathtt{M}_r^{\mathsf{T}}[s_\Omega]_\times \mathtt{M}_r$, and $\mathtt{M}_r\mathtt{M}_r^{\mathsf{T}} = \mathtt{K}_r\mathtt{R}_r\mathtt{R}_r^{\mathsf{T}}\mathtt{K}_r^{\mathsf{T}} \ = \ \mathtt{K}_r\mathtt{K}_r^{\mathsf{T}}$, so

$$\dot{s}_\Omega = -\frac{1}{m}\mathtt{K}_r\mathtt{K}_r^{\mathsf{T}}\ [s_\Omega]_\times\ \mathtt{M}_r\ \boldsymbol{\Omega} \tag{4.11}$$

This equation shows the independency of the state vector s_Ω from the translation velocity.

Now, the equation (4.10), gives

$$\begin{aligned} w &= \mathtt{M}_r^{\mathsf{T}} s_V \\ s_V &= \mathtt{M}_r^{-\mathsf{T}} w \\ \dot{s}_V &= \mathtt{M}_r^{-\mathsf{T}} \dot{w} \\ \dot{s}_V &= \mathtt{M}_r^{-\mathsf{T}}(\boldsymbol{\Omega} \times w + \mathbf{V} \times v) \\ \dot{s}_V &= \ -\mathtt{M}_r^{-\mathsf{T}}[w]_\times \boldsymbol{\Omega} - \mathtt{M}_r^{-\mathsf{T}}[v]_\times \mathbf{V} \end{aligned}$$

but, $[w]_\times \ = \ [\mathtt{M}_r^{\mathsf{T}} s_\Omega]_\times \ = \ m\mathtt{M}_r^{-1}[s_\Omega]_\times \mathtt{M}_r^{-\mathsf{T}}$, so

$$\dot{s}_V = -m\mathtt{K}_r^{-\mathsf{T}}\mathtt{K}_r^{-1}[s_V]_\times \mathtt{M}_r^{-\mathsf{T}}\boldsymbol{\Omega} \ - \ m[s_\Omega]_\times \mathtt{M}_r \mathbf{V} \tag{4.12}$$

Equations (4.11) and (4.12) are useful to define a partial decoupled control laws for rotation and translation or sequential control as we will see in the next section.

4.4 Sequential control laws

The kinematic of the state vectors s_Ω, s_V given by the equations (4.11) and (4.12), verifies a partial decoupling between the rotation and the translation velocities. Therefore to design the control laws, we now start by firstly defining two task functions.
The difference between the state vector $s_\Omega(t)$ in the current stereo image, and the one s^*_Ω in the desired stereo image, defines the rotational task function for one 3D straight line as the following:

$$e_\Omega(t) = s_\Omega(t) - s^*_\Omega \ . \tag{4.13}$$

While with the translational task function, one can do similar choice considering the error between the current state vector $s_V(t)$ and the desired one s^*_V for one 3D straight line too

$$e_V(t) = s_V(t) - s^*_V \ . \tag{4.14}$$

With these task functions, the rotational and translational control laws can now be designed separately.

4.4.1 The rotation control law

Supposed an exponential decrease to the rotation task function given in (4.13), which means $\dot{e}_\Omega = -\lambda_\Omega \ e_\Omega$. This gives $\dot{s}_\Omega = -\lambda_\Omega \ e_\Omega$ as $\dot{s}^*_\Omega = 0, \lambda_\Omega \in \mathbb{R}^+$.

$$\dot{s}_\Omega = -\frac{1}{m} \mathtt{K}_r \mathtt{K}_r^\mathsf{T} \ [s_\Omega]_\times \ \mathtt{M}_r \ \boldsymbol{\Omega} \ = \ -\lambda_\Omega \ e_\Omega$$

Let $\mathtt{h}_\Omega = -\frac{1}{m} \mathtt{K}_r \mathtt{K}_r^\mathsf{T} \ [s_\Omega]_\times \mathtt{M}_r$, be a (3×3) matrix, then the rotation control law is given by

$$\boldsymbol{\Omega}(t) = -\lambda_\Omega \ \mathtt{h}^+_\Omega \ e_\Omega \ .$$

In the case of a unique line, the matrix $\mathtt{h}^+_\Omega$ does not exist. Let's define the task function and the interaction matrix for the general case of n lines. Then, we will look for the minimal number n of state vectors necessary to invert the interaction matrix.

Given n current state vectors $s_{\Omega 1}, ..., s_{\Omega n}$ and n desired ones $s^*_{\Omega 1}, ..., s^*_{\Omega n}$, the task function and the interaction matrix are

$$e_\Omega = \begin{pmatrix} s_{\Omega 1} - s^*_{\Omega 1} \\ \vdots \\ s_{\Omega n} - s^*_{\Omega n} \end{pmatrix} \qquad \mathtt{H}_\Omega = \begin{bmatrix} -\frac{1}{m} \mathtt{K}_r \mathtt{K}_r^\mathsf{T} \ [s_{\Omega 1}]_\times \mathtt{M}_r \\ \vdots \\ -\frac{1}{m} \mathtt{K}_r \mathtt{K}_r^\mathsf{T} \ [s_{\Omega n}]_\times \mathtt{M}_r \end{bmatrix}$$

The new form of the rotational control law is

$$\boldsymbol{\Omega} = -\lambda_\Omega \, \mathtt{H}_\Omega^+ \, e_\Omega \tag{4.15}$$

This control law can be defined only if the matrix $\mathtt{H}_\Omega^+ = (\mathtt{H}_\Omega^\mathsf{T}\mathtt{H}_\Omega)^{-1}\mathtt{H}_\Omega^\mathsf{T}$ exists.

4.4.2 Stability and convergence of rotation control law

To prove the stability of the closed loop system with the rotation control law defined in (4.15), and the convergence of the task function (4.13), we use the Lyapunov function $\mathcal{L}(t) = 1/2||e_\Omega(t)||^2$ and we study the sign of its time derivative:

$$\begin{aligned}
\dot{\mathcal{L}}(t) &= \dot{e}_\Omega^\mathsf{T} e_\Omega \\
&= \begin{pmatrix} \dot{s}_{\Omega 1} \\ \vdots \\ \dot{s}_{\Omega n} \end{pmatrix}^\mathsf{T} e_\Omega \\
&= \begin{pmatrix} -\frac{1}{m}\mathtt{M}_r \; [v_1]_\times \boldsymbol{\Omega} \\ \vdots \\ -\frac{1}{m}\mathtt{M}_r \; [v_n]_\times \boldsymbol{\Omega} \end{pmatrix}^\mathsf{T} e_\Omega \\
&= (\mathtt{H}_\Omega \boldsymbol{\Omega})^\mathsf{T} e_\Omega \;=\; \boldsymbol{\Omega}^\mathsf{T} \mathtt{H}_\Omega^\mathsf{T} \, e_\Omega
\end{aligned}$$

We have $\boldsymbol{\Omega} = -\lambda_\Omega \, \mathtt{H}_\Omega^+ \, e_\Omega$ so $\boldsymbol{\Omega}^\mathsf{T} = -\lambda_\Omega \, e_\Omega^\mathsf{T} \, \mathtt{H}_\Omega \, (\mathtt{H}_\Omega^\mathsf{T} \, \mathtt{H}_\Omega)^{-1}$, then

$$\dot{\mathcal{L}}(t) = -\lambda_\Omega \, e_\Omega^\mathsf{T} \, \mathtt{H}_\Omega \, (\mathtt{H}_\Omega^\mathsf{T} \, \mathtt{H}_\Omega)^{-1} \, \mathtt{H}_\Omega^\mathsf{T} \, e_\Omega$$

To look for the sign of $\dot{\mathcal{L}}(t)$ the matrix $\mathtt{H}_\Omega^\mathsf{T} \, \mathtt{H}_\Omega$ must be invertible and its inverse should be positive definite, but it is known that any positive definite matrix is invertible, and its inverse is positive definite too, so it is enough to prove that $\mathtt{H}_\Omega^\mathsf{T} \, \mathtt{H}_\Omega$ is positive definite.

$$\begin{aligned}
\mathtt{H}_\Omega^\mathsf{T}\mathtt{H}_\Omega &= \frac{1}{m^2} \begin{bmatrix} \mathtt{M}_r \; [v_1]_\times \\ \vdots \\ \mathtt{M}_r \; [v_n]_\times \end{bmatrix}^\mathsf{T} \begin{bmatrix} \mathtt{M}_r \; [v_1]_\times \\ \vdots \\ \mathtt{M}_r \; [v_n]_\times \end{bmatrix} \\
&= -\frac{1}{m^2} \sum_i [v_i]_\times \mathtt{M}_r^\mathsf{T} \mathtt{M}_r [v_i]_\times
\end{aligned}$$

Therefore, for any vector $a \in \mathbb{R}^{*3}$, we have

$$\begin{aligned} a^{\mathsf{T}}\ \mathtt{H}_{\Omega}^{\mathsf{T}}\mathtt{H}_{\Omega}\ a &= -a^{\mathsf{T}} \sum_i [v_i]_{\times}\mathtt{M}_r^{\mathsf{T}}\mathtt{M}_r[v_i]_{\times}a \\ &= -\sum_i a^{\mathsf{T}}[v_i]_{\times}\mathtt{M}_r^{\mathsf{T}}\mathtt{M}_r[v_i]_{\times}a \\ &= \sum_i (\mathtt{M}_r[v_i]_{\times}a)^{\mathsf{T}}\ \mathtt{M}_r[v_i]_{\times}a \\ &= \sum_i ||\mathtt{M}_r\ v_i \times a||^2 \end{aligned}$$

So $\mathtt{H}_{\Omega}^{\mathsf{T}}\mathtt{H}_{\Omega}$ is positive definite if there are at least two non-parallel vectors v_i, v_j, which means that $(\mathtt{H}_{\Omega}^{\mathsf{T}}\mathtt{H}_{\Omega})^{-1}$ always exists and is positive definite too.
Now the part $e_{\Omega}^{\mathsf{T}}\ \mathtt{H}_{\Omega}\ \mathtt{H}_{\Omega}^{\mathsf{T}}\ e_{\Omega}$ is positive semi-definite because any matrix of the form $\mathtt{A}\mathtt{A}^{\mathsf{T}}$ is positive (semi definite) which means that the control law is globally stable.

At this stage, to study the convergence, we must look for the equilibrium points.

$$\begin{aligned} e_{\Omega}^{\mathsf{T}}\mathtt{H}_{\Omega} &= -\frac{1}{m}\begin{pmatrix} s_{\Omega 1} - s_{\Omega 1}^* \\ \vdots \\ s_{\Omega n} - s_{\Omega n}^* \end{pmatrix}^{\mathsf{T}} \begin{bmatrix} \mathtt{M}_r\ [v_1]_{\times} \\ \vdots \\ \mathtt{M}_r\ [v_n]_{\times} \end{bmatrix} \\ &= -\frac{1}{m}\sum_i e_i^{\mathsf{T}}\mathtt{M}_r[v_i]_{\times} \qquad \text{with} \quad e_i = s_{\Omega i} - s_{\Omega i}^* \end{aligned}$$

and $\mathtt{H}_{\Omega}^{\mathsf{T}}e_{\Omega} = \frac{1}{m}\sum_i [v_i]_{\times}\mathtt{M}_r^{\mathsf{T}}e_i$.
Finally, the scalar $e_{\Omega}^{\mathsf{T}}\ \mathtt{H}_{\Omega}\mathtt{H}_{\Omega}^{\mathsf{T}}\ e_{\Omega} = \frac{1}{m^2}\left|\left|\sum_i [v_i]_{\times}\mathtt{M}_r^{\mathsf{T}}e_i\right|\right|^2 \geq 0$, which means $\dot{\mathcal{L}} \leq 0$. It is null when $e_i = 0$ for all i, but we can not guarantee that it is unique. There are maybe other local minima (equilibriums points), which verifies the equation $\sum_i [v_i]_{\times}\mathtt{M}_r^{\mathsf{T}}e_i = 0$. So the control law $\boldsymbol{\Omega} = \lambda_{\Omega}\ m\ \mathtt{H}_{\Omega}^{+}\ e_{\Omega}$ globally stabilizes the system in the closed loop, but the asymptotic convergence of the rotational servoing error is not ensured.

4.4.3 The translation control law

In the sequential control, we suppose that the translation servoing is started after the rotational servoing was finished, and the current rotation state vectors are converging toward the desired state vectors.
The translation task function given in (4.14), with an exponential decrease supposed to this function ($\dot{e}_V = -\lambda_V\ e_V$) give $\dot{s}_V = -\lambda_V\ e_V$, as $\dot{s}_V^* = 0$.

$$\begin{aligned} \dot{s}_V &= -\lambda_V\ e_V \\ \dot{s}_V &= -m[s_{\Omega}]_{\times}\mathtt{M}_r\mathbf{V} \qquad \text{because} \quad \boldsymbol{\Omega} = 0 \end{aligned}$$

Let the (3×3) matrix $\mathtt{h}_V = -m[s_\Omega]_\times \mathtt{M}_r$ then, the translation control law is given by

$$\mathbf{V} = -\lambda_V \ \mathtt{h}_V^+ \ e_V \ .$$

Again, for a unique line (one state vector), the matrix $\mathtt{h}_V^+$ did not exist. So let's define the task function and the interaction matrix for the case of n lines.
For n current state vectors $s_{V1}, ..., s_{Vn}$ and their desired values $s^*_{V1}, ..., s^*_{Vn}$. The task function and the interaction matrix are

$$e_V = \begin{pmatrix} s_{V1} - s^*_{V1} \\ \vdots \\ s_{Vn} - s^*_{Vn} \end{pmatrix} \qquad \mathtt{H}_V = \begin{bmatrix} -m[s_{\Omega 1}]_\times \mathtt{M}_r \\ \vdots \\ -m[s_{\Omega n}]_\times \mathtt{M}_r \end{bmatrix}$$

Then, the translational control law is

$$\mathbf{V} = -\lambda_V \ \mathtt{H}_V^+ e_V \ . \tag{4.16}$$

This control law can be defined if the matrix $\mathtt{H}_V^+ = (\mathtt{H}_V^\mathsf{T}\mathtt{H}_V)^{-1}\mathtt{H}_V^\mathsf{T}$ exists.

4.4.4 Stability and convergence of translation control law

To prove the stability of the closed loop system with the translation control law defined in (4.16), and the convergence of the task function (4.14), we use the Lyapunov function $\mathcal{L}(t) = 1/2||e_V(t)||^2$ and study the sign of its time derivative:

$$\begin{aligned}
\dot{\mathcal{L}}(t) &= \dot{e}_V^\mathsf{T} e_V \\
&= (\dot{s}_{v1}, \ldots, \dot{s}_{vn})^\mathsf{T} \, e_V \\
&= \begin{pmatrix} -\mathtt{M}_r^{-\mathsf{T}} \ [v_1]_\times \mathbf{V} \\ \vdots \\ -\mathtt{M}_r^{-\mathsf{T}} \ [v_n]_\times \mathbf{V} \end{pmatrix}^\mathsf{T} e_V \\
&= -(\mathtt{H}_V \mathbf{V})^\mathsf{T} e_V \\
&= -\mathbf{V}^\mathsf{T} \mathtt{H}_V^\mathsf{T} e_V
\end{aligned}$$

Given $\mathbf{V} = \lambda_V \ \mathtt{H}_V^+ \ e_V$, $\mathbf{V}^\mathsf{T} = \lambda_V \ e_V^\mathsf{T} \ \mathtt{H}_V \ (\mathtt{H}_V^\mathsf{T}\mathtt{H}_V)^{-1}$, then

$$\dot{\mathcal{L}}(t) \ = \ -\lambda_V \ e_V^\mathsf{T} \ \mathtt{H}_V \ (\mathtt{H}_V^\mathsf{T}\mathtt{H}_V)^{-1} \ \mathtt{H}_V^\mathsf{T} \ e_V \ .$$

To look for the sign of the time derivative of Lyapunov function, we started with $(\mathtt{H}_V^\mathsf{T}\mathtt{H}_V)$, and similarly to the rotation case, it is positive definite, if there are at least

two non-parallel lines in the set of the object lines.
Now the part $e_V^\top \ \mathtt{H}_V \mathtt{H}_V^\top \ e_V$ is positive semi-definite which means that the control closed loop is globally stable.
Now to study the convergence, we seek for the equilibrium points

$$\begin{aligned} e_V^\top \mathtt{H}_V &= \begin{pmatrix} s_{V1} - s_{V1}^* \\ \vdots \\ s_{Vn} - s_{Vn}^* \end{pmatrix}^\top \begin{bmatrix} \mathtt{M}_r^{-\top} \ [v_1]_\times \\ \vdots \\ \mathtt{M}_r^{-\top} \ [v_n]_\times \end{bmatrix} \\ &= -\sum_i e_i^\top \mathtt{M}_r^{-\top} [v_i]_\times \end{aligned}$$

and $\mathtt{H}_V^\top \ e_V = \sum_i [v_i]_\times \mathtt{M}_r^{-1} \ e_i$.
Finally,

$$\begin{aligned} e_V^\top \mathtt{H}_V \mathtt{H}_V^\top e_V &= \Big|\Big|\sum_i [v_i]_\times \mathtt{M}_r^{-1} e_i\Big|\Big|^2 \ \geq \ 0 \\ \dot{\mathcal{L}} &\leq \ 0 \ . \end{aligned}$$

Like for the rotational control law, the task function converges towards zero, but may stop at another local minimum, an equilibrium point verifies $\sum_i [v_i]_\times \mathtt{M}_r^{-1} e_i = 0$.

4.4.5 Simulations for sequential control laws

This section presents the simulation results of the system behaviour in the closed loop, using the translational and rotational control laws (equations 4.16, 4.15)

$$\begin{aligned} \mathbf{V} &= \lambda_V \ \mathtt{H}_V^+ e_V \\ \mathbf{\Omega} &= \lambda_\Omega \ m \ \mathtt{H}_\Omega^+ \ e_\Omega \end{aligned}$$

As the simulation does not interested of the robot dynamics, the robot is considered as a perfect integrator. The stereo system consists of two parallel identical cameras. The object of interest consists of three non coplanar straight lines. The intrinsic parameters of the two cameras are same as the previous simulations in section 4.2.

The goal of the simulations is to verify the stability and the convergence of the closed loop system in the perfect case, and with the presence of a white noise. Many simulations have been done, with large rotation and translation between the initial and final state vectors, all the results show the GAS, even that the theoretical study did not prove that. But we will present one simulations (with and without noise), as no interest of showing many simulations with the same results.
In all simulations, the gains are set to $\lambda_V = \lambda_\Omega = 0.15$. Figure (4.7) illustrates an

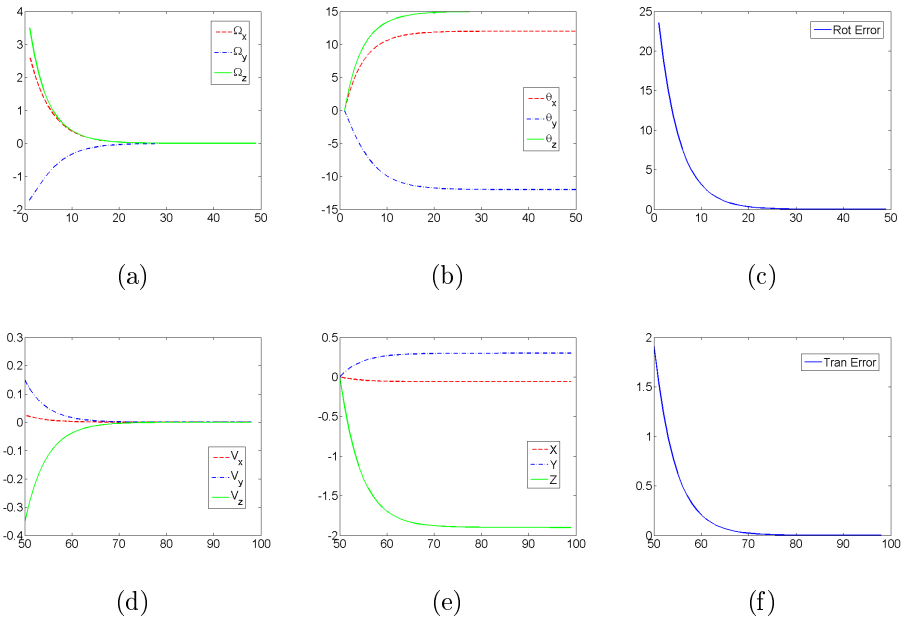


Figure 4.7: *Simulation results for the SVS with sequential control, rotational servoing first, then translational servoing. The object of interest consists of three 3D straight lines. The horizontal axis of the figures is the iteration number. a) Rotational velocity (deg/s). b) Rotation angles (deg). c) Rotation error (task function e_Ω). d) Translation velocity (m/s). e) Translation values (m). f) Translation error (task function e_V). The rotational and translational errors (the task functions) are null at the end of servoing, which means that the system asymptotically converges.*

example with large rotation and translation. In the first phase of the simulation, the rotation was controlled alone, then the translation servoing is started $\lambda_\Omega = 0$.

Figures (4.7 - a, d) show the rotational and translational velocities. One can see the difference from the case of the coupled control laws, and how much the system is close to a first-order dynamic system. The final pose is shown in figures (4.7 - b, e). It is really a large rotation and a large translation achieved by the system (the trajectory traveled by the end effector is very long).
Finally, figures (4.7 - c, f) show the asymptotic convergence of the rotation and the translation task functions. The state vectors are converging toward their desired values.

This simulation is repeated with the presence of white noise in the image. We add a random value (less than one pixel) for each point in the image lines, then, the image lines are estimated by the least square method. The results are figured in (fig. 4.8). Figures (4.8 - a, d) show the rotational and translational velocities. One can

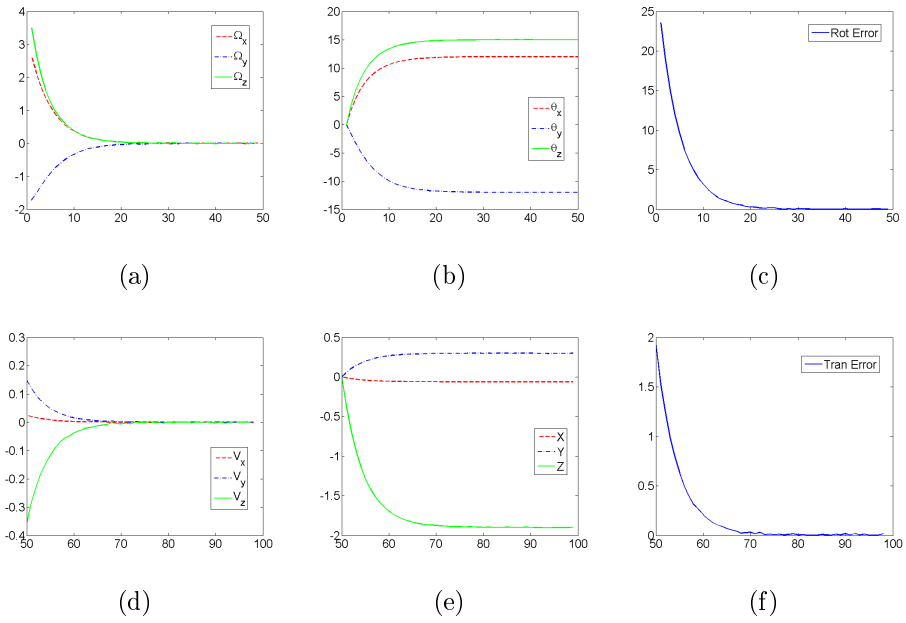


Figure 4.8: *Simulation results for the SVS with sequential control, rotational servoing first, then translational servoing. Case of image with white noise. The object of interest consists of three 3D straight lines. The horizontal axis of the figures is the iteration number. a) Rotational velocity (deg/s). b) Rotation angles (deg). c) Rotation error (task function e_Ω). d) Translation velocity (m/s). e) Translation values (m). f) Translation error (task function e_V). The rotational and translational errors (the task functions) are almost null at the end of servoing, which means that the system asymptotically converges.*

see that the noise did not affect the stability of the system, and the one can notice that these sequential control laws are more robust against the white noise than the coupled control laws experimented in section 4.2. The final pose is shown in figures (4.8 - b, e), these figures show the large rotation and the large translation achieved by the end effector of the robot.

Finally, figures (4.8 - c, f) show the asymptotic convergence of the rotational and translational task functions. The error in the translational task function is more affected by the noise, that because the error of rotation task function is not null at the end of the rotation servoing.

With this simulation, we finish the sequential control laws, and now, we will look for state vectors to achieve the full decoupled control laws.

4.5 Decoupled control laws

The sequential control laws need to divide the visual servoing in two phases. In the first phase, the rotational control law works to servo the rotation error, after that, the second phase starts with the translational control law to serve the translation error. This takes almost twice the time for servoing, and as the simulation show, the error in the rotation task function accumulates in the translation error. These reasons show how it is important to design fully decoupled control laws.

The rotation control law defined in (4.15) is independent of the translational error, so it is suitable for the full decoupled servoing.
We need to define a translational state vector independent of the rotational error. To do that, recall the equations (4.10), (4.12)

$$\begin{aligned} w &= \mathtt{M}_r^{\mathsf{T}} s_V \\ \dot{s}_V &= -\mathtt{M}_r^{-\mathsf{T}}[w]_\times \boldsymbol{\Omega} - \mathtt{M}_r^{-\mathsf{T}}[v]_\times \mathbf{V} \end{aligned}$$

developing this equation yields

$$\begin{aligned} \mathtt{M}_r^{\mathsf{T}} s_V &= w \\ \mathtt{M}_r^{\mathsf{T}} \dot{s}_V &= \boldsymbol{\Omega} \times w + \mathbf{V} \times v \\ (\mathtt{M}_r^{\mathsf{T}} s_V)^{\mathsf{T}} (\mathtt{M}_r^{\mathsf{T}} \dot{s}_V) &= w^{\mathsf{T}} (\boldsymbol{\Omega} \times w + \mathbf{V} \times v) \\ (\mathtt{M}_r^{\mathsf{T}} s_V)^{\mathsf{T}} (\mathtt{M}_r^{\mathsf{T}} \dot{s}_V) &= (v \times w)^{\mathsf{T}} \mathbf{V} \\ s_V^{\mathsf{T}}\, \mathtt{M}_r \mathtt{M}_r^{\mathsf{T}}\, \dot{s}_V &= u^{\mathsf{T}} \mathbf{V} \end{aligned}$$

But $\mathtt{M}_r\mathtt{M}_r^{\mathsf{T}} = \mathtt{K}_r\mathtt{R}_r\mathtt{R}_r^{\mathsf{T}}\mathtt{K}_r^{\mathsf{T}} = \mathtt{K}_r\mathtt{K}_r^{\mathsf{T}}$, so $s_V^{\mathsf{T}}\mathtt{K}_r\mathtt{K}_r^{\mathsf{T}}\dot{s}_V = u^{\mathsf{T}}\mathbf{V}$
$\mathtt{K}_r\mathtt{K}_r^{\mathsf{T}} = \omega^{-1}$ is the image of the absolute conic placed at the plane at infinity π_∞. Using

$$s = \frac{1}{2}||\mathtt{K}_r^{\mathsf{T}} s_V||^2 \tag{4.17}$$

$$\dot{s} = s_V^{\mathsf{T}}\mathtt{K}_r\mathtt{K}_r^{\mathsf{T}}\dot{s}_V = u^{\mathsf{T}}\mathbf{V} \tag{4.18}$$

We can define the translation task function $e = s - s^*$. Assuming an exponential decrease of the error ($\dot{e} = -\lambda e$) defining a translation control law in the case of n state vectors

$$\mathbf{V} = -\lambda\, \mathtt{H}^{+}\, e \tag{4.19}$$

with

$$e = \begin{pmatrix} s_1 - s_1^* \\ \vdots \\ s_n - s_n^* \end{pmatrix} \qquad \mathtt{H} = \begin{bmatrix} u_1^{\mathsf{T}} \\ \vdots \\ u_n^{\mathsf{T}} \end{bmatrix}$$

This control law is independent of the rotational error. So, one can start the servoing for both control laws simultaneously.

4.5.1 Stability and convergence of decoupled translation control law

Considering the Lyapunov function $\mathcal{L}(t) = 1/2||e||^2$, its time derivative $\dot{\mathcal{L}}(t)$ is

$$\begin{aligned}
\dot{\mathcal{L}}(t) &= \dot{e}^{\mathsf{T}} e \\
&= \begin{pmatrix} \dot{s}_1 & \cdots & \dot{s}_n \end{pmatrix} e \\
&= \begin{pmatrix} u_1^{\mathsf{T}}\mathbf{V} & \cdots & u_n^{\mathsf{T}}\mathbf{V} \end{pmatrix} e \\
&= (\mathtt{H}\mathbf{V})^{\mathsf{T}} e \\
&= \mathbf{V}^{\mathsf{T}}\mathtt{H}^{\mathsf{T}} e
\end{aligned}$$

With $\mathbf{V} = -\lambda\, \mathtt{H}^{+} e$, we find $\mathbf{V}^{\mathsf{T}} = -\lambda e^{\mathsf{T}}\mathtt{H}(\mathtt{H}^{\mathsf{T}}\mathtt{H})^{-1}$ so

$$\dot{\mathcal{L}}(t) = -\lambda\, e^{\mathsf{T}}\mathtt{H}(\mathtt{H}^{\mathsf{T}}\mathtt{H})^{-1}\mathtt{H}^{\mathsf{T}} e\ .$$

To study the sign of the time-derivative of the Lyapunov function, we start with $\mathtt{H}^{\mathsf{T}}\mathtt{H}$. In fact, one can easily see that $\mathtt{H}^{\mathsf{T}}\mathtt{H} = \sum_i u_i u_i^{\mathsf{T}}$. Furthermore, for any vector $a \in \mathbb{R}^3$, we have

$$\begin{aligned}
a^{\mathsf{T}}\mathtt{H}^{\mathsf{T}}\mathtt{H}a &= a^{\mathsf{T}} \sum_i u_i u_i^{\mathsf{T}} a = \sum_i a^{\mathsf{T}} u_i u_i^{\mathsf{T}} a \\
&= \sum_i ||a^{\mathsf{T}} u_i||^2
\end{aligned}$$

which is positive definite if and only if there are three non coplanar vectors u_i. Now, computing $e^{\mathsf{T}}\mathtt{H}\mathtt{H}^{\mathsf{T}} e$:

$$\begin{aligned}
e^{\mathsf{T}}\mathtt{H} &= \begin{pmatrix} s_1 - s_1^* \\ \vdots \\ s_n - s_n^* \end{pmatrix}^{\mathsf{T}} \begin{pmatrix} u_1^{\mathsf{T}} \\ \vdots \\ u_n^{\mathsf{T}} \end{pmatrix} \\
&= \sum_i e_i\ u_i^{\mathsf{T}} \ \text{ as } \ e_i = s_i - s_i^* \ \text{ is a scalar.}
\end{aligned}$$

and, $\mathtt{H}^{\mathsf{T}} e = \sum_i e_i\ u_i$ So

$$\begin{aligned}
e^{\mathsf{T}}\mathtt{H}\mathtt{H}^{\mathsf{T}} e &= ||\sum_i e_i u_i||^2 \ \geq \ 0 \\
\Longrightarrow \dot{\mathcal{L}}(t) &\leq 0
\end{aligned}$$

The time derivative of the Lyapunov function is not negative definite, because of the dimension of the input vector $(s_1, \ldots, s_n)^\top$. If we consider an object with only three visual features $n = 3$ (or if we select three of the most relevant ones), we get an interaction matrix $\mathtt{H}$ of dimensions 3×3, then $\mathtt{H}^+ = \mathtt{H}^{-1}$ and the translation control law is $\mathbf{V} = -\lambda\ \mathtt{H}^{-1}\ e$. The Lyapunov derivative provides $\dot{\mathcal{L}}(t) = -\lambda e^\top e < 0$, which means the convergence of the task function e is asymptotic.
Finally, we note that we need only three state vectors to define this translation control law.

4.5.2 Simulations for decoupled control laws

This section presents the simulation results of the system behaviour in the closed loop, after applied the translational and rotational control laws (equations 4.19, 4.15)

$$\begin{aligned} \mathbf{V} &= \lambda\ \mathtt{H}^+ e \\ \mathbf{\Omega} &= \lambda_\Omega\ m\ \mathtt{H}_\Omega^+\ e_\Omega \end{aligned}$$

As the simulation does not interested of the robot dynamics, the robot is considered as a perfect integrator. The stereo system consists of two parallel identical cameras. The object of interest consists of three non coplanar straight lines. The intrinsic parameters of the two cameras are the same as for the previous simulations in section 4.2.

The goal of the simulations is to verify the stability and the convergence of the closed loop system in the perfect case, and with the presence of a white noise. Many simulations have been done, with large rotation and translation between the initial and final state vectors, all the results are in accordance with the GAS, even that the theoretical study did not prove that. Again we present only one (with and without noise).
In all simulations, the gains of the control laws are set to $\lambda = \lambda_\Omega = 0.15$.

An example of full decoupled SVS with large rotation and translation is illustrated in figure (4.9). In this simulation, both of rotation and translation servoing are started together, simultaneously.
Figures (4.9 - a, d) show the rotational and translational velocities, one can see all the velocity components decrease to zero, their time evaluation are close to a first-order system. The large rotation (about $60deg$) and the large translation (about $15m$) achieved by the system are shown in (4.9 - b, e). Finally the asymptotic convergence of the rotational and translational task functions without static error

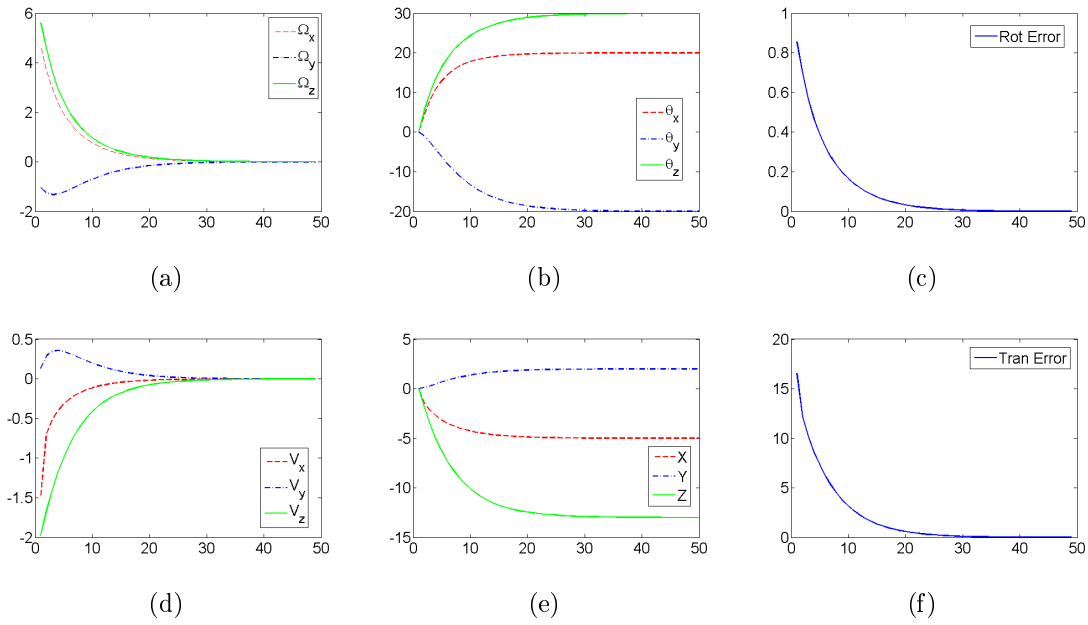


Figure 4.9: *Simulation results for the full decoupled SVS. The object of interest consists of three 3D straight lines. The horizontal axis of the figures is the iteration number. a) Rotational velocity (deg/s). b) Rotation angles (deg). c) Rotation error (task function e_Ω). d) Translation velocity (m/s). e) Translation values (m). f) Translation error (task function e). The rotational and translational errors (the task functions) are null at the end of servoing, which means that the system asymptotically converges.*

is shown in (4.9 - c, f).

This simulation is repeated with adding a white noise to the image measurements $(l_{ri}, l_{ri}, i = 1, 2, 3)$ in the same way presented in previous section 4.2. The results are shown in figure (4.10).

Figures (4.10 - a, d) show the rotational and translational velocities. The noise does not affect the stability of the system. The Euclidean final pose is displayed in figures (4.10 - b, e), to see the large rotation and translation achieved by the end effector of the robot in this simulation.

Finally the asymptotic convergence of the rotational and translational task functions with a small error due to the noise is shown in (4.10 - c, f).

With this simulations, we finish the stereo visual servoing (SVS) using two types of control laws, the first is the sequential control laws which are partially decoupled, the second is fully decoupled control laws. With the decoupled control laws, the global stability of the system in closed loop is proved, and the system behaviour is

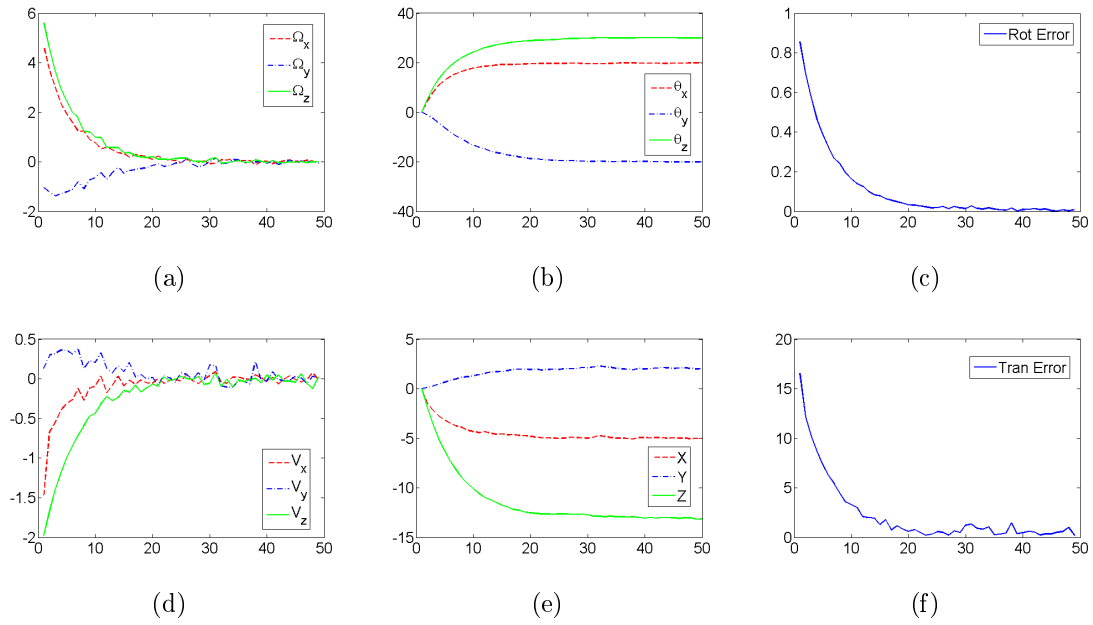


Figure 4.10: *Simulation results for the full decoupled SVS after adding a white noise to the image lines. The object of interest consists of three 3D straight lines. The horizontal axis of the figures is the iteration number. a) Rotational velocity (deg/s). b) Rotation angles (deg). c) Rotation error (task function e_Ω). d) Translation velocity (m/s). e) Translation values (m). f) Translation error (task function e). The rotational and translational errors (the task functions) are almost null at the end of servoing, which means that the system asymptotically converges.*

close to a first-order system. We did not prove the asymptotic convergence of the task functions, but we have derived the formula of the equilibrium points. Simulations with large rotation and large translations have been done without convergence toward a local minimum, we always get an asymptotic convergence.

4.6 Conclusion

This chapter is addressing the IBVS with sterovision and 3D lines. First, we started with the classical IBVS from lines, using interaction matrix defined by Chaumette, and using a stereo interaction matrix defined in this chapter. The idea was to define the velocity screw of the stereo system by $\tau = -\lambda\ \mathtt{L_I}^{+}\mathtt{e}$.
The simulations results are not convincing. There was minima local, large rotation or large translation are not possible, analytical solution or theoretical prove of stability and convergence are very difficult with the high dimensions of the interaction matrices.
These reasons creates a real need of new interaction matrices, with small dimensions, easy to develop for obtaining its pseudo-inverse.

The solution is the decoupling between the rotational and translational velocities control laws. The classical IBVS is converted into stereo visual servoing (SVS). The pair of images (stereo image) forms together the space of work. New state vectors are defined from a combination of visual measurements in the stereo image and thanks to the infinite homography.
For that end, the Euclidean model of the camera which contents the intrinsic and extrinsic parameters is replaced by infinite homography $\mathtt{H}_\infty$ and the Fundamental matrix $\mathtt{F}$.
In the stereo space, the new state vectors s_Ω, s_V are proposed to define partially decoupled control laws for rotation and translation.

$$\begin{aligned} \mathbf{V} &= \lambda_V\ \mathtt{H}_V^{+}\ e_V \\ \mathbf{\Omega} &= \lambda_\Omega\ m\ \mathtt{H}_\Omega^{+}\ e_\Omega \end{aligned}$$

The global stability of the system in closed loop are proved, while the asymptotic convergence was not proved, but the analytic expression of the equilibrium points has been provided.
The simulations show the asymptotic convergence of the state vectors towards their desired value, even with large rotations and large translations.

The work is continued with the definition of a translational state vector and a translational task function independent of rotation error. This task function allows to design a translational control law fully decoupled to the rotational control law.

$$\mathbf{V} = -\lambda\ \mathtt{H}^{+}\ e$$

Simulations with large rotation and large translation are done for all the control laws presented in this chapter and the results are illustrated in every section.

Chapter 5

Conclusions and Perspectives

In this final chapter, a summary of the thesis is given. Then some future works are highlighted. The main theoretical results are presented according to their position in the thesis.

5.1 Conclusions

The work we have presented here tried to solve the problems presented in the introduction. It concerns the decoupling between the rotational and translational control laws, for stereo vision-based robot control. To this end, we investigated the stereo visual servoing from straight lines. The work is addressing the two famous kinds of visual servoing, position based visual servoing (PBVS) and image based visual servoing (IBVS), to define new velocities control laws and to look for a better behaviour of the servoing.
After denoting the problem of decoupling control in the introduction, we illustrate some interesting works in the literature, which show the importance of video servoing in the vision applications and mobile robots. Then some works related to the geometrical modeling and the kinematics of a 3D straight line are reported. We finished the chapter with some works about the decoupling control laws.

Our work started in chapter 3, in this chapter, the object of interest is represented by 3D straight lines, and we select the Plücker coordinates to represent these 3D lines. We gives some explanations about the choice of this representation among others.

The 3D information is needed in PBVS, and in IBVS. These 3D measurements are always available in our servoings thanks to the stereo vision, while all the drawbacks of using stereo system like enormeous data flow and system calibration are

overcome nowadays. For example, in our practical experiments, we have acquired 75fps from each camera, let's say 150 images by second, of dimensions 600×400. The image processing and the robot controller are executed in real time.

The stereo system enabled to decrease the necessary number of 3D straight lines (the features) to define the 6 DOF of a rigid real object in the Euclidean space. Even for an object of interest composed with only one straight line, with a stereo rig, we succeeded in servoing the relative position (Object - stereo rig) with less degrees of freedom.
We did not focused the work on the calibration of the camera parameters, nor the calibration of the stereo system, but the effect of the calibration errors are included in the simulations results for every control law designed according to the PBVS approach. It is important to get an accurate calibrated system, especially for practical experiments with PBVS, but we voluntarily considered that calibration is out of the scope of this thesis work.

In this chapter we provide a survey on the PBVS techniques, starting by computing the transformation matrix between two Euclidean position of a 3D straight line, then use the (classical) "Euclidean control laws"

$$\begin{aligned} \mathbf{V} &= -\lambda_V \, \mathtt{R}^\mathsf{T} \, t \ , \\ \mathbf{\Omega} &= -\lambda_\Omega \, \theta u \end{aligned}$$

to control the camera motion velocities with respect to a rigid object and for servoing the relative pose (Object - stereo rig). The independency between the rotational and the translational state vectors ($s_\Omega = \theta \ u$, $s_V = \mathtt{R}^\mathsf{T} \ t$), implies the decoupling between the rotational and the translational velocities control laws. These velocities control laws are verifying the global asymptotic stability (GAS) of the system in the closed loop.

The Plücker coordinates of a 3D straight line can used directly in the state vectors. The rotational state vector is the direction vector v, $s_\Omega = v$, and the translational state vector is the orthogonal distance d between the straight line and the frame origin (or the norm of the vector w), $s_V = d$. Then two independent task functions have been designed according to the number of state vectors, that is the number of required visual features. This independency in the task functions allows the design of fully decoupled velocities control laws:

$$\begin{aligned} \mathbf{V} &= -\lambda_V \, \mathtt{H}_V^+ \, e_V \ , \\ \mathbf{\Omega} &= -\lambda_\Omega \, \mathtt{H}_\Omega^+ \, e_\Omega \ , \end{aligned}$$

with

$$\mathtt{H}_V = \begin{bmatrix} u_1^\mathsf{T} \\ \vdots \\ u_n^\mathsf{T} \end{bmatrix} \quad , \quad \mathtt{H}_\Omega = \begin{bmatrix} -[v_1]_\times \\ \vdots \\ -[v_n]_\times \end{bmatrix} ,$$

and

$$e_V = \begin{pmatrix} d_1 - d_1^* \\ \vdots \\ d_n - d_n^* \end{pmatrix} \quad , \quad e_\Omega = \begin{pmatrix} v_1 - v_1^* \\ \vdots \\ v_n - v_n^* \end{pmatrix} \quad .$$

These two control laws are called "Plücker coordinates control laws".
The global asymptotic stability (GAS) of the system provided with the "Plücker coordinates control laws" is proved with the help of Lyapunov theory. The proof was possible and simple to establish, due to the decoupling between the rotational and translational velocities control laws, which reduces the dimensions of the state representation of the system in the closed loop.

Simulation results confirmed the theoretical studies. They confirm the GAS of the system in the closed loop for many configurations. Adding a white noise to the line parameters does not affect much the asymptotic stability of the system, the final precision is largely depending on the number of features used in the estimations of the inverse of the interaction matrix. These results prove the robustness of these control laws against the noise and the errors of features estimation.
The "Euclidean control laws" and the "Plücker coordinates control laws" are applied to a 6 DOF robot manipulator of type Adept Viper S650. This robot has 6 axes (6 revolute joints), which means 6 DOF in the joint space. As it is a full anthropomorphic manipulator, it has also 6 DOF in the Euclidean space. The goal of the experiments were to control the relative pose between the robot end-effector (which holds the stereo rig) and the object of interest. The results were very similar to those of simulations with a synthetic white noise.

The PBVS applied to a non-rigid object is considered as an interesting application for the stereo PBVS presented in this chapter. First, we designed the velocity control laws for an object composed with only one 3D straight line:

$$\begin{aligned} \boldsymbol{\Omega} &= -\lambda_\Omega \, \mathtt{H}_\Omega^+ e_\Omega \quad , \\ \mathbf{V} &= -\lambda_V \, (d - d^*) u \end{aligned}$$

with

$$e_\Omega = \begin{pmatrix} v - v^* \\ w - w^* \end{pmatrix} \qquad \text{and} \qquad \mathtt{H}_\Omega = \begin{bmatrix} -[v]_\times \\ -[w]_\times \end{bmatrix} \quad .$$

These velocities control laws are fully decoupled and verify the GAS of the system in closed loop. That was proved through the Lyapunov theory then shown by simulations and experiments on the robot Viper S650.

Second, we addressed the visual servoing of an articulated object. The goal was to control both the joints of an articulated object and the relative pose (articulated object - stereo rig) simultaneously. For this application, we have considered a simple articulated object, consisting of two 3D straight lines and one revolute joint. This object is modelled with 7 DOF, in the Euclidean space. 6 DOF were concerning the relative pose (articulated object - stereo rig) to servo, and the seventh DOF is used to control the revolute joint of the object.
An additional task function is designed to define the error of the joint angle $e_\psi = \psi - \psi_\infty$. Then the angular velocity control law is given by $\mathbf{\Omega}_\psi = -\lambda_\psi e_\psi$.
The Rodrigues' rotation formula gives back the articulated object into rigid object after the estimation of the joint angle. This allows to turn to the "Euclidean control laws" for the position servoing of the object relative to the stereo rig.
The simulations and the experiments on a robot manipulator and a real articulated object show the global asymptotic stability of the visual closed loop and the angular closed loop.
For a more complex articulated object (with several articulations), we use the first two 3D lines to estimate the Euclidean transformation $[\mathtt{R}, t]$, then control the relative pose (articulated object - stereo rig) using the "Euclidean control laws", just like in the case of a single articulated object.
The simulations show the convergence of all the joint angles ψ_i towards their final values in the desired angular position, and the convergence of the robot end-effector toward the desired position and orientation.

The IBVS has been presented in the chapter 4. This chapter divides the IBVS in two categories, the classical IBVS, and the stereo IBVS which is noted as SVS.
In the IBVS, we look for the interaction matrix $\mathtt{L}$ and applied the velocity control law $\tau = -\lambda \mathtt{L}^+ e$ with e is the task function. This task function is defining the servoing error through the difference between the current state vector s and s^* in the desired one.
In the classical IBVS, the state vector is defined in the image plane or planes (stereo vision), while in the SVS, we have defined our own state vectors in a "stereo space" thanks to the homography at infinity, $\mathtt{H}_\infty$, the epipoles and the measurements issued from the two images.

The classical IBVS with Plücker modeling is called and simulated in this thesis,

first with the interaction matrix defined in Chaumette's thesis [Cha90], then by providing a new interaction matrix with observers.
The simulation results are not convincing, as there were several local minima. Besides the theoretical analysis of the stability and the convergence were very difficult with these interaction matrices.
These reasons show a real need of new interaction matrices, with low dimensions, easy to develop for obtaining its pseudo inverse.
The decoupling between the rotational and translational velocities control laws makes the dimensions of the interaction matrix divided in two parts, and we always got an interaction matrix with dimensions of $n \times 3$ rather than $n \times 6$ without decoupling. With this dimension reduction, the analytical analyses of the interaction matrices were possible, and the theoretical prove of stability and convergence were done with the help of the Lyapunov theory. To do that, The classical IBVS is converted into a Stereo Visual Servoing (SVS). The two images (stereo images) are used together to build the "stereo space". New state vectors are defined in that space from a combination of visual measurements in the two images and the knowledge of the two-views affine and projective properties.
The Euclidean model of the camera represented with the intrinsic and extrinsic matrices $(\mathtt{K}, \mathtt{R}, t)$ is replaced by a model which involves the infinite homography $\mathtt{H}_\infty$ and the Fundamental matrix $\mathtt{F}$ (or the epipoles).
The first pair of state vectors $s_\Omega = (\mathtt{H}_\infty^{\mathsf{T}}\ l_l \times l_r), s_V = (l_r\ +\ \mathtt{H}_\infty^{\mathsf{T}}\ l_l)$ defined in the stereo space, are used to design partially decoupled control laws for rotational and translational velocities:

$$\begin{aligned} \mathbf{V} &= \lambda_V\ \mathtt{H}_V^+\ e_V \quad , \\ \mathbf{\Omega} &= \lambda_\Omega\ m\ \mathtt{H}_\Omega^+\ e_\Omega \end{aligned}$$

with

$$\mathtt{H}_V = \begin{bmatrix} \mathtt{M}_r^{-\mathsf{T}}[v_1]_\times \\ \vdots \\ \mathtt{M}_r^{-\mathsf{T}}[v_n]_\times \end{bmatrix} \quad , \quad \mathtt{H}_\Omega = \begin{bmatrix} \mathtt{M}_r[v_1]_\times \\ \vdots \\ \mathtt{M}_r[v_n]_\times \end{bmatrix}$$

and

$$e_V = \begin{pmatrix} s_{V1} - s_{V1}^* \\ \vdots \\ s_{Vn} - s_{Vn}^* \end{pmatrix} \quad , \quad e_\Omega = \begin{pmatrix} s_{\Omega 1} - s_{\Omega 1}^* \\ \vdots \\ s_{\Omega n} - s_{\Omega n}^* \end{pmatrix}$$

The global stability of the system in closed loop has been proved, but not the asymptotic convergence, the analytical form of the equilibrium points was found.

The simulations show the convergence for the state vectors for large rotation and translation motions.
To look for further decoupling, a new translational state vector $s = \frac{1}{2}||\mathtt{K}_r^{\mathsf{T}} s_V||^2$ has been designed. This state vector is independent from the rotation motion, but needs the knowledge of the camera calibration parameters. Then the translational task function $e = s - s^*$ defines the translational servoing error. With this new translational state vector we designed a new translational velocity control law

$$\mathbf{V} \;=\; -\lambda\, \mathtt{H}^{+}\, e \qquad \text{with} \qquad \mathtt{H} = \begin{bmatrix} u_1^{\mathsf{T}} \\ \vdots \\ u_n^{\mathsf{T}} \end{bmatrix} .$$

With these two translational velocity control laws and the rotational velocity control law, we have performed several simulations with large rotation and large translation motions. The results have been largely illustrated and discussed in every section.

5.2 Contributions

The main contributions of this PhD thesis are:

- The design of decoupled rotational and translational velocities control laws, for the stereo image based visual servoing. That is done in the "stereo space" using the homography at infinity $\mathtt{H}_\infty$ and the epipoles (or the fundamental matrix).
- The definition of new state vectors for the rotational and translational motion, in both position based and image based stereo servoing. These new state vectors are used in the task functions to design partial and full decoupled velocities control laws.
- A first step to the visual servoing of an articulated object. All the studied joints of the articulated object studied here are revolute.
- A true stereo position based visual servoing (4 DOF) with only one-line object.
- The computation of the transformation matrix between two poses using the Plücker coordinates of a straight line.
- The extension of the Umeyama's lemma to be used with Plücker coordinates for lines.
- The computation of the coordinates of the two closest points of two 3D straight lines represented by the normalized Plücker coordinates, and the minimal distance between these lines.

- The implementation of a fast real-time position based stereo visual servoing, applied to the 6 DOF Adept Viper S650 robot manipulator.

Here, maybe the end of this thesis, however many things remain to be done, and immediate perspectives of the presented scientific and technical methodology are manifold.

5.3 Perspectives

At the end of this work, we would like to propose some future perspectives of this work:

- 3D Visual tracking may be tackled with IBVS. It has a wide application in the industry, media, medical, and other environments. This gives a wide range of applications to the new control laws, in particular for those in the SVS category.

- The stereo visual servoing (SVS) defined in this thesis is a new approach in visual servoing. It can be developed and used in the servoing tasks that we did in position based visual servoing, like the visual servoing on an articulated object, and the visual servoing with one 3D straight line.

- In this thesis, we were interested in a simple application with articulated object with revolute joints only, but other types of joints with manipulator robots and mobile robots could be investigated. The approach proposed here could be extended to carry out the visual servoing an articulated object with prismatic joints or ball joints.

- The servoing with one straight line gives us a way to identify the type of the joint for an articulated or deformable object. This can be done without the help of image processing or object recognition, but rather by performing several virtual stereo visual servoings.

- Study of the trajectory of the visual indexes in SVS. It is known that the visual indices are always visible in the VSBI but not with SVS nor PBVS.

References

[AC09] G. Allibert and E. Courtial. What can prediction bring to image-based visual servoing ? In *proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5210–5215, St. Louis, USA, Oct 2009.

[ACLS98] J. Aggarwal, Q. Cai, W. Liao, and B. Sabata. Articulated and elastic motion. *Computer Vision and Image Understanding*, 70:142–156, 1998.

[AEH00] N. Andreff, B. Espiau, and R. Horaud. Visual servoing from lines. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2070–2075, San Francisco, USA, 2000.

[AEH02] N. Andreff, B. Espiau, and R. Horaud. Visual servoing from lines. *International Journal of Robotics Research*, 21(8):679–700, Aug 2002.

[Agi77] G. J. Agin. Servoing with visual feedback. In *proceedings of 7th Symposium on Industrial Robots*, pages 551–560, Tokyo, Japan, Oct 1977.

[Agi79] G. J. Agin. Real time control of a robot with a mobile camera. In *proceedingsTechnical Note 179, SRI International*, Feb 1979.

[AMAM06] H.H. Abdelkader, Y. Mezouar, N. Andreff, and P. Martinet. Decoupled homography-based visual servoing with omnidirectional cameras. In *proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, Oct 2006.

[And89] R. Andersson. Dynamic sensing in ping-pong playing robot. *IEEE Transactions on Robotics and Automation*, 5(6):728–739, 1989.

[And99] N. Andreff. *Asservissement visuel à partir de droites et auto-étalonnage pince-caméra*. PhD thesis, INP Grenoble et INRIA, France, 1999.

[BDVK10] J.C. Bazin, C. Demonceaux, P. Vasseur, and I.S. Kweon. Motion estimation by decoupling rotation and translation in catadioptric vision. *Computer Vision and Image Understanding*, 114(2), Feb 2010.

[BH05] D. Burschka and G.D. Hager. Vision-based 3d scene analysis for driver assistance. In *proceedings of IEEE Int. Conf. on Robotics and Automation*, pages 812–816, Barcelona, Spain,, Apr 2005.

[BS01] A. Bartoli and P. Sturm. The 3d line motion matrix and alignment of line reconstructions. In *proceedings of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 287–292, Hawaii, USA, Sep 2001.

[BZR95] P.A. Beardsley, A. Zisserman, and I. Reid. Metric calibration of a stereo rig. In *proceedings of IEEE workshop on representation of visual scenes*, pages 93–100, Cambridge, Mass, Jun 1995.

[Can86] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[CBM02] E. Cervera, F. Berry, and P. Martinet. Is 3d useful in stereo visual control? In *proceedings of IEEE International Conference on Robotics and Automation*, volume 2, pages 1630–1635, Washington D.C., USA, May 2002.

[CH06] F. Chaumette and S. Hutchinson. Visual servo control part i: Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4):82–90, Dec 2006.

[CH07] F. Chaumette and S. Hutchinson. Visual servo control part ii: Advanced approaches. *IEEE Robotics and Automation Magazine*, pages 109–118, Mar 2007.

[CH09] A. Chilian and H. Hirschmuller. Stereo camera based navigation of mobile robots on rough terrain. In *proceedings of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 4571–4576, St. Louis, MO, USA, Oct 2009.

[Cha90] F. Chaumette. *La relation vision-commande: théorie et application à des tâches robotiques.* PhD thesis, Université de Rennes 1, IRISA, Juillet 1990.

[CLIM05] C. Charron, O. Labbani-Igbida, and El M. Mouaddib. Qualitative localization using omnidirectional images and invariant features. In *proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3009 –3014, Edmonton, Canada, Aug 2005.

[CLK09] M. Chandraker, J. Lim, and D. Kriegman. Moving in stereo: Efficient structure and motion using lines. In *proceedings of IEEE 12th International Conference on Computer Vision*, pages 1741 – 1748, Oct 2009.

[CM00] F. Chaumette and E. Malis. 2 1/2 d visual servoing: a possible solution to improve image-based and position-based visual servoings. In *proceedings of IEEE International Conference on Robotics and Automation*, pages 630–635, San Francisco, CA, USA, Apr 2000.

[CMC05] A. Comport, E. Marchand, and F. Chaumette. Complex articulated object tracking. *Electronic Letters on Computer Vision and Image Analysis*, 5(3):21–31, 2005.

[CMC07] A. Comport, E. Marchand, and F. Chaumette. Kinematic sets for real-time robust articulated object tracking. *Image and Vision Computing*, 25:374–391, 2007.

[DA99] C. Doignon and G. Abba. A practical multi-plane method for a low-cost camera calibration technique. In *proceedings of the European Control Conference*, Karlsruhe, Allemagne, Sep 1999.

[DAAAM08] R. Dahmouche, O. Ait-Aider, N. Andreff, and Y. Mezouar. High-speed pose and velocity measurement from vision. In *proceedings of IEEE International Conference on Robotics and Automation*, pages 107–112, CA, USA, May 2008.

[Dan96] K. Daniilidis. The dual quaternion approach to hand-eye calibration. In *proceedings of 13th International Conference on Pattern Recognition*, volume 1, pages 318–322, Vienna, Austria, Aug 1996.

[Dan99] K. Daniilidis. Hand-eye calibration using dual quaternions. *Int'l journal of Robots Research*, 18(3):286–298, Jun 1999.

[DAO94a] C. Doignon, G. Abba, and E. Ostertag. Recognition and localization of solid objects by a monocular vision system for robotic tasks. In *proceedings of IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, pages 2007–2014, Munich, Germany, Sep 1994.

[DAO94b] C. Doignon, G. Abba, and E. Ostertag. Recognition and localization of solid objects by a monocular vision system for robotic tasks. In *Proceedings of IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, pages 2007–2014, Munich, Germany, Sep 1994.

[DC00] T. Drummond and R. Cipolla. Real-time tracking of multiple articulated structures in multiple views. In *proceedings of Int'l 6th European Conference on Computer Vision*, Jun 2000.

[DdM07] C. Doignon and M. de Mathelin. A degenerate conic-based method for a direct fitting and 3-d pose of cylinders with a single perspective view. In *proceedings of the IEEE Int. Conf. on Robotics and Automation*, pages 4220–4225, Rome, Italie, Apr 2007.

[DM07] F. Dionnet and E. Marchand. Robust stereo tracking for space applications. In *proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3373 – 3378, San Diego, CA, USA, Oct 2007.

[Doi07] C. Doignon. *An Introduction to Model-Based Pose Estimation and 3D Tracking Techniques.* International Journal of Advanced Robotic Systems, 2007.

[DRLR89] M. Dhome, M. Richetin, J. Lapreste, and G. Rives. Determination of the attitude of 3-d objects from a single perspective view. *IEEE transactions on pattern analysis and machine intelligence*, 11(12):1265–1278, 1989.

[ECR92] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326, June 1992.

[Fus01] A. Fusiello. A new autocalibration algorithm: Experimental evaluation. *InW. Skarbek, editor, Computer Analysis of Images and Patterns*, 2124:717–724, 2001.

[FW67] P. L. Falb and W. A. Wolovich. Decoupling in the design and synthesis of multivariable control systems. *IEEE Transactions on automatic control*, 12(6):651–659, 1967.

[GMM09] C. Guo, S. Mita, and D. McAllester. Stereovision-based road boundary detection for intelligent vehicles in challenging scenarios. In *proceedings of IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, pages 1723–1728, St. Louis, MO, USA, Oct 2009.

[HA96] A. Heyden and K. Aström. Euclidean reconstruction from constant intrinsic parameters. In *proceedings of 13th International Conference on Pattern Recognition*, volume I, pages 6–343, Vienna, Austria, Aug 1996.

[Hag97] G. D. Hager. A modular system for robust positioning using feedback from stereo vision. *IEEE Trans. on Robotics and Automation*, 13(4):582–595, 1997.

[Har97a] R. Hartley. In defence of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, Feb 1997.

[Har97b] R. Hartley. Kruppa's equations derived from the fundamental matrix. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):133–135, Feb 1997.

[HC98] R. Horaud and G. Csurka. Self-calibration and euclidean reconstruction using motions of a stereo rig. In *proceedings of Sixth International Conference on Computer Vision*, pages 96 –103, Bombay, Jan 1998.

[HCM95] G.D. Hager, W.C. Chang, and A.S. Morse. Robot hand-eye coordination based on stereo vision. *IEEE Control System Magazine*, 15(1):30–39, Feb 1995.

[HDE98] R. Horaud, F. Dornaika, and B. Espiau. Visually guided object grasping. *IEEE transactions on robotics and automation*, 14(4):525–532, Aug 1998.

[HEK96] K. Hashimoto, T. Ebine, and H. Kimura. Visual servoing with hand-eye manipulator - optimal control approach. *IEEE Trans. on Robotics and Automation*, 12(5):766–774, 1996.

[HHC96] S. Hutchinson, G. D. Hager, and P.I. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, Oct 1996.

[HP79] J. Hill and W. Park. Real time control of a robot with a mobile camera. In *proceedings of 9th International Symposium on Industrial Robots*, pages 233–246, 1979.

[HT98] G. Hager and K. Toyama. X vision : A portable substrate for real-time vision applications. *Computer Vision and Image Understanding*, 69(1):23–37, 1998.

[HZ00] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press,Cambridge, 2000.

[Kav57] R. J. Kavangh. Noninteracting control in linear multivariable systems. *Transaction of the American institute of electronic engineers, Part II, application and industry*, 76(5):95–100, 1957.

[KGdM+03] A. Krupa, J. Gangloff, M. de Mathelin, C. Doignon, G. Morel, L. Soler, J. Leroy, and J. Marescaux. Autonomous retrieval and 3d positioning of surgical instruments in robotized laparoscopic surgery. *IEEE Trans. on Robotics and Automation*, 19(5):842–853, October 2003.

[KR00] J. Knight and I. Reid. Self-calibration of a stereo rig in a planar scene by data combination. In *proceedings of International Conference on pattern recognition*, pages 1411–1414, Spain, Sep 2000.

[LD00] M. I. Lourakis and R. Deriche. Camera self-calibration using the singular value decomposition of the fundamental matrix. In *proceedings of Asian Conference on Computer Vision*, volume 1, pages 403 – 408, Jan 2000.

[LDV10] S. Ly, C. Demonceaux, and P. Vasseur. Translation estimation for single viewpoint cameras using lines. In *proceedings of IEEE International Conference on Robotics and Automation*, pages 1928–1933, Alaska, USA, May 2010.

[LH88] Y. Liu and T.S. Huang. A linear algorithm for motion estimation using straight line correspondences. *Computer Vision, Graphics, and Image Processing*, 44(1):35–57, Oct 1988.

[LHF90] Y. Liu, T.S. Huang, and O.D. Faugeras. Determination of camera location from 2d to 3d line and point correspondences. *proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):28–37, 1990.

[LM05] X. Lu and R. Manduchi. Detection and localization of curbs and stairways using stereo vision. In *proceedings of IEEE International Conference on Robotics and Automation*, pages 4648 – 4654, Barcelona, Spain, Apr 2005.

[Low87] D.G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, Mar 1987.

[LXXX03] H. Li, C. Xu, Q. Xiao, and X. Xu. Visual navigation of an autonomous robot using white line recognition. In *proceedings of IEEE Int'l Conference on Robotics and Automation*, volume 3, pages 3923 –3928, Taipei, Taiwan, Sep 2003.

[LZG10] S. Ly, D. Zhang, and J. Gu. Research of automatic needle locating based on stereo visual servoing. In *proceedings of 3rd International Conference on Biomedical Engineering and Informatics (BMEI 2010)*, volume 4, pages 1779–1783, Yantai, China, Oct 2010.

[Mal98] Ezio Malis. *Contribution à la modèlisation et à la commande en asservissement visuel*. PhD thesis, Université de Rennes I, 1998.

[MC00] E. Malis and F. 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.

[MC01] P. Martinet and E. Cervera. Stacking jacobians properly in stereo visual servoing. In *proceedings of IEEE International Conference on Robotics and Automation*, pages 717 –722, Seoul, Korea, May 2001.

[MC08] M. Marey and F. Chaumette. Analysis of classical and new visual servoing control laws. In *proceedings of IEEE International Conference on Robotics and Automation*, pages 3244–3249, Pasadena, CA, USA, May 2008.

[MCB99] E. Malis, F. Chaumette, and S. Boudet. 2d 1/2 d visual servoing. *IEEE Trans. on Robotics and Automation*, 20(1):238–250, 1999.

[MCB00] E. Malis, F. Chaumette, and S. Boudet. Multi-cameras visual servoing. In *proceedings of IEEE International Conference on Robotics and Automation*, volume 4, pages 3183 –3188, San Francisco, CA, USA, Apr 2000.

[MDGD97] P. Martinet, N. Daucher, J. Gallice, and M. Dhome. Robot control using monocular pose estimation. In *proceedings of Workshop New Trends Image-Based Robot Servoing*, Grenoble, France, 1997.

[MHGdM05] K. Miura, K. Hashimoto, J. Gangloff, and M. de Mathelin. Visual servoing without jacobian using modified simplex optimization. In *proceedings of IEEE/RSJ International Conference on Intelligent Robots and Automation*, pages 3504–3509, Barcelona, Spain, Apr 2005.

[Mit94] Amar Mitiche. *Computational Analysis of Visual Motion*. Plenum Press, New York, 1994.

[MLIM11] P. Merveilleux, O. Labbani-Igbida, and E.-M. Mouaddib. Real-time free space detection and navigation using omnidirectional vision and parametric and geometric active contours. In *proceedings of IEEE International Conference on Robotics and Automation*, pages 6312–6317, Shanghai, China, May 2011.

[Mou05] E. Mouaddib. Introduction à la vision panoramique catadioptrique. *Traitement du Signal*, 22(5), Sep 2005.

[MSEY05] E. Mouaddib, R. Sagawa, T. Echigo, and Y. Yagi. Stereovision with a single camera and multiple mirrors. In *proceedings of IEEE International Conference on Robotics and Automation*, pages 800–805, Barcelona, Spain, Apr 2005.

[NF97] N. Navab and O.D. Faugeras. The critical set of lines for camera displacement estimation: A mixed euclidean-projective and constructive approach. *Int'l Journal of Computer Vision*, 23(1):17–44, May 1997.

[NFV93] N. Navab, O.D. Faugeras, and T. Vieville. The critical set of lines for camera displacement estimation: A mixed euclidean-projective and constructive approach. In *proceedings of Fourth International Conference on Computer Vision*, pages 713–723, Berlin, Germany, May 1993.

[PG99] M. Pollefeys and L. V. Gool. Stratified self-calibration with the modulus constraint. *proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):707–724, Aug 1999.

[PL08] H. Pan and Y. Liu. Motion estimation of elastic articulated objects from points and contours with volume invariable constraint. *Pattern Recognition*, 41:458–467, 2008.

[Plu65] J. Plucker. On a new geometry of space. *Philosophical Transactions of the Royal Society of London*, pages 155:725–791, 1865.

[PPT08] L.M. Paz, P. Piniés, and J.D. Tardós. Large-scale 6-dof slam with stereo-in-hand. *IEEE transactions on robotics*, 24(5):946–957, Oct 2008.

[PS09] L. Pari and J.M. Sebastiàn. Image base visual servoing: Estimation of the image jacobian by using lines in a stereo vision system. In

proceedings of 3d IEEE International Conference on E-Learning in Industrial Electronics, pages 109 –114, Nov 2009.

[PSSA08] L. Pari, J.M. Sebastian, A. Sebastian, and L. Angel. Image based visual servoing: Estimation of the image jacobian by using fundamental matrix vs analytic jacobian. In *proceedings of International Conference on Image Analysis and Recognition*, pages 706 – 717, Jun 2008.

[QSL+06] L. Qiu, Q. Song, J. Lei, Y. Yu, and Y. Ge. Multi-camera based robot visual servoing system. In *proceedings of IEEE International Conference on Robotic and Automation*, pages 1509 –1514, Luoyang, China, Jun 2006.

[RE87] P. Rives and B. Espiau. Recursive estimation of 3d primitives by means of a mobile camera. *INRIA Research report*, 351, 1987.

[RH99] A. Ruf and R. Horaud. Rigid and articulated motion seen with an uncalibrated stereo rig. In *Seventh International Conference on Computer Vision*, volume 2, pages 789–796, Kerkyra, Greece, Sep 1999.

[RMLH99] A. Ruf, F. Martin, B. Lamiroy, and R. Horaud. Visual control using projective kinematics. In *proceedings of Ninth International Symposium of Robotics Research (ISRR 1999)*, pages 66–73, Snowbird, Utah, USA, Oct 1999.

[RRSM05] A. H. Rivera-Rios, F. Shih, and M. Marefat. Stereo camera pose determination with error reduction and tolerance satisfaction for dimensional measurements. In *proceedings of IEEE International Conference on Robotics and Automation*, pages 423–428, Barcelona, Spain, Apr 2005.

[SA90] M. Spetsakis and J. Aloimonos. Structure from motion using line correspondences. *International Journal of Computer Vision*, 4:171–183, 1990.

[SBE91] C. Samson, M. Le Borgne, and B. Espiau. *Robot control: The task function approach.* Oxford university press, 1991.

[SH96] Y. Seo and K. S. Hong. Sequential reconstruction of lines in projective space. In *proceedings of 13th International Conference on Pattern Recognition*, volume 1, pages 503–507, Vienna, Austria, Aug 1996.

[SI73] Y. Shirai and H. Inoue. Guiding a robot by visual feedback in assembling tasks. *Pattern Recognition*, 5(2):99–106, 1973.

[SMDVC08] J. Solà, A. Monin, M. Devy, and T. Vidal-Calleja. Fusing monocular information in multicamera slam. *IEEE transactions on robotics*, 24(5):958–968, Oct 2008.

[SVCD09] J. Solá, T. Vidal-Calleja, and M. Devy. Undelayed initialization of line segments in monocular slam. In *proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1553–1558, St. Louis, USA, Oct 2009.

[SW80] A.C. Sanderson and L.E. Weiss. Image based visual servo control using relational graph error signals. In *proceedings of IEEE International Conference on Robotics and Automation*, pages 1074–1077, 1980.

[SZ97] C. Schmid and A. Zisserman. The critical sets of lines for camera displacement estimation: a mixed euclidean-projective and constructive approach. In *proceedings of Conference on Computer Vision and Pattern Recognition*, pages 666–671, Puerto Rico, USA, 1997.

[TK95] C.J. Taylor and D.J. Kriegman. Structure and motion from line segments in multiple images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1021–1032, 1995.

[TMCG02] B. Thuilot, P. Martinet, L. Cordesses, and J. Gallice. Position based visual servoing: Keeping the object in the field of vision. In *proceedings of IEEE Int'l Conf. on Robotic and automation*, pages 1624–1629, Washington, DC, USA, May 2002.

[TN00] M. Tonko and H.-H. Nagel. Model-based stereo-tracking of non-polyhedral objects for automatic disassembly experiments. *International Journal on Computer Vision*, 37(1):99–118, 2000.

[Tsa87] R.Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, 1987.

[Ume91] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(4):376–380, Apr 1991.

[WHB96] W.J. Wilson, C.W. Hulls, and G.S. Bell. Relative end-effector control using cartesian position-based visual servoing. In *proceedings of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 12, pages 684–696, Oct 1996.

[WLW08] H. Wang, Y.H. Liu, and Z. Wang. Uncalibrated dynamic visual servoing using line features. In *proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3046–3051, Nice, France, Sep 2008.

[ZBR95] A. Zisserman, P. Beardsley, and I. Reid. Metric calibration of a stereo rig. In *proceedings of IEEE Workshop on Representation of Visual Scenes*, pages 93–100, Cambridge, Massachusetts, Jun 1995.

[ZF90] Z. Zhang and O. D. Faugeras. Tracking and grouping 3d line segments. In *proceedings of 3rd International Conference on Computer Vision*, pages 577–580, Osaka, Japan, Dec 1990.

[Zha94] Z. Zhang. Estimating motion and structure from correspondences of line segments between two perspective images. *proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1129–1139, Jun 1994.

[Zha99] Z. Zhang. A flexible new technique for camera calibration. In *Proceedings of the Int. Conf. on Computer Vision*, Corfou, Grèce, Sep 1999.

[ZO02] H. Zhang and J.P. Ostrowski. Visual motion planning for mobile robots. *proceedings of IEEE transactions on robotics and automation*, 18(2):199–208, Apr 2002.