Contributions to control modeling in visual servoing, task redundancy, and joint limits avoidance.

Mohammed Marey

N° d'ordre : 4249 **ANNÉE 2010**

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Européenne de Bretagne

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Informatique

Ecole doctorale Matisse

présentée par

Mohammed Abdel-Rahman MAREY

préparée à l'INRIA Rennes – Bretagne Atlantique & IRISA.
Equipe d'accueil : LAGADIC

Contributions to control modeling in visual servoing, task redundancy, and joint limits avoidance.

Thèse soutenue à Rennes le 16 Décembre 2010

devant le jury composé de :

Seth HUTCHINSON
Professeur, University of Illinois at Urbana-Champaign, USA / *rapporteur*

Philippe MARTINET
Professeur, Institut Français de Mécanique Avancée - Clermont-Ferrand, France / *rapporteur*

Wisama KHALIL
Professeur, Ecole Centrale de Nantes, France / *examinateur*

Patrick RIVES
Directeur de Recherche, INRIA Sophia Antipolis - Méditerranée, France / *examinateur*

Eric MARCHAND
Professeur, Université de Rennes 1, France / *examinateur*

François CHAUMETTE
Directeur de Recherche, INRIA Rennes - Bretagne Atlantique, France/ *directeur de thèse*

To my brother Wael ...

Acknowledgment

The research described in this thesis was carried out from October 2006 to December 2010 at the Institut National de Recherche en Informatique et Automatique INRIA, centre Rennes - Bretagne Atlantique and at the Institut de Recherche en Informatique et Systèmes Aléatoires IRISA in Rennes. The thesis was undertaken in the Lagadic group directed by François Chaumette. The four years of my thesis work were financially supported by the Egyptian Government through a scientific mission from the Ministry of High Education and Scientific Research, in addition to a supplementary support from the Lagadic group during the last year.

First of all, I am deeply indebted to my supervisor ***François Chaumette***, INRIA Senior Researcher, for his encouragement, helpful guidance, great care, valuable advices, continuous and indispensable assistance, and his great help in the interpretation of the results of this work. I would like to thank him for his confidence in me and his availability through my thesis. The numerous discussions and his motivating investment in the research surrounding my thesis is more than greatly appreciated. It is really my pride to perform my Ph.D. research under his supervision and it is a fact that this thesis would not have been possible unless his supervision.

I would like to thank the members of the jury for having accepted to assess my thesis:

Seth HUTCHINSON, Professor at the University of Illinois at Urbana-Champaign, USA, for the fruitful discussions that we had, for his time spent in particularly rigorous reading, analyzing the thesis, and writing his report.

Philippe MARTINET, Professor at the Institut Français de Mécanique Avancée - Clermont-Ferrand, France, for his interest in my work and time spent in writing his report.

Wisama KHALIL, Professor at Ecole Centrale de Nantes, France, to have honored me by accepting to be a member of the jury and for his interest in my work.

Patrick RIVES, Senior Researcher at INRIA Sophia Antipolis-Méditerranée, France, for accepting to be a member of the jury and for his useful comments.

Eric MARCHAND, Professor at the Université de Rennes 1, France, for presiding the jury.

I would like to thank again Prof. ***Seth HUTCHINSON*** for his valuable and useful comments about this work through several discussions with him either when he visited our group in Rennes or at Beckman Institute at Champaign, during the week I spent in his lab which was a great opportunity for me to have several long and deep discussions with him about my work. It was a great chance to know the ongoing research in his lab in high level control system, artificial intelligence and control systems security by his wonderful team members James Davidson, Sal Candido and the others. My sincere thanks are due to Seth for his detailed review and excellent advice while performing this work.

I would like to thank Prof. ***Abderrahmane Kheddar***, director of the Joint Robotics Laboratory at the National Institute of Advanced Industrial Science and Technology (CNRS-AIST JRL), Tsukuba, Japan, for facilitating my visit to his lab. I would like also to express my deepest thanks to Olivier Stasse for our discussions and for his help and for his demonstration of the HRP2, I woud like also to thank all researchers, Ph.D. students and master students in the lab for having many useful discussions about their work in the field of humanoid robot.

I would like to thank Prof. ***Farrokh J. Sharifi***, director of the Robotics and Manufacturing Automation Laboratory in Ryerson University, Toronto, Ontario, Canada, who visited our lab for six months during which we had several useful discussions and works concerning classical Kalman filters and a new one for pose estimation in visual servoing.

I would also like to thank our perfect engineer ***Fabien Spindlier*** for his continuous and great support during experimental phases that allows me to get accurate results thanks to the accurate calibrated camera and/or robot system; Prof. ***Eric Marchand*** for valuable discussion about ViSP library and joint limits avoidance; Dr. ***Alexandre Krupa*** for useful discussions and his last medical robot demonstration; Prof. ***El Mustapha Mouaddib*** for his useful comments about the final PhD presentation.

I would like to thank my friends and colleagues for their friendship and support including those who have shared their office with me giving me the chance to discuss several issues with them: Fabien Servant who was the first person I met in the train station in Rennes, Mahindra the first person I went out with in Rennes, Albert Diosi for several useful discussions, Anthony Saunier for his help, support, remarkable kindness and deeply understanding, Nicolas Melchior for his continuous help and of course for his friendship as well as Andrea Cherubini, Guillaume Fortier and Laurent Coutard. And those who complete the shape of Lagadic life including Christophe Collewet, Nicolas Mansard, Claire Dune, Roméo Tatsambon-Fomena, Odile Bourquardez, Rafik Mebarki, Muriel Pressigout, Thi Thanh Hai Tran, Amaury Dame, Céline Teulière, David Folio, Xiang Wang, Olivier Kermorgant, Tao Li, Pierre Martin, Caroline Nadeau, Antoine Petit, Guillaume Caron, Deukhee Lee, Hideaki Uchiyama, Manikandan Bakthavatchalam, Filip Novotny, Romain Tallonneau, Jean Laneu-

rit, Ravo and Emilio.

I would like to thank Claire Dune for her efforts in correcting my French paper; Nicolas Mansard for his advices at the first time I tried to implement a new control law and his encouragements; Roméo Tatsambon-Fomena for his helping me at the first time I tried to use the robot; Andrea Cherubini for several useful discussions about mobile robot, Caroline Nadeau for her nice demonstration of medical robot. I would like also to thank Ryuta Ozawa, professor at the Ritsumeikan University, Japon, for his friendship and several useful discussions during the year he spent in our group.

I would like to thank our assistant Céline Ammoniaux-Gharsalli for her perfect and great responsibility about the administrative work that saves a lot of time for performing this research, as well as Odile Mazuret and Angélique Jarnoux. I would like also to thank the Library staff for making relevant literature easily accessible during my Ph.D. work.

I would like also to thank all my friends in Rennes, especially Jennifer, Sofian, Alberto, Ciara, Patricia, Salome, Alexandro, Duygu, Arnaud, Maher, Bilal, Ghehan, Gawad, Adel, Ahmed and Mohammed Ghraba, Gehad, Mohammed El Agan, Rafik Zouari, Dr. Hany, Samar and Mahmoud for their support and accompany during my staying in France. I would like to thank Emma Chaironi without her I will not choose to come to study my PhD in France.

I would like also to thank Prof. ***Camelia Sobhi***, the Egyptian cultural and scientific consul in the Egyptian embassy, for her supports, continuous care and encouragements as well as all members in the Egyptian cultural office in Paris.

I would like also to thank Prof. ***Mohammed F. Tolba*** and Prof ***Ashraf Saad*** my Masters supervisors in the faculty of Computer and Information Sciences (FCIS), Ain Shams University, Cairo for their continuous support and encouragements.

I believe that in the life we are results of each passed moment we had lived in and that our mentality is created and formed based on difficulties we have to deal with. The more the difficulties we had and succeeded to overcome the more the ability to deal with other difficulties that may appear in the future even if they are not belonging to the same sort of difficulties we had experienced before. We just gain the logic of thinking how to avoid difficulties

In my opinion, the only important point is to have the ability to free our mind from any constraints that may set any limits to our way of thinking about those problem issues considered.

In this view, I would like to think of every moment in my life, and every person who had a positive signature kept in my mentality and my life. In fact they are a lot, I will try to mention those who come too fast to my mind, and I hope that the persons I did not mention

there names will forgive me because I have a bad memory to remember all of you so please accept my excuse and forgive me

First, my primary school, I would like to thank Mr. **Mohammed Zakaria** who created the hope to be an engineer inside my mind and my heart. Also, I would like to thank Mr. Mohammed Shedid, Miss. Zenab, Awatef, Ragaa and Khayria for their care about me and encourage me as I was 6 to 12 years old which allowed me to be one of the best students in Al-Shahed Adnan Al-Madany primary school, not for any thing but as a result of their efforts and encouragements. As for my friends during this time I would like to thank Adel Telib, Osama Salah, Mohammed Barakat, Mohammed Yousif, Mohammed Ryad, Wael Hassan and Mohammed Mostafa.

Secondary, my intermediate schools, I would like to thank all teachers who had taught me a lot of things and encouraged me to keep working hard. First Mr. Abd Alhamed in Al-Gahra intermediate school in Kiewit. As for my friends during this time I would like to thank Ryad Edham and Refaat Yaakoob. Then, Fathala Basha Barakat intermediate school in Menyet Al-Mourshed, Kafr Al-Sheekh, Egypt. I would like to thank Mr. Khaled Marey and Mm. Seham Al-Somody for their encouragements. As for my friends during this time, I would like to thank Mohammed Afify, Ola Marey and all other friends.

In the secondary schools, I would like to thank many teachers especially Mr. Mohammed Abdal-Rahman in Ismaeil Al-Kabany Secondary School and Mr. Amr and Mostafa and all my teachers without any exception in Al-Ahram Secondary School. As for my friends during this time I would like to thank my best friends Husam Al-Talyawy and Magdy Al-Bagoury.

In the faculty of sciences, where I obtained my B. Sc. degree in Pure Mathematics and Computer Science, I would like to thank all professors and doctors and teaching assistants. Prof. Khayrat, Prof. Nashat, Prof. Hatem Al-Henawy, Prof. Al-zahar, Prof. Bayoumy, Prof. Souwd, Prof. Lobna, Prof. Entesarat, Prof. Al-Herbety and Prof. Ilham Al-Sheref. In addition to all teaching assistant in that time who became great doctors including Dr. Ahmed, Dr. Yaser. Dr. Hatem, Dr. Christen, Dr. Maged, Dr. Khaled, Dr. Ghada, Dr. Sayed and all other doctors. And my friends Mohammed Abd el Megeed, Rania Adel, Ahmed Gawesh, Yassen, Ossama Terra, Mohammed Amen, Hala Mosher and all other friends.

I would like also to thank all staff members in FCIS including Prof. ***Saied Abd Al wahab***, Prof. ***Essam Khalefa***, Prof. ***Hatem Al Ayady***, Prof. ***Saied Al ghonemy***, Prof. ***Ahmed Hamad***, Prof. ***Sayed Fadel***, Prof. ***Ali Al Naem***, Prof. ***Taha el Areef***, Prof. ***Mahmoud Al gamal*** and Prof. ***Mohammed Roshdy*** as well as all my friends and colleagues.

I would like also to thank those persons who have great impacts in my personality, first of all the person who I will never forget Eng. ***Magdy Ashour*** as well as Mr. Abdel Megeed and Eng. Aktham for all what I had learned from them.

I owe my loving thanks to my parents. They have lost a lot due to my research abroad. Without their encouragement and understanding it would have been impossible for me to finish this work. My special gratitude is due to my sister Walaa and my brothers Tamer, Alaa and before all to my brother Wael who passed away one year after I started this work. For him I present this work.

Mohammed Abd Al Rahman Marey

Contents

Chapter 1

Introduction

Many researches in robotics aim at the involvement of exteroceptive additional sensors which is strongly contributing to the ever growing demand to improve autonomy, robustness, flexibility and precision. Nevertheless good sensing abilities are essential to gain a higher flexibility and autonomy of robots. When robots operate in unstructured environments, the sensory information is included in the control loop. Many different sensors have been developed over the past years to fit the requirements of different but very specific tasks. Measurements gained by employing these sensors, that reflect variation in some phenomena, are used in the control loop to adapt the task execution to these variations. The most common sensors in robotics are force/torque sensors, laser range finders, ultrasonic sensors and camera sensors. Among all sensors, vision is the most complementary one with respect to the sensory information it provides to be included in the control loop since visual sensors are powerful means for a robot to perceive its environment. In particular, the use of visual feedback from sensor camera, when it is used in a correct manner, guarantees accurate positioning, robustness to calibration uncertainties, and reactivity to environmental changes.

Visual servoing is a viable method for robotic control based on the utilization of visual information extracted from images to close the robot control loop. Various areas are involved in visual servoing as shown in Fig. 1.1. Visual information obtained from the image processing can be used to extracting 2D features. It can also be used to estimating pose parameters by employing a pose estimation algorithm from computer vision. The estimated pose is transformed into the 3D features. These 2D and/or 3D features are then used in the control scheme. Optimization techniques, robot dynamic and/or robot kinematics are used in the modeling phase of the control schemes. Advances in the power and availability of image processing have made possible the computing required at frame rate. This makes that the use of real time video information for robotic guidance is increasingly becoming a more attractive proposition and is the subject of much research. It enables visual servoing with

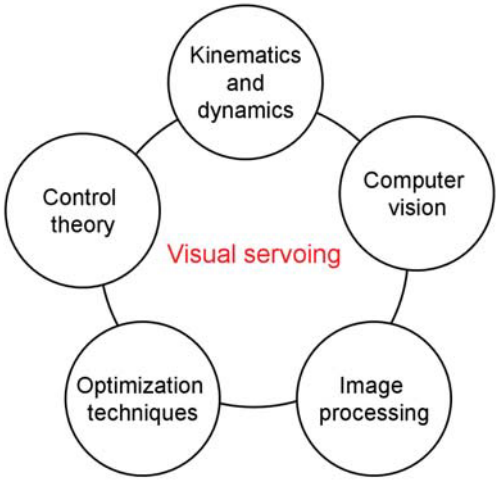


Figure 1.1 – Visual servoing related areas

sufficient accuracy and many useful tasks can be accomplished. This makes visual servoing, that is the closed loop control of motion on the basis of image analysis, more and more recognized as an important tool in modern robotics control. Two types of digital imaging sensors are usually used in computer vision and can be used for visual servoing. First type encodes light intensity to give intensity images acquired by cameras while the second type encodes shape and distance to give range images acquired by laser or sonar scanners.

With a vision sensor, which provides 2D measurements, the nature of the potential visual features is extremely rich, since it is possible to design visual servoing using both 2D features, such as the coordinates of characteristic points in the image, and 3D features, provided by localization module operating on the extracted 2D measurements. The wide range of possibilities is the reason behind the major difficulty in visual servoing, that is to build, select and design as best as possible the visual data needed and the control scheme used for obtaining a suitable behavior of the system, based on all the available measurements.

Using vision in robot control makes it possible to solve different problems, that can be handled safely based on the sensory visual data without any contact with the environment, for example the obstacle avoidance problem when the 3D models of the obstacles are known. However, some issues should be considered when vision sensors are used within robot system. These issues include local or global stability, robustness behavior, suitable trajectories for the robot and for the measurement in the image, a maximum decoupling between the visual information and the controlled degrees of freedom, avoiding singularity and local minima for the control (interaction) matrix, keeping features in the camera field of view, occlusion avoidance and avoiding robot joint limits.

1.1 Motivations

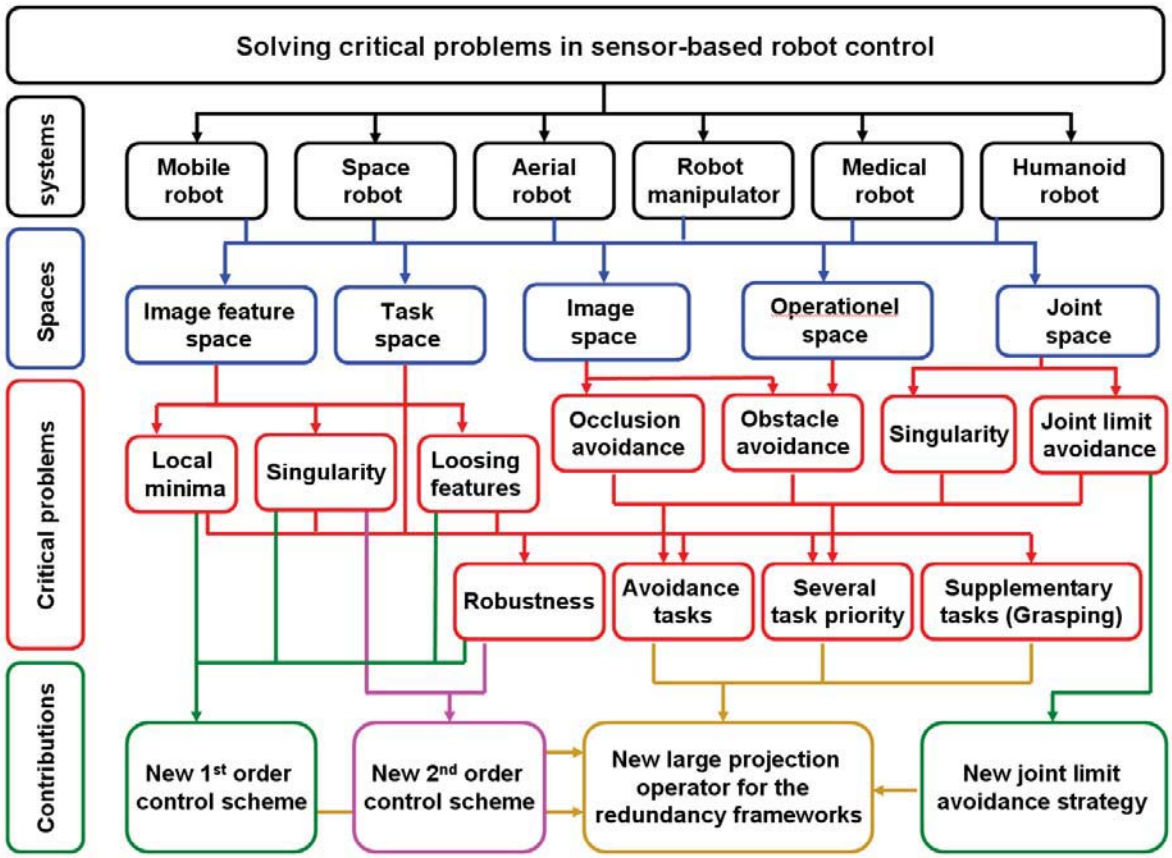


Figure 1.2 – Thesis scopes: problem domains and contributions

Visual servoing methods can be employed in almost all robotic systems that perform positioning and interactions tasks (for example space robot, mobile robot, robot arm, humanoid robot,..., (see Fig. 1.2). It involves working with different spaces in order to acquire the input signal in addition to transform a motion from one space to another in order to execute the required tasks. Visual servoing encounters different sorts of problems like local minima, singularities and loosing features from the camera field of view that may be avoided either by changing the feature set or by modeling suitable control schemes. These problems motivate the research in modeling new visual servoing control schemes. Another type of problem that requires defining avoidance tasks and selecting a mechanism to ensure its execution is classified as constrained problems, for example joint limit avoidance, obstacle avoidance, and occlusion avoidance. Avoiding this kind of problems is usually realized by the utilization of gradient projection methods. That is why it is important to have a gradient projection method that allows better performing of these avoidance tasks. These sorts of problems motivated us to accomplish the present work.

1.2 Contributions and thesis Outline

The thesis is written in two parts after the introductory chapter (see Fig. 1.3). The first part is about visual servoing control schemes and contains chapters 2, 3 and 4. It starts by reviewing the state of the art in visual servoing control schemes, then new control schemes are proposed and their experimental evaluations are presented. The second part is about redundancy and joint limits avoidance. It starts by presenting the state of the art in redundancy formalisms and joint limits avoidance strategies in robot control. A new large projection operator and a new joint limit avoidance strategy are then proposed. This part contains chapter 5, 6 and 7. Finally, the conclusions of these two parts are collected in chapter 8.

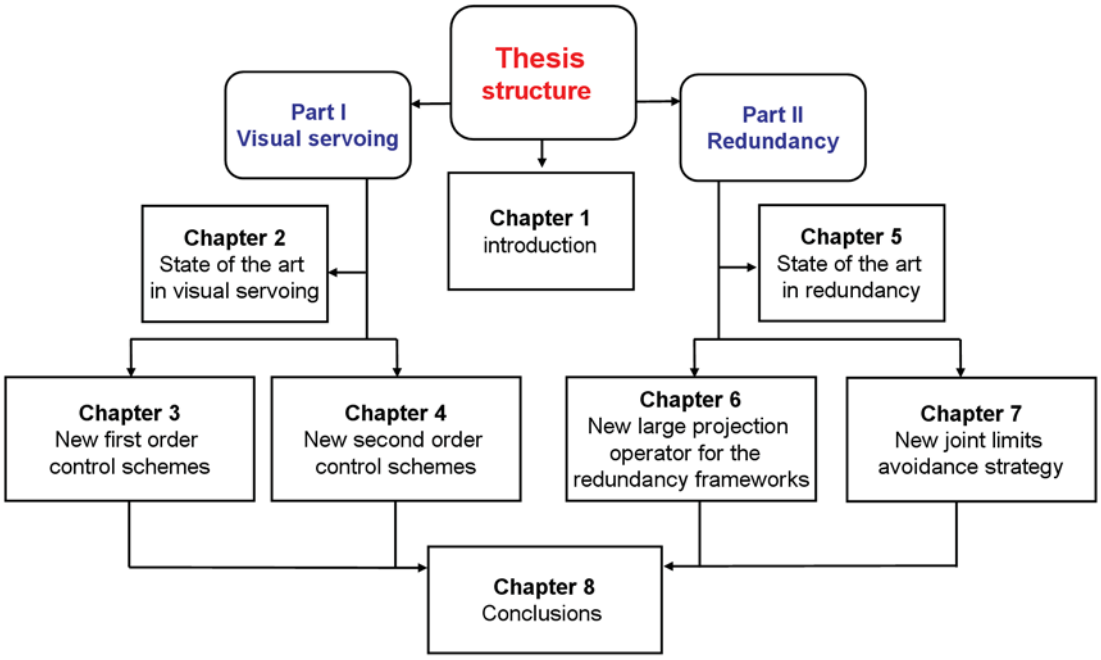


Figure 1.3 – Thesis structure

Part one: Visual servoing (chapter 2 to 4)

Chapter 2: State of the art in visual servoing: In this chapter, state of the art in visual servoing in the light of visual features used, control schemes designed and problems that could appear is presented.

Chapter 3: New first order control scheme: In this chapter we analyze different control schemes by considering the most usual and simple features, that are the Cartesian coordinates of image points. As for the control schemes, we consider three classical control laws and we propose in this chapter two new ones. The first new control law follows an

hybrid strategy. It is based on a behavior parameter that can be used to tune the weight of the current and the desired interaction matrix in the control law. We will see that in some configurations where all other control schemes fail due to local minima or singularity problems, this new control law allows the system to converge. The second control law that we propose is an attempt towards global asymptotic stability (GAS). Unfortunately, if GAS can be obtained in the space of the chosen task function, we will see that it is not ensured in the configuration space when redundant image point coordinates are used. This control scheme is indeed subject to attractive local minima. The analysis of the control laws included in this chapter is performed with respect to translational motion along and rotational motion around the optical axis. As we will see, a singularity of the control law proposed in [Malis 04] will be exhibited thanks to this analysis.

Chapter 4: New second order control schemes: In this chapter, we propose new control schemes to try to reach a singular configuration. It is based on Halley's method, which uses a second order minimization step. After analyzing the behavior of the classical control schemes in a singular configuration, we present the new control schemes. They are based on the Hessian matrices of the selected features. We thus determine the Hessian matrices of the Cartesian coordinates of an image point. Finally, experimental results are presented.

Part two: Redundancy (chapter 5 to 7)

Chapter 5: State of the art in task redundancy and joint limits constraints: In this chapter, we present the state of arts of the utilization of gradient projection methods (GPM) in robot control to cope with injecting constraints and additional tasks to the main task. The state of arts about avoidance frameworks and joint limits avoidance strategies are also presented.

Chapter 6: New large projection operator for the redundancy framework: In this chapter, we propose a new projection operator. Instead of considering all the components of the main task, only the norm of this task is used. Considering the norm of the errors allows the corresponding projection operator to enlarge the permitted motions, at least when the errors are still large. As we will see, using this new projection operator leads to a less constrained problem since the task based on the norm is of rank one at maximum. Our analytical studies show that this operator has to switch to the classical projection operator when the norm of the total error approaches zero. A switching strategy for the projection operator has thus been developed. Finally, an adaptive gain is also proposed to slow down the convergence of the main task. It allows the secondary tasks to be active for longer, which may be useful in practice when the secondary tasks have really to be taken into account (for obstacles or joint limits avoidance for instance). The recursive augmented projection operator for redundancy-based task-priority is presented and its efficiency is compared with the classical one.

Chapter 7: New joint limits avoidance strategy: In this chapter, we use the projection operator proposed in chapter 6 for injecting a joint limits avoidance task into the main task. We first present a new avoidance technique that uses the gradient projection method. This avoidance technique is based on three functions proposed: an activation function that activates the avoidance task and sets its direction; an adaptive gain function that controls the magnitude of the avoidance task; and a tuning function that ensures the smoothness of the injection of the avoidance task into the main task. Then, the problem of adding additional secondary tasks to the main task to be performed simultaneously while ensuring the joint limits avoidance is solved. These additional tasks could be used for moving the robot away from the neighborhood of the joint limits, avoiding occlusion and obstacles, performing emergency needed motion, or keeping object features in the camera field of view. Finally, the new avoidance strategy is integrated with the iterative approach to ensure the smoothness of the injecting of the avoidance task into the main task and a solution to the discontinuity problem of this approach is proposed by performing an inner switching loop inside the visual servoing loop.

Part I

Visual servoing

Chapter 2

State of the art of visual servoing

2.1 Visual servoing

Visual servoing is a well known flexible and robust technique to increase the accuracy and the versatility of a vision-based robotic system [Weiss 87, Feddema 89a, Hutchinson 96, Chaumette 06, Chaumette 07]. It consists in using vision in the feedback loop of a robotic system control with fast image processing to provide reactive behavior. Single or multiple cameras can be used to track visual information in order to control a robot with respect to a target [Shirai 73, Weiss 87, Hashimoto 93a, Hutchinson 96, Kragic 02]. Robotic task in visual servoing is specified in terms of image features extracted from a target object and their use for controlling the robot/camera motion through the scene [Espiau 92, Hashimoto 93a]. The positioning task is expressed as the regulation to zero of a task function. The task function in visual servoing is defined to control the pose of the robot's end-effector relative to either a world coordinate frame or a frame of a manipulated object. Visual features extracted in real time from the image are used to define a task function that depends on the robot configuration and the time [Samson 91].

Several methods have been developed for controlling the robot using visual information. Two basic approaches have been proposed namely position-based visual servoing (PBVS) and image-based visual servoing (IBVS) [Sanderson 80, Weiss 87, Hutchinson 96]. Both approaches have been used as bases for developing other schemes such as 2 1/2 D visual servoing [Malis 99]. In position-based visual servoing (PBVS or 3D VS), the image measures are processed in order to estimate the relative 3D pose between the camera and the target, which is then used as an error signal for controlling the motion of the robot/camera system toward its desired goal [Martinet 97] [Wilson 96]. 3D coordinates of points can also be used [Martinet 96]. In image-based visual servoing (IBVS or 2D VS), the error is directly computed in terms of features expressed in image [Weiss 87, Feddema 89b, Espiau 92]. Two

main aspects have a great impact on the behavior of any visual servoing scheme: the selection of the visual features used as input of the control law and the designed form of the control scheme. On one hand, the same feature set gives different behavior when employed in different control schemes and on the other hand, the same control law gives different behavior when it considers different feature sets. The behavior obtained is not always the required one: selecting a specific feature set or a specific control scheme may lead to some stability and convergence problems. In the following, a review in visual servoing is presented by focusing on modeling issues.

2.2 Camera-robot configurations

Two main configurations exist to combine camera(s) and robot for visual servoing applications. The first configuration called eye-in-hand, consists in camera(s) mounted on the end effector of the robot. A constant transformation between the camera frame and the end effector frame is defined and used for transforming the motions from the camera frame into the end effector frame, the robot frame, or any other reference frame attached to the robot. In the second configuration, which is called eye-to-hand, one or several cameras are placed in the workspace to monitor a target, end effector or both. This configuration needs to compute the transformation between cameras and robot frames at each iteration. The former configuration is commonly used in visual servoing since it allows keeping target(s) in the field of view (for example when a grasping task requires to ensure monitoring the object and a grasping tool attached to the end effector [Allen 93]). Hybrid configurations can be constructed when eye-to-hand and eye-in-hand configurations are used, [Flandin 00]. Within the work presented in this thesis, we use the eye-in-hand camera-robot configuration.

2.2.1 Eye-in-hand configuration

In the eye-in-hand configuration (see Fig. 2.1), the camera is attached to the end-effector of the robot and visual servoing control schemes are designed such that the velocity vectors $\mathbf{v}$ or $\dot{\mathbf{q}}$ are defined to control the required motion in the camera space or in the robot's joints space respectively. This velocity vector is sent to the robot controller in order to perform the required motion. Thus camera calibration, robot calibration and the end-effector to camera calibrations are first needed in order to have the set of camera intrinsic parameters, the robot Jacobian ${}^{e}\mathbf{J}_{q}$ and the transformation matrix ${}^{c}\mathbf{M}_{e}$ between the camera frame $\mathbf{F}_{c}$ and the end-effector frame $\mathbf{F}_{e}$ [Tsai 89]. Let us note that visual servoing is robust with respect to the calibration errors.

2.2.2 Eye-to-hand configuration

In the eye-to-hand configuration, the camera is fixed in the workspace (see Fig. 2.2). Unlike the eye-in-hand configuration where the image of the target changes based on both the

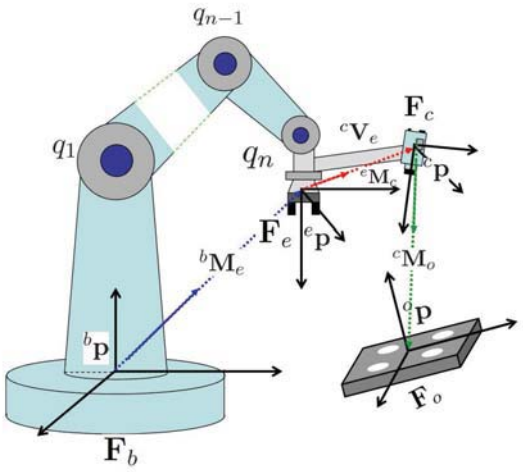


Figure 2.1 – Eye-in-hand camera-robot configuration

movement of the target and the movement of the end effector, in eye-to-hand configuration, the image of the target changes based on the movement of the target. The transformation matrix ${}^c\mathbf{M}_b$ between the camera frame $\mathbf{F}_c$ and the base coordinate system of the robot frame $\mathbf{F}_b$ is constant and is computed one time. The relative position between the robot end effector frame and the camera frame can be computed using transformations between different frames. More details concerning camera/robot configurations can be found in [Hutchinson 96] [Kragic 02] [Chaumette 07].

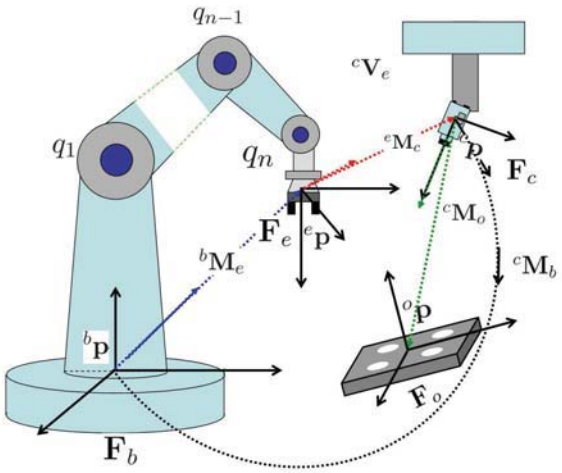


Figure 2.2 – Eye-to-hand camera-robot configuration

2.3 Task functions in visual servoing

Tasks in visual servoing are defined to describe how the robot can interact with its environment using vision sensors. All attained positions by the robot end effector's tools are belonging to a configuration space called task space $\mathbf{e}$. In three dimensional workspace when translation motions, rotation motions or translation motions combined with rotation motions are required to be performed respectively by tasks, the corresponding task space is defined respectively by $\mathbf{e} \in \mathbb{R}^3$ where $\mathbb{R}^3$ is the Cartesian space, $\mathbf{e} \in SO^3$ where SO^3 is the spatial orthogonal space or by $\mathbf{e} \in SE_3$ where $SE_3 = \mathbb{R}^3 \times SO^3$, the special Euclidean space denoted . In order to apply these motions with respect to a specified reference frame (for example camera frame, end-effector frame, tool frame, manipulated object frame, or robot frame), a transformation between different frames is required to transform the motion from one frame to another [Dombre 07][Spong 06]. In order to perform a task, the object must exhibit visual features which can be extracted from different points of view depending on the object appearance.

2.3.1 Camera space control

Typically, when $\mathbf{s}^*$ is the desired feature vector, $\mathbf{s}$ is the current feature vector, and ${}^c\mathbf{p}(t)$ is the relative pose between the camera and the object at instance t, the regulation scheme of task function $\mathbf{e}$ is defined as:

$$\mathbf{e} = \mathbf{s}({}^c\mathbf{p}(t)) - \mathbf{s}^* \tag{2.1}$$

When the control variables are defined in the camera frame, the variation of the visual feature $\mathbf{s}$ related to the relative movements between the camera and the scene is given by:

$$\dot{\mathbf{s}} = \frac{\partial \mathbf{s}}{\partial {}^c\mathbf{p}} \frac{d{}^c\mathbf{p}}{dt} + \frac{\partial \mathbf{s}}{\partial t} = \mathbf{L}_\mathbf{s} {}^c\mathbf{v} + \frac{\partial \mathbf{s}}{\partial t} \tag{2.2}$$

where $\frac{\partial \mathbf{s}}{\partial t}$ is the variation of $\mathbf{s}$ due to the object own motion, $\mathbf{L}_\mathbf{s} \in \mathbb{R}^{k \times n}$ is the visual features interaction matrix that represents the differential relationship between the features $\mathbf{s}$ and the camera frame [Hutchinson 96] [Dombre 07] , ${}^c\mathbf{v} = \mathbf{v}_c - \mathbf{v}_o$ is the relative instantaneous velocity between the camera frame $\mathbf{F}_c$ and the object frame $\mathbf{F}_o$ expressed in the camera frame $\mathbf{F}_c$, $\mathbf{v}_c$ is the instantaneous camera velocity and $\mathbf{v}_o$ is the instantaneous object velocity. When the object is motionless, $\frac{\partial \mathbf{s}}{\partial t} = 0$ and we get:

$$\dot{\mathbf{s}} = \mathbf{L}_\mathbf{s} \mathbf{v} \tag{2.3}$$

where $\mathbf{v} = {}^c\mathbf{v}$.

2.3.2 Joint space control

For an eye-in-hand system, the relation between $\dot{\mathbf{s}}$ and the velocity of the joint variables $\dot{\mathbf{q}}$ can be obtained as follows [Dombre 07]:

$$\dot{\mathbf{s}} = \frac{\partial \mathbf{s}}{\partial {}^c\mathbf{p}} \frac{\partial {}^c\mathbf{p}}{\partial {}^e\mathbf{p}} \frac{\partial {}^e\mathbf{p}}{\partial \mathbf{q}} \dot{\mathbf{q}} + \frac{\partial \mathbf{s}}{\partial t} = \mathbf{L_s} {}^c\mathbf{V}_e \, {}^e\mathbf{J_q} \dot{\mathbf{q}} + \frac{\partial \mathbf{s}}{\partial t} \tag{2.4}$$

where ${}^e\mathbf{J_q}$ is the Jacobian of the robot expressed in the end effector frame $\mathbf{F}_e$ and ${}^c\mathbf{V}_e$ is the twist transformation matrix between the end effector frame $\mathbf{F}_e$ and the camera frame $\mathbf{F}_c$ defined by [Dombre 07]:

$${}^c\mathbf{V}_e = \begin{bmatrix} {}^c\mathbf{R}_e & [{}^c\mathbf{t}_e]_\times {}^c\mathbf{R}_e \\ \mathbf{0}_3 & {}^c\mathbf{R}_e \end{bmatrix} \tag{2.5}$$

where ${}^c\mathbf{R}_e$ and ${}^c\mathbf{t}_e$ are the rotation matrix and the translation vector from the end effector frame $\mathbf{R}_e$ to the camera frame $\mathbf{R}_c$. The matrix ${}^\mathbf{c}\mathbf{V_e}$ remains constant for eye-in-hand configurations, while it has to be estimated at each iteration for eye-to-hand configurations.

If the object is not moving $\frac{\partial \mathbf{s}}{\partial t} = 0$, from (2.4)we get:

$$\dot{\mathbf{s}} = \mathbf{J_s} \dot{\mathbf{q}} \tag{2.6}$$

where $\mathbf{J_s} = \mathbf{L_s} {}^c\mathbf{V}_e \, {}^e\mathbf{J_q}$ is the Jacobian of the visual features.

2.4 Selection of visual features

Visual features observed by the vision sensor define the inputs to the control scheme. A feature can be any property that represents a part of the scene and can be extracted from the image. Vision sensor can be conventional camera (as usually used in visual servoing), 2-D ultrasound camera [Mebarki 10] or omni-directional cameras that is motivated in robotics applications to avoid visibility problems due to the restricted field of view of conventional camera [OkamotoJr 02] [HadjAbdelkader 07]. Selection of good visual features is a crucial aspect of visual servoing as it is necessary for achieving optimal speed, accuracy, and reliability of image measurements, consequently, affects performance and robustness of visual servoing [Feddema 91] [JanabiSharifi 97] [Chaumette 98]. Imaging measurements are either used directly in the control loop or used for relative pose estimation of a workpiece with respect to a camera. The number of degrees of freedom (DOF) to be controlled by the employed control scheme determines the minimum number of independent features required. Therefore, it is desirable to choose features which will be highly correlated with movements of the camera. In the following, we classify image features, that define the input signal to visual servoing control scheme, into three classes named geometric features, photometric features, and velocity field features.

2.4.1 Geometric features

Geometric features are defined to describe geometrically contents involved in a scene (2D visual features) or to relate a frame attached to a robot system with a frame attached to an object in a scene (3D visual features). Both 2D and 3D visual features can be used simultaneously in a hybrid form as we will see in Section 2.5.3.1.

2D visual features

Visual features can be selected in 2D image space as point coordinates, parameters representing straight lines or ellipses, region of interest, contours, [Feddema 91] [Espiau 92] [JanabiSharifi 97] [Corke 01] [Gans 03c]. These features are defined from image measurements. In case of image points, Cartesian coordinates are generally used but it may be also possible to use their polar and cylindrical coordinates [Iwatsuki 05]. In all cases, the parameters defining the internal camera calibration are required.

Image moments can also be used in visual servoing [Chaumette 04] [Tahri 05]. Using image moments, the improvements with respect to classical visual servoing seem to be significant, since it allows a generic representation not only able to handle simple geometrical primitives, but also complex objects with unknown shapes. It is shown in [Tahri 05] that moment invariants can be used to design a decoupled 2D visual servoing scheme and to minimize the nonlinearity of the interaction matrix related to the selected visual features.

3D visual features

Visual features can also be selected in 3D Cartesian space such as pose or coordinates of 3D points [Martinet 96] [Wilson 96] [Deng 03]. Usually, object model and image measurements are used to compute or to estimate the relative pose between object and camera frames in the Cartesian space or to reconstruct the 3D coordinates. In [Cervera 03], the 3D coordinates of the points of the object are used as the feature vector. A priori knowledge about the camera calibration parameters is required. In PBVS, orientation in pose vector can be represented by the total format, roll-pitch-yaw or axis-angle formats, [Wang 92] or quaternion formulate [Guoqiang 10].

Hybrid visual features

Several combinations between visual feature types can also be considered: for example, a mixture composed of both kinds of 2D and 3D features is presented in [Malis 99] [Deng 02a] [Cervera 03] [Marchand 05a], and finally polar and Cartesian parameterizations of image points coordinates as presented in [Corke 09].

Gaussian mixture features

In [Hafez 08a], an approach that does not require tracking nor matching has been presented. Collectively features points extracted from the image are modeled as a mixture of Gaussian.

The error function is defined as the distance between the current and desired Gaussian mixture models. As presented in [Hafez 08a], three degrees of freedom can be controlled using this approach while six degrees of freedom can be considered under the assumptions that a depth distribution is available.

Redundant features

Utilizing redundant features can improve the performance of visual servoing control and increase the positioning accuracy by improving the corresponding minimum singular value of the extended image Jacobian [Hashimoto 96]. However, processing a large feature set can sometimes be computationally infeasible, and therefore the focus must be on the selection of an information-intensive and reliable set that possesses a minimum number of features.

Automatic feature selection

A methodology for automatic visual feature selection among a set of features is presented in [Feddema 91] [Papanikolopoulos 93] [JanabiSharifi 97]. This approach selects the more reliable combination from a given feature set depending on some control criteria based on a weighted criteria function of robustness, completeness, uniqueness and computational time complexity of the image features.

Interaction matrix

The analytical form of the interaction matrix is based on the type of the used camera (conventional camera, 2D or 3D ultrasound cameras or omni-directional camera) and the projection model used [Hutchinson 96][TatsambonFomena 09]. The most common geometric model for usual cameras is the perspective projection model [Horn 86], (see Fig. 2.3). In this model, the center of projection is considered at the origin of the camera frame $\mathbf{F}_c$ and the image plane is at $Z = f$, where f is the camera focal length. By considering a 3D point with coordinates $\mathbf{X} = (X, Y, Z)$ in the camera frame and using a perspective projection model [Forsyth 03], the point $\mathbf{X}$ is projected on a 2D point $\mathbf{x}$ of coordinates (x, y) in the image plane such that:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} X/Z \\ Y/Z \end{bmatrix} = \begin{bmatrix} (u - c_u)/f\alpha \\ (v - c_v)/f \end{bmatrix} \tag{2.7}$$

where $\mathbf{x}_p = (u, v)$ is the image point coordinates in pixel unit, $\mathbf{c} = (c_u, c_v)$ is the coordinates of the principle point, f is the focal length of the camera lens and α is the ratio of pixel dimension.

Interaction matrix of image feature points (2D): As for the interaction matrix $\mathbf{L_s}$ of an image feature point it can be obtained as following: By taking the time derivative of (2.7) we get:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \dot{X}/Z - X\dot{Z}/Z^2 \\ \dot{Y}/Z - Y\dot{Z}/Z^2 \end{bmatrix} \tag{2.8}$$

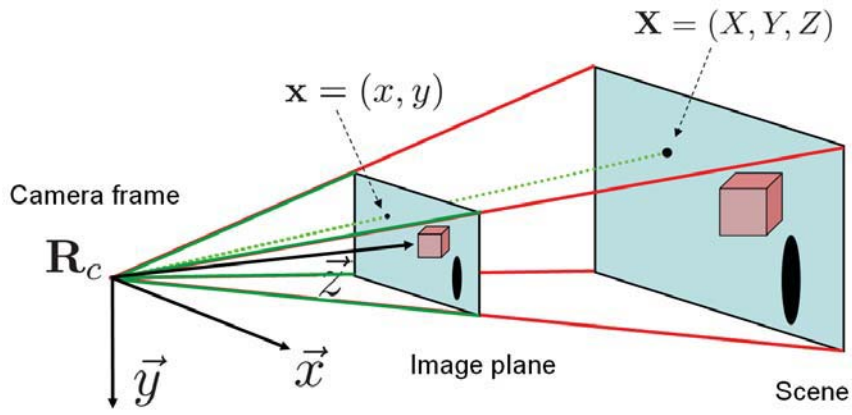


Figure 2.3 – Camera model

If the spatial velocity of the camera is given by $\mathbf{v} = (\boldsymbol{v}, \boldsymbol{\omega})$ where $\boldsymbol{v} = (v_x, v_y, v_z)$ and $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$ are the instantaneous linear and angular velocities of the origin of the camera frame both expressed in $\mathbf{F}_c$, then the velocity of the 3D point $\mathbf{X}$ related to the camera velocity is defined using the fundamental kinematic equation $\dot{\mathbf{X}} = -\boldsymbol{v} - \boldsymbol{\omega} \times \mathbf{X}$ such that:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} v_x - \omega_y Z + \omega_z Y \\ v_y - \omega_z X + \omega_x Z \\ v_z - \omega_x Y + \omega_y X \end{bmatrix} \tag{2.9}$$

By injecting the values of $\dot{X}$, $\dot{Y}$ and $\dot{Z}$ from (2.9) in (2.8) and grouping for $\boldsymbol{v}$, and ω we get the classical result [Weiss 87]:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \frac{-1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & \frac{-1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix} \tag{2.10}$$

which can be written as

$$\dot{\mathbf{x}} = \mathbf{L}_{\mathbf{x}} \ \mathbf{v} \tag{2.11}$$

where $\mathbf{L}_{\mathbf{x}}$ is the interaction matrix related to $\mathbf{x}$. If there is a set of k feature points $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_k)$, the interaction matrix $\mathbf{L}_{\mathbf{x}}$ of the set $\mathbf{x}$ is obtained by stacking $\mathbf{L}_{\mathbf{x}_i}$ for all $x_i \in \mathbf{x}$ to get:

$$\mathbf{L}_{\mathbf{x}} = \begin{bmatrix} \mathbf{L}_{\mathbf{x}_1} \\ \vdots \\ \mathbf{L}_{\mathbf{x}_k} \end{bmatrix} \tag{2.12}$$

Interaction matrix of $\theta\mathbf{u}$ parameterization (3D): If the rotation matrix ${}^{c^*}\mathbf{R}_c \in SO(3)$ relating the two views of the camera is defined in $\theta\mathbf{u}$ parameterization, the error function

is given by $\mathbf{e} = ({}^{c^*}\mathbf{t}_c, \theta\mathbf{u})$ where ${}^{c^*}\mathbf{t}_c \in \mathbb{R}^k$ is the translation vector. The camera velocity related to the pose error is given by [Malis 99]:

$$\dot{\mathbf{e}} = \begin{bmatrix} {}^{c^*}\mathbf{R}_c & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_\omega(\mathbf{u}, \theta) \end{bmatrix} \mathbf{v} \tag{2.13}$$

where

$$\mathbf{L}_\omega(\mathbf{u}, \theta) = \mathbf{I}_3 + \frac{\theta}{2}[\mathbf{u}]_\times + \left(1 - \frac{\text{sinc}(\theta)}{\text{sinc}^2(\frac{\theta}{2})}\right)[\mathbf{u}]_x^2 \tag{2.14}$$

with $[\mathbf{u}]_\times$ is the skew symmetric matrix associated with $\mathbf{u}$ and $\mathbf{I}_3 \in \mathbb{R}^{3\times3}$ is the identity matrix.

2.4.2 Photometric features

Recently, different researches focus on the utilization of photometric features computed from pixel intensities. The utilization of the photometric features does not rely on complex image processing such as feature extraction, matching, and tracking process, contrary to utilizing geometric visual features such as points, straight lines, pose, homography, etc. Moreover, it is not very sensitive to partial occlusions and to coarse approximations of the depths required to compute the interaction matrix. This approach is employed by considering the whole image as a feature set from which the control input signal is defined. The input to the controller can belong to eigenspace or kernel of image pixel or can also be defined as the set of all image pixels itself.

Eigenspace-based photometric features

In [Nayar 96], image intensity is not used directly but an eigenspace decomposition is performed first to reduce the dimensionality of image data. Then the control is performed in the eigenspace. This method requires off-line computation of this eigenspace and the projection of the image on this subspace for each new frame. No analytical form is available for the interaction matrix, indeed it is learned during an off-line step. The learning process has to be done for each new object and requires the acquisition of many images of the scene at various camera positions.

Kernel-based photometric features

Another approach that considers the pixels intensity is based on the use of kernel methods. It is presented in [Kallem 07]. This method, valid for 4 DOFs, leads to a decoupled control law that decouples translation and rotation around the optical axis.

Pixel-based photometric features

It is also possible to use the luminance of all the pixels in the image as visual feature set. In that case, $\mathbf{s} = \mathbf{I}(\mathbf{r})$ where $\mathbf{I}$ is a vector of the same size $M_i \times N_i$ as the image, [Collewet 08]

[Collewet 09b]. In the same manner, instead of using the luminance $\mathbf{I}(\mathbf{r})$, color attributes $\mathbf{C} = (R,G,B)$ can be used as visual features by selecting the three components R, G or B, [Collewet 09a]. A combination of different components can also be performed.

2.4.3 Velocity field features

In [Crétual 01], the visual features are specified with dynamic criteria which is homogeneous to speed in the image and the relation between the variations of velocity field features and the camera velocity is modeled. The camera motions are controlled so that the current velocity field in the image becomes equal to motion field in the desired configuration. This approach is used for positioning a camera parallel to a plane and following trajectory. In [Kelly 04], the application of the velocity field control is applied to visual servoing of robot manipulator under fixed camera configuration. In that case, the task to be accomplished by the robot is coded by means of a smooth desired velocity vector field defined in the image space by $\mathbf{v}(\mathbf{x}) : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, where $\mathbf{x} \in \mathbb{R}^2$ is an image point and $\mathbf{v}$ is assumed to be bounded. This desired velocity field $\mathbf{v}$ defines a tangent vector that represents the desired image feature velocity $\dot{\mathbf{x}}$ at every point of the image space. The velocity field error $\mathbf{e}$ is defined as the difference between the desired velocity field $\mathbf{v}(\mathbf{x})$ and the image feature velocity $\dot{\mathbf{x}}$. Thus, instead of requiring the image feature to be at a specific location at each instant time as it is imposed in trajectory tracking control, in velocity field control the image feature is guided towards the desired flow defined by the desired velocity field $\mathbf{v}(\mathbf{x})$ i.e. the image feature will match with the flow lines of the desired velocity field. The velocity field is also used in [Kelly 06], for controlling wheeled nonholonomic mobile robots by a fixed camera.

2.5 Visual servoing control laws

As for the choice of the control law [Espiau 92] [Wilson 96] [Hutchinson 96] [Chaumette 98] [Deng 02b] [Malis 04] [Chaumette 06] [Chaumette 07], it affects the behavior of the system. As described in the previous section, the nature of the potential visual features is extremely rich. In the control design phase, a number of properties should be considered such as local and global stability, robust behavior with respect to measurement and modeling errors, local or global exponential decrease, order of convergence, absence of local minima and singularities, obtaining suitable robot trajectory, and finally the degree of decoupling between the visual information and the controlled degrees of freedom.

2.5.1 Classical approaches

2.5.1.1 Position-based visual servoing

In Position-based visual servoing (PBVS) or 3D visual servoing, the task function is expressed in the Cartesian space. The translation and the rotation of the camera in Cartesian space are explicitly reconstructed using pose estimation, [Hutchinson 96] [Wilson 96] [Martinet 97] [Basri 99] [Taylor 00] [Deng 03]. PBVS is known to have global asymptotic

stability referring to the ability of a controller to stabilize the pose of the camera from any initial condition if 3D estimation is perfect. The analytical proof is evident if the pose is perfect, otherwise it is impossible. When accurate 3D estimation is employed, decoupling rotation and translation is obtained. Errors in calibration propagate to errors in the 3D world, so it is required to ensure robustness of PBVS [Kyrki 04a].

No mechanism in PBVS ensures keeping the features visible within the camera field of view (FOV) when the translation is defined in the desired fixed end effector frame [Wilson 96]. While, if the translation is expressed in the camera frame, the trajectory in the image plane is improved under large camera rotation motion and features can be kept in the image plane for small rotation, [Deng 03]. Several control schemes have been proposed to overcome the latter problem (for example 2.5 D visual servoing presented in [Malis 02] or nonlinear approach using a new 3D translation features in the control loop as presented in [Martinet 99] [Thuilot 02]). In PBVS, the task function to be regulated is usually defined as the error between current and desired poses. The pose can also be selected as the pose of the camera or the end effector with respect to the object or any other reference frame in the world space. When the pose between the camera and the object is considered, the task function is given by $\mathbf{e} = {}^{c^*}\mathbf{P}_c$. After executing the task, the camera reaches the desired position and the task function $\mathbf{e} = {}^{c^*}\mathbf{P}_{c^*} = \mathbf{0}$.

Pose estimation

One of the central problems in position-based visual servoing is the determination of the relative position and orientation of the observed object, that is the pose with respect to the camera. For real time pose estimation of the object, image measurements are combined with the known object CAD description. The pose can be estimated using image points [Haralick 89] [Dementhon 95][Liu 99] [Ansar 03] [Chesi 09a], using point and line correspondence [Dornaika 99], using point to region correspondence [Mcinroy 08], using curves [Safaee-Rad 92], or using other different geometrical features as in virtual visual servoing [Marchand 02]. For obtaining more accurate pose estimation, different filters are usually used to estimate its translational and rotational parameters [Myers 76] [Wang 92] [Ficocelli 01] [Lippiello 04] [Shademan 05] and recently [JanabiSharifi 10b].

Partial pose estimation

Camera translation (up to a scale factor) and camera rotation can be estimated through the Essential matrix [LonguetHiggins 81][Faugeras 93] [Hartley 97]. However, when the target is planar or when the motion performed by the camera between the desired and the current pose is a pure rotation, essential matrix cannot be estimated and it is more appealing to estimate the pose using a homography matrix [Malis 00]. Indeed, if all points belong to a plane π, the partial pose can be extracted from the homography matrix $\mathbf{H}$ that relates homogeneous coordinates of points in the desired and current image planes π using the

relation:

$$\mathbf{H} = {}^{c^*}\mathbf{R}_c + \frac{{}^{c^*}\mathbf{t}_c \mathbf{n}^{*\top}}{d^*} \tag{2.15}$$

where ${}^{c^*}\mathbf{R}_c$ is the rotation matrix and ${}^{c^*}\mathbf{t}_c$ is the translation vector between the two camera frames, $\mathbf{n}^*$ is the unit normal vector of the plane π expressed in the desired camera frame, and d^* is the distance from the camera origin at the desired position to the plane π.

2.5.1.2 Image-based visual servoing

IBVS uses feature extracted directly from the image as input of the control law, without any supplementary estimation step. Different from PBVS, IBVS does not need to estimate the pose at each iteration which helps to provide a robust positioning control against calibration and modeling errors. For example, lens distortions can be neglected when the set of desired image feature locations is obtained via an off-line teach-by-showing procedure by moving the robot end-effector to a desired location and storing the camera measurements at this location.

IBVS is characterized by several advantages. Firstly, when the set of features is composed of the Cartesian coordinates of image points, these image points are controlled in the image plane to move approximately along straight lines. Therefore the target can be constrained to remain visible during the execution of a task if both the initial and desired image feature locations are within the camera field of view [Chaumette 98]. Another advantage of IBVS is that the positioning accuracy is insensitive to camera and target modeling errors [Espiau 93] [Hutchinson 96] [Hager 97] [Mezouar 02]. It is essentially a model-free method without explicit requirement of the target model in practical applications when the depths of the feature points are available. IBVS systems are usually preferred to position-based systems since they are usually less sensitive to image noise. However, some knowledge of the transformation between the sensor and the robot frame [Tsai 89] is still required.

IBVS is however subject to some shortcomings. Firstly, it is hard to predict the trajectory of the end effector and robot may reach its joint limits. Secondly, the end-effector translational and rotational motions are not directly controlled and the usual coupling existing between these motion makes it difficult to plan a pure rotation or a pure translation (for example, due to the camera retreat problem [Chaumette 98]). Also, since the system is usually highly coupled, the analytical domain of system stability is impossible to determine in the presence of camera modeling errors. Furthermore, usual IBVS is only locally asymptotically stable and may fail in the presence of large displacement to realize [Chaumette 98] [Cervera 99], which necessitates a path planning step to split a large displacement up in smaller local movements [Mezouar 02]. Finally, potential failure occurs when IBVS is subject to image singularities or local minima [Chaumette 98].

Depth estimation

In IBVS, providing some information about the depth of the object in the camera frame is usually necessary for the computations required to obtain the interaction matrix. Since the stability region for the error in depth estimation is not very large [Malis 10], it is necessary to accurately estimate the depth. For static objects, this estimation of the depth value can be obtained from the measurement of the current values of the feature points x and y and their image motion $\dot{x}$ and $\dot{y}$, and of the camera velocity [Matthies 89] [Marcé 87]. The depth parameters of planar and volumetric parametric primitives like points, lines, cylinders, spheres, etc. can be also obtained [Chaumette 96]. Another depth estimation method for static points without the explicit need for image motion estimation can be found in [DeLuca 08b]. In [Xie 09], a laser pointer is used and the depth estimation can be achieved through a triangulation method.

In response to the difficulties appearing in IBVS and PBVS, several methods that do not rely solely on the interaction matrix have been devised to improve the behavior of the visual servoing controls.

2.5.2 Enhanced visual servoing approaches

In order to overcome drawback of visual servoing control schemes, different modeling approaches have been considered such as sliding approaches [Zanne 00], partitioning approaches [Corke 01] [Kyrki 04b] [Pages 06], trajectory planning approaches [Mezouar 02], varying-feature-set approaches as presented in [Comport 06] [GarciaAracil 05] [Mansard 09b], and hybrid and switching approaches which will be discussed in a particular section (Section 2.5.3).

Sliding approaches

Sliding mode control theory has been used to design a controller which is robust to bounded parametric estimation errors (uncertainties) [Zanne 00]. Two sources of error are considered, the control input error due to uncertainties in the positioning of the camera relative to the robot end-effector, and the estimation error of the pose reconstruction algorithm used in the control loop. The formulation of the sliding mode controller is based on the quaternion representation of rotations. The controller is designed such that the estimated state precisely tracks the desired trajectory in the presence of uncertainties.

Partitioning approaches

A partitioned approach had been proposed for decoupling the rotation and the translation around and along the z-axis from the rotation and the translations regarding x-axis and y-axis, [Corke 01]. This decoupling is used to solve the problem corresponding to a pure rotation by 180 degrees around the camera optical axis which leads to a singularity [Chaumette 98]. In

[Corke 01], a controller is designed for the motion along and around the optical axis which can avoid the backward motion of the camera.

$$\dot{\mathbf{s}} = \mathbf{J}_{xy}\dot{\mathbf{r}}_{xy} + \mathbf{J}_z\dot{\mathbf{r}}_z \Longleftrightarrow \dot{\mathbf{r}}_{xy} = \mathbf{J}_{xy}^{+}(\dot{\mathbf{s}} - \mathbf{J}_z\dot{\mathbf{r}}_z) \tag{2.16}$$

where $\dot{\mathbf{r}}_{xy} = (\boldsymbol{v}_x, \boldsymbol{v}_y, \omega_x, \omega_y)$ and $\dot{\mathbf{r}}_z = (\boldsymbol{v}_z, \omega_z)$. Selecting suitable $\boldsymbol{v}_z$ and ω_z, the partitioned control scheme eliminates the retreat problem. This partitioning approach is also used to keep the feature points in the camera field of view by establishing a repulsive potential function at the boundary of the image to enforce the camera to move in the negative direction of its optical z-axis to keep the feature points within the image plane.

In [Pages 06], a structured light sensor is designed for image-based visual servoing such that the projected pattern on the object is invariant to depth and the corresponding image is symmetric. This approach, similar to a position-based approach, decouples the rotational part from the translational one. Object plane is not required to be reconstructed by using triangulation or a non-linear minimization algorithm. This method requires a coarse calibration of camera and lasers location with respect to the robot end effector. To improve the robustness with respect to misalignment between the camera and the lasers, an image transformation is defined.

Trajectory planning

In [Mezouar 02], rather than the utilization of direct point-to-point image-based visual servoing when a large displacement has to be realized, a coupling between a path planning step based on the potential field approach [Khatib 85] and a purely image-based control is performed to extend the local robustness and stability of image-based control. The required trajectory of the camera is planned with respect to stationary desired camera end-effector frame. This ensures the shortest straight line in the Cartesian space in the absence of constraints. This method considers the joint limits problems and keeping the features in the field of view of the camera.

In [Chesi 09b], a new framework solution is presented for avoidance in visual servoing using homogeneous forms and linear matrix inequality. This method makes it possible to consider different kind of constraints by imposing them as a positively conditions on homogeneous form into the proposed general parameterization of the trajectory planned from the initial to the desired position. Convex optimization is used in this framework in order to determine the solution and maximize the performance of these path planning constraints.

Origin-shift in cylindrical coordinates

As already said in section 2.4.1, in [Iwatsuki 05], a formulation of visual servoing based on the cylindrical coordinate system has been proposed to avoid the camera backward motion when the motion from the initial to the desired poses is a pure rotation of 180 around the optical axis. In that paper, a decision method of a shifted position of the origin by estimating

the camera rotational motion is also proposed. Simulations and experimental results have shown the effectiveness of the proposed approach by comparing the conventional approaches and the new one.

Varying-feature-set control schemes

In [Comport 06], a weighted error function $\mathbf{e}_w$ is used in the control law. It is defined as $\mathbf{e}_w = \mathbf{W}\mathbf{e}$ where $\mathbf{e} = \mathbf{s} - \mathbf{s}_d$ is the usual error between the current and desired visual features and $\mathbf{W} = \text{Diag}(w_1, w_2, ..., w_k)$ is a diagonal weighting matrix. The corresponding control scheme is:

$$\mathbf{v} = -\lambda(\mathbf{W}\mathbf{J})^{+}\mathbf{W}\mathbf{e} \tag{2.17}$$

where $(\mathbf{W}\mathbf{J})^{+}$ is the Moore-Penrose pseudo inverse of $\mathbf{W}\mathbf{J}$. The values of $w_i \in [0, 1]$ is smoothly changing to a value that indicates the level of confidence in the corresponding feature (w_i with 1 refers to the highest confidence level). This removes the outliers from the feature set $\mathbf{s}$. Errors in feature extraction, tracking and matching are rejected at the low level control which improves the behavior of the positioning task. In [GarciaAracil 05], to ensure the continuity of the control law when some features leave the camera field of view, i.e., when the number of features changes, a diagonal weighted matrix is also introduced into the definition of the task function.

In [Mansard 09b], a theoretical study of using the weighted matrix when a classical pseudo inverse is used shows that the continuity of the control scheme is not ensured when the activated feature set is non redundant except when the activated feature is fully decoupled. A control law based on a new inverse operator that is able to keep the continuity of the control law regardless the change of the rank of the Jacobian is presented.

2.5.3 Hybrid and switching strategies in visual servoing

To combine the advantages of both image-based (2D) and position-based (3D) visual servoing, numerous approaches have been proposed to model control schemes based on the utilization of hybrid and switching strategies. As for the hybrid approaches when both 2D and 3D features are used simultaneously in the same control scheme, we find 2 1/2 visual servoing [Malis 99] [Chaumette 00] [Deguchi 98]. Switching approaches between PBVS and IBVS are used to capture the strength of each one when the other being in a weakness configuration, [Gans 03a] [Gans 03b] [Gans 07]. Hybrid and switching approaches are used together by integrating PBVS and IBVS in a switching hybrid scheme, [Deng 05] [Hafez 06] [Hafez 07b] [Hafez 08b]. A hybrid approach in feature parameterization is presented in [Corke 09]. And finally, planning step-switching using laser spot is presented in [Xie 09].

2.5.3.1 Hybrid approaches

A 2 1/2 D visual servoing approach developed in [Malis 99] [Chaumette 00] combines visual features expressed in 2D image and 3D Cartesian spaces. This hybrid controller tries

to take advantage of both approaches by decoupling rotation and translation and keeping the features in the field of view. The task function is expressed in both Cartesian space and in the image space. These approaches make it possible to demonstrate also the global asymptomatically stability and the robustness of the control law when a minimal number of visual features remain in the camera field of view [Malis 02]. The camera displacement between two views is estimated using the correspondence between several points in two images to extract the relative rotation between the two images and the translation up to a scale factor [Malis 00] [Benhimane 07]. This decoupling between the orientation and the translation makes it possible to design two task functions: $\mathbf{e_v}$ for the translation and $\mathbf{e}_\omega$ for the rotation which improves the system behavior in the 3D space. A very interesting aspect in 2 1/2 D visual servoing is that, thanks to projective reconstruction, the knowledge of the 3D structure of the considered targets is no more necessary. However, the use of projection reconstruction implies that the corresponding control laws are more sensitive to image noise than classical image-based visual servoing.

In [Kyrki 04b], a solution similar to [Chaumette 00] is presented. A hybrid control scheme between 3D and 2D visual servoing is presented based on controlling the translation using the estimated 3-D translation between the current and desired poses while the rotation is controlled using a single feature point that is driven towards its desired location.

Another hybrid system between IBVS and PBVS is presented in [Hafez 07b]. Instead of partially combining 2D and 3D visual information, a hybrid objective function to be optimized is defined by concatenating both 2D and 3D errors. The solution thus minimizes the errors in both image space and pose space simultaneously. This is obtained by defining a cost function as:

$$\mathbf{E}(\mathbf{s}(\mathbf{p})) = \frac{\lambda_{2d}}{2}\mathbf{e}_{2d}(\mathbf{p})^\top \mathbf{e}_{2d}(\mathbf{p}) + \frac{\lambda_{3d}}{2}\mathbf{e}_{3d}(\mathbf{p})^\top \mathbf{e}_{3d}(\mathbf{p}) \tag{2.18}$$

where $\mathbf{e}_{2d}(\mathbf{p}) = \mathbf{s}_{2d}(\mathbf{p}) - \mathbf{s}_{2d}(\mathbf{p}^*)$ is the image error vector, $\mathbf{e_{3d}}(\mathbf{p}) = \mathbf{s}_{3d}(\mathbf{p}) - \mathbf{s}_{3d}(\mathbf{p}^*)$ is the pose error vector, and λ_{2d} and λ_{3d} are positive scalar factors to define the integration weight of 2D and 3D spaces. Using the classical approach, the control scheme is given by:

$$\mathbf{v} \;=\; -\lambda \begin{bmatrix}\mathbf{L}_{2d}(\mathbf{p})\\ \mathbf{L}_{3d}(\mathbf{p})\end{bmatrix}^+ \begin{bmatrix}\Gamma_{2d}\\ \Gamma_{3d}\end{bmatrix} \begin{bmatrix}\mathbf{e}_{2d}(\mathbf{p})\\ \mathbf{e}_{3d}(\mathbf{p})\end{bmatrix} \tag{2.19}$$

where $\mathbf{L}_{2d}$ and $\mathbf{L}_{3d}$ are respectively the interaction matrices to $\mathbf{e}_{2d}$ and $\mathbf{e}_{3d}$ and $\Gamma_* = \text{diag}(\lambda_*), * = 2d, 3d$. This cost function has common global minima as 2D and 3D approaches. This hybrid control scheme improves the performance in image and Cartesian spaces by decreasing the probability of losing image feature and enhances the camera trajectory in the Cartesian space.

In [Hafez 07a], a boosted visual control that considers 2D and 3D control as weak controls is presented. A linear combination of these two control produce a strong one with satisfactory performance. Weights which are used in the combination are computed based on error function associated with each control.

2.5.3.2 Switching approaches

In [Gans 03a, Gans 03b, Gans 07], switching is performed in order to avoid the problems of loosing feature points using PBVS and reaching robot joint limits when IBVS is considered. In this switching system, a threshold region is defined as a circle in the image space to start switching to IBVS if the features nears the image boundaries. Another threshold region is defined in the robot configuration space in which the object is visible so that the switching to PBVS starts when the robot nears its joint limits. Two metric functions are defined to decide when switching should start. Finally, for the position-based and image-based visual servoing given by $\mathbf{v}_{3d} = -\lambda_{3d}\mathbf{A}_{3d}\mathbf{e}_{3d}$ and $\mathbf{v}_{2d} = -\lambda_{2d}\mathbf{A}_{2d}\mathbf{e}_{2d}$ (where λ_{3d} and λ_{2d} are positive gains, $\mathbf{e}_{3d}$ and $\mathbf{e}_{2d}$ are control input signals and $\mathbf{A}_{3d}$ and $\mathbf{A}_{2d}$ are the control matrices), a common state space is defined by concatenating the state vectors such that:

$$\begin{bmatrix} \dot{\mathbf{e}}_{3d} \\ \dot{\mathbf{e}}_{2d} \end{bmatrix} = \lambda_\sigma \mathbf{A}_\sigma \begin{bmatrix} \mathbf{e}_{3d} \\ \mathbf{e}_{2d} \end{bmatrix}, \quad \sigma \in 3d, 2d \tag{2.20}$$

where

$$\mathbf{A}_{3d} = -\begin{bmatrix} \mathbf{I}_6 & \mathbf{0} \\ \mathbf{L}_{2d}\mathbf{L}_{3d}^{-1} & \mathbf{0} \end{bmatrix}, \qquad \mathbf{A}_{2d} = -\begin{bmatrix} \mathbf{0} & \mathbf{L}_{3d}\mathbf{L}_{2d}^{+} \\ \mathbf{0} & \mathbf{I}_6 \end{bmatrix} \tag{2.21}$$

The hybrid switching system presented in [Gans 03a] [Gans 03b] is defined by a convex combination:

$$\mathbf{A}_c = \alpha\mathbf{A}_{3d} + (1-\alpha)\mathbf{A}_{2d}, \quad \alpha \in [0,1] \tag{2.22}$$

where α is a time-based switching parameter. The system control is thus partitioned along the time while in [Gans 07] a state-based switching parameter is used. Stability analysis presented in [Gans 07] shows that under arbitrary switching within a sufficiently small neighborhood of the origin, a hybrid switching visual servo system is asymptotically stable in the sense of Lyapunov. Moreover, the state based switching local stability is proved using a technique for establishing the stability of switched control systems [Branicky 97] [Wicks 94].

The switching method proposed in [Gans 07] is used for mobile robots navigation in corridors in [Toibero 09] to enhance the performance of the navigation system. The switching helps to deal with adverse initial conditions that could be difficult to handle when only PBVS or IBVS is considered. The switching helps also to handle the sudden change on the position of the features due to disturbances. By considering Multiple Lyapunov Functions, the stability of the controller used has been proved.

In [Deng 05], a system was designed to resolve the problems of local minima and singularities. This is achieved by performing the switching from IBVS to PBVS as soon as the end-effector motion nears the neighborhood of image singularity or image local minima configurations. The evaluation of the image singularity is performed by comparing the image motion perceptibility measure with image singularity threshold values, [Sharma 97]. As for the local minima, to avoid the exhaustive searching for image local minima, they evaluate the local minima only on the planned image trajectories.

In [Xie 09], a switching approach to decouple camera rotation and camera translation motions is proposed to apply IBVS to a robotic assembly system composed of a 6-DOF robot with a camera and a laser pointer mounted on the robot end effector. The laser pointer is used in order to decouple the camera rotation and translation. A required motion to be performed is considered as a composition of two steps, rotation step and translation step. In the first step, an imaginary image feature defined by laser-spot is used to control the rotational motion of the end effector to reach the configuration where the laser spot is projected on the object. In the second step, the interaction matrix constructed from the features of the object and laser spot is used to control the end-effector translational motion with respect to the object. This control scheme was used for eye-in-hand robotic manufacturing system to detect, grasp, and assemble a planar object on a main body.

2.5.3.3 Hybrid switching approaches

An integration between PBVS and IBVS proposed in [Hafez 06] is performed by computing the weighted sum of the velocity values obtained from each controller individually. The integrated controller given by this framework is defined by

$$\mathbf{v} = \omega \mathbf{v}_{2d} + (1 - \omega)\mathbf{v}_{3d} \tag{2.23}$$

where $\mathbf{v}_{2d}$ and $\mathbf{v}_{3d}$ are the velocity vectors of the IBVS and PBVS controllers and are given by $\mathbf{v}_{2d} = -\lambda_{2d}\mathbf{L}_{2d}^{+}\mathbf{e}_{2d}$, and $\mathbf{v}_{3d} = -\lambda_{3d}\mathbf{L}_{3d}^{-1}\mathbf{e}_{3d}$, respectively. ω is defined as a function of the constraints in image and Cartesian spaces and the energy of the task function. This function is designed such that near to image local minima and singularities its value decreases to switch to PBVS while when one of the image features nears the image boundary, its value increases to switch to IBVS. A probabilistic general formula can represent this framework by defining $\mathbf{v}$ as:

$$\mathbf{v} = \sum_{\alpha_i} \mathbf{v}(\alpha_i) p(\alpha_i|x) \tag{2.24}$$

where $p(\alpha_i|x)$ is the discrete probability for visual servoing control law parameters $\alpha_i, i = 1, .., k$ conditioned on the image measurement x and $\sum_{\alpha_i} p(\alpha_i|x) = 1$.

In [Hafez 08b], the same methodology used in [Hafez 06] is used for avoiding joint limits and keeping features in the camera field of view. By considering the hybrid approach as a boosting which produces a strong algorithm from two weak algorithms represented by the IBVS and PBVS ones. In that case, the weight ω introduced in (2.23) is given by $\omega = \frac{\alpha_{2d}}{\alpha_{2d}+\alpha_{3d}}$ where the thresholds of joint limits and the threshold representing the nearest image border are used to compute the weights α_{2d} and α_{3d} that give the current importance to each algorithm.

2.5.3.4 Hybrid and/or switching strategies involving other approaches

Switching approaches previously mentioned combine two different visual servoing approaches. Now, we address some cases when switching is performed between visual servoing and other approaches: for example switching between visual servoing and artificial potential [Hashimoto 00a], hybrid between visual and kinematic measurements [Taylor 04] and finally switching between PBVS and other selected rotational or translational control as in [Chesi 04a].

In order to avoid local minima problems in IBVS, in [Hashimoto 00a], a switching method based on defining artificial potential $\mathbf{v}^a$ using interpolation between the initial and the desired images is proposed. When a visual servoing system nears a neighborhood of the local minima, switching from the visual servoing velocity $\mathbf{v}$ to $\mathbf{v}^a$ is performed to reach a local maximum. Then the system switches again to the original $\mathbf{v}$.

A switching approach between PBVS and some selected rotational or translational motions is presented in [Chesi 04a] for keeping features in the field of view. It consists of realizing a visual servoing when all points lie inside the image. When at least one point lies on the image boundary, system control switches to the other selected controls in order to push such a point inside the image and to guarantee the final convergence. Only rotational control or translational control is applied when the corresponding produced motions moves the points lying on the image boundary inside the image. Otherwise, a backward translational motion along the optical axis is applied to sent the camera away from the observed object.

To increase the accuracy and robustness of controlling humanoid robot having a grasping system and working in an unstructured domestic environment, a hybrid approach that combine kinematic measurements and visual measurements is presented in [Taylor 04]. In this approach, using kinematic measurements increases the robustness to visual distractions and occlusion while visual measurements increase the positioning accuracy. They introduced online estimation of the eye-in-hand transformation to enhance the influence of calibration errors on camera and kinematic models. Iterative extended Kalman filter (IEKF) was used to fuse the visual and kinematic measurements obtained from the camera-robot system to overcome the problems of initial pose error in long term operations and to decrease the influences of the calibration errors.

A hybrid controller for robust mobile manipulation is developed in [Wang 10] by integrating the classical IBVS controller and a machine-learning approach called reinforcement learning or Q-learning controller through a rule-based arbitrator which defines the control behavior policy. This integration autonomously improves its performance by ensuring visibility of visual features. This is thought to be the first paper that integrates Q-learning with visual servoing to achieve robust operation. Experimental validation carried out to this approach shows that the hybrid controller possesses the capabilities of self-learning and fast response, and provides a balanced performance with respect to robustness and accuracy.

2.5.4 Visual servoing as a minimization problem

Visual servoing can be viewed as a minimization problem that finds a camera displacement vector $\Delta\mathbf{p}$ to minimize a cost function $\mathbf{E}(\mathbf{s}(\mathbf{p}))$ of the error vector $\mathbf{s}(\mathbf{p})$ where $\mathbf{p} \in \mathbb{R}^3 \times \mathbf{SO^3}$ is the camera pose [Malis 04] [Hafez 07b]. By considering the problem as a nonlinear least squares minimization such that $\mathbf{E}(\mathbf{s}(\mathbf{p})) = \frac{1}{2}\Delta\mathbf{s}^\top\Delta\mathbf{s}$, where $\Delta\mathbf{s} = (\mathbf{s}(\mathbf{p}_i) - \mathbf{s}(\mathbf{p}_d))$ with $\mathbf{p}_i$ and $\mathbf{p}_d$ the initial and the desired pose, Taylor expansion performed on the gradient of $\mathbf{E}(\mathbf{s}(\mathbf{p}))$, that is on $\Delta\mathbf{s}(\mathbf{p})$, can be used to obtain first and second order minimizations.

2.5.4.1 Steepest decent minimization method (SDM)

Using first order Taylor expansion of $\Delta\mathbf{s}(\mathbf{p})$ evaluated at $\mathbf{p}_d$ around $\mathbf{p}_i$ we get:

$$\mathbf{s}(\mathbf{p}_d) = \mathbf{s}(\mathbf{p}_i) + \left.\frac{\partial\mathbf{s}(\mathbf{p})}{\partial\mathbf{p}}\right|_{\mathbf{p}=\mathbf{p}_i} \Delta\mathbf{p} \tag{2.25}$$

where $\frac{\partial\mathbf{s}(\mathbf{p})}{\partial\mathbf{p}} = \mathbf{L}_{\mathbf{s}(\mathbf{p})}$ and $\Delta\mathbf{p} = \mathbf{p}_d - \mathbf{p}_i$. In the gradient decent minimization, the movement is in the opposite direction to the gradient and the change in the pose is given by [Malis 04]:

$$\Delta\mathbf{p} = -\lambda\mathbf{L}^\top_{\mathbf{s}(\mathbf{p})}\Delta\mathbf{s} \tag{2.26}$$

where λ is a positive gain and $\mathbf{L}_\mathbf{s} = \mathbf{J}_\mathbf{s}\mathbf{P}$ such that $\dot{\mathbf{p}} = \mathbf{P}\mathbf{v}$. The method presented above is called Jacobian transpose method JTM and has been used in [Hashimoto 93b, Kelly 00, Hafez 07b]. This method shows linear convergence rate.

2.5.4.2 Newton minimization method (NM)

In order to obtain quadratic convergence rate, second order Taylor expansion is used for evaluating the cost function at the desired pose $\mathbf{p}_d$:

$$\mathbf{s}(\mathbf{p}_d) = \mathbf{s}(\mathbf{p}_i) + \left.\frac{\partial\mathbf{s}(\mathbf{p})}{\partial\mathbf{p}}\right|_{\mathbf{p}=\mathbf{p}_i} \Delta\mathbf{p} + \frac{1}{2}\Delta\mathbf{p}^\top \left.\frac{\partial^2\mathbf{s}(\mathbf{p})}{\partial\mathbf{p}^2}\right|_{\mathbf{p}=\mathbf{p}_i} \Delta\mathbf{p} \tag{2.27}$$

where the second order derivative is a function of Hessian matrix $\mathbf{H}(\mathbf{s}(\mathbf{p}))$ given by:

$$\frac{\partial^2\mathbf{s}(\mathbf{p})}{\partial\mathbf{p}^2} = \mathbf{L}^\top_{\mathbf{s}(\mathbf{p})}\mathbf{L}_{\mathbf{s}(\mathbf{p})} + \sum_{j=0}^{n} \mathbf{H}^j_{\mathbf{s}(\mathbf{p})}\Delta\mathbf{s} \tag{2.28}$$

By injecting (2.28) in (2.27) and solving for $\mathbf{p}$ we get:

$$\mathbf{v} = -\lambda\left(\mathbf{L}^\top_{\mathbf{s}(\mathbf{p}_i)}\mathbf{L}_{\mathbf{s}(\mathbf{p}_i)} + \sum_{j=0}^{n} \mathbf{H}^j_{\mathbf{s}(\mathbf{p_i})}\Delta\mathbf{s}_j\right)^{-1} \mathbf{L}^\top_{\mathbf{s}(\mathbf{p}_i)}\Delta\mathbf{s} \tag{2.29}$$

Even if the Newton minimization method has a higher convergence rate than Steepest decent minimization method, convergence problems can appear when Hessian matrix is negative definite.

2.5.4.3 Approximation to Newton minimization method (NMM)

The convergence problem mentioned above is avoided by using a positive definite approximation to replace Hessian matrix. This replacement gives approximations to Newton minimization method such as Gauss-Newton minimization method, Levenberg-Marquardt method and Quasi-Newton Method.

Gauss-Newton minimization method (GNM)

The first one is Gauss-Newton minimization method (GNM) in which the second order derivative $\frac{\partial^2 \mathbf{s(p)}}{\partial \mathbf{p}^2}$ is set as $\mathbf{L}_{\mathbf{s(p)}}^{\top}\mathbf{L}_{\mathbf{s(p)}}$ by neglecting $\sum_{j=0}^{n} \mathbf{H}_{\mathbf{s(p)}}^{j}\Delta\mathbf{s}$ in (2.29). The control law obtained is:

$$\mathbf{v} = -\lambda(\mathbf{L}_{\mathbf{s(p)}}^{\top}\mathbf{L}_{\mathbf{s(p)}})^{-1}\mathbf{L}_{\mathbf{s(p)}}^{\top}\Delta\mathbf{s} \tag{2.30}$$

which is the classical control scheme using the pseudo inverse, [Espiau 92].

Levenberg-Marquardt method (LMM)

The second method is Levenberg-Marquardt method (LMM) or Damped Least Squares method (DLS). It has been firstly proposed in [Wampler 86]. This method is obtained from Newton method by replacing the Hessian part by $\gamma\mathbf{D}$:

$$\mathbf{v} = -\lambda \left(\mathbf{L}_{\mathbf{s(p)}}^{\top}\mathbf{L}_{\mathbf{s(p)}} + \gamma\mathbf{D}\right)^{-1} \mathbf{L}_{\mathbf{s(p)}}^{+}\Delta\mathbf{s} \tag{2.31}$$

where $\mathbf{D}$ is any selected diagonal positive matrix and γ is a parameter used to determine the convergence rate. When γ is large, Levenberg-Marquardt method shows convergence rate similar to that of Steepest decent minimization method. While when γ is small, the convergence rate obtained approaches the convergence rate of Gauss-Newton minimization method. Finally, when $\gamma = 0$, the LMM method is equivalent to GNM method. This method is usually used to solve the singularity problem in robotics [Deo 92] [Chiaverini 97].

Quasi-Newton Method (QNM)

In this method, an approximation of Hessian matrix is reached by defining a sequence of symmetric definite matrices $\mathbf{A}_k$ that converges to the Hessian matrix [Shanno 70].

$$\mathbf{v} = -\lambda {\mathbf{A}_k}^{-1}\mathbf{L}_{\mathbf{s(p)}}^{\top}\Delta\mathbf{s} \tag{2.32}$$

where the initial value of $\mathbf{A}_0$ is set to be $\mathbf{I}$ or $\mathbf{L}_{\mathbf{s(p)}}^{\top}\mathbf{L}_{\mathbf{s(p)}}$ to start by coinciding with steepest method or Gauss-Newton method respectively. This method has been applied for visual servoing in [Piepmeier 99]. Although using transpose of $\mathbf{L}_{\mathbf{s(p)}}$ ensures the continuity in the control scheme even when the rank of $\mathbf{L}_{\mathbf{s(p)}}$ changes (when some features are lost for instance), it can lead to non optimal control. That is why it is preferable usually to use the pseudo inverse of $\mathbf{L}_{\mathbf{s(p)}}$ [Espiau 92].

Efficient second order method (ESM)

Classical image-based visual servoing methods can be considered as problems of low convergence rate and their performance can be studied in terms of the rate of convergence of their minimization problems. However, convergence problems can be obtained when computing the second order derivatives which is subject to ill-conditions. In [Malis 04], two control schemes are proposed by formulating the positioning problem as a minimization problem. The two schemes proposed give higher convergence rate while requiring only computing the first order derivatives. The first proposed second order approximation is the mean of the pseudo inverse of $\mathbf{L}_{\mathbf{s}(\mathbf{p}_i)}$ and $\mathbf{L}_{\mathbf{s}(\mathbf{p}_d)}$. It is given by:

$$\mathbf{v} = -\frac{\lambda}{2}(\mathbf{L}^{+}_{\mathbf{s}(\mathbf{p}_i)} + \mathbf{L}^{+}_{\mathbf{s}(\mathbf{p}_d)})\Delta\mathbf{s} \tag{2.33}$$

The second scheme is obtained by computing the pseudo inverse of the mean of $\mathbf{L}_{\mathbf{s}(\mathbf{p}_i)}$ and $\mathbf{L}_{\mathbf{s}(\mathbf{p}_d)}$. It is given by:

$$\mathbf{v} = -2\lambda(\mathbf{L}_{\mathbf{s}(\mathbf{p}_i)} + \mathbf{L}_{\mathbf{s}(\mathbf{p}_d)})^{+}\Delta\mathbf{s} \tag{2.34}$$

Using these two control schemes, better results can be obtained even for large displacement leading to local minima and retreat problem using basic approaches. However, we exhibited in the next chapter a singularity of control scheme (2.34). Finally, the 3D camera trajectory is also improved.

Hessian approximation

In [Lapresté 04], a method that approximates the control matrix up to second order is proposed. This approximation is achieved by considering Taylor expansions at first and second order, and by defining $\tilde{\mathbf{s}} \in \mathbb{R}^N$, where $N = \frac{n}{2}(n+1)$, as a function of p_i, and finally by introducing supplementary parameters $p_i p_j$ where $0 \leq i \leq j \leq n$ where n is the number of degrees of freedom. The Jacobian of $\tilde{\mathbf{s}}$ as a function of $\mathbf{L}_{\mathbf{s}}$ and $\mathbf{H}_{\mathbf{s}}$ is given by:

$$\mathbf{J}_{\tilde{\mathbf{s}}} = \mathbf{L}_{\mathbf{s}(\mathbf{p})}\Delta\mathbf{p} + \Delta\mathbf{p}^{\top}\mathbf{H}_{\mathbf{s}(\mathbf{p})}\Delta\mathbf{p} \tag{2.35}$$

where $\mathbf{J}^{i}_{\tilde{\mathbf{s}}} = [\mathbf{L}^{i}_{\mathbf{s}(\mathbf{p})}, \tilde{\mathbf{H}}^{i}_{\mathbf{s}(\mathbf{p})}]$ with $\tilde{\mathbf{H}}^{i}_{\mathbf{s}(\mathbf{p})}$ is obtained from the coefficients of the lower triangular part of $\mathbf{H}_{\mathbf{s}_i}(\mathbf{p})$ by dividing its diagonal by 2 and putting these terms in a line vector. Finally, the control law is defined as:

$$\mathbf{v} = \mathbf{K}^{+}\boldsymbol{\Delta}\mathbf{s} \tag{2.36}$$

where $\mathbf{K}^{+}$ is composed of the first n lines of the pseudo inverse of the Jacobian matrix $\mathbf{J}_{\tilde{\mathbf{s}}}$. This method is used in [Lapresté 04] to overcome the retreat and advance problem in IBVS.

2.5.5 Problems in visual servoing

Selecting a suitable set of visual features and designing good control schemes should be taken into account for avoiding system failures. Very usual problems in visual servoing that are directly influenced by this selection and can be enhanced by a good selection are local minima, singularity and visibility problems.

Local minima

Generally, local minima problems occur only with very specific configurations [Chaumette 98] [Gans 03a]. Getting trapped in a local minima, camera velocity is null $\mathbf{v} = \mathbf{0}$, while the feature errors have not been minimized such that $\mathbf{s} - \mathbf{s}^* \neq \phi \in \text{Ker}(\mathbf{L}^+)$. This results in converging to a final pose that is different from the desired one. When $\mathbf{s}$ is composed in three image points and $\mathbf{L_s}$ is full rank, then we have $\text{Ker}(\mathbf{L_s^+}) = \mathbf{0}$, implying that there is no local minima. However, when three points are used, the same image of the three points can be seen from four different camera poses, which means four camera poses exist such that $\mathbf{s} = \mathbf{s}^*$, that correspond to four global minima. When at least four points are used, unique pose can theoretically be obtained. However, $\dim(\mathbf{L_s}) = 8 \times 6$, implying that dim $\text{Ker}(\mathbf{L}^+) = 2$. Using four points, the control law tries to enforces 8 constraints on the image trajectory while the system has only six degrees of freedom. In that case, due to the existence of unrealizable motions in the image that may be computed by the control law, a local minima may be reached [Chaumette 98].

Several control strategies have been used to avoid local minima in visual servoing. For example, and as already said before, in [Deng 05] and [Gans 07], a hybrid motion control strategy that considers the local minima problem is presented while in [Mezouar 02] a path planning strategy is developed.

Singularity

When the interaction matrix is singular causing a task singularity, the velocity tends to infinity and the system is unstable. It may become singular if image points are chosen as visual features. For instance, when four points are used and the required camera motion is defined by a pure rotation of 180^o around its optical axis, the image trajectory obtained of each point is such that the points move concurrently in a straight line at the principal point, where the interaction matrix is singular [Chaumette 98]. For the considered motion, the choice of image points coordinates is really inadequate. Indeed, if the four points in the image are replaced by cylindrical parameters (ρ, θ), the singularity can be avoided when the same initial position is used, showing a pure rotation motion around the optical axis of the camera. Other singular configurations will be described in Chapter 4.

In PBVS [Wilson 96] [Martinet 96] [Deng 03], most of representations avoid the problems of local minima and/or singularities of the corresponding interaction matrices depending on the chosen $\mathbf{e}$, a 3D straight line between the initial and final camera pose is obtained when $\dot{\mathbf{e}}$ is defined by (2.13). This problem can also be solved by using potential function [Hashimoto 00b], partitioning approach [Corke 01], switching approach [Deng 05], hybrid approach [Malis 99], and PBVS [Wilson 96].

Feature visibility

Using classical 2-D and 3-D visual servoing and assuming a bad calibration and a large initial camera displacement, the target may leave the camera field of view [Malis 99] [Chesi 04b]. That is why, it is desirable to have servoing controls able to keep features in the camera field of view to obtain reliable feedback signal during the servoing process. To minimize the probability that the object leaves the FOV, a repulsive potential field can be adopted [Chesi 04b], a path planning strategy [Mezouar 02], switching strategies [Gans 07], as well as using structure light [Xie 09].

2.6 Performance and robustness

A nice dynamic performance of PBVS is presented in [Deng 03]. It considers different orientation formats in controlling the robot end effector in the desired and in the current end-effector frames. In [Kyrki 04a], the effect of vision system properties to the performance of the control system are investigated by analyzing the propagation of image errors through pose estimation and visual servoing control law.

Quantitative performance metrics for specific tasks can be performed using some metric measures as in [Gans 03c]. These measures include the number of iterations required to converge, error at termination, maximum feature excursion, maximum camera excursion and maximum camera rotation. The evaluation and the comparison between different visual servoing approaches can thus be performed for a given task. (But I will not use these matrices in the following chapters).

Recently in [Malis 10], the robustness of standard image-based visual servoing control and efficient second order approximation method (ESM) is studied theoretically with respect to errors on the 3-D parameters introduced in the interaction matrix when any central catadioptric camera is used as a sensor and when points coordinates are used as input of the control scheme. It has been noticed that the stability region is similar for all catadioptric cameras and it is not expected to enlarge it by simply changing the type of central camera used.

In [JanabiSharifi 10a], a comprehensive comparison of IBVS and PBVS is presented by comparing system stability and dynamic performance in the Cartesian and image spaces on a common framework using both predefined and taught references . The robustness and sensitivity analyses are investigated with respect to all the camera, target, and robot modeling errors. Furthermore, other fundamental characteristics of the two methods, such as sensory task space singularity and local minima are also compared.

2.7 Conclusion

In this chapter, we presented a review of the state of the art concerning two main components in visual servoing: features selection and design of the control scheme. This review can be summarized in two graphs as illustrated in Fig. (2.4) for features selection and in Fig. (2.5) for visual servoing control schemes. For a chosen combination of both, different behaviors can be obtained by the robot system. That is why several works have been accomplished concerning the robustness with respect to selected features, structure of the employed control scheme, existence of errors and uncertainty in robot or camera calibration, and errors and uncertainty in input signals and in object model.

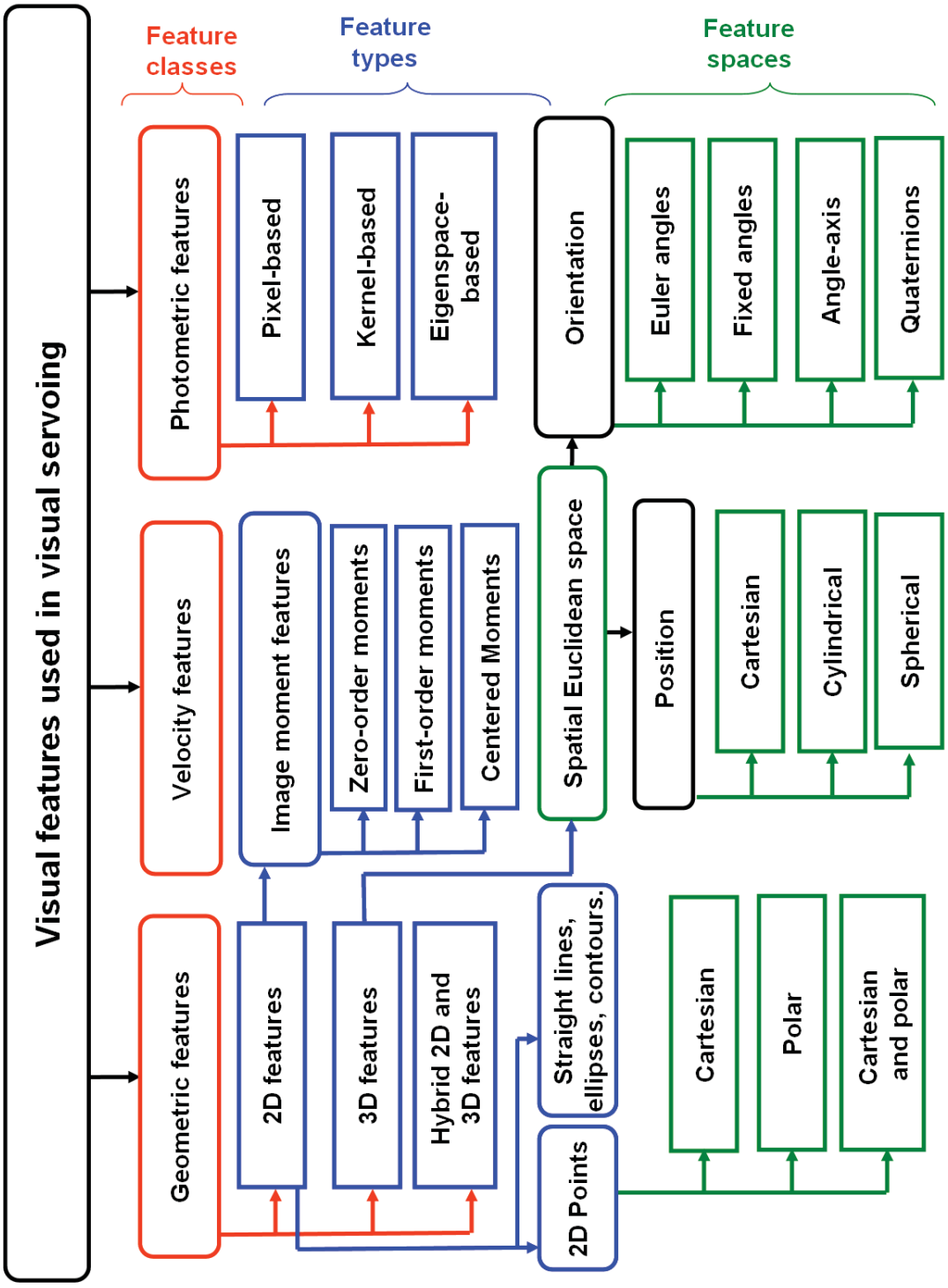


Figure 2.4 – Visual feature selection

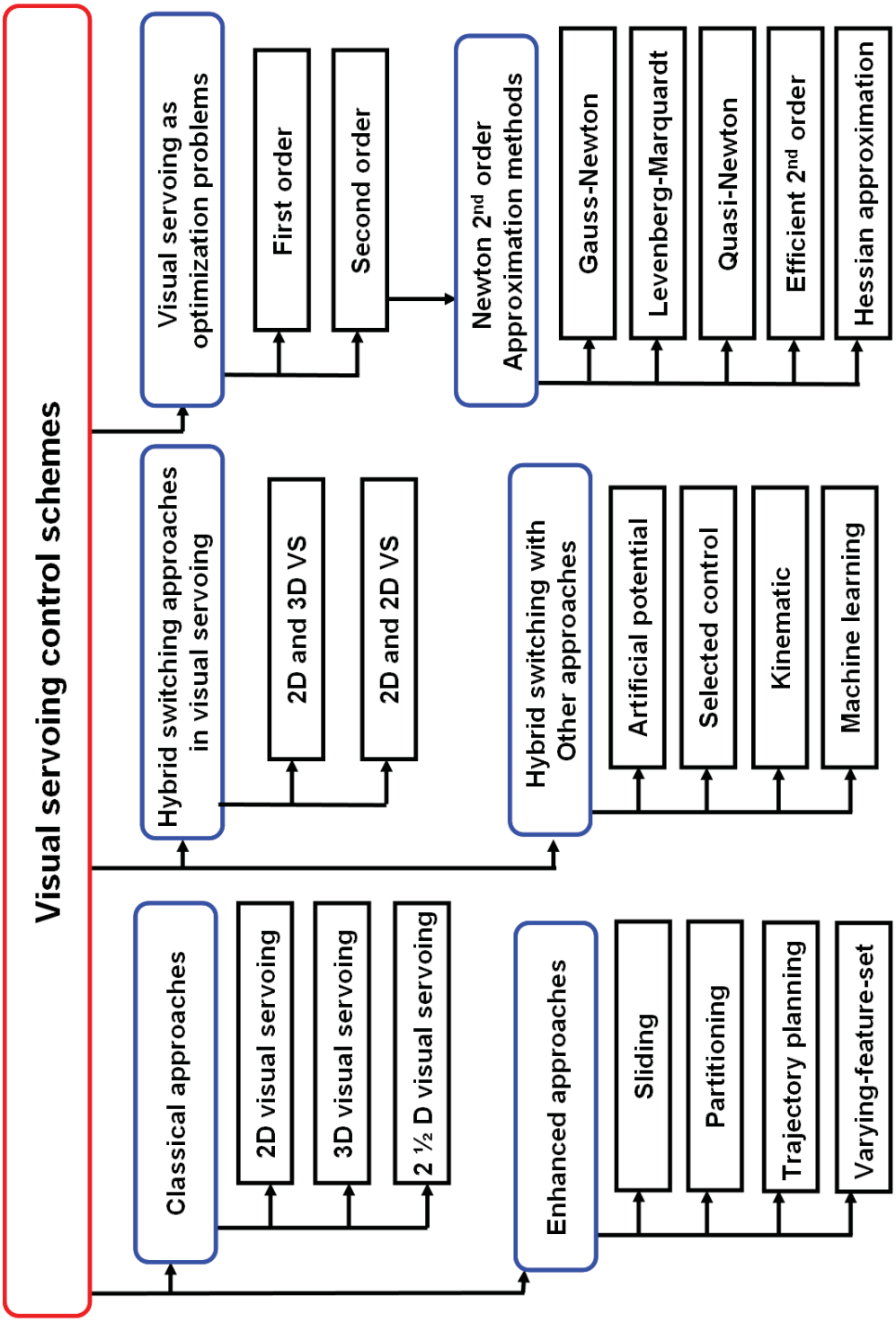


Figure 2.5 – Visual servoing control schemes

Chapter 3

New first order control scheme

In this chapter, we analyze and compare five image-based visual servoing control laws. Three of them are classical while two new ones are proposed. The first new control law is based on a behavior controller to adjust the movement of the camera. It can also be used to switch between the classical methods. The second control law is designed to try to obtain the global stability of the system. An analytical study of all control schemes when translational motion along and rotational motion around the optical axis is also presented. Finally, simulation and experimental results show that the new control law with a behavior controller has a wider range of success than the other control schemes and can be used to avoid local minima and singularities. The work described in this chapter leads to the following publications [Marey 08b][Marey 08a][Marey 08c].

This chapter is organized as follows: In Section 3.1, classical control schemes are recalled. In Section 3.2, the control law with a behavior controller is proposed while in Section 3.3, another control law is proposed and it global stability is studied. In Section 3.4, an analysis of the control laws in the presence of rotation and translation w.r.t. the camera optical axis is presented. Finally, simulation and experimental results are presented in Section 3.5.

3.1 Classical image-based control matrix

Let $\mathbf{s} \in \mathbb{R}^k$ be the vector of the selected k visual features, $\mathbf{s}^*$ their desired value and $\mathbf{v} \in \mathbb{R}^6$ the instantaneous velocity of the camera. Most classical control laws have the following form:

$$\mathbf{v} = -\lambda \, \widehat{\mathbf{L}_\mathbf{s}}^{+} (\mathbf{s} - \mathbf{s}^*) \tag{3.1}$$

where λ is a gain and $\widehat{\mathbf{L}_\mathbf{s}}^{+}$ is the pseudoinverse of an estimation or an approximation of the interaction matrix related to $\mathbf{s}$. Different forms for $\widehat{\mathbf{L}_\mathbf{s}}$ have been proposed in the past [Espiau 92,

Chaumette 06]. For simplicity, we consider that all values can be computed accurately, leading to the following choices

$$1) : \widehat{\mathbf{L_s}} = \mathbf{L}_{\mathbf{s}^*} \quad (3.2)$$

$$2) : \widehat{\mathbf{L_s}} = \mathbf{L}_{\mathbf{s}(t)} \quad (3.3)$$

$$3) : \widehat{\mathbf{L_s}} = (\mathbf{L}_{\mathbf{s}^*} + \mathbf{L}_{\mathbf{s}(t)})/2. \quad (3.4)$$

In the first case, $\widehat{\mathbf{L_s}} = \mathbf{L}_{\mathbf{s}^*}$ is constant during all the servo since it is the value of the interaction matrix computed at the desired configuration. It is thus not required to estimate the depth of each feature at each iteration, the interaction matrix is computed off line either by knowing the 3D model of the object or by computing it from an image acquired at the desired configuration. In the second case, $\widehat{\mathbf{L_s}} = \mathbf{L}_{\mathbf{s}(t)}$ changes at each iteration of the servo since the current value of the interaction matrix is used. The depth in this case can be estimated using camera motion measurements [Chaumette 96] or using depth observer [DeLuca 08b] or from pose estimation. Finally, in the third case, the average of these two values is used [Malis 04], and similar to the second case the depth is needed at each iteration. As explained in [Tahri 10], it is possible to improve the form given in (3.4) by using $\widehat{\mathbf{L_s}} = (\mathbf{L}_{\mathbf{s}^*}\ {}^{c^*}\mathbf{T_c} + \mathbf{L}_{\mathbf{s}(t)})/2$ where ${}^{c^*}\mathbf{T_c}$ is the spatial motion transform matrix to transform velocities expressed in the desired camera frame to the current camera frame. However, we will not consider this supplementary control scheme in the following, since it would be necessary to know the pose between the current and the desired camera frame. The three usual choices (3.2), (3.3) and (3.4) for $\widehat{\mathbf{L_s}}$ when used with (3.1) define three distinct control laws:

$$1) : \mathbf{v} = -\lambda\, \mathbf{L}^+_{\mathbf{s}^*}(\mathbf{s} - \mathbf{s}^*) \quad (3.5)$$

$$2) : \mathbf{v} = -\lambda\, \mathbf{L}^+_{\mathbf{s}(t)}(\mathbf{s} - \mathbf{s}^*) \quad (3.6)$$

$$3) : \mathbf{v} = -2\lambda\, (\mathbf{L}_{\mathbf{s}^*} + \mathbf{L}_{\mathbf{s}(t)})^+(\mathbf{s} - \mathbf{s}^*). \quad (3.7)$$

We will denote these control schemes D, C and A (for desired, current and average respectively) in the remainder of this chapter. On one hand, near the desired pose where the error $\mathbf{s} - \mathbf{s}^*$ is low, the same behavior is obtained whatever the choice of $\widehat{\mathbf{L_s}}$ since we have in that case $\mathbf{L}_{\mathbf{s}(t)} \approx \mathbf{L}_{\mathbf{s}^*}$. On the other hand, as soon as $\mathbf{s} - \mathbf{s}^*$ is large, it is well known that the choice of $\widehat{\mathbf{L_s}}$ induces a particular behavior of the system since we thus have $\mathbf{L}_{\mathbf{s}(t)} \neq \mathbf{L}_{\mathbf{s}^*}$. This motivates the current research on the determination of visual features such that the interaction matrix is constant in all the configuration space of the camera, but it is clearly still an open problem, and it not the subject of this work to determine new visual features.

3.2 New controller with a behavior parameter

In our proposed control law, a hybrid matrix obtained by introducing a behavior controller β is used to partially combine the two interaction matrices $\mathbf{L}_{\mathbf{s}^*}$ and $\mathbf{L}_{\mathbf{s}(t)}$:

$$\widehat{\mathbf{L_s}} = \mathbf{L}_\beta = (\beta \mathbf{L}_{\mathbf{s}^*} + (1 - \beta)\mathbf{L}_{\mathbf{s}(t)}). \quad (3.8)$$

Using (3.8) in (3.1), we obtain a new control law, denoted G in the following (for "general").

$$\mathbf{v} = -\lambda \left(\beta \, \mathbf{L}_{\mathbf{s}^*} + (1-\beta) \, \mathbf{L}_{\mathbf{s}}\right)^{+} (\mathbf{s} - \mathbf{s}^*) \tag{3.9}$$

Of course, if $\beta = 1$, we find again control law D; if $\beta = 0$, we obtain control law C, and if $\beta = 1/2$, we obtain control law A. Control law G could thus be used to switch between the different control schemes during the execution of the task. As described in the previous chapter, switching strategies have already been proposed in [Gans 07] [Hafez 07b] but, in these works, switching is performed between image-based and position-based approaches, that is between different features in order to improve the performance of the visual servoing, while here the features are the same but their control matrix is different.

More generally, the main interesting property of control law G (3.9) is that the behavior of the system changes gradually from the behavior using control law C to the behavior using control law A when β varies from 0 to $1/2$, and similarly, the behavior changes gradually from the behavior using control law A to the behavior using control law D when β varies from $1/2$ to 1. Hence, this new control scheme allows us to adapt the behavior of the system based on the selected value of β. We will see in Section 3.5 that particular values of β indeed allows the system to converge while the other control schemes fail.

Let us finally note that in case of modeling or calibration errors, the matrices $\mathbf{L}_{\mathbf{s}^*}$ and $\mathbf{L}_{\mathbf{s}(t)}$ have to be respectively replaced by approximations $\widehat{\mathbf{L}_{\mathbf{s}^*}}$ and $\widehat{\mathbf{L}_{\mathbf{s}(t)}}$, but that does not change the general properties of the control schemes as long as the approximations are not too coarse.

3.3 Pseudo-Gas control law

Control laws D, C, and A are known to be locally asymptotically stable only [Chaumette 06]. The same is of course true for control law G. In this section, an attempt to obtain a globally asymptotically stable (GAS) control scheme is presented.

3.3.1 Modeling

Let us choose as task function $\mathbf{e} \in \mathbb{R}^{\mathbf{6}}$ the following error

$$\mathbf{e} = \mathbf{L}_{\mathbf{s}^*}^{+} (\mathbf{s} - \mathbf{s}^*) \tag{3.10}$$

where, as usual, $\mathbf{s}^*$ is chosen such that $\mathbf{L}_{\mathbf{s}^*}$ is a full rank matrix. Since $\mathbf{L}_{\mathbf{s}^*}$ is constant, the time variation of $\mathbf{e}$ is given by $\dot{\mathbf{e}} = \mathbf{L}_{\mathbf{e}} \mathbf{v}$ where $\mathbf{L}_{\mathbf{e}} \in \mathbb{R}^{6\times 6}$ is given by

$$\mathbf{L}_{\mathbf{e}} = \mathbf{L}_{\mathbf{s}^*}^{+} \mathbf{L}_{\mathbf{s}}.$$

We can note that $\mathbf{L}_{\mathbf{e}}$ is of full rank 6 as soon as $\mathbf{L}_{\mathbf{s}}$ is also of full rank 6. To achieve an exponential decreasing of $\mathbf{e}$ (that is, $\dot{\mathbf{e}} = -\lambda \mathbf{e}$), we obtain immediately as control scheme

$$\mathbf{v} = -\lambda \mathbf{L}_{\mathbf{e}}^{-1} \mathbf{e}, \tag{3.11}$$

which is nothing but

$$\mathbf{v} = -\lambda(\mathbf{L}_{\mathbf{s}^*}^{+}\mathbf{L}_{\mathbf{s}})^{-1}\mathbf{L}_{\mathbf{s}^*}^{+}(\mathbf{s}-\mathbf{s}^*). \tag{3.12}$$

3.3.2 Stability analysis

To study the stability of the control scheme (3.12), let us consider as candidate Lyapunov function $\mathcal{L} = \frac{1}{2}\|\mathbf{e}(t)\|^2$. We have $\dot{\mathcal{L}} = \mathbf{e}^\top\dot{\mathbf{e}} = \mathbf{e}^\top\mathbf{L}_\mathbf{e}\mathbf{v}$. Applying (3.11), we obtain

$$\begin{aligned}\dot{\mathcal{L}} &= -\lambda\ \mathbf{e}^\top\mathbf{L}_\mathbf{e}\mathbf{L}_\mathbf{e}^{-1}\mathbf{e} = -\lambda\ \mathbf{e}^\top\mathbf{e} \\ &< 0\ ,\ \forall\mathbf{e}\neq 0.\end{aligned}$$

The control scheme (3.12) seems thus to be very promising since it is GAS in the task function space. Indeed, $\mathcal{L}$ always decrease to 0 whatever the initial value of $\mathbf{e}$. Furthermore, $\mathbf{e}$ ensures the specified behavior $\dot{\mathbf{e}} = -\lambda\mathbf{e}$ as soon as $\mathbf{L}_{\mathbf{s}^*}$ and $\mathbf{L}_\mathbf{s}$ are computed accurately. Unfortunately, to end the demonstration of the global stability, we should demonstrate that $\mathbf{e} = 0$ if and only if $\mathbf{s} = \mathbf{s}^*$. That is usually impossible since, as soon as

$$(\mathbf{s}-\mathbf{s}^*) \in \text{Ker}\ \mathbf{L}_{\mathbf{s}^*}^{+}, \tag{3.13}$$

we have $\mathbf{e} = 0$, which implies $\mathbf{v} = 0$, and $\mathbf{s} \neq \mathbf{s}^*$, which corresponds to a local minimum [Chaumette 98]. In other words, GAS in the task function space does not necessarily imply GAS in SE_3 or in the visual features space. The task function (3.10) forms a local diffeomorphism with SE_3, but not a global one as soon as a configuration such that (3.13) is satisfied exists. Control law (3.12) will be denoted PG in the following (for "pseudo-GAS").

In Section 3.5, we will exhibit some configurations which lead to local minima. We can thus conclude that, as for all the previous control schemes, only the local asymptotic stability of PG can be demonstrated when image point coordinates are used as visual features. In spite of this disappointing result, control scheme PG is still interesting, since it may be possible in the future to determine visual features such that Ker $\mathbf{L}_{\mathbf{s}^*}^{+} = \mathbf{0}$, leading to GAS.

3.4 Motion along and around the optical axis

This section presents an analytical analysis of all the control laws described previously when the camera displacement is a combination of a translation t_z and a rotation r_z w.r.t. the camera optical axis. As usually done in IBVS, we have considered an object composed of four points forming a square (see Fig. 3.1). The study includes three cases in which the movement along z-axis is from Z to Z^*. In the first case $r_z = 0^o$, which means no rotation around the camera optical axis, while $r_z = 90^o$ in the second case and $r_z = 180^o$ in the third case. In all cases, the object plane is parallel to the image plane and the desired pose is common. Using four points, the initial visual feature vector $\mathbf{s}$ and the desired one $\mathbf{s}^*$ are:

$$\begin{aligned}\mathbf{s} &= (x_0, x_1, x_2, x_3, y_0, y_1, y_2, y_3) \\ \mathbf{s}^* &= (x_0^*, x_1^*, x_2^*, x_3^*, y_0^*, y_1^*, y_2^*, y_3^*)\end{aligned} \tag{3.14}$$

where $x_i = X_i/Z$, $y_i = Y_i/Z$, $x_i^* = X_i^*/Z^*$ and $y_i^* = Y_i^*/Z^*$ for $i = 1, .., 4$.

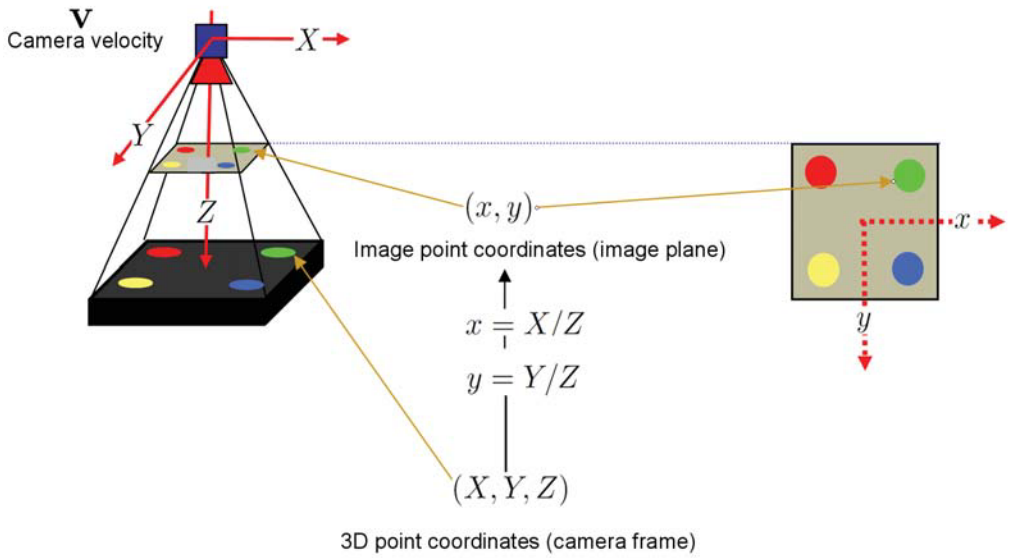


Figure 3.1 – Motions along the camera optical axis

3.4.1 Case 1: $t_z = (Z \rightarrow Z^*)$

We consider in this case a pure translation along the camera optical axis. Let the coordinates of the four points in the camera frame at the initial and the desired configurations be defined by $p_{i0} = (-L, L, Z)$, $p_{i1} = (L, L, Z)$, $p_{i2} = (L, -L, Z)$, $p_{i3} = (-L, -L, Z)$, $p_{d0} = (-L, L, Z^*)$, $p_{d1} = (L, L, Z^*)$, $p_{d2} = (L, -L, Z^*)$ and $p_{d3} = (-L, -L, Z^*)$. Let $l = L/Z$ and $l^* = L/Z^*$. By defining the feature vector as in (3.14), we obtain the initial feature vector $\mathbf{s} = (-l, l, l, -l, l, l, -l, -l)$ and the corresponding desired feature vector $\mathbf{s}^* = (-l^*, l^*, l^*, -l^*, l^*, l^*, -l^*, -l^*)$ as depicted in Fig. 3.2.

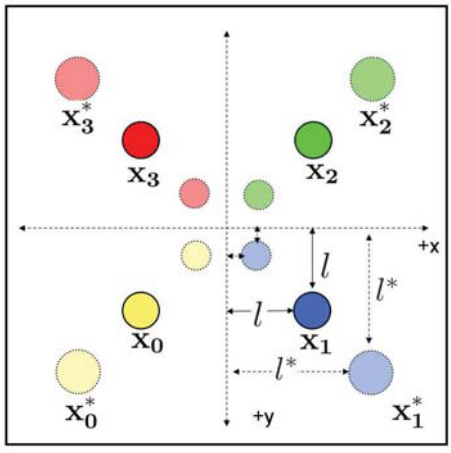


Figure 3.2 – Case 1: Initial and desired image feature points when $t_z = (Z \rightarrow Z^*)$

Using $\mathbf{s}^*$ and the analytical form of $\mathbf{L_x}$, the stacked interaction matrix $\mathbf{L_s}$ computed at the desired configuration for $\mathbf{s}^*$ is given as:

$$\mathbf{L}_{\mathbf{s}^*} = \begin{bmatrix} \frac{-1}{Z^*} & 0 & \frac{-l^*}{Z^*} & -l^{*2} & -(1+l^{*2}) & l^* \\ \frac{-1}{Z^*} & 0 & \frac{l^*}{Z^*} & l^{*2} & -(1+l^{*2}) & l^* \\ \frac{-1}{Z^*} & 0 & \frac{l^*}{Z^*} & -l^{*2} & -(1+l^{*2}) & -l^* \\ \frac{-1}{Z^*} & 0 & \frac{-l^*}{Z^*} & l^{*2} & -(1+l^{*2}) & -l^* \\ 0 & \frac{-1}{Z^*} & \frac{l^*}{Z^*} & 1+l^{*2} & l^{*2} & l^* \\ 0 & \frac{-1}{Z^*} & \frac{l^*}{Z^*} & 1+l^{*2} & -l^{*2} & -l^* \\ 0 & \frac{-1}{Z^*} & \frac{-l^*}{Z^*} & 1+l^{*2} & l^{*2} & -l^* \\ 0 & \frac{-1}{Z^*} & \frac{-l^*}{Z^*} & 1+l^{*2} & -l^{*2} & l^* \end{bmatrix} \tag{3.15}$$

and in this particular case, the interaction matrix $\mathbf{L_s}$ of initial set of visual features is given by:

$$\mathbf{L_s} = \mathbf{L}_{\mathbf{s}^*}|_{(l^*=l, Z^*=Z)} \tag{3.16}$$

It is possible to compute the analytical form of $\mathbf{L}_\beta^+$ by using $Z = l^*Z^*/l$ and injecting (3.16) and (3.15) in (3.8). We obtain:

$$\mathbf{L}_\beta^+ = (\beta\mathbf{L}_{\mathbf{s}^*} + (1-\beta)\mathbf{L_s})^+ = \begin{bmatrix} -c_0 & -c_0 & -c_0 & -c_0 & -c_1 & c_1 & -c_1 & c_1 \\ -c_1 & c_1 & -c_1 & c_1 & -c_0 & -c_0 & -c_0 & -c_0 \\ -c_2 & c_2 & c_2 & -c_2 & c_2 & c_2 & -c_2 & -c_2 \\ -c_3 & c_3 & -c_3 & c_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_3 & -c_3 & c_3 & -c_3 \\ c_4 & c_4 & -c_4 & -c_4 & c_4 & -c_4 & -c_4 & c_4 \\ \end{bmatrix} \tag{3.17}$$

where

$$\begin{aligned} c_0 &= \frac{l^*Z^*}{4(l^*\beta + l(1-\beta))} \\ c_1 &= \frac{l^*Z^*(1+l^{*2}\beta + l^2(1-\beta))}{4(l^*\beta + l(1-\beta))(l^{*2}\beta + l^2(1-\beta))} \\ c_2 &= \frac{l^*Z^*}{8(l^{*2}\beta + l^2(1-\beta))} \\ c_3 &= \frac{1}{4(l^{*2}\beta + l^2(1-\beta))} \\ c_4 &= \frac{1}{8(l^*\beta + l(1-\beta))} \end{aligned}$$

Analyzing (3.17) shows that singularities exist for $\mathbf{L}_\beta$ when $l^*\beta + l(1-\beta) = 0$ or when $\beta l^{*2} + (1-\beta)l^2 = 0$. These situations occur when $\beta = 0$ and $l = 0$ which corresponding to initial pose at infinity. Another singularity is obtained when $\beta = 1$ and $l^* = 0$ corresponding to setting the desired position at infinity. When $\beta = 1/2$ singularity is also obtained if either

$l^* = 0$ or $l = 0$, corresponding to setting the desired or the initial position at infinity similarly to controls C and D respectively. Let us note that none of these configurations are realistic in practice,

The error vector $\mathbf{e}$ to be used in the control scheme is given by:

$$\mathbf{e} = \mathbf{s} - \mathbf{s}^* = (-l + l^*, l - l^*, l - l^*, l - l^*, l - l^*, -l + l^*, -l + l^*) \tag{3.18}$$

If the gain of the control law is λ then the values for the initial instantaneous velocity $\mathbf{v_i}$ of the camera can be deduced by injecting (3.17) and (3.18) in (3.9) to get:

$$\mathbf{v_i} = \left(0, 0, \frac{-\lambda(l-l^*)l^*Z^*}{\beta l^{*2} + (1-\beta)l^2} 0, 0, 0\right) \tag{3.19}$$

As expected, there is only a translational velocity component v_z for all control schemes D, C, M and G. The direction of motion depends on the relative relation between l and l^*. When $Z > Z^*$, which is equivalent to $l < l^*$, a forward motion is performed making the camera move toward the object frame to near its desired position. While, when $Z < Z^*$ the direction of initial motion is to make the camera move backward far from the object. The magnitude of v_z is proportional to the difference between l and l^*. To conclude, there is no particular difference in that case between the control schemes C, D and A, and all provide a satisfactory behavior.

3.4.2 Case 2: $r_z = 90^o$ and $t_z = (Z \rightarrow Z^*)$

By considering the coordinates of the four points w.r.t. the camera frame at the desired pose as in case 1, while the coordinates at the initial pose are now $p_{i0} = (-L, -L, Z)$, $p_{i1} = (-L, L, Z)$, $p_{i2} = (L, L, Z)$ and $p_{i3} = (L, -L, Z)$, the initial value of $\mathbf{s}$ is then $\mathbf{s_i} = (-l, -l, l, l, -l, l, l, -l)$ and $\mathbf{s_i} - \mathbf{s}^* = (-l+l^*, -l-l^*, l^*-l, l+l^*, -l-l^*, l-l^*, l+l^*, l^*-l)$ is the error vector (see Fig. 3.3).
The interaction matrix of the features at the initial configuration is:

$$\mathbf{L_s} = \begin{bmatrix} \frac{-1}{Z} & 0 & \frac{-l}{Z} & l^2 & -(1+b^2) & -l \\ \frac{-1}{Z} & 0 & \frac{-l}{Z} & -l^2 & -(1+l^2) & l \\ \frac{-1}{Z} & 0 & \frac{l}{Z} & l^2 & -(1+l^2) & l \\ \frac{-1}{Z} & 0 & \frac{l}{Z} & -l^2 & -(1+l^2) & -l \\ 0 & \frac{-1}{Z} & \frac{-l}{Z} & 1+l^2 & -l^2 & l \\ 0 & \frac{-1}{Z} & \frac{l}{Z} & 1+l^2 & l^2 & l \\ 0 & \frac{-1}{Z} & \frac{l}{Z} & 1+l^2 & -l^2 & -l \\ 0 & \frac{-1}{Z} & \frac{-l}{Z} & 1+l^2 & l^2 & -l \end{bmatrix} \tag{3.20}$$

Injecting the value of the interaction matrix for the desired features, as given in (3.15), and the interaction matrix (3.20) we find easily the analytical form of $\mathbf{L}_\beta$, from which $\mathbf{L}_\beta^+$ is

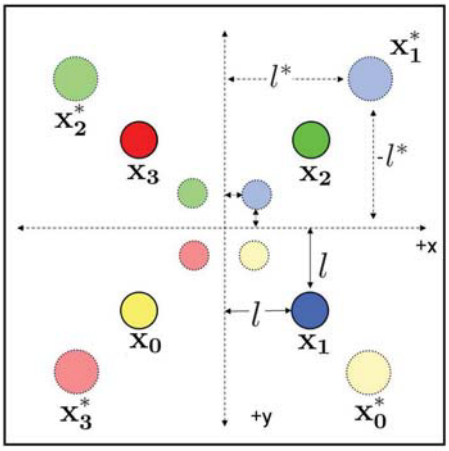


Figure 3.3 – Case 2: Initial and desired image feature points when $r_z = 90^o$ *and* $t_z = (Z \rightarrow Z^*)$

obtained after computations and simplifications:

$$\mathbf{L}_\beta^+ = (\beta \mathbf{L}_{\mathbf{s}^*} + (1-\beta)\mathbf{L}_\mathbf{s})^+ = \begin{bmatrix} -c_0 & -c_0 & -c_0 & -c_0 & -c_1 & c_1 & -c_1 & c_1 \\ -c_1 & c_1 & -c_1 & c_1 & -c_0 & -c_0 & -c_0 & -c_0 \\ -c_3 & c_4 & c_3 & -c_4 & c_4 & c_3 & -c_4 & -c_3 \\ -c_5 & c_5 & -c_5 & c_5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_5 & -c_5 & c_5 & -c_5 \\ c_7 & c_6 & -c_6 & -c_7 & c_6 & -c_7 & -c_6 & c_7 \end{bmatrix} \quad (3.21)$$

where

$$c_0 = \frac{l^* Z^*}{4(\beta l^* + (1-\beta) l)}$$

$$c_1 = \begin{cases} 0 & \text{if } \beta l^{*2} = (1-\beta) l^2 \\ c_0 \frac{\beta(1+l^{*2}) + (1-\beta)(1+l^2)}{(\beta l^{*2} - (1-\beta) l^2)} & \text{else.} \end{cases}$$

$$c_3 = \frac{l^* Z^* (\beta l^* + (1-\beta) l)}{8((1-\beta)^2 l^3 + \beta^2 l^{*3})} \ ,$$

$$c_4 = \frac{l^* Z^* (\beta l^* - (1-\beta) l)}{8((1-\beta)^2 l^3 + \beta^2 l^{*3})}$$

$$c_5 = \begin{cases} 0 & \text{if } \beta l^{*2} = (1-\beta) l^2 \\ \frac{-1}{4(\beta l^{*2} - (1-\beta) l^2)} & \text{else.} \end{cases}$$

$$c_6 = \frac{\beta l^{*2} + (1-\beta) l^2}{8((1-\beta)^2 l^3 + \beta^2 l^{*3})},$$

$$c_7 = \frac{\beta l^{*2} - (1-\beta) l^2}{8((1-\beta)^2 l^3 + \beta^2 l^{*3})}$$

Using the value of $\mathbf{s_i} - \mathbf{s}^*$, the initial velocity $\mathbf{v_i}$ is easily deduced from (3.1) as:

$$\mathbf{v_i} = \left(0, 0, v_z, 0, 0, \omega_z,\right) \tag{3.22}$$

where

$$v_z = \frac{\lambda Z^* l^* (\beta l^{*2} - (1-\beta) l^2)}{\beta^2 l^{*3} + (1-\beta)^2 l^3} \ , \ \omega_z = \frac{\lambda l l^* (\beta l^* + (1-\beta) l)}{\beta^2 l^{*3} + (1-\beta)^2 l^3}$$

As expected, the initial camera motion consists in performing a translation combined with a rotation whose value only depends on image data and on the chosen value for β and λ. We can note that $\mathbf{L}_\beta$ is singular if $\beta l^{*2} = (1-\beta) l^2$. For instance, such a singularity occurs when $l = l^*$ (i.e. $Z = Z^*$) and $\beta = 1/2$, which is very surprising. The control law A proposed in [Malis 04] is thus singular for a pure rotation of 90^o, which had not been exhibited before as far as we know. In fact, the only way to avoid this singularity whatever the value of l and l^* is to select $\beta = 0$ or $\beta = 1$. As can be seen on (3.22), this singularity has no effect on the computed velocity in perfect conditions, but, as we will see in Section 3.5, a quite unstable behavior is obtained in the presence of image noise or for configurations near that singularity (such that for instance the object plane is almost parallel to the image plane).

When $Z = Z^*$ then $l = l^*$ and the initial velocity $\mathbf{v_i}$ becomes

$$\mathbf{v_i} = \left(0, 0, \tfrac{\lambda Z^* (2\beta - 1)}{2\beta^2 - 2\beta + 1}, 0, 0, \tfrac{\lambda}{2\beta^2 - 2\beta + 1}\right).$$

In that classical case, the velocity $\mathbf{v_i}$ contains an unexpected translation whose direction depends on the value of β ($v_z < 0$ if $\beta < 1/2$ and $v_z > 0$ if $\beta > 1/2$). The only way to avoid this nonzero translation is to select $\beta = 1/2$ as already shown in [Malis 04], but $\mathbf{L}_\beta$ is singular in that case...

Coming back to the more general case and setting $\beta = 1$ in $\mathbf{L}_\beta^+$, the initial velocity $\mathbf{v_i}$ using control law D is given by

$$\mathbf{v_i} = \left(0, 0, \lambda Z^*, 0, 0, \tfrac{\lambda l}{l^*}\right). \tag{3.23}$$

Whatever the value of Z, that is even when $Z < Z^*$ in which case the camera has to move backward, the initial camera motion contains a forward translational term. This surprising result extends the same property obtained when $Z = Z^*$ [Chaumette 06].

Setting $\beta = 0$, the initial velocity $\mathbf{v_i}$ using the control law C is now

$$\mathbf{v_i} = \left(0, 0, \tfrac{-\lambda l^* Z^*}{l}, 0, 0, \tfrac{\lambda l^*}{l}\right). \tag{3.24}$$

In that case, the initial camera motion contains a backward translational term whatever the value of Z, that is even when $Z \geq Z^*$. We can even note that, more l is small, *i.e.* more Z is large, more the initial backward motion is large, which is even more surprising than the result obtained for $\beta = 1$. These results extend thus largely the property exhibited in [Corke 01]

when $Z = Z^*$. By comparing (3.23) and (3.24), we can also note that the amplitude of the rotational motion using control laws D and C is surprisingly not the same as long as $l \neq l^*$, that is as soon as $Z \neq Z^*$.

Setting $\beta = 1/2$, the velocity $\mathbf{v_i}$ using control law A is

$$\mathbf{v_i} = \left(0, 0, \frac{2\lambda Z^* l^* (l^{*2} - l^2)}{l^{*3} + l^3}, 0, 0, \frac{2\lambda l l^* (l + l^*)}{l^3 + l^{*3}}\right).$$

In that case, a good behavior is obtained since the translational motion is always in the expected direction ($v_z < 0$ when $l^* < l$, that is when $Z < Z^*$, $v_z > 0$ when $l^* > l$ ($Z > Z^*$), and, as already said, $v_z = 0$ when $l = l^*$ (where $Z = Z^*$ but where $\mathbf{L}_\beta$ is singular).

Finally, the velocity $\mathbf{v_i}$ of the new control law PG is

$$\mathbf{v_i} = \left(0, 0, \frac{-\lambda l^* Z^*}{l}, 0, 0, \frac{\lambda l^*}{l}\right)$$

that is exactly the same velocity $\mathbf{v_i}$ given in (3.24) by the control law C.

3.4.3 Case 3: $r_z = 180^o$ & $t_z = (Z \rightarrow Z^*)$

We now consider the more problematic case where the camera displacement is composed of a translation and of a rotation of 180^o around the camera optical axis (see Fig 3.4). The initial coordinates of the four points in the camera frame at the initial pose are $p_{i0} = (L, L, Z)$, $p_{i1} = (-L, L, Z)$, $p_{i2} = (-L, -L, Z)$ and $p_{i3} = (L, -L, Z)$.

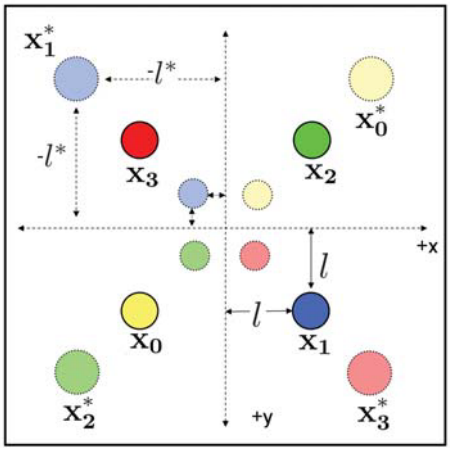


Figure 3.4 – Case 3: Initial and desired image feature points when $r_z = 180^o$ and $t_z = (Z \rightarrow Z^*)$

In that case, $\mathbf{s_i} = (l, -l, -l, l, -l, -l, l, l)$ and the corresponding interaction matrix is:

$$\mathbf{L_s} = \begin{bmatrix} \frac{-1}{Z^*} & 0 & \frac{l}{Z^*} & -l^2 & -(1+l^2) & -l \\ \frac{-1}{Z^*} & 0 & \frac{-l}{Z^*} & l^2 & -(1+l^2) & -l \\ \frac{-1}{Z^*} & 0 & \frac{-l}{Z^*} & -l^2 & -(1+l^2) & l \\ \frac{-1}{Z^*} & 0 & \frac{l}{Z^*} & l^2 & -(1+l^2) & l \\ 0 & \frac{-1}{Z^*} & \frac{l}{Z^*} & 1+l^2 & l^2 & -l \\ 0 & \frac{-1}{Z^*} & \frac{l}{Z^*} & 1+l^2 & -l^2 & l \\ 0 & \frac{-1}{Z^*} & \frac{-l}{Z^*} & 1+l^2 & l^2 & l \\ 0 & \frac{-1}{Z^*} & \frac{-l}{Z^*} & 1+l^2 & -l^2 & -l \end{bmatrix} \tag{3.25}$$

The control matrix $\mathbf{L}_\beta^+$ is now given by:

$$\mathbf{L}_\beta^+ = \begin{bmatrix} -c_0 & -c_0 & -c_0 & -c_0 & -c_1 & c_1 & -c_1 & c_1 \\ -c_1 & c_1 & -c_1 & c_1 & -c_0 & -c_0 & -c_0 & -c_0 \\ -c_3 & c_3 & c_3 & -c_3 & c_3 & c_3 & -c_3 & -c_3 \\ -c_4 & c_4 & -c_4 & c_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_4 & -c_4 & c_4 & -c_4 \\ c_5 & c_5 & -c_5 & -c_5 & c_5 & -c_5 & -c_5 & c_5 \end{bmatrix} \tag{3.26}$$

where

$$c_0 = \frac{l^* Z^*}{4(\beta l^* + (1-\beta) l)} \ , \ c_1 = c_0 \frac{\beta(1+l^{*2}) + (1-\beta)(1+l^2)}{\beta l^{*2} + (1-\beta) l^2}$$

$$c_3 = \begin{cases} 0 & \text{if } \beta l^{*2} = (1-\beta) l^2 \\ \frac{l^* Z^*}{8(\beta l^{*2} - (1-\beta) l^2)} & \text{else} \end{cases}$$

$$c_4 = \frac{1}{4(\beta l^{*2} + (1-\beta) l^2)}$$

$$c_5 = \begin{cases} 0 & \text{if } \beta l^* = (1-\beta) l \\ \frac{1}{8(\beta l^* - (1-\beta) l)} & \text{else} \end{cases}$$

Proceeding as before, using $\mathbf{s_i} - \mathbf{s}^* = (l+l^*, -l-l^*, -l-l^*, l+l^*, -l-l^*, -l-l^*, l+l^*, l+l^*)$ we obtain

$$\mathbf{v_i} = \big(0, 0, v_z, 0, 0, 0,\big)$$

$$\text{where } v_z = \begin{cases} 0 & \text{if } \beta l^{*2} = (1-\beta) l^2 \\ \frac{\lambda Z^* l^* (l+l^*)}{\beta l^{*2} - (1-\beta) l^2} & \text{else.} \end{cases}$$

In all cases, no rotational motion is produced while a translational motion is generally obtained, but when $\beta l^{*2} = (1-\beta) l^2$ in which case $\mathbf{L}_\beta$ is singular, leading to a repulsive local minimum where $v_z = 0$. Such a case occurs for instance when $Z = Z^*$ (i.e. $l = l^*$) and $\beta = 1/2$, which corresponds to the control law proposed in [Malis 04]. Another singularity occurs when $\beta l^* = (1-\beta) l$, which is also the case when $l = l^*$ and $\beta = 1/2$.

Of course, when $Z = Z^*$, we find again the results given in [Chaumette 98]: a pure forward motion is involved when $\beta = 1$ and a pure backward motion is involved when $\beta = 0$. More generally, for $\beta = 1$ and $\beta = 0$, the direction of motion is the same (i.e. forward or backward) whatever the value of l and l^*, that is whatever the value of Z with respect to Z^*.

For any other value of β, the direction of motion depends on the relative value of Z with respect to Z^*, but unfortunately, there does not exist any value of β that will give a good behavior in that case since no rotational motion is computed by the control law. Finally, no better results are obtained using control law PG since we have in that case

$$\mathbf{v_i} = \left(0, 0, \frac{-\lambda l^* Z^* (l+l^*)}{l^2}, 0, 0, 0\right)$$

which is the same as the one obtained when $\beta = 0$.
We will validate the results obtained in this section through experimental results presented at the end of the next section.

3.5 Results

In this section, simulation and experimental results are given. They have been obtained using the ViSP library [Marchand 05b] in which the new control schemes have been implemented.

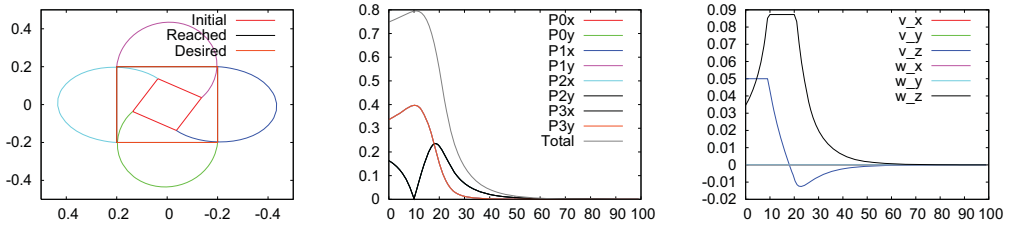
3.5.1 Simulation results: Motion along and around camera optical axis

In this section, the behavior of the camera movement w.r.t. the camera optical axis is studied. A pose is denoted as $\mathbf{p} = (\mathbf{t}, \mathbf{r})$ where $\mathbf{t}$ is the translation expressed in meter and $\mathbf{r}$ the roll, pitch and yaw angles expressed in degrees. The initial camera pose is $(0, 0, 1, 0, 0, r_z)$ and the desired camera pose is $(0, 0, 0.5, 0, 0, 0)$. As in the previous section, we have considered a square and we have set $L = 0.1$ so that $a = 0.2$ and $b = 0.1$. We have also set $\lambda = 0.1$.

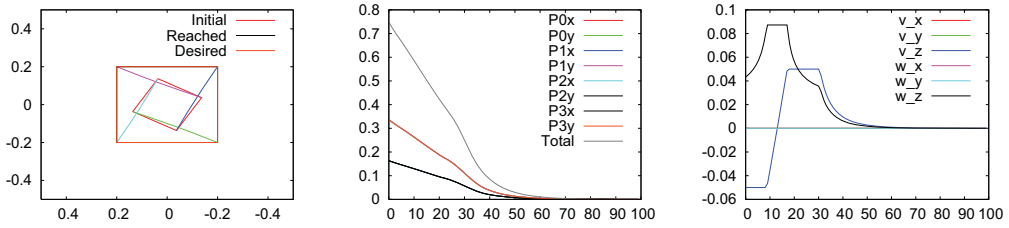
It is important here to mention that the simulation results are not always physically coherent because the simulation software can accept a very high velocity which can not exist in reality to be used with most of robotic systems. So, it is necessary to saturate the very high velocities resulted from the simulation by specifying translational and rotational saturation terms T_{max} and R_{max}. We consider that the real robot acts normally when the velocity sent to the robot controller is given by $T_{max} = 0.05$ m/s and $R_{max} = 5$ deg/s. All velocity components are thus normalized when needed so that the maximal one is 0.05 m/s or 5 deg/s. Implementing these saturation values forbids the application of too high and thus unfeasible velocities and causes a more realistic behavior of the control law when the system is near to a critical situation. Such a situation may occur when $\widehat{\mathbf{L_s}}$ is ill conditioned, that is near a singularity.

A general description of the camera behavior when employing the five control laws with all possible values of r_z is given in the following, as well as the results for three cases: case A when $r_z = 120^o$, case B when $r_z = 170^o$ and case C when $r_z = 180^o$. The results of the three cases are depicted in Figs. 3.5, 3.6 and 3.7 respectively.

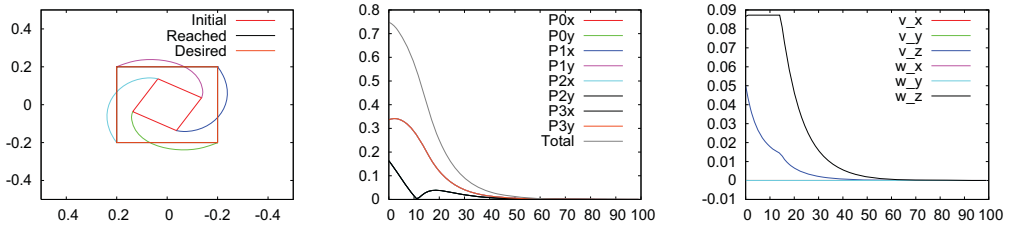
Applying control law D, the camera translates toward the desired pose without any additional movement as soon as $r_z \leq 78^o$. When $r_z > 78^o$, the camera continues its translational



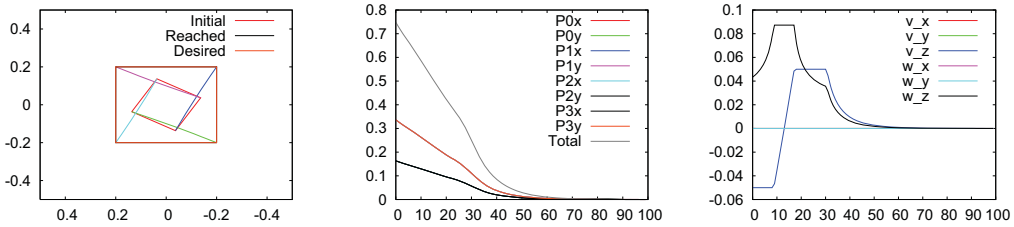
(a) control law D



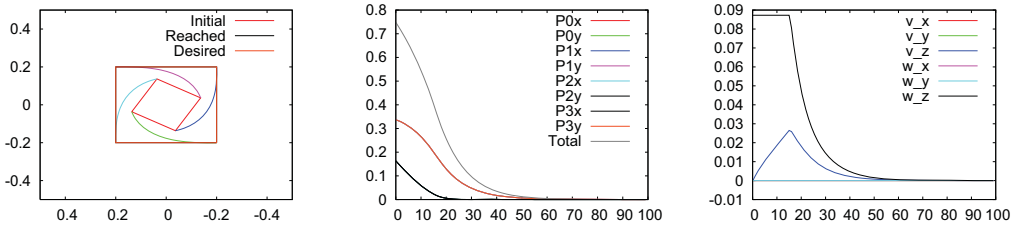
(b) control law C



(c) control law A

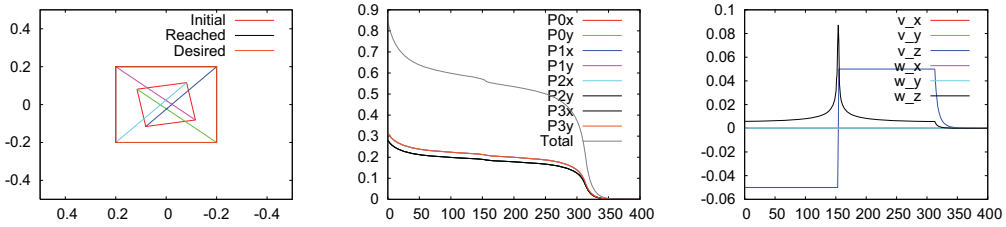


(d) control law PG)

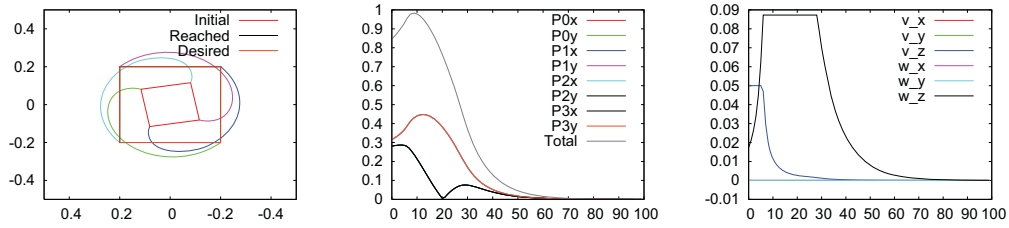


(e) control law G, $\beta = 0.285$

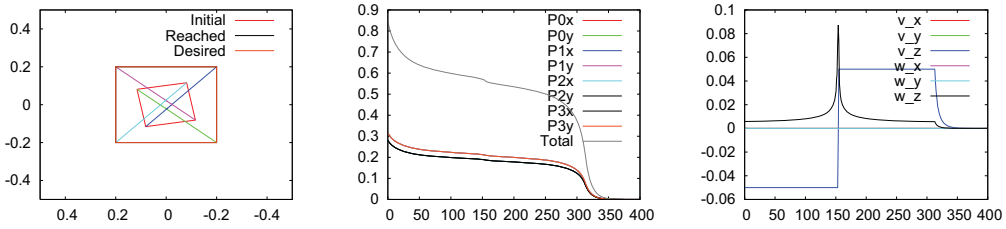
Figure 3.5 – Results for case A: Desired pose is $(0, 0, 0.5, 0, 0, 0)$ *and initial pose is* $(0, 0, 1, 0, 0, 120)$*. First column: image points trajectories, middle column: camera velocity components (in m/s and rad/s), last column: visual features error components and global error.*



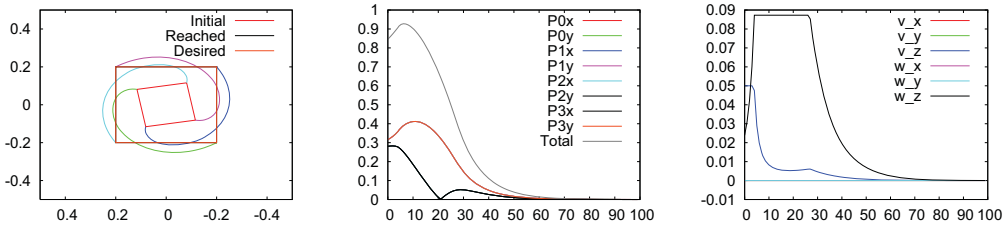
(a) control law C



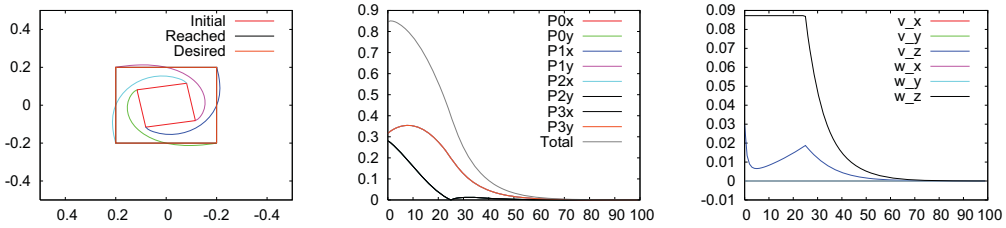
(b) control law A



(c) control law PG



(d) control law G, $\beta = 0.4$



(e) control law G, $\beta = 0.35$

Figure 3.6 – Results for case B: Desired pose is $(0, 0, 0.5, 0, 0, 0)$ *and initial pose is* $(0, 0, 1, 0, 0, 170)$*. First column: image points trajectories, middle column: camera velocity components (in m/s and rad/s), last column: visual features error components and global error.*

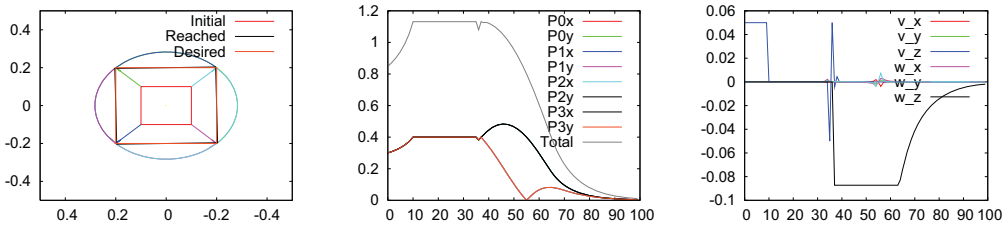


Figure 3.7 – Results for case C: Desired pose is $(0, 0, 0.5, 0, 0, 0)$ *and initial pose is* $(0, 0, 1, 0, 0, 180)$*; control laws (A and G),* $\beta = 0.5$

motion after reaching $Z = Z^*$ and then moves back toward the desired pose as depicted in Fig. 3.5(a). The translation increases as r_z increases. When $r_z \geq 155^o$, the control law fails since the camera reaches the object plane where $Z = 0$. Finally, as can be seen on Figs. 3.5(a), v_z reaches its maximal value at the first iteration and ω_z after some iterations of the control scheme. Both are saturated during few iterations.

Applying control laws C and PG which have exactly the same behavior in the situations considered here, the camera translates correctly as long as $r_z \leq 61^o$. When $r_z > 61^o$, the camera starts moving backward with maximum v_z and then translates forward. The backward translation increases as r_z increases. Maximum r_z is reached and saturated when the camera changes its translation from backward to forward as depicted in Fig. 3.5(b) and 3.5(d). The number of iterations required to reach the desired pose increases rapidly when $r_z > 150$. Finally, when $r_z \geq 178.6^o$, the backward translation is so large that the camera is not able to reach the desired pose.

Applying control law A, there is no additional translation as long as $r_z < 172^o$. This can be checked on Figs. 3.5(c). As discussed before, control law A has a singularity when $r_z = 180^o$, that is why the velocity components are saturated at the beginning of the servo for large values of r_z (see Fig. 3.6(b)). Applying the saturation allows control law A to converge with a perfect behavior as long as $r_z < 180^o$. Indeed as expected from our analytical demonstration in the previous section, when $r_z = 180^o$, only a translational velocity component $v_z \neq 0$ is involved, so the camera performs a forward translation until $Z = Z^*$ where $a = b$. The analytical condition for $v_z = 0$ is then satisfied since we have $\beta = 1/2$. That is why v_z is damped suddenly to zero as illustrated in Fig. 3.7. Finally, the movement of the camera after iteration number 34 is due to the accumulated numerical noise during the previous iterations.

Applying the new control law G, different behaviors are obtained based on the value selected for β. When the value of β is near to 0, 1 and $1/2$, the behavior of the control law approaches the behavior of control laws D, C and A respectively. Best selection of β leads to enhance the behavior of the control law for a given displacement. For example, in case A where $r_z = 120^o$, control law G allows the camera to reach its desired pose when $\beta \in [-0.08, 1.19]$ with the best behavior obtained when $\beta = 0.285$ as depicted in Fig. 3.5(e). In that case, the

rotational velocity ω_z is saturated after reaching its maximum value at the first iteration. The error on each point coordinates starts also to decrease at the first iteration. When $r_z = 170^o$, the camera reaches its desired pose as long as $\beta \in [0.33, 0.85]$ with best behavior obtained when $\beta = 0.4$. Nice results are also obtained with $\beta = 0.35$.

3.5.2 Experimental results: Singularities

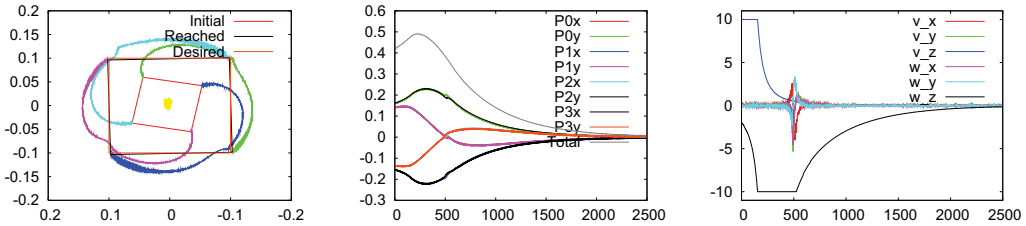
The experimental results have been obtained on a six degrees of freedom robot (see Fig. 3.8). They allow to validate the analysis presented in Section 3.4 about the motion along and around the optical axis.

Figure 3.8 – Experimental system: Cartesian robot

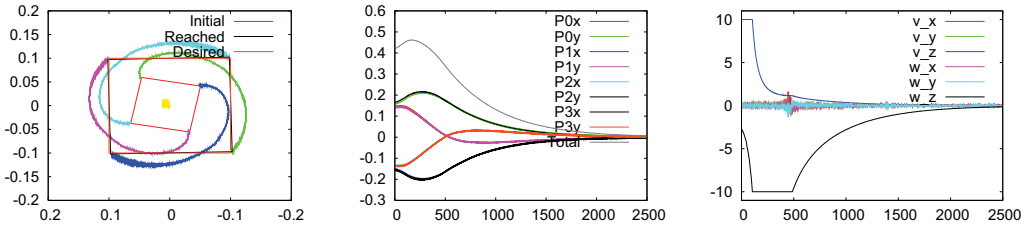
3.5.2.1 Case E1

In the first experimental case, the required camera motion is composed of a rotation of 170^o around the optical axis combined with a translation of 0.5 m along the optical axis toward the object (a square once again). As usual, gain λ has been set to 0.1.

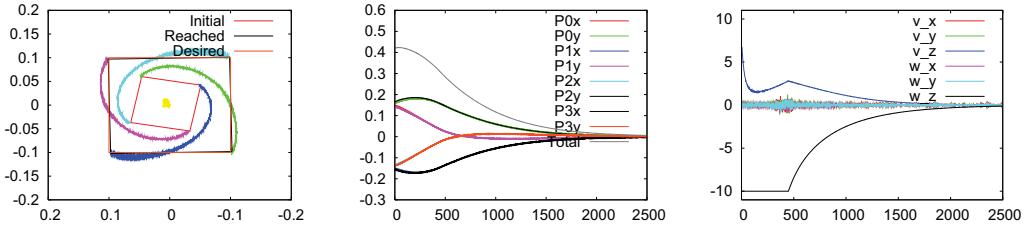
As expected unfortunately, control law D makes the points leave the camera field of view due to a forward motion, while control laws C and PG make the robot reach its joints limits due to a backward motion. As can be seen in Fig. 3.9.a, control law A starts with high value of v_z toward the object, while ω_z increases until the translational motion is almost finished. Since the pure rotation $r_z = 90^o$ corresponds to a singularity of control law A, as demonstrated in the analytical study, the behavior of the camera is quite unstable near this configuration, that is during 400 iterations (from iterations 800 to 1200) as can be observed in the velocity components in Fig. 3.9.a. As can be seen in Fig. 3.9.b, using control law G with $\beta = 0.4$ allows to decrease significantly the effect of the singularity near $r_z = 90^o$, while its effect completely disappears for $\beta = 0.35$ (see Fig. 3.9.c).



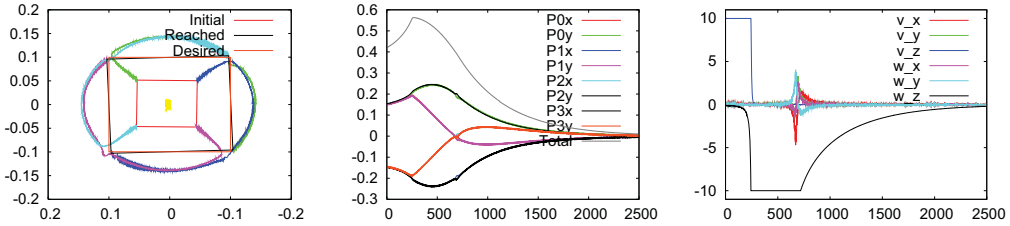
(a) control law A



(b) control law G, $\beta = 0.45$



(c) control law G, $\beta = 0.35$



(d) control laws A and G, $\beta = 0.5$

Figure 3.9 – Experimental results for cases E1 and E2: $r_z = 170^o$ in (a), (b) and (c); for case B: $r_z = 180^o$ in (d). First column: image points trajectories, middle column: camera velocity components (in m/s and rad/s), last column: visual features error components and global error.

3.5.2.2 Case E2

In this second case, the task is still to perform a translation of 0.5 m toward the object but combined now with a rotation of 180^o. Figure 3.9.d shows the results obtained for control law A (that is G with $\beta = 0.5$). The velocity components show that the motion of the camera starts with a pure translation toward Z^*. From the analytical study, no rotational motion should occur. However, due to small image noise and to the use of a real robot, that is a non perfectly calibrated system, the robot moves away from the repulsive local minimum and

starts to rotate. The effect of the singularity at 90^o is clearly visible, but after its crossing, the system converges to the desired pose.

3.5.3 Simulation results: Local minima

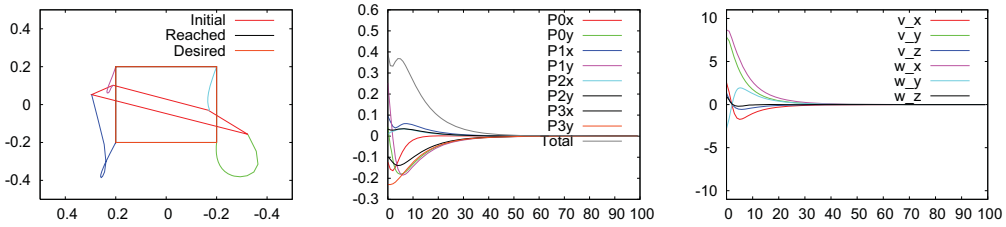
First, we consider two difficult configurations and compare the results obtained with the different control schemes described previously.

3.5.3.1 Case 1

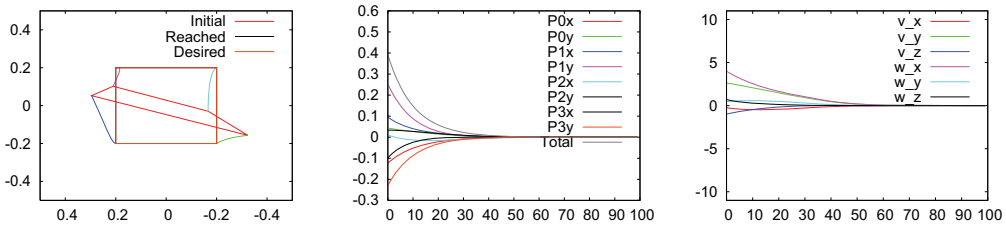
The desired camera pose is $(0, 0, 0.5, 0, 0, 0)$, which means that the camera has to be at 0.5 m in front of the square and such that the square appears as a centered square in the image. In this case, the configurations where $(\mathbf{s} - \mathbf{s}^*) \in \text{Ker } \mathbf{L}_{\mathbf{s}^*}^+$ correspond to very particular cases where the four points are aligned in the image [Chaumette 98]. The initial camera pose is $(0, 0, 0.4, 80, 20, 10)$ and has thus a very different orientation than the desired one. The simulation results for the control laws D, C, A and PG are depicted on Fig. 3.10. Classical schemes D, C and A lead the camera to converge to its desired pose while, using the new control law PG, the camera reaches a configuration where the four points are aligned in the image. In fact, such local minima are attractive for PG while they are not for all other control schemes. As expected, the task function $\mathbf{e}$ defined in (3.10) converges exponentially to zero as shown in Fig. 3.12.a, but that is not sufficient to obtain a good behavior of the system... Finally, we have checked with additional simulations that control law G converges to the desired configuration for any value of $\beta \in [-1.9; 1.04]$ (see Fig. 3.10.e where the result for $\beta = 0.4$ is given). It is thus not necessary that $\beta \in [0; 1]$ and negative values can even be chosen. For this configuration, the value of β is thus not a crucial issue.

3.5.3.2 Case 2

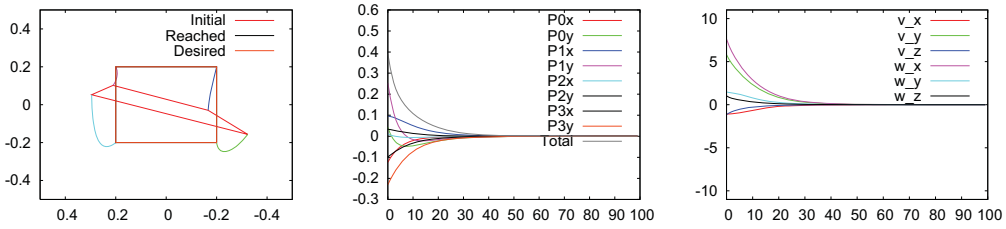
The desired camera pose is now given by $(0, 0, 1, 45, -30, 30)$ which means that the desired position of the image plane is not parallel to the object. The initial camera pose is given by $(0, 0, 1, -46, 30, 30)$. In that case where the desired position of the image plane is not parallel to the object, the control law D is known to be subject to local minima. As can be seen on Fig. 3.11.a, using control law D, the camera is first motionless, as in a local minimum, and then starts to diverge so that the points leave the camera field of view. Even if we do not consider this constraint (we are here in simulation where an image plane of infinite size can be assumed), the camera then reaches the object plane where $Z = 0$, leading of course to a failure. From the results depicted in Fig. 3.11.c, 3.11.d, and 3.11.e, we can see that control laws C, A, and PG all fail in a local minimum. For PG, we can note once again that the task function $\mathbf{e}$ converges exponentially to zero as shown in Fig. 3.12.b. As for control law A, it is the first time, as far as we know, that such a local minimum problem has been exhibited. Finally, control law G with a behavior controller is the only one to converge to the desired position as soon as $0.515 < \beta < 0.569$ (see Fig. 3.11.e). The oscillations observed in the camera velocity and in the visual features allow the camera to go out from the workspace corresponding to the attractive area of the local minimum for the other control schemes.



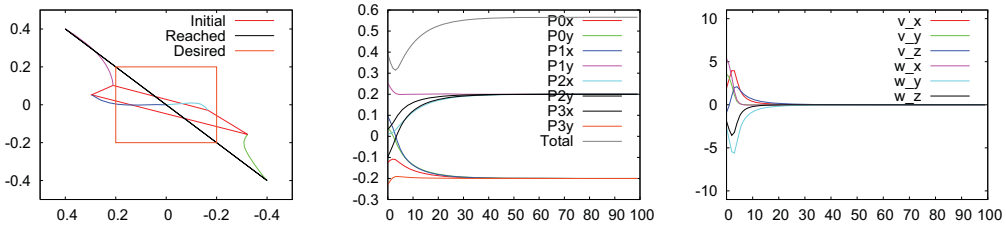
(a) control law D



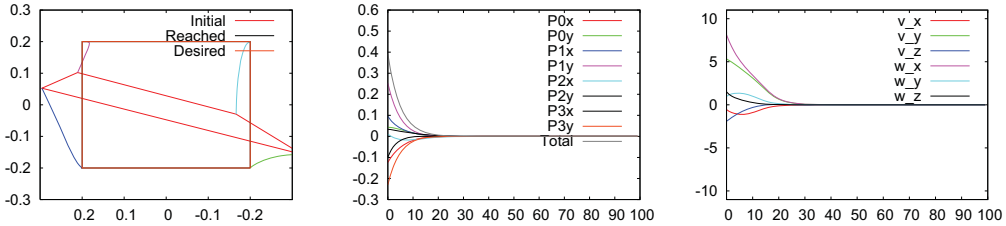
(b) control law C



(c) control law A

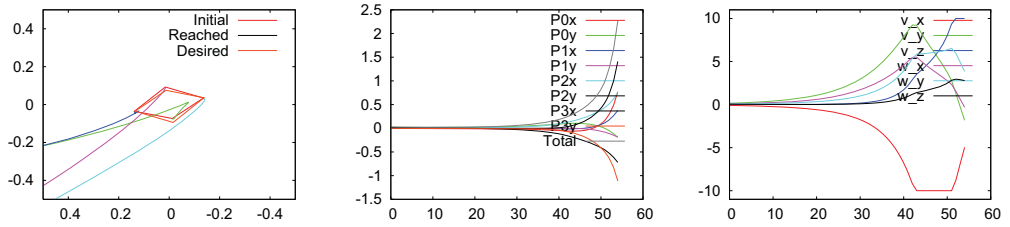


(d) control law PG

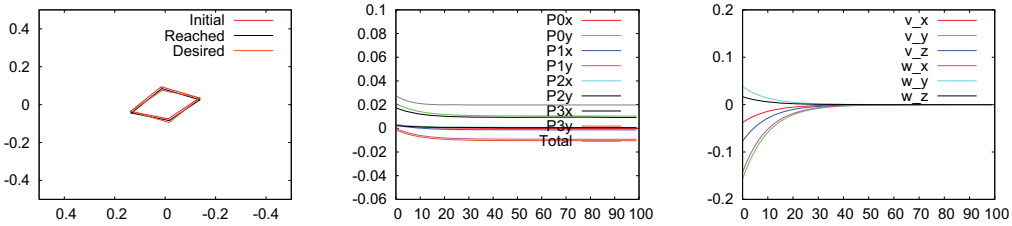


(e) control law G, $\beta = 0.4$

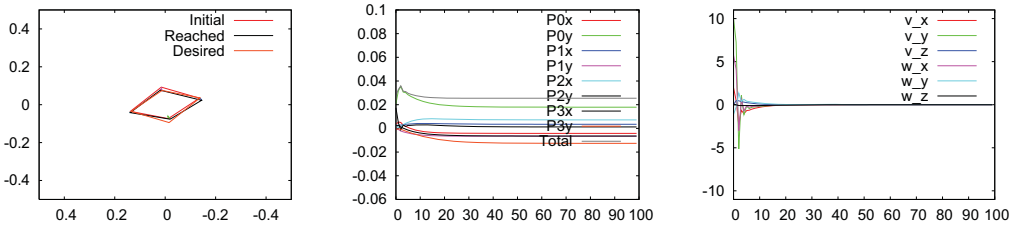
Figure 3.10 – Results for case 1. Desired pose is $(0, 0, 0.5, 0, 0, 0)$ *and initial pose is* $(0, 0, 0.4, 80, 20, 10)$*. First column: image points trajectories, middle column: camera velocity components (in m/s and rad/s), last column: visual features error components and global error.*



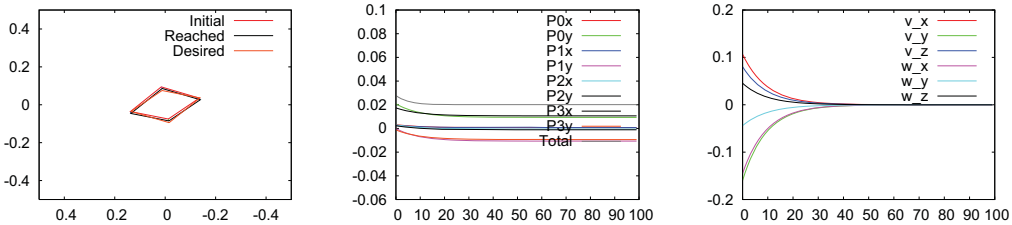
(a) control law D



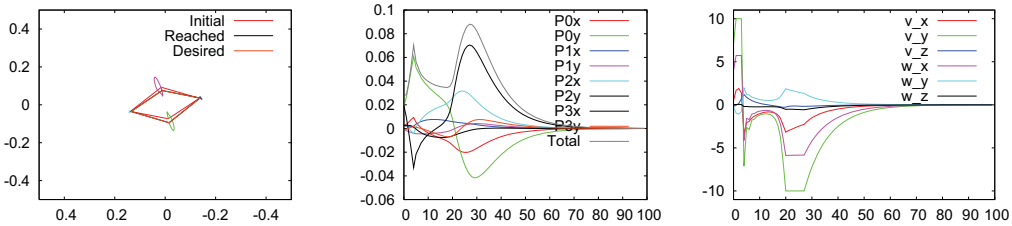
(b) control law C



(c) control law A



(d) control law PG



(e) control law G, $\beta = 0.54$

Figure 3.11 – Results for case 2. Desired pose is $(0, 0, 1, 45, -30, 30)$ *and initial pose is* $(0, 0, 1, -46, 30, 30)$*. First column: image points trajectories, middle column: camera velocity components (in m/s and rad/s), last column: visual features error components and global error.*

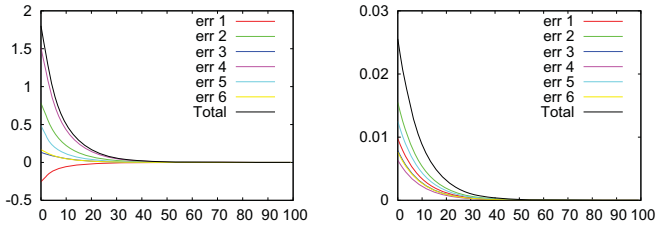
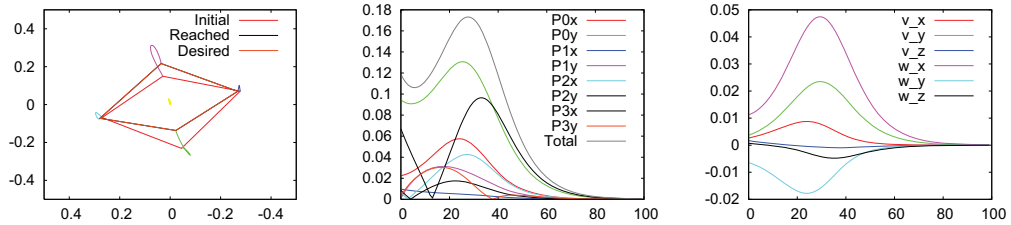


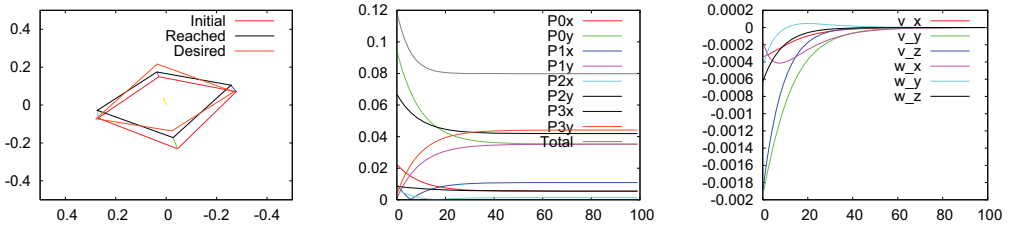
Figure 3.12 – Task function $\mathbf{e}$ *for control law PG for case 1 (on the left) and case 2 (on the right)*

3.5.3.3 Case 3

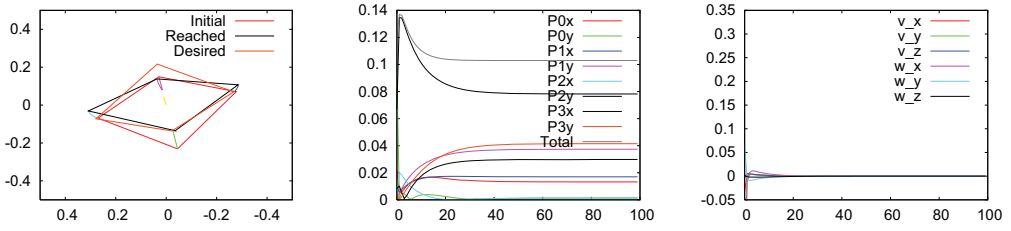
The desired camera pose is now given by $(0, 0, 0.5, -45, 30, 30)$. The initial camera pose is given by $(0, 0, 0.5, 42, -27, 30)$. Similar to case 2, the desired position of the image plane is not parallel to the object and the control law C is subject to local minima. The results are depicted on Fig. 3.13. We can see that all control schemes fail and reach a local minimum, but the control law D, which is indeed known to be less subject to this problem than C [Chaumette 98]. As for A, it is the second time, as far as we know, that such a local minimum problem is exhibited, (case 2 is the first time). For new control law PG, we can note once again that the task function $\mathbf{e}$ converges exponentially to zero as shown in Fig. 3.12. Finally, control law G converges to the desired position as soon as $0.75 < \beta < 1.7$, which is coherent to the fact that control law D (where $\beta = 1$) converges while control laws C and A (where $\beta = 0$ and 1/2) fail.



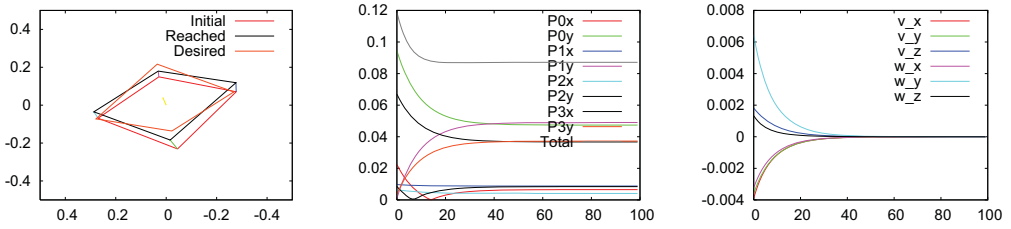
(a) control laws D and G, $\beta = 1$



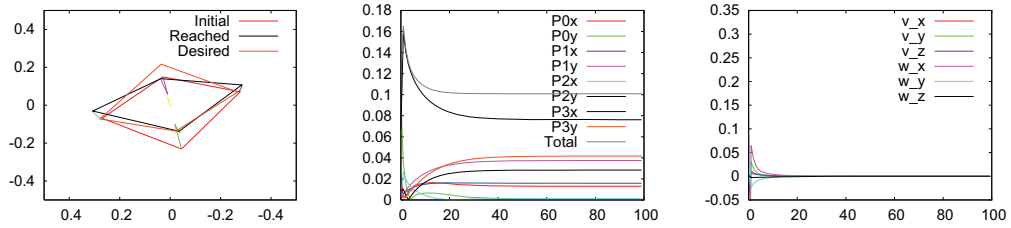
(b) control law C



(c) control law A



(d) control law PG



(e) control law G with $\beta = 0.49$

Figure 3.13 – Results for case 3. Desired pose is $(0, 0, 0.5, -45, 30, 30)$ *and initial pose is* $(0, 0, 0.5, 42, -27, 30)$*. First column: image points trajectories, middle column: camera velocity components (in m/s and rad/s), last column: visual features error components and global error.*

3.6 Conclusion

In this chapter, new configurations have been exhibited, for the first time as far as we know: a local minimum for all classical control schemes, especially for the control law proposed in [Malis 04]. This configuration has been found by studying a new control scheme built to try to obtain its global asymptotic stability.

A singularity of the control scheme proposed in [Malis 04] has also been exhibited and its effects have been emphasized through experiments obtained on a 6 DOF robot. New surprising results have also been obtained for the other classical control schemes for motion combining translation along and rotation around the optical axis. Finally, a new control law based on a behavior controller has also been proposed. Setting $\beta = 0, 1,$ or $1/2$ would allow to switch between the three most classical schemes but we have preferred to analyse the behavior of the control scheme for all possible values of this parameter. In all considered cases (difficult configurations subject to local minima for all classical schemes, motion along and around the optical axis), it has always been possible to determine values of this parameter that provide a satisfactory behavior of the control scheme.

In this chapter, we also demonstrated the superiority of the proposed control law with a behavior controller over all other classical control laws. As shown in simulation results, only control law G is able to avoid the local minima and reach the desired pose while all other classical control laws fail to avoid this local minima.

The experimental results show that control law G helps to decrease the bad effect due to a singularity situation and avoid it completely when the behavior controller β of the hybrid interaction matrix is selected perfectly. In fact, the suitable values of the behavior controller rely on the displacement that the camera has to realize, as illustrated on Fig. 3.14 for the rotation around the optical axis.

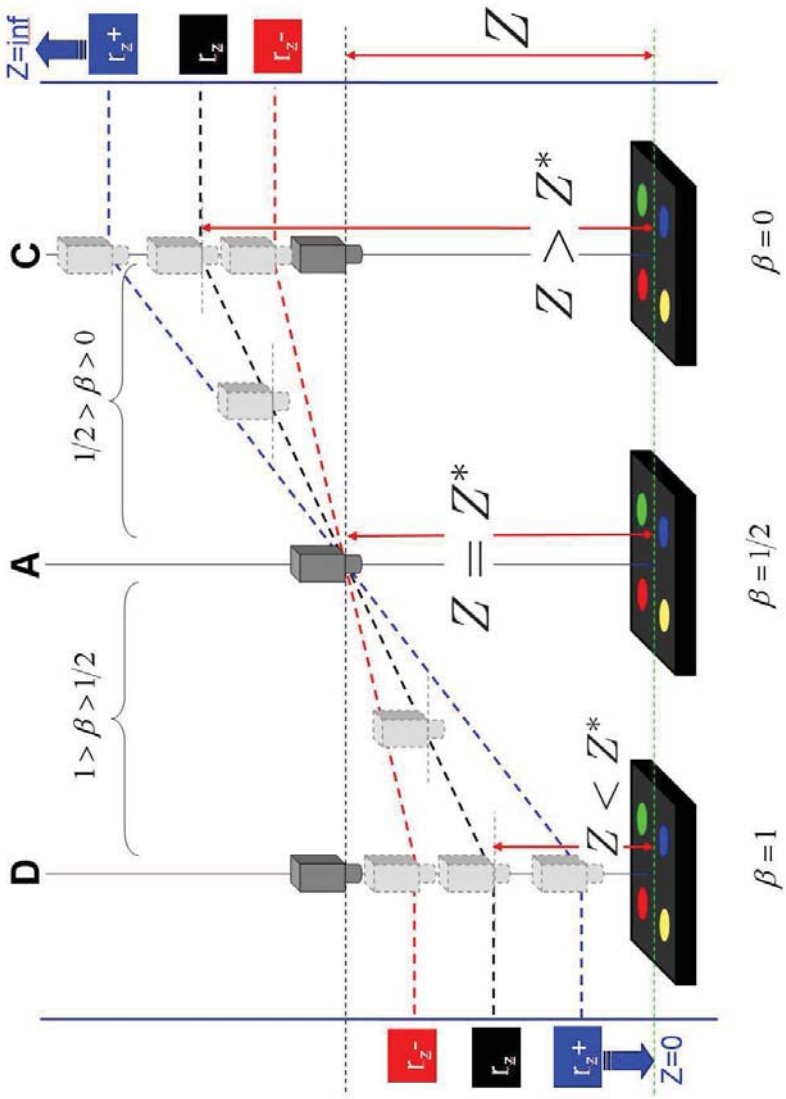


Figure 3.14 – Behavior for a rotation around the optical axis

Chapter 4

New second order control schemes

Generally, image-based visual servoing has been found to give satisfactory, accurate and robust results. However, and as established in the previous chapter, singularity and local minima may appear causing stability and convergence problems. In this chapter, we present new control schemes based on Halley's method as a temptative to obtain a robust system even when the desired configuration is singular. The new control scheme uses the first and the second order derivatives of the error to be regulated to zero. Hessian matrices of an image point are thus determined to be used in the control schemes. Preliminary experimental results obtained on a 6 dof eye-in-hand system shows that a more accurate positioning can be obtained compared with classical methods. The work described in this chapter leads to the following publication [Marey 09]

This chapter is organized as follow: In Section 4.1, singular configurations are discussed followed by analysing the classical control schemes in Section 4.2. The new second order control schemes is proposed in Section 4.3 and the Hessian matrices is determined in Section 4.4. Finally, experimental results are presented in Section 4.5.

4.1 Introduction

Singular configurations correspond to a loss of rank of the Jacobian matrix that relates the features used as input of the control scheme to the control parameters. The classical control schemes are known to be unstable and very sensitive to noise and perturbations around singular configurations. We will discuss in this section different singularity configurations in IBVS and strategies used for avoiding this situation.

4.1.1 Different singular configurations in IBVS

Several singular configurations in IBVS have been exhibited in the literature. As explained in chapter 2, a singularity is encountered by all classical controls schemes for a pure rotation of 180^o around the camera optical axis, and as exhibited in chapter 3, a singularity also exists for control A when a pure rotation of 90^o around the optical axis has to be performed. In the following, additional singularity configurations are discussed.

Singularity of three points

The most well known singularity appears for a target composed of three points. Indeed, for that target, when the camera optical center lies on the surface of a cylinder built from these three points (see Fig. 4.1), the interaction matrix related to the Cartesian coordinates of the three image points is singular (with rank 5) [Michel 93], while it is of full rank 6 as soon as the camera optical center lies outside of this surface. The same singular configurations exist whatever the image features selected to represent the three points (cylindrical coordinates of the points, parameters representing the three straight lines that can be defined from the three points, etc.)

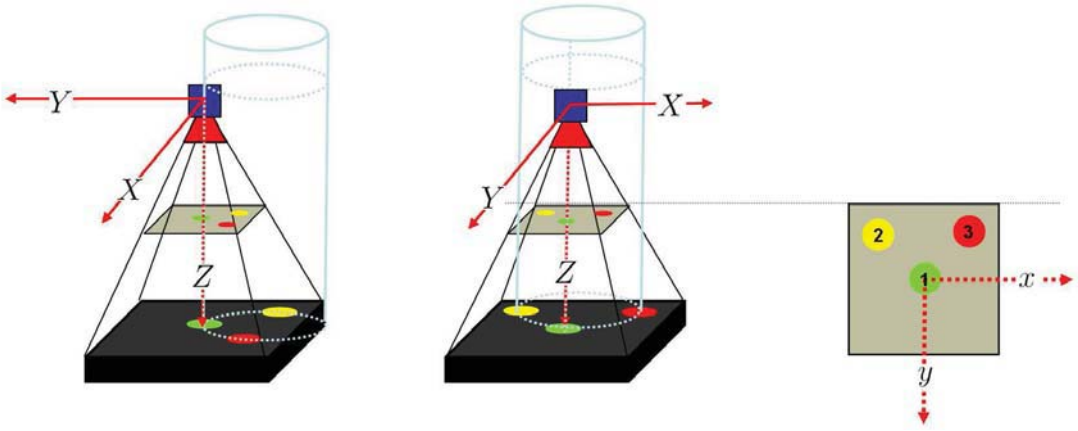


Figure 4.1 – Three point singularity for cylinder configuration

Another singularity of course occurs when the three points are aligned. But we are in that case in a degenerate configuration whatever the camera pose.

Singularity of a circle

Another singular configuration has been exhibited in [Chaumette 93]: if the target is a circle, (see Fig. 4.2), then the interaction matrix related to any set of parameters that represents the image of that circle, which is an ellipse in the general case, is always of rank 5, but when

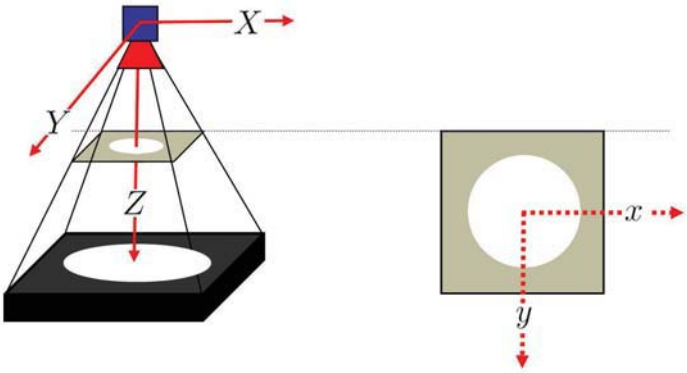


Figure 4.2 – Centered circle singularity

the circle appears in the image as a centered circle, in which case the interaction matrix if of rank 3.

Singularity of the norm

More general singular configurations can be exhibited: whatever the set $\mathbf{s}$ of features selected and its desired value $\mathbf{s}^*$, the interaction matrix related to $\|\mathbf{s} - \mathbf{s}^*\|$ is always of full rank 1 but when $\mathbf{s} = \mathbf{s}^*$, in which case the interaction matrix is null [Fruchard 06]. This case is extremely problematic since the singularity occurs at the desired configuration where we would like the system to be stable and robust. As we will see in Chapter 6, this singularity can be solved by defining a switching strategy when the system nears its desired configuration.

4.1.2 Avoiding singularities in IBVS

Adding feature

Usually, these singular configurations are avoided trivially by selecting features such that their interaction matrix is always of full rank, that is by considering a fourth point when the original target is a set of three points, or by considering two circles or a circle and a point instead of just a circle, or by using $\mathbf{s}$ instead of $\|\mathbf{s} - \mathbf{s}^*\|$. This is not completely satisfactory from a scientific point of view, and may not be always possible in practice, when only three points can be extracted in the image for instance.

Redundancy

When the redundancy framework can be applied, that is when the main task does not control all the robot degrees of freedom (dof), a secondary objective can be designed to try to avoid the singular configurations [Nelson 95, Marchand 96]. Once again, this method can not always be used, typically when the task constrains all the robot dof. We will see that the

new projection operator that we propose in the second part of this thesis can be used to solve this problem.

Damped least square

The previous solution is not efficient if the goal is to reach a singular configuration. To deal with this problem, a classical approach in robotics is to use the damped-least-squares inverse [Wampler 86, Nakamura 86, Egeland 90, Chiaverini 97] instead of the Moore-Penrose pseudo inverse. This method, which artificially increases the lowest singular values of the Jacobian matrix, reduces the effect of the singularity in terms of robustness, but decreases the precision of the control.

Regularization

Finally, a regularization technique has recently been introduced in [Fruchard 06]. It also allows reducing the effect of the singularity, but with the price of decreasing the convergence speed, which is inefficient if the task consists in tracking a moving target.

4.2 Analysis of classical control schemes

Again, let $\mathbf{s}, \mathbf{s}^* \in \mathbb{R}^k$ be the vectors of current and desired set of selected k visual features and $\mathbf{v} \in \mathbb{R}^6$ the instantaneous velocity of the camera. As before, the classical form of the control laws is given by:

$$\mathbf{v} = -\lambda \, \widehat{\mathbf{L}_\mathbf{s}}^{+} (\mathbf{s} - \mathbf{s}^*) \tag{4.1}$$

In this chapter, we are interested in the case where the interaction matrix is singular at the desired configuration, that is when $\mathbf{L}_{\mathbf{s}^*}$ is singular. In that case, control law D is of course inefficient since it is subject to numerous local minima. Indeed, all configurations such that:

$$(\mathbf{s} - \mathbf{s}^*) \in \mathcal{N}\left(\mathbf{L}_{\mathbf{s}^*}^{+}\right)$$

(where $\mathcal{N}(\mathbf{A})$ is the null space of matrix $\mathbf{A}$) correspond to a local minimum, and such configurations are generally numerous since $\mathcal{N}\left(\mathbf{L}_{\mathbf{s}^*}^{+}\right)$ is at least of dimension 1. If the initial error $\mathbf{s}_i - \mathbf{s}^*$ is large and $\mathbf{L}_{\mathbf{s}_\mathbf{i}}$ is not singular, control law C can be used at the beginning of the servo, but, as soon as $\mathbf{s} - \mathbf{s}^*$ will become small, the system will be unstable since it nears the singularity. The same comment can unfortunately be done for control law A, even if we could hope for some smoothing effects of the singularity thanks to the use of the constant matrix $\mathbf{L}_{\mathbf{s}^*}$ in A. This simple analysis of the behavior of control laws D, C, and A will be confirmed in Section 4.5 through experimental results.

To avoid the unstability near the singularity of all control schemes based on the interaction matrix only, we could think of using second order schemes, such as the one based on the classical Newton minimization method. It is given by (see section 2.5.4):

$$\mathbf{v} = -\lambda \, \mathbf{K}_1^{+} \mathbf{L}_\mathbf{s}^{\top} (\mathbf{s} - \mathbf{s}^*) \tag{4.2}$$

where

$$\mathbf{K}_1 = \mathbf{L}_\mathbf{s}^\top \mathbf{L}_\mathbf{s} + \sum_{i=1}^{k} \mathbf{H}_{s_i}(s_i - s_i^*)$$

$\mathbf{H}_{s_i}$ being the Hessian matrix of the i-th component of $\mathbf{s}$. Unfortunately, the convergence domain of this control scheme is generally very limited due to the fact that the Hessian is not always positive definite (see Section 4.4.2). Furthermore, all configurations such that $\mathbf{L}_\mathbf{s}$ is singular and

$$(\mathbf{s} - \mathbf{s}^*) \in \mathcal{N}\left(\mathbf{L}_\mathbf{s}^\top\right)$$

correspond to a local minimum since we have in that case $\mathbf{v} = \mathbf{L}_\mathbf{s}^\top(\mathbf{s} - \mathbf{s}^*) = 0$. This is also of course the same for the basic control scheme

$$\mathbf{v} = -\lambda\, \mathbf{L}_\mathbf{s}^\top(\mathbf{s} - \mathbf{s}^*)$$

based on the steepest descent and usually named gradient method.
All control schemes described above being not satisfactory when trying to reach a singular configuration, we propose in the next section new control schemes based on Halley's method.

4.3 New second order control schemes

4.3.1 Halley's Method for scalar case

Halley's method is well known in the numerical analysis community to find a root of a function $f(x)$ (that is to find x_r such that $f(x_r) = 0$) [Ortega 00]. As classical gradient and Newton methods, it is an iterative algorithm that can be applied if function f is continuous and twice differentiable. It is based on the second order Taylor expansion of f:

$$f(x) = f(x_n) + f'(x_n)(x - x_n) + \frac{1}{2}\, f''(x_n)(x - x_n)^2 \tag{4.3}$$

where x_n is the estimate of x_r at iteration n of the algorithm. Let x_{n+1} be the root of $f(x) = 0$. It can be written:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n) + \frac{1}{2} f''(x_n)(x_{n+1} - x_n)}$$

This equation can not be used directly since x_{n+1} appears both in its left and right sides. However, using on the right side, the result of the Newton-Raphson step (which is easily obtained by solving the first order Taylor expansion $f(x) = f(x_n) + f'(x_n)(x - x_n)$), that is

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

we obtain

$$x_{n+1} = x_n - \frac{2 f(x_n) f'(x_n)}{2\left[f'(x_n)\right]^2 - f(x_n) f''(x_n)} \tag{4.4}$$

which is known as the Halley' rational formula. We can note that, thanks to the term $f(x_n)f''(x_n)$, there is no inversion problem when $f'(x_n) = 0$ as long as $f(x_n) \neq 0$ and $f''(x_n) \neq 0$.

We now apply exactly the same reasoning for the case where x and $f(x)$ are not scalar values but vectors.

4.3.2 Modeling the new control schemes

As before, let $\mathbf{p}$ and $\mathbf{p}^*$ be the parameters that represent the current and the desired camera poses as before and $s : \mathbb{R}^3 \times SO_3 \mapsto \mathbb{R}^n$ a mapping function from the camera pose space to the feature space. The first order Taylor expansion of $\mathbf{s}(\mathbf{p})$ is given by

$$\mathbf{s}^* = \mathbf{s} + \mathbf{J_s}\Delta\mathbf{p}$$

where $\mathbf{s}^* = \mathbf{s}(\mathbf{p}^*)$, $\mathbf{s} = \mathbf{s}(\mathbf{p})$ and $\Delta\mathbf{p}$ represents the displacement between $\mathbf{p}^*$ and $\mathbf{p}$. As presented in Chapter 2, we immediately deduce the following control law using the Newton-Raphson method:

$$\mathbf{v}_1 = -\lambda_1 \mathbf{L}_\mathbf{s}^+ (\mathbf{s} - \mathbf{s}^*) \tag{4.5}$$

where $\mathbf{J_s}$ and $\mathbf{L_s}$ are linked by $\mathbf{L_s} = \mathbf{J_s}\mathbf{P}$, where $\mathbf{P}$ is defined such that $\dot{\mathbf{p}} = \mathbf{P}\mathbf{v}$. Note that control law (4.5) is nothing but the classical control law C.

Let us now consider the second order Taylor expansion of $\mathbf{s}$. It is given by

$$\mathbf{s}^* = \mathbf{s} + \mathbf{K}_2 \, \Delta\mathbf{p} \tag{4.6}$$

where matrix $\mathbf{K}_2$ is

$$\mathbf{K}_2 = \mathbf{J_s} + \frac{1}{2} \begin{bmatrix} \Delta\mathbf{p}^\top \mathbf{H}_{s_1} \\ \cdots \\ \cdots \\ \cdots \\ \Delta\mathbf{p}^\top \mathbf{H}_{s_k} \end{bmatrix} \tag{4.7}$$

Solving (4.6) for $\Delta\mathbf{p}$, we obtain

$$\Delta\mathbf{p} = -\mathbf{K}_2^+ (\mathbf{s} - \mathbf{s}^*) \tag{4.8}$$

from which we deduce the following control law:

$$\mathbf{v} = -\lambda \mathbf{K}_\mathbf{s}^+ (\mathbf{s} - \mathbf{s}^*) \tag{4.9}$$

where the output (4.5) of control law C is used to go from $\mathbf{K}_2$ to $\mathbf{K_s}$:

$$\mathbf{K_s} = \mathbf{L_s} - \frac{\lambda_1}{2} \begin{bmatrix} (\mathbf{s} - \mathbf{s}^*)^\top \mathbf{L}_\mathbf{s}^{+^\top} \mathbf{H}_{s_1} \\ \cdots \\ \cdots \\ \cdots \\ (\mathbf{s} - \mathbf{s}^*)^\top \mathbf{L}_\mathbf{s}^{+^\top} \mathbf{H}_{s_k} \end{bmatrix} \tag{4.10}$$

This control law is named K in the following. Let us note that, if $\mathbf{L_s}$ is singular, it does not necessarily imply that $\mathbf{K_s}$ is singular thanks to the Hessian part involved in this control scheme. Furthermore and contrarily to the control law (4.2) based on the Newton method, when $\mathbf{L_s}$ is singular, the configurations such that

$$(\mathbf{s} - \mathbf{s}^*) \in \mathcal{N}\left(\mathbf{L}_\mathbf{s}^\top\right)$$

does not generally correspond to a local minimum. That are for the good points of K. Unfortunately, some bad points also exist. First, $\mathbf{K_s}$ may be singular for some configurations where $\mathbf{L_s}$ is not singular. Then, for $\mathbf{s} = \mathbf{s}^*$, $\mathbf{K_s}$ is singular when $\mathbf{L_{s^*}}$ is singular (since $\mathbf{K_s} = \mathbf{L_{s^*}}$ in that case). However, since we never have $\mathbf{s} = \mathbf{s}^*$ in practice, due to unavoidable image noise, we will never have exactly $\mathbf{K_s} = \mathbf{L_{s^*}}$, which makes appealing the use of K when $\mathbf{L_{s^*}}$ is singular. Furthermore, near the singularity, the low conditioning of $\mathbf{L_s}$ in the first part of $\mathbf{K_s}$ is compensated by the high conditioning of $\mathbf{L_s}$ in its second part. However, we will see in Section 4.5 that if the rank of $\mathbf{K_s}$ is indeed improved, it increases the sensitivity of the control scheme to the image noise.

Following the same idea than going from control law C to D, we could think of using:

$$\mathbf{v} = -\lambda \mathbf{K}_{\mathbf{s}^*}^+ (\mathbf{s} - \mathbf{s}^*) \tag{4.11}$$

where $\mathbf{K_{s^*}}$ is given by:

$$\mathbf{K_{s^*}} = \mathbf{L_{s^*}} - \frac{\lambda_1}{2} \begin{bmatrix} (\mathbf{s} - \mathbf{s}^*)^\top \mathbf{L}_{\mathbf{s}^*}^{+\top} \mathbf{H}_{s_1^*} \\ \cdots \\ \cdots \\ \cdots \\ (\mathbf{s} - \mathbf{s}^*)^\top \mathbf{L}_{\mathbf{s}^*}^{+\top} \mathbf{H}_{s_k^*} \end{bmatrix}$$

However, that is definitively not a good idea since this control scheme has exactly the same bad properties of D that all configurations such that

$$(\mathbf{s} - \mathbf{s}^*) \in \mathcal{N}\left(\mathbf{L}_{\mathbf{s}^*}^+\right)$$

will lead to a local minima (to check that, just note that in that case $(\mathbf{s}-\mathbf{s}^*)^\top \mathbf{L}_{\mathbf{s}^*}^{+\top} = \mathbf{0}$, which implies $\mathbf{K_{s^*}} = \mathbf{L_{s^*}}$).

A last control scheme can be obtained by considering each feature independently in the second part of $\mathbf{K_s}$, that is using

$$\mathbf{v}_i = -\lambda_1 \mathbf{L}_{s_i}^+ (s_i - s_i^*) \tag{4.12}$$

instead of (4.5). In that case we obtain

$$\mathbf{v} = -\lambda \mathbf{K}_i^+ (\mathbf{s} - \mathbf{s}^*) \tag{4.13}$$

where

$$\mathbf{K}_i = \mathbf{L_s} - \frac{\lambda_1}{2} \begin{bmatrix} (s_1 - s_1^*)^\top \mathbf{L}_{s_1}^{+\top} \mathbf{H}_{s_1} \\ \cdots \\ \cdots \\ \cdots \\ (s_k - s_k^*)^\top \mathbf{L}_{s_k}^{+\top} \mathbf{H}_{s_k} \end{bmatrix}$$

This control law will be named Ki in the following. Even if we are currently unable to give any theoretical explanation, we will see in Section 4.5 that this control law allows improving the accuracy of the positioning in a singular configuration.

4.4 Hessian Matrices of an image point

In the experiments presented in the next section, we will compare the behavior of the control schemes presented in this chapter in the case of a target composed of three points. The analytical form of the Hessian matrices $\mathbf{H}_x$ and $\mathbf{H}_y$ of the coordinates (x, y) of an image point are thus needed. Let us note that these matrices have already been used in [Lapresté 04], but the analytical form given in that paper contains unfortunately few typos errors.

4.4.1 Modeling

We recall that an image point with coordinates (x, y) results from the perspective projection of a 3D point such that $x = X/Z$ and $y = Y/Z$ where (X, Y, Z) are the coordinates of this 3D point expressed in the camera frame. We also recall that the velocity $(\dot{x}, \dot{y})$ of an image point is linked to the camera velocity $\mathbf{v} = (v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)$ through the well known equations:

$$\dot{x} = \mathbf{L}_x \mathbf{v} \ , \ \dot{y} = \mathbf{L}_y \mathbf{v} \tag{4.14}$$

where the interaction matrices $\mathbf{L}_x$ and $\mathbf{L}_y$ are given by:

$$\mathbf{L}_x = \begin{bmatrix} \frac{-1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \end{bmatrix} \tag{4.15}$$

$$\mathbf{L}_y = \begin{bmatrix} 0 & \frac{-1}{Z} & \frac{y}{Z} & (1+y^2) & -xy & -x \end{bmatrix} \tag{4.16}$$

The Hessian matrices $\mathbf{H}_x$ and $\mathbf{H}_y$ can easily be determined by differentiating (4.14). Indeed, for any feature s, we have:

$$\ddot{s} = \mathbf{L}_s \dot{\mathbf{v}} + \mathbf{v}^\top \mathbf{H}_s \mathbf{v} \tag{4.17}$$

where $\mathbf{H}_s$ is a symetrix matrix. Using (4.14), (4.15) and (4.16), we obtain:

$$\ddot{x} = \mathbf{L}_x \dot{\mathbf{v}} + \frac{\dot{Z}}{Z^2} v_x + \frac{\dot{x}Z - \dot{Z}x}{Z^2} v_z + (\dot{x}y + x\dot{y})\ \omega_x - 2x\dot{x}\ \omega_y + \dot{y}\ \omega_z$$

$$\ddot{y} = \mathbf{L}_y \dot{\mathbf{v}} + \frac{\dot{Z}}{Z^2} v_y + \frac{\dot{y}Z - \dot{Z}y}{Z^2} v_z + 2y\dot{y}\ \omega_x - (\dot{x}y + x\dot{y})\ \omega_y - \dot{x}\ \omega_z$$

By substituting (4.14) for $\dot{x}$ and $\dot{y}$, and knowing that:

$$\dot{Z} = -v_z - yZ\omega_x + xZ\omega_y$$

we obtain after simple developments:

$$\begin{aligned}\ddot{x} &= \mathbf{L}_x\dot{\mathbf{v}} - \frac{2}{Z^2}\,v_xv_z - \frac{2y}{Z}\,v_x\omega_x + \frac{3x}{Z}\,v_x\omega_y - \frac{x}{Z}\,v_y\omega_x - \frac{1}{Z}\,v_y\omega_z \\ &\quad + \frac{2x}{Z^2}\,v_zv_z + \frac{4xy}{Z}\,v_z\omega_x - \frac{(1+4x^2)}{Z}\,v_z\omega_y + \frac{2y}{Z}\,v_z\omega_z \\ &\quad + x(1+2y^2)\,\omega_x\omega_x - y(1+4x^2)\,\omega_x\omega_y + (1+2y^2-x^2)\,\omega_x\omega_z \\ &\quad + 2x(1+x^2)\,\omega_y\omega_y - 3xy\,\omega_y\omega_z - x\,\omega_z\omega_z\end{aligned}$$

$$\begin{aligned}\ddot{y} &= \mathbf{L}_y\dot{\mathbf{v}} + \frac{y}{Z}\,v_x\omega_y + \frac{1}{Z}\,v_x\omega_z - \frac{2}{Z^2}\,v_yv_z - \frac{3y}{Z}\,v_y\omega_x + \frac{2x}{Z}\,v_y\omega_y \\ &\quad + \frac{2y}{Z^2}\,v_zv_z + \frac{1+4y^2}{Z}\,v_z\omega_x - \frac{4xy}{Z}\,v_z\omega_y - \frac{2x}{Z}\,v_z\omega_z \\ &\quad + 2y(1+y^2)\,\omega_x\omega_x - x(1+4y^2)\,\omega_x\omega_y + y(1+2x^2)\,\omega_y\omega_y \\ &\quad + (1+2x^2-y^2)\,\omega_y\omega_z - y\,\omega_z\omega_z\end{aligned}$$

from which we deduce by identification with (4.17)

$$\mathbf{H}_x = \begin{bmatrix} 0 & 0 & \frac{-1}{Z^2} & \frac{-y}{Z} & \frac{3x}{2Z} & 0 \\ 0 & 0 & 0 & \frac{-x}{2Z} & 0 & \frac{-1}{2Z} \\ \frac{-1}{Z^2} & 0 & \frac{2x}{Z^2} & \frac{2xy}{Z} & \frac{-1-4x^2}{2Z} & \frac{y}{Z} \\ \frac{-y}{Z} & \frac{-x}{2Z} & \frac{2xy}{Z} & x(1+2y^2) & -y(\frac{1}{2}+2x^2) & \frac{1-x^2+2y^2}{2} \\ \frac{3x}{2Z} & 0 & \frac{-1-4x^2}{2Z} & -y(\frac{1}{2}+2x^2) & 2x(1+x^2) & \frac{-3xy}{2} \\ 0 & \frac{-1}{2Z} & \frac{y}{Z} & \frac{1-x^2+2y^2}{2} & \frac{-3xy}{2} & -x \end{bmatrix}$$

and

$$\mathbf{H}_y = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{y}{2Z} & \frac{1}{2Z} \\ 0 & 0 & \frac{-1}{Z^2} & \frac{-3y}{2Z} & \frac{x}{Z} & 0 \\ 0 & \frac{-1}{Z^2} & \frac{2y}{Z^2} & \frac{1+4y^2}{2Z} & \frac{-2xy}{Z} & \frac{-x}{Z} \\ 0 & \frac{-3y}{2Z} & \frac{1+4y^2}{2Z} & 2y(1+y^2) & -x(\frac{1}{2}+2y^2) & \frac{-3xy}{2} \\ \frac{y}{2Z} & \frac{x}{Z} & \frac{-2xy}{Z} & -x(\frac{1}{2}+2y^2) & y(1+2x^2) & \frac{1-y^2+2x^2}{2} \\ \frac{1}{2Z} & 0 & \frac{-x}{Z} & \frac{-3xy}{2} & \frac{1-y^2+2x^2}{2} & -y \end{bmatrix}$$

4.4.2 Positiveness of $\mathbf{H}_x$ and $\mathbf{H}_y$

Using the determinant test to study the positiveness of the Hessian matrices $\mathbf{H}_x$ and $\mathbf{H}_y$, we found that the determinants of the leading principal minor vectors of $\mathbf{H}_x$ and $\mathbf{H}_y$ are $\mathbf{M}_x = (0,0,0,\frac{x^2}{4Z^6},\frac{x^3(1+x^2)}{Z^6},0)$ and $\mathbf{M}_y = (0,0,0,0,\frac{y^3(1+y^2)}{Z^6},0)$ respectively, where $\mathbf{M}_s[i] = |\mathbf{H}_s[1..i,1..i]|$. This means that the necessary and sufficient condition for both $\mathbf{H}_x$ and $\mathbf{H}_y$ to be positive semi-definite is that x and y are positive, which is of course not always achieved. This may explain the fact that Newton and Halley's methods based on image point coordinates have a small convergence domain.

4.5 Results

The experimental results presented in this section have been obtained on a 6 DOF eye-in-hand system (see Fig. 3.8). As before, all the control schemes have been easily implemented thanks to the open source ViSP library [Marchand 05b]. The task consists of positioning the camera with respect to a target composed of three points (in practice, three white dots on a black background to avoid any image processing problem) using the Cartesian coordinates of the perspective projection of these points in the image. The three points form a rectangle isosceles triangle whose side lengths are equal to 0.06 m, 0.06 m and $0.06 \times \sqrt{2}$ m. The interaction matrix is thus of dimension 6×6 and of full rank 6 but for the singularities exhibited in [Michel 93] in which case it is of rank 5. We recall that the singularities occur when the optical center of the camera belongs to the surface of the right circular cylinder whose basis is defined by the circle to which the three points belong. From this general result, it is easy to see that if one of the three points appears at the principal point in the image (which corresponds to the image of the optical axis), then the interaction matrix is singular.

The desired pose between the camera and the triangle has thus been chosen such that they are parallel (at a distance of 0.5 m) and one point appears at the principal point (see Fig. 4.3). The initial pose has been chosen very near from the desired one, that is $\mathbf{p_i} = (0.0022, 0.001, 0.501, 0.8, 0.4, 0.6)$ where the first three components represent the translation expressed in meter, and the last three ones represent the rotation expressed in degree. We are indeed interested in the behavior near the singularity.

Let us finally note that the depth of the points, which appears in the translational term of the interaction matrix and in the Hessian matrix, are estimated at each iteration of the control scheme using a classical pose estimation method. Let us also note that the gain λ_1 involved in K and Ki has been set to 1, and the gain λ involved in all the control schemes has been set to 0.5. This value has voluntarily been chosen small to show the effects of the singularity and to avoid any unstability due to a too high value of λ. The very large number of iterations in each experiment is thus not significant. As for the singular value decomposition used to compute the pseudo-inverses involved in the different control schemes, the condition number threshold has been set to 0.0001. We recall that the condition number is the ratio between the minimal and the maximal singular values of a matrix, and the threshold is used to compute its rank and to consider if a singular value is zero or not. This value has been chosen to not damage the robot by forbidding high values in the outputs of the control scheme.

The results obtained for control laws D, C, M, K and Ki are given on Figs. 4.3, 4.4, 4.5, 4.6 and 4.7 respectively. As expected, control law D is always of rank 5. It is thus not surprising that it reaches a local minimum. Contral law C is of rank 6 at the beginning of the servo, which allows the system to near the desired position. It is then of rank 5 due to the high condition number threshold, but for some iterations where it becomes again of rank 6, due to image noise, producing high robot velocities at these iterations. Control law A has not a very satisfactory behavior: even if it is of rank 6 at the beginning of the servo, it fails,

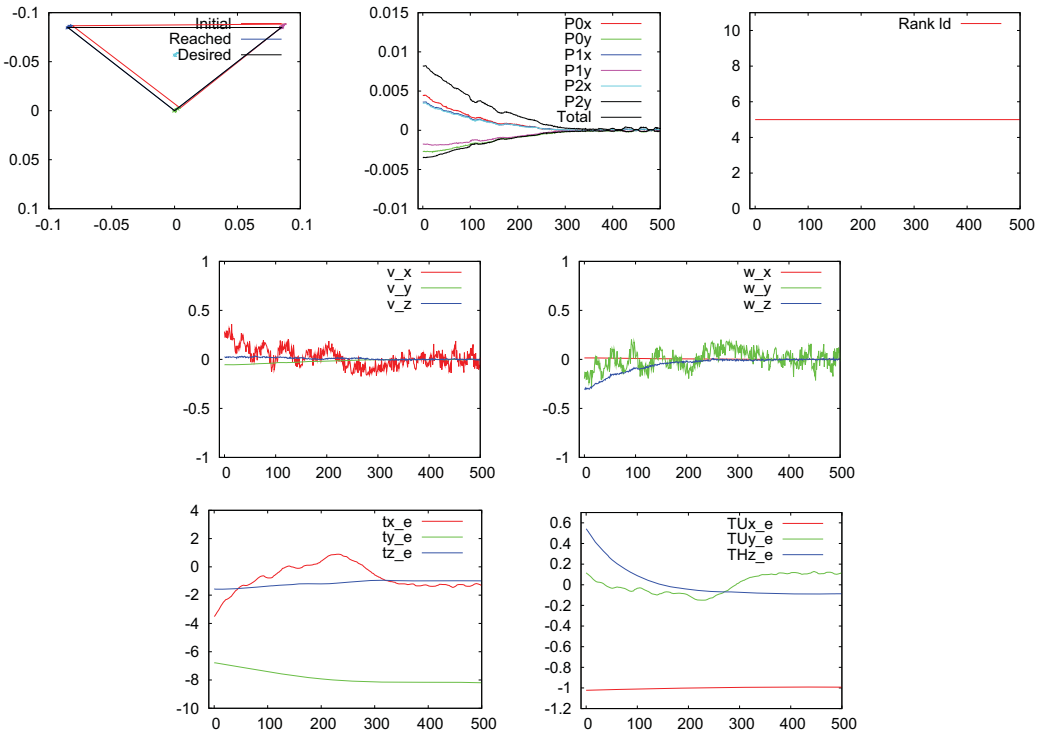


Figure 4.3 – Experimental results of control law D. First line from the left: initial and desired images, visual features errors and rank of the control matrix; second line: translational (cm/s) and rotational components (dg/s) components of the control law; third line: translational (in mm) and rotational (in dg) error components

as D, in a local minimum. As for K, it is almost always of rank 6, as expected, but it reaches also a local minimum and is quite unstable due to the fact that it is of rank 6. Finally, control law Ki provides with the best behavior, similar to the one of C, but with a better positioning accuracy and less noise.

4.6 Conclusion

In this chapter, we have been interested by the difficult problem of reaching a visual singular configuration. Without any surprise, all classical control schemes have been shown to be unsatisfactory. Control schemes based on second order minimization Halley's method have been proposed to try to improve the behavior of the system near the desired singular position. The experimental results obtained have shown that it is possible to improve the accuracy of the positioning using one of these control schemes.

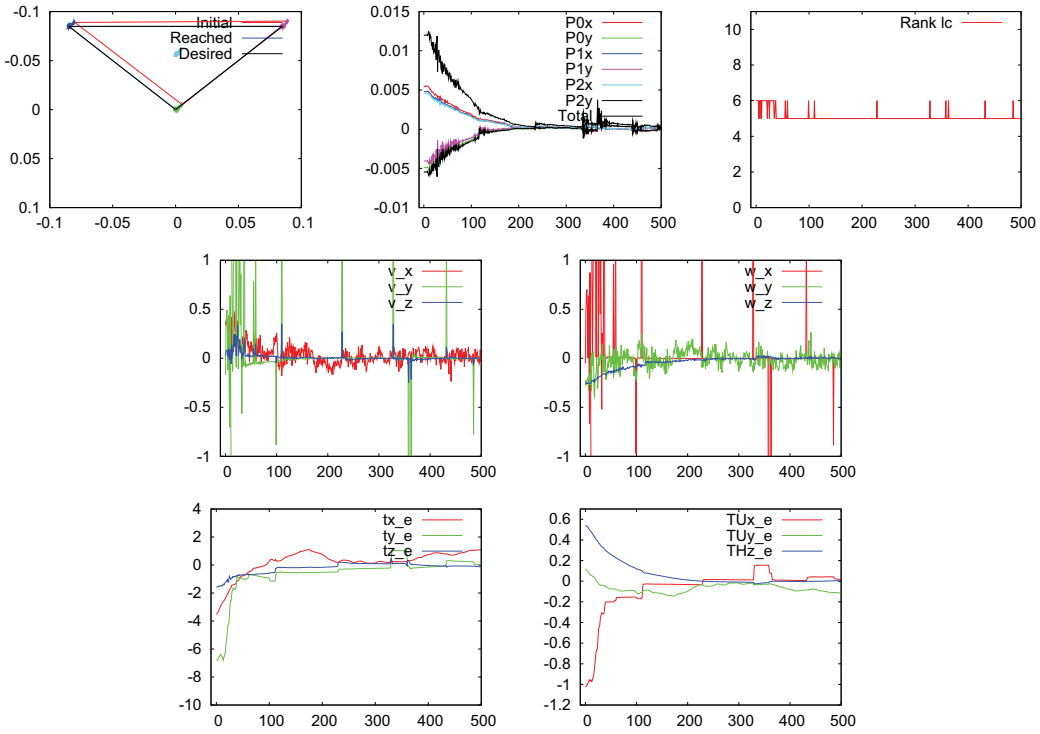


Figure 4.4 – Experimental results of control law C.

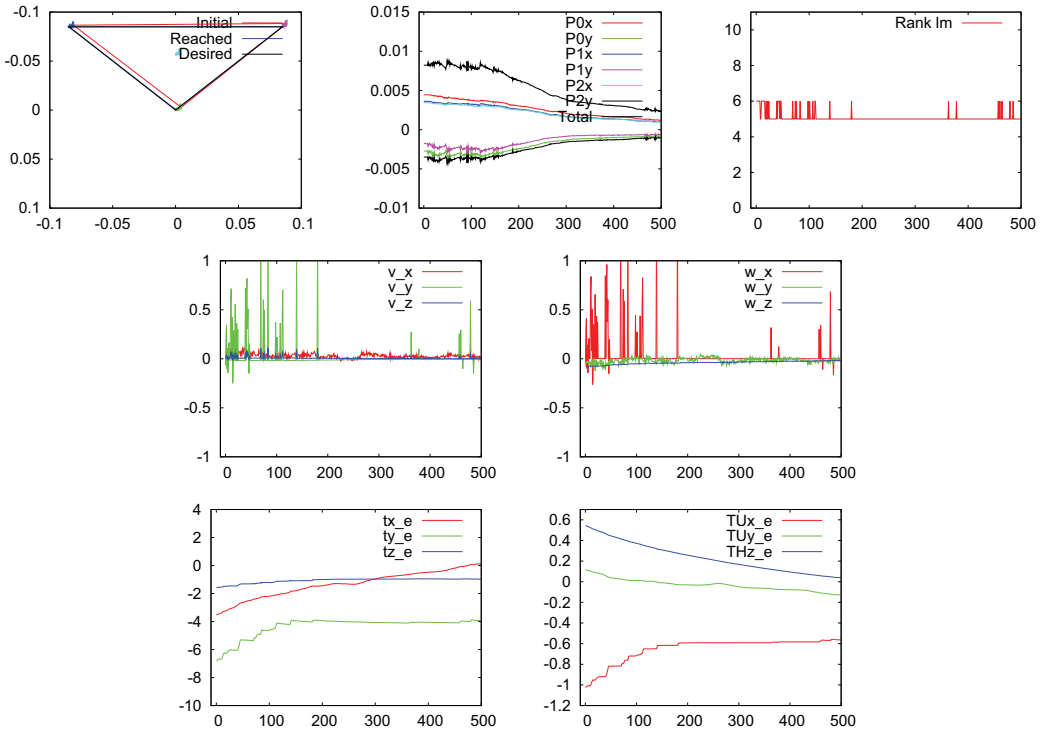


Figure 4.5 – Experimental results of control law M

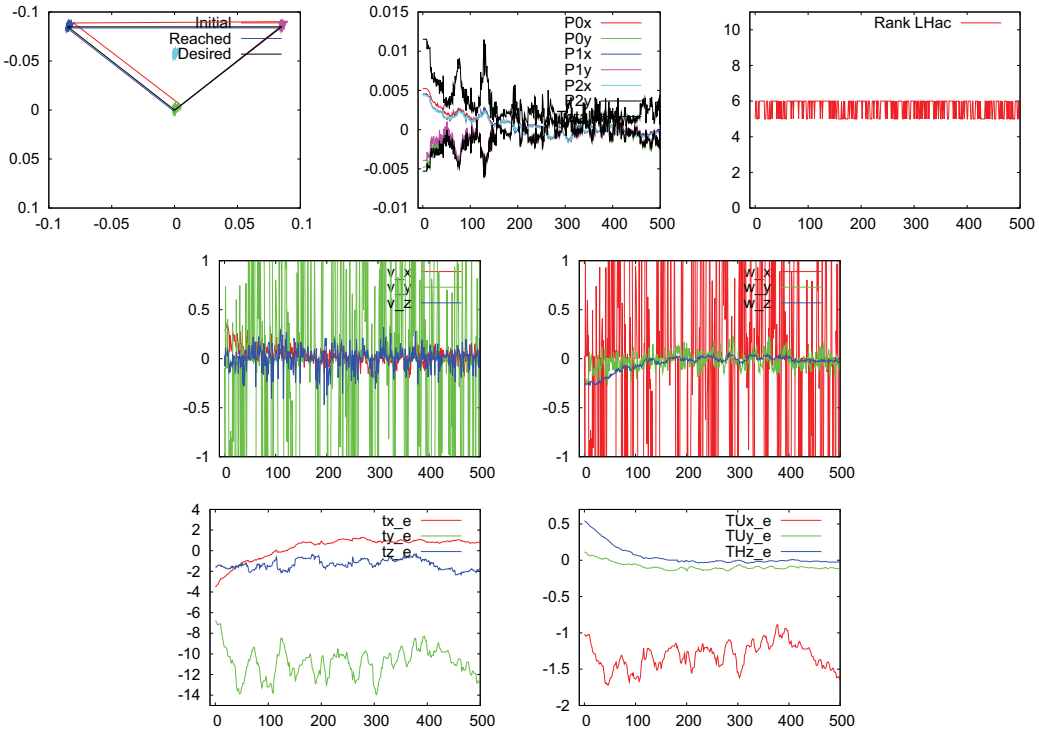


Figure 4.6 – Experimental results of control law K

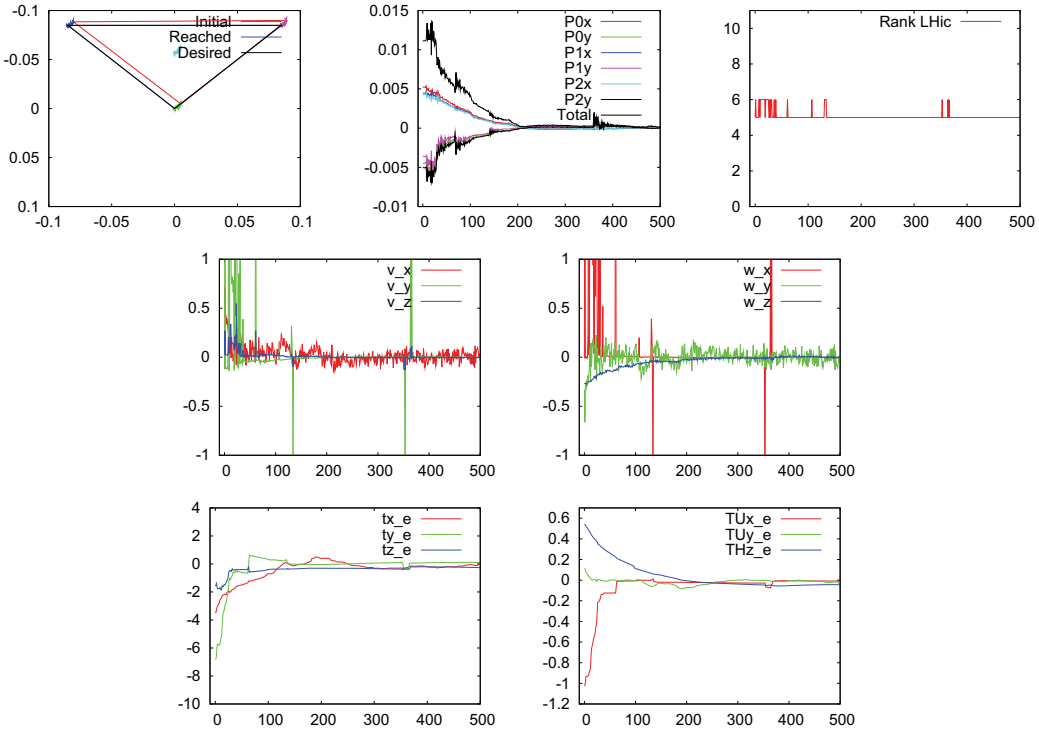


Figure 4.7 – Experimental results of control law Ki

Part II

Redundancy formalism

Chapter 5

State of the art in redundancy

This chapter presents the state of the art of the redundancy framework in robotic. The state of the art of the most common avoidance strategies used for solving the problem of robot joint limits avoidance are also detailed.

5.1 Redundancy

Redundant robots attract special attention due to their dexterity and versatility for addressing sophisticated tasks, particularly for underactuated systems [Shkolnik07]. There are several types of redundancy for a robot: Redundancy with respect to the task when the number of independent parameters required by a task is less than the DOFs of the robot and redundancy with respect to the end effector when the manipulator has more DOFs than those required to place its end effector at a given position and orientation within the task space [Samson 91] [Khatib 96] [Siciliano 91] [Hanafusa 81]. When robots have more DOFs than those required to execute a given task, the excess DOFs can be profitably used for performing other subtasks such as singularity and obstacles avoidance, keeping the joints within their limits, torque optimization, and energy minimization. However, they pose some difficulty in solving the inverse kinematics problem because they provide an infinite number of joint values for a certain end-effector position and orientation.

In robot control, actions are generally operated in the joint space. However, in most practical applications and as mentioned in Part I, it is desired to design a task space controller and to allow the robot considering other constraints. The solution of this problem requires the joint variables to satisfy not only the original task, but also the constraints. This makes interesting the use of a redundant system [Siciliano 91].

The use of redundancy can be formulated in the framework of task with order of priority.

When the main task is defined as to following a given trajectory of the end effector while avoiding joint limits [Euler 89], avoiding singular configurations [Paulin 04], avoiding occlusion [Marchand 98] and/or avoiding obstacles in the workspace [Choi 00] [Perdereau 02], trajectory following has the first priority while joint limits, singularity and obstacle avoidance are given the second priority [Nakamura 87]. In visual servoing applications, the redundancy has been exploited for example to avoid joint limits or occlusions [Chaumette 01] [Marchand 96][Mansard 09a].

5.2 Projection operators

5.2.1 Classical projection operator $\mathbf{P_e}$

Let $\mathbf{e} \in \mathbb{R}^k$ be the main task function where k is the number of its components. The classical approach that tries to ensure an exponential decrease of all components of $\mathbf{e}$ while considering a secondary task leads to the following control scheme [Liegeois 77]:

$$\begin{aligned} \dot{\mathbf{q}} &= \dot{\mathbf{q}}_\mathbf{e} + \mathbf{P_e}\,\mathbf{g} && (5.1)\\ &= \mathbf{J_e^+}\dot{\mathbf{e}}^* + (\mathbf{I}_n - \mathbf{J_e^+}\mathbf{J_e})\,\mathbf{g} && (5.2) \end{aligned}$$

where $\dot{\mathbf{q}} \in \mathbb{R}^\mathbf{n}$ is the robot joint velocity sent as input of the low level robot controller, $\mathbf{J_e} \in \mathbb{R}^{k\times n}$ is the task Jacobian defined such that $\dot{\mathbf{e}} = \mathbf{J_e}\dot{\mathbf{q}}$, n is the number of robot DOFs, $\mathbf{J_e^+}$ is the Moore-Penrose pseudoinverse of $\mathbf{J_e}$, $\dot{\mathbf{e}}^*$ is usually set as $-\lambda\mathbf{e}$ to have an exponential decrease of each component of $\mathbf{e}$, $\mathbf{g}$ represents the motion induced by the secondary task, and $\mathbf{P_e} = (\mathbf{I}_n - \mathbf{J_e^+}\mathbf{J_e})$ is a projection operator on the null space of $\mathbf{J_e}$ so that $\mathbf{g}$ is realized at best under the constraint that it does not perturb the regulation of $\mathbf{e}$ to $\mathbf{0}$ (we have $\mathbf{J_e}\mathbf{P}\mathbf{g} = \mathbf{0}, \forall \mathbf{g}$). This projection operator may be too much constraining: it has no component available when the error $\mathbf{e}$ constrains all the n DOFs of the system (in that case $\mathbf{P_e} = \mathbf{0}$), and only $n - r$ (where r is the rank of $\mathbf{J_e}$) in the general case.

5.2.2 Bidirectional projection operator $\mathbf{P_z}$

A nonlinear projection operator has been recently proposed in [Mansard 09a] to enlarge the free space on which the gradient is projected. This is achieved by decreasing the error of the system when the secondary task goes in the same direction than the main task. This operator is built such that when $\dot{\mathbf{q}} = \mathbf{J}^+\dot{\mathbf{e}}^* + \dot{\mathbf{q}}_\mathbf{2}$, where $\dot{\mathbf{q}}_\mathbf{2} = \mathbf{P}\mathbf{g}$, $\dot{\mathbf{q}}_\mathbf{2}$ respects the condition $\nabla\mathcal{L}^\top\mathbf{J}\dot{\mathbf{q}}_\mathbf{2} \leq \mathbf{0}$, where $\nabla = \frac{\partial}{\partial \mathbf{e}}$ and $\mathcal{L}$ is the Lyapunov function associated to $\dot{\mathbf{q}} = \mathbf{J}^+\dot{\mathbf{e}}^*$ such that $\dot{\mathcal{L}} = \frac{\partial}{\partial \mathbf{e}}\dot{\mathbf{e}} < \mathbf{0}$. By considering the singular values decomposition of $\mathbf{J} = \mathbf{U}\Sigma\mathbf{V}^\top$, where $\mathbf{U}$ is a basis of the task function space, $\mathbf{V}$ is a basis of the joint space, $\mathbf{\Sigma} = [\mathbf{\Delta}_\sigma\ \mathbf{0}]$, and Δ_σ is the diagonal matrix whose coefficients are the m singular values of $\mathbf{J}$, the condition can be written as $\widetilde{\nabla\mathcal{L}}^\top\mathbf{\Sigma}\dot{\widetilde{\mathbf{q}}}_\mathbf{2} \leq \mathbf{0}$, where $\widetilde{\nabla\mathcal{L}} = \mathbf{U}^\top\nabla\mathcal{L}$ and $\dot{\widetilde{\mathbf{q}}}_\mathbf{2} = \mathbf{V}^\top\dot{\mathbf{q}}_\mathbf{2}$. This leads to a restricted condition:

$$\forall i \in [1 \cdots m], \widetilde{\mathbf{v}}\sigma\dot{\widetilde{q}}_{2_i} \leq \mathbf{0} \qquad (5.3)$$

where $\widetilde{\mathcal{L}} = (v_1, \cdots, v_m)$.

For a given $\mathbf{g}$, $\mathbf{q_2}$ that respect the condition (5.3) is built by keeping the components that respect (5.3) and nullifying the other components. This leads to obtaining the projection operator as:

$$\mathbf{P_g} = \mathbf{V} \begin{bmatrix} p_1(\tilde{g}) & & 0 \\ & \ddots & \\ 0 & & p_n(\tilde{g}) \end{bmatrix} \mathbf{V}^\top \tag{5.4}$$

where $p_i(\tilde{\mathbf{g}})$ is defined by:

$$p_i(\mathbf{g}) = \begin{cases} 1 & \text{if } i > m \text{ or } \tilde{g}_i = 0 \\ 1 & \text{if } \tilde{g}_i \text{ and } \tilde{v}_i\sigma_i \text{ have opposite signs} \\ 0 & \text{if } \tilde{g}_i \text{ and } \tilde{v}_i\sigma_i \text{ have the same sign} \end{cases} \tag{5.5}$$

where $\tilde{\mathbf{g}} = \mathbf{V}\mathbf{g}$. Finally, to realize the task $\mathbf{e}$ and respect the condition, the control law is given by:

$$\dot{\mathbf{q}} = \mathbf{J}^+\dot{\mathbf{e}}^* + \mathbf{P_g}\mathbf{g} \tag{5.6}$$

The second term $\mathbf{P_g}\mathbf{g}$ has a proper sign however it can be arbitrary large when the main task converges to zero which may introduce oscillation. This problem is corrected by computing the projection operator from the second-order Taylor expansion which introduces an upper bound on the value of the secondary term. In this case, when considering the norm of the error as a Lyapunov function $\mathcal{L} = \frac{1}{2}\mathbf{e}^\top\mathbf{e}$, the condition to be respected is:

$$\nabla\mathcal{L}^\top\mathbf{J}\Delta\mathbf{q_2} + \frac{1}{2}\Delta\mathbf{q_2}^\top\mathbf{J}^\top\mathbf{J}\Delta\mathbf{q_2} < \mathbf{0} \tag{5.7}$$

where $\nabla\mathcal{L}^\top = (\mathbf{e} + \Delta\mathbf{e}^*)^\top$.

When introducing the SVD bases and performing some simplifications, the condition can be written as: $\forall i \in [1 \cdots m],\ \ 2(\tilde{e}_i + \widetilde{\Delta e_i^*})\sigma_i\widetilde{\Delta q_{2_i}} + \sigma_i^2\widetilde{\Delta q_{2_i}}^2 < 0$, where $\widetilde{\Delta\mathbf{q_2}}$ is a possible secondary motion that belongs to the free space of the main task if and only if:

$$\forall i \in [1 \cdots m] = \begin{cases} \widetilde{\Delta\mathbf{q_{2_i}}} = 0 \\ \text{or } 0 < \widetilde{\Delta\mathbf{q_{2_i}}} \leq -\frac{2}{\sigma_i}(\tilde{e}_i + \widetilde{\Delta e_i^*}) \\ \text{or } -\frac{2}{\sigma_i}(\tilde{e}_i + \widetilde{\Delta e_i^*}) \leq \widetilde{\Delta\mathbf{q_{2_i}}} \end{cases} \tag{5.8}$$

In order to ensure that $\widetilde{q_{2_i}}$ respects condition (5.8), the projection operator is defined by:

$$\mathbf{P_g} = \mathbf{V} \begin{bmatrix} p_1(\tilde{g}) & & 0 \\ & \ddots & \\ 0 & & p_n(\tilde{g}) \end{bmatrix} \mathbf{V}^\top \tag{5.9}$$

where

$$p_i(\tilde{g}) = \begin{cases} \tilde{g}_i & \text{if } i > m \text{ or } \tilde{g}_i = 0 \\ \tilde{g}_i & \text{if } 0 < \tilde{g}_i - \frac{2}{\sigma_i}(\widetilde{e_i} + \widetilde{\Delta e_i^*}) \\ \tilde{g}_i & \text{if } -\frac{2}{\sigma_i}(\widetilde{e_i} + \widetilde{\Delta e_i^*}) \leq \tilde{g}_i \\ 0 & \text{if } \sigma_i \tilde{g}_i > 0 \text{ and } \widetilde{e_i} + \widetilde{\Delta e_i^*} < 0 \\ 0 & \text{if } \sigma_i \tilde{g}_i < 0 \text{ and } \widetilde{e_i} + \widetilde{\Delta e_i^*} > 0 \\ -\frac{2(\widetilde{e_i} + \widetilde{\Delta e_i^*})}{\sigma_i \tilde{g}_i} & \text{else} \end{cases} \tag{5.10}$$

This projection operator (5.10) obtained by considering the robot as a discrete system allows to avoid the oscillations that appear when the projection operator (5.5) is used. Finally, let us note that even if this projection operator improves the performance of the system, the number of DOF can be insufficient to realize the constraints. We will see in the next chapter that the new projection operator we propose can be used even if all DOFs are used by the main task.

5.3 Redundancy-based task-priority formalism

Various frameworks were proposed in order to manage the redundancy among two or several tasks to control their levels of priority. There are classical and efficient formalisms [Hanafusa 81], [Baerlocher 98] [Lenarcic 98]; Inverse-based and transpose-based projection methods [Hanafusa 81] [Chiacchio 91] [Baerlocher 98]; Successive and augmented projections [Siciliano 91] [Baerlocher 98] [Mansard 04] [Antonelli 09]; and Kinematic and dynamic controllers [Siciliano 91] [Hanafusa 81]. In the following we present a review concerning these issues.

5.3.1 Inverse-based projection methods (successive, augmented)

For tasks with order of priority, the problem is formulated using the kinematic relationship between the joint variable $\mathbf{q} \in \mathbb{R}^{\mathbf{n}}$ and the manipulation variables [Liegeois 77]. As illustrated in Fig. 5.1, the final solution is a homogeneous solution belonging to the null space of $\mathbf{J_1}$.

Successive inverse-based projections

In [Mansard 04], several approaches were discussed to propose suitable ones that enable stacking n different tasks $\mathbf{e_1}, \mathbf{e_2}, ..., \mathbf{e_n}$ using redundancy. In successive inverse-based projections the last task is projected on the null space of its previous task and then project this composed task on the null space of their previous tasks to get:

$$\mathbf{e} = \mathbf{e_1} + \mathbf{P_1}.\mathbf{e_2} + \mathbf{P_1}\mathbf{P_2}.\mathbf{e_3} + \ldots \prod_{i=1}^{n-1} .\mathbf{P_i}\mathbf{e_n} \tag{5.11}$$

The null space projectors are not commutative and the solution obtained by (5.11) may lead to conservative stability conditions. Since $\mathbf{P_1}...\mathbf{P_{n-1}}\mathbf{e_k}$ is not in the null space of $\mathbf{e_2}$,

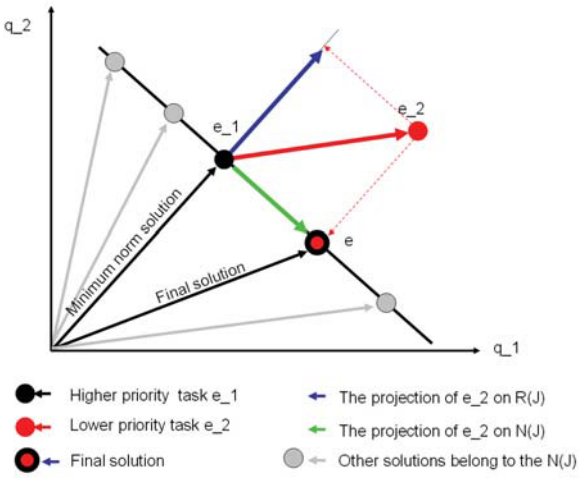


Figure 5.1 – Two priority level architecture

...,$\mathbf{e_{n-1}}$, the low level priority task $\mathbf{e_n}$ may modify tasks $\mathbf{e_2}$, ...,$\mathbf{e_{n-1}}$ of higher level priority. This stability problem can be solved by the augmented or best the intersection methods [Antonelli 09].

Augmented inverse-based projections

In this approach, the generic task is projected onto the null space of the task achieved by considering the augmented Jacobian of all the higher priority ones. By assuming that $\mathbf{e_{1...n}}$ is a task that realize the n first tasks while respecting their priorities, the orthogonal projection operator $\mathbf{P_{1..n}}$ is given by [Baerlocher 98]:

$$\mathbf{P_{1..n}} = \mathbf{I} - \mathbf{J^{+}_{1..n}}\mathbf{J_{1..n}} \tag{5.12}$$

where $\mathbf{J_{1..n}}$ is the Jacobian of $\mathbf{e_{1..n}}$ defined by:

$$\mathbf{J_{1..n}} = \frac{\partial \mathbf{e_{1..k}}}{\partial \mathbf{q}} = \frac{\partial(\mathbf{e_1} + \mathbf{P_1}.\mathbf{e_2} + \ldots + \mathbf{P_{1..n}}\mathbf{e_n})}{\partial \mathbf{q}} \tag{5.13}$$

The approximation value $\mathbf{J_{1..n}} = \mathbf{J_1} + \mathbf{P_1}.\mathbf{J_2} + \ldots + \mathbf{P_{1..n}}\mathbf{J_n}$ is used to compute $\mathbf{P_{1..n}}$ by assuming that $\frac{\partial \mathbf{P_{1..n}}}{\partial \mathbf{q}} = \mathbf{0}$ since it is difficult to compute. This approximation however does not guarantee that the error of the first n tasks remains $\mathbf{0}$ when the control due to $\mathbf{e_{n+1}}$ is high.

Intersection inverse-based projections method

To ensure that the motion will be projected if it is in the null space of all n tasks, the new task $\mathbf{e_{n+1}}$ is projected onto the space $\mathbf{N_{i...n}}$ defined as the intersection of the null spaces for

all n previous tasks . $\mathbf{N_{i...n}}$ is given by:

$$\mathbf{N_{i...n}} = \bigcap_{k=1}^{n} \text{Null}(\mathbf{J_k}) \tag{5.14}$$

The null space can be computed by stacking the Jacobian of the n tasks such that:

$$\mathbf{N_{i...n}} = \text{Null} \begin{pmatrix} \mathbf{J_1} \\ \vdots \\ \mathbf{J_n} \end{pmatrix} \tag{5.15}$$

5.3.2 Transpose-based projection methods (successive, augmented)

In [Chiacchio 91], instead of using inverse kinematics schemes that use the Jacobian pseudo inverse, a transpose-based task-priority redundancy resolution has been proposed for the two tasks in which Jacobian transpose is used such that:

$$\dot{\mathbf{q}} = \mathbf{J}^{\top}\dot{\mathbf{e}}^{*} + \mathbf{P}\mathbf{J}_{\mathbf{2}}^{\top}\mathbf{e_2} \tag{5.16}$$

Successive transpose-based projections

In [Antonelli 09], transpose projection approach is generalized by following the same guidelines previously mentioned for the inverse-based projection approaches, i.e., by projecting successively the lower priority tasks onto the null space of higher priority task:

$$\mathbf{e} = \mathbf{J}_{\mathbf{1}}^{\top}\dot{\mathbf{e}}_{\mathbf{1}} + \mathbf{P_1}.\mathbf{J}_{\mathbf{2}}^{\top}\dot{\mathbf{e}}_{\mathbf{2}} + \mathbf{P_1}\mathbf{P_2}.\mathbf{J}_{\mathbf{3}}^{\top}\dot{\mathbf{e}}_{\mathbf{3}} + \ldots \prod_{i=1}^{n-1}.\mathbf{P_i}\mathbf{J}_{\mathbf{n}}^{\top}\dot{\mathbf{e}}_{\mathbf{n}} \tag{5.17}$$

Augmented transpose-based projections

The lower priority task is projected onto the null space of the augmented Jacobian obtained by stacking all the higher priority tasks while the Jacobian transpose is used, [Antonelli 09]:

$$\mathbf{e} = \mathbf{J}_{\mathbf{1}}^{\top}\dot{\mathbf{e}}_{\mathbf{1}} + \mathbf{P_1}.\mathbf{J}_{\mathbf{2}}^{\top}\dot{\mathbf{e}}_{\mathbf{2}} + \mathbf{P}_{1\cdots 2}.\mathbf{J}_{\mathbf{3}}^{\top}\dot{\mathbf{e}}_{\mathbf{3}} + \mathbf{P_{1\cdots n}}\mathbf{J}_{\mathbf{n}}^{\top}\dot{\mathbf{e}}_{\mathbf{n}} \tag{5.18}$$

5.3.3 Unified formula for redundancy

In [Antonelli 09], four priority-based approaches have been grouped in an unified formula, and their stability analysis has been discussed. For n tasks this unified formula is written as:

$$\mathbf{e} = \mathbf{J}_{\mathbf{1}}^{\#}\dot{\mathbf{e}}_{\mathbf{1}} + \bar{\mathbf{P}}_{\mathbf{1}}.\mathbf{J}_{\mathbf{2}}^{\#}\dot{\mathbf{e}}_{\mathbf{2}} + \bar{\mathbf{P}}_{\mathbf{2}}.\mathbf{J}_{\mathbf{3}}^{\#}\dot{\mathbf{e}}_{\mathbf{3}} + \bar{\mathbf{P}}_{\mathbf{n}}\mathbf{J}_{\mathbf{n}}^{\#}\dot{\mathbf{e}}_{\mathbf{n}} \tag{5.19}$$

where for the successive projection $\bar{\mathbf{P}}_{\mathbf{k}} = \mathbf{P_1}\mathbf{P_2}\cdots\mathbf{P_k}$, and for the augmented projection $\bar{\mathbf{P}}_{\mathbf{k}} = \mathbf{P_{1\cdots k}}$, while for transpose-based or for inverse-based method $(\#)$ is replaced by $(\top)$ or $(+)$ respectively.

5.3.4 Efficient task-priority using the classical operator $\mathbf{P_e}$

Using the classical approach as given by (5.2) to treat priority among several tasks may distort tasks of lower priority. In order to execute the lower priority tasks more efficiently, a nice approach has been proposed in [Siciliano 91] [Baerlocher 98]. We refer this approach as efficient redundancy-based task-priority.

5.3.4.1 Two tasks

When two tasks $\mathbf{e_1}$ and $\mathbf{e_2}$ are considered where $\mathbf{e_i} \in \mathbb{R}^{m_i}, 1 = 1, 2$, $\mathbf{J_i} \in \mathbb{R}^{\mathbf{m_i \times n}}$ are their Jacobian such that $\dot{\mathbf{e}}_\mathbf{i} = \mathbf{J_i}\dot{\mathbf{q}}_\mathbf{i}$ and $\mathbf{e_1}$ has priority over $\mathbf{e_2}$. For $i = 1, 2$ we have:

$$\dot{\mathbf{q}}_\mathbf{i} = \mathbf{J}_\mathbf{i}^+ \mathbf{e_i} \tag{5.20}$$

When the robot is controlled using its articular velocity, the general solution of the primary task is

$$\dot{\mathbf{q}} = \mathbf{J}_\mathbf{1}^+ \dot{\mathbf{e}}_\mathbf{1}^* + \mathbf{P_1 g} \tag{5.21}$$

Using (5.21) and $\dot{\mathbf{e}}_\mathbf{2} = \mathbf{J_2}\dot{\mathbf{q}}$ we get:

$$\dot{\mathbf{e}}_\mathbf{2} = \mathbf{J_2 J}_\mathbf{1}^+ \dot{\mathbf{e}}_\mathbf{1}^* + \mathbf{J_2 P_1 g} \tag{5.22}$$

Solving (5.22) for $\mathbf{g}$ and injecting the computed $\mathbf{z}$ in (5.21) we get:

$$\dot{\mathbf{q}} = \mathbf{J}_\mathbf{1}^+ \dot{\mathbf{e}}_\mathbf{1}^* + \mathbf{P_1}(\mathbf{J_2 P_1})^+ (\dot{\mathbf{e}}_\mathbf{2}^* - \mathbf{J_2 J}_\mathbf{1}^+ \dot{\mathbf{e}}_\mathbf{1}^*) \tag{5.23}$$

Since $\mathbf{P_1}$ is idempotent and Hermitian then (5.23) becomes:

$$\dot{\mathbf{q}} = \mathbf{J}_\mathbf{1}^+ \dot{\mathbf{e}}_\mathbf{1}^* + (\mathbf{J_2 P_1})^+ (\dot{\mathbf{e}}_\mathbf{2}^* - \mathbf{J_2 J}_\mathbf{1}^+ \dot{\mathbf{e}}_\mathbf{1}^*) \tag{5.24}$$

Using (5.20) when $i = 1$, (5.24) is written as:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_\mathbf{1} + (\mathbf{J_2 P_1})^+ (\dot{\mathbf{e}}_\mathbf{2}^* - \mathbf{J_2}\dot{\mathbf{q}}_\mathbf{1}) \tag{5.25}$$

where $\mathbf{J_2 P_1}$ is the limited Jacobian of $\mathbf{e_2}$ and $(\dot{\mathbf{e}}_\mathbf{2}^* - \mathbf{J_2}\dot{\mathbf{q}}_\mathbf{1})$ is the secondary control component after removing the part of $\mathbf{e_2}$ already accomplished by $\mathbf{e_1}$.

5.3.4.2 Several tasks

Kinematics controller

Now, we consider l tasks. Let $\mathbf{e_1} \in \mathbb{R}^{\mathbf{m_1}}, \cdots, \mathbf{e_l} \in \mathbb{R}^{\mathbf{m_l}}$ be l tasks where the i^{th} task Jacobian $\mathbf{J_i} \in \mathbb{R}^{\mathbf{m_i \times n}}$ is defined by $\dot{\mathbf{e}}_\mathbf{i} = \mathbf{J_i}\dot{\mathbf{q}}$ where $\sum_{i=1}^{l} \text{rank}(\mathbf{J_i}) \leq n$. If the i^{th} task is to be executed with lower priority than the $(i-1)^{th}$ task, the general control scheme is given by $\dot{\mathbf{q}} = \dot{\mathbf{q}}_\mathbf{l}$ where $\dot{\mathbf{q}}_\mathbf{i}$ is recursively defined by, [Baerlocher 98]:

$$\dot{\mathbf{q}}_\mathbf{i} = \begin{cases} \dot{\mathbf{q}}_{\mathbf{i-1}} + \left(\mathbf{J}_i \mathbf{P}_{\mathbf{i-1}}^{\mathbf{A}}\right)^+ (\dot{\mathbf{e}}_\mathbf{i}^* - \mathbf{J_i}\dot{\mathbf{q}}_{\mathbf{i-1}}) & \text{if } i = 2, .., l \\ \mathbf{J}_\mathbf{1}^+ \dot{\mathbf{e}}_\mathbf{1}^*, & \text{if } i = 1 \end{cases} \tag{5.26}$$

where $\mathbf{P_i^A} = \mathbf{I}_n - \mathbf{J_i^{A^+}} \mathbf{J_i^A}$ is the orthogonal projection operator on the null space of the augmented Jacobian $\mathbf{J_i^A} = (\mathbf{J_1}, ..., \mathbf{J_i})$ and is recursively defined by:

$$\mathbf{J_i^A} = (\mathbf{J_{i-1}^A}, \mathbf{J_i}) \tag{5.27}$$

Using the generalized inverse of partitioned matrix we get [Cline 64] [Baerlocher 04]:

$$\mathbf{J_i^{A^+}} = \begin{bmatrix} (\mathbf{J_{i-1}^{A^+}} - \mathbf{T_i} \mathbf{J_i} \mathbf{J_{i-1}^{A^+}}) & \mathbf{T_i} \end{bmatrix} \tag{5.28}$$

where

$$\mathbf{T_i} = \widetilde{\mathbf{J}}_\mathbf{i}^+ + X(\mathbf{I_n} - \widetilde{\mathbf{J}}_\mathbf{i} \widetilde{\mathbf{J}}_\mathbf{i}^+) \tag{5.29}$$

such that

$$\widetilde{\mathbf{J}}_\mathbf{i} = \mathbf{J_i}(\mathbf{I_n} - \mathbf{J_{i-1}^{A^+}} \mathbf{J_{i-1}^A}) \tag{5.30}$$

and X is a complex term. Since $\mathbf{P}_i^\mathbf{A}$ is defined by

$$\mathbf{P_i^A} = \mathbf{I_n} - \mathbf{J_i^{A^+}} \mathbf{J_i^A} \tag{5.31}$$

By injecting (5.28) and (5.27) in (5.31)

$$\begin{aligned} \mathbf{P_i^A} &= \mathbf{I_n} - \begin{bmatrix} (\mathbf{J_{i-1}^{A^+}} - \mathbf{T_i} \mathbf{J_i} \mathbf{J_{i-1}^{A^+}}) & \mathbf{T_i} \end{bmatrix} (\mathbf{J_{i-1}^A}, \mathbf{J_i}) \\ &= \mathbf{I_n} - (\mathbf{J_{i-1}^{A^+}} \mathbf{J_{i-1}^A} - \mathbf{T_i} \mathbf{J_i} \mathbf{J_{i-1}^{A^+}} \mathbf{J_{i-1}^A} + \mathbf{T_i} \mathbf{J_i}) \\ &= \mathbf{I_n} - (\mathbf{J_{i-1}^{A^+}} \mathbf{J_{i-1}^A} + \mathbf{T_i} \ \ \mathbf{J_i}(\mathbf{I_n} - \mathbf{J_{i-1}^{A^+}} \mathbf{J_{i-1}^A}) \ \ ) \end{aligned} \tag{5.32}$$

By injecting (5.29) and (5.30) in (5.32) we get:

$$\begin{aligned} \mathbf{P_i^A} &= \mathbf{I_n} - (\mathbf{J_{i-1}^{A^+}} \mathbf{J_{i-1}^A} - (\widetilde{\mathbf{J}}_\mathbf{i}^+ + X(\mathbf{I_n} - \widetilde{\mathbf{J}}_\mathbf{i} \widetilde{\mathbf{J}}_\mathbf{i}^+)) \widetilde{\mathbf{J}}_\mathbf{i}) \\ &= (\mathbf{I_n} - \mathbf{J_{i-1}^{A^+}} \mathbf{J_{i-1}^A}) - \widetilde{\mathbf{J}}_\mathbf{i}^+ \widetilde{\mathbf{J}}_\mathbf{i} + X(\widetilde{\mathbf{J}}_\mathbf{i} - \widetilde{\mathbf{J}}_\mathbf{i} \widetilde{\mathbf{J}}_\mathbf{i}^+ \widetilde{\mathbf{J}}_\mathbf{i}) \end{aligned} \tag{5.33}$$

By using the property ($\widetilde{\mathbf{J}}_\mathbf{i} = \widetilde{\mathbf{J}}_\mathbf{i} \widetilde{\mathbf{J}}_\mathbf{i}^+ \widetilde{\mathbf{J}}_\mathbf{i}$) of the pseudoinverse, we get:

$$\begin{aligned} \mathbf{P_i^A} &= (\mathbf{I_n} - \mathbf{J_{i-1}^{A^+}} \mathbf{J_{i-1}^A}) - \widetilde{\mathbf{J}}_\mathbf{i}^+ \widetilde{\mathbf{J}}_\mathbf{i} \\ &= \mathbf{P_{i-1}^A} - \widetilde{\mathbf{J}}_\mathbf{i}^+ \widetilde{\mathbf{J}}_\mathbf{i} \end{aligned} \tag{5.34}$$

The recursive formula of the augmented classical projection operator is thus written as:

$$\mathbf{P_i^A} = \begin{cases} \mathbf{P_{i-1}^A} - \left(\mathbf{J_i} \mathbf{P_{i-1}^A}\right)^+ \left(\mathbf{J}_i \mathbf{P_{i-1}^A}\right) & \text{if } i = 2, .., l \\ \mathbf{I} - \mathbf{J_1^+} \mathbf{J_1}, & \text{if } i = 1 \end{cases} \tag{5.35}$$

If the classical projection operator is used to manage several tasks, the higher level priority tasks $\mathbf{e_1}, ..., \mathbf{e_{i-1}}$ should leave some DOFs to the lower level priority tasks $\mathbf{e_i}, ..., \mathbf{e_l}$. Usually, this is ensured by selecting $\mathbf{e_1}, ..., \mathbf{e_{i-1}}$ such that $Rank(\mathbf{J}_{i-1}^A) < n$.

Dynamic controller

In [Siciliano 91], the recursive form is expressed in terms of joint acceleration. Instead of deriving joint acceleration solutions from the second-order differential kinematics and then substituting in the robot dynamic model for designing a computed torque control, which may lead to internal unstable behavior for a redundant manipulator, [Nakamura 87], it is derived from the joint velocity solution by direct differentiation with respect to time, [KaBerounian 88] [Siciliano 91]. By taking the derivative of (5.26) we get:

$$\ddot{\mathbf{q}}_{\mathbf{i}} = \begin{cases} \ddot{\mathbf{q}}_{\mathbf{i-1}} + \bar{\mathbf{J}}_{\mathbf{i}}^{+}\left(\ddot{\mathbf{e}}_{\mathbf{i}}^{*} - \dot{\mathbf{J}}_{\mathbf{i}}\dot{\mathbf{q}}_{\mathbf{i-1}} - \mathbf{J}_{\mathbf{i}}\ddot{\mathbf{q}}_{\mathbf{i-1}}\right) + \dot{\bar{\mathbf{J}}}_{\mathbf{i}}^{+}\left(\dot{\mathbf{e}}_{\mathbf{i}}^{*} - \mathbf{J}_{\mathbf{i}}\dot{\mathbf{q}}_{\mathbf{i-1}}\right) & \text{if } i = 2,..,l \\ \mathbf{J}_{\mathbf{1}}^{+}\ddot{\mathbf{e}}_{\mathbf{1}}^{*} + \dot{\mathbf{J}}_{1}^{+}\dot{\mathbf{e}}_{\mathbf{1}}^{*}, & \text{if } i = 1 \end{cases} \tag{5.36}$$

where $\bar{\mathbf{J}} = \mathbf{J_i P_{i-1}^A}$ and $\dot{\bar{\mathbf{J}}}_{\mathbf{i}}^{+}$ is given by:

$$\dot{\bar{\mathbf{J}}}_{\mathbf{i}}^{+} = \bar{\mathbf{J}}_{\mathbf{i}}^{+}\dot{\bar{\mathbf{J}}}_{\mathbf{i}}\bar{\mathbf{J}}_{\mathbf{i}}^{+} + \left(\mathbf{I} - \bar{\mathbf{J}}_{\mathbf{i}}^{+}\bar{\mathbf{J}}_{\mathbf{i}}\right)\dot{\bar{\mathbf{J}}}_{\mathbf{i}}^{\top}\left(\bar{\mathbf{J}}_{\mathbf{i}}\bar{\mathbf{J}}_{\mathbf{i}}^{\top}\right)^{-1} \tag{5.37}$$

These frameworks based on equations (5.26) and (5.36) have been studied in [Siciliano 91] by simulations on snake-like robot with an obstacle avoidance constraint.

Handling singularities

In [Chiaverini 97], singular configurations in the framework of the redundancy-based task-priority resolution technique are addressed for two tasks. First singularity, the algorithmic singularity, appears in the task-priority formulation given by (5.26) that needs to compute an inverse of $\mathbf{J}_i\mathbf{P_{i-1}}$ which becomes singular when the secondary task cannot be achieved without preventing the primary task from being realized. Close to such singularities the priority levels may even be inverted. Algorithmic singularity is handled by defining the formulation of $\mathbf{q}$ as:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_{\mathbf{1}} + \mathbf{P_1}\mathbf{J}_2^{+}\dot{\mathbf{e}}_{\mathbf{2}} \tag{5.38}$$

Using (5.38), algorithmic singularities are decoupled from the singularities of $\mathbf{J_2}$. On one hand, when $\mathbf{J_2}$ is singular, it is replaced by its corresponding damped least square inverse ${\mathbf{J_2}}^{+\lambda_2}$ defined by:

$${\mathbf{J_2}}^{+\lambda_2} = {\mathbf{J_2}}^{\top}(\mathbf{J_2}{\mathbf{J_2}}^{\top} + \lambda_2^2\mathbf{I})^{-1} \tag{5.39}$$

where $\lambda_2 \in \mathbb{R}$ is the damping factor. On the other hand, when $\mathbf{J_1}$ has a kinematic singularity, ${\mathbf{J_1}}^{+}$ can be replaced by the damped least square inverse ${\mathbf{J_1}}^{+\lambda_1}$. This formulation helps to avoid the singularity, however, the tracking error for the secondary task increases to be higher than that of (5.25).

The kinematic singularity can also be be handled by introducing the damping factor λ_1 such that, [Chiaverini 97]:

$$\dot{\mathbf{q}} = \mathbf{J}_{\mathbf{1}}^{+\lambda_1}\dot{\mathbf{e}}_{\mathbf{1}}^{*} + (\mathbf{I} - \mathbf{J}^{+\lambda_1}\mathbf{J_1})\mathbf{J}_2^{+\lambda_2}\dot{\mathbf{e}}_{\mathbf{2}}^{*} \tag{5.40}$$

In [Baerlocher 04], the algorithmic singularity of the recursive formula (5.26) is handled by:

$$\dot{\mathbf{q}}_\mathbf{i} = \begin{cases} \dot{\mathbf{q}}_{\mathbf{i-1}} + \left(\mathbf{J}_i \mathbf{P}^{\mathbf{A}}_{\mathbf{i-1}}\right)^{+\lambda_i} \left(\dot{\mathbf{e}}^*_\mathbf{i} - \mathbf{J}_\mathbf{i}\dot{\mathbf{q}}_{\mathbf{i-1}}\right) & \text{if } i = 2, .., l \\ \mathbf{J}_1^+ \dot{\mathbf{e}}^*_\mathbf{1}, & \text{if } i = 1 \end{cases} \tag{5.41}$$

Finally, the expression that considers algorithmic and kinematic singularities corresponding to (5.25) is given by:

$$\dot{\mathbf{q}} = \mathbf{J}_\mathbf{1}^{+\lambda_1} \dot{\mathbf{e}}^*_\mathbf{1} + (\mathbf{J_2 P_1})^{\lambda_2} (\dot{\mathbf{e}_\mathbf{2}}^* - \mathbf{J_2} (\mathbf{J}_\mathbf{1}^{+\lambda_1} \mathbf{e}^*_\mathbf{1})) \tag{5.42}$$

However, additional algorithmic singularity appears when the low priority task can not be realized when projected onto the null space of $\mathbf{J_1}$.

5.3.5 Tasks priority and tasks sequencing

Both task priority and task sequencing have been validated by applying them to a unicycle-like mobile platform carrying on board a pan-tilt unit [DeLuca 08a]. Both redundancy resolution schemes show effectiveness against noise and unmodeled effects present in realistic conditions, as well as the ability to correctly execute regulation of ill-conditioned tasks. With the task priority method, it is possible to simultaneously regulate all the task variables. However, no additional requirements could be specified for the motion execution. On the other hand, the task sequencing approach allowed such a possibility thanks to the artificial redundancy introduced in the first phase, at the expense of a longer execution time for completing the original task.

In [Mansard 07], a global architecture is proposed to sequence a stack of tasks in order to reach a goal taking into account several environment constraints. Redundancy and stacking are involved in a proposed framework which consists in four controller layers. The first controller is composed of a stack which orders the active subtasks. The subtask at the bottom level of the stack has higher priority where the priority decreases as the stack level increases. The second controller ensures that enough DOF remain free to take the constraints into account and selects the optimal subtask to be removed from the stack. The third controller observes the removed subtasks from the stack and try to put them back in the stack as soon as possible.Finally, the forth controller ensures the convergence of the global system by solving the dead locks of the bottom controllers. Several sets of experiments are realized to validate this architecture. It have been shown that this approach is able to converge to the desired position despite various kinds of constraints.

5.4 Joint limit avoidance

Almost all robots including robot arms and humanoid robots have physical joint limits. Avoiding those limits while performing a task is one of the most important problems to

be solved by redundancy. In the following, we present the state of the art of joint limits avoidance strategies including potential field approaches [Khatib 85], the gradient projection method [Liegeois 77], joint position specifications and joint velocity specifications in task function approaches [Samson 91] [Marchand 96], redundancy-based iterative approach [Chaumette 01], manipulability measure approach [Nelson 95], weighted least norm solution approach [Chan 95], general weighted least norm solution approach [Xiang 10], constrained quadratic optimization methods [Ellekilde 07], improved damped least square approach [Na 08], fuzzy logic-based approach [Ahson 96], and neural network-based approach [Assal 06a] [Assal 06b]. In the following, we denote the value of the i^{th} joint j_i by q_i and the lower and upper limits for each joint j_i by $q_i^{\min}$ and $q_i^{\max}$ respectively.

5.4.1 Potential field approach

In [Khatib 85], artificial potential field is used to satisfy robot joint limits constraints. Each joint j_i can be kept within its minimum and maximum boundary q_i^{min} and q_i^{max} by defining barriers of potential and defining their corresponding potential forces by:

$$\Gamma_{q_i^{min}} = \begin{cases} \eta \left(\frac{1}{p_i^{min}} - \frac{1}{p_i^{min}(0)} \right) \frac{1}{p_i^{min\,2}}, & \text{if } p_i^{min} \leq p_i^{min}(0) \\ 0, & \text{else} \end{cases} \tag{5.43}$$

$$\Gamma_{q_i^{max}} = \begin{cases} -\eta \left(\frac{1}{p_i^{max}} - \frac{1}{p_i^{max}(0)} \right) \frac{1}{p_i^{max\,2}}, & \text{if } p_i^{max} \leq p_i^{max}(0) \\ 0, & \text{else} \end{cases} \tag{5.44}$$

where $p_i^{min}(0)$ and $p_i^{max}(0)$ represent the distance of the potential field influence and $p_i^{min} = q_i - q_i^{min}$ and $p_i^{max} = q_i^{max} - q_i$ are the distance limits. This potential field is integrated with the operational space approach.

5.4.2 Gradient projection method

The most widely used method to consider constraints or some criteria is the Gradient projection method (GPM). This method has been originally introduced for non-linear optimization [Rosen 60][Rosen 61], then applied in robotics [Liegeois 77] [Chang 87] [Nakamura 87] [Dubey 88] [Zghal 90] [Marchand 96] [Luya 01]. It requires defining a function representing the performance criterion and projecting the gradient of this function using the projection operator presented in Section 5.2. The self-motions generated allows the robot to perform the additional task as best as possible.

In GPM, the control law used to solve joint limits avoidance problem is usually given by:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_\mathbf{1} + \dot{\mathbf{q}}_\mathbf{a} = \dot{\mathbf{q}}_\mathbf{1} + \mathbf{P}_\mathbf{e} \frac{\partial h_s}{\partial \mathbf{q}} \tag{5.45}$$

where $\dot{\mathbf{q}}_\mathbf{a}$ is the articular velocity to perform the joint limits avoidance and h_s is a cost function designed to be minimal at safe configuration and maximal in the vicinity of the joint limits.

Joint position specification

In [Liegeois 77], an active utilization of redundancy of robot manipulator by gradient projection method is presented and the generalized inverse of the Jacobian matrix is used to present a general solution of joint velocity problem to keep the physical joint values within its allowable limits. The cost function used has a quadratic form (see Fig. 5.2) and is given by:

$$h_s = \frac{1}{6}\sum_{i=1}^{6}\left(\frac{q_i - q_i^{mean}}{q_i^{mean} - q_i^{max}}\right)^2 \tag{5.46}$$

where $q_i^{mean} = (q_i^{max} + q_i^{min})/2$. When the robot is not redundant, the desired end-effector task becomes impossible to accomplish in addition to the damage that may occur to the manipulators if the commanded robot joint angles exceed the joint limits.

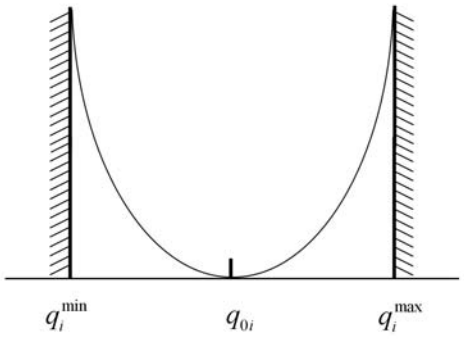


Figure 5.2 – Cost function, [Liegeois 77][Samson 91][Marchand 96]

In [Samson 91], the avoidance task aims to keep the joints of the manipulator at their optimal positions. The cost function is designed such that it is minimal when the manipulator reaches a specified desired joint position q_i^{mean}, and is maximal in the vicinity of joint limits. The robot moves the nearest of that position under the constraint that the visual task is realized.

$$h_s = \frac{1}{6}\sum_{i=1}^{n}(q_i - q_i^{mean})^2 \tag{5.47}$$

The cost functions defined by (5.46) and (5.47) are active as long as $(q_i - q_i^{mean}) \neq 0$, even if the joint is not in the vicinity of its limits as can be seen in Fig. 5.2.

In [Chaumette 01], to solve the problem mentioned above, the configuration $\mathbf{q}$ of the robot is considered safe with respect to its joint limits if for all joints j_i, $q_i \in [\, q_{\ell_0 i}^{\min}, q_{\ell_0 i}^{\max} \,]$ (see Fig. 5.3), where :

$$\begin{aligned} q_{\ell_0 i}^{\min} &= q_i^{\min} + \rho\Delta q_i \\ q_{\ell_0 i}^{\max} &= q_i^{\max} - \rho\Delta q_i \end{aligned} \tag{5.48}$$

define the safe domain of articulation j_i with $\rho \in [0, \frac{1}{2}]$ is a tuning parameter (typically $\rho = 0.1$) and $\Delta q_i = q_i^{\max} - q_i^{\min}$. It is generally defined by:

$$h_s = \frac{\beta}{2} \sum_{i=1}^{n} \frac{{\Delta_i}^2}{\Delta q_i} \tag{5.49}$$

where

$$\Delta_i = \begin{cases} q_i - q_{\ell_0 i}^{\min}, & \text{if } q_i < q_{\ell_0 i}^{\min} \\ q_i - q_{\ell_0 i}^{\max}, & \text{if } q_{\ell_0 i}^{\max} < q_i \\ 0, & \text{else} \end{cases} \tag{5.50}$$

Therefore

$$g_{s_i} = \frac{\partial h_s}{\partial q_i} = \beta \frac{\Delta_i}{\Delta q_i} \tag{5.51}$$

The classical avoidance task is thus given by:

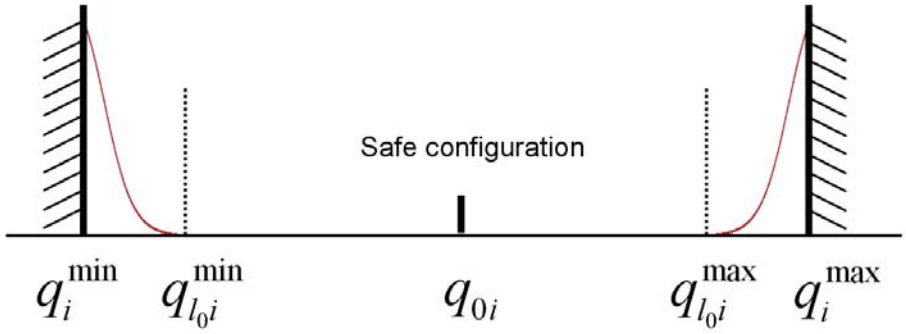


Figure 5.3 – Cost function, [Chaumette 01]

$$\dot{\mathbf{q}}_\mathbf{a} = \mathbf{P_e g} = \mathbf{P_e} \frac{\partial h_s}{\partial \mathbf{q}} = \beta \mathbf{P_e \Delta} \tag{5.52}$$

where $\mathbf{\Delta} = (\Delta_1/\Delta q_1, .., \Delta_n/\Delta q_n)$.

Joint velocities specification.

In [Marchand 96], the behavior of the manipulator is specified in term of velocities by introducing more constraints on the performance of the system to ensure an exponential decrease toward the final joint position q_i^{mean} such that:

$$\dot{\mathbf{q}}_\mathbf{i}^*(\mathbf{t}) = -\alpha \frac{q_i^*(t) - q_i^{mean}}{q_i^{max} - q_i^{min}} \tag{5.53}$$

where α is a constant value. The cost function defined for velocity specification is given by:

$$h_i = \frac{\beta}{2} \sum_{i=1}^{n} [(q_i(t) - q_i^*(t))]^2 \tag{5.54}$$

and therefore

$$g_{s_i} = \frac{\partial h_s}{\partial q_i} = \beta(q_i(t) - q_i^*(t)) \tag{5.55}$$

Experimental results showed that velocity specification method allows the robot to reach a final position closer to the middle of the joint limits faster than the position specification one.

Let us finally note that, in these methods, the tuning of β is extremely difficult. If too large, it results in oscillations. If too small, it can not allow avoiding the joint limits. Also, a suitable value for a given configuration may be either too large or too small for other configurations.

5.4.3 Redundancy-based iterative approach

In [Chaumette 01], a redundancy-based iterative approach is proposed to avoid the robot joint limits. This method does not affect the main task achievement and ensures the avoidance problem by automatically generating a robot motion compatible with the main task by iteratively solving a system of linear equations to cut any motion on the axis that are in a critical situation. In this method, if the robot system has n joints and the main task has m independent constraints then the global task is given by:

$$\mathbf{e} = \mathbf{J_1}^+ \mathbf{e}_1 \; + \; \sum_{i=1}^{n_a} a_i \, \mathbf{E}_{\bullet i} \tag{5.56}$$

where $n_a = \dim \text{Ker } \mathbf{J_1} = n - m$, $\sum_{i=1}^{n_a} a_i \, \mathbf{E_{\bullet i}}$ define the available motions to avoid the joint limit or to perform any other secondary task, $\mathbf{E}$ is a basis of Ker $\mathbf{J_1}$ of dimension $n \times n_a$. $\mathbf{a}$ is the vector of gain that will be automatically computed. The effect of this gain vector is to stop any motion of all joints that are in a critical situation. The global velocity vector obtained using (5.56) for the i^{th} joint is given by:

$$\dot{\mathbf{q}}[i] = -\lambda \left((\mathbf{J_1}^+ \mathbf{e_1})[i] + \sum_{k=1}^{n_a} a_k \, \mathbf{E}_{ik} \right) \tag{5.57}$$

In order to stop the motion of the robot joint j_i which nears its limits, $\dot{\mathbf{q}}[i]$ is set to be zero then we get:

$$(\dot{\mathbf{q}}[i] = 0) \Longleftrightarrow \left(\sum_{k=1}^{n_a} a_k \, \mathbf{E}_{ik} = -(\mathbf{J_1}^+ \mathbf{e_1})[i] \right) \tag{5.58}$$

If there are l joints near their limits then we need to ensure (5.58) for each of these joints. The common adaptive gain vector that works for all joints is automatically computed by solving the system of linear equations defined by:

$$\begin{bmatrix} \vdots \\ \mathbf{E}_{i\bullet} \\ \vdots \end{bmatrix} \mathbf{a} = \begin{bmatrix} \vdots \\ -(\mathbf{J_1}^+ \mathbf{e_1})[i] \\ \vdots \end{bmatrix} \equiv \mathbf{A}\mathbf{a} = \mathbf{d} \tag{5.59}$$

where $\mathbf{a}$ and $\mathbf{d}$ are vectors of dimension n_a and $\mathbf{A}$ is of dimension $n \times n_a$. The solution for (5.59) has the form $\mathbf{a}^* = \mathbf{A}^+\mathbf{d}$ and the control law is written as:

$$\dot{\mathbf{q}} = -\lambda \left(\mathbf{J_1}^+ \mathbf{e_1} + \sum_{k=1}^{n_a} a_k^* \mathbf{E}_{\bullet k} \right) = \dot{\mathbf{q}}_\mathbf{1} - \lambda \left(\sum_{k=1}^{n_a} a_k^* \mathbf{E}_{\bullet k} \right) \tag{5.60}$$

When the number of axis in critical situation n_b is less than n_a, other axes may enter in the critical situation. This problem is handled by choosing $\mathbf{a}^*$ as:

$$\mathbf{a}^* = \mathbf{A}^+\mathbf{d} + \sum_{k_2=1}^{n_b} \mathbf{b}_{k_2} \mathbf{B}_{\bullet k_2} \tag{5.61}$$

where $\mathbf{B} = \mathbf{I}_{n_a} - \mathbf{A}^+\mathbf{A}$ is the basis of Ker $\mathbf{A}$ and $n_b = \dim$ Ker $\mathbf{A}$. Let us note that $\sum_{k_2=1}^{n_b} \mathbf{b}_{k_2} \mathbf{B}_{\bullet k_2}$ has no effect on the main task. Finally, by injecting (5.61) in (5.60) we get:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_\mathbf{1} - \lambda \sum_{i=k}^{n_a} \left(\mathbf{A}^+\mathbf{d} + \sum_{k_2=1}^{n_b} \mathbf{b}_{k_2} \mathbf{B}_{\bullet k_2} \right)_k \mathbf{E}_{\bullet k} \tag{5.62}$$

This control scheme ensures that the motion of all joints near their limits will stop. However, a problem of discontinuity appears because each time a new joint has to be considered in the joint limit avoidance, a sudden stopping for this joint occurs. To deal with this discontinuity problem, the condition for $\dot{\mathbf{q}}[i]$ given in (5.58) is changed as:

$$\sum_{k=1}^{n_a} a_k \mathbf{E}_{ik} = -\gamma_i \left(\mathbf{J}^+ \mathbf{e} \right)[i] \tag{5.63}$$

where the coefficient γ_i is used to produce a smooth decay of the axis velocity and is given by:

$$\gamma_i = \begin{cases} \frac{q_{i(t+1)} - q_{\ell_0 i}^{\min}}{q_i^{\min} - q_{\ell_0 i}^{\min}}, & \text{if } q^{\min} \leq q_{i(t+1)} \leq q_{\ell_0 i}^{\min} \\ 0, & \text{if } q_{\ell_0 i}^{\min} < q_{i(t+1)} < q_{\ell_0 i}^{\max} \\ \frac{q_{i(t+1)} - q_{\ell_0 i}^{\max}}{q_i^{\max} - q_{\ell_0 i}^{\max}}, & \text{if } q_{\ell_0 i}^{\max} \leq q_{i(t+1)} \leq q^{\max} \end{cases} \tag{5.64}$$

The predicted value $\dot{\mathbf{q}}_{i(t+1)}$ is used for computing γ_i to ensure that the joint j_i will not reach any of its limits $q^{\min}$ or $q^{\max}$. In the implementation described in [Chaumette 01], it is proposed to reduce the range of each joint by 20%. Furthermore, as the control law is discontinuous, it also relies on the low level robot controller to smooth the resulting trajectory.

5.4.4 Manipulability measure approach

In [Nelson 95], a global objective function that realizes a compromise between the main task and secondary tasks is used by exploiting the robot redundant DOFs with respect to the main task. In this approach, a manipulability measure is determined to compare different manipulator configurations in order to indicate nearness to joint limits. This measure combines a

penalty function which approaches zero as soon as the joints approach their limits. It is given by:

$$w(\mathbf{q})' = \left(1 - e^{-k\Pi_{i=1}^{n} \frac{(q_i - q_i^{min})(q_i^{max} - q_i)}{(q_i^{max} - q_i^{min})}}\right) (\det(\mathbf{J}(\mathbf{q})))^2 \tag{5.65}$$

where k is a constant used to change the envelop of the penalty function. This approach was used to avoid kinematic singularity and joint limits in a target tracking system by introducing the manipulability measure into the visual tracking objective function to define a new objective function. However, important perturbations can be produced by the obtained motions, which are generally not compatible with the regulation to zero of the main task. Also, the global task can fail when the same joints are used for the avoidance and for achieving the main task.

5.4.5 Weighted least norm solution approach

In [Chan 95], weighted least-norm solution (WLN) to avoid joint limits for redundant joint manipulators is presented. Unlike the basic gradient projection method, WLN guarantees joint limit avoidance as well as minimizes unnecessary self-motion. The weighted norm of the joint velocity vector is defined by:

$$|\dot{\mathbf{q}}_\mathbf{W}| = \sqrt{\dot{\mathbf{q}}^\top \mathbf{W} \dot{\mathbf{q}}} \tag{5.66}$$

where $\mathbf{W} \in \mathbb{R}^{n \times n}$ is a symmetric and positive definite weighting matrix. If the relationship between the joint velocities $\dot{\mathbf{q}}$ and the end effector velocity $\mathbf{v}$ is $\mathbf{v} = \mathbf{J}\dot{\mathbf{q}}$, the weighted least square solution is given by:

$$\dot{\mathbf{q}}_\mathbf{W} = \mathbf{W}^{-1}\mathbf{J}^\top [\mathbf{J}\mathbf{W}^{-1}\mathbf{J}^\top]^{-1}\mathbf{v} \tag{5.67}$$

provided $\mathbf{J}$ is full ranked. To avoid joint limits, the weighted norm matrix $\mathbf{W}$ assumed to be diagonal matrix is given by $\mathbf{W} = \text{diag}(w_1, w_2, ..., w_n)$, where the i^{th} element w_i is defined as:

$$w_i = 1 + \left| \frac{\partial \mathbf{H}(\mathbf{q})}{\partial q_i} \right| \tag{5.68}$$

where $\mathbf{H}(\mathbf{q})$ is the performance criteria (cost function) and is defined as:

$$\mathbf{H}(\mathbf{q}) = \sum_{i=1}^{n} \frac{(q_i^{max} - q_i^{min})^2}{4(q_i^{max} - q_i)(q_i - q_i^{min})} \tag{5.69}$$

In order to determine wither the joint is moving away from or toward the limits, the weighting factors is redefined as follows:

$$w_i = \begin{cases} 1 + \left| \frac{\partial \mathbf{H}(\mathbf{q})}{\partial q_i} \right| & \text{if } \Delta \left| \frac{\partial \mathbf{H}(\mathbf{q})}{\partial q_i} \right| \geq 0 \\ 1 & \text{else} \end{cases} \tag{5.70}$$

In this case, if the joint is moving away from the limit, no penalization occurs to its motion and the joint moves freely. This allows the redundancy to be useful for other purposes such as avoiding obstacles. Finally, let us note that using WLN, the self-motion depends on the configuration and the end-effector velocity vector $\mathbf{v}$. The magnitudes of $\dot{\mathbf{q}}_{\mathbf{W}}$ changes according to the magnitude and the direction of the end-effector velocity vector q_i. When the end-effector is not moving, WLN scheme does not change its configuration.

5.4.6 General weighted least norm solution approach

In [Xiang 10], a general version of the weighted least norm solution approach is presented. In this method, the performance criterion is selected to satisfy that $\mathbf{H_g}(\mathbf{q}) > \mathbf{h_g}$, where $\mathbf{H_g}(\mathbf{q})$ is the performance function and $\mathbf{h_g}$ is a predefined threshold. By regarding $\mathbf{H_g}(\mathbf{q})$ as a virtual joint variable, the solution is to make $\dot{\mathbf{H}}_\mathbf{g}(\mathbf{q}) = \mathbf{0}$ when the joint $\mathbf{H_g}(\mathbf{q}) = \mathbf{h_g}$. The velocity of the virtual joint variable becomes $\dot{\mathbf{q}}_v = \mathbf{T}(\mathbf{q})\dot{\mathbf{q}}$ where $\mathbf{T}(\mathbf{q}) = [\nabla\mathbf{H_g}(\mathbf{q})\ \ \mathbf{K_g}(\mathbf{q})]^\top$, $\nabla\mathbf{H_g}(\mathbf{q})$ being the gradient vector of $\mathbf{H_g}(\mathbf{q})$, and $\mathbf{K_g}(\mathbf{q})$ the orthogonal complementary matrix of $\nabla\mathbf{H_g}(\mathbf{q})$. The kinematic equation relating the end-effector velocity and virtual joint velocities is given as follows $\dot{\mathbf{x}} = \mathbf{J}_v\dot{\mathbf{q}}_v$ where $\mathbf{J}_v = \mathbf{J}\mathbf{T}^{-1}(\mathbf{q})$ is the Jacobian matrix with respect to the virtual joint variable. The velocity command of the virtual joints is given by:

$$\dot{\mathbf{q}}_v = \mathbf{W}_v^{-1}\mathbf{J}_v^\top[\mathbf{J}_v\mathbf{W}_v^{-1}\mathbf{J}_v^\top]^{-1}\dot{\mathbf{x}}_d \tag{5.71}$$

The joint-velocity command is then given by:

$$\dot{\mathbf{q}} = \mathbf{W}_T^{-1}\mathbf{J}^\top[\mathbf{J}\mathbf{W}_T^{-1}\mathbf{J}^\top]^{-1}\dot{\mathbf{x}}_d \tag{5.72}$$

where $\mathbf{W}_T = \mathbf{T}^{-1}(\mathbf{q})\mathbf{W}_v^{-1}\mathbf{T}^{-T}(q)$ and

$$\tilde{w}_{v1} = \begin{cases} 1, & \text{if } q_{v1} - h_g > \epsilon \text{ or } \dot{q}_{v1} > 0 \\ 0, & \text{if } q_{v1} - h_g \leq \epsilon \text{ or } \dot{q}_{v1} \leq 0 \end{cases} \tag{5.73}$$

This method effectively meets the multiple constraints, whose number might be even larger than the number of joints while guaranteeing the good performance of the main task [Xiang 10]. However, a drawback of this method is the inability to avoid the singularity configurations because the transformation on the joint space will change the criterion for avoidance.

5.4.7 Constrained Newton optimization methods

In [Shahamiri 05], the control scheme proposed is based on constrained optimization using a null space-biased Newton technique. It is shown that, by formulating the control law as an optimization problem, singularities and obstacles can be detected online, and an avoidance command can be computed in the null space of the main task. Similar to the GPM, this approach exploits redundant degrees of freedom to execute a secondary task in the null space of the main task.

5.4.8 Constrained quadratic optimization method

In [Craig 86] [Cheng 92] [Cheng 94] [Ellekilde 07], the problem of joint limits avoidance is addressed using the quadratic optimization method. In this method, it is assumed that the value q_i of the joint j_i respects position, velocity and acceleration constraints defined by: $q_i^{\min} < q_i < q_i^{\max}$, $v_i^{\min} < \dot{q}_i < v_i^{\max}$, and $a_i^{\min} < \ddot{q}_i < a_i^{\max}$, where $v_i^{\min}$ and $v_i^{\max}$ are the minimum and maximum velocity and $a_i^{\min}$ and $a_i^{\max}$ are the minimum and maximum acceleration of the joint j_i respectively. To determine the feasible joint velocity $\dot{\mathbf{q}}$, the three constraints are expressed as constraints of $\dot{\mathbf{q}}$ as $\mathbf{A} < \dot{\mathbf{q}} < \mathbf{B}$. In order to consider joint limits when computing $\dot{\mathbf{q}}_\mathbf{i}$, each joint is treated individually by defining the lower and upper limits of its velocity. When the joint j_i is in an arbitrary position q_i^*, to ensure that a motion of j_i toward q_i^{max} can be stopped before actually reaching the limit, the velocity of the joint must be limited when arriving at q_i^*. This joint velocity upper bound is defined by the braking capabilities of the robot as well as the distance from the current joint position to the upper limit such that.

$$d_i = q_i^{max} - q_i^* \tag{5.74}$$

The worst case approximation of the distance traveled while braking with the maximal acceleration $\dot{q}_i^{min}$ is given by:

$$d_i = h(j+1)\dot{q}_i^{end} - h^2 a_i^{min} \frac{j(j+1)}{2} \tag{5.75}$$

where h is the cycle time. By setting $\dot{q}_i^{end} = 0$, and $\dot{q}_i^{end} = -ha_i^{min}$, the upper and lower bound for j can be obtained by solving (5.75) for j_{max} from which the integer between them can be defined as:

$$j(d_i) = \text{ROUND}\left(\frac{1}{2}\sqrt{1 - \frac{8d_i}{h^2 a_i^{min}}} - 1\right) \tag{5.76}$$

By injecting (5.76) in (5.75) we get:

$$\dot{q}_i^{end}(d_i) = \frac{d_i + h^2 a_i^{min} \frac{j(d_i)(j(d_i)+1)}{2}}{h(j(d_i)+1)} \tag{5.77}$$

The maximal velocity allowable at current time t is $\dot{Q}_i^{max}(d_i) = \dot{q}_i^{end}(d_i) - j(d_i)\ddot{q}_i^{min}h$. The worst-case estimate of the position at time $t+h$ is given by $d_i^* = d_i - h\dot{Q}_i^{max}(d_i)$, and the limit velocity needed at time t is $\dot{q}_i^{max}(d_i) = \dot{Q}_i^{max}(d_i^*) = \dot{Q}_i^{max}(d_i - h\dot{Q}_i^{max}(d_i))$. The same steps can be followed to determine the lower velocity limit when moving toward the lower joint limit. $\dot{q}_i^{min}(d_i) = \dot{Q}_i^{min}(d_i^*) = \dot{Q}_i^{min}(d_i - h\dot{Q}_i^{min}(d_i))$. At each control step, $\dot{q}_i^{min}(d_i)$ and $\dot{q}_i^{max}(d_i)$ are computed and then used in $\mathbf{A} < \dot{q} < \mathbf{B}$ in order to use the entire joint range.

5.4.9 Improved damped least square approach

In [Na 08], an improved damped least square solution is used for joint limits avoidance. In this method, to ensure that joint limits will not be violated and $q_i^{min} < q_i < q_i^{max}$, a diagonal

matrix $\mathbf{D}(\lambda)$ is introduced to replace the constant damping factor λ in the damped least square approach (remember equation (5.39)):

$$\mathbf{D}(\lambda) = \text{diag}(\lambda) = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \tag{5.78}$$

where

$$\lambda_i = c\left[\frac{2q_i - q_i^{max} - q_i^{min}}{q_i^{max} - q_i^{min}}\right]^p \tag{5.79}$$

where $c, p \in \mathbb{R}^+$ are constants and p is an even number.

When the joint value is within its motion range, small value of λ_i gives accurate solutions; and when the joint is near or moves away from its limits, large value of λ_i results in a feasible solution. Therefore, each λ_i makes a restriction to each joint value q_i. The larger the p is, the flatter the bottom of the curve of damping factor is, which implies the approximate low cost to the reasonable joint values.

5.4.10 Fuzzy logic-based approach

In [Ahson 96], a fuzzy logic-based method is presented to automatically choose an appropriate magnitude of a self-motion to avoid joint limits in kinematically redundant manipulators. For a given robot configuration, membership functions and linguistic rules are defined. A planner articulated arm having four DOFs is considered in simulation test where the limits of each joint are defined relative to its previous joint. The normalized cost function is defined as

$$h_i = \frac{(q_i - a_i)}{(a_i - q_i^{max})} \tag{5.80}$$

where

$$a_i = q_{i-1} + \frac{1}{2}(\Delta q_i^{min} + \Delta q_i^{max}) \tag{5.81}$$

is the average link position such that Δq_i^{min} and Δq_i^{max} are the joint limits with respect to the previous link. This method helps in choosing appropriate joint angles by selecting the right amount of self-motion using linguistic rules. The effectiveness of this method has been demonstrated by computer simulation.

5.4.11 Neural network-based approach

Classical methods for avoiding the joint limits using GPM do not guarantee a minimization of a certain performance criterion for each individual joint, particularly when the number of degrees of redundancy becomes less than the number of critical axes for a given task. In [Assal 06a] [Assal 06b], an intelligent control system is proposed which relies on exploiting the neural network as an advanced nonlinear controller to solve the kinematic inversion to

avoid the joint limits of a redundant manipulator. The fuzzy-neuro system works as a hint generator to generate an approximate solution $\mathbf{q_h} = (q_{h_1}, q_{h_2}, ...q_{h_n})$ of the inverse kinematic problem that presents joint angle vector for the next step of the manipulator. All elements of $\mathbf{q_h}$ are within the values required for the avoidance of joint angle limits. Two input vectors are presented to the fuzzy-neuro system to generate the hint vector $\mathbf{q_h}$, namely $\mathbf{q_{rel}} = (q_{r_1}, q_{r_2}, ..., q_{r_n}) \in \mathbb{R}^n$ and $\mathbf{v_r} = (v_{r_1}, v_{r_2}, ..., v_{r_n}) \in \mathbb{R}^n$, where q_{r_i} is the relative position of joint j_i to its limits given by:

$$q_{r_i} = \begin{cases} \frac{q_i(t)-q_i^m}{q_i^{\max}-q_i^m}, & \text{if } q_i > q_i^{\min} \\ -\frac{q_i(t)-q_i^m}{q_i^{\max}-q_i^m}, & \text{else} \end{cases} \tag{5.82}$$

where $q_i^m = \frac{q_i^{\max}+q_i^{\min}}{2}$, and v_{r_i} is the absolute relative velocity of joint j_i to its maximum velocity limit given by:

$$v_{r_i} = \frac{|v_i|}{v_{i_{max}}} \tag{5.83}$$

where $v_{i_{max}}$ is the maximum velocity of joint j_i and v_i is the velocity of joint j_i given by:

$$v_i = \begin{cases} \frac{q_i(t)-q_i(t-1)}{\Delta t}, & \text{if } q_i > q_i^{\min} \\ -\frac{q_i(t-1)-q_i(t)}{\Delta t}, & \text{else} \end{cases} \tag{5.84}$$

where Δt is the sampling time.

The general concept of constructing the fuzzy rules for each joint angle is to generate an approximate value for each joint angle from the current and previous ones for the next step according to the situation of the relative position of joint j_i as:

$$q_{h_i}(t+1) = q_i(t) + c(q_i(t) - q_i(t-1)) \tag{5.85}$$

where c is a constant.

The output of the neural network is guided by feeding an additional hint input vector to the NN, which is an approximate value vector for the required joint angle vector for the next step. Then, the neural network converges rapidly to the required joint angle vector that is close to the hint vector. In addition, it solves the redundancy resolution problem since the hint values are the values that can achieve the joint limits avoidance.

5.4.12 Comparison between different JLA methods

In [Ellekilde 07], a nice comparison is presented among different joint limits avoidance methods including iterative gradient projection method, weighted least norm method and quadratic optimization method. This comparison shows that, as the iterative GPM method reaches its critical area located at 0.8, it could be 0.95, in normalized joint coordinates it stops any motion of this joint toward the limit. However, the chosen limit may be reached with

maximum speed using iterative GPM method. The WLN method does not completely block the joint, but simply increases the cost of moving toward its limit causing the joint approach the limit slower than when the robot is controlled using the previous iterative GPM method. The characteristic behaviors of the QP controller, the WLN and the iterative GPM methods are similar, however, the QP controller exploits the entire joint range without attempting to break the limit while the basic GPM controller stops any motion of this joint toward the limit as soon as the critical limit is reached.

The advantage of projecting the gradient of the secondary task onto the null space of the Jacobian used for the main task is that the joint limit process have no influence on the main task and hence is executed under the constraint that the main task is realized. Unfortunately, it turns out that the success of these methods depends on a parameter, which specifies the influence of the secondary task on the final robot command sequence [Chan 95] [Chaumette 01]. This parameter must be tuned very precisely to ensure the effectiveness of the joint limit process. If this parameter is too small, the secondary task will not be able to influence the robot motions in time to avoid limits or singularities. If, on the other hand, the parameter is too large, the robot motions will be disturbed by the secondary task which results in undesired end-effector oscillations [Euler 89]. To make bad even worse, it turns out that a parameter suitable for a given robot configuration is often either too large or too small for other configurations. This effectively renders such basic gradient projection methods unsuitable for most applications. We will bring a significant contribution to this approach in the next chapter.

5.5 Conclusion

In this chapter, state of the art considering redundancy in robotic control as well as joint limits avoidance strategies were presented. The main advantage that redundant manipulators over non-redundant ones is the property of self-motion i.e. the ability to move joints without moving the end-effector. Self-motion makes redundant manipulators capable of optimizing various performance criteria in addition to the main task motion [Siciliano 91]. This is not possible in case of non-redundant manipulators since the joint configuration for any given end-effector position and orientation is unique except for finite variation. A summarization of our review of the state of the art is summarized by the two graphs illustrated in Fig. 5.4 and Fig. 5.5 for redundancy and joint limits avoidance respectively.

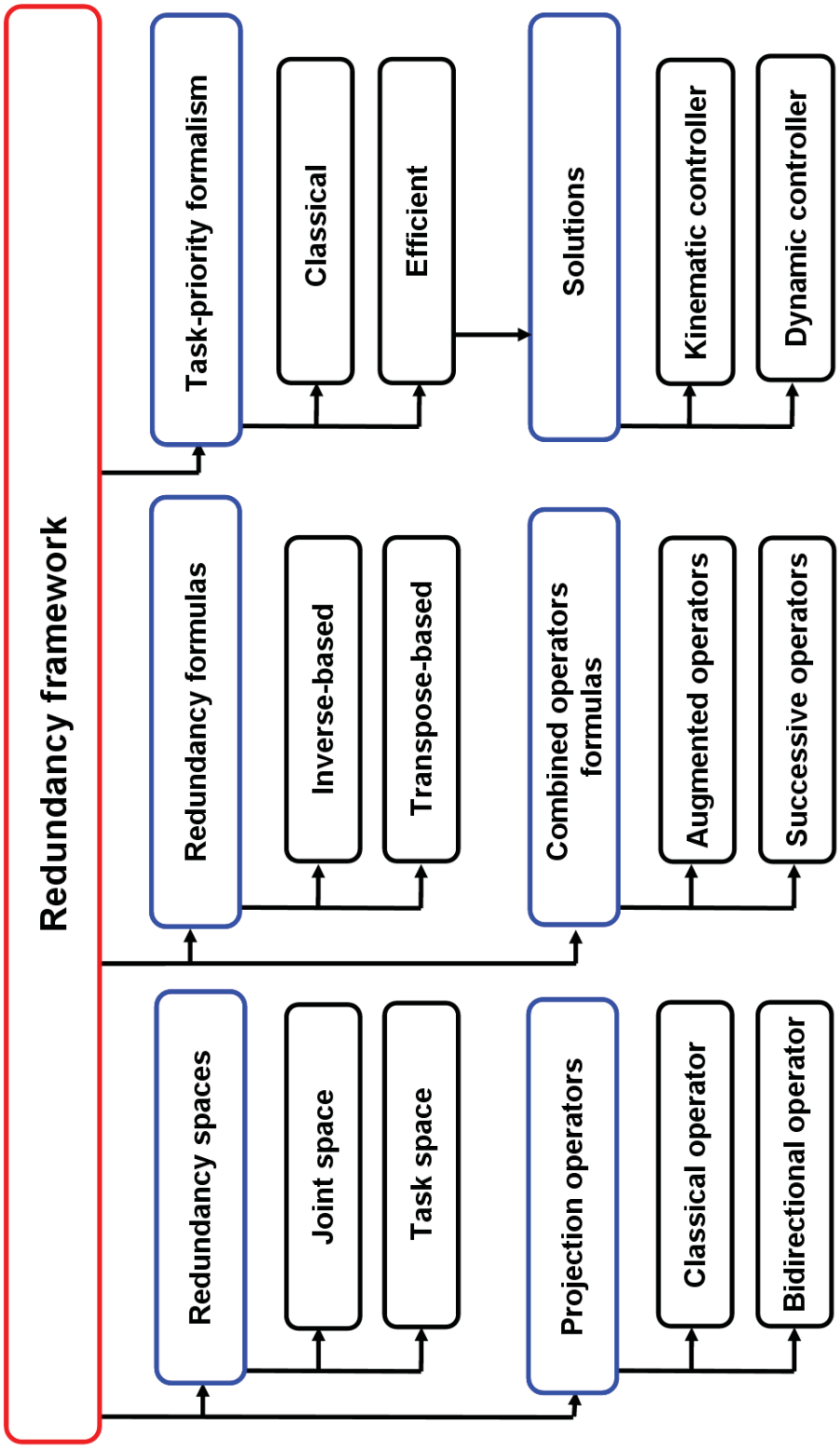


Figure 5.4 – Redundancy

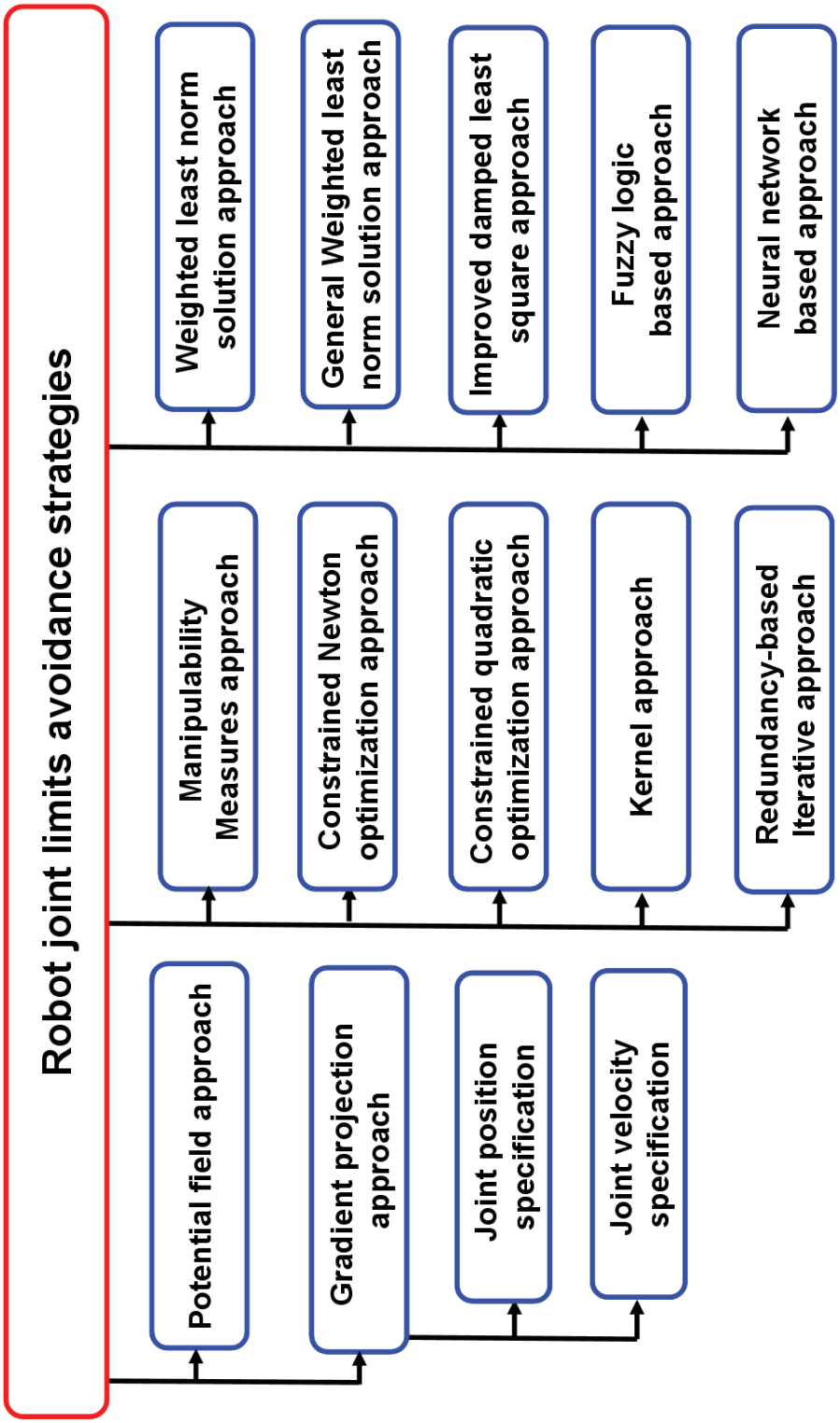


Figure 5.5 – Joint limits avoidance

Chapter 6

New large projection operator for the redundancy framework

In this chapter, we propose a new projection operator for the redundancy framework based on a task function defined as the norm of the usual error. This projection operator allows performing secondary tasks even when the main task is full rank. To ensure the convergence of the system, a switching strategy is then defined to switch from the new projection operator to the classical one before the norm of the total error reaches zero. An adaptive gain is also defined to slow down the convergence of the main task. It allows the secondary tasks to be active for longer. Treating priority among several tasks of different priority levels is also discussed and the recursive formula for the new projection operator is deduced. The experimental results obtained show the agreement with the analytical study and demonstrate the effectiveness of the proposed projection operator with respect to the classical one. The work described in this chapter leads to the following publication [Marey 10a].

6.1 Introduction

The methods presented in the previous chapter and based on the GPM approach require that the main task does not constrain all the robot DOFs. Indeed, in that case, the main task Jacobian becomes of full rank and no redundancy space is left for projecting any constraint. This is a limitation of the classical gradient projection method. That is why a proper selection of the projection operator is required to provide secondary motions of the manipulator that respect the constraints and keep the projected vector from being distorted as much as possible [Khatib 96], [Yoshikawa 96]. We will see that the method we propose leads to significant improvements.

This chapter is organized as follow: in Section 6.2, the new projection operator is devel-

oped and discussed. In Section 6.2.1, different test cases with respect to varying number of task components and robot DOFs are presented and analytically studied as $\mathbf{e}$ nears zero. In Section 6.2.2, a switching strategy is given to solve the problems exhibited in Sections 6.2 and 6.2.2. In Section 6.3, the recursive formula of the new projection operator is deduced for several task priorities levels and articular task priority is presented in Section 6.3.3. Finally, a description of the implementation of the new projection operator followed by experimental results in visual servoing are given in Section 6.4.

6.2 New projection operator $\mathbf{P}_{\|\mathbf{e}\|}$

The main original idea of this chapter is to consider $\eta = \|\mathbf{e}\|^\gamma$, where $\gamma \in \mathbb{R} - \{0\}$ to build a new projection operator $\mathbf{P}_\eta$. Since the error norm and the error vector are linked by $\|\mathbf{e}\|^2 = \mathbf{e}^\top\mathbf{e}$, we have $\|\mathbf{e}\|\dot{\|\mathbf{e}\|} = \mathbf{e}^\top\dot{\mathbf{e}}$ where $\dot{\|\mathbf{e}\|}$ is the variation of the norm of the total error $\|\mathbf{e}\|$, from which we obtain:

$$\dot{\eta} = \gamma\|\mathbf{e}\|^{\gamma-1}\dot{\|\mathbf{e}\|} = \gamma\|\mathbf{e}\|^{\gamma-2}\ \mathbf{e}^\top\dot{\mathbf{e}} \tag{6.1}$$

Since $\dot{\mathbf{e}} = \mathbf{J}_\mathbf{e}\dot{\mathbf{q}}$, we obtain by injecting $\dot{\mathbf{e}}$ in (6.1) :

$$\dot{\eta} = \gamma\|\mathbf{e}\|^{\gamma-2}\ \mathbf{e}^\top\mathbf{J}_\mathbf{e}\dot{\mathbf{q}} \tag{6.2}$$

from which we deduce:

$$\mathbf{J}_\eta = \gamma\|\mathbf{e}\|^{\gamma-2}\ \mathbf{e}^\top\mathbf{J}_\mathbf{e} \tag{6.3}$$

Note that $\mathbf{J}_\eta \in \mathbb{R}^{1\times n}$ is at most of rank 1. For all $\mathbf{e}$ such that $\|\mathbf{e}\| \neq \mathbf{0}$, we then obtain

$$\mathbf{J}_\eta^+ = \frac{1}{\gamma\|\mathbf{e}\|^{\gamma-2}\ (\mathbf{e}^\top\mathbf{J}_\mathbf{e}\mathbf{J}_\mathbf{e}^\top\mathbf{e})}\ \mathbf{J}_\mathbf{e}^\top\mathbf{e} \tag{6.4}$$

If we want η to have an exponential decrease, i.e. $\dot{\eta} = -\lambda\eta$, then the least square solution $\dot{\mathbf{q}}_\eta$ of $\mathbf{J}_\eta\ \dot{\mathbf{q}}_\eta = -\lambda\ \eta$ is given by:

$$\dot{\mathbf{q}}_\eta = -\lambda\ \|\mathbf{e}\|^\gamma\ \mathbf{J}_\eta^+ \tag{6.5}$$

and the general control law will be:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_\eta + \dot{\mathbf{q}}_\eta^\perp = \dot{\mathbf{q}}_\eta + \mathbf{P}_\eta\mathbf{g} \tag{6.6}$$

where $\mathbf{P}_\eta = (\mathbf{I}_n - \mathbf{J}_\eta{}^+\mathbf{J}_\eta)$ is a projection operator on the null space of $\mathbf{J}_\eta$ and $\mathbf{g}$ is any vector that can be designed to try to realize secondary tasks. Using (6.3) and (6.4), we directly get:

$$\mathbf{P}_\eta = \mathbf{P}_{\|\mathbf{e}\|} = \mathbf{I}_n - \frac{1}{\mathbf{e}^\top\mathbf{J}_\mathbf{e}\mathbf{J}_\mathbf{e}^\top\mathbf{e}}\ \mathbf{J}_\mathbf{e}^\top\mathbf{e}\mathbf{e}^\top\mathbf{J}_\mathbf{e} \tag{6.7}$$

We can note that $\mathbf{P}_\eta$ does not depend of the value of η. It is thus denoted $\mathbf{P}_{\|\mathbf{e}\|}$ in the following. Since $\mathbf{J}_\eta$ is at most of rank 1, $\mathbf{P}_{\|\mathbf{e}\|}$ is at least of rank $n-1$, which will thus not filter a lot the secondary task $\mathbf{g}$. That is the main contribution of this work especially if we remember that, in the classical approach, the rank of $\mathbf{P}_\mathbf{e}$ is equal to $n-r$. As soon as $r > 1$, supplementary directions of motions are thus available to achieve the secondary tasks. That is particularly true when $\mathbf{J}_\mathbf{e}$ is of full rank n, in which case $\mathbf{P}_\mathbf{e} = \mathbf{0}$ and no secondary task at all can be considered in that usual case.

6.2.1 Analytical study of $\mathbf{P}_{\|\mathbf{e}\|}$

This section presents an analytical analysis of the projection operator $\mathbf{P}_{\|\mathbf{e}\|}$ given by (6.7). On one hand, we can note that, as $\mathbf{e} \to \mathbf{0}$, the value of $\mathbf{P}_{\|\mathbf{e}\|}$ is unstable since the denominator tends to zero. On the other hand, we would like that $\mathbf{P}_{\|\mathbf{e}\|}$ tends to $\mathbf{P}_{\mathbf{e}}$ when $\mathbf{e} \to \mathbf{0}$. Indeed, when $\mathbf{e} \to \mathbf{0}$, no perturbation has to be introduced by the secondary tasks on each component of $\mathbf{e}$ to preserve its convergence and stability. In the following, we illustrate this point by two examples.

Case when k=2 and n=2

If we consider a system with two DOFs and a task with two components $\mathbf{e} = (x, y)$ then $\mathbf{J}_{\mathbf{e}} \in \mathbb{R}^{2\times 2}$. By assuming $\mathbf{J}_{\mathbf{e}}$ is full rank and given by:

$$\mathbf{J}_{\mathbf{e}} = \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} \tag{6.8}$$

we get:

$$\mathbf{J}_{\mathbf{e}}^{\top}\mathbf{e}\mathbf{e}^{\top}\mathbf{J}_{\mathbf{e}} = \begin{bmatrix} X^2 & XY \\ XY & Y^2 \end{bmatrix} \tag{6.9}$$

and

$$\mathbf{e}^{\top}\mathbf{J}_{\mathbf{e}}\mathbf{J}_{\mathbf{e}}^{\top}\mathbf{e} = X^2 + Y^2 \tag{6.10}$$

where $X = (a_1 x + b_1 y)$ and $Y = (a_2 x + b_2 y)$.

By injecting (6.9) and (6.10) in (6.7) and assuming for simplicity that $x = y$ then taking the limit as $\mathbf{e} \to \mathbf{0}$ for the upper left entry of $\mathbf{P}_{\|\mathbf{e}\|}$ we get:

$$\lim_{\mathbf{e}\to\mathbf{0}} \mathbf{P}_{\|\mathbf{e}\|}[1,1] = 1 - \frac{(a_1 + b_1)^2}{(a_1 + b_1)^2 + (a_2 + b_2)^2} \neq 0 \tag{6.11}$$

while we have $\mathbf{P}_{\mathbf{e}} = \mathbf{I}_n - \mathbf{J}_{\mathbf{e}}^{-1}\mathbf{J}_{\mathbf{e}} = \mathbf{0}$, which implies $\lim_{\mathbf{e}\to\mathbf{0}} \mathbf{P}_{\|\mathbf{e}\|} \neq \mathbf{P}_{\mathbf{e}}$. Similarly, the same result can be obtained for a higher DOFs system when the number of features is equal to the number of the robot DOFs and the task Jacobian is full rank.

Case when k=2 and n=3

If a system of three DOFs is considered with the same task $\mathbf{e} = (x, y)$, whose task Jacobian $\mathbf{J}_{\mathbf{e}} \in \mathbb{R}^{2\times 3}$ is now given by:

$$\mathbf{J}_{\mathbf{e}} = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{bmatrix} \tag{6.12}$$

then we get:

$$\mathbf{J}_{\mathbf{e}}^{\top}\mathbf{e}\mathbf{e}^{\top}\mathbf{J}_{\mathbf{e}} = \begin{bmatrix} X^2 & XY & XZ \\ XY & Y^2 & YZ \\ XZ & YZ & Z^2 \end{bmatrix} \tag{6.13}$$

and

$$\mathbf{e}^\top \mathbf{J_e} \mathbf{J_e}^\top \mathbf{e} = x^2\ A + 2xy\ C + y^2\ B \tag{6.14}$$

where $X = a_1 x + b_1 y$, $Y = a_2 x + b_2 y$, $Z = a_3 x + b_3 y$, $A = \sum_{i=1}^{3} a_i^2$, $B = \sum_{i=1}^{3} b_i^2$, $C = \sum_{i=1}^{3} a_i b_i$.
Injecting (6.13) and (6.14) in (6.7) we get:

$$\mathbf{P}_{\|\mathbf{e}\|} = \begin{bmatrix} 1 - \frac{X^2}{D} & \frac{XY}{D} & \frac{XZ}{D} \\ \frac{XY}{D} & 1 - \frac{Y^2}{D} & \frac{YZ}{D} \\ \frac{XZ}{D} & \frac{YZ}{D} & 1 - \frac{Z^2}{D} \end{bmatrix} \tag{6.15}$$

where $D = \mathbf{e}^\top \mathbf{J_e} \mathbf{J_e}^\top \mathbf{e}$. By considering again the particular case $x = y$ and taking the limit of $\mathbf{P}_{\|\mathbf{e}\|}$ when $\mathbf{e} \to \mathbf{0}$, then multiplying the first column of the result by $\mathbf{J_e}$ we get:

$$\begin{aligned} \mathbf{J_e} \lim_{\mathbf{e}\to\mathbf{0}} \mathbf{P}_{\|\mathbf{e}\|}[1, 1..3] &= \begin{bmatrix} a_1 - \frac{a_1 X_0^2}{D_0} + \frac{a_2 X_0 Y_0}{D_0} + \frac{a_3 X_0 Z_0}{D_0} \\ b_1 - \frac{b_1 X_0^2}{D_0} + \frac{b_2 X_0 Y_0}{D_0} + \frac{b_3 X_0 Z_0}{D_0} \end{bmatrix} \\ &\neq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{aligned} \tag{6.16}$$

where $D_0 = \sum_{i=1}^{3} (a_i^2 + a_i b_i + b_i^2)$, $X = a_1 + b_1$, $Y = a_2 + b_2$, $Z = a_3 + b_3$. This result shows that once again $\lim_{\mathbf{e}\to\mathbf{0}} \mathbf{P}_{\|\mathbf{e}\|} \neq \mathbf{P_e}$ since we have of course $\mathbf{J_e P_e} = \mathbf{0}$.

The previous study directs us to the following conclusion: as soon as the system nears its goal (that is when $\mathbf{e} \to \mathbf{0}$) we have to switch $\mathbf{P}_{\|\mathbf{e}\|}$ to the classical projection operator $\mathbf{P_e}$. This switching will ensure the convergence of the system since it allows solving the instability problem of $\mathbf{P}_{\|\mathbf{e}\|}$ as $\mathbf{e} \to \mathbf{0}$ due to the singularity of $\mathbf{J}_{\|\mathbf{e}\|}$ when $\mathbf{e} = \mathbf{0}$.

6.2.2 Switching based projection operator

The switching strategy designed consists in defining a convex combination $\mathbf{P}_\lambda$ between the classical and the new projection operator such that:

$$\mathbf{P}_\lambda = \overline{\lambda}(\|\mathbf{e}\|)\ \ \mathbf{P}_{\|\mathbf{e}\|} + (1 - \overline{\lambda}(\|\mathbf{e}\|))\ \ \mathbf{P_e} \tag{6.17}$$

where the proposed formula for the switching function $\overline{\lambda}(\|\mathbf{e}\|) : \mathbb{R} \to [0, 1]$ is defined by:

$$\overline{\lambda}(\|\mathbf{e}\|) = \begin{cases} 1 & \text{if } e_1 < \|\mathbf{e}\| \\ \frac{\lambda(\|\mathbf{e}\|) - \lambda_0}{\lambda_1 - \lambda_0} & \text{if } e_0 \leq \|\mathbf{e}\| \leq e_1 \\ 0 & \text{if } \|\mathbf{e}\| < e_0 \end{cases} \tag{6.18}$$

where e_1 and e_0 are two threshold values that define the starting and the ending conditions for the switching period. $\lambda(t) : \mathbb{R} \to \mathbb{R}$ is a continuous monotonically increasing function, such that $\lambda_1 = \lambda(e_1) \approx 1$ and $\lambda_0 = \lambda(e_0) \approx 0$. The sigmoid function $\lambda(t) = \frac{1}{1+\exp(-t)}$ shows early exponential growth from zero for negative t, which slows to linear growth of slope 1/4

near $t = 0$, then approaches one with an exponentially decaying gap for positive t [Pearl 20]. A good selection for the function $\lambda(\|\mathbf{e}\|)$ is then the sigmoid function given as:

$$\lambda(\|\mathbf{e}\|) = \frac{1}{1 + \exp(-12\frac{\|\mathbf{e}\| - e_0}{e_1 - e_0} + 6)} \tag{6.19}$$

where values of e_0 and e_1 have to be selected such that the system does not converge too fast during the interval $[e_0, e_1]$. This allows the switching to be performed smoothly during a sufficient number of iterations. Figure (6.1) shows the shape of the switching function $\overline{\lambda}(\|\mathbf{e}\|)$ when $e_0 = 0.1$ and $e_1 = (0.3, 0.5, 0.7, 0.9)$.

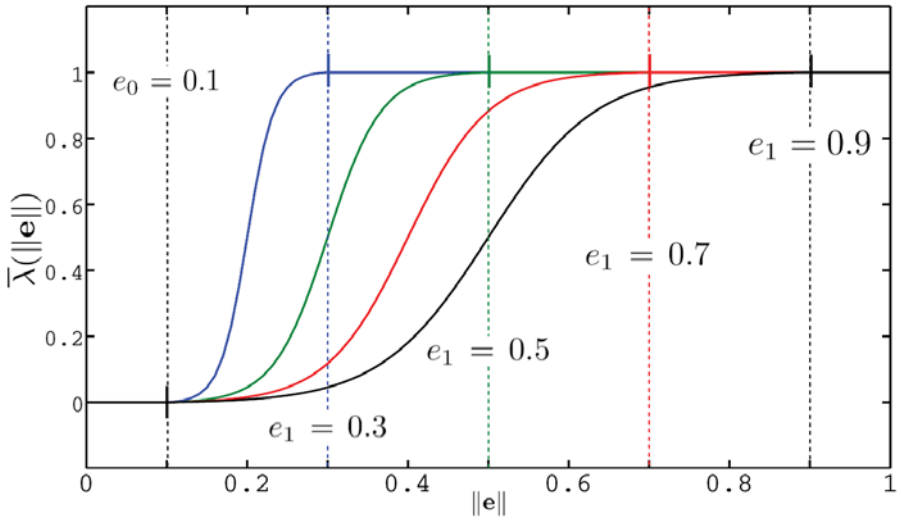


Figure 6.1 – Switching function $\overline{\lambda}(\|\mathbf{e}\|)$.

6.2.3 Stability analysis of $\dot{\mathbf{q}}_\eta$

Let us now consider $\dot{\mathbf{q}}_\eta$. After injecting (6.4) in (6.5) we obtain:

$$\dot{\mathbf{q}}_\eta = -\frac{\lambda}{\gamma}\frac{\|\mathbf{e}\|^2}{(\mathbf{e}^\top \mathbf{J}_\mathbf{e}\mathbf{J}_\mathbf{e}^\top \mathbf{e})}\ \mathbf{J}_\mathbf{e}^\top \mathbf{e} \tag{6.20}$$

Using control scheme $\dot{\mathbf{q}}_\eta$ given by (6.20), a singular configuration is obtained if $\mathbf{e} \in \text{Ker}(\mathbf{J}_\mathbf{e}^\top)$. That is clear from (6.3). This case corresponds to a local minimum of the classical control scheme $\dot{\mathbf{q}}_\mathbf{e} = -\lambda \mathbf{J}_\mathbf{e}^+ \mathbf{e}$ (see (5.1) and (5.2)), since $\text{Ker}(\mathbf{J}_\mathbf{e}^\top) = \text{Ker}(\mathbf{J}_\mathbf{e}^+)$. Another singularity occurs when $\mathbf{e} \to \mathbf{0}$ if the denominator $\mathbf{e}^\top \mathbf{J}_\mathbf{e}\mathbf{J}_\mathbf{e}^\top \mathbf{e}$ has a convergence rate to zero faster than that of the nominator. If the denominator and the nominator have the same convergence rate to zero when $\mathbf{e} \to \mathbf{0}$, then $\lim_{\mathbf{e}\to\mathbf{0}} \dot{\mathbf{q}}_\eta$ is indeterminate and the system will not be stable nor robust with respect to any perturbation. Of course, we obtain the same results by studying the stability analysis of the control scheme (6.20). Let us consider the candidate Lyapunov

function $V(t) = \eta^2$. By taking the derivative of $V(t)$ and injecting (6.2) in the result we obtain:

$$\dot{V}(t) = 2\eta\,\dot{\eta} = 2\gamma\,\|\mathbf{e}\|^{2\gamma-2}\mathbf{e}^\top\mathbf{J_e}\dot{\mathbf{q}} \tag{6.21}$$

Injecting (6.20) in (6.21), we get:

$$\begin{aligned}\dot{V}(t) &= \frac{-2\lambda\|\mathbf{e}\|^{2\gamma}}{\mathbf{e}^\top\mathbf{J_e}\mathbf{J_e}^\top\mathbf{e}}\,\mathbf{e}^\top\mathbf{J_e}\mathbf{J_e}^\top\mathbf{e} \\ &= -2\lambda\|\mathbf{e}\|^{2\gamma} \quad \text{when } \mathbf{J_e}^\top\mathbf{e} \neq 0\end{aligned} \tag{6.22}$$

We have $\dot{V}(t) < 0$ as soon as $\mathbf{e} \neq \mathbf{0}$ and $\mathbf{e} \notin \text{Ker}(\mathbf{J_e}^\top)$, thus ensuring the same asymptotic stability properties of the system as in the classical case [Chaumette 06], but when $\mathbf{e} = \mathbf{0}$. Let us note that these results are obtained whatever the value of γ, for example by considering the main task functions such as $\eta = \sqrt{\|\mathbf{e}\|}$ when $\gamma = 0.5$ or $\eta = \|\mathbf{e}\|^2$ when $\gamma = 2$. We will see how to deal with these problems of singularity and stability in Section 6.2.5.

6.2.4 Studying $\dot{\mathbf{q}}_\eta$ and $\dot{\mathbf{P}}_\eta$ as $\mathbf{e} \to \mathbf{0}$ ($t \to \infty$)

Now, we study the behavior of $\dot{\mathbf{q}}_\eta$ and $\dot{\mathbf{P}}_\eta$ when defined as a function of t, where t is the time parameter.

Finding η and $\mathbf{e}$ as a function of the time t

By injecting (6.20) in $\dot{\mathbf{e}} = \mathbf{J_e}\dot{\mathbf{q}}$ we get:

$$\dot{\mathbf{e}} = -\frac{\lambda}{\gamma}\frac{\|\mathbf{e}\|^2}{\mathbf{e}^\top\mathbf{J_e}\mathbf{J_e}^\top\mathbf{e}}\;\mathbf{J_e}\mathbf{J_e}^\top\mathbf{e}$$

If we assume that $\mathbf{J_e}\mathbf{J_e}^\top = \mathbf{I}_k$ (which is a strong assumption), then we get:

$$\dot{\mathbf{e}} = \begin{cases} -\frac{\lambda}{\gamma}\mathbf{e} & \text{if } \mathbf{e} \neq 0 \\ \mathbf{0} & \text{if } \mathbf{e} \to 0 \end{cases} \tag{6.23}$$

Solving this differential equation, we obtain:

$$\mathbf{e}(t) = \mathbf{e}(0)\exp(-\tfrac{\lambda}{\gamma}t) \tag{6.24}$$

from which we deduce:

$$\begin{aligned}&\eta(t) = \|\mathbf{e}(t)\|^\gamma = \|\mathbf{e}(0)\|^\gamma\exp(-\lambda t) \\ &\Rightarrow \quad \eta(t) \to 0 \;\text{ when }\; t \to 0\end{aligned} \tag{6.25}$$

Studying $\mathbf{v}_{\|\mathbf{e}\|}$ as $\mathbf{e} \to \mathbf{0}$ ($t \to \infty$)

Assuming again that $\mathbf{J_e}\mathbf{J_e}^\top = \mathbf{I}_k$, we obtain from (6.20):

$$\begin{aligned} \mathbf{v}_{\|\mathbf{e}\|} &= -\frac{\lambda}{\gamma}\frac{\|\mathbf{e}\|^2}{\|\mathbf{e}\|^2}\ \mathbf{J_e}^\top\mathbf{e} \\ &= -\frac{\lambda}{\gamma}\ \mathbf{J_e}^\top\mathbf{e} \quad \text{when } \mathbf{e} \neq \mathbf{0} \end{aligned} \tag{6.26}$$

Injecting (6.24) and (6.25) in (6.26) gives

$$\mathbf{v}_{\|\mathbf{e}\|}(t) = -\frac{\lambda}{\gamma}\exp(-\lambda t)\ \mathbf{J_e}^\top\mathbf{e}(0) \tag{6.27}$$

If $\mathbf{J_e}$ is constant, then as $t \to \infty$ we get:

$$\lim_{\mathbf{t}\to\infty} \mathbf{v}_{\|\mathbf{e}\|}(t) = \mathbf{0} \tag{6.28}$$

Studying $\mathbf{P}_{\|\mathbf{e}\|}$ as $\mathbf{e} \to \mathbf{0}$ ($t \to \infty$)

For the projection operator

$$\mathbf{P}_{\|\mathbf{e}\|} = \mathbf{I}_n - \frac{1}{\mathbf{e}^\top\mathbf{J_e}\mathbf{J_e}^\top\mathbf{e}}\ \mathbf{J_e}^\top\mathbf{e}\mathbf{e}^\top\mathbf{J_e} \tag{6.29}$$

If $\mathbf{J_e}\mathbf{J_e}^\top = \mathbf{I}_k$, then:

$$\mathbf{P}_{\|\mathbf{e}\|} = \mathbf{I}_n - \frac{1}{\|\mathbf{e}\|^2}\ \mathbf{J_e}^\top\mathbf{e}\mathbf{e}^\top\mathbf{J_e} \tag{6.30}$$

Writing $\mathbf{P}_{\|\mathbf{e}\|}$ as a function of t by injecting (6.24) and (6.25) in (6.30) and assuming $\mathbf{J_e}$ is constant, we get:

$$\mathbf{P}_{\|\mathbf{e}\|}(t) = \mathbf{I}_n - \frac{1}{\|\mathbf{e}(0)\|^2}\ \mathbf{J_e}^\top\mathbf{e}(0)\mathbf{e}^\top(0)\mathbf{J_e} \tag{6.31}$$

Since this expression does not depend of the time, we have of course:

$$\lim_{\mathbf{t}\to\infty} \mathbf{P}_{\|\mathbf{e}\|}(t) = \mathbf{I}_n - \frac{1}{\|\mathbf{e}(0)\|^2}\ \mathbf{J_e}^\top\mathbf{e}(0)\mathbf{e}^\top(0)\mathbf{J_e} \tag{6.32}$$

To study the limit of the classical projection operator $\mathbf{P_e} = \mathbf{I_n} - \mathbf{J_e}^+\mathbf{J_e}$ as $t \to \infty$, three cases arise. These cases are based on the number m of features and the number n of DOF of the system. For the first case when $m > n$ then $\mathbf{J_e}^+ = (\mathbf{J_e}^\top\mathbf{J_e})^{-1}\mathbf{J_e}^\top$ then we get:

$$\begin{aligned} \mathbf{P_e}(t) &= \mathbf{I}_n - \mathbf{J_e}^+\mathbf{J_e} \\ &= \mathbf{I}_n - (\mathbf{J_e}^\top\mathbf{J_e})^{-1}\mathbf{J_e}^\top\mathbf{J_e} \\ &= \mathbf{0} \end{aligned} \tag{6.33}$$

Similarly, for the second case when $m = n$ then $\mathbf{J}_\mathbf{e}^+ = \mathbf{J}_\mathbf{e}^{-1}$ and we have:

$$\begin{aligned} \mathbf{P}_\mathbf{e}(t) &= \mathbf{I}_n - \mathbf{J}_\mathbf{e}^{-1}\mathbf{J}_\mathbf{e} \\ &= \mathbf{0} \end{aligned} \tag{6.34}$$

Finally, for the third case when $m < n$ then $\mathbf{J}_\mathbf{e}^+ = \mathbf{J}_\mathbf{e}^\top(\mathbf{J}_\mathbf{e}\mathbf{J}_\mathbf{e}^\top)^{-1}$ and since we assumed $\mathbf{J}_\mathbf{e}\mathbf{J}_\mathbf{e}^\top = \mathbf{I}_k$ then we get:

$$\begin{aligned} \mathbf{P}_\mathbf{e}(t) &= \mathbf{I}_n - \mathbf{J}_\mathbf{e}^+\mathbf{J}_\mathbf{e} \\ &= \mathbf{I}_n - \mathbf{J}_\mathbf{e}^\top(\mathbf{J}_\mathbf{e}\mathbf{J}_\mathbf{e}^\top)^{-1}\mathbf{J}_\mathbf{e} \\ &= \mathbf{I}_n - \mathbf{J}_\mathbf{e}^\top\mathbf{J}_\mathbf{e} \end{aligned} \tag{6.35}$$

These results show that if $\mathbf{J}_\mathbf{e}\mathbf{J}_\mathbf{e}^\top = \mathbf{I}_k$ then the classical and the new projection operators may have non-zero values as $t \to \infty$ when $m < n$ which allows secondary tasks to be projected. If $m \geq n$ and $t \to \infty$ then $\mathbf{P}_\mathbf{e} = \mathbf{0}$. That is why the classical projection operator does not filter any secondary task as soon as the system nears the desired configuration. While for the new projection operator $\mathbf{P}_\mathbf{e} \neq \mathbf{0}$. This leads to the same conclusion obtained in 6.2.1, i.e., when the system approach the desired configuration, the new operator should switch to the classical one to ensure the convergence of the system.

6.2.5 Main task function

Possible control schemes

As discussed in Section 6.2.3, stability problem and singularity appear when $\mathbf{e} \to \mathbf{0}$ if the control scheme $\dot{\mathbf{q}}_\eta$ defined in (6.20) is used as output of the main task. These problems can be avoided by performing a switching from $\dot{\mathbf{q}}_\eta$ to the classical $\dot{\mathbf{q}}_\mathbf{e}$ given in (5.2) by defining as control scheme:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_\lambda + \mathbf{P}_\lambda \mathbf{g} \tag{6.36}$$

where

$$\dot{\mathbf{q}}_\lambda = \overline{\lambda}(\|\mathbf{e}\|)\ \ \dot{\mathbf{q}}_\eta + (1 - \overline{\lambda}(\|\mathbf{e}\|))\ \ \dot{\mathbf{q}}_\mathbf{e} \tag{6.37}$$

with $\overline{\lambda}(\|\mathbf{e}\|)$ given by (6.18) for which e_0 is selected such that as long as $\|\mathbf{e}\| > e_0$, the singularity effect of $\dot{\mathbf{q}}_{\|\mathbf{e}\|}$ does not appear. Applying this switching strategy ensures that the main task will avoid the singularity situation when $\mathbf{e} \to \mathbf{0}$ since

$$\lim_{\mathbf{e}\to\mathbf{0}} \dot{\mathbf{q}}_\lambda = \lim_{\mathbf{e}\to\mathbf{0}} \dot{\mathbf{q}}_\mathbf{e} = \mathbf{0} \tag{6.38}$$

In order to avoid the indetermination problem when the denominator of $\dot{\mathbf{q}}_{\|\mathbf{e}\|}$ is equal to zero (that is when $\mathbf{e} \in \text{Ker}(\mathbf{J}_\mathbf{e}^+)$, a direct switching without a transition interval from $\dot{\mathbf{q}}_{\|\mathbf{e}\|}$ to $\dot{\mathbf{q}}_e$ can be employed by setting $e_0 = e$, where e is the value of the error norm when the indetermination problem occurs.

Finally, the classical control scheme $\dot{\mathbf{q}}_\mathbf{e}$ can also be used directly with the new projection operator $\mathbf{P}_\lambda$ instead of $\dot{\mathbf{q}}_\eta$ or $\dot{\mathbf{q}}_\lambda$. In that case, the control scheme will be:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_\mathbf{e} + \mathbf{P}_\lambda \mathbf{g} \tag{6.39}$$

Starting the task with the classical control scheme ensures an exponential decreasing of each error component when $\mathbf{g} = \mathbf{0}$, which may be useful in practice as will be seen in Section 6.4.

Adaptive gain $\beta(\|\mathbf{e}\|)$

Usually, the gain λ involved in the classical control scheme $\dot{\mathbf{q}}_\mathbf{e} = -\lambda \mathbf{J}_\mathbf{e}^+ \mathbf{e}$ is tuned so that the convergence rate of the main task is as fast as possible while preserving the stability of the system. This leads to increase λ as $\mathbf{e}$ decreases (for instance could be an equation $\lambda(\|\mathbf{e}\|) = \lambda_{\min} + K \exp(-B\|\mathbf{e}\|)$ where $\lambda_{\min}$, K and B are constant positive scalar values). However, having a fast convergence rate for the main task may not be adequate in our case. Indeed, since $\mathbf{P}_{\|\mathbf{e}\|}$ switches to $\mathbf{P}_\mathbf{e}$ when $\mathbf{e} \to \mathbf{0}$, this switch may occur too early to have enough time to take the secondary tasks into account. This is especially the case when the secondary tasks are critical (such as obstacles and joints limits avoidance for example). That is why it may be useful to slow down the convergence rate of the main task. For that, the norm of the total error may be introduced in the control gain and we may think to use $\lambda = \lambda_0 \|\mathbf{e}\|$ where λ_0 is a constant. However, as soon as $\|\mathbf{e}\| \to 0$, the convergence rate of the main task becomes too slow. To avoid this problem, we define a gain function $\beta(\|\mathbf{e}\|)$ which returns the norm of the total error as long as $\mathbf{P}_\lambda = \mathbf{P}_{\|\mathbf{e}\|^2}$ and switches smoothly to 1 as soon as the norm of the error reaches a specified threshold value. The scheme of the adaptive gain function $\beta(\|\mathbf{e}\|)$ can be deduced and written as:

$$\beta(\|\mathbf{e}\|) = 1 - \overline{\lambda}(\|\mathbf{e}\|) + \|\mathbf{e}\| \ \overline{\lambda}(\|\mathbf{e}\|) \tag{6.40}$$

with the same switching conditions used to switch from $\mathbf{P}_{\|\mathbf{e}\|}$ to $\mathbf{P}_e$ used for $\beta(\|\mathbf{e}\|)$. Setting $\lambda = \lambda_0 \beta(\|\mathbf{e}\|)$, the control scheme (6.39) becomes:

$$\dot{\mathbf{q}} = -\lambda_0 \, \beta(\|\mathbf{e}\|) \, \mathbf{J}^+ \mathbf{e} + \mathbf{P}_\lambda \, \mathbf{g} \tag{6.41}$$

Using this control scheme increases the time during which the secondary tasks will be active thanks to the use of $\mathbf{P}_{\|\mathbf{e}\|}$.

Let us finally note that the new projection operator $\mathbf{P}_\lambda$ does not modify the stability properties of the system. More precisely, if the control schemes (6.36), (6.39) and (6.41) are globally or locally stable in the Lyapunov sense using $\mathbf{P}_\mathbf{e}$, they are also with $\mathbf{P}_\lambda$. This is due to the fact that, by construction, the variation of the Lyapunov function $\eta^2 = \|\mathbf{e}\|^2$ does not depend of $\mathbf{g}, \forall \mathbf{g}$.

6.3 Efficient task-priority using the new operator $\mathbf{P}_{\|\mathbf{e}\|}$

Since the projection operator $\mathbf{P}_\lambda$ provides a highly redundant system, it is possible to introduce several constraints and to manage the order of priority between them. When using

the new projection operator $\mathbf{P}_\lambda$, each subtask is equivalent to a scalar constraint and it is thus possible to consider a large number of subtasks, whatever their corresponding dimension and rank are (rank$(\mathbf{J_i}) \leq n$ for $i = 1, .., l$). This enlarges the applicability domain of redundancy-based task-priority.

6.3.1 Two tasks

Recalling (5.25), when only two tasks are considered then the general control is given as:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_\mathbf{1} + \left(\mathbf{J_2}\mathbf{P}_{\|\mathbf{e_1}\|}\right)^+ (\dot{\mathbf{e}}_\mathbf{2}^* - \mathbf{J_2}\dot{\mathbf{q}}_\mathbf{1}) \tag{6.42}$$

which allows us to use the efficient scheme for task-priority even when the main task uses all the robot degrees of freedom.

6.3.2 Several tasks

Similarly as in Section 5.3.4.2, For l tasks, we define the general control scheme as $\dot{\mathbf{q}} = \dot{\mathbf{q}}_\mathbf{l}$ where $\dot{\mathbf{q}}_\mathbf{i}$ is given by (see 5.26):

$$\dot{\mathbf{q}}_\mathbf{i} = \begin{cases} \dot{\mathbf{q}}_{\mathbf{i-1}} + \left(\mathbf{J_i}\mathbf{P}^{\mathbf{A}}_{\|\mathbf{e_{i-1}}\|}\right)^+ (\dot{\mathbf{e}}_\mathbf{i}^* - \mathbf{J_i}\dot{\mathbf{q}}_{\mathbf{i-1}}), & \text{if } i = 2, .., l \\ \mathbf{J}_\mathbf{1}^+\dot{\mathbf{e}}_\mathbf{1}^*, & \text{if } i = 1 \end{cases} \tag{6.43}$$

where $\mathbf{P}^{\mathbf{A}}_{\|\mathbf{e_i}\|}$ is the new projection operator on the null space of the augmented Jacobian $\mathbf{J}^{\mathbf{A}}_\mathbf{i}$. When $i = 1$ then $\mathbf{P}^{\mathbf{A}}_{\|\mathbf{e_1}\|} = \mathbf{P}_{\|\mathbf{e_1}\|}$ while if $i = 2..l$ then $\mathbf{P}^{\mathbf{A}}_{\|\mathbf{e_i}\|}$ is defined by:

$$\mathbf{P}^{\mathbf{A}}_{\|\mathbf{e}^{\mathbf{A}}_\mathbf{i}\|} = \mathbf{I}_n - \frac{1}{\mathbf{e}^{\mathbf{A}\top}_\mathbf{i}\mathbf{J}^{\mathbf{A}}_\mathbf{i}\mathbf{J}^{\mathbf{A}\top}_\mathbf{i}\mathbf{e}^{\mathbf{A}}_\mathbf{i}}\ \mathbf{J}^{\mathbf{A}\top}_\mathbf{i}\mathbf{e}^{\mathbf{A}}_\mathbf{i}\mathbf{e}^{\mathbf{A}\top}_\mathbf{i}\mathbf{J}^{\mathbf{A}}_\mathbf{i} \tag{6.44}$$

where $\mathbf{e}^{\mathbf{A}}_\mathbf{i}$ is the augmented task and $\mathbf{J}^{\mathbf{A}}_\mathbf{i}$ is the augmented Jacobian::

$$\mathbf{e}^{\mathbf{A}}_\mathbf{i} = (\mathbf{e}^{\mathbf{A}}_{\mathbf{i-1}}, \mathbf{e_i}),\ \ \mathbf{e}^{\mathbf{A}}_\mathbf{1} = \mathbf{e_1} \tag{6.45}$$

$$\mathbf{J}^{\mathbf{A}}_\mathbf{i} = (\mathbf{J}^{\mathbf{A}}_{\mathbf{i-1}}, \mathbf{J_i}),\ \ \mathbf{J}^{\mathbf{A}}_\mathbf{1} = \mathbf{J_1} \tag{6.46}$$

Equation (6.44) can be written as:

$$\mathbf{P}^{\mathbf{A}}_{\|\mathbf{e}^{\mathbf{A}}_\mathbf{i}\|} = \mathbf{I}_n - \frac{1}{\mathbf{a}^{\mathbf{A}}_\mathbf{i}\mathbf{a}^{\mathbf{A}\top}_\mathbf{i}}\ \mathbf{a}^{\mathbf{A}\top}_\mathbf{i}\mathbf{a}^{\mathbf{A}}_\mathbf{i} \tag{6.47}$$

where $\mathbf{a}^{\mathbf{A}}_\mathbf{i}$ is defined as:

$$\mathbf{a}^{\mathbf{A}}_\mathbf{i} = \mathbf{e}^{\mathbf{A}\top}_\mathbf{i}\mathbf{J}^{\mathbf{A}}_\mathbf{i} \tag{6.48}$$

The recursive formula of $\mathbf{a}^{\mathbf{A}}_\mathbf{i}$ is obtained by injecting (6.45) and (6.46) in (6.48) to get:

$$\begin{aligned} \mathbf{a}^{\mathbf{A}}_\mathbf{i} &= \left[\mathbf{e}^{\mathbf{A}\top}_{\mathbf{i-1}} \quad \mathbf{e}^{\top}_\mathbf{i}\right](\mathbf{J}^{\mathbf{A}}_{\mathbf{i-1}}, \mathbf{J_i}) \\ &= \mathbf{e}^{\mathbf{A}\top}_{\mathbf{i-1}}\mathbf{J}^{\mathbf{A}}_{\mathbf{i-1}} + \mathbf{e}^{\top}_\mathbf{i}\mathbf{J_i} \\ &= \mathbf{a}^{\mathbf{A}}_{\mathbf{i-1}} + \mathbf{e}^{\top}_\mathbf{i}\mathbf{J_i} \end{aligned} \tag{6.49}$$

We thus obtain:

$$\mathbf{a_i^A} = \begin{cases} \mathbf{a_{i-1}^A} + \mathbf{e_i}^\top \mathbf{J_i}, & \text{if } i = 2, .., l \\ \mathbf{e_1}^\top \mathbf{J_1}, & \text{if } i = 1 \end{cases} \tag{6.50}$$

Another recursive formula of $\mathbf{P}^{\mathbf{A}}_{\|\mathbf{e_i^A}\|}$ can directly be obtained from (5.35) by using the particular form $\mathbf{J}_{\|\mathbf{e_i}\|} = \frac{1}{\|\mathbf{e}\|}\mathbf{e_i}^\top \mathbf{J_i}$:

$$\mathbf{P}^{\mathbf{A}}_{\|\mathbf{e_i^A}\|} = \begin{cases} \mathbf{P}^{\mathbf{A}}_{\|\mathbf{e_{i-1}^A}\|} - \Pi, & \text{if } i = 2, .., l \\ \mathbf{I}_n - \frac{1}{\mathbf{e_i}^\top \mathbf{J_{e_i}} \mathbf{J_{e_i}^\top} \mathbf{e_i}} \ \mathbf{J_{e_i}^\top} \mathbf{e_i} \mathbf{e_i}^\top \mathbf{J_{e_i}}, & \text{if } i = 1 \end{cases} \tag{6.51}$$

where

$$\Pi = \frac{1}{\mathbf{b_i^A} \mathbf{b_i^A}^\top} \ \mathbf{b_i^A}^\top \mathbf{b_i^A} \tag{6.52}$$

and

$$\mathbf{b_i^A} = \mathbf{e_i}^\top \mathbf{J_i} \mathbf{P}^{\mathbf{A}}_{\|\mathbf{e_{i-1}^A}\|} \tag{6.53}$$

As can be seen, the computational cost required for computing the recursive formula (6.51), which requires the computation of term Π, is higher compared to the other recursive form defined by (6.47) and (6.50). That is why we prefer to use (6.47) and (6.50) in practice.

Finally, let us note that a switching from $\mathbf{P}^A_{\|\mathbf{e_i}\|}$ to $\mathbf{P_i}^A$ has to be performed as soon as $\mathbf{e_i^A} \to \mathbf{0}$. This can be done using the same strategy than the one proposed in Section 6.2.2.

6.3.3 Redundancy-based articular-task-priority

Now, we consider the case when a task $\mathbf{e_i}$ describes a desired motion in the articular space given by an articular velocity $\dot{\mathbf{q}}_\mathbf{i}^*$. Instead of replacing the Jacobian $\mathbf{J_i}$ by the identity matrix $\mathbf{I}_n$ as in [Hanafusa 81], we propose to replace $\mathbf{J_i}$ by a diagonal matrix $\bar{\mathbf{I}}_n$ having zero value in all components not used by $\mathbf{e_i}$. In that case, the control schemes (5.26) and (6.43) are given by:

$$\dot{\mathbf{q}}_\mathbf{i} = \dot{\mathbf{q}}_{\mathbf{i-1}} + \left(\bar{\mathbf{I}}_\mathbf{n} \mathbf{P}^{\mathbf{A}}_{*_{\mathbf{i-1}}}\right)^+ \left(\dot{\mathbf{q}}_\mathbf{i}^* - \bar{\mathbf{I}}_\mathbf{n} \dot{\mathbf{q}}_{\mathbf{i-1}}\right), \quad i = 2..l \tag{6.54}$$

where $\mathbf{P}^A_{*_{\mathbf{i-1}}}$ is $\mathbf{P}^A_{\|\mathbf{e_{i-1}}\|}$ or $\mathbf{P}^A_{i-1}$. In order to analyses the behavior of (6.54), let us consider a simple case when $\mathbf{e_i} \in \mathbb{R}^n$ is to send a motion to a joint j_0. The above equation (6.54) indicates that if the joint j_0 is not used by the higher level priority tasks, then $\dot{\mathbf{q}}_\mathbf{i}^*[j_0]$ will not be modified before being projected by $\left(\bar{\mathbf{I}}_\mathbf{n} \mathbf{P}^{\mathbf{A}}_{*_{\mathbf{i-1}}}\right)^+$. If the joint j_0 is already used by the first (i-1) tasks, then $\dot{\mathbf{q}}_{\mathbf{i-1}}[j_0] \neq 0$ is subtracted from $\dot{\mathbf{q}}_\mathbf{i}^*[j_0]$ and the difference is then projected by $\left(\bar{\mathbf{I}}_\mathbf{n} \mathbf{P}^{\mathbf{A}}_{*_{\mathbf{i-1}}}\right)^+$. When $j \neq j_0$ and $\dot{\mathbf{q}}_{\mathbf{i-1}}[j] \neq 0$, then $(\bar{\mathbf{I}}_n \dot{\mathbf{q}}_{\mathbf{i-1}})[j] = 0$, which ensures that there is no unrequired motion from $\dot{\mathbf{q}}_{\mathbf{i-1}}[j]$ to be projected, while using the identity matrix $\mathbf{I_n}$ as proposed in [Hanafusa 81] we get $(\dot{\mathbf{q}}_\mathbf{i}^* - \bar{\mathbf{I}}_n \dot{\mathbf{q}}_{\mathbf{i-1}})[j] = -\dot{\mathbf{q}}_{\mathbf{i-1}}[j]$.

Now, if a zero articular value is to be sent to the joint j_0, then $\dot{\mathbf{q}}_\mathbf{i}^*[j_0] = 0$ and $\bar{\mathbf{I}}_\mathbf{n}[j, j] = 1$.

When $\left(\bar{\mathbf{I}}_{\mathbf{n}}\mathbf{P}^{\mathbf{A}}_{*_{\mathbf{i-1}}}\right)^{+}$ lets the required movement in the j^{th} component of $(\dot{\mathbf{q}}^{*}_{\mathbf{i}} - \dot{\mathbf{q}}_{\mathbf{i-1}})$ to be projected completely, then we get $\dot{\mathbf{q}}_{\mathbf{i}}[j] = \dot{\mathbf{q}}^{*}_{\mathbf{i}}[i]$.

If two tasks $\mathbf{e_1}$ and $\mathbf{e_2}$ are considered while $\mathbf{e_2}$ is defined in the articular space, the control schemes (5.25) and (6.42) are given by:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_{\mathbf{1}} + \left(\bar{\mathbf{I}}\mathbf{P}^{\mathbf{A}}_{*_{\mathbf{i-1}}}\right)^{+}\left(\dot{\mathbf{e}}_{\mathbf{2}} - \bar{\mathbf{I}}\dot{\mathbf{q}}_{\mathbf{1}}\right) \tag{6.55}$$

where $\mathbf{P}^{A}_{*_{\mathbf{i-1}}}$ is $\mathbf{P}^{A}_{\|\mathbf{e}_{\mathbf{i-1}}\|}$ or $\mathbf{P}^{A}_{\mathbf{e}_{\mathbf{i-1}}}$. For a 6 DOFs robot manipulator, if $\mathbf{e_2}$ is used to change the first three joints, we set $\text{diag}(\bar{\mathbf{I}}) = (1,1,1,0,0,0)$. When it is required to change the last three joints by $\mathbf{e_2}$ then $\text{diag}(\bar{\mathbf{I}}) = (0,0,0,1,1,1)$.

This analysis shows that task priority based redundancy gives two directions of priority. The first direction is a task-priority, which ensures the performance of the most significant task defined as the tasks of higher priorities while using the remaining redundancy to perform the lower priority tasks if it is possible. The second direction is a behavior-priority of the global motion obtained by the control scheme which tries to give the motion required to perform tasks of lower level of priority whenever it is possible, based on the available redundancy.

6.4 Experimental Results

The experimental results presented in this section have been obtained after applying the proposed methods in visual servoing using a six degrees of freedom Gantry robot. The task function is defined by $\mathbf{e} = \mathbf{s} - \mathbf{s}^{*}$ where $\mathbf{s}$ and $\mathbf{s}^{*} \in \mathbb{R}^{k}$ are two vectors representing the current and the desired selected visual features. The task Jacobian $\mathbf{J_e} = \mathbf{L_s}\mathbf{M}\mathbf{J_q}$, where $\mathbf{J_q}$ is the robot Jacobian and $\mathbf{M}$ is the matrix that relates $\mathbf{v}$ to the variation of the camera pose $\mathbf{p}$ by $\mathbf{v} = \mathbf{M}\dot{\mathbf{p}}$. Two different objects are used. The first object (obj_1) is a square of length 0.1 m composed of four points. The Cartesian coordinates of these points in the image define the visual features $(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$ used in the visual servoing system. In that case, we have a system of six DOFs and eight visual features with a main task of full rank 6. The second object (obj_2) is a cylinder of radius 0.08 m. The visual features are now defined by parameters $(\rho_1, \theta_1, \rho_2, \theta_2)$ that describe the configuration of the cylinder in the image plane and the main task is of full rank four. The camera pose is represented by $\mathbf{p} = (\mathbf{t}, \mathbf{r})$ where $\mathbf{t}$ is a translational vector expressed in meter and $\mathbf{r} = \theta\mathbf{u}$ is the rotational vector expressed in degree using the classical angle axis representation.

Two sets of experiments are presented: Set A in Section 6.4.1 using obj_1 compares different control schemes and studies the behavior of the new projection operator by applying simple secondary tasks; Set B in Section 6.4.2 using obj_2 compares $\mathbf{P_e}$ and $\mathbf{P}_{\lambda}$ when classical and efficient redundancy-based task-priority control schemes are employed.

6.4.1 New Projection operator using obj_1

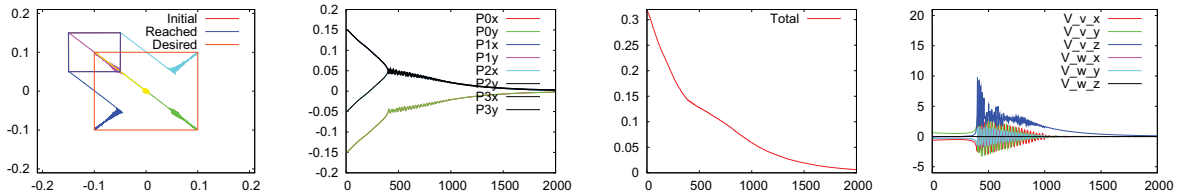
Three different cases have been implemented to validate and investigate the efficiency of the new projection operator when a full rank main task is considered. The chosen task consists of positioning the camera with respect to obj_1. In all cases, the desired camera pose is (0, 0, 0.5, 0, 0, 0), which means that the camera has to be at 0.5 m in front of the square obj_1 so that it appears as a centered square in the image.

6.4.1.1 Case A1: Control schemes $\dot{\mathbf{q}}_{\|\mathbf{e}\|}$ and $\dot{\mathbf{q}}_e$

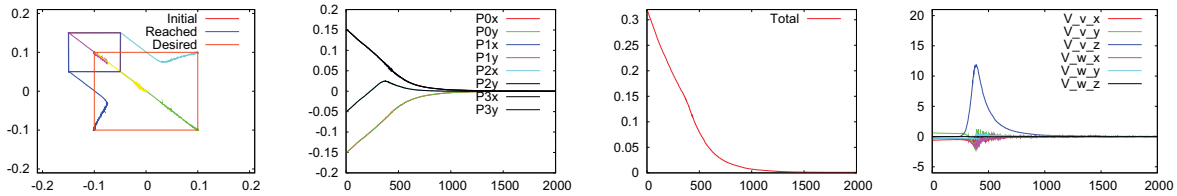
In this case, we study the behavior of $\dot{\mathbf{q}}_{\|\mathbf{e}\|}$ and the effect of the adaptive gain $\beta(\|\mathbf{e}\|)$ with the classical control scheme $\dot{\mathbf{q}}_\mathbf{e}$. The initial camera pose is I=(-0.1,0.1,1.0,0,0,0) and no secondary task is added to the main task. Applying $\dot{\mathbf{q}}_{\|\mathbf{e}\|}$, the initial movement consists of translations along x-axis and y-axis combined with a small rotation around y-axis till iteration number 80, as depicted in Fig. 6.2(a). Then the translational movement along z-axis starts to increase. Figure 6.2(a) shows also that as soon as $\|\mathbf{e}\|$ nears zero, the control law $\dot{\mathbf{q}}_{\|\mathbf{e}\|}$ is completely unstable, as explained in Section 6.2. As expected, the norm of the total error is exponentially decreasing but during the instability due to the perturbations it induces. If the switching to $\dot{\mathbf{q}}_e$ is performed when $\|\mathbf{e}\| \rightarrow \mathbf{0}$, that is using (6.37), the system converges as can be seen in Fig. 6.2(b). As shown in Fig. 6.2(c), applying the classical control scheme $\dot{\mathbf{q}}_\mathbf{e}$, each error component converges exponentially to zero (as can be seen from the image points trajectories, which are now pure straight lines), as well as the norm of the error. Setting the gain $\lambda = \lambda_0\|\mathbf{e}\|$, the convergence rate is extremely slow, as depicted in Fig. 6.2(d). Finally, using the adaptive gain function $\beta(\|\mathbf{e}\|)$ with the classical control law ensures the convergence of the system as shown in Fig. 6.2(e).

6.4.1.2 Case A2: Secondary task g = (0.1, 0, 0, 0, 0, 0)

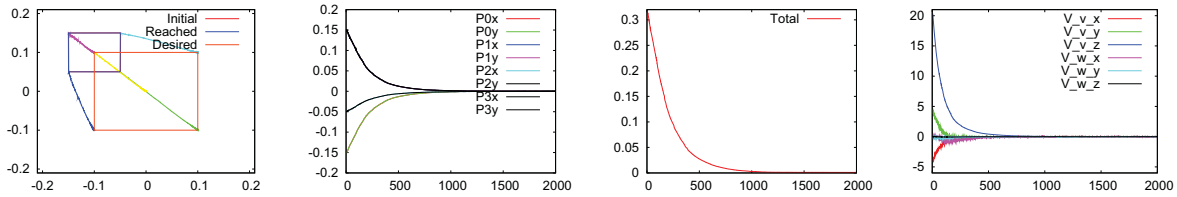
Now, we use the new projection operator and apply a simple secondary task consisting of a translation of 10 cm/s along x-axis. Let us first recall that using $\mathbf{P}_\mathbf{e}$ would not allow any secondary task to be achieved. As expected, using the projection operator $\mathbf{P}_{\|\mathbf{e}\|}$ during all the servo does not produce a satisfactory behavior (see Fig. 6.3). Indeed, some secondary motions are produced while the main task tries the robot to reach the desired pose, where it has then to be motionless. This explains the oscillating behavior on the velocity components of the global task. In Fig 6.4, the projection operator $\mathbf{P}_\lambda$ is used and the system switches automatically to the classical projection operator. This allows the secondary task to be taken into account at the beginning of the servo and ensures the convergence of the system to the desired position. Fig. 6.5 shows the results obtained when the switching gain function $\beta(\|\mathbf{e}\|)$ is used. It is clear that the secondary task is considered during a longer number of iterations.



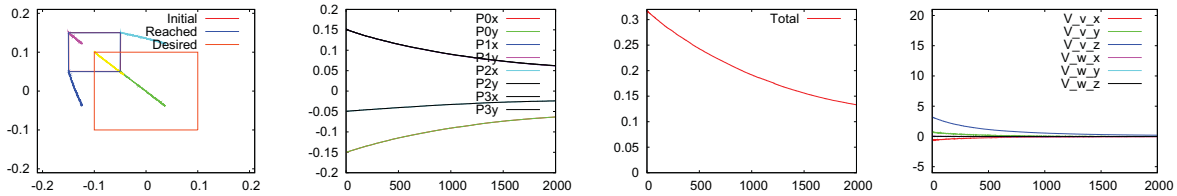
(a) $\dot{\mathbf{q}}_{\|\mathbf{e}\|}$



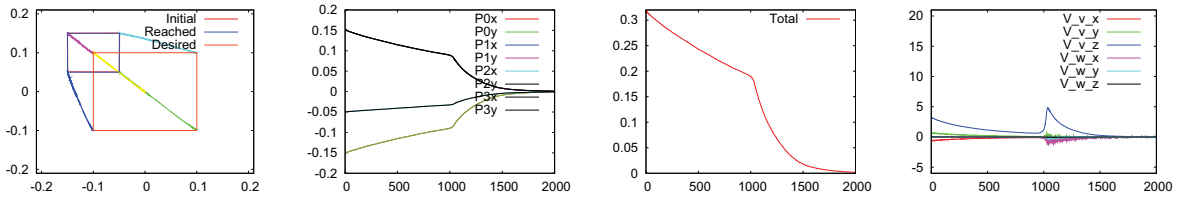
(b) $\dot{\mathbf{q}}_{\|\mathbf{e}\|}$ & switching



(c) $\dot{\mathbf{q}}_{\mathbf{e}}$, $\lambda = \lambda_0$



(d) $\dot{\mathbf{q}}_{\mathbf{e}}$, $\lambda = \lambda_0\|\mathbf{e}\|$



(e) $\dot{\mathbf{q}}_{\mathbf{e}}$, $\lambda = \lambda_0\beta(\|\mathbf{e}\|)$

Figure 6.2 – Results for case A1. Comparison between the different control schemes. Line 1:image points trajectories, line 2: image point error, line 3: norm of the total error, line 4: translational camera velocity (cm/s) and rotational camera velocity (deg/s).

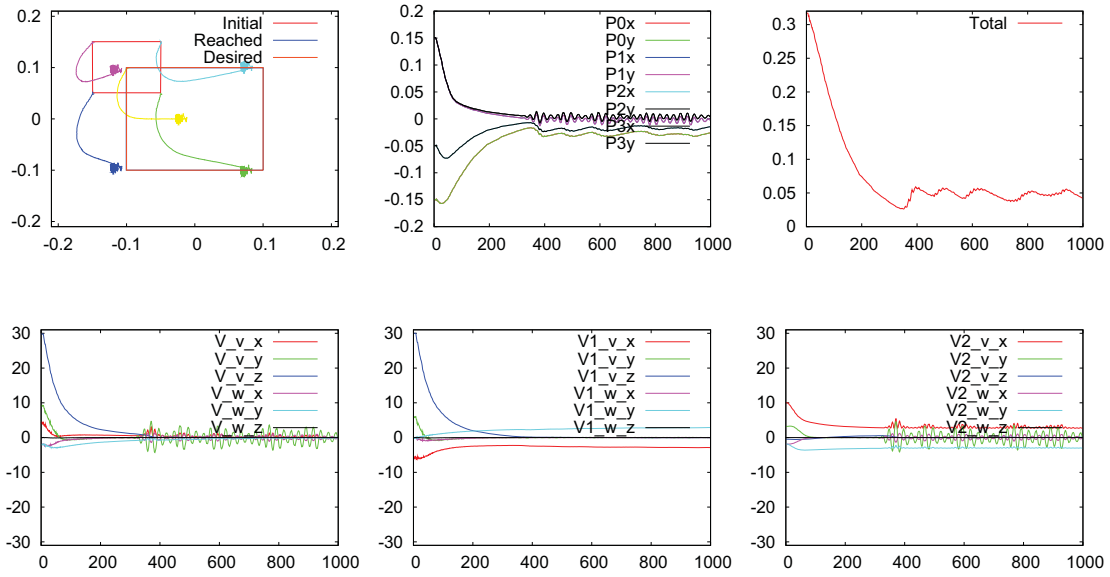


Figure 6.3 – Results for case A2: $\dot{\mathbf{q}}_\mathbf{e}$, $\mathbf{P}_{\|\mathbf{e}\|}$, Line 1 from left to right: image points trajectories, image point error and norm of the total error, lines (2) from left to right: camera velocity components in cm/s and deg/s of the general, main and secondary tasks respectively.

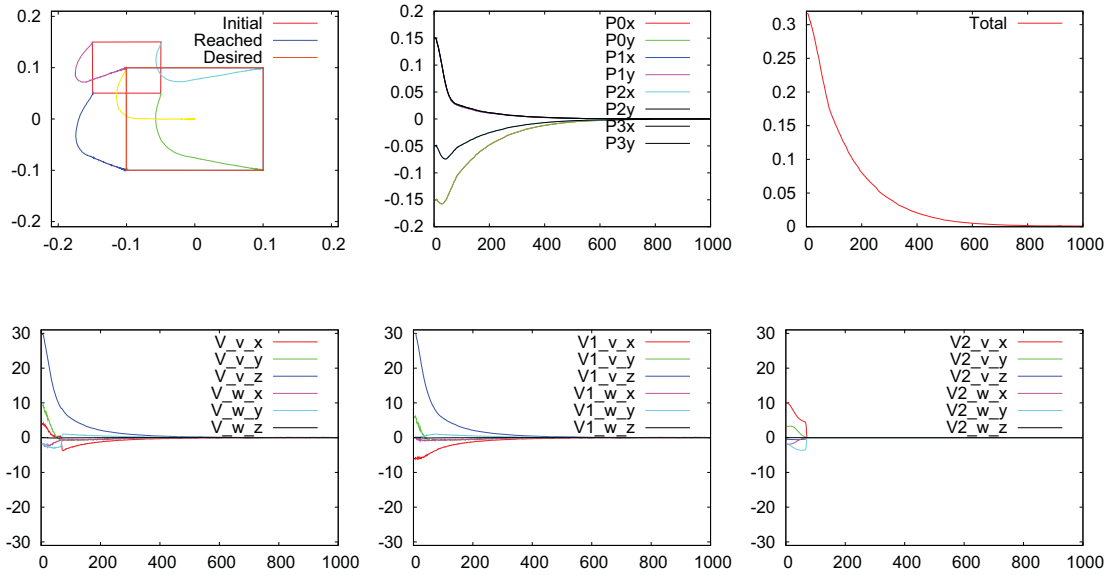


Figure 6.4 – Results for case A2: $\dot{\mathbf{q}}_\mathbf{e}$, P_λ, $\lambda = \lambda_0$.

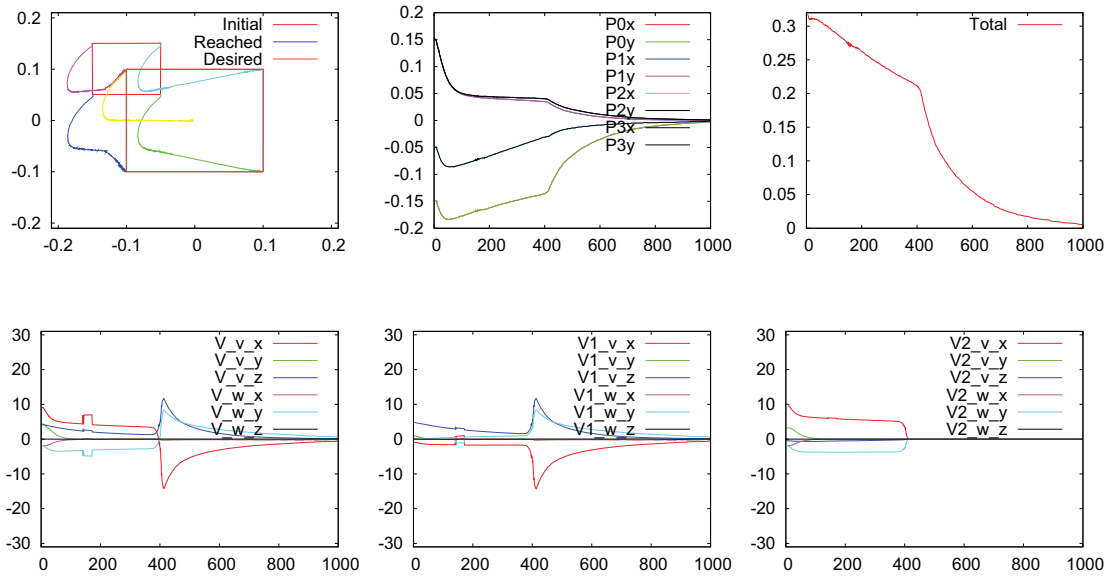


Figure 6.5 – Results for case A2: $\dot{\mathbf{q}}_\mathbf{e}$, P_λ, $\lambda = \lambda_0 \beta(\|\mathbf{e}\|)$.

6.4.1.3 Case A3: Secondary task g = (-0.02,0.04,0.02, 4,4,4)

In this case, a more general secondary task is used with non-zero value in all its components. Using the new projection operator $\mathbf{P}_\lambda$ with the classical control scheme, most of the secondary task components are projected successfully onto the main task (see Fig.6.6). At each iteration, the main task tries keeping the exponential decrease of each error component while the projection operator allows keeping the exponential decrease of the norm of the total error, which leads to a nice behavior of the system. Then, thanks to the switching strategy to the classical projection operator, the system converges to the desired pose. When the switching gain function $\beta(\|\mathbf{e}\|)$ is used (see Fig. 6.7), the number of iterations where the secondary task is active increases significantly.

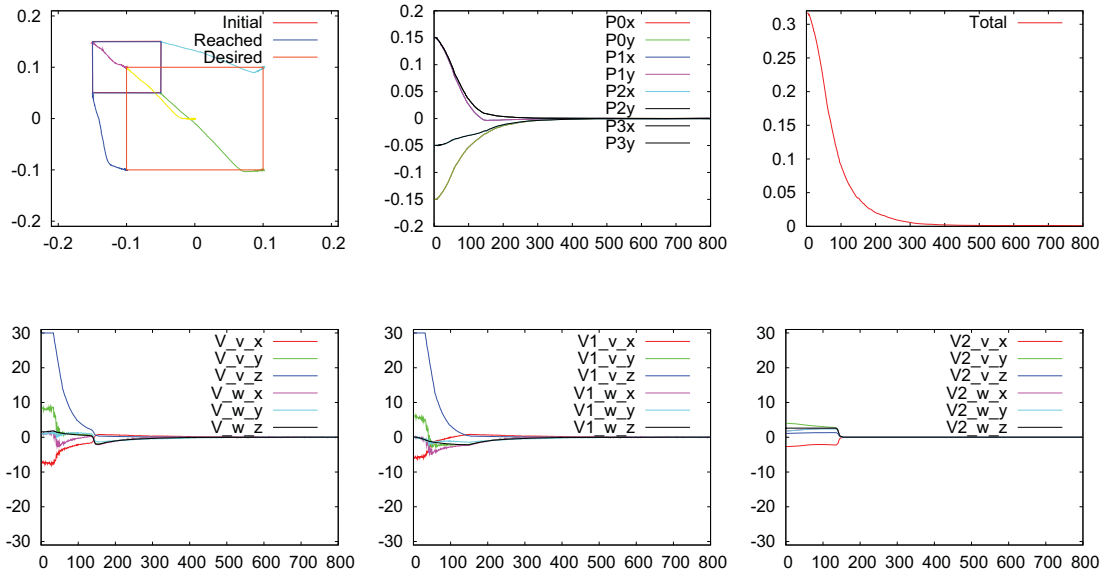


Figure 6.6 – Results for case A3: $\dot{\mathbf{q}}_\mathbf{e}$, P_λ, $\lambda = \lambda_0$.

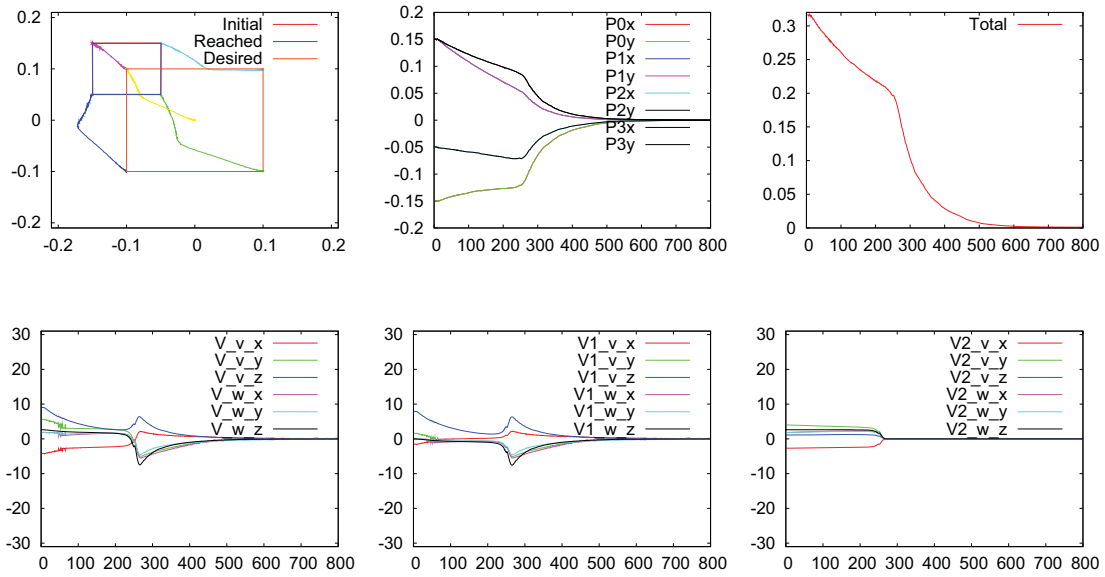
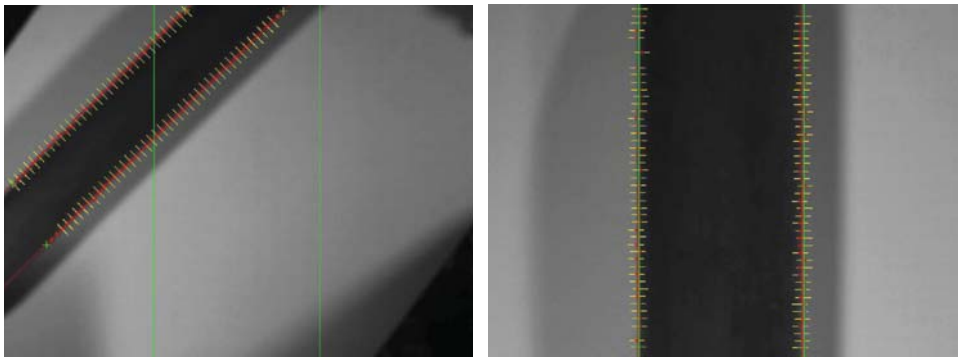


Figure 6.7 – Results for case A3: $\dot{\mathbf{q}}_\mathbf{e}$, P_λ, $\lambda = \lambda_0 \beta(\|\mathbf{e}\|)$.

6.4.2 Comparison between $\mathbf{P}_\lambda$ and $\mathbf{P}_\mathbf{e}$, classical and efficient task-priority, using obj_2

In these experiments, the robot has to reach a desired position where the optical axis of the camera is normal to the axis of the cylinder and y-axis of the camera frame is parallel to this axis (see Fig. 6.8). In that case, the rank of $\mathbf{P_e}$ is two while the rank of $\mathbf{P}_{\|\mathbf{e}\|}$ is five. Different motions moving the robot with a velocity 0.03 m/s in the camera frame are specified as secondary tasks. The vector $\mathbf{g}$ that produces this motion is defined as $\mathbf{g} = (0, 0, 0.03, 0, 0, 0)$ when $(1 \leq iter < 100)$, $\mathbf{g} = (0, 0, -0.03, 0, 0, 0)$ when $(100 \leq iter < 200)$, $\mathbf{g} = (0.03, 0, 0, 0, 0, 0)$ when $(200 \leq iter < 300)$, $\mathbf{g} = (0, 0.03, 0, 0, 0, 0)$ when $(300 \leq iter < 400)$, else $\mathbf{g} = (0, 0, 0, 0, 0, 0)$. The two components available in $\mathbf{P_e}$ are the translation along y axis and a combination of translation along x axis and rotation around y axis so that the image of the cylinder is not modified despite these motions.



(a) Initial image (b) Desired image

Figure 6.8 – Initial and desired camera images of obj_2, (cases B1, B2, B3 and B4)

6.4.2.1 Cases B1 and B2, two tasks, classical task-priority

In this case, the control scheme used is given by:

$$\dot{\mathbf{q}} = -\lambda \mathbf{J}_1^+ \mathbf{e_1} + \lambda_{*1} \mathbf{P}_* \mathbf{g} \tag{6.56}$$

where $\mathbf{P}_*$ represents the projection operator used ($\mathbf{P_e}$ or $\mathbf{P}_\lambda$) and the adaptive gain function λ_{*1} is given by:

$$\lambda_{*1} = \frac{|\mathbf{g}[i_0]|}{|(\mathbf{P}_*\mathbf{g})[i_0]|} \tag{6.57}$$

where i_0 is the axis where the secondary motion projected is the most significant one. Using $\mathbf{P_e}$, motions along x-axis and y-axis are projected as depicted in Fig. 6.9 while no redundancy remains for projecting any motion along z-axis. Using $\mathbf{P}_\lambda$, the vector $\mathbf{g}$ is completely projected into the x, y, and z components of $\mathbf{v_2}$ as depicted in Fig. 6.10. Results also show that the classical redundancy-based task-priority control scheme leads to accumulate the motions needed by the secondary tasks with the motion needed by the main task provided that the convergence of the main task is ensured.

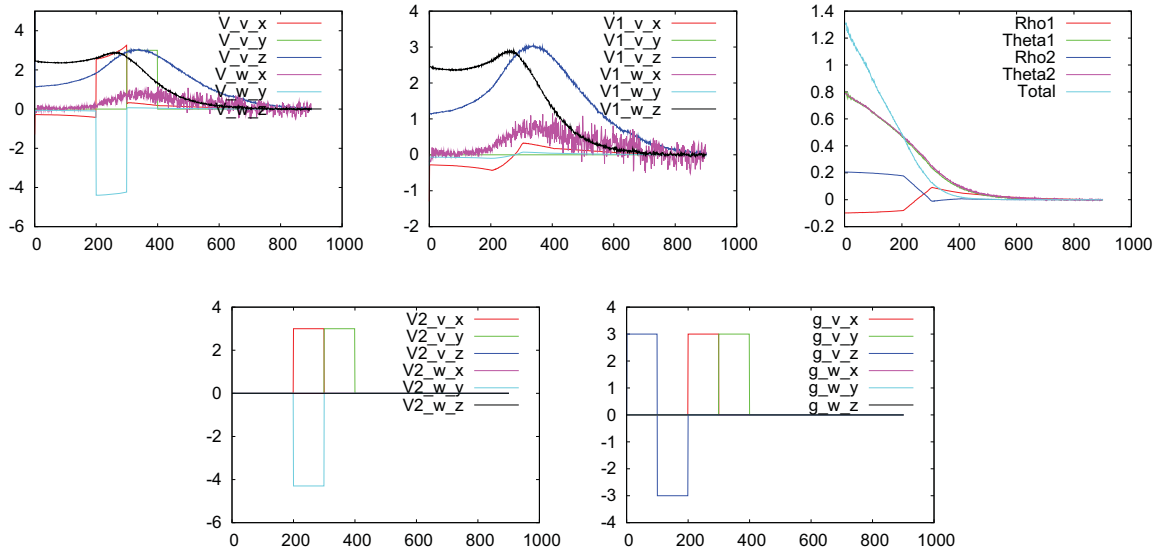


Figure 6.9 – Results for case B1: $\mathbf{P_e}$ and classical redundancy-based task-priority are considered; lines (1) : camera velocity components in cm/s and deg/s of the general and main tasks respectively and image parameter errors for the cylinder, line (2): the projected and the required additional motion in cm/s.

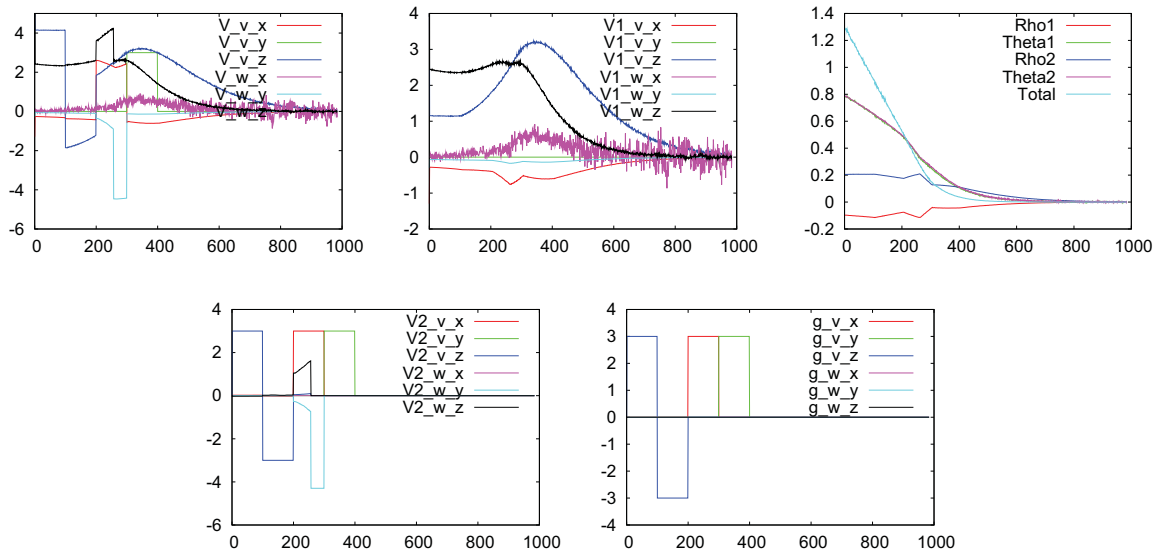


Figure 6.10 – Results for case B2: $\mathbf{P}_\lambda$ and efficient redundancy-based task-priority are considered.

6.4.2.2 Cases B3 and B4, two tasks, efficient task-priority

In this case, the task $\mathbf{e_2}$ is defined to produce some motion in the camera frame by $\dot{\mathbf{e}}_\mathbf{2}^* = \mathbf{g}$. This can be realized using control schemes (5.26) and (6.43) while, instead of replacing the Jacobian $\mathbf{J_2}$ by the identity matrix $\mathbf{I}_n$ as in [Hanafusa 81], which constrains all DOFs, we replace $\mathbf{J_2}$ by a diagonal matrix $\bar{\mathbf{I}}_n$ having zero value in all components not used by $\dot{\mathbf{e}}_\mathbf{2}^*$ (see Section 6.3.3). By considering two tasks $\mathbf{e_1}$ and $\mathbf{e_2}$, the control schemes (5.26) and (6.43) are now given by:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_\mathbf{1} + \lambda_{*2} \left(\bar{\mathbf{I}}_\mathbf{n}\mathbf{P}_*\right)^+ \left(\dot{\mathbf{e}}_\mathbf{2}^* - \bar{\mathbf{I}}_\mathbf{n}\dot{\mathbf{q}}_\mathbf{1}\right) \tag{6.58}$$

where the adaptive gain λ_{*2} is given by:

$$\lambda_{*2} = \frac{\left|\left(\dot{\mathbf{e}}_\mathbf{2}^* - \bar{\mathbf{I}}_\mathbf{n}\dot{\mathbf{q}}_\mathbf{1}\right)[i_0]\right|}{\left|\left(\left(\bar{\mathbf{I}}_\mathbf{n}\mathbf{P}_{*_\mathbf{1}}^\mathbf{A}\right)^+ \left(\dot{\mathbf{e}}_\mathbf{2}^* - \bar{\mathbf{I}}_\mathbf{n}\dot{\mathbf{q}}_\mathbf{1}\right)\right)[i_0]\right|} \tag{6.59}$$

Defining the adaptive gain λ_{*2} by (6.59) ensures producing the required motion on the axis i_0 of the camera.

Again, using $\mathbf{P_e}$, no motion is produced along z-axis as depicted in Fig. 6.11 while using $\mathbf{P}_\lambda$ lets the three motions x, y, and z to be projected. Figure 6.12 shows that the projected motions required by the lower level priority task are perfectly performed by the global task components.

By comparing the results obtained from cases B1 and B2 and the results obtained from cases B3 and B4, we conclude that, when classical task-priority is enabled, the projected motion corresponds to the desired one $\mathbf{g}$ on $\mathbf{v_2}$ but not on $\mathbf{v}$, while when the efficient task-priority is enabled the desired motion $\mathbf{g}$ is realized through $\mathbf{v}$.

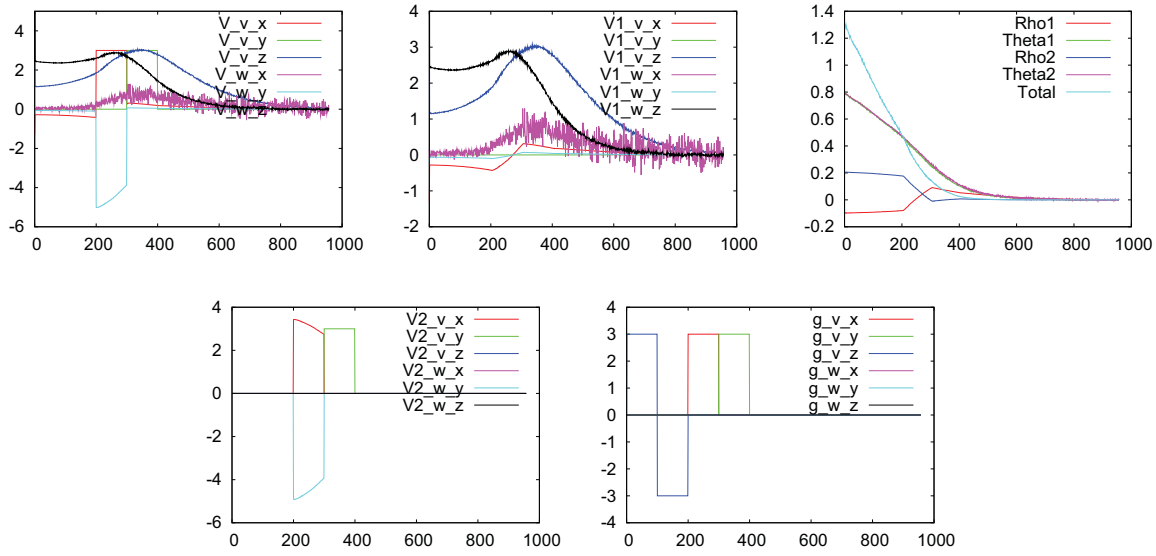


Figure 6.11 – Results for case B3: $\mathbf{P_e}$ and classical redundancy-based task-priority are considered.

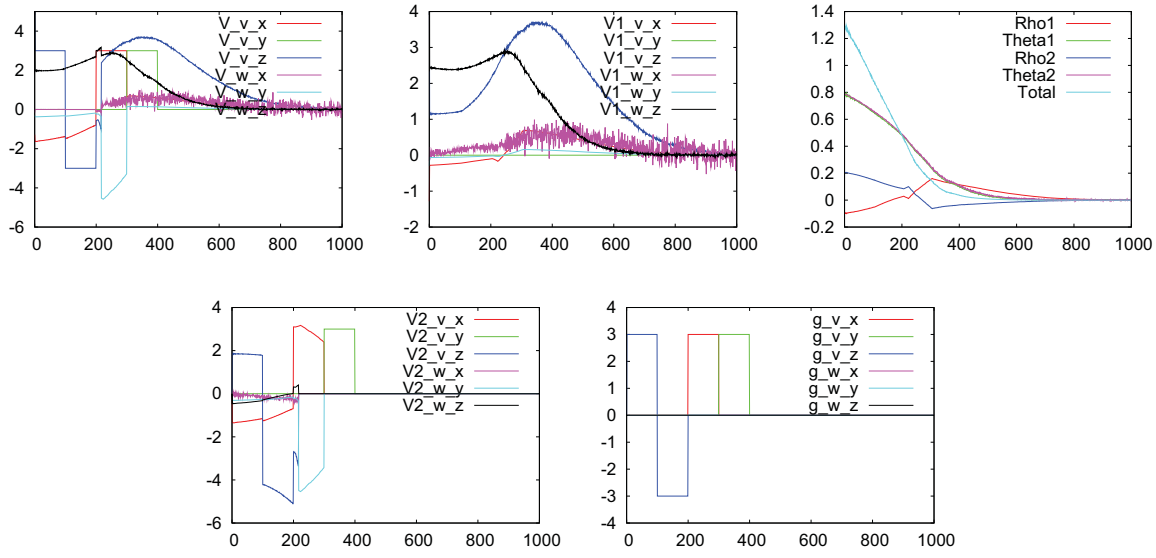


Figure 6.12 – Results for case B4: $\mathbf{P}_\lambda$ and efficient redundancy-based task-priority are considered.

6.5 Conclusion

In this chapter we have proposed a new large projection operator for the redundancy framework. The new projection operator is obtained by considering the norm of the total error. A switching strategy has been proposed to ensure that the new projection operator smoothly switches to the classical projection operator as soon as the error nears zero. An adaptive gain has also been developed so that the secondary task is effective during a long period. That may be useful when the secondary tasks have really to be considered (which is the case for instance for joints limits and obstacle avoidance).

The main interest of the new projection operator is that it is always at least of rank $n-1$. Hence it can be used even if the main task is full rank. The new projection operator has been employed within two different redundancy-based task-priority approaches. A comparative study of results obtained using $\mathbf{P}_\lambda$ and $\mathbf{P}_\mathbf{e}$ shows that $\mathbf{P}_\lambda$ provides more versatility when the main task is not full rank. The effectiveness of the new projection operator has been addressed when the task is defined in the camera space or in the articular space. All experiments have been implemented on a six DOFs robot arm.

These theoretical developments enlarges the applicability domain of the redundancy framework. These enlargements cover all relevant problems solved using the redundancy formalism when it is required to add system constraints (joint limit avoidance, as we will be seen in the next chapter, obstacle avoidance, occlusion avoidance, etc.), to add supplementary tasks (trajectory following, etc.) or to manage the levels of priority among several tasks, using for instance the stack of tasks proposed in [Mansard 07].

Chapter 7

New joint limits avoidance strategy

In this chapter, we present a new redundancy-based strategy for avoiding joint limits of a robot arm. This strategy is based on defining three functions for each joint: an activation function; an adaptive gain function ; and a tuning function. These functions allow to determine automatically the required sign and the suitable magnitude for the avoidance process at each joint. The problem of adding an additional task with the main task and the avoidance process is also considered and solved. As for the redundancy framework, the large projection operator based on the norm of the usual error proposed in the previous chapter is used to enlarge the redundancy domain for applying our proposed avoidance strategy. The three functions defining the new avoidance strategy are also used with kernel-based approach in order to decreasing the discontinuity effect when a new joint is nearing its limits and when a joint becomes very close to one of its two limits. The experimental results obtained on a 6 DOF robot arm in eye-in hand visual servoing show that the new avoidance strategy gives smooth joint avoidance behavior without any tuning step. Using the new projection operator allows a significant improvement of the joint avoidance process, especially in the case of a full rank task function. The work described in this chapter leads to the following publication [Marey 10b].

7.1 Introduction

Joints limit avoidance is a classical and crucial issue in robot control. As already described in Section 5.4, the utilization of redundancy has been widely used for solving this problem. The general solution by this technique is obtained as a minimum norm solution together with a homogeneous solution, which is referred to as self-motion.

This chapter is organized as follows: In Section 7.2, the new avoidance strategy is presented and discussed. In Section 7.2.5, the problem of adding additional secondary tasks to

the avoiding process is considered. In Section 7.3, the simualtaneous avoidance of several joints is considered. Finally, the experimental results are presented in Section 7.4.

7.2 New joint limits avoidance strategy

We recall that the classical control law used to solve joint limits avoidance problem is given by:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_\mathbf{1} + \dot{\mathbf{q}}_\mathbf{a} \tag{7.1}$$

where $\dot{\mathbf{q}}_1$ and $\dot{\mathbf{q}}_\mathbf{a}$ are the velocity vectors of the main task and the avoidance task of the robot arm. We also recall that a safe robot configuration $\mathbf{q}$ is reached when $\forall j_i \in \{j_1, ..., j_n\}$, $q_i \in [\, q_{\ell_0 i}^{\min}, q_{\ell_0 i}^{\max}\,]$, where

$$\begin{aligned} q_{\ell_0 i}^{\min} &= q_i^{\min} + \rho \Delta q_i \\ q_{\ell_0 i}^{\max} &= q_i^{\max} - \rho \Delta q_i \end{aligned} \tag{7.2}$$

where $\rho \in [0, \frac{1}{2}]$ is a tuning parameter and $\Delta q_i = q_i^{\max} - q_i^{\min}$.

The new proposed scheme for $\dot{\mathbf{q}}_\mathbf{a}$ is given by:

$$\dot{\mathbf{q}}_\mathbf{a} = \sum_{i=1}^{n} \dot{\mathbf{q}}_\mathbf{a}^i = -\sum_{i=1}^{n} \lambda_{sec_i} \lambda_{\ell_i} \mathbf{P} \mathbf{g}_i^\ell \tag{7.3}$$

where $\mathbf{P}$ is either the classical projection operator $\mathbf{P_e}$, or the new one $\mathbf{P}_\lambda$, λ_{sec_i} is an adaptive gain function to control the magnitude of the avoidance task, λ_{ℓ_i} is a tuning function to ensure the smoothness of injecting the avoidance task into the main task, and $\mathbf{g}_i^\ell$ is a vector indexing function that controls the activation of the avoidance task and determines its sign.

7.2.1 Activation and sign function

If the configuration is not safe due to the joint j_i, a secondary task is activated by the vector $\mathbf{g}_i^\ell = (\mathbf{g}_i^\ell[1], \mathbf{g}_i^\ell[2], ..., \mathbf{g}_i^\ell[n])$ defined by:

$$\mathbf{g}_i^\ell[i_0] = \begin{cases} -1, & \text{if } q_i < q_{\ell_0 i}^{\min} \text{ and } i = i_0 \\ 1, & \text{if } q_{\ell_0 i}^{\max} < q_i \text{ and } i = i_0 \\ 0, & \text{else} \end{cases} \tag{7.4}$$

where $i_0 \in \{1, 2, ..., n\}$. If $\mathbf{g}_i^\ell[i] = 0$, no avoidance task for the joint j_i is activated. The values 1 and -1 for $\mathbf{g}_i^\ell$ determine the sign of the avoidance task. If $\mathbf{g}_i^\ell[i] = 1$, the avoidance task is negative and causes the joint to move away from its maximum limit $q_i^{\max}$. If $\mathbf{g}_i^\ell[i] = -1$, the avoidance task is positive and causes the joint to move away from its minimum limit $q_i^{\min}$. This can be explained by recalling that the projection operator has its diagonal with elements of non-negative values. Since the vector $\mathbf{g}_i^\ell$ for the joint j_i that nears its limits has only one non-zero element at its i^{th} component, then the sign of $(\mathbf{P}\mathbf{g}_i^\ell)[i]$ is the sign of the i^{th} component of $\mathbf{g}_i^\ell$ since both λ_{sec_i} and λ_{ℓ_i} have positive sign, as we will see in the following.

7.2.2 Tuning function

The tuning function $\lambda_{\ell_i}(q_i)$ depicted in Figure 7.1 is used as a scaling parameter for the secondary task to control its injection onto the main task such that the final behavior of the system avoids any sudden movement due to the secondary task. For the joint under consideration j_i we define this tuning function based on two sigmoid functions as follows:

$$\lambda_{\ell i}(q_i) = \begin{cases} 1, & \text{if } q_i < q_{\ell_1 i}^{\min} \text{ or } q_{\ell_1 i}^{\max} < q_i \\ \frac{\lambda_{\ell i}^{\min}(q_i) - \lambda_{\ell_0 i}^{\min}}{\lambda_{\ell_1 i}^{\min} - \lambda_{\ell_0 i}^{\min}}, & \text{if } q_{\ell_1 i}^{\min} \leq q_i \leq q_{\ell_0 i}^{\min} \\ \frac{\lambda_{\ell i}^{\max}(q_i) - \lambda_{\ell_0 i}^{\max}}{\lambda_{\ell_1 i}^{\max} - \lambda_{\ell_0 i}^{\max}}, & \text{if } q_{\ell_0 i}^{\max} \leq q_i \leq q_{\ell_1 i}^{\max} \\ 0, & \text{if } q_{\ell_0 i}^{\min} < q_i < q_{\ell_0 i}^{\max} \end{cases} \tag{7.5}$$

where as previously defined, $q_{\ell_0 i}^{\min}$ and $q_{\ell_0 i}^{\max}$ are the threshold values to start the secondary task for avoidance. $q_{\ell_1 i}^{\min} = q_{\ell_0 i}^{\min} - \rho_1 \rho \Delta q_i$ and $q_{\ell_1 i}^{\max} = q_{\ell_0 i}^{\max} + \rho_1 \rho \Delta q_i$ are the threshold values at which the smoothing function equals one so that the avoidance task is totally injected into the main task, where $\rho_1 \in]0, 1]$. $\lambda_{\ell i}^{\min}(q_i) : \mathbb{R} \rightarrow \mathbb{R}$ and $\lambda_{\ell i}^{\max}(q_i) : \mathbb{R} \rightarrow \mathbb{R}$ are two continuous monotonically increasing functions such that $\lambda_{\ell_1 i}^{\max} = \lambda_{\ell i}^{\max}(q_{\ell_1 i}^{\max}) \approx 1$, $\lambda_{\ell_0 i}^{\max} = \lambda_{\ell i}^{\max}(q_{\ell_0 i}^{\max}) \approx 0$, $\lambda_{\ell_1 i}^{\min} = \lambda_{\ell i}^{\min}(q_{\ell_1 i}^{\min}) \approx 1$ and $\lambda_{\ell_0 i}^{\min} = \lambda_{\ell i}^{\min}(q_{\ell_0 i}^{\min}) \approx 0$. A good selection for

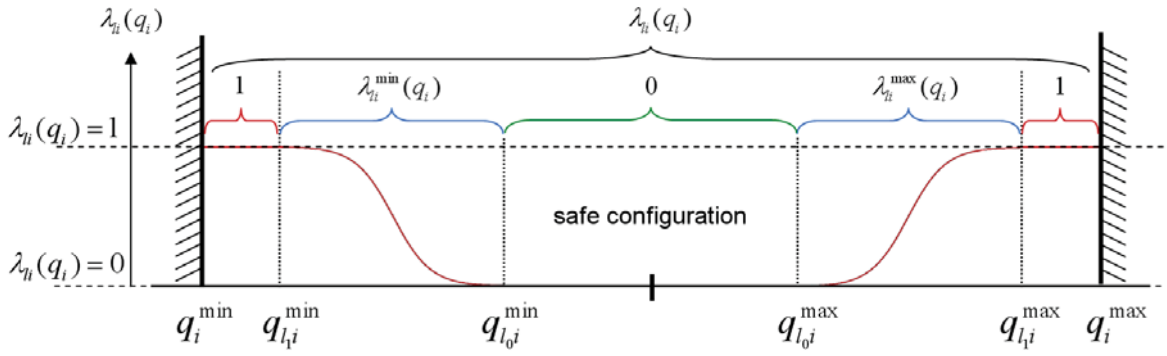


Figure 7.1 – Tunning function $\lambda_{\ell_i}(q_i)$ for the joint j_i

the functions $\lambda_{\ell i}^{\max}(q_i)$ and $\lambda_{\ell i}^{\min}(q_i)$ can be obtained using a sigmoid function as:

$$\lambda_{\ell i}^{\max}(q_i) = \frac{1}{1 + \exp(-12 \frac{q_i - q_{\ell_0 i}^{\max}}{q_{\ell_1 i}^{\max} - q_{\ell_0 i}^{\max}} + 6)} \tag{7.6}$$

$$\lambda_{\ell i}^{\min}(q_i) = \frac{1}{1 + \exp(-12 \frac{q_i - q_{\ell_0 i}^{\min}}{q_{\ell_1 i}^{\min} - q_{\ell_0 i}^{\min}} + 6)} \tag{7.7}$$

Introducing $q_{\ell_1 i}^{\min}$ and $q_{\ell_1 i}^{\max}$ gives the following advantage. By selecting these values different from $q_i^{\min}$ and $q_i^{\max}$ respectively, the joint will never reach the joint limit so that there is no need to predict the next joint position in order to avoid passing the joint limit.

7.2.3 Adaptive gain function

The adaptive gain function λ_{sec_i} is used to adapt the magnitude of $|\dot{\mathbf{q}}^i_{\mathbf{a}}[i]|$ in order to compensate the corresponding component of the main task near a joint limit. For the joint j_i, λ_{sec_i} is defined from the current value of the i^{th} component of both $\dot{\mathbf{q}}_1$ and $(\mathbf{P}\mathbf{g}^\ell_i)$. It is defined by:

$$\lambda_{sec_i} = \begin{cases} (1+\lambda_i)\frac{|\dot{\mathbf{q}}_1[i]|}{|(\mathbf{P}\mathbf{g}^\ell_i)[i]|}, & \text{if } \mathbf{g}^\ell_i[i] \neq 0 \\ 0, & \text{else} \end{cases} \tag{7.8}$$

where $\lambda_i \geq 0$.

An advantage of using this gain function is that the value of $\dot{\mathbf{q}}^i_{\mathbf{a}}$ given by (7.3) is a function of $\dot{\mathbf{q}}_{\mathbf{1}}$, so that at any time step, the value of $\dot{\mathbf{q}}^i_{\mathbf{a}}$ is compatible with $\dot{\mathbf{q}}_{\mathbf{1}}$ to be sent to the joint j_i. It can be demonstrated that using the adaptive gain function λ_{sec_i} allows ensuring that the i^{th} velocity component of the main task and of the avoidance task satisfy the inequality $|\dot{\mathbf{q}}^i_{\mathbf{a}}[i]| > |\dot{\mathbf{q}}_{\mathbf{1}}[i]|$ before reaching the limit $q_i^{\max}$ or $q_i^{\min}$. Indeed, since near a joint limit $\lambda_{\ell i}(q_i) = 1$ and $\lambda_{sec_i} = (1+\lambda_i)\frac{|\dot{\mathbf{q}}_1[i]|}{|(\mathbf{P}\mathbf{g}^\ell_i)[i]|}$ then by recalling (7.3) we obtain $|\dot{\mathbf{q}}^i_{\mathbf{a}}[i]| = |-(1+\lambda_i)\dot{\mathbf{q}}_{\mathbf{1}}[i]| > |\dot{\mathbf{q}}_{\mathbf{1}}[i]|$ where $\lambda_i > 0$. Therefore, the gain function λ_{sec_i} ensures that the joint limit will never be reached, this property being obtained without any gain tuning step.

7.2.4 Behavior analysis of $\dot{\mathbf{q}}^i_{\mathbf{a}}$

Now, we study more precisely the behavior of $\dot{\mathbf{q}}^i_{\mathbf{a}}$ for avoiding the limits of the joint j_i. By injecting (7.30) and (7.5) in (7.3) , we get:

$$\dot{\mathbf{q}}^i_{\mathbf{a}} = \begin{cases} -(1+\lambda_i)\frac{|\dot{\mathbf{q}}_1[i]|}{|(\mathbf{P}\mathbf{g}^\ell_i)[i]|}\mathbf{P}\mathbf{g}^\ell_i, & \text{if } C_1 \\ -\lambda_{\ell i}(q_i)(1+\lambda_i)\frac{|\dot{\mathbf{q}}_1[i]|}{|(\mathbf{P}\mathbf{g}^\ell_i)[i]|}\mathbf{P}\mathbf{g}^\ell_i, & \text{if } C_2 \\ 0, & \text{if } C_3 \end{cases} \tag{7.9}$$

where

$$\begin{aligned} C_1 &\equiv (q_i < q^{\min}_{\ell_1 i} \text{ or } q^{\max}_{\ell_1 i} < q_i) \\ C_2 &\equiv (q^{\min}_{\ell_1 i} \leq q_i \leq q^{\min}_{\ell_0 i} \text{ or } q^{\max}_{\ell_0 i} \leq q_i \leq q^{\max}_{\ell_1 i}) \\ C_3 &\equiv (q^{\min}_{\ell_0 i} < q_i < q^{\max}_{\ell_0 i}) \end{aligned} \tag{7.10}$$

By investigating (7.9), we can study the behavior of the i^{th} component of the secondary task $|\dot{\mathbf{q}}^i_{\mathbf{a}}[i]|$ (see Figure 7.2). Within the two intervals $[q^{\max}_{\ell_0 i}, q^{\max}_{\ell_1 i}]$ and $[q^{\min}_{\ell_1 i}, q^{\min}_{\ell_0 i}]$, the value of $|\dot{\mathbf{q}}^i_{\mathbf{a}}[i]|$ is changing continuously from 0 to $|(1+\lambda_i)\dot{\mathbf{q}}_{\mathbf{1}}[i]|$ as the value q_i of the joint j_i is changing continuously from $q^{\max}_{\ell_0 i}$ to $q^{\max}_{\ell_1 i}$ and from $q^{\min}_{\ell_0 i}$ to $q^{\min}_{\ell_1 i}$ respectively. Within the two intervals $[q^{\max}_{\ell_1 i}, q^{\max}_i]$ and $[q^{\min}_i, q^{\min}_{\ell_1 i}]$, the value of $|\dot{\mathbf{q}}^i_{\mathbf{a}}[i]|$ is greater than or equal to $|\dot{\mathbf{q}}_{\mathbf{1}}[i]|$. The value of the constant λ_i is used to tune the difference in the magnitude between $|\dot{\mathbf{q}}_{\mathbf{1}}[i]|$ and $|\dot{\mathbf{q}}^i_{\mathbf{a}}[i]|$. For example, if $\lambda_i = 0$, we get $|\dot{\mathbf{q}}^i_{\mathbf{a}}[i]| = |\dot{\mathbf{q}}_{\mathbf{1}}[i]|$ as soon as the value of the joint j_i

reaches $q_{\ell_1 i}^{\max}$ or $q_{\ell_1 i}^{\min}$. While if $\lambda_i = 1$, as soon as the value of q_i approaches $q_{\ell_1 i}^{\max}$ or $q_{\ell_1 i}^{\min}$ we obtain $|\dot{\mathbf{q}}_{\mathbf{a}}^{i}[i]| = 2|\dot{\mathbf{q}}_{\mathbf{1}}[i]|$. Figure 7.2 illustrates the relations between i^{th} velocity component of $\dot{\mathbf{q}}_{\mathbf{a}}$ and the i^{th} velocity component of $\dot{\mathbf{q}}_{\mathbf{1}}$ for all values of $q_i \in [q_i^{\min}, q_i^{\max}]$. Red and blue arrows mean that the corresponding velocity component causes the joint to move away from its minimum and maximum limit respectively. It is ensured by this investigation that, near to a limit of a joint j_i, $|\dot{\mathbf{q}}_{\mathbf{a}}^{i}[i]| \geq |\dot{\mathbf{q}}_{\mathbf{1}}[\mathbf{i}]|$.

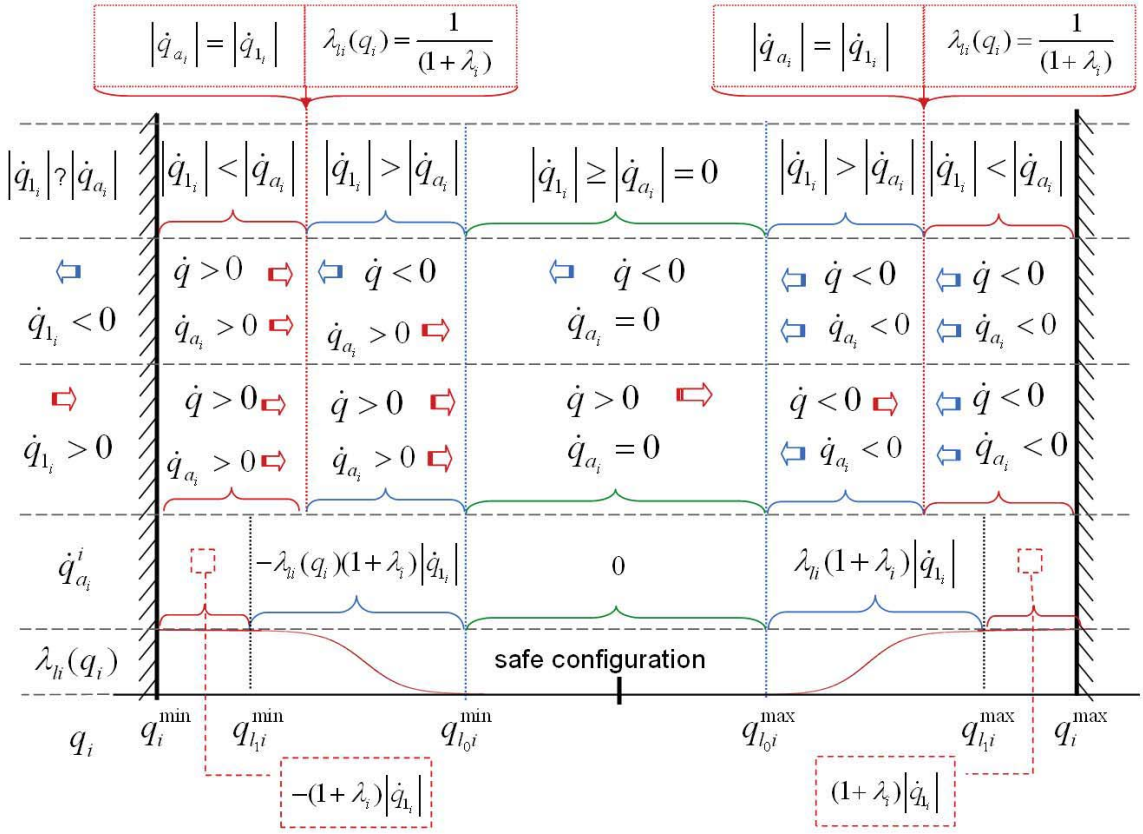


Figure 7.2 – Comparison between the magnitudes of $\dot{\mathbf{q}}_{\mathbf{a}}[i]$ and $\dot{\mathbf{q}}_{\mathbf{1}}[i]$ within the different values for q_i of the joint j_i and when $\lambda_i > 0$; Illustration of the relation between the direction of $\dot{\mathbf{q}}_{\mathbf{1}}[i]$, $\dot{\mathbf{q}}_{\mathbf{a}}^{i}[i]$, and $\dot{\mathbf{q}}[i]$.

7.2.5 Additional secondary tasks

7.2.5.1 Additional secondary tasks having the same priority level

Now, we consider the case when there are three tasks to be performed by the system. The three tasks consist of the main task; the secondary task for robot joint limits avoidance, and an additional secondary task. In this case, the control law is defined as:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_\mathbf{1} + \dot{\mathbf{q}}_\mathbf{a} + \dot{\mathbf{q}}_\mathbf{2} \tag{7.11}$$

where $\dot{\mathbf{q}}_\mathbf{1}$ is the velocity vector due to the main task, $\dot{\mathbf{q}}_\mathbf{a}$ is the velocity vector due to the joint limit avoidance and finally $\dot{\mathbf{q}}_\mathbf{2}$ is the velocity vector to produce the required secondary task. More precisely, we get:

$$\dot{\mathbf{q}} = -\lambda_1 \mathbf{J_e}^+ \mathbf{e_1} - \sum_{i=1}^{n} \lambda^a_{sec_i} \lambda_{\ell_i} \mathbf{P}\mathbf{g}^\ell_i - \lambda_2 \mathbf{P}\mathbf{g_2} \tag{7.12}$$

Since the additional secondary task $\dot{\mathbf{q}}_\mathbf{2}$ can lead to reach a joint limit, it is necessary to change the definition of $\lambda^a_{sec_i}$ in (7.30) to consider $\dot{\mathbf{q}}_\mathbf{2}$. In that case $\lambda^a_{sec_i}$ is defined by:

$$\lambda^a_{sec_i} = \begin{cases} (1+\lambda_i)\frac{|\dot{\mathbf{q}}_*[i]|}{|\mathbf{P}\mathbf{g}^\ell_i[i]|} & \text{if } \ \mathbf{g}^\ell_i[i] > 0 \\ 0, & \text{else} \end{cases} \tag{7.13}$$

where $\dot{\mathbf{q}}_* = \dot{\mathbf{q}}_\mathbf{1} + \dot{\mathbf{q}}_\mathbf{2}$. This form for the adaptive gain ensures that the total velocity component is taken into account by $\dot{\mathbf{q}}_\mathbf{a}$ for joint limit avoidance. The avoidance task in this case is given by:

$$\dot{\mathbf{q}}^i_\mathbf{a} = \begin{cases} -(1+\lambda_i)\frac{|\dot{\mathbf{q}}_*[i]|}{|(\mathbf{P}\mathbf{g}^\ell_i)[i]|}\mathbf{P}\mathbf{g}^\ell_i, & \text{if } C_1 \\ -\lambda_{\ell i}(q_i)(1+\lambda_i)\frac{|\dot{\mathbf{q}}_*[i]|}{|(\mathbf{P}\mathbf{g}^\ell_i)[i]|}\mathbf{P}\mathbf{g}^\ell_i, & \text{if } C_2 \\ 0, & \text{if } C_3 \end{cases} \tag{7.14}$$

To analyze the behavior of the avoidance task $\dot{\mathbf{q}}^i_\mathbf{a}$ near a joint limit, it is sufficient to replace $\dot{\mathbf{q}}_1$ by $\dot{\mathbf{q}}^*$, which will lead to the same behavior as studied in 7.2.4.

Finally, when more additional secondary tasks have to be considered, the avoidance task (7.14) can be used by setting:

$$\dot{\mathbf{q}}_* = \sum_{k=1}^{k_{max}} \dot{\mathbf{q}}_k \tag{7.15}$$

and the global control scheme is given by:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_* + \sum_{i=1}^{n} \dot{\mathbf{q}}^i_\mathbf{a} \tag{7.16}$$

7.2.5.2 Additional secondary tasks having several priority levels

When a global task consists of l tasks having different priorities, the avoidance task $\dot{\mathbf{q}}_{\mathbf{a}}^{i}$ is given by:

$$\dot{\mathbf{q}}_{\mathbf{a}}^{i} = \begin{cases} -(1+\lambda_i)\frac{|\dot{\mathbf{q}}_{\mathbf{l}}[i]|}{|(\mathbf{P}\mathbf{g}_i^{\ell})[i]|}\mathbf{P}\mathbf{g}_i^{\ell}, & \text{if } C_1 \\ -\lambda_{\ell i}(q_i)(1+\lambda_i)\frac{|\dot{\mathbf{q}}_{\mathbf{l}}[i]|}{|(\mathbf{P}\mathbf{g}_i^{\ell})[i]|}\mathbf{P}\mathbf{g}_i^{\ell}, & \text{if } C_2 \\ 0, & \text{if } C_3 \end{cases} \tag{7.17}$$

where $\dot{\mathbf{q}}_{\mathbf{l}}$ is the articular velocity to perform the execution of the l tasks with their required priority computed using (6.43). The global control scheme is:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_{\mathbf{l}} + \sum_{i=1}^{n} \dot{\mathbf{q}}_{\mathbf{a}}^{i} \tag{7.18}$$

Let us finally note that, as in the classical approaches, when several joints are not in their safe configuration, a potential problem may occur if motions required to avoid a limit of one joint move another joint nearer to one of its limits. However, the new avoidance approach is better than the classical one. Indeed, on one hand, the classical approach induces very high values for the avoidance task components $\dot{\mathbf{q}}_{\mathbf{a}}$ when a joint is very close to its limit. On the other hand, the new approach ensures a smooth behavior and decrease of the velocity $\dot{\mathbf{q}}_{\mathbf{a}}[i]$ when a joint nears its limit, which reduces the amplitude of the motions projected on the other joints, and thus the possibility that other joints near their limits due to these motions.

7.3 Integrating the new joint limits avoidance with kernel approach

In this section, we deal with the problem of avoiding several joint limits simultaneously. For that, we integrate the proposed activation and sign function and the tuning function previously presented in 7.2 as well as a new adaptive gain function in the method given in [Chaumette 01]. This ensures the smoothness of the injection of the avoidance task into the main task with a reasonable magnitude and correct direction. We also propose a method to consider additional secondary tasks to the proposed avoidance control system so that the final control scheme ensures that the joint limits will not be reached. Finally, we present a solution to discontinuity effect in that approach.

7.3.1 Gain vector $\mathbf{a}$

We recall that the i^{th} component of the control (5.56) can be written as:

$$\dot{\mathbf{q}}[i] = -\lambda(\mathbf{J}^{+}\,\mathbf{e}_1)[i] \;+\; \sum_{k=1}^{n_a} a_k\,\mathbf{E}_{ik} = \dot{\mathbf{q}}_{\mathbf{1}}[i] \;+\; \sum_{k=1}^{n_a} a_k\,\mathbf{E}_{ik} \tag{7.19}$$

where value of the gain vector $\mathbf{a}$ is computed to ensure that the control stops any motion of the joint j_i toward its limits.

Here, the value of the gain vector $\mathbf{a}$ is computed so that it ensures the same behavior obtained using our avoidance method proposed in Section 7.2. Let us consider that there are several joints within their critical intervals and near their limits. The global task is given by:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_1 + \dot{\mathbf{q}}_a \tag{7.20}$$

where $\dot{\mathbf{q}}_a$ is the new proposed avoidance scheme. It is defined by:

$$\dot{\mathbf{q}}_a = -\sum_{i=1}^{n} \mathbf{g}_i^{\ell} \;\; \lambda_{\ell i}(q_i) \;\; \bar{\lambda}_{sec_i} \tag{7.21}$$

where $\mathbf{g}_i^{\ell}$ is to activate the avoidance and to determine its direction, $\lambda_{\ell i}(q_i)$ is to allow a smooth behavior as explained in Sections 7.2.1 and 7.2.2 respectively. $\bar{\lambda}_{sec_i}$ is a new adaptive gain function to determine the magnitude of the avoidance task and is defined by:

$$\bar{\lambda}_{sec_i} = (1 + \lambda_i) \left| \dot{\mathbf{q}}_1[i] \right|, \quad \lambda_i \geq 0 \tag{7.22}$$

Our proposed method to determine $\mathbf{a}$ consists in defining the velocity $\dot{\mathbf{q}}[i]$ of the joint j_i as:

$$\dot{\mathbf{q}}[i] = \dot{\mathbf{q}}_1[i] + \dot{\mathbf{q}}_a[i] \tag{7.23}$$

To compute the vector $\mathbf{a}$ which ensures that the velocity for each joint near its limits is equal to $\dot{\mathbf{q}}[i]$ given by (7.23) and satisfies (7.19) we set:

$$\sum_{k=1}^{n_a} a_k \; \mathbf{E}_{ik} = \dot{\mathbf{q}}_a[i] = -\mathbf{g}_i^{\ell}[i] \; \lambda_{\ell i}(q_i)(1 + \lambda_i) \left| \dot{\mathbf{q}}_1[i] \right| \tag{7.24}$$

If there are several joints near their limits then the gain vector $\mathbf{a}$ that ensures (7.24) for all of those joints is the solution of the system of linear equation defined by:

$$\begin{bmatrix} \vdots \\ \mathbf{E}_{i\bullet} \\ \vdots \end{bmatrix} \mathbf{a} = \begin{bmatrix} \vdots \\ -\mathbf{g}_i^{\ell}[i] \; \lambda_{\ell i}(q_i)(1 + \lambda_i) \left| \dot{\mathbf{q}}_1[i] \right| \\ \vdots \end{bmatrix} \equiv \mathbf{A}\mathbf{a} = \mathbf{b} \tag{7.25}$$

The solution of (7.25) has the form $\mathbf{a}_0 = \mathbf{A}^{+}\mathbf{b}$. Finally, the control law will be:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_{\mathbf{1}} + \sum_{k=1}^{n_a} a_{0_k}^{*} \; \mathbf{E}_{\bullet \mathbf{k}} \tag{7.26}$$

The velocity given in (7.26) ensures a smooth behavior for each joint nears its joint limits when injecting the avoidance into the main task.

7.3.2 Gain vector a and additional secondary tasks

In this section we deal with the joint limit avoidance problem when some additional tasks should be considered. If these secondary tasks are given by $\mathbf{e}_2, \mathbf{e}_3, ..., \mathbf{e}_{d_{max}}$ and their velocity vectors are computed as $\dot{\mathbf{q}}_2, \dot{\mathbf{q}}_3, ..., \dot{\mathbf{q}}_{d_{max}}$ then the global task is given by:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_\mathbf{1} + \ \dot{\mathbf{q}}_a + \sum_{d=2}^{d_{max}} \dot{\mathbf{q}}_d \tag{7.27}$$

and the i^{th} component of $\dot{\mathbf{q}}$ will be:

$$\dot{\mathbf{q}}[i] = \dot{\mathbf{q}}_a[i] + \sum_{d=1}^{d_{max}} \dot{\mathbf{q}}_d[i] \tag{7.28}$$

In this case, the required avoidance task is given by:

$$\dot{\mathbf{q}}_a = -\sum_{i=1}^{n} \mathbf{g}_i^\ell \ \lambda_{\ell i}(q_i) \ \bar{\lambda}^a_{sec_i} \tag{7.29}$$

where $\bar{\lambda}^a_{sec_i}$ is defined by:

$$\bar{\lambda}^a_{sec_i} = (1+\lambda_i) \left| \sum_{d=1}^{d_{max}} \dot{\mathbf{q}}_d[i] \right|, \ \ \lambda_i \geq 0 \tag{7.30}$$

To compute the vector $\mathbf{a}$ and take into account the velocity components $\sum_{d=2}^{d_{max}} \dot{\mathbf{q}}_d[i]$ due to the additional secondary tasks we need to ensure that:

$$\sum_{i=1}^{n_a} a_i \ \mathbf{E}_{ik} = \dot{\mathbf{q}}_a[i] = -\mathbf{g}_i^\ell \ \lambda_{\ell i}(q_i)(1+\lambda_i) \left| \sum_{d=1}^{d_{max}} \dot{\mathbf{q}}_d[i] \right| \tag{7.31}$$

Again, if there are several joints near their limits, the gain vector $\mathbf{a}$ is computed by solving a system of linear equations constructed from (7.31) defined for each joint, so we get:

$$\begin{bmatrix} \vdots \\ \mathbf{E}_{i\bullet} \\ \vdots \end{bmatrix} \mathbf{a} = \begin{bmatrix} \vdots \\ -\mathbf{g}_i^\ell[i] \ \lambda_{\ell i}(q_i)(1+\lambda_i) \left| \sum_{d=1}^{d_{max}} \dot{\mathbf{q}}_d[i] \right| \\ \vdots \end{bmatrix} \equiv \mathbf{A}\mathbf{a} = \mathbf{b} \tag{7.32}$$

The system of linear equations that consider all joints near their limits is solved to obtain a vector $\mathbf{a}_1$. If the solution of (7.32) is $\mathbf{a}_1 = \mathbf{A}^+\mathbf{b}$, then the global control scheme will be:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_\mathbf{1} + \sum_{i=1}^{n_a} a^*_{1_i} \ \mathbf{E}_{\bullet \mathbf{i}} + \sum_{d=2}^{d_{max}} \dot{\mathbf{q}}_d \tag{7.33}$$

7.3.3 Transition switching inner loop to solve the discontinuity problem

In this section, we propose a solution to solve the discontinuity problem occurring in the iterative method when a new joint enters its critical situation or when a joint leaves its critical situation [Chaumette 01]. This solution consists in introducing a transition switching inner loop inside the visual servoing control loop.

Let us assume that the system velocity is given by $\dot{\mathbf{q}}^{t_{k-1}}$ during iteration $k-1$ of the visual servoing loop, where t_{k-1} is the starting time of iteration $k-1$. If the basic method is applied (see Section 5.4.3) and a new joint j_i enters its critical configuration during iteration $k-1$ the discontinuity in the system behavior occurs when the new computed velocity $\dot{\mathbf{q}}^{t_k}$ is sent to the robot controller at the beginning of iteration k, i.e. at time t_k as illustrated in Fig. 7.3.

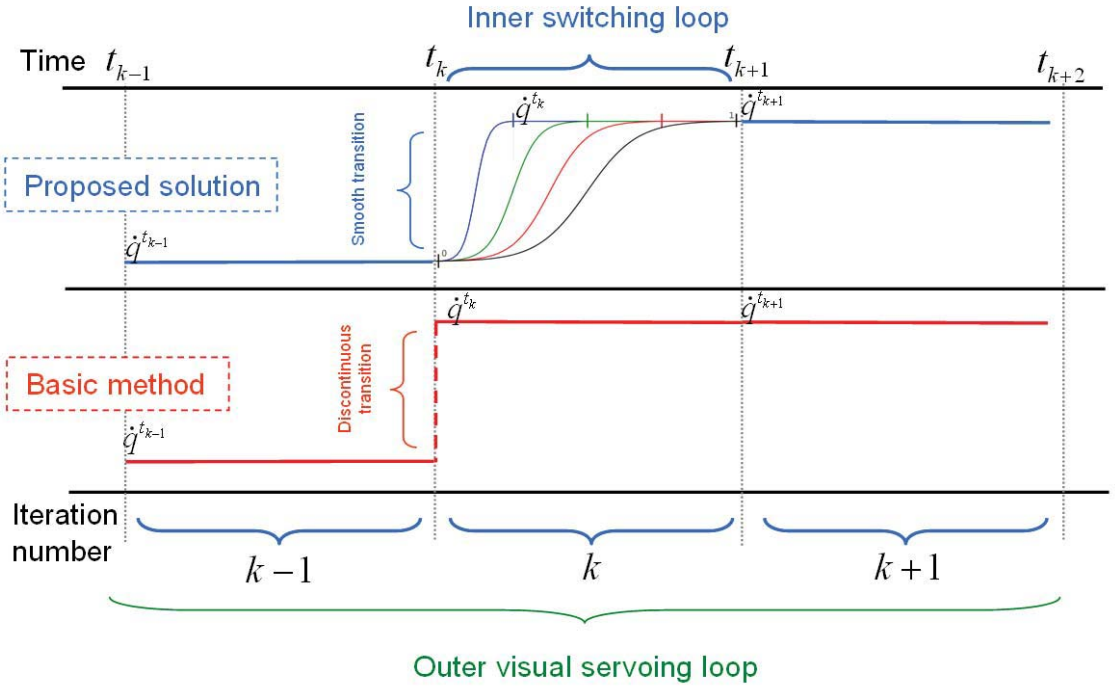


Figure 7.3 – Inner switchng loop

Instead of sending $\dot{\mathbf{q}}^{t_k}$ directly to the robot controller, we apply a transition switching from $\dot{\mathbf{q}}^{t_{k-1}}$ to $\dot{\mathbf{q}}^{t_k}$ using the following convex combination:

$$\dot{\mathbf{q}}^{\lambda} = (1 - \overline{\lambda}(n_{\text{iter}}))\dot{\mathbf{q}}^{t_{k-1}} + \overline{\lambda}(n_{\text{iter}})\dot{\mathbf{q}}^{t_k} \tag{7.34}$$

where n_{iter} is the iteration number of the inner loop and the switching function $\overline{\lambda}(n_{\text{iter}})$ can be defined as:

$$\overline{\lambda}(n_{\text{iter}}) = \frac{\lambda(n_{\text{iter}}) - \lambda(0)}{\lambda(N_{\text{iter}}) - \lambda(0)} \tag{7.35}$$

where

$$\lambda(n_{\text{iter}}) = \frac{1}{1+\exp(-12\frac{n_{\text{iter}}}{N_{\text{iter}}}+6)} \tag{7.36}$$

To ensure that the switching operation from $\dot{\mathbf{q}}^{t_{k-1}}$ to $\dot{\mathbf{q}}^{t_k}$ is performed during the iteration i_k of the outer loop, the number of iterations N_{iter} of the inner loop can easily be specified such that $N_{\text{iter}} < \frac{\Delta t_{\text{vs}}}{\Delta t_r}$, where Δt_{vs} is the time of a visual servoing iteration and Δt_r is the required time to execute a single velocity commend by the robot.

7.4 Results

Set A: New avoidance strategy

The joint limits avoidance strategy proposed in Section 7.2 has been applied to a six DOF Cartesian robot, see Fig. 3.8, for a full rank main task using obj_1. The values of the lower and upper limits for each joint are $q_1^{\min} = -0.7$, $q_1^{\max} = 0.7$, $q_2^{\min} = -0.6$, $q_2^{\max} = 0.63$, $q_3^{\min} = -0.5$, $q_3^{\max} = 0.46$, $q_4^{\min} = -156.41$, $q_4^{\max} = 156.0$, $q_5^{\min} = -5.73$, $q_5^{\max} = 142.0$, $q_6^{\min} = -91.0$, and $q_6^{\max} = 91.0$, where q_1, q_2 and q_3 are in meter and q_4, q_5 and q_6 are in degrees. The initial pose between the camera and the object has been chosen such that the camera is at a distance of 0.5 m from obj_1 and the optical axis of the camera passes vertically through the center of obj_1 such that the initial camera pose is (0,0,0.5,0,0,130) for cases {A1, A2} and (0,0,0.5,0,0,100) for case A3. The desired camera pose is (0,0,0.5,0,0,0), which corresponds to a pure rotation of 130 and 100 degrees around the camera optical axis respectively. These initial and desired configurations have been selected such that the main task causes the robot to perform a backward motion when Cartesian image points coordinates are selected as visual features. By setting the initial configuration of the robot system such that the optical axis of the camera is parallel to the joint j_3 of the robot, a limit of the joint j_3 is reached.

All limit thresholds $q_{\ell_0 i}^{\min}$ and $q_{\ell_1 i}^{\min}$ are computed using the parameters $\rho = 0.1$ and $\rho_1 = 0.5$ as explained in Section 7.2 except for the lower limit of the 5^{th} joint. Indeed, since the articular position for the initial configuration of the robot for our experiment is $\mathbf{q} = (-0.51, 0.09, -0.02, -128.89, 1.46, 0.01)$, we set $q_{\ell_0 5}^{\min} = -2$ and $q_{\ell_1 5}^{\min} = -5$ so that j_5 is not in a critical situation at the beginning of the experiments. The avoidance and secondary tasks describing a trajectory to follow in the articular space are projected using the new projection operator $\mathbf{P}_\lambda$. Let us recall that it is impossible to consider joint limits avoidance with the classical projection operator $\mathbf{P_e}$ in all cases presented here, since $\mathbf{P_e} = \mathbf{0}$.

7.4.1 Case A1: Single joint limits avoidance

In this case, the system starts its movement as expected from control (7.1) by a backward motion. When the threshold value $q_{\ell_0 3}^{\min}$ is reached, the avoidance is activated for the joint j_3 and the robot avoids reaching $q_3^{\min}$. As depicted in Figure 7.4, $\dot{\mathbf{q}}_\mathbf{a}$ computed using (7.3)

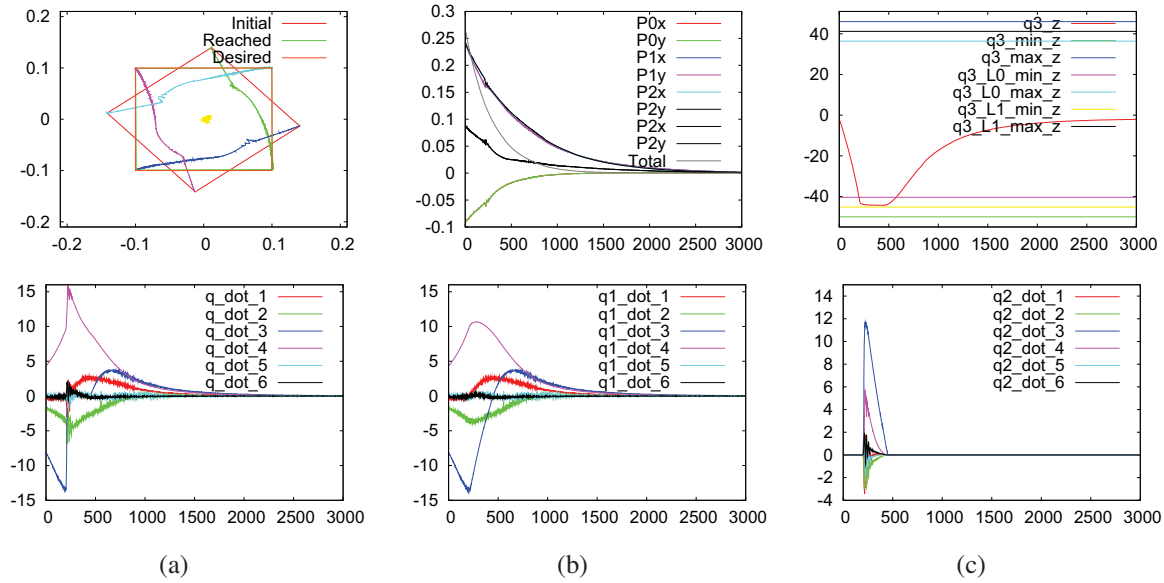


Figure 7.4 – Results for cases A1; Single joint limit avoidance; Line 1 (a,b,c): image points trajectories, image point error and articular values and joint limits of robot joint q_3;; Line 2 (a,b,c): articular velocity components in cm/s and deg/s of the general task $\dot{\mathbf{q}}$, main task $\dot{\mathbf{q}}_\mathbf{1}$ (high level priority) and secondary task $\dot{\mathbf{q}}_\mathbf{2}$ (low level priority) respectively.

starts to have a non-zero value in the opposite direction of the main task within the iteration interval (250,420). During this interval, the velocity component of the global task $\dot{\mathbf{q}}$ for the joint j_3 is approximately equal to zero because of the effect of $\dot{\mathbf{q}}_\mathbf{a}$, while the rotation around the revolute joint j_4, which corresponds to the camera optical axis, is performed by the main task. Then, the direction of $\dot{\mathbf{q}}_\mathbf{1}[3]$ of the main task is inverted near to iteration number 420, so that it causes the robot to move away from its joint limit, that is why $\dot{\mathbf{q}}_\mathbf{a}$ stops since it becomes useless, although the value of $q_3 \in [q_{\ell_0 3}^{\min}, q_3^{\min}]$ within the iteration interval (420,500). This can be seen in Figure 7.4 by noticing the values of the third component of $\dot{\mathbf{q}}_\mathbf{a}$ and q_3. Finally, the system converges to its desired position while ensuring the exponential decreasing for $\|\mathbf{e}\|$ all along the task.

7.4.2 Case A2: Additional secondary task sent to revolute joint, classical task-priority

Now, we consider the case when an additional task is considered describing a trajectory to follow in the articular space inducing an articular velocity $\mathbf{g}$. A sinusoidal motion defined by $f(t) = \frac{-3\pi}{180}(\sin(\frac{\pi t}{180}))$ on the revolute joint j_5 is chosen as additional secondary task causing two joints to near their limits simultaneously. Through the first 360 iterations, the vector of this secondary task is given by $\mathbf{g} = (0, 0, 0, 0, f(t), 0)$ where t is the number of iterations. Control scheme (7.11) is used where $\dot{\mathbf{q}}_\mathbf{a}$ is given by (7.14) and $\dot{\mathbf{q}}_\mathbf{2} = \mathbf{g}$. The obtained results depicted on Fig. 7.5, in which q2_dot = $\dot{\mathbf{q}}_\mathbf{a} + \dot{\mathbf{q}}_\mathbf{2}$, show that the system succeeds to avoid two joint limits simultaneously, a joint limit of j_3 at $q_{\ell_0 3}^{\min}$ due to the main task and a joint limit

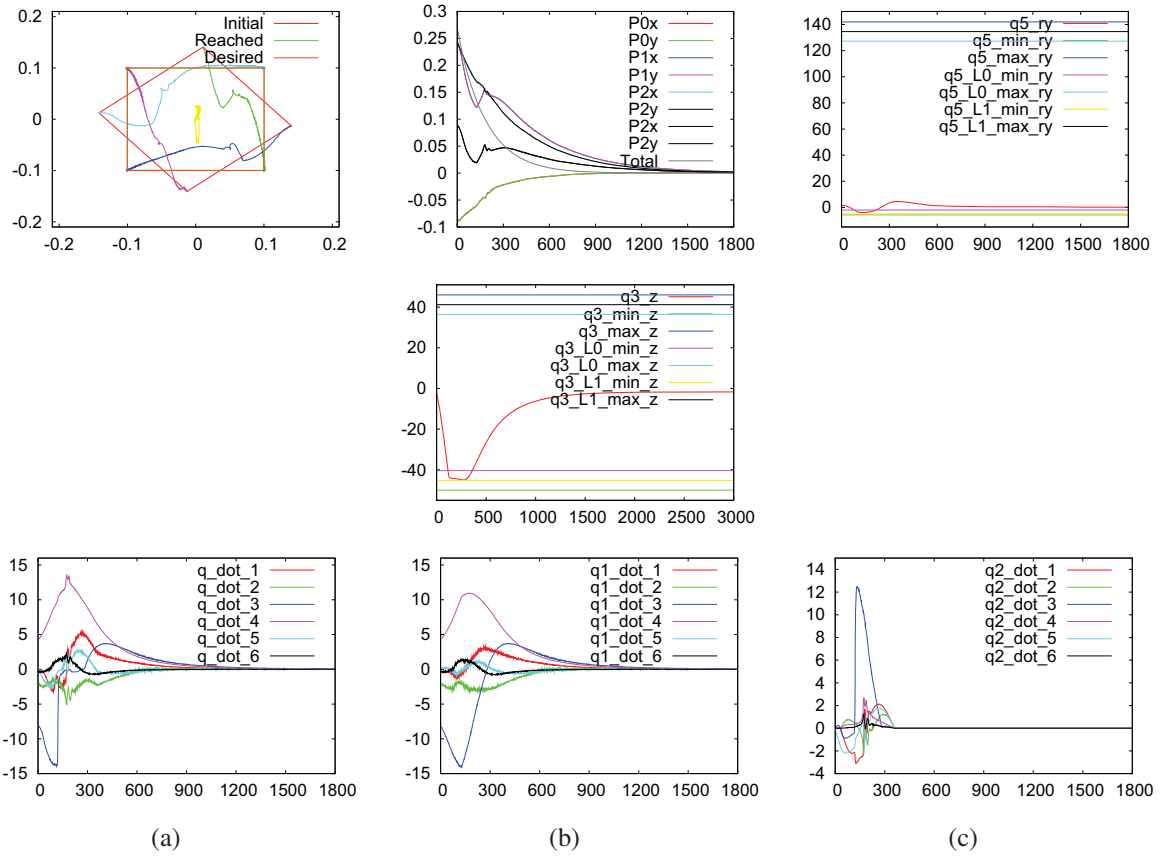


Figure 7.5 – Results for cases A2; Classical priority; Two joint limits avoidance; Additional secondary task sent to a revolute joint q_5; Line 1 (a,b,c): image points trajectories, image point error and articular values and joint limits of robot joint q_5;; Line 2 (a,b,c): articular velocity components in cm/s and deg/s of the general task $\dot{\mathbf{q}}$, main task $\dot{\mathbf{q}}_\mathbf{1}$ (high level priority) and secondary task $\dot{\mathbf{q}}_\mathbf{2}$ (low level priority) respectively; q2_dot = $\dot{\mathbf{q}}_\mathbf{a} + \dot{\mathbf{q}}_\mathbf{2}$

of j_5 at $q^{\min}_{\ell_0 5}$ due to the additional task. Again, as illustrated in Fig. 7.5, the system keeps the exponential decreasing of the norm of the total error while projecting the additional secondary task $\mathbf{g}$ and avoiding the joint limits.

7.4.3 Case A3: Additional secondary task sent to revolute joint, efficient task-priority

We now consider the efficient-task priority approach with the same additional secondary task as before. The control scheme to manage $\mathbf{e_1}$ and $\mathbf{e_2}$ is similar to (6.58) except that it is defined in the articular space. Using $\bar{\mathbf{I}}$ in (6.58) indicates that, if the $i_0{}^{th}$ joint j_{i_o} is not used by $\mathbf{g}$, then $\mathbf{g}[i_0]$ will not be modified by $\dot{\mathbf{q}}_\mathbf{1}[i_0]$ before being projected by $\left(\bar{\mathbf{I}}_\mathbf{n}\mathbf{P}_{*_\mathbf{1}}\right)^+$. This prevents any unnecessary motions when $\dot{\mathbf{q}}_\mathbf{1}[i_0] \neq 0$. In this case, the higher level priority task is the visual servoing task while the lower level priority additional task is given by $\mathbf{g} = (0, 0, 0, 0, f(t), 0)$ where $f(t) = \frac{3\pi}{180}(\sin(\frac{\pi t}{180}))$. It is activated during the first 360 iterations. The initial camera

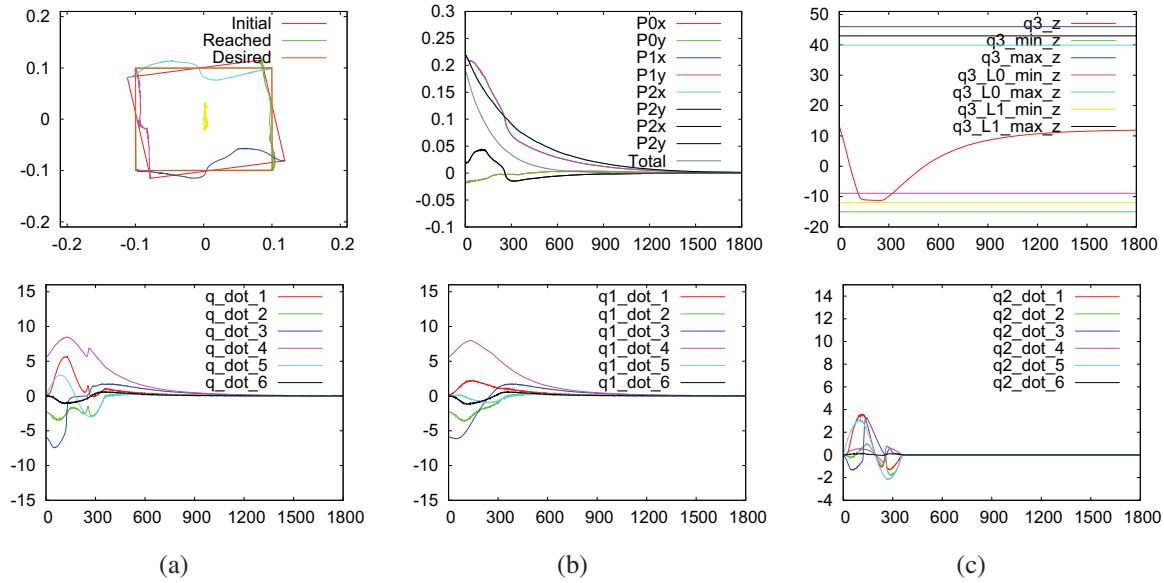
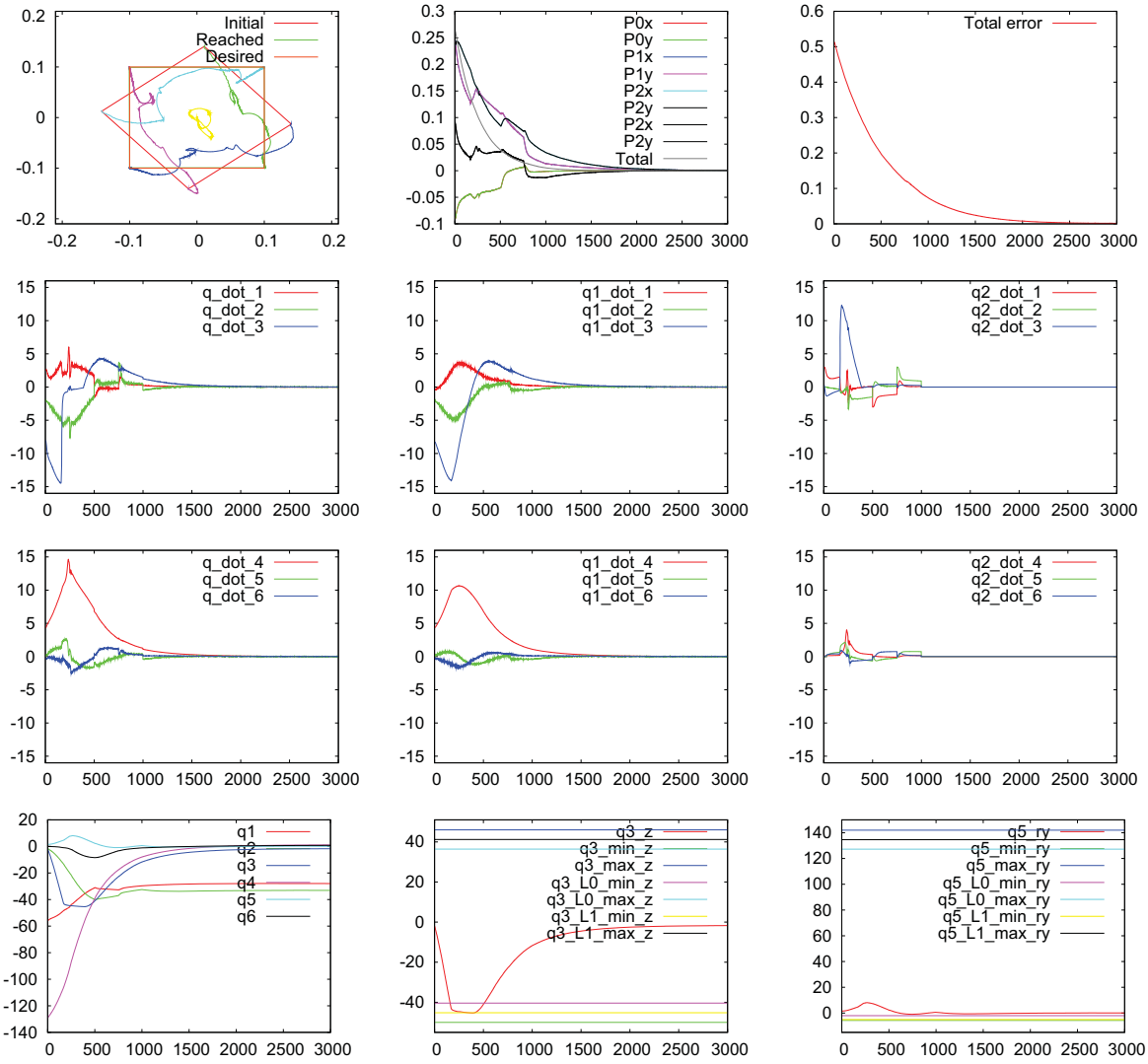


Figure 7.6 – Results for cases A3; Classical priority; Avoidance and additional secondary task sent to a revolute joint q_5; Line 1 (a,b,c): image points trajectories, image point error and articular values and joint limits of robot joint q_5;; Line 2 (a,b,c): articular velocity components in cm/s and deg/s of the general task $\dot{\mathbf{q}}$, main task $\dot{\mathbf{q}}_\mathbf{1}$ (high level priority) and secondary task $\dot{\mathbf{q}}_\mathbf{2}$ (low level priority) respectively; q2_dot = $\dot{\mathbf{q}}_\mathbf{a} + \dot{\mathbf{q}}_\mathbf{2}$

pose is (0,0,0.5,0,0,100). A lower joint limit for j_3 is virtually considered to be 15 degrees. Applying (6.58) on $\dot{\mathbf{q}}_\mathbf{1}$ and $\mathbf{g}$ gives $\dot{\mathbf{q}}_\mathbf{l}$ which is used in (7.14) to compute the avoidance task. The obtained results depicted on Fig. 7.6 show that the system succeeds to avoid j_3 at $q_{\ell 03}^{\min}$ and to keep the exponential decreasing of $\|\mathbf{e}\|$ while projecting the articular velocity task $\mathbf{g}$ using the efficient redundancy-based task-priority approach. As depicted in Fig 7.6, the motion required by $\mathbf{g}[4]$ appears on the corresponding component $\mathbf{q}[4]$ of the global task.

7.4.4 Case A4: Additional secondary task sent to prismatic joint, classical task-priority

In this case, a motion that tries to move the end effector in a square of length 0.03 in xy-plane of the articular frame is specified as additional secondary task. The vector $\mathbf{g_3}$ that produces this motion is defined as $\mathbf{g} = (0.03, 0, 0, 0, 0, 0)$ when $(1 < iter < 250)$, $\mathbf{g} = (0, -0.03, 0, 0, 0, 0)$ when $(250 < iter < 500)$, $\mathbf{g} = (-0.03, 0, 0, 0, 0, 0)$ when $(500 < iter < 750)$, $\mathbf{g} = (0, 0.03, 0, 0, 0, 0)$ when $(750 < iter < 1000)$, else $\mathbf{g_3} = (0, 0, 0, 0, 0, 0)$. It concerns only the two prismatic joints j_1 and j_2. As depicted in Fig. 7.7, q2_dot represents the avoidance task combined with the projection of $\mathbf{g_3}$. As expected, secondary motions along x-axis and y-axis are produced during the first 1000 iterations due to the projection of the vector $\mathbf{g_3}$. It is clear that even if the error of each feature lost its exponential decreasing behavior mainly due to the effect of $\dot{\mathbf{q}}_\mathbf{3}$, the servo system keeps the exponential decreasing for the norm of the total error, as expected thanks to $\mathbf{P}_{\|\mathbf{e}\|}$. Again, the visual servoing task is



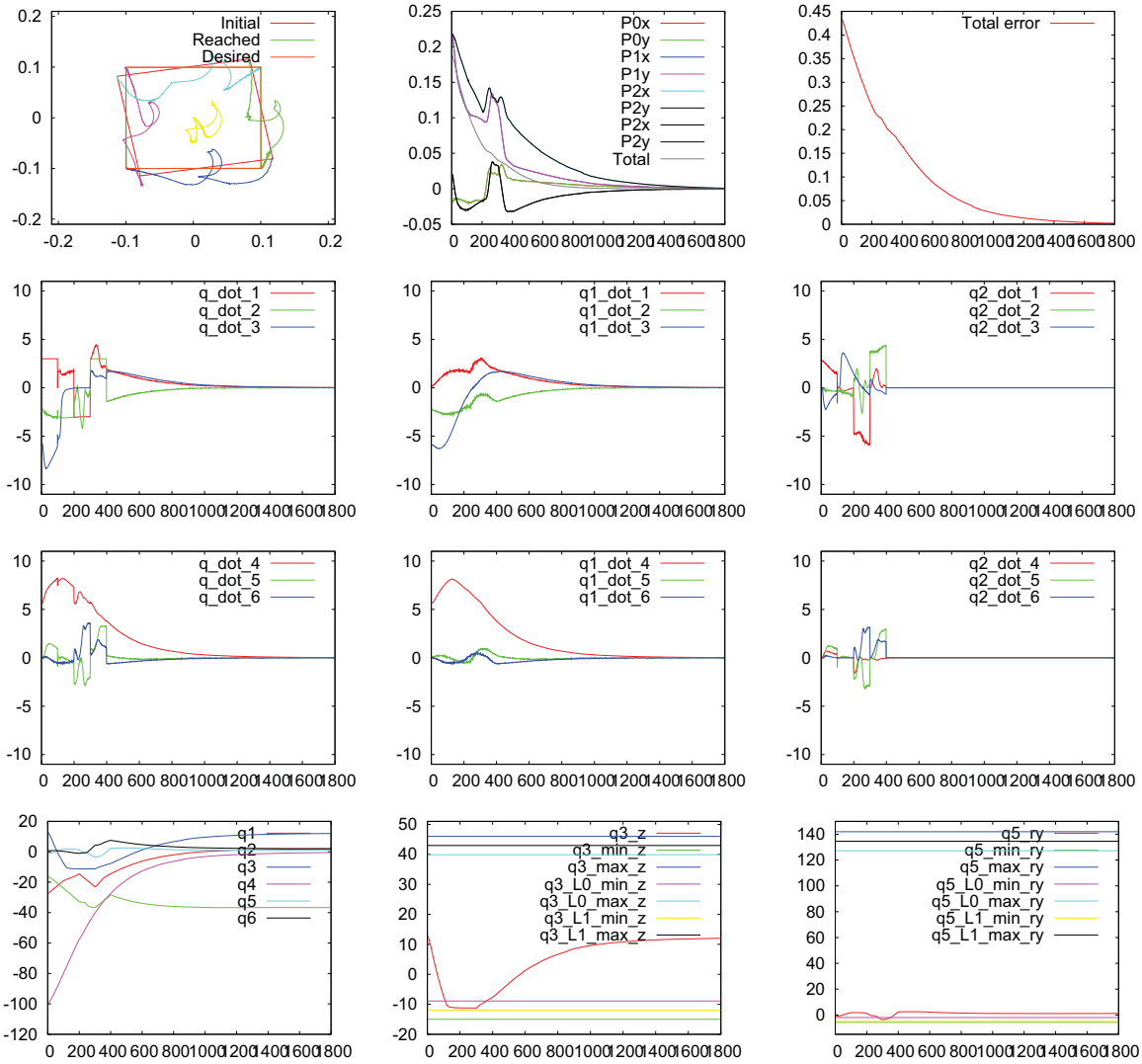
(a) (b) (c)

Figure 7.7 – Results for case A4: Avoidance and additional secondary task sent to translational joints; q2_dot = $\dot{\mathbf{q}}_2 + \dot{\mathbf{q}}_3$, Line 1 (a,b,c): image points trajectories, image point error and the norm of the total error; Line 2 (a,b,c) and Line 3 (a,b,c): articular velocity components for translational and rotational robot joints respectively in cm/s and deg/s of the general task $\dot{\mathbf{q}}$, main task $\dot{\mathbf{q}}_1$ (high level priority) and secondary task $\dot{\mathbf{q}}_2$ (low level priority) respectively; Line 4 (a,b,c): articular values for the six joints, joint limits of robot joints q_3 and q_5;

performed successfully by avoiding the joint limit of j_3.

7.4.5 Case A5: Additional secondary task sent to prismatic joint, efficient task-priority

In this case, as in case A3, a lower joint limit for j_3 is virtually considered to be 15 degrees. Secondary task similar to A4 is considered. As depicted in Fig. 7.8, q2_dot represents



(a) (b) (c)

Figure 7.8 – Results for case A5, Avoidance and additional secondary task sent to a revolute joint; q2_dot = $\dot{\mathbf{q}}_2 + \dot{\mathbf{q}}_3$ where $\dot{\mathbf{q}}_2$ considers task priority

the avoidance task combined with the projection of g. As expected, Fig. 7.8 shows that secondary motions along x-axis and y-axis are produced during the first 400 iterations due to considering the priority when projecting the vector g. As expected, the servo system keeps the exponential decreasing for the norm of the total error. The joint limit of j_3 is avoided and the visual servoing task is performed successfully.

Set B: New avoidance strategy integrated with kernel approach

In this set of experiments, the new avoidance strategy integrated with the kernel approach to deal with the problem of avoiding several joint limits simultaneously is implemented to a six DOFs viper robot arm (see Fig. 7.9). The lower and upper limits for each joint have the following values: $q_1^{\min} = -50$, $q_1^{\max} = 50$, $q_2^{\min} = -135$, $q_2^{\max} = -40$, $q_3^{\min} = 40$, $q_3^{\max} = 215$, $q_4^{\min} = -190$, $q_4^{\max} = 190$, $q_5^{\min} = -110$, $q_5^{\max} = 110$, $q_6^{\min} = -184.125$, and $q_6^{\max} = 184.125$, where q_1, q_2, q_3, q_4, q_5 and q_6 are in degrees. Similar to set A, we set $\rho = 0.1$ and $\rho_1 = 0.5$ to compute $q_{\ell_0 i}^{\min}$ and $q_{\ell_1 i}^{\min}$. The features considered by the main task are the image coordinates of a moving point. This means that only two DOFs of the robot are constrained by the main task. Three experiments are performed and their results are presented in cases B1, B2, and B3. In all cases, the motion of the point is the same. It has been designed so that the robot moves toward several of its joint limits.

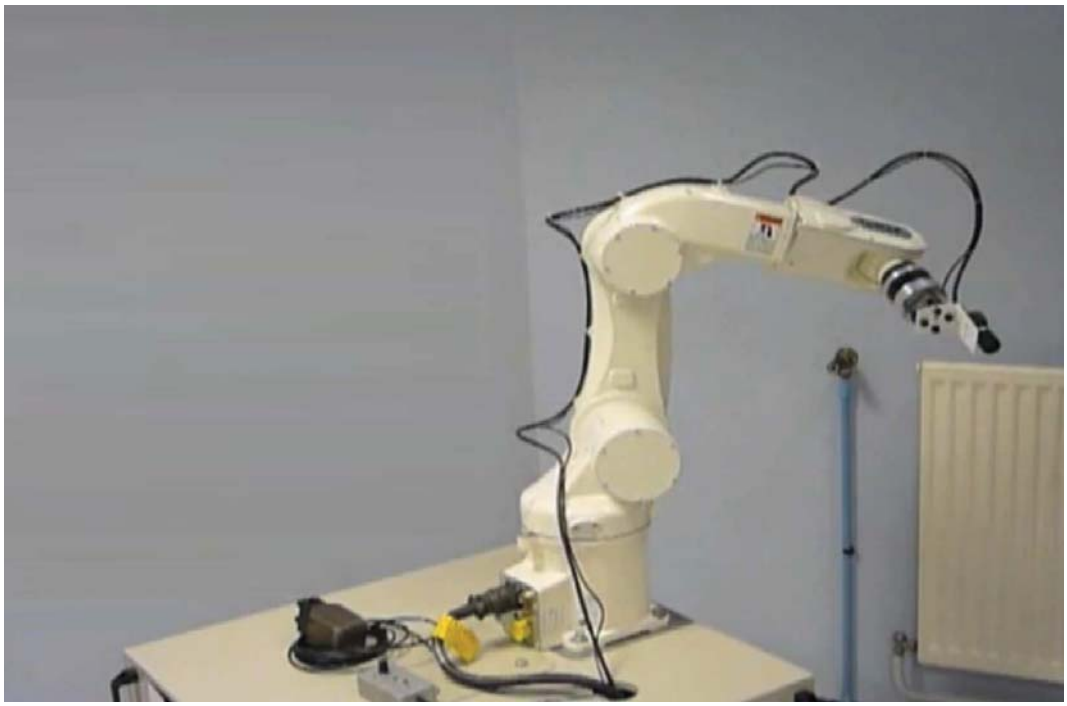


Figure 7.9 – Experimental system: Robot arm

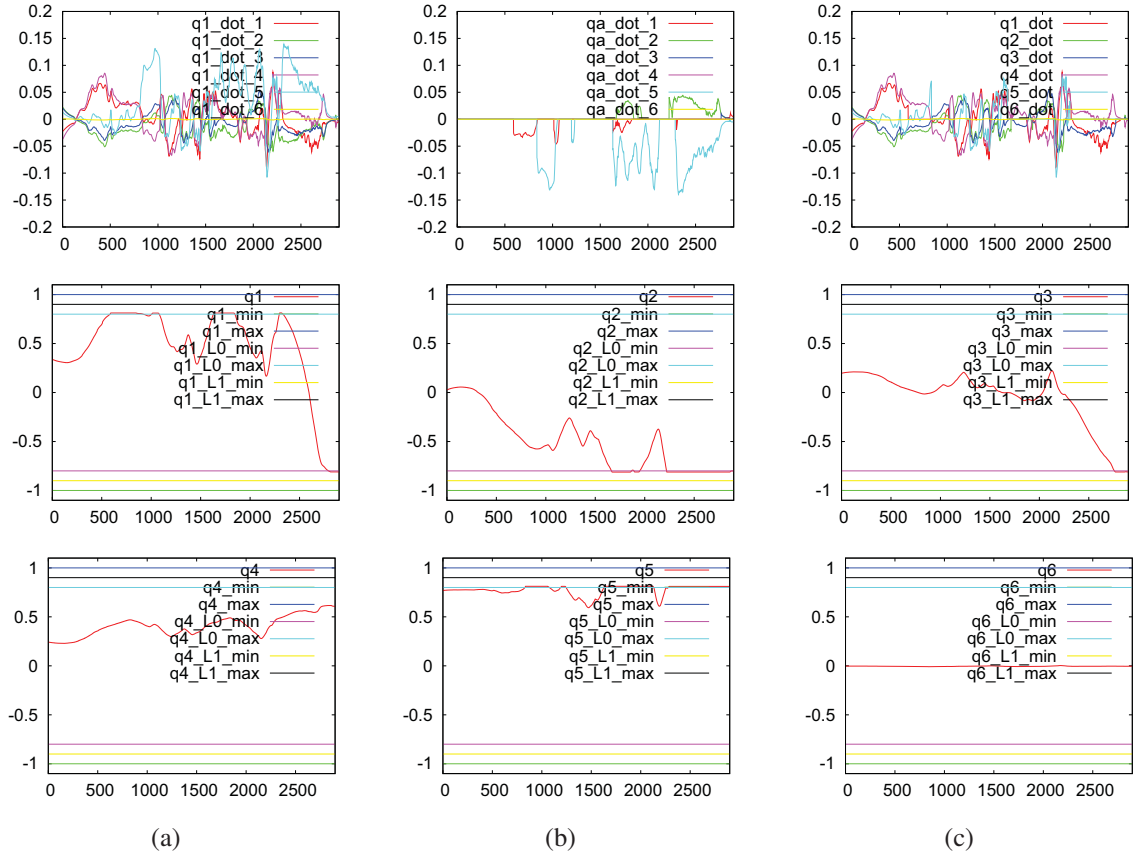


Figure 7.10 – Results for case B1; Kernel approach, Basic method; Line 1 (a,b,c): articular velocity components in rad/s of the main task $\dot{\mathbf{q}}_1$, avoidance task $\dot{\mathbf{q}}_a$ and general task $\dot{\mathbf{q}}$ respectively; q_dot = $\dot{\mathbf{q}}_1 + \dot{\mathbf{q}}_a$; Line 2 (a,b,c) and Line 3 (a,b,c) are the normalized articular values and joint limits of the six robot joints q_1, q_2, q_3, q_4, q_5, q_6 respectively.

7.4.6 Case B1: Basic kernel approach

In this first case, the basic method described in Section 5.4.3 is used to avoid the joint limits. We recall that $\sum_{k=1}^{n_a} a_k \, \mathbf{E}_{ik}$ is defined by (5.58). As can be seen in Fig. 7.10, as soon as the values of the joints q_1, q_2, q_3 and q_5 approach their thresholds $q_{\ell_0 1}^{\max}$ or $q_{\ell_0 1}^{\min}$, $q_{\ell_0 2}^{\min}$, $q_{\ell_0 3}^{\min}$ and $q_{\ell_0 5}^{\max}$ respectively, the avoidance task is activated causing the corresponding joint to suddenly stop any motion toward its nearest limits.

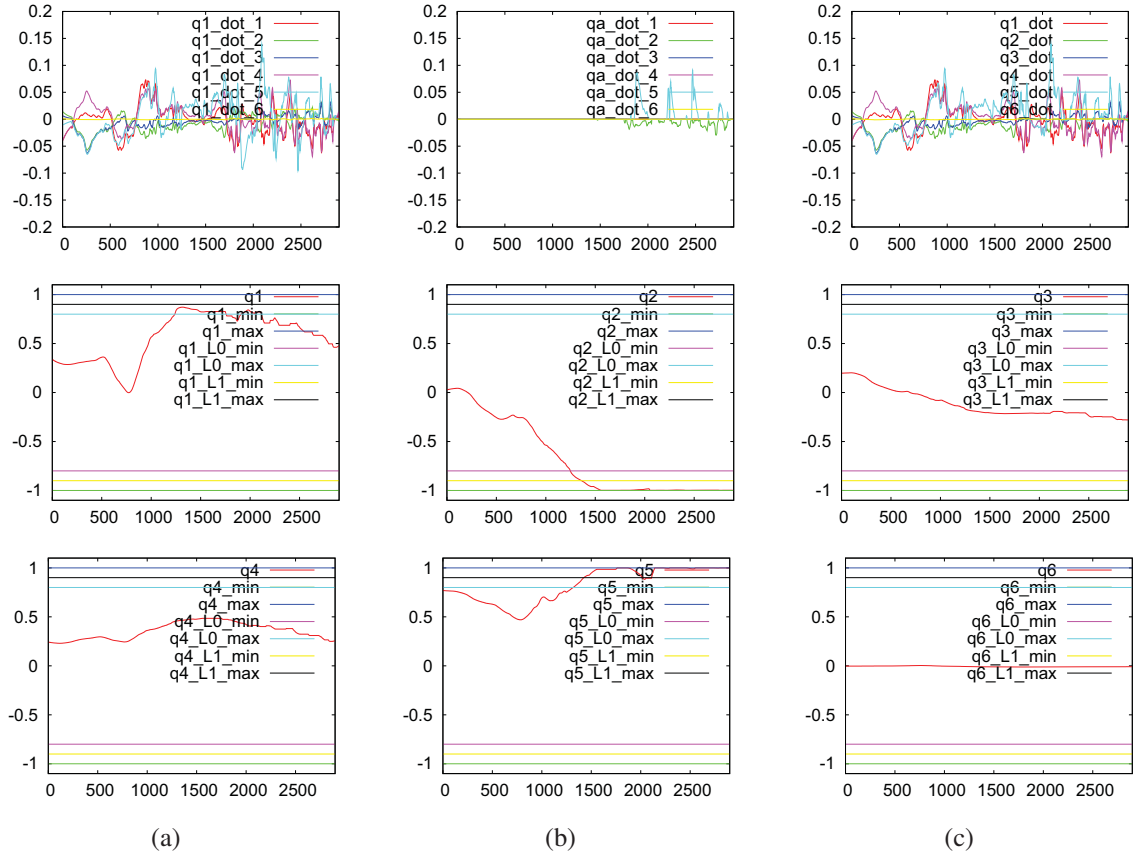
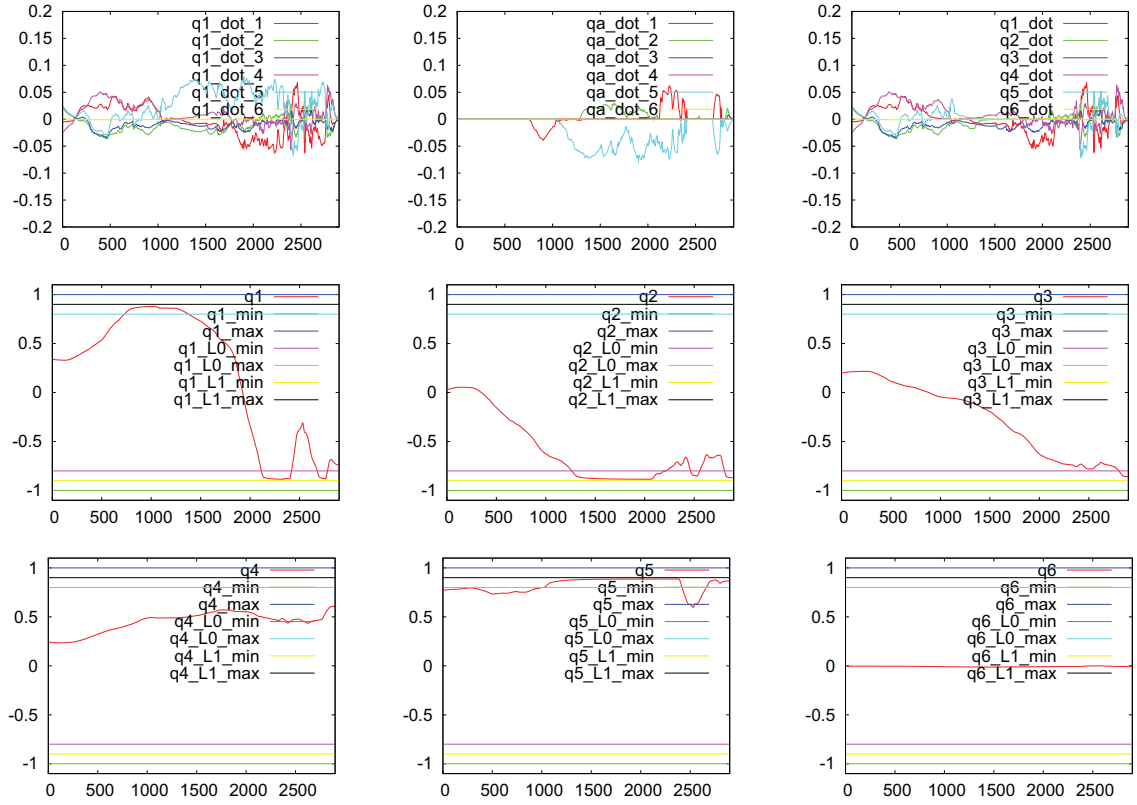


Figure 7.11 – Results for case B2, kernel approach, basic kernel approach with gamma coefficients

7.4.7 Case B2: Basic kernel approach with gamma coefficients

The basic kernel approach uses the coefficients γ_i defined in Section 5.4.3. As can be seen in Fig. 7.11, using γ_i allows to smooth the initial value of the avoidance task component $\mathbf{q_a}[i]$ when a joint j_i approaches its threshold limits $q_{\ell_0 i}^{\min}$ or $q_{\ell_0 i}^{\max}$. It allows to avoid the discontinuities appearing in previous case B1. However, a discontinuity is still obtained when a joint becomes very close to its limits $q_i^{\min}$ or $q_i^{\max}$, i.e. when $\gamma_i \approx 1$. As can be seen in Fig. 7.11, when j_2 and j_5 approach $q_2^{\max}$ and $q_5^{\min}$ respectively, a discontinuity is obtained in the behavior of the joint j_1.



(a) (b) (c)

Figure 7.12 – Results for case B3, new joint limit avoidance integerated with kernel approach

7.4.8 Case B3: New joint limits avoidance strategy integrated to kernel approach

In this case, the results of the new avoidance strategy coupled with the kernel approach, as proposed in Section 7.3, are presented. As can be seen in Fig. 7.12, a smooth behavior for all joints is obtained when the joints near their limits. The avoidance task component is smoothly activated as soon as a joint q_i nears any of its thresholds $q_{\ell_0 i}^{\min}$ or $q_{\ell_0 i}^{\max}$. Using the new strategy ensures that the magnitudes for each component of the avoidance task compensate the main task components. This ensures that no joint will reach its limit.

7.5 Conclusion

A new avoidance scheme for avoiding joint limits has been presented based on three functions proposed: an activation function $\mathbf{g}_i^{\ell}$ that activates the avoidance task and sets the direction of its actions; adaptive gain functions λ_{sec_i} and $\lambda_{sec_i}^{a}$ that controls the magnitude of the avoidance task; and a tuning function $\overline{\lambda}_{\ell_i}(q^i)$ that ensures the smoothness of the injection of the avoidance task into the main task. Considering several additional secondary tasks has also been proposed. The new avoidance strategy has also been integrated with the kernel approaches to avoid several joint limits simultaneously.

The avoidance method proposed in Section 7.2 has been implemented and validated on a six DOFs robot arm having three prismatic and three rovolute joints as illustrated in 3.8. The investigation of the new large projection operator $\mathbf{P}_{\lambda}$ has been performed by employing $\mathbf{P}_{\lambda}$ to deal with the problem of robot joint limit avoidance and projecting additional secondary tasks. A nice behavior has been obtained that makes the system avoid the joint limits very smoothly even when the main task constrains all the robot degrees of freedom and when additional secondary tasks are considered. The main advantage of the new avoidance method over classical gradient projection method is that the magnitude of the self motion is automatically computed without any tuning step.

The integration of the new avoidance strategy with the kernel approach has been implemented on another six DOFs robot platform. This approach allows the system to automatically generate a suitable magnitude for each component of the avoidance task such that it is greater than the magnitude of the main task near robot joint limits.

Chapter 8

General conclusions

In this chapter, we present the general conclusions of this work. The research described in this thesis has been concerned with two main issues in sensor-based robot control. The first issue, presented in Part I, concerned the domain of visual servoing, while the second issue, presented in part II, concerned the domain of redundancy and avoidance in robot control.

Contributions

The works presented in Part I and II provide several contributions as illustrated in Fig. 8.1. For the main tasks, we can employ the new first order control scheme to obtain a better behavior for a selected value of the behavior controller that can avoid local minima configurations or employ the new second order control schemes to approach a singular configurations and achieve more accurate positioning tasks in general. The new projection operator can be used for adding supplementary tasks and/or avoidance tasks. For example, secondary tasks can be defined as the proposed joint limits avoidance strategies. The contributions are presented in two parts as to the organization of the dissertation.

Part I: Visual servoing

The control laws used in 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. Different behaviors may explain these failures. For example, the camera moves to infinity, the camera moves to be too near to the object, the system reaches a local minimum or a singular configuration. The first part of this thesis aimed towards more robust control schemes. This was achieved by introducing two new control schemes. None of the classical control schemes allow to enhance its behavior against a given problem of local minima or singularity. The classical solutions for these situations

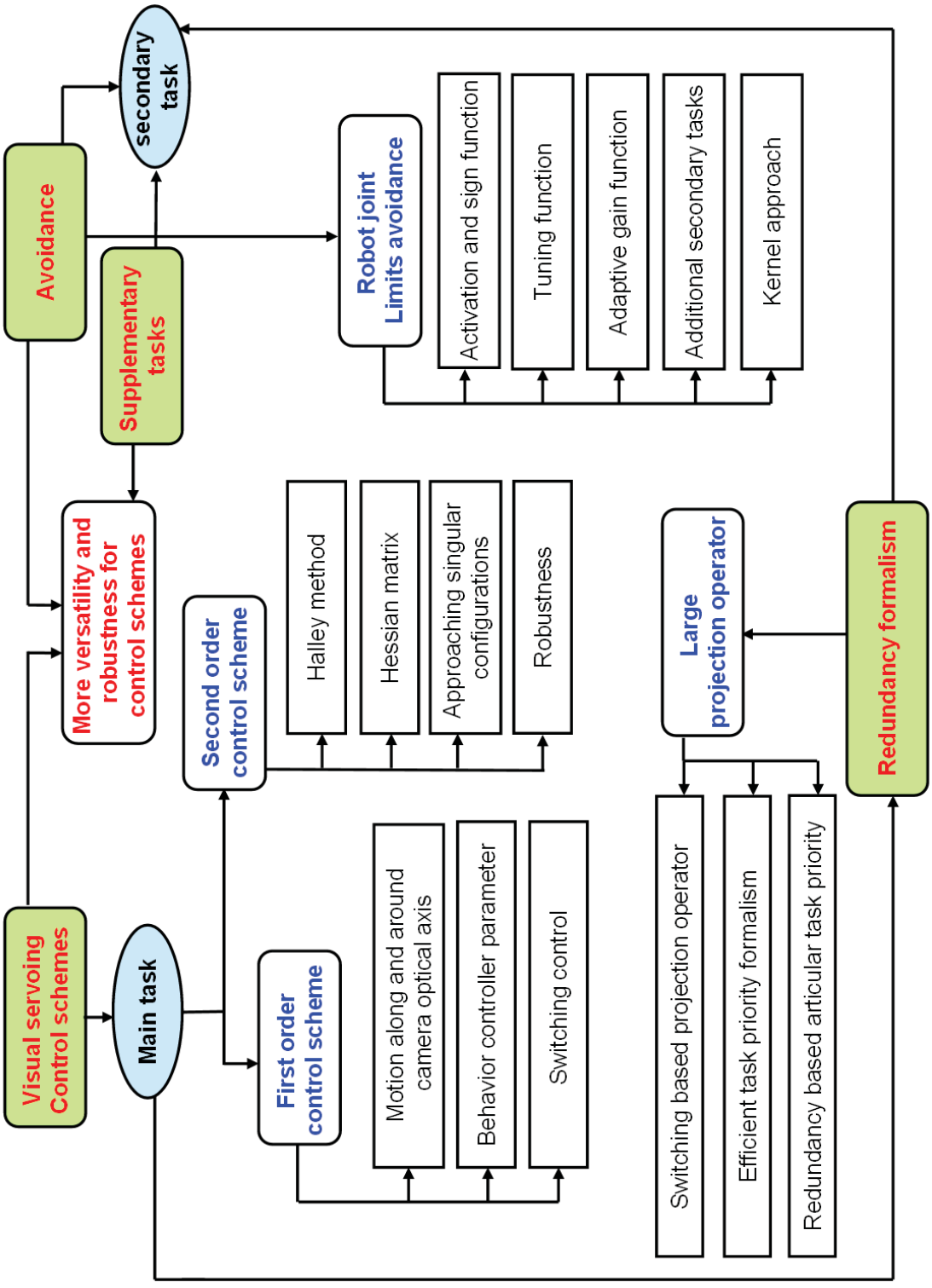


Figure 8.1 – Thesis conclusions

as mentioned in Chapter 2 were by utilizing sliding approaches, partitioning approaches, trajectory planning approaches, or origin-shift approaches. Solutions also were provided by defining hybrid and/or switching approaches among IBVS, PBVS and other approaches. Finally, these different solutions allow to obtain a better control behavior for a given configuration.

First order control scheme: The first proposed control law is based on introducing a behavior controller β to the control matrix as presented in Chapter 3. The control matrix is obtained based on the convex combination between the interaction matrices computed at the current and the desired configurations after introducing the parameter β. Setting $\beta = 0, 1,$ or $1/2$ allows to switch between the three most classical schemes. Another control law has been introduced to try to obtain the global stability of the system. We presented also an analytical analysis of the new control schemes and of the most common ones in image-based visual servoing when a required camera displacement is a combination of a translation and a rotation with respect to the camera optical axis. We perform this analysis for all possible values of the behavior parameter β. The analysis exhibited, for the first time as far as we know a singularity of the control scheme proposed in [Malis 04] when a pure rotation of degrees 90 around the camera optical axis is required. The experimental analysis also exhibited a local minimum configuration for all classical control schemes. New surprising results have also been obtained for the other classical control schemes for motion combining translation along and rotation around the optical axis. In all considered cases (difficult configurations subject to local minima for all classical schemes, motion along and around the optical axis), a satisfactory behavior of the control scheme can be obtained for some values of this parameter, which demonstrate the superiority of the proposed control law with a behavior controller over all other classical control laws.

Second order control schemes: Dealing with singularity configuration is one of the most critical problems in IBVS. In Chapter 4, we have been interested by the difficult problem of reaching a visual singular configuration where all classical control schemes have been shown to be unsatisfactory. To try to improve the behavior of the system near the desired singular position, we proposed a new second order control schemes based on the second order Taylor expansion similar to Halley's method minimization. The experimental results obtained have shown that it is possible to improve the accuracy of the positioning using one of these control schemes.

Part II: Redundancy and avoidance

Redundancy formalism: We presented in Chapter 6 theoretical developments of a new large projection operator for the redundancy framework by considering the norm of the total error. The redundancy framework proposed is built using this projection operator. The main contribution here is that the new projection operator enlarges the applicability domain of the task-redundancy in robot control to all problems that can be solved by adding system constraints. It can also be used to manage the levels of priority among several tasks, such as the

stack of tasks. A switching strategy has been proposed to ensure that the new projection operator smoothly switches to the classical projection operator as soon as the error nears zero. An adaptive gain has also been developed so that the secondary task is effective during a long period. That may be useful when the secondary tasks have really to be considered (which is the case for instance for joints limits and obstacle avoidance). The main interest of the new projection operator is that it is always at least of rank $n-1$. Hence it can be used even if the main task is full rank. The new projection operator has been employed within two different redundancy-based task-priority approaches. A comparative study of results obtained using $\mathbf{P}_\lambda$ and $\mathbf{P}_\mathbf{e}$ shows that $\mathbf{P}_\lambda$ provides more versatility when the main task is not full rank. The effectiveness of the new projection operator has been addressed when the task is defined in the camera space or in the articular space.

Joint limit avoidance strategies: A new avoidance scheme for avoiding joint limits has been presented in Chapter 7 based on three functions proposed: an activation function $\mathbf{g}_i^\ell$ that activates the avoidance task and sets the direction of its actions; adaptive gain functions λ_{sec_i} and $\lambda_{sec_i}^a$ that controls the magnitude of the avoidance task; and a tuning function $\overline{\lambda}_{\ell_i}(q^i)$ that ensures the smoothness of the injection of the avoidance task into the main task. Considering several additional secondary tasks has also been proposed. The new avoidance strategy has been integrated with the kernel approach to control the magnitude and ensure smoothness of the avoidance task when several joint limits have to be considered simultaneously. As for the discontinuity problem of the iterative method that occurs when a new joint enters its critical situation, a solution proposed is by introducing an inner switching loop inside the visual servoing loop to ensure smooth transition when needed. The avoidance method proposed in this work has been implemented and validated on a six DOFs robot manipulator and the investigation of the new large projection operator $\mathbf{P}_{\|\mathbf{e}\|}$ has been performed by employing it to deal with the problem of robot joint limit avoidance and projecting additional secondary tasks. A nice behavior has been obtained that makes the system avoid the joint limits very smoothly even when the main task constrains all the robot degrees of freedom and when additional secondary tasks are considered. The main advantage of the new avoidance method over classical gradient projection method is that the magnitude of the self motion is automatically computed. The problem of avoiding several joint limits simultaneously has also been validated using the integrated approach presented in Section 7.3 on a six DOFs robot arm.

Limitations and future work

The work performed in this thesis and the contributions obtained open several research issues to be performed in the future concerning visual servoing control schemes, redundancy and joint limits avoidance.

First order control scheme

The future work for this control scheme will be devoted to determining how to select automatically the value of the behavior controller in order to avoid singularity, local minima or other problems and to obtain a good behavior in all cases. Modifying on line the value of the behavior controller during the task execution will be also studied. Let us mention again that this modeling strategy for visual servoing can be used for switching between the classical image-based visual servoing control schemes based on specified criteria such as the current image feature set and robot system configuration. Other future works are expressed on Fig. 8.2.

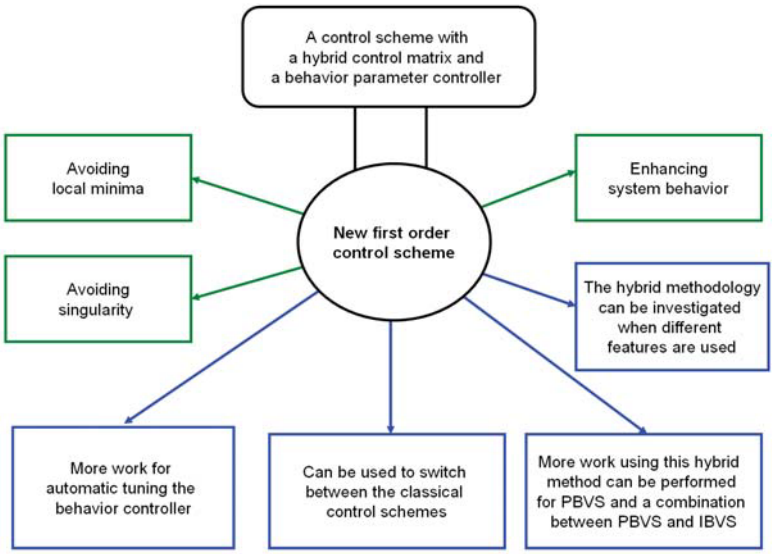


Figure 8.2 – First order control scheme

Second order control schemes

It would be interesting as a future work to see for these control schemes if it can improve the positioning accuracy for other singular configurations, such as the case of the centered circle for instance or when different kind of visual features such as straight lines or ellipse are considered. It would also be interesting to see if any improvement can be obtained by combining Halley's method with the damped-least-squares method. Although this method is able to position accurately the robot in the desired configuration, it is sensitive to the noise in the image. This can be avoided by a preprocessing step for image noise removal to perfectly tracking the accurate position of the features in the image plane (see Fig. 8.3).

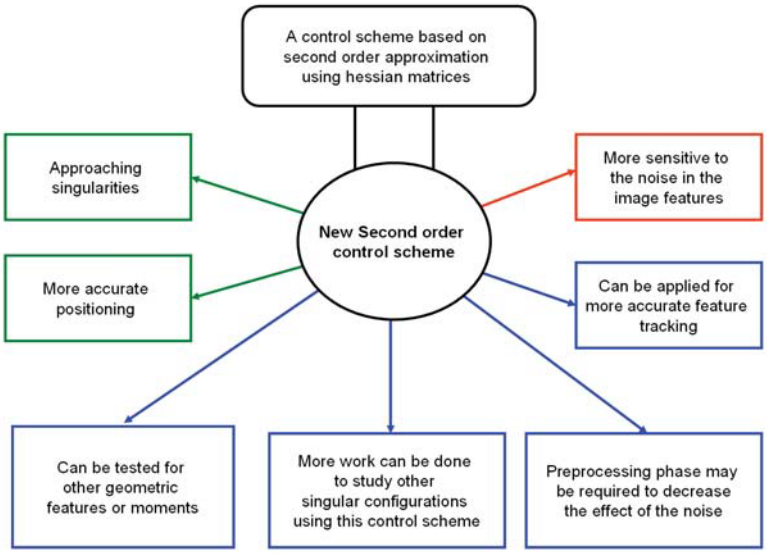


Figure 8.3 – Second order control scheme

Redundancy formalism and task-priority

The fact that the proposed projection operator has a singularity as the error nears zero has been solved within this work by designing a switching strategy that make the system switch to the classical projection operator as the error approaches zero. The theoretical achievements concerning the proposed redundancy frameworks can be improved by applying the discretization method to ensure a higher limit for the secondary task such that the main task is respected. The new redundancy formalism can be used to integrate several actions when multi-sensors are used in robot control. More experiments by concerning the priorities among several tasks can be performed by adding supplementary tasks for grasping or trajectory following. The proposed redundancy framework can be used in task planning strategies to manage levels of priority among stack of tasks for example. This would allow better system performances especially when not enough degrees of freedom are available. Let us finally note that, the new redundancy frameworks is applicable to a wide range of positioning tasks. It can be employed on mobile platforms such as helicopters, underwater vehicles [Tang 09] [Antonelli 98], space based robots [Russkow 92] or terrestrial vehicles to guide the robot to a desired position and orientation with respect to one or more target objects. These issues are illustrated on Fig. 8.4.

Joint limits avoidance

Finally, the new joint limit avoidance strategy can be integrated with other avoidance approaches. The proposed joint limit avoidance and its integrated version with the kernel iterative method can be implemented to other robot configurations including mobile robot and humanoid robot. It can be possible to use the same avoidance strategies for different avoidance problems (see Fig. 8.5).

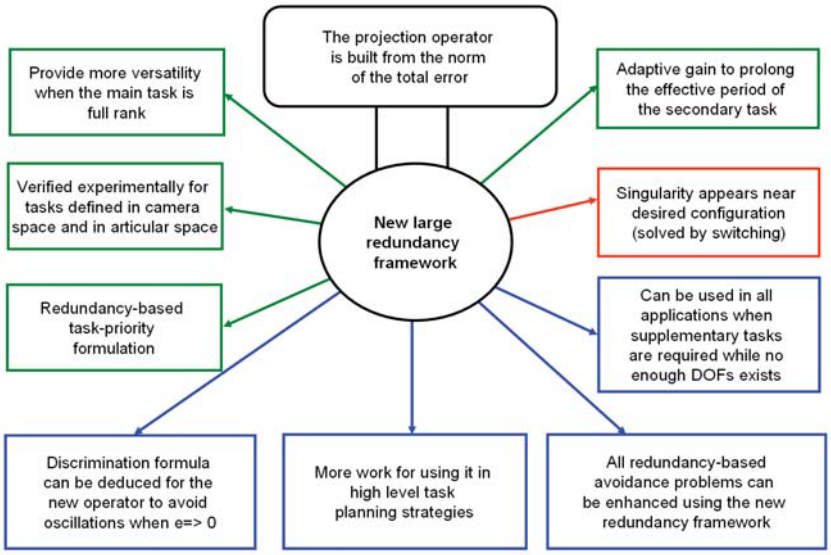


Figure 8.4 – Redundancy frameworks

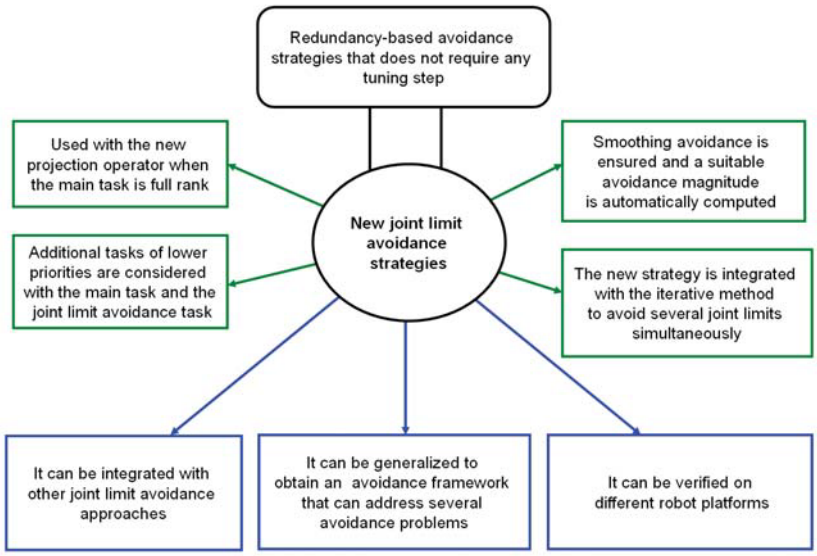


Figure 8.5 – Joint limit avoidance strategies

List of publications

[Marey] M. Marey, F. Chaumette. – A new large projection operator for tasks redundancy - Applications to joint limits avoidance. *IEEE Transactions on Robotics*, (Submitted to IEEE Transactions on Robotics, May 2010).

[Marey 08a] M. Marey, F. Chaumette. – Analyse de lois de commande d'asservissement visuel. – *16e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle, RFIA'08*, Amiens, France, January 2008.

[Marey 08b] M. Marey, F. Chaumette. – Analysis of classical and new visual servoing control laws. – *IEEE International Conference on Robotics and Automation, ICRA'08*, pp. 3244–3249, Pasadena, California, May 2008.

[Marey 08c] M. Marey, F. Chaumette. – A behavior-based visual servoing control law. – *Mediterranean Conference on Intelligent Systems and Automation, CISA'08*, Annaba, Algeria, June 2008.

[Marey 09] M. Marey, F. Chaumette. – A temptative to reach a visual singular configuration using halley's method. – *IEEE International Conference on Robotics and Automation, ICRA'09*, pp. 1122–1127, Kobe, Japan, May 2009.

[Marey 10a] M. Marey, F. Chaumette. – A new large projection operator for the redundancy framework. – *IEEE International Conference on Robotics and Automation, ICRA'10*, pp. 3727–3732, Anchorage, Alaska, May 2010.

[Marey 10b] M. Marey, F. Chaumette. – New strategies for avoiding robot joint limits: Application to visual servoing using a large projection operator. – *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'10*, Taipei, Taiwan, October 2010.

Bibliography

[Ahson 96] S. I. Ahson, N. E. Sharkey, B. Nicolas. – Avoiding joint limits and obstacles for kinematically redundant manipulators: a fuzzy logic based approach. – *IEEE International Conference on Fuzzy Systems*, vol. 3, pp. 1777–1781, September 1996.

[Allen 93] P.K. Allen, A. Timcenko, B. Yoshimi, P. Michelman. – Automated tracking and grasping of a moving object with a robotic hand-eye system. *IEEE Transactions on Robotics and Automation*, 9(2):152–165, avril 1993.

[Ansar 03] A. Ansar, K Daniilidis. – Linear pose estimation from points or lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):578–589, May 2003.

[Antonelli 98] G. Antonelli, S. Chiaverini. – Task-priority redundancy resolution for underwater vehicle-manipulator systems. – *IEEE International Conference on Robotics and Automation, ICRA'98*, vol. 1, pp. 768–773, 1998.

[Antonelli 09] G. Antonelli. – Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems. *IEEE Transactions on Robotics and Automation*, 25(5):985–994, October 2009.

[Assal 06a] S.F.M. Assal, K. Watanabe, K. Izumi. – Intelligent control for avoiding the joint limits of redundant planar manipulators. – *Artif Life Robotics*, vol. 10, pp. 141–148, 2006.

[Assal 06b] S.F.M. Assal, K. Watanabe, K. Izumi. – Neural network-based kinematic inversion of industrial redundant robots using cooperative fuzzy hint for the joint limits avoidance. *IEEE/ASME Transactions on Mechatronics*, 11(5):593–603, October 2006.

[Baerlocher 98] P. Baerlocher, R. Boulic. – Task priority formulations for the kinematic control of highly redundant articulated structures. –

IEEE/RSJ International Conference on Intelligent Robots Systems, IROS'98, vol. 98, pp. 323–329, 1998.

[Baerlocher 04] P. Baerlocher, R. Boulic. – An inverse kinematics architecture enforcing an arbitrary number of strict priority levels. *The visual computer*, 20(6):402–417, 2004.

[Basri 99] R. Basri, E. Rivlin, I. Shishoni. – Visual homing: Surfing on the epipoles. *International journal of Computer Vision*, 33(2):117–137, February 1999.

[Benhimane 07] S. Benhimane, E. Malis. – Homography-based 2d visual tracking and servoing. *International Journal of Robotics Research*, 26(7):661–676, July 2007.

[Branicky 97] M.S. Branicky. – Stability of hybrid systems: State of the art. – *IEEE Conference on Decision and Control, CDC'97*, vol. 1, pp. 120–126, 1997.

[Cervera 99] E. Cervera, P. Martinet. – Visual servoing with indirect image control and a predictable camera trajectory. – *IEEE/RSJ International Conference on Intelligent Robots Systems, IROS'99*, vol. 1, pp. 381–386, 1999.

[Cervera 03] E. Cervera, A. P. del Pobil, F. Berry, P. Martinet. – Improving image-based visual servoing with three-dimensional features. *International Journal of Robotics Research*, 22(10-11):821–840, 2003.

[Chan 95] T.F. Chan, R.V. Dubey. – A weighted least-norm solution based scheme for avoiding joints limits for redundant manipulators. *IEEE Transactions on Robotics and Automation*, 11(2):286–292, April 1995.

[Chang 87] P. Chang. – A closed-form solution for inverse kinematics of robot manipulators with redundancy. *IEEE Journal of Robotics and Automation*, 3(5):393 –403, October 1987.

[Chaumette 93] F. Chaumette, P. Rives, B. Espiau. – Classification and realization of the different vision-based tasks. *Visual Servoing*, editor K. Hashimoto, pp. 199–228. – Singapour, World Scientific Series in Robotics and Automated Systems, 1993.

[Chaumette 96] F. Chaumette, S. Boukir, P. Bouthemy, D. Juvin. – Structure from controlled motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5):492 –504, May 1996.

[Chaumette 98] F. Chaumette. – Potential problems of stability and convergence in image-based and position-based visual servoing. *The Confluence of Vision and Control*, editor D. Kriegman, G . Hager, A.S. Morse, pp. 66–78. – LNCIS Series, No 237, Springer-Verlag, 1998.

[Chaumette 00] F. Chaumette, E. Malis. – 2 1/2 D visual servoing: a possible solution to improve image-based and position-based visual servoings. – *IEEE International Conference on Robotics and Automation, ICRA'00*, vol. 1, pp. 630–635, San Francisco, CA, April 2000.

[Chaumette 01] F. Chaumette, E. Marchand. – A redundancy-based iterative scheme for avoiding joint limits: Application to visual servoing. *IEEE Transactions on Robotics and Automation*, 17(5):719–730, October 2001.

[Chaumette 04] F. Chaumette. – Image moments: a general and useful set of features for visual servoing. *IEEE Transactions on Robotics*, 20(4):713–723, August 2004.

[Chaumette 06] F. Chaumette, S. Hutchinson. – Visual servo control, part I : Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4):82–90, December 2006.

[Chaumette 07] F. Chaumette, S. Hutchinson. – Visual servo control, part II : Advanced approaches. *IEEE Robotics and Automation Magazine*, 14(1):109–118, March 2007.

[Cheng 92] F.-T. Cheng, T.-H. Chen, Y.-Y Sun. – Efficient algorithm for resolving manipulator redundancy the compact QP method. – *IEEE International Conference on Robotics and Automation, ICRA'92*, vol. 1, pp. 508–513, Nice , France, May 1992.

[Cheng 94] F.-T. Cheng, R.-J. Sheu, T.-H. Chen, Kung F.-C. – The improved compact QP method for resolving manipulator redundancy. – *IEEE/RSJ/GI International Conference on Intelligent Robots and Systems, IROS'94*, vol. 2, pp. 1368–1375, Munich, Germany, 1994.

[Chesi 04a] G. Chesi, K. Hashimoto, D. Prattichizzo, A. Vicino. – Keeping features in the field of view in eye-in-hand visual servoing: a switching approach. *IEEE Transactions on Robotics*, 20(5):908 – 914, October 2004.

[Chesi 04b] G. Chesi, A. Vicino. – Visual servoing for large camera displacements. *IEEE Transactions on Robotics*, 20(4):724–735, 2004.

[Chesi 09a] G. Chesi. – Estimation of the camera pose from image point correspondences through the essential matrix and convex optimization. – *IEEE International Conference on Robotics and Automation, ICRA'09*, pp. 35 –40, 12-17 2009.

[Chesi 09b] G. Chesi. – Visual servoing path planning via homogeneous forms and LMI optimizations. *IEEE Transactions on Robotics*, 25(2):281 –291, April 2009.

[Chiacchio 91] P. Chiacchio, S. S. Chiaverini, L. Sciavicco, B. Siciliano. – Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy. *International Journal of Robotics Research*, 10(4):410–425, 1991.

[Chiaverini 97] S. Chiaverini. – Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Transactions on Robotics*, 13(3):398–410, June 1997.

[Choi 00] S.I. Choi, B.K. Kim. – Obstacle avoidance for redundant manipulators using directional-collidability / temporal-collidability measure. *Journal of Intelligent and Robotic Systems*, 28(3):213–229, 2000.

[Cline 64] R.E. Cline. – Representation for the generalized inverse of a partitioned matrix. *Journal of the Society for Industrial and Applied Mathematics*, 12:588–600, 1964.

[Collewet 08] C. Collewet, E. Marchand, F. Chaumette. – Visual servoing set free from image processing. – *IEEE International Conference on Robotics and Automation, ICRA'08*, pp. 81–86, Pasadena, California, May 2008.

[Collewet 09a] C. Collewet, E. Marchand. – Colorimetry-based visual servoing. – *IEEE International Conference on Intelligent Robots and Systems, IROS'09*, pp. 5438–5443, St Louis, USA, October 2009.

[Collewet 09b] C. Collewet, E. Marchand. – Photometry-based visual servoing using light reflexion models. – *IEEE International Conference on Robotics and Automation, ICRA'09*, pp. 701–706, Kobe, Japan, May 2009.

[Comport 06] A.I. Comport, E. Marchand, F. Chaumette. – Statistically robust 2D visual servoing. *IEEE Transactions on Robotics*, 22(2):415–421, April 2006.

[Corke 01] P.I. Corke, S. Hutchinson. – A new partitioned approach to image-based visual servo control. *IEEE Transactions on Robotics and Automation*, 17(4):507–515, August 2001.

[Corke 09] P. I. Corke, F. Spindler, F Chaumette. – Combining cartesian and polar coordinates in IBVS. – *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'09*, pp. 5962–5967, St. Louis, USA, October 2009.

[Craig 86] J.J. Craig. – *Introduction to robotics: mechanics and control*. – Addison-Wesley Reading, MA, 1986.

[Crétual 01] A. Crétual, F. Chaumette. – Visual servoing based on image motion. *International Journal of Robotics Research*, 20(11):857–877, November 2001.

[Deguchi 98] K. Deguchi. – Optimal motion control for image-based visual servoing by decoupling translation and rotation. – *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'98*, vol. 2, pp. 705–711, Amsterdam, October 1998.

[DeLuca 08a] A. De Luca, M. Ferri, G. Oriolo, P. R. Giordano. – Visual servoing with exploitation of redundancy: An experimental study. – *IEEE International Conference on Robotics and Automation, ICRA'08*, pp. 3231 –3237, May 2008.

[DeLuca 08b] A. De Luca, G. Oriolo, P. R. Giordano. – Feature depth observation for image-based visual servoing: Theory and experiments. *International Journal of Robotics Research*, 27(10):1093–1116, 2008.

[Dementhon 95] D. F. Dementhon, L. S. Davis. – Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15(1-2):123–141, 1995.

[Deng 02a] L. Deng, F. Janabi-Sharifi, W. Wilson. – Hybrid strategies for image constraints avoidance in visual servoing. – *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'02*, pp. 348–353, October 2002.

[Deng 02b] L. Deng, W. Wilson, F. Janabi-Sharifi. – Characteristics of robot visual servoing methods and target model estimation. – *IEEE International Symposium on Intelligent Control*, pp. 684 – 689, October 2002.

[Deng 03] L. Deng, W. Wilson, F. Janabi-Sharifi. – Dynamic performance of the position-based visual servoing method in the cartesian and image spaces. – *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'03*, vol. 1, pp. 510 – 515, December 2003.

[Deng 05] L. Deng, F. Janabi-Sharifi, W.J. Wilson. – Hybrid motion control and planning strategies for visual servoing. *IEEE Transactions on Industrial Electronics*, 52(4):1024 – 1040, August 2005.

[Deo 92] A. Deo, I. Walker. – Robot subtask performance with singularity robustness using optimal damped least squares. – *IEEE International Conference on Robotics and Automation, ICRA'92*, pp. 434–441, Nice, France, May 1992.

[Dombre 07] E. Dombre, W. Khalil. – *Robot Manipulator Modeling, performance analysis and control*. – ISTE, Cambridge, Massachusetts, 2007.

[Dornaika 99] F. Dornaika, G. Garcia. – Pose estimation using point and line correspondences. *Real-Time Imaging*, 5(3):215–230, 1999.

[Dubey 88] RV Dubey, JA Euler, SM Babcock. – An efficient gradient projection optimization scheme for a seven-degree-of-freedom redundant robot with spherical wrist. – *IEEE International Conference on Robotics and Automation, ICRA'88*, pp. 24–29, 1988.

[Egeland 90] O. Egeland, M. Ebdrup, S. Chiaverini. – Sensory control in singular configurations-application to visual servoing. – *IEEE International Workshop on Intelligent Motion Control*, pp. 401–405, Istanbul, August 1990.

[Ellekilde 07] L.P. Ellekilde, P. Favrholdt, M. Paulin, H.G. Petersen. – Robust control for high-speed visual servoing applications. *International Journal of Advanced Robotic Systems*, 4(3):279–292, 2007.

[Espiau 92] B. Espiau, F. Chaumette, P. Rives. – A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326, June 1992.

[Espiau 93] B. Espiau. – Effect of camera calibration errors on visual servoing in robotics. – *International Symposium on experimental Robotics, ISER'93*, pp. 182–192, Kyoto, Japan, 1993.

[Euler 89] JA Euler, RV Dubey. – A comparison of two real-time control schemes for redundant manipulators with bounded joint velocities. – *IEEE International Conference on Robotics and Automation, ICRA'89*, vol. 1, pp. 701–706, May 1989.

[Faugeras 93] O. Faugeras. – *Three-dimensional computer vision: a geometric viewpoint*. – MIT Press, Cambridge, Massachusetts, 1993.

[Feddema 89a] J. Feddema, O. Mitchell. – Vision-guided servoing with feature-based trajectory generation. *IEEE Trans. on Robotics and Automation*, 5(5):691–700, October 1989.

[Feddema 89b] J.T Feddema, C.S.G. Lee, O.R. Mitchell. – Automatic selection of image features for visual servoing of a robot manipulator. – *IEEE International Conference on Robotics and Automation, ICRA'89*, vol. 2, pp. 832–837, Scottsdale, Arizona, May 1989.

[Feddema 91] J.T Feddema, C.S.G. Lee, O.R. Mitchell. – Weighted selection of image features for resolved rate visual feedback control. *IEEE Transactions on Robotics and Automation*, 7(1):31–47, February 1991.

[Ficocelli 01] M. Ficocelli, F. Janabi-Sharifi. – Adaptive filtering for pose estimation in visual servoing. – *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'03*, pp. 19–24, Maui, Hawaii, USA, November 2001.

[Flandin 00] G. Flandin, F. Chaumette, E. Marchand. – Eye-in-hand / eye-to-hand cooperation for visual servoing. – *IEEE International Conference on Robotics and Automation, ICRA'00*, vol. 3, pp. 2741–2746, San Francisco, California, April 2000.

[Forsyth 03] D. Forsyth, J. Ponce. – *Computer Vision: A Modern Approach.* – Upper Saddle River, NJ: Prentice Hall,, 2003.

[Fruchard 06] M. Fruchard, P. Morin, C. Samson. – A fremawork for the control of nonholonomic mobile manipulators. *International Journal of Robotics Research*, 25(8):398–410, August 2006.

[Gans 03a] N. Gans, S. Hutchinson. – An asymptotically stable switched system visual controller for eye in hand robots. – *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'03*, vol. 1, pp. 735 – 742, October 2003.

[Gans 03b] N. Gans, S. Hutchinson. – An experimental study of hybrid switched system approaches to visual servoing. – *IEEE. International Conference Robotics and Automation, ICRA'03*, vol. 1, pp. 3061 – 3068, September 2003.

[Gans 03c] N.R. Gans, S.A. Hutchinson, P.I. Corke. – Performance tests for visual servo control systems, with application to partitioned approaches to visual servo control. *International Journal of Robotics Research*, 22(10-11):955–981, 2003.

[Gans 07] N. Gans, S. Hutchinson. – Stable visual servoing through hybrid switched-systems contro. *IEEE Transactions on Robotics*, 23(3):530–540, June 2007.

[GarciaAracil 05] N. Garcia-Aracil, E. Malis, R. Aracil-Santonja, C. Perez-Vidal. – Continuous visual servoing despite the changes of visibility in image features. *IEEE Transactions on Robotics*, 21(6):415–421, April 2005.

[Guoqiang 10] H. Guoqiang, N. Gans, N. Fitz-Coy, W. Dixon. – Adaptive homography-based visual servo tracking control via a quaternion formulation. *IEEE Transactions on Control Systems Technology*, 18(1):128 –135, January 2010.

[HadjAbdelkader 07] H. Hadj-Abdelkader, Y. Mezouar, P. Martinet. – Decoupled Visual Servoing from a set of points imaged by an omnidirectional camera. – *IEEE International Conference on Robotics and Automation, ICRA'07*, pp. 1697–1697, Roma, April 2007.

[Hafez 06] A.H.A. Hafez, C.V. Jawahar. – Probabilistic integration of 2D and 3D cues for visual servoing. – *IEEE International Conference on Robotics and Automation, ICRA'06*, pp. 1691–1696, April 2006.

[Hafez 07a] A.H.A. Hafez, E. Cervera, C.V. Jawahar. – Optimizing image and camera trajectories in robot vision control using on-line boosting. – *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'07*, pp. 352 –357, San Diego, USA, October 2007.

[Hafez 07b] A.H.A. Hafez, C.V. Jawahar. – Visual servoing by optimization of a 2D/3D hybrid objective function. – *IEEE International Conference on Robotics and Automation, ICRA'07*, pp. 1691 –1696, Roma, Italy, 2007.

[Hafez 08a] A.H.A. Hafez, S. Achar, C.V. Jawahar. – Visual servoing based on gaussian mixture models. – *IEEE International Conference on Robotics and Automation, ICRA'08*, Pasadena, California, May 2008.

[Hafez 08b] A.H.A. Hafez, E. Cervera, C.V. Jawahar. – Hybrid visual servoing by boosting IBVS and PBVS. – *International Conference on Information and Communication Technologies: From Theory to Applications,ICTTA'08*, pp. 1–6, April 2008.

[Hager 97] G. Hager. – A modular system for robust positioning using feedback from stereo vision. *IEEE Transactions on Robotics and Automation*, 13(4):582–595, August 1997.

[Hanafusa 81] H. Hanafusa, T. Yoshikawa, Y. Nakamura. – Analysis and control of articulated robot with redundancy. – *IFAC, 8th Triennal World Congress*, vol. 4, pp. 1927–1932, 1981.

[Haralick 89] R. Haralick, H. Joo, C. Lee, X. Zhuang, V Vaidya, M. Kim. – Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6):1426–1445, November 1989.

[Hartley 97] R.I. Hartley. – In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580 –593, June 1997.

[Hashimoto 93a] K. Hashimoto. – Real time control of robot manipulators based on visual sensory feedback. *Visual Servoing*, editor K. Hashimoto. – Singapour, World Scientific Series in Robotics and Automated Systems, 1993.

[Hashimoto 93b] K. Hashimoto, H. Kimura. – LQ optimal and nonlinear approaches to visual servoing. *Visual Serving: Real Time Control of Robot Manipulators Based on Visual Sensory Feedback*, 7:165–198, 1993.

[Hashimoto 96] K. Hashimoto, T. Ebine, H. Kimura. – Visual servoing with hand-eye manipulator: Optimal control approach. *IEEE Transactions on Robotics and Automation*, 12(5):766–774, 1996.

[Hashimoto 00a] K. Hashimoto, T. Noritsugu. – Enlargement of stable region in visual servo. – *IEEE Conference on Decision and Control*, vol. 4, pp. 3927–3932, 2000.

[Hashimoto 00b] K. Hashimoto, Noritsugu T. – Potential switching control in visual servoing. – *IEEE International Conference on Robotics and Automation, ICRA'98*, vol. 3, pp. 2765–2770, San Francisco, USA, April 2000.

[Horn 86] B.K. Horn. – *Robot vision*. – McGraw-Hill Higher Education, 1986.

[Hutchinson 96] S. Hutchinson, G. Hager, P. Corke. – A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, October 1996.

[Iwatsuki 05] M. Iwatsuki, N. Okiyama. – A new formulation of visual servoing based on cylindrical coordinate system. *IEEE Transactions on Robotics*, 21(2):266–273, 2005.

[JanabiSharifi 97] F. Janabi-Sharifi, W Wilson. – Automatic selection of image features for visual servoing. *IEEE Transactions on Robotics and Automation*, 13(6):890–903, December 1997.

[JanabiSharifi 10a] F. Janabi-Sharifi, L. Deng, W Wilson. – Comparison of basic visual servoing methods. *IEEE/ASME Transactions on Mechatronics*, pp(91):1–17, September 2010.

[JanabiSharifi 10b] F. Janabi-Sharifi, M. Marey. – A kalman-filter-based method for pose estimation in visual servoing. *IEEE Transactions on Robotics*, 26(5):939 – 947, October 2010.

[KaBerounian 88] K. KaBerounian, Z. Wang. – Global versus local optimisation in redundancy resolution of robotic manipulators. *International Journal of Robotics Research*, 7(5):3–12, 1988.

[Kallem 07] V. Kallem, M. Dewan, J. Swensen, G. Hager, N. Cowan. – Kernel-based visual servoing. – *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'07*, pp. 1975 – 1980, San Diego, USA, October 2007.

[Kelly 00] R. Kelly, R. Carelli, O. Nasisi, B. Kuchen, F. Reyes. – Stable visual servoing of camera-in-hand robotic systems. *IEEE/ASME Transactions on mechatronics*, 5(1):39–48, 2000.

[Kelly 04] R. Kelly, J. Moreno, R. Campa. – Visual servoing of planar robots via velocity fields. – *IEEE Conference on Decision and Control, CDC'04*, vol. 4, pp. 4028–4033, December 2004.

[Kelly 06] R. Kelly, E. Bugarin, V. Sanchez. – Image-based visual control of nonholonomic mobile robots via velocity fields: Case of partially calibrated inclined camera. – *IEEE Conference on Decision and Control, CDC'06*, pp. 3071–3076, December 2006.

[Khatib 85] O. Khatib. – Real-time obstacle avoidance for manipulators and mobile robots. – *IEEE International Conference on Robotics and Automation, ICRA'85*, vol. 2, pp. 500–505, March 1985.

[Khatib 96] O. Khatib. – The impact of redundancy on the dynamic performance of robots. – *Laboratory Robotics and Automation*, vol. 8, pp. 37–48, 1996.

[Kragic 02] D. Kragic, H.I. Christensen. – *Survey on visual servoing for manipulation*. – Research report, Centre for Autonomous Systems, Stockholm, Sweden - Research report, 2002.

[Kyrki 04a] V. Kyrki, D. Kragic, H.I. Christensen. – Measurement errors in visual servoing. – *IEEE International Conference on Robotics and Automation, ICRA'04*, vol. 2, pp. 1861 – 1867, April 2004.

[Kyrki 04b] V. Kyrki, D. Kragic, H.I. Christensen. – New shortest-path approaches to visual servoing. – *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'04*, vol. 1, pp. 349–354, 2004.

[Lapresté 04] J.-T. Lapresté, Y. Mezouar. – A hessian approach to visual servoing. – *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'04*, vol. 1, pp. 998–1003, Sendai, Japan, September 2004.

[Lenarcic 98] J. Lenarcic. – Effective secondary task execution of redundant manipulators. *Robotica*, 16(4):457–462, 1998.

[Liegeois 77] A. Liegeois. – Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Transactions on Systems, Man and Cybernetics*, 7(12):868–871, December 1977.

[Lippiello 04] V. Lippiello, B. Siciliano, L. Villani. – Visual motion estimation of 3D objects: an adaptive extended kalman filter approach. – *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'04*, vol. 1, pp. 957 – 962, Sendai, Japan, September 2004.

[Liu 99] M. L. Liu, K. H. Wong. – Pose estimation using four corresponding points. *Pattern Recognition Letters*, 20(1):69–74, 1999.

[LonguetHiggins 81] H.C. Longuet-Higgins. – A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, September 1981.

[Luya 01] Li. Luya, W.A. Gruver, Z. Qixian, Y. Zongxu. – Kinematic control of redundant robots and the motion optimizability measure. *IEEE Transactions on Systems, Man and Cybernetics*, 31(1):155–160, February 2001.

[Malis 99] E. Malis, F. Chaumette, S. Boudet. – 2 1/2 d visual servoing. *IEEE Transactions on Robotics and Automation*, 15(2):238–250, April 1999.

[Malis 00] E. Malis, F. Chaumette, S. Boudet. – 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, June 2000.

[Malis 02] E. Malis, F. Chaumette. – Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods. *IEEE Transactions on Robotics*, 18(2):176186, 2002.

[Malis 04] E. Malis. – Improving vision-based control using efficient second-order minimization techniques. – *IEEE International Conference on Robotics and Automation, ICRA'04*, vol. 2, pp. 1843–1848, New Orleans, April 2004.

[Malis 10] E. Malis, Y. Mezouar, P. Rives. – Robustness of image-based visual servoing with a calibrated camera in the presence of uncertainties in the three-dimensional structure. *IEEE Transactions on Robotics*, 26(1):112 –120, February 2010.

[Mansard 04] N. Mansard, F. Chaumette. – Tasks sequencing for visual servoing. – *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'04*, pp. 992–997, October 2004.

[Mansard 07] N. Mansard, F. Chaumette. – Task sequencing for high-level sensor-based control. *IEEE Transactions on Robotics*, 23(1):60–72, 2007.

[Mansard 09a] N. Mansard, F. Chaumette. – Directionnal redundancy for robot control. *IEEE Transactions on Automatic Control*, 54(6):1179–1192, June 2009.

[Mansard 09b] N. Mansard, A. Remazeilles, F. Chaumette. – Continuity of varying-feature-set control laws. *IEEE Transactions on Automatic Control*, 54(11):2493–2505, November 2009.

[Marcé 87] L. Marcé, P. Bouthemy. – Determination of a depth map from an image sequence. – *3rd International Conference on Advanced Robotics*, pp. 221–232, October 1987.

[Marchand 96] E. Marchand, F. Chaumette, A. Rizzo. – Using the task function approach to avoid robot joint limits and kinematic singularities in visual servoing. – *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'96*, vol. 3, pp. 1083–1090, Osaka, Japan, November 1996.

[Marchand 98] E. Marchand, G.-D. Hager. – Dynamic sensor planning in visual servoing. – *IEEE International Conference on Robotics and Automation, ICRA'98*, vol. 3, pp. 1988–1993, Leuven, Belgium, May 1998.

[Marchand 02] E. Marchand, F. Chaumette. – Virtual visual servoing: A framework for real-time augmented reality. *Computer Graphics Forum*, 21(3):289–298, September 2002.

[Marchand 05a] E. Marchand, F. Chaumette. – Feature tracking for visual servoing purposes. *Robotics and Autonomous Systems*, 52(1):53–70, June 2005.

[Marchand 05b] E. Marchand, F. Spindler, F. Chaumette. – Visp for visual servoing: A generic software platform with a wide class of robot control skills. *IEEE Robotics and Automation Magazine*, 12(4), December 2005. – Special Issue on "Software Packages for Vision-Based Control of Motion", P. Oh, D. Burschka (Eds.).

[Martinet 96] P. Martinet, J. Gallice, D. Khadraoui. – Vision based control law using 3D visual features. – *Econometrica Committees, WAC 96*, pp. 497–502, Montpellier, May 1996.

[Martinet 97] P. Martinet, N. Daucher, J. Gallice, M. Dhome. – Robot control using monocular pose estimation. – *Workshop on New Trends In Image-Based Robot Servoing*, p. 112, Grenoble, France, September 1997.

[Martinet 99] P. Martinet, J. Gallice. – Position-based visual servoing using a nonlinear approach. – *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'99*, vol. 1, p. 531536, Kyongju, Korea, October 1999.

[Matthies 89] L. Matthies, R. Szelinski, T. Kanade. – Kalman filter-based algorithms for estimating depth from image sequences. *International journal of Computer Vision*, 3:209–236, 1989.

[Mcinroy 08] J. E. Mcinroy, Z. Qi. – A novel pose estimation algorithm based on points to regions correspondence. *Journal of Mathematical Imaging and Vision*, 30(2):195–207, 2008.

[Mebarki 10] R. Mebarki, A. Krupa, F. Chaumette. – 2-d ultrasound probe complete guidance by visual servoing using image moments. *IEEE Transactions on Robotics*, 26(2):296 –306, April 2010.

[Mezouar 02] Y. Mezouar, F. Chaumette. – Path planning for robust image-based control. *IEEE Transactions on Robotics and Automation*, 18(4):534 – 549, August 2002.

[Michel 93] H. Michel, P. Rives. – *Singularities in the determination of the situation of a robot effector from the perspective view of three points*. – Research report n1850, INRIA - Research report, February 1993.

[Myers 76] K. A. Myers, B. D. Tapley. – Adaptive sequential estimation with unknown noise statistics. *IEEE Transactions on Automatic Control*, 21:520–523, August 1976.

[Na 08] M. Na, B. Yang, P. Jia. – Improved damped least squares solution with joint limits, joint weights and comfortable criteria for controlling human-like figures. – *IEEE Conference on Robotics, Automation and Mechatronics*, pp. 1090 –1095, September 2008.

[Nakamura 86] Y. Nakamura, H. Hanafusa. – Inverse kinematics solutions with singularity robustness for robot manipulator control. *Transactions ASME journal of Dynamic System, Measures and Control*, 108:163–171, September 1986.

[Nakamura 87] Y. Nakamura, H. Hanafusa, T. Yoshikawa. – Task-priority based redundancy control of robot manipulators. *International Journal of Robotics Research*, 6(2):3–15, 1987.

[Nayar 96] S.K. Nayar, S.A. Nene, H. Murase. – Subspace methods for robot vision. *IEEE Transactions on Robotics*, 12(5):750 – 758, October 1996.

[Nelson 95] B. Nelson, P.K. Khosla. – Strategies for increasing the tracking region of an eye-in-hand system by singularity and joint limits avoidance. *International Journal of Robotics Research*, 14(3):255–269, June 1995.

[OkamotoJr 02] J. Okamoto Jr, V. Grassi Jr. – Visual servo control of a mobile robot using omnidirectional vision. *Proceedings of Mechatronics*, pp. 413–422, 2002.

[Ortega 00] J.M. Ortega, W.C. Rheinboldt. – *Iterative Solution of Nonlinear Equations in Several Variables*. – Society for Industrial Mathematics, SIAM, 2000.

[Pages 06] J. Pages, C. Collewet, F. Chaumette, J. Salvi. – Optimizing plane-to-plane positioning tasks by image-based visual servoing and structured light. *IEEE Transactions on Robotics*, 22(5):1000 – 1010, October 2006.

[Papanikolopoulos 93] N. P. Papanikolopoulos, P. K Khosla. – Selection of features and evaluation of visual measurements for 3D robotic visual tracking. *International Symp. on Intelligent Control.*, pp. 320–325, August 1993.

[Paulin 04] M. Paulin. – The VirtualWorkCell–The Missing Link Between Robot Motion Planning and Visual Servoing. – *IEEE International Conference on Mechatronics and Robotics*, vol. 2, pp. 326–331, 2004.

[Pearl 20] R. Pearl, L. Reed. – On the rate of growth of the population of the united state since 1790 and its mathematical representation. – *Proc. of the National Academy of Sciences*, vol. 6, pp. 275–288, June 1920.

[Perdereau 02] V. Perdereau, C. Passi, M. Drouin. – Real-time control of redundant robotic manipulators for mobile obstacle avoidance. *Robotics and Autonomous Systems*, 41(1):41–59, 2002.

[Piepmeier 99] J. Piepmeier, G. McMurray, H. Lipkin. – A dynamic quasi-newton method for uncalibrated visual servoing. – vol. 2, pp. 1595–1600, 1999.

[Rosen 60] JB Rosen. – The gradient projection method for nonlinear programming. Part I. Linear constraints. *Journal of the Society for Industrial and Applied Mathematics*, 8(1):181–217, 1960.

[Rosen 61] JB Rosen. – The gradient projection method for nonlinear programming. Part II. Linear constraints. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):514–532, 1961.

[Russkow 92] J. Russkow, O. Khatib. – A New Control Structure for Free-Flying Space Robots. *Symposium on Artificial Intelligence, Robotics and Automation, in Space, i-SAIRAS*, pp. 395–403, September 1992.

[Safaee-Rad 92] R. Safaee-Rad, I. Tchoukanov, B. Benhabib, K.C. Smith. – 3D-pose estimation from a quadratic-curved feature in two perspective views. – *IAPR International Conference on Pattern Recognition, ICPR'92*, vol. 1, pp. 341–344, Netherlands, August 1992.

[Samson 91] C. Samson, M. Le Borgne, B. Espiau. – *Robot Control: The Task Function Approach*. – Clarendon Press, Oxford, United Kingdom, 1991.

[Sanderson 80] A. C. Sanderson, L. E. Weiss. – Image based visual servo control using relational graph error signals. – *IEEE International Conference on Cybernetics and Society*, pp. 1074–1077, 1980.

[Shademan 05] A. Shademan, F. Janabi-Sharifi. – Sensitivity analysis of EKF and iterated EKF pose estimation for position-based visual servoing. – *IEEE Conference on Control Applications*, pp. 755 – 760, August 2005.

[Shahamiri 05] M. Shahamiri, M. Jagersand. – Uncalibrated visual servoing using a biased newton method for on-line singularity detection and avoidance. – *IEEE/RSJ International Conference on Intelligent Robots*

and Systems, IROS'05, pp. 2682–2687, Edmonton, Canada, August 2005.

[Shanno 70] DF Shanno. – Conditioning of quasi-Newton methods for function minimization. *Mathematics of Computation*, 24(111):647–656, 1970.

[Sharma 97] R. Sharma, S. A. Hutchinson. – Motion perceptibility and its application to active vision-based servo control. *IEEE Transactions on Robotics and Automation*, 13(4):607–617, August 1997.

[Shirai 73] Y. Shirai, H. Inoue. – Guiding a robot by visual feedback in assembling tasks. *Pattern Recognition*, 5(2):99–106, 1973.

[Siciliano 91] B. Siciliano, J-J. Slotine. – A general framework for managing multiple tasks in highly redundant robotic systems. – *IEEE International Conference on Robotics and Automation, ICRA'91*, pp. 1211–1216, 1991.

[Spong 06] M.W. Spong, S. Hutchinson, M. Vidyasagar. – *Robot modeling and control*. – Wiley New Jersey, 2006.

[Tahri 05] O. Tahri, F. Chaumette. – Point-based and region-based image moments for visual servoing of planar objects. *IEEE Trans. on Robotics*, 21(6):1116–1127, December 2005.

[Tahri 10] O. Tahri, Y. Mezouar. – On visual servoing based on efficient second order minimization. *Robotics and Autonomous Systems*, 58(5):712–719, 2010.

[Tang 09] L. Tang, Y. Li. – Study on kinematical control based on fuzzy inference for underwater work system. – *Chinese Control and Decision Conference, 2009. CCDC'09*, pp. 684 –688, June 2009.

[TatsambonFomena 09] R. Tatsambon Fomena, F. Chaumette. – Improvements on visual servoing from spherical targets using a spherical projection model. *IEEE Transactions on Robotics*, 25(4):874–886, August 2009.

[Taylor 00] C. J. Taylor, J. P. Ostrowski, S. H. Jung. – Robust vision-based pose control. – *IEEE International Conference on Robotics and Automation, ICRA'00*, vol. 1, p. 27342740, 2000.

[Taylor 04] G. Taylor, Kleeman L. – Hybrid position-based visual servoing with online calibration for humanoid robot. – *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'04*, vol. 1, pp. 686 – 691, Sendai, Japan, September 2004.

[Thuilot 02] B. Thuilot, P. Martinet, L. Cordesses, J. Gallice. – Position based visual servoing: keeping the object in the field of vision. – *IEEE International Conference on Robotics and Automation, ICRA'02*, vol. 2, pp. 1624 – 1629, 2002.

[Toibero 09] J.M. Toibero, C.M. Soria, F. Roberti, R. Carelli, P. Fiorini. – Switching visual servoing approach for stable corridor navigation. – *International Conference on Advanced Robotics, ICAR'09*, pp. 1 –6, June 2009.

[Tsai 89] R.Y. Tsai, R.K. Lenz. – A new technique for fully autonomous and efficient 3 D robotics hand-eye calibration. *IEEE Transactions on Robotics and Automation*, 5(3):345–358, 1989.

[Wampler 86] C.W. Wampler. – Manipulator inverse kinematic solutions based on vector formulations and damped least squared method. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(1):93–101, January 1986.

[Wang 92] J. Wang, W.J. Wilson. – 3D relative position and orientation estimation using kalman filter for robot control. – *IEEE International Conference on Robotics and Automation, ICRA'92*, vol. 3, pp. 2638 – 2645, Nice, May 1992.

[Wang 10] Y. Wang, H. Lang, C. W. de Silva. – A hybrid visual servo controller for robust grasping by wheeled mobile robots. *IEEE/ASME Transactions on Mechatronics*, PP(99):1 –13, 2010.

[Weiss 87] L.E. Weiss, A.C. Sanderson, C.P. Neuman. – Dynamic sensor-based control of robots with visual feedback. *IEEE journal of Robotics and Automation*, 3(5):404–417, October 1987.

[Wicks 94] M. Wicks, P. Peleties, R. DeCarlo. – Construction of piecewise lyapunov functions for stabilizing switched systems. – *Proc. Conference Decision and Control*, pp. 3492–3497, 1994.

[Wilson 96] W. Wilson, C. Hulls, G. Bell. – Relative end-effector control using cartesian position-based visual servoing. *IEEE Transactions on Robotics and Automation*, 12(5):684–696, October 1996.

[Xiang 10] J. Xiang, C. Zhong, W. Wei. – General-weighted least-norm control for redundant manipulators. *IEEE Transactions on Robotics*, 26(4):660 –669, August 2010.

[Xie 09] W. Xie, Z. Li, X. Tu, Perron C. – Switching control of image-based visual servoing with laser pointer in robotic manufacturing

systems. *IEEE Transactions on Industrial Electronics*, 56(2):520–529, February 2009.

[Yoshikawa 96] T. Yoshikawa. – Basic optimization methods of redundant manipulators. – *Laboratory Robotics and Automation*, vol. 8, pp. 49–60, 1996.

[Zanne 00] P. Zanne, G. Morel, F. Piestan. – Robust vision based 3D trajectory tracking using sliding mode control. – *IEEE International Conference on Robotics and Automation, ICRA'00*, vol. 3, pp. 2088 –2093, April 2000.

[Zghal 90] H. Zghal, R. V. Dubey, J. A. Euler. – Efficient gradient projection optimization for manipulators with multiple degrees of redundancy. – *IEEE International Conference on Robotics and Automation, ICRA'90*, vol. 2, pp. 1006–1011, 1990.

Résumé

L'asservissement visuel est devenu une approche classique dans le cadre de la commande de robots exploitant les informations fournies par un capteur de vision dans une boucle de commande. La recherche décrite dans cette thèse vise à résoudre des problèmes d'asservissement et à améliorer la capacité de gérer plus efficacement les tâches supplémentaires. Cette thèse présente tout d'abord l'état de l'art en asservissement visuel, redondance et évitement des butées articulaires. Elle propose ensuite les contributions suivantes:

Un schéma de commande est obtenu en introduisant un paramètre de comportement dans un contrôle hybride. Il permet un meilleur comportement du système lorsque des valeurs appropriées du paramètre sont sélectionnées. Une étude analytique des lois de commandes les plus courantes et de la nouvelle loi proposée est effectuée dans le cas de mouvements de translation et de rotation selon l'axe optique. De nouveaux schémas de commande sont également proposés pour améliorer le comportement du système lorsque la configuration désirée est singulière.

Les contributions théoriques concernant le formalisme de la redondance reposent sur l'élaboration d'un opérateur de projection obtenu en ne considérant que la norme de la tâche principale. Cela conduit à un problème moins contraint et permet d'élargir le domaine d'application. De nouvelles stratégies d'évitement des butées articulaires du robot fondées sur la redondance sont développées. Le problème d'ajouter des tâches secondaires à la tâche principale, tout en assurant l'évitement des butées articulaires, est également résolu.

Tous ces travaux ont été validés par des expérimentations dans le cadre d'applications d'asservissement visuel.

Abstract

Visual servoing has become a popular paradigm in robot control by exploiting the information provided by a vision sensor in a feedback control loop. The research described in this dissertation aims mainly to solve problems in visual servoing control and to improve the ability to consider more effectively additional tasks. This thesis presents the state of art in visual servoing, redundancy, and joint limits avoidance and the following contributions:

A new first order control scheme obtained by introducing a behavior parameter in a hybrid control matrix is first proposed. It allows a better behavior of the system when suitable values for the behavior parameter are satisfied. An analytical analysis of the most common control scheme and the new one is performed when the camera displacement is a combination of a translation along and a rotation around the camera optical axis. New second order control schemes are also proposed to enhance the behavior of the system when the problem of reaching a desired singular configuration is considered.

Theoretical contributions regarding redundancy-based task-priority are achieved by developing a new projection operator obtained by considering only the norm of the main task. This leads to a less constrained problem and enlarges the applicability domain of redundancy-based task-priority. New strategies for redundancy-based robot joint limits avoidance are developed. The problem of adding additional secondary tasks to the main task while ensuring the joint limits avoidance is finally solved.

All these works have been investigated experimentally using an eye-in-hand system and visual servoing applications.