Automatic guidance of robotized 2D ultrasound probes with visual servoing based on image moments.

Rafik Mebarki

N° d'ordre : 4080 ANNÉE 2010

UNIVERSITÉ DE RENNES 1

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 : Traitement du Signal

Ecole doctorale Matisse

présentée par

Rafik Mebarki

préparée à l'IRISA. Équipe d'accueil : LAGADIC
Composante universitaire : IFSIC

Automatic guidance of robotized 2D ultrasound probes with visual servoing based on image moments

Thèse soutenue à Rennes
le 25 Mars 2010

devant le jury composé de :

Christian Barillot
Directeur de Recherhe, CNRS / président

Guillaume Morel
Professeur, ISIR, Paris / rapporteur

Philippe Poignet
Professeur, LIRMM, Montpellier / rapporteur

Pierre Dupont
Professor, Harvard Medical School, Boston University, USA / examinateur

Alexandre Krupa
Chargé de Recherche, INRIA / co-directeur de thèse

François Chaumette
Directeur de Recherche, INRIA / directeur de thèse

Automatic guidance of robotized 2D ultrasound probes with visual servoing based on image moments

Doctoral Thesis

by

Rafik Mebarki

Thesis submitted in partial satisfaction of
the requirements for the degree of

Doctor of Philosophy (Ph.D.)

in

Signal Processing

from

Université de Rennes 1

This work has been prepared in

IRISA/INRIA Rennes

Committee in charge:

Guillaume Morel
Philippe Poignet
Pierre Dupont
Christian Barillot
Alexandre Krupa
François Chaumette

March 2010

To my parents, my wife, my brothers, and my family.

Acknowledgements

I would like to thank Guillaume Morel, Philippe Poignet, Pierre Dupont, and Christian Barillot for having accepted to review this Ph.D. work and for their time.

I would like to thank Alexandre Krupa and François Chaumette for their advices and the discussions we have had about this Ph.D. work. Moreover, I would like to thank the latter for accepting that I write this dissertation in english.

I would like to thank my team colleagues Romeo Tatsambon-Fomena, Mohammed Marey, Ryuta Ozawa, Céline Teulière, and Nicolas Mansard. Especially, I would like to thank the latter for the wealthy discussions we have had and for different things.

My warmest thanks to my colleague and friend Hamza Drid.

My thanks to Boris D.

To my brave friends of Toulouse, especially Yassine C.

My warmest thanks and gratitude to Mr. Boudour, my former teacher of Mathematics.

My warmest thanks to Riad K. for his valuable help.

To my best friends Samir and Kamal.

My thanks to Riad B. for all his advices and encouragements.

My thanks to my colleagues of the National Polytechnic School of Algeria.

Contents

3

1 Introduction **5**

2 Prior Art **13**
2.1 Medical robotics 14
2.1.1 Human-machine interfaces 17
2.1.2 Operator-robot interaction paradigms 18
2.2 Optical imaging-based guidance: microsurgery robotics 21
2.3 X-ray-based guidance 22
2.4 MRI-guided robotics 24
2.5 Ultrasound-based guidance 26
2.5.1 Ultrasound-based simulations 26
2.5.2 3D ultrasound-guided robotics 27
2.5.3 2D ultrasound-guided position-based visual servoing 30
2.5.4 2D ultrasound-guided image-based visual servoing 38
2.6 Conclusion 47

3 Modeling **49**
3.1 Image moments: a brief state-of-the-art 50
3.2 Discussion with regards to image moments 53
3.3 Image moments-based visual servoing with optical systems: state of the art 56
3.4 Modeling objectives 58
3.5 Image point velocity modeling 62
3.5.1 First constraint 67
3.5.2 Second constraint 69
3.5.3 Virtual point velocity 73
3.6 Image moments time variation modeling 75
3.7 Interpretation for simple shapes 77
3.7.1 Spherical objects 78

3.7.2 Cylindrical objects . . . 84
3.7.3 Interaction with a 3D straight line . . . 87
3.8 Conclusion . . . 88

4 Normal vector on-line estimation **91**
4.1 On-line estimation methods based on lines . . . 91
4.1.1 Straight line-based estimation method . . . 93
4.1.2 Curved line-based estimation method . . . 99
4.2 Quadric surface-based estimation method . . . 102
4.3 Sliding least squares estimation algorithm . . . 105
4.4 Simulation results . . . 108
4.4.1 Interaction with straight lines . . . 108
4.4.2 Interaction with curved lines . . . 110
4.4.3 Interaction with quadric surfaces . . . 111
4.4.4 Ellipsoid objects: perfect and noisy cases . . . 114
4.5 Discussion . . . 120
4.6 Conclusion . . . 123

5 Visual Servoing **125**
5.1 Visual features selection . . . 127
5.2 Simulation results with an ellipsoidal object . . . 131
5.2.1 Model-based visual servoing . . . 132
5.2.2 Model-free visual servoing using the curved line-based normal vector estimation . . . 139
5.3 Simulation results with realistic ultrasound images . . . 145
5.4 Simulation results with a binary object . . . 153
5.5 Experimental results . . . 156
5.5.1 Experimental results with a spherical object . . . 157
5.5.2 Exprimental results with an ultrasound phantom . . . 159
5.5.3 *Ex-vivo* experimental results with a lamb kidney . . . 161
5.5.4 Experimental results with a motionless soft tissue . . . 161
5.5.5 Tracking two targets . . . 164
5.6 Conclusion . . . 166

6 Conclusions **169**

177

A Some fundamentals in coordinate transformations **177**
A.1 Scalar product . . . 177
A.2 Skew-symmetric matrix . . . 177
A.3 Vector cross-product . . . 178
A.4 Points Projection . . . 179
A.5 Rotation matrix properties . . . 179

180

B Calculus **180**
B.1 Integral of trigonometric functions . . . 180
B.2 Calculus of n_{ij}, spherical case . . . 182

185

C Supplementary simulation results of model-free visual servoing **185**
C.1 Model-free servoing on the ellipsoid . . . 185
C.1.1 Using the straight line-based method . . . 185
C.1.2 Using the quadric surface-based method . . . 193
C.2 Simulations with realistic ultrasound images . . . 198
C.2.1 Straight line-based estimation . . . 198
C.2.2 Quadric surface-based estimation . . . 198
C.3 Simulations with the binary volume . . . 203
C.3.1 Straight line-based estimation . . . 203
C.3.2 Quadric surface-based estimation . . . 203

Chapter 1

Introduction

Robots are machines but dedicated to perform not a unique static task. They are designed and endowed with a relative monitored freedom in such a way they can deal with dynamic requirements. Their designed body structure allows them performing different kind of autonomous actions and therefore interacting with their environment with predefined goals. These interactions can also lead to exchanged forces between the robot and the environment. Robotic actions are generated by actuators embedded in the robot structure. The robot can perform an action only if the latter is ordered and well formulated according to robots's own language, provided of course that the required action fits and lies within the robot's capabilities. This language is that the robot's actuators understand and thus accordingly generate an action, that will be transmitted to the robot's structure. The actions separately generated by each of the actuators will result in an action at the structure's end-element. The robot is servoed to perform a task in its environment, and therefore needs information about this latter in order to be able to interact with it. Such information are generally afforded thanks to sensors attached to the structure of the robot. They can be either proprioceptive or exteroceptive allowing respectively sensing the state of the robot or sensing that of the environment. The task to be performed by the robot is conceived in a language different from that understandable by the robot's actuators. Such task orders can be formulated, as for examples, by: move to position A then to position B; perform motion with a certain velocity and then smoothly stop right arriving to a certain position; grab the door and then correctly fix it in the car body; push the surface with a certain force and perform back-and-forth motions for polishing; perform welding by following a certain path; etc. The task orders can not be directly communicated to the robot since the latter's actuators do not understand the language with which the ordered task is formulated. The actuators can perform according to orders formulated only in actuator's language. A buffer between the two languages is consequently crucial to translate the orders to be thus understood and then accomplished by the robot. The technical field related to such buffers is well known by the term *Automatic Control* in general, when dealing with machines, and more particularly by

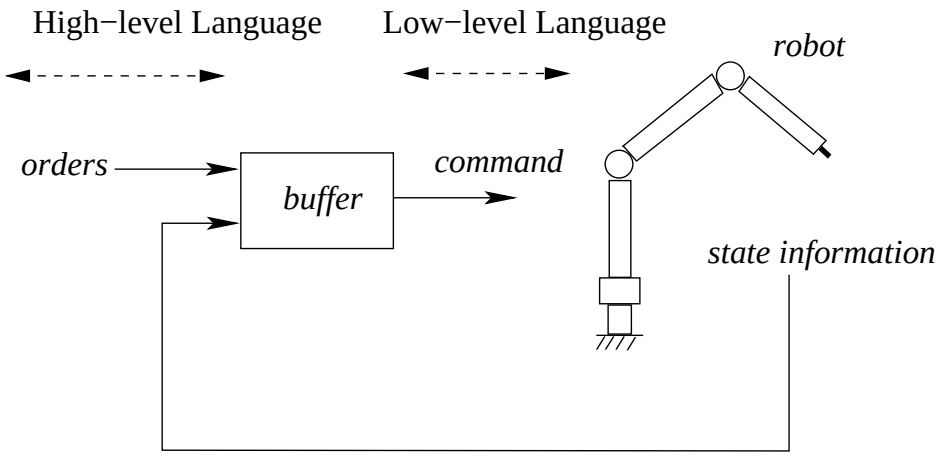


Figure 1.1: Sketch about robotics control

Robot Control when dealing with robots. The sensors provide with robot's or environment's state information that are fed back to the buffer, that then computes the commands which finally are sent to the robot. A sketch is given in Fig. 1.1.

Depending on kind of the task to be performed by the robot, different types of sensors are considered. In the case only the proprioceptive sensors, as the robot's encoders for example, are used to convey the information relative to the pose of the robot, the servoing technique is known as *Position-based Servoing*. Such techniques require prior knowledge about the considered environment, as a CAD model representing its geometry for instance. They are prone to errors in the task accomplishment if a change has occurred in a considered part of the environment. An alternative consists in using exteroceptive sensors, as vision ones that can enable the robot perceiving the environment with which it is interacting. This approach is well known as *Visual Servoing* (VS) technique, that we draw a global scheme on Fig. 1.2, grossly representing the different involved steps with the corresponding data flow.

Visual sensors provide an image of the environment, thus reflecting its state. The information contained in the image is extracted and then fed back for robot servoing. In the case the information is directly used to compute the command to the robot, the visual servoing technique is referred by *Image-based* visual servoing (IBVS) technique. If however the information is processed to be transformed in 3D poses information, that is used to compute the command, then the visual servoing technique is referred by *Position-based* visual servoing (PBVS) one. Otherwise, part of the information is transformed in poses inputs which are then compounded with other image information to compute the command. In this case we refer to *Hybrid Visual Servoing* technique. Reviews are presented in [41] and [17, 18] . In

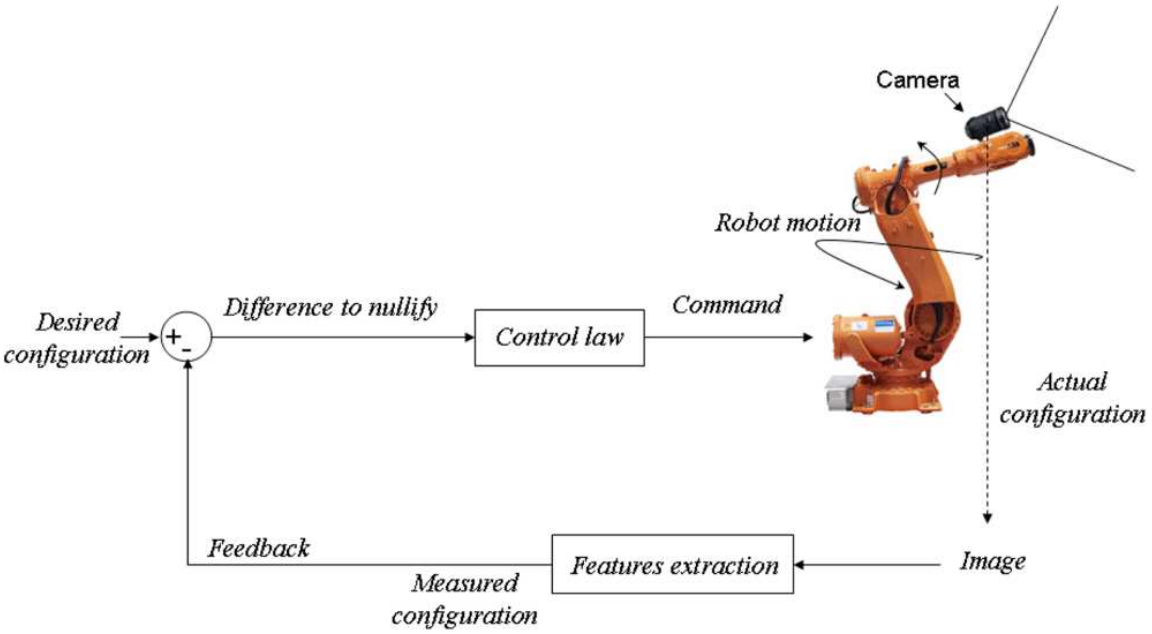


Figure 1.2: A typical visual servoing scheme.

visual servoing, the feedback information used for computing the command is referred to as *visual feature*.

Robotics has come into being with a main objective to enhance the capabilities of humans and to afford what the latter could not. It was in fact a follow-up of the development of mechanical machines, which at that time already afforded the human with valuable services. Such machines were however restrained for performing a unique task and were limited in autonomy. This fueled the desire to make them versatile with a broad range of services and with as higher as possible autonomy. More, investigations have already been undertaken to make these machines smart, even with higher skills than human. Much of the efforts therefore has been, and still are being in an increasing rate, devoted for enhancing the robots autonomy and capabilities, as we have taken part through this thesis.

Robotics finds applications in numerous areas ranging from, but not limited to, the field of automotive industry, aerospace, under-water, nuclear, military, and recently in the medical intervention field. The latter represents the field this thesis is mainly targeting. We introduce this area in Chapter 2. Visual sensors afford robotic systems with perception of their environment and consequently with more abilities for autonomous actions with enhanced safety. Such sensors thus are of great interest, perhaps indispensable, for many applications

of the medical robotics field, where the environment with which the robot is interacting is typically difficult to model. Possible continual environment's state changes, that may occur, make such difficulties stronger. Many of the medical robotic systems use, indeed, visual sensors, and therefore are endowed with capabilities of interacting with their environment. Those sensors are generally based on modalities such as optical, magnetic resonance (MR), X-ray fluoroscopy or CT-scan, ultrasound, etc. We provide in the next chapter a review about robotic systems guided with these imaging modalities, that we present in more details for the case of ultrasound, since our work concerns this latter field.

A gap, however, still remains to be addressed before medical robotics become common place for large applications range, due mainly to the fact that the information provided by most of such sensors is not yet well exploited in servoing. Efforts are therefore needed to deal with such issue and investigate how those sensors could be used, their information exploited and translated in a language understood by the robot (i. e., new modeling along with visual servoing techniques needs to be developed), so the latter behaves accordingly and achieves the required medical task. This thesis concerns such objectives, and more particularly it investigates how 2D ultrasound sensors, through their valuable information, can be exploited in medical robotic systems in order to afford the latter with enhanced autonomy and capabilities.

Contributions

Our work concerns the exploitation of 2D ultrasound images in the closed loop of visual servoing scheme for automatic guidance of a robot arm, that carries at its end-effector a 2D ultrasound probe; we consider in this work 6 degrees of freedom (DOFs) anthropomorphic medical robot arms. We develop a new visual servoing method that allows for automatic positioning of a robotized 2D ultrasound probe with respect to an observed soft tissue [54] [57] [55], and [56]. It allows to control both the in-plane and out-of-plane motions of the 2D ultrasound probe. This method makes direct use of the observed 2D ultrasound images, continuously provided by the probe transducer, in the servoing loop (see Fig. 1.3). It exploits the shape of the cross-section lying in the 2D image, by translating it in feedback signals to the control loop. This is achieved by making use of image moments, that after being extracted are compounded to build up the feedback visual features (an introduction about image moments is given in Chapter 3). The choice of the components of the visual features vector is also determinant. These features are transformed in a command signal to the probe-carrier robot. To do so, we first develop the interaction matrix that relates the image moments time variation to the probe velocity. This interaction matrix is subsequently used to derive that related to the chosen visual features. The latter matrix is crucial

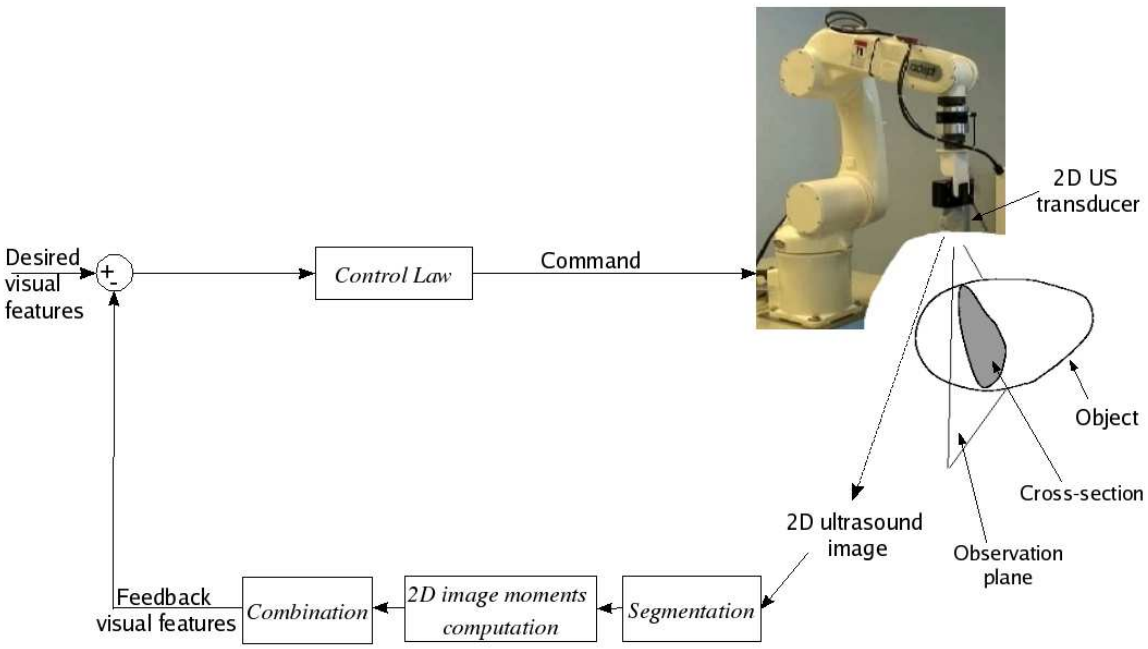


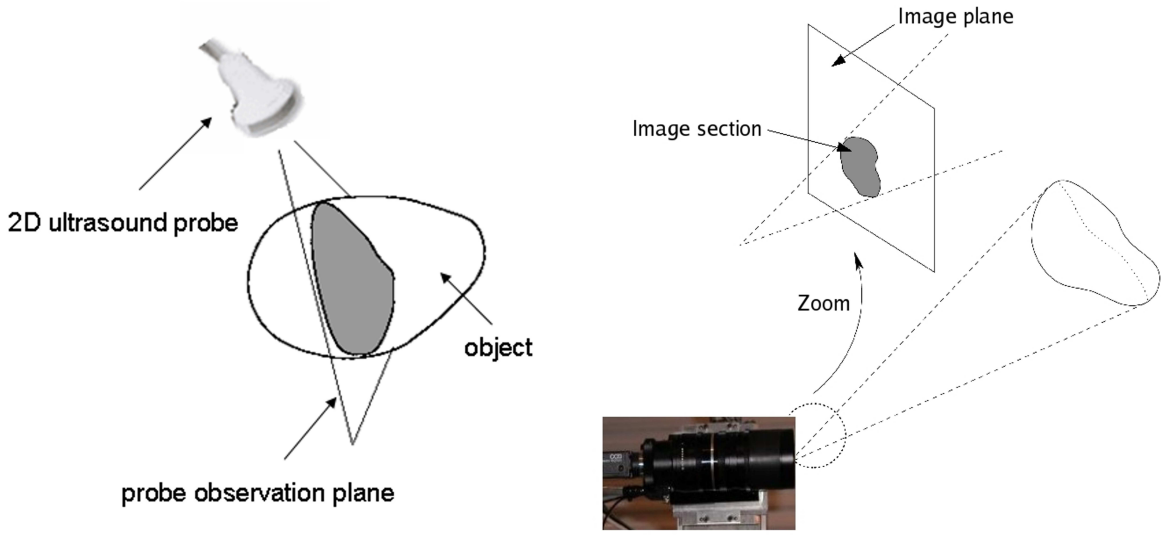
Figure 1.3: An overall scheme of the ultrasound (US) visual servoing method using image moments, with the corresponding data flow.

in the design of the visual servo scheme, since it is involved in the control law. We propose six relevant visual features to control the 6 DOFs of the robot. The method we develop allows for automatic reaching a target image starting from one totally different, and does not require a prior calibration step with regard to parameters representing the environment with which the probe transducer is interacting. It is furthermore based on visual features that can be readily computed after having segmented the cross-section of interest in the image. These features do not warp but truly reflect the information conveyed by the image. They are unlikely to misrepresenting the actual information of an image from which they are extracted. These features are moreover relatively robust to image noise, which is of great interest when dealing with the ultrasound modality whose images are, inherently, very noisy. An image moments-based servoing system, namely the one presented in the present dissertation, will then be, at its turn, robust to image noise. We will see this in Chapter 5.

The method we propose has numerous potential medical applications. First, it can be used for diagnosis by providing an appropriate view of the organ of interest. As instance, in [1] only the probe in-plane motions are automatically compensated to keep tubes centered in the image. However, if the tubes are for example curved, they may vanish from the image while the robotized probe is manipulated by the operator. Indeed, compensating only in-plane motions is not enough to follow such tubes. With the method we propose, however, it would be possible that the probe automatically follows the tubes's curvatures thanks to the compensation of the out-of-plane motions. Another potential applications is needle insertion. Since the method we propose allows to keep the actuated probe on an organ desired cross-section, it therefore would afford to stabilize an actuated needle with respect to the targeted organ. This would prevent the needle from eventual bending or breaking when the organ moves. The assumption and constraint assumed for example in [38], where the needle is mechanically constrained to lie in the probe observation plane, thus would be overcome since the system would automatically stabilize the needle in the desired plane (organ's slice). Another application is image 3-D registration, where currently in the Lagadic group we have a colleague who works to exploit this method for that topic.

This thesis brings and states new modeling of the ultrasound visual information with respect to the environment with which the robot is interacting. It is important to notice the difference from the modeling of optical systems visual information, for example, which can be found in different literature works. In case of optical systems, like a camera for example, the transmitted image conveys information of 3D world scenes that are projected on the image plane. In contrast, a 2D ultrasound transducer transmits a 2D image of the section resulting from the intersection of the probe observation beam with the considered object. In practice, the ultrasound beam is approximated with a perfect plane. A 2D ultrasound probe thus provides information only in its observation plane but none outside of it. Consequently, the modeling in case of optical systems quite differs from that of 2D ultrasound systems (this contrast is sketched in Fig. 1.4). Most of the visual interaction modeling, and thus visual servoing methods, are however devoted for optical systems. Therefore, they can not be applied in case of 2D ultrasound due to the highlighted difference. New modeling need therefore to be developed in order to design visual servoing systems using 2D ultrasound. We first derive the image velocity of points of the cross-section ultrasound image. This velocity is analytically modeled, and is related as function of the probe velocity. It is then used for deriving the analytical form of the image moments time variation as function of the probe velocity. This latter formulae we obtain is nothing but the crucial interaction matrix required in the control law of the visual servoing scheme. The modeling is developed and presented in Chapter 3.

Another challenging issue is that the interaction matrix strongly depends on the 3D shape



(a) (b)

Figure 1.4: Difference between an optical system and a 2D ultrasound one in the manner they interact with their respective environments: (a) a 2D ultrasound probe observes an object, through the cross-section resulting from the intersection of its planar beam with that object - (b) a perspective camera observes two 3D objects, which reflect rays that are projected on the camera's lens. (The camera picture, at the top, is from http://www.irisa.fr/lagadic/).

of the soft tissue with which the robotic system is interacting, when probe out-of-plane motions are involved. A first resolution that could be proposed is the use of a pre-operative 3D model, of the considered soft tissue, that would be used to derive the interaction. However, doing so would arise difficulties along with more challenges. Firstly, the pre-operative model should be available. This suggest an off-line procedure in order to obtain it. Furthermore, it would also require to register the pre-operative model with the current observed image. The above issue is addressed in the present dissertation. Indeed, we develop an efficient model-free visual servoing method that allows the system for automatic positioning without any prior knowledge of the shape of the observed object, its 3D parameters, nor its location in the 3D space. This model-free method efficiently estimates the 3D parameters involved in the control law. The estimation is performed on-line during the servoing is applied. This is presented in Chapter 4.

The developed methods have been validated from simulations and experiments, where promising results have been obtained. This is presented in Chapter 5. The simulations

consist in scenarios where a 2D virtual probe is interacting with either a 3D mathematical model, a realistic object reconstructed from a set of real B-scan ultrasound images previously captured, or a binary object reconstructed from a set of binary images. The experiments have been conducted using a 6 DOFs medical robot arm carrying a 2D ultrasound probe transducer. The robot arm was interacting with an ultrasound phantom which, inside, contained a soft tissue object, and also with soft tissue objects immersed in a water-filled tank.

We finally conclude this document by providing some orientations for prospective investigations.

Chapter 2

Prior Art

The focus of this thesis is robot automatic guidance from 2D ultrasound images. More precisely, the objective of our investigations is to develop new modeling for image-based visual servoing. It is therefore necessary to position our work between the former ones that dealt with robot guidance from 2D ultrasound, and thus the contributions that this thesis brings can also be contrasted from those of the literature works. This is the scope of the present chapter. In this dissertation, in fact, we develop new methods aimed at more effective and broad exploitation of an imaging modality, namely the ultrasound imaging, for medical robotics control. Consequently, it seems fundamental to first provide an overview about medical robotics, from the point of view of robotics control, and to introduce medical robot guidance performed with main imaging modalities. After doing so, we finally can start dealing in more details with works that investigate the use of the ultrasound images for robot control.

The remainder of the chapter is organized as follows. We present in the next section a short introduction to medical robotics, along to human-machine interfaces. These latter are commonly used for the intercommunication between the clinician and the medical robotic system for procedure monitoring. We also provide a classification that each of which reflects a specific manner that, according to, the clinician interacts and orders the robotic system for task achievements. Subsequently, we introduce the most used imaging modalities as optical, X-ray and/or CT, MRI, and ultrasound. The ultrasound modality represents the imaging whose employing, in guiding automatic robotic procedures, is investigated in the present thesis. Therefore, those remaining imaging modalities are briefly presented. The examples of literature investigations related to those modalities are provided only to illustrate their corresponding field. We thus generally cite only one work for each of those fields, since they are beyond the focus of this thesis. As for works dealing with ultrasound-based automatic guidance, we finally present and organize them according to a certain classification, as can be seen later. We afterwards briefly recall the contributions that this thesis brings to the

Figure 2.1: Da Vinci robot (Photo: www.intuitivesurgical.com)

field of 2D ultrasound-based robotic automatic guidance.

2.1 Medical robotics

Some parts of this section are inspired from [78].

Medical robotics has come into being to enhance and extend the clinician capabilities in order to perform medical applications with better precision, dexterity, and speed leading to medical procedures of shortened operative time, reduced error rate, and of reduced morbidity (see [78]); its goal is not to replace the clinician. As examples to illustrate such objectives, robotic systems could compensate for the surgeon's hand tremors to remove them during an intervention, or could be used to carry heavy tools with care. These systems could assist and provide the clinician with valuable information which are organized and displayed on screens for visualization. The clinician could interact with the system to obtain desired information, on which correct decisions can be made. The conveyed information have therefore to be pertinent with at the same time not overwhelming the clinician.

Medical robots can be classified according to different ways [78]: by manipulator design (e. g., kinematics, actuation); level of autonomy (e. g., programmed, teleoperated, constrained cooperative control); targeted anatomy or technique (e. g., cardiac, intravascular percutaneous, laparoscopic, microsurgical); intended operating environment (e. g., in-scanner, conventional operating room); or by the devices used for sensing the information (e. g., camera, ultra-

sound, MR, CT, etc). An example of a well known medical robot is shown on Fig. 2.1. Such robot is used for minimally invasive surgical procedures.

In contrast to industrial robots that generally deal with manufactured objects, medical robots instead interact with human patients. Therefore, much constraints and difficulties arise when dealing with medical robotics. The security is one of the requirements that medical robotics typically must fulfill. Consequently, such robots are rigorously expected to possess accuracy, and dexterity. The versatility is also of great interest allowing to perform a range of robotized medical procedures with minimal changes to the medical room setup. The robot should not be cumbersome in order to allow the clinical staff unimpeded access to the patient, especially for the surgeon during the procedure. It can be ground-, ceiling-, or patient-mounted. Such choice is subject to the tradeoff between the robot size, heaviness, and access to the patient. Sterilization also must be addressed, especially for surgical procedures. The patient can be in contact with parts of the robot, and consequently all precautions must be taken in order to prevent any possible contamination of the surgical field. The common practice for sterilization is the use of bags to cover the robot, and either gas, soak, or autoclave steam to sterilize the end-effector holding the surgical instrument.

As introduced above, medical robotic systems use mainly visual sensors, whose modality is chosen depending on the kind of the application to perform. Each modality presents specific advantages but also suffers from drawbacks. Soft tissues, for example, are well imaged and their structures well discriminated with the Magnetic Resonance Imaging (MRI). This modality is extensively used to detect and then localize tumors for their treatment, and is subject to different investigations to exploit it for robotized tumor treatment, where the robot could assist needle insertion for better tumor targeting (e. g., [30]). Such imaging is afforded by scanners of high intensity magnetic field. Therefore, ferromagnetic materials exposed to such field undergo intense forces and could became dangerous projectiles. Consequently, common robotic components do not apply since they are generally made from such materials, and are therefore precluded for this imaging modality. Moreover, the streaming rate at which the image are provided by the current MRI systems is relatively low to envisage real-time robotic applications. As for bones, they are well imaged with X-ray modality (or CT). Such imaging has been therefore the subject to investigations and has found its use, for example, in robotically-assisted orthopedic surgery as spine surgery, joint replacement, etc. This modality can, however, be harmful to the patient body due to its radiation. Optical imaging sensors have also been considered. One of the most medical application using such sensors concerns endoscopic surgery, where generally a small camera is carried and passed inside the patient's body through a small incision port, while two or more surgical instruments are passed through separate other small incisions (see Fig. 2.2). The camera is positioned in such a way it gives an appropriate view of the surgical instruments. The

Figure 2.2: Example of endoscopic surgery robot (Da Vinci robot) in action. (Photo: `http://biomed.brown.edu/.../Roboticsurgery.html`)

surgeon thus can handle those surgical instruments and can observe their interaction with soft tissues thanks to the conveyed images by the camera. Such procedures have already been robotized, where each instrument is separately carried by a robot arm. Both instruments are remotely operated by the surgeon through haptic devices. This kind of robotic systems is already commercialized, as the one shown in Fig. 2.2, and these robotized procedures have become commonplace in some medical centers. Research works are however still being conducted in order to automatically assist the surgeon, by visually servoing the instrument-holder arms (e. g., [47], [60]).

Another application of optical systems which new works have started to investigate is the microsurgery robotics (e. g., [31]). It is introduced in Section 2.2. Other applications could be considered but are however extremely invasive (e. g., [36], [7]). Therefore, the range of potential applications based on optical imaging sensors seems to be restrained to few applications as endoscopic surgery, wherein at least two incisions are required, leading to possible hemorrhage and trauma for the patient. Bleeding can also hinder and, perhaps, preclude visualization if blood encounters the camera lens, thus compromising the procedure. Optical sensors require free space up to the region to visualize, which represents a strong constraint that generally could not be satisfied when dealing with medical procedures; where the camera is inside the body and encounters soft tissue walls from either sides. The camera also needs to be passed inside the body up to the region to operate on, which is however not always possible for some regions. We can note indeed that, as instance, most of endoscopic procedures are laparoscopically performed (i.e., through the abdomen), and thus the camera along with the instruments is passed through a patient body's region that is relatively less

(a) (b)

Figure 2.3: An example of a typical robotic system teleoperated through a human-machine interface: three medical slave robot arms (left) are teleoperated by a user thanks to a master handle device, and the procedure is monitored by the user through display screens (right). (Photo: `http://www.dlr.de/`).

complicated in term of access since, for example, the fewer presence of bones. In contrast, MR, X-ray, and ultrasound imaging modalities provide internal body images without any incision, and thus circumvent the constraints imposed when using optical systems and their effects. But as introduced above, MRI and X-ray present drawbacks. The former modality currently does not provide images in real-time, and precludes ferromagnetic materials. The latter is harmful. Ultrasound modality, however, provides internal body images non-invasively and is considered healthy for patient. More particularly, 2D ultrasound provides images with high streaming rate. This latter trait is of great interest when dealing with robot servoing for real-time applications. This thesis concerns this modality, where it aims at addressing the issue of exploiting 2D ultrasound images for automatically performing robotized medical applications.

During a medical procedure, it is crucial that the clinician is present to supervise and monitor the application. Therefore the clinician should be able to order and interact with the robot. This is performed through an interface well known by the term of *Human-machine interface*.

2.1.1 Human-machine interfaces

Human-machine interfaces (HMI) play an important role in medical robotics, more particularly they allow the clinician for supervising the procedure. An HMI is grossly composed of a display screen on which different information are displayed, and a handle device with which the clinician can send orders to the robotic system. Such device could be a joystick, or sim-

ply a mouse with which hand clicks are performed on the display screen. The clinician thus can interactively send the orders to the robot through the HMI, and inversely, can receive information about the clinical field's state (see Fig. 2.3). However, the clinician should receive important and precise information, while at the same time not be overwhelmed by such data in order to take decisions based only on pertinent information. An issue is the ability of the system to estimate the imprecision of the conveyed information, such as registration errors, in order to prevent the clinician making decisions based on wrong information [78]. An example of a human-machine interface developed for robotically assisted laparoscopic surgery is presented in [61].

2.1.2 Operator-robot interaction paradigms

Depending on the configuration reflecting the manner the operator commands the robotic system, different paradigms could be considered, as those presented in the following.

Self-guided robotic system paradigm

In such a configuration, the robot autonomously performs a series of actions after a clinician had previously indicated required objectives. That operator is in fact out-of-loop with regard to the interaction of the robot with its environment, except for restrained actions such as monitoring the development of the procedure and defining new objectives for the robot, or stopping the procedure. Endowed with such a paradigm, a robotic system could afford with valuable services that otherwise could not be performed. Such a system requires therefore intelligent closed-loop servoing techniques to enable the robot undertaking autonomous actions, especially when interacting with complex environments. The servoing techniques developed through this thesis are ranged mainly within this paradigm class.

In contrast to this configuration, the below presented paradigms consist is the case where the operator is involved within the interaction loop. Such configurations can therefore be considered, with regard to the task to perform, belonging to the open-loop servoing classes.

Haptic interfaces: master-slave paradigm

Haptic interface systems have brought pertinent assistance for medical interventions. Typical systems consist of robot arms that can carry different variety of medical instruments (see Fig. 2.3 top). By handling master devices, the clinician manipulates the instrument carried by the robot end-effector (see Fig. 2.3 bottom). The clinician can remotely manipulate the robot, and can feel what is being done thanks to reflected forces from the instrument (e. g., [49]). Force sensors attached between the carried instrument and its holder estimate the

Figure 2.4: Cooperative manipulation: a microsurgical instrument held by both an operator and a robot. Device, developed by JHU robotics group, aimed at injecting vision-saving drugs into tiny blood vessels in the eye (Photo: `http://www.sciencedaily.com`).

forces applied on the manipulated patient's tissue. The forces encountered by the instrument are sensed, scaled, and then sent to the master handle. This latter moves according to these sent forces, and thus it reflects the sensed forces to the clinician who is operating on it. The clinician therefore can feel the sensed forces and consequently can be aware about the effects of the interaction between the instrument and the patient's tissue. Inversely, the forces applied by the clinician on the master handle are scaled, transmitted, and then transformed in motions of the slave instrument. Intercommunicating forces as such allows to effectively slowing down abrupt motions that could be the result from backlash movements of the operator, and to attenuate hand tremor which can be of great interest for surgical procedures. It however does not allow the operator direct access to the instrument, which thus can not be freely manipulated (see [78]).

One known application of the master-slave paradigm concerns endoscopic surgery. Such procedures (they have been introduced above), whether robotically or freehand performed, suffer from low dexterity because of the effect of the entry port placement, through which the surgical instrument or the camera holder is passed. Another application concerns microsurgery robotics (it is introduced in Section 2.2). It suffers however from the fact that current master-slave systems are not reactive to small forces.

Figure 2.5: Hand-held instrument for microsurgery. (Photo: `http://www3.ntu.edu.sg/`).

Cooperative manipulation

In this case, both the clinician and the robot hold the same instrument, e. g. [31], (see Fig. 2.4). This paradigm keeps some advantages of the master-slave one, since it allows effectively slowing down abrupt surgeon's hand motions, and attenuating surgeon's hand tremor. In contrast to master-slave, this paradigm allows the surgeon to directly manipulate the instrument, and be more closer to the patient, which is really appreciated by surgeons [78].

Hand-held configuration

Another configuration consists in hand-held instruments (see Fig. 2.5), that find success in hand tremor cancellation (e. g. [85]). Embedded inside the instrument are inertial sensors that detect tremor motions and speed which both, by low amplitude actuators, are then inertially canceled. The advantage of such a configuration is that beyond of leaving the surgeon completely unimpeded, it lets the operating room uncumbersome, with less setup changes. However, heavier tools are not supported and the instrument can not be left stationary in position [78].

After we have presented an introduction to the medical robotic field, we now survey exploitation of main imaging modalities in guiding such systems. We first introduce medical robotic systems guided with optical images. Then, we present robotic guidance with X-ray (or CT-scan) and MRI imaging modality, respectively. They are discussed briefly, such that we present only few examples for illustration, since they are beyond the scope of this thesis. Finally, we consider guidance using the ultrasound modality. We discuss it with more details, since it represents the focus of this thesis. In particular, we provide a

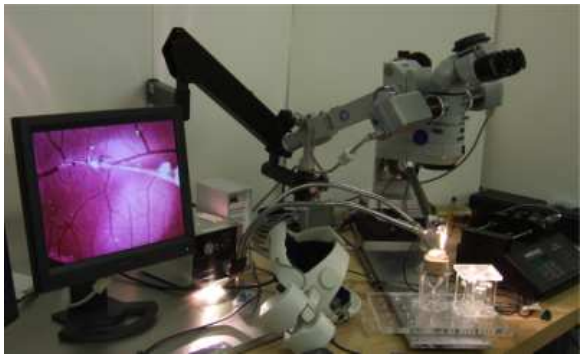


Figure 2.6: Microsurgery robotics: micro-surgical assistant workstation with retinal-surgery model. (Photo: `http://www.cs.jhu.edu/CIRL/`).

detailed survey on works that are investigating the exploitation of 2D ultrasound imaging for automatic guidance of medical robotic systems, as the work presented in this dissertation.

2.2 Optical imaging-based guidance: microsurgery robotics

Since endoscopic robotics, introduced above in Section 2.1, have become commonplace in the medical field, only microsurgery robotics is considered in this section. Microsurgical robotics is nothing but surgical robotics related to tasks performed at a small scale, e. g. [31], (see Fig. 2.6). The typical sensor used to provide visual information about the soft tissue environment is the microscope. In contrast to free hand performed microsurgery, robots enhance the surgeon capabilities for performing tasks with fine control and precise positioning. In many cases, microsurgical robots are based on force-reflecting master-slave paradigm. The clinician remotely moves the slave by manipulating the master and applying forces on it. Inversely, the forces encountered by the slave are scaled, amplified, and sent back to the master manipulator that moves accordingly. The operator thus can feel the encountered forces, and therefore is aware about the forces applied on the manipulated soft tissue. Furthermore, this configuration allows to produce reduced motions on the slave. Accordingly, this paradigm considerably prevents the manipulated soft tissue from possible damages that can be the result of abrupt operator's hand motion with/or high applied forces. This configuration however suffers from two main disadvantages. One disadvantage consists in the complexity and the cost of such systems, since they are composed of two main mechanical systems: the master and the slave. Also, such a configuration does not allow the clinician directly manipulate the instrument [78]. Micorsurgery robotics finds application,

Figure 2.7: ACROBAT robot in orthopaedic surgery aimed at hip reparation. (Photo: `http://medgadget.com`).

as instance, in the domain of ophthalmic surgery (e. g., [31]).

2.3 X-ray-based guidance

A well-known application of X-ray imaging is orthopaedic surgery. In orthopaedic surgery robotics (see Fig. 2.7), the surgeon is assisted by the robot in order to enhance the procedure performance. As in knee or hip replacement, rather than the bone is manually cut, it is automatically performed by the robot, under the supervision of the surgeon. This allows to effectively cut the bone in such a way to appropriately machine the desired hole for the implant. Preoperative x-ray images provide key 3D points used for planning a path that the robot will then follow during the cutting procedure.

Since bones are easily well imaged with computed X-ray tomography (CT) or X-ray fluoroscopy modalities, the employed visual sensors are based on these modalities. During the surgical procedure, the patient's bones are attached rigidly to the robot's base with specially designed fixation tools. The image frame pose is estimated either by touching different points on the surface of the patient's bones or by touching preimplanted fiducial markers. The surgeon manually brings and position the robot surgical instrument at the bone surface to operate on. Then, the robot automatically moves the instrument to cut the desired shape, while in the same the robot computer controls the trajectory and the forces applied on the bones. Since security must be rigorously addressed in surgical robotics, different checkpoints are predefined in order to allow the surgical procedure to be restarted

if it was prematurely stopped or paused for whether reasons.

For better security of bone machining, the presented robotic system configuration can be enhanced with the constrained hand guiding configuration. The robot is constrained by the computer so that the cutter remains within a volume to be machined [42].

One of the first prototype of orthopaedic surgery robotics was developed in the late 1980's, named ROBODOC system [77], and its first clinical use was in 1992 [78]. A similar robot is shown in Fig. 2.7. Nowadays, hundreds of orthopaedic robots are present in different hospital centers, and over thousands of surgical operations have been performed with such systems. However, before a medical robot system is clinically used, battery of tests have to be performed to validate the system and thus, ensure total security of the patient and the clinician staff during the surgical operation. Of course, the system must demonstrate enhancements in the surgical procedure performance as precision, dexterity, etc, to justify its use rather than the surgical operation is manually performed.

X-ray images have also been considered for image-based visual servoing. A robotic system for tracking stereotactic rode fiducials within CT images is presented in [24]. The image consists in a cross-section plane wherein the rods appear as spots. Those rods are radiopaque in order to ease their visualization in the X-ray (CT) images. The objective is to automatically position the robot in such a way the spots are kept at desired positions in the image. To do so, an image-based visual servoing was used, where the spots image coordinates constitute the feedback visual features. From each new acquired image the spots are extracted to update the actual visual features, which then are compared to that of the desired configuration. The according inferred error is used to compute the control law which, at its turn, is ordered to the robot in form of control velocity. Since the jacobian matrix relating the changes of the visual features to the probe velocity is required, that related to the spots image coordinates is presented in [24]. To do so, the rodes are represented with 3D straight lines whose intersection with the image plane is analytically formulated. The system has been tested for small displacements from configuration where the desired image related to desired spot's coordinates is captured. The issue investigated in [24], the modeling aspect more precisely, in fact can be ranged within the scope of this thesis. Indeed, in [24], the image used in the servoing loop provides a cross-section sight of the environment with which the robot is interacting. Similarly, this thesis deals with cross-section images in the servoing loop, except that these images are provided by a 2D ultrasound transducer. A big difference is that only simple geometrical primitives, namely straight lines, are considered in [24], while this thesis deals with whatever-shaped volume objects. We present in this document a general modeling method, that, indeed, can be applied to the simple case of straight lines, as described in Section 3.7.3.

(a) (b)

Figure 2.8: MRI-based needle insertion robot (a) High field MRI scanner (Photo: `http://www.bvhealthsystem.org`) - (b) MRI needle placement robot [30] (Photo: `www2.me.wpi.edu/AIM-lab/index.php/Research`).

2.4 MRI-guided robotics

MR imaging systems, as X-ray ones, provide in-depth images of observed elements. However, MRI systems provide images non-invasively and thus are considered not harmful for patient body. Moreover, they provide well contrasted images of soft tissues. This advantages stimulated different investigations in order to exploit this modality for automatically guiding robotized procedures. In [30], for example, a pneumatically-actuated robotic system guided by MRI for needle insertion in prostate interventions is presented. A 2 DOFs robot arm is used to automatically position a passive stage, on which a manually-inserted needle is held [see Fig. 2.8(b)]. Inside the room of a MRI scanner [e. g., see Fig. 2.8(a)], the patient is lying in a semi-lithotomy position on a bed. Both the robot arm holder, a needle insertion stage, and the robot controller are also inside the scanner room, while the surgeon is in a separated room to monitor the procedure through a human machine interface. The main issue while dealing with a MRI scanner consists in the difficulty for the choice of compatible devices. Due to the high magnetic field in the MRI scanners, ferromagnetic or conductive materials are precluded. Such materials can for instance either be dangerously projected, cause artifacts and distortion in the MRI image, or create heating near the patient's body. Most of the standard available devices are however made from either materials, and therefore are not compatible with the MR modality.

It is proposed in [30] the use of pneumatic actuators, that have been tailored since the non

total MRI-compatibility at their original state. The manipulator is located near the bed in the scanner room, for the interaction with the patients body, where its end-effector pose is detected thanks to attached fiducial markers extracted from the observed MRI image. In order to avoid electrical signals passing through the scanner room and thus keeping the image quality, the robot controller is placed in a shielded enclosure near the robot manipulator, and the communication between the control room and the controller is through a fiber optic Ethernet connection. A PID control law is used for the pneumatic actuators servoing.

During the procedure, the surgeon indicates both a target and a skin entry point. Accordingly, the robot automatically brings the needle tip up to the entry point with a corresponding orientation. Subsequently, through the slicers of the human-machine software interface, the surgeon monitors the manual insertion of the needle, which then slides along its holder axis to reach the target. The use of the MR images is limited to detect the target and needle tip locations. The automatic positioning of the robot up to the entry point is afforded with a position-based visual servoing. Such an approach however is well-known for its relatively low positioning accuracy, if compared for example to the image-based visual servoing. The main contribution presented in [30] seems in fact consisting in the design of a MRI-compatible robotic system.

The propulsion effect that a magnetic field can apply on ferromagnetic materials has been exploited to perform automatic positioning and tracking of untethered ferromagnetic object, using its MRI images in a visual servoing loop [28]. The MR field is used both to measure the position of the object and to propel the latter to the desired location. Prior that the procedure takes place, a path through which the object has to move is planned off-line. It is represented by successive waypoints to be followed by the object. During the procedure that is performed under a MR field, the actual position is measured and compared to the desired one of the planned path, and the difference is sent to a controller that uses it to compute the magnetic propulsion field to be applied on the object. That propulsion is expected to move the object from the actual position to that desired. Experimental results are reported, such that the system was tested using both a phantom and a live swine under general anesthesia. The feedback was updated at a rate of 24 Hz for the phantom case. The in-vivo objective was to continuously track and position the object in such a way it travels within and along the swine's carotid artery by following the pre-planned path. The object consisted in a 1.5 mm diameter sphere made of chrome and steel. The proposed visual servoing method consists however in a position-based one. As mentioned just above, it is well-known for its relatively low positioning accuracy.

The limitation that MRI systems currently suffer from consists (to our knowledge) in

the low streaming rate at which the images are provided. This considerably hinders the exploitation of such images for real-time robotic guidance applications. Image-based visual servoing, for example, requires that the update along with the processing of the image has to be performed within the rate at which the robot operates. The 2D ultrasound modality, nevertheless, beyond of being non-invasive, provides the images at a relatively high streaming rate. This makes such a modality a relevant candidate for real-time robotic automatic-guidance applications where in-depth images are required.

2.5 Ultrasound-based guidance

Ultrasound imaging represents an important modality of medical practice, and is being the subject of different investigations for enhanced use. Ten years ago, one out of four imaging-based medical procedures was performed with this modality and the proportion is increasing for different applications in the foreseeable future [84].

We report in this section investigations that deal with automatic guidance from the ultrasound imaging modality. In particular, we survey in more details works dealing with the use of 2D ultrasound images for automatically guiding robotic applications, as it is the scope of our work presented in this document. The remainder of this section is organized as follows. First, in Section 2.5.1, we present an example of an investigation about the use of the ultrasound modality to simulate and then to plan the insertion of needle in soft tissue. Then, we present in Section 2.5.2 works that exploit 3D ultrasound images to guide surgical instruments, where the objective was either positioning or tracking. Afterwards, the works that deal with guidance using 2D ultrasound are surveyed. We classify them into two main categories depending on whether the 2D ultrasound image is only used to extract and thus to estimate 3D poses of features used in position-based visual servoing, or the 2D ultrasound image is directly used in the control law. The former, namely 2D ultrasound-guided position-based visual servoing, is presented in section 2.5.3, while the latter, namely 2D ultrasound-guided image-based visual servoing, is presented in Section 2.5.4.

2.5.1 Ultrasound-based simulations

In [23], a simulator of stiff needle insertion for 2D ultrasound-guided prostate brachytherapy is presented. The objective is to simulate the interaction effect between the needle and the tissue composed of the prostate and its surrounding region. For that, the forces, applied

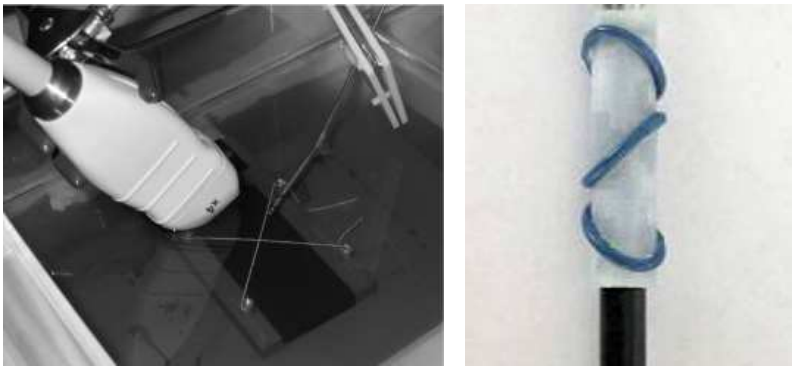
by the stiff needle on the tissue, and the tissue are modelled by making use of the information provided by the ultrasound image. A non-homogeneous phantom, composed from two layers and a hollow cylindrical object, has been made up to mimic a real configuration. The external and internal layers are designed to mimic respectively the prostate and its surrounding soft tissue, while the cylinder is designed to simulate the rectum. To mimic prostate rotation around the pubic bone, the internal layer is composed of a cylinder, with a hemisphere at each end, connected to the base of another cylinder. The elasticity of each of the two layers is represented with Young's moduli and Poisson ratios. While the Poisson ratios are pre-assigned, the objective is to estimate the Young's moduli of each layer. The forces are fitted with a piece-wise linear model of three parameters, that are identified using Nelder-Mead search algorithm [3]. When the needle interacts with the tissue, the displacements of this latter are measured from the images provided by the ultrasound probe, using *time delay estimator with prior estimates* (TDPE) [87, 88], without any prior markers inside the tissue. These measurements together with the probe positions and the measured forces are used to estimate the Young's moduli and the force model parameters. The soft tissue displacements are then simulated by making up a mesh of 4453 linear tetrahedral elements and 991 nodes, using the linear finite element method [89] with linear strain.

2.5.2 3D ultrasound-guided robotics

In the ultrasound modality, in fact, we distinguish two main modalities, that are 3D ultrasound and 2D ultrasound modalities. Works related to the former modality are presented in this section, while those related to the latter are subsequently considered. In the following, we present works where 3D ultrasound images have been exploited for automatic positioning of surgical instruments or for tracking moving target.

3D ultrasound-based positioning of surgical instrument

Subsequently in [75] and [62], a 3D ultrasound-guided robot arm-actuated system for automatic positioning of surgical instrument is presented (see Fig. 2.9). The second work follows-up and improves the system streaming speed of the first work, where 25 Hz rate is obtained instead of 1 Hz streaming rate at which the first prototype operated. The presented system consists of a surgical instrument sleeve actuated by a robot arm, a motionless 3D ultrasound transducer, and a host computer for 3D ultrasound monitoring with the corresponding image processing and for robot controlling. The objective was to automatically position the instrument tip at a target 3D position indicated in the 3D ultrasound image volume, from which the current instrument tip 3D position is estimated. A marker is at-



(a) (b)

Figure 2.9: 3D ultrasound-guided robot. (a) Experimental setup for robot tests - (b) Marker attached to the instrument tip. (Photos: (a) taken from [62], and (b) from `http://biorobotics.bu.edu/CurrentProjects.html`).

tached to the tip of the instrument in order to detect its 3D pose with respect to a cartesian frame attached to the 3D ultrasound image volume. This marker consists of three ridges of same size surrounding a sheath that fits over the instrument sleeve [see Fig. 2.9(b)]. An echogenic material is used to coat the marker in order to improve the visibility of this latter, and thus to facilitate its detection. The ridges are coiled on the sleeve in such a way they form successive sinusoids lagged by $2\pi/3$ rad. From the 3D ultrasound volume, a lengthwise cross-section 2D image of the instrument shaft along with the marker is sought to then be extracted. In such 2D image, the ridges appear as successive crests whose respective distances from a reference point lying on the shaft are used to determine the instrument sleeve 3D pose. For image detection of the crest, the extracted image is rotated in such a way the instrument appears horizontal, and then a sub-image centered on the instrument is extracted to be super-sampled by a factor of 2 using linear interpolation. The error between the estimated instrument position and the target one is fed back, through the host computer, to a position-based servo scheme based on a proportional-derivative (PD) law, with which the robot arm is servoed to position the instrument tip to the specified target. Experiments have been carried out using a stick immersed in a water-filled tank. The stick passes through a spherical bearing to mimic the physical constraints of minimally invasive surgical procedures, where the instrument passes through an incision port and consequently its movements are constrained accordingly [see Fig. 2.9(a)]. With a motion range of about 20 mm of the instrument, it is reported that the system performed with less than 2 mm of positioning error.

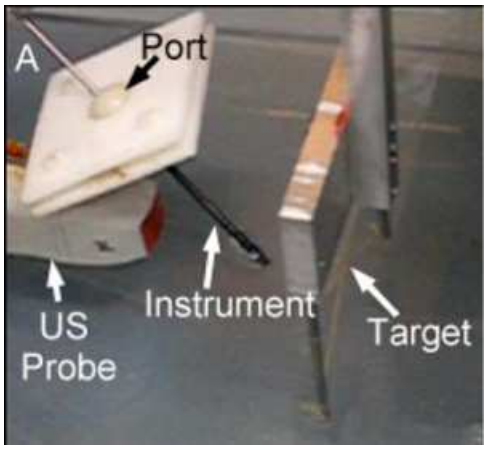


Figure 2.10: An estimator model [86] for synchronization with beating hear motions using 3D ultrasound is tested with the above photographed experimental setup. (Photo: taken from [86]).

Synchronization with beating heart motions

In [86], an estimator model for synchronization to beating heart motions using 3D ultrasound imaging is presented. The objective is to predict mitral valve motions, and then use that estimation to feed-forward the controller of a robot actuating an instrument, whose motions are to be synchronized with the heart beatings. This could allow the surgeon to operate on the beating heart as on a motionless organ. Moreover, such a system could overcome, for example, the requirements of using a cardiopulmonary bypass, and thus would spare patients its adverse effects. It was assumed that the mitral valve periodically translates along one axis, while its rotational motions have been neglected. The translational motions are then represented with a time varying Fourier series model that allows for rate and signal morphology evolving over time [63]. For the identification of the model parameters, three estimators have been tested: an Extended Kalman filter (EKF), an autoregressive model with least squares (AR), and an auto regressive model with fading memory estimator. Their performances are assessed with regards to prediction accuracy of time-changing motions. From conducted simulations, it was noted that the EKF outperformed the two other estimators, by more mitigating the estimation error especially for motions with rate changing. Experiments have been conducted on an artificial target immersed in a water-filled tank (see Fig. 2.10). The target was continuously actuated in such a way to mimic the heart mitral valve beating motions, at 60 beating per minute average rate for constant motions. A position-based proportional-derivative (PD) controller is employed for robot servoing. The system was submitted to both constant and changing rate motions. As concluded from the simulations, it was noted from the experiments that the EKF provided well predictions of

the beating heart motions compared to the others estimation approaches, with an obtained prediction error of less than 2 mm. This error is of about 30% less than that obtained with the two other estimators. In other but separate works, [36] and [7], low tracking errors have been obtained but, however, that was achieved using extremly invasive systems. In the former work, fiducial markers attached to the heart are tracked by employing a high speed eye-to-hand camera of 500 Hz streaming rate; the chest is being opened in such a way the fiducial points can be viewed by that external camera. The information conveyed by this latter are used to visually servo a robot arm that accordingly has to compensate for heart motions. As for the latter work, sonomicrometry sensors operating at 257 Hz streaming rate have been sutured to a porcine heart. Currently, 3D ultrasound modality suffers from low imaging quality along with time delayed streaming of the order of 60 ms, which could account for the relatively lower obtained performances compared to those two works (i. e., [36] and [7]).

2.5.3 2D ultrasound-guided position-based visual servoing

As has been already highlighted in this document, the 2D ultrasound imaging systems provide images at a sufficient rate to envisage real-time automatic robotic guidance. In the following, we present a survey of works that investigated the use this imaging modality in guiding automatic medical procedures. In particular, this section is dedicated to works where the image is used only in position-based visual servoing schemes. We classify these works according to the targeted medical procedure. We distinguish: kidney stones treatment; brachytherapy treatment; and tumor biopsy and ablation procedure.

Kidney stones treatment

An ultrasound-based image-guided system for kidney stone lithotripsy therapy is presented in [48]. The lithotripsy therapy aims to erode the kidney stones, while preventing collateral damages of organs and soft tissue of the vicinity. The stones are fragmented thanks to high intensity focused ultrasound (HIFU). The HIFU transducer extracorporeally emits high intensive ultrasound waves that strike the stones. The crushed stones are then naturally evacuated by the patient through urination.

For the success and effectiveness of the procedure, that can lead to shortened time of patient treatment and to spare the organs of the vicinity from being harmed, it is important to keep the stone under the pulse of the HIFU throughout the procedure. However, the kidney is subject to displacements caused by patient respiration, heartbeat, etc, and consequently the kidney stone may get out of the beam focus.

The objective of the proposed system is to keep track of the kidney stone under the HIFU transducer, throughout the lithotripsy procedure, by visual servoing using ultrasound images. The system is mainly composed of two 2D ultrasound transducers, a HIFU transducer, a stage cartesian robot whose end effector holds the HIFU transducer rigidly linked to the two ultrasound transducers, and a host computer. This latter monitors the visual servoing and the data flow through the different corresponding steps. The end-effector can apply translational motions along its three orthogonal axes in the 3D space. The two ultrasound probes, whose respective beam planes are orthogonal to each other, provide two ultrasound B-scan images of the stone in the kidney. By image processing on both the two images, the stone is identified and its position in the 3D space is determined. The inferred location represents the target 3D position on which the HIFU focal has to be. The error, between the desired position and the current position of the HIFU transducer, is fed back to the host computer that derives the control law. The command is sent to the cartesian robot that moves accordingly along its three axes in order to keep the kidney stone under its focus (i. e., thus the focus of the HIFU).

Ultrasound-guided brachytherapy treatment

A robot manipulator guided by 2D ultrasound for percutaneous needle insertion is presented in [6]. The objective is to automatically position the needle tip at a prostate desired location in order to inject the radioactive therapy seeds. The target is manually selected from a preoperative image volume. It is chosen in such a way (which is the goal of the brachytherapy) the seeds have as important as possible effect on the lesion while at the same time not harming the surrounding tissues. The robotic system is mainly composed of two robotic parts corresponding respectively to a macro and a micro robotic system, and of a 2D ultrasound probe for the imaging. The macro robot allows to bring and position the needle tip at the skin entry point, while subsequently the micro robot performs fine motions to insert and then position the needle tip at the desired location. By visualizing the volume image of the prostate, displayed on a human-machine interface, the surgeon indicates to the robot the target location where the seeds have to be dropped (see Fig. 2.11). Before that, the volume is first made up from successive cross-section images of the prostate. While the robot's end effector is rotating the 2D ultrasound probe, the latter scans the region containing the prostate by acquiring successive 2D ultrasound images at 0.7 degree intervals. The needle target position is expressed with respect to the robot frame, thanks to a previous registration of the volume image. A position-based proportional-integral-derivative (PID) controller is then fed back with the error between the needle tip current position, measured from the robot encoders, and the desired one. The command is sent to the robot, that

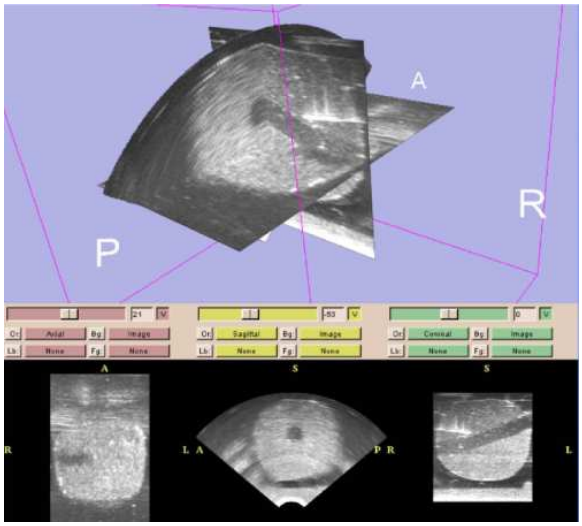


Figure 2.11: Ultrasound volume visualization through a graphical interface. Three sights (bottom) of an ultrasound volume are respectively provided by three slicer planes (top). (Photo: taken from [6]).

moves accordingly to position the needle tip at the target location. The proposed technique however is position-based, where the image is only used to determine the target location. Compared therefore to image-based servoing techniques, this method can be considered as an open-loop servoing method. As such, it has the drawback of not compensating displacements of the target that can occur during the servoing. Such displacements can be caused, as instance, by patient's body motion resulting from breathing, or by the prostate tissue shifting due to the forces it undergoes from the needle during the insertion. This lack of observed images in the servoing scheme could account for the errors obtained in the conducted experiments. The needle deflection is also not addressed. The deflection is mainly due to the forces endured by the needle during the insertion.

Ultrasound-guided procedures for tumor biopsy and ablation

A 2D ultrasound-guided computer-assisted robotic system for needle positioning in biopsy procedure is presented in [58]. The objective is to assist the surgeon in orienting the needle for the insertion. The system is mainly composed of a robot arm, a needle holder mounted on the robot's end-effector, a 2D ultrasound probe, and a host computer. The needle can linearly slide on its holder. Firstly, the eye-to-hand 2D ultrasound probe is manually positioned and oriented in order to have an appropriate view of the region to be targeted. It is then kept motionless at that configuration throughout the procedure. The observed

images are continuously displayed through a human machine interface on which the surgeon indicates the target position to be reached by the needle tip. Subsequently, the surgeon also indicates the patient's skin entry point, through which the needle will enter to reach the target. A 3D straight line trajectory is planned to then be performed by the needle tip, starting from the skin entry point to reach the target point. That trajectory is determined from the 3D coordinates of those selected two points (the entry and target point) after being expressed in an appropriate frame. The robot automatically brings the needle tip to the patient's skin entry point, in such a manner the direction of the needle intersects the target point (i.e., the needle is collinear with that straight line). The active robotic assistance ends at this stage, where the surgeon then manually inserts the needle by sliding it down to reach the target, while in the same time observing the corresponding image displayed in the interface screen. Experiments have been conducted in ideal conditions, where the target consists of a wooden stick immersed in water-filled tank. The ultrasound image is only used to determine the two target points, but is not involved in the servoing scheme. Errors of a millimeter order had been reported. Since the experiments are conducted in water, the needle does not undergo forces, which is however not the case in clinical conditions, due as instance to the interaction with soft tissue. Such forces can cause deflection of the needle, which had also been highlighted in that work.

Combining 2D ultrasound images to other imaging modalities could enhance the quality of the obtained images. In [29], an X-ray-assisted ultrasound-based imaging system for breast biopsy is presented. The principle consists in combining stereotactic X-ray mammography (SM) with ultrasound imaging in order to detect as well as possible the lesions location, and then be able of harvesting relevant samples for the biopsy. The X-ray modality provides images with high sensitivity for most lesions, but is not as safe and fast as 2D ultrasound. The presented procedure begins by first keeping motionless the patient tissue for diagnosis, by using a special apparatus. A 2D ultrasound probe scans that region of interest with constant velocity by acquiring successive 2D ultrasound images at similar distance intervals. A corresponding 3D volume is made up from those acquired images, and interactively displayed through a human-machine interface. A clinician can then inspect the volume, by continuously visualizing its cross-section 2D ultrasound images. This is performed by sliding a cross-sectional plane. Any detected lesion can be indicated to the host computer by mouse hand clicking (a prior registration of the 3D volume and the tissue is assumed to be already performed). Then, both the 2D ultrasound probe and the needle guide are positioned in such a way they are aligned on the indicated lesion to biopsy. Subsequently, the needle is automatically inserted trough the tissue to target the lesion, while at the same time being monitored by the clinician that observes the corresponding 2D ultrasound image. Another image volume of the region of interest is taken in order to verify if the

needle has well and truly targeted the lesion, by means of a similar acquisition-construction-visualization process detailed above. Combining the SM modality to the ultrasound one, the system precision is claimed to be increased.

An ultrasound-guided robotically-assisted system for ablative treatment is presented in [11]. The objective is to assist the surgeon for such a medical procedure, by firstly affording a relevant view of the lesion within the soft tissue to facilitate its detection with enhanced precision. Then, it would consist in robotizing the needle insertion for accurate targeting, rather than doing it manually. The setup is composed of a freehand-actuated conventional 2D ultrasound probe, a needle for the insertion actuated by a 5 DOFs robot arm, and a host computer for the monitoring of the application. The 2D ultrasound probe is handled by a clinician and swept to take a 3D scan of the region of interest, by continually acquiring successive 2D ultrasound images. Thanks to a marker attached to the probe, the path followed by this latter along with the recorded images is intra-operatively registered to reconstruct a corresponding 3D ultrasound volume. This volume is then interactively explored and visualized by the clinician for inspection of the region of interest, and thus detection of any possible tumors. The image point position of a detected lesion accompanied with a patient's skin entry point is manually indicated by the clinician, and then transmitted to the host computer. An algorithm was developed for aligning the direction of the needle, in such a way it has to perform a 3D straight line to reach the target tumor location from the skin entry point. The robot then automatically brings the tip of the needle up to the entry point, while in the same time performing the alignment, and finally the needle is inserted to reach the target location. Experiments have been carried out both on a calf liver embedded with an olive for tumor mimicking and on a set of 8 mm of diameter pins immersed in water-filled tank. According to the pin experiments, it is reported that the system performed with an accuracy of about 2.45 mm with 100% of success rate.

Similarly, but with improvements with respect to the manner the successive 2D ultrasound images are acquired then registered, another work is presented in [10]. It is proposed to hold the 2D ultrasound probe by a second robot arm, rather than doing it by free-hand. A scan performed robotically is expected to result in a more better 3D volume image quality, in alignment of the successive slices and in consistency of distances between successive slices, than if it would has been done free-hand. To compare the scan performance whether it is robotically or free-hand performed, experiments have been conducted using a mechanical phantom composed of four pins. An electromagnetic tracker has been attached to each of the ultrasound probe and the needle guide robot tip, for extraction of their respective 3D poses with respect to a base frame. This latter is attached to a tracker located on the operating table. It is claimed that the use of such sensors rather than, for instance, the robot

encoders is more advantageous is the sense that it permits quick configuration of the experimental setup when using more or less robots, and that it simplifies modular replacement of the end-effector. Three robotic scans and three free-hand scans have been conducted on the phantom. It has been concluded that the robotic scan approach outperformed that of free-hand, where besides of obtaining a 3D image of better quality with the former, a rate of 7 success out of 7 trials has been obtained with the robotic scan for a rate of 3 success out of 4 trials with the free-hand scan.

Using 2D ultrasound imaging modality to position instrument tip at desired target location has been considered in [74], where a 2D ultrasound-guided robotically-actuated system is presented. The system consists of two personal computers, a 2D ultrasound probe, an electromagnetic tracking device, and a robot arm. One computer monitors ultrasound image acquisition and processing, whereas the latter computer insures robot control. This control computer conveys the different data, consisting of the target and current control features with corresponding variables of the control servoing scheme, through a serial link running at 155.200 bps. Image acquisition is performed at a rate of 30 frames per second. The electromagnetic tracking device consists of a fixed base transmitter and two remote tracking receivers. Each receiver provides its corresponding 3D space pose with respect to the transmitter base, by transmitting its six degrees of freedom to the computer through a serial line connection. One receiver is mounted on the ultrasound scan head, while the second was initially used for calibration and then is attached to the robot for registration and tracking. The target to be reached by the robot tip consists in the center of an object of interest. It is detected using the 2D ultrasound probe. Firstly, a scan of the region containing the target object is performed by acquiring successive 2D ultrasound images. Then, each acquired image is segmented to extract the corresponding object cross-section. From the set of all those segmented cross-sections, the center of the target object is estimated. The center 3D coordinates represent the target 3D location at which the robot tip has to be positioned. For image segmentation, each 2D ultrasound image is first segmented according to an empirically chosen threshold, then subsampled by 1/4 factor to reduce the computational time of the next step, wherein the image is convolved by a 2D Gaussian kernel of 10 radius and of 5 pixels deviation, and finally an automatic identification of the image section of interest is applied by searching pixels of high intensity. The target is assumed roughly spherical. The robot is servoed in position by a proportional derivative (PD) control law, with an error limit-based rule is added in order to prevent possible velocity excess relative to important displacements orders.

Experiments have been carried out using a tank containing a salt water layer at its bottom and an oil layer at its top. A grape, of approximately 20 mm diameter, served as a roughly

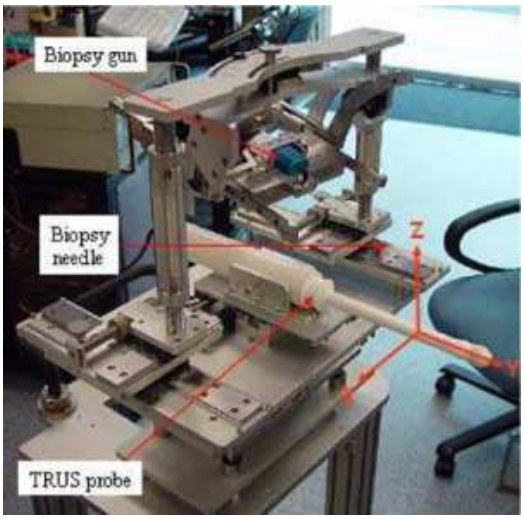


Figure 2.12: A biopsy robot. (Photo: taken from [64]).

spherical target. It was put between these two layers of respectively oil and water. Thanks to gravity and buoyancy forces and the immiscibility between the two liquids, the grape floated within the plane delineating the water surface from the oil one, and can freely slide along this plane. To detect the target location, a scan centered on the grape is performed by taking successive cross-section ultrasound images as described above. In conditions where the grape is maintained fixed, the robot tip touched the target with a rate of 53 out of 60 trials.

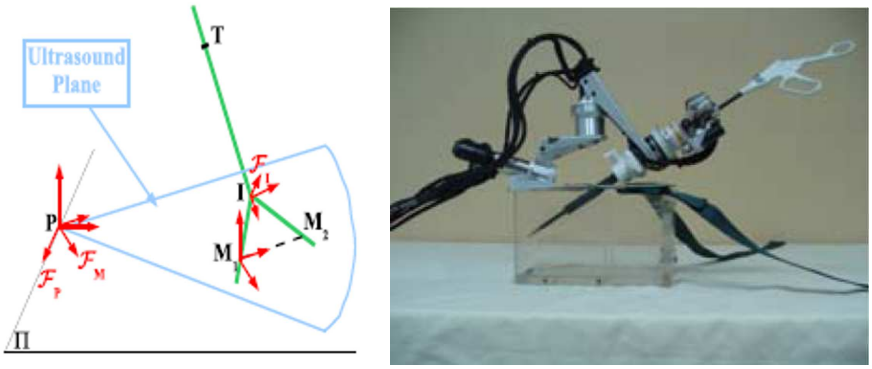
For needle placement in prostate biopsy procedure, a 2D ultrasound-guided robotic system is presented in [64] (see Fig. 2.12). The objective is to perform needle positioning of enhanced accuracy. The system consists of a biopsy needle gun, a robot holder platform, a host computer, and a 2D ultrasound probe. The functions of the computer consist mainly in the monitoring of the procedure. This ranges from ultrasound image acquisition, processing along with registration, screen-displaying for visualization, needle motion planning, and robot motion control. The robot can be moved and thus positioned appropriately near the patient's perineal wall, prior to an intervention, thanks to 4 wheels on which it can translate. It can subsequently be maintained motionless with enhanced stability, after the operator had depressed a foot pedal, which causes the robot to be slightly raised and be supported by 4 rubber-padded legs in place of the wheels. The robot can be further adjusted, by tuning the height and tilt of its operating table. This will allow to position the ultrasound probe horizontally with respect to patient's rectum, in order to obtain as good as possible quality of the ultrasound images, and also to prevent the probe transducer ramming into the rectal

wall during the procedure which could lead to possible damages. The robot table's base is kept in the adjusted pose throughout the procedure, by means of locks. Subsequently, the needle is manually positioned at the skin entry point by adjusting 2 pairs of linear slides and a pair of lead screws. Following, successive transverse ultrasound images of the prostate are acquired at 1 mm distance interval of robot motion, and recorded. They are used to make up a 3D model of the prostate. This is performed semi-automatically, where firstly the urologist have to delineate the prostate's boundary in each of several selected images, among those acquired, by indicating boundary points with hand-clicks. A NURBS (non-uniform rational B-splines) modeling algorithm then processes separately each slice with its assigned set of indicated points, in order to extract the corresponding boundary. The algorithm, finally, fits the successive created edges with a surface simulating that of the prostate. A designed graphical interface allows for the display of the 2D ultrasound images along with the constructed 3D surface of the prostate. The urologist can thus interactively indicate on the interface the biopsy target and needle entry points, by visualizing observed images. These two points are thereby expressed in 3D space with respect to robot frame. The computer then calculates the 3D straight line path, that the needle has to perform to reach the biopsy point from the entry point. This path with the indicated points and the 3D surface are interactively simulated and displayed on the interface. This latter provides, also, a functionality that allows to position the ultrasound probe where an indicated image has been previously acquired and recorded, and thus to verify if the observed image do corresponds to that recorded. This aims to check whether or not the prostate has deformed or shifted. After the robotic system being tested in phantom and cadaveric trials, clinical experiments have been conducted. The patient, under general anesthesia, is lying in lithotomy position on the operating bed. Inside the patient's prostate, copper seeds are dropped. They serve as fiducial targets in order to be able of prospectively assessing the performance of the procedure. Upon a 3D path is planned, the needle is slid down manually along its holder to reach the seed target location. The urologist will then only have to trigger the biopsy fire gun, causing sequential actuation of needle's inner core and outer sheath, and thus a tissue sample is cut off and housed in a slot at the needle's distal end. To verify needle positioning accuracy, the prostate is at the end of the positioning imaged with C-arm fluoroscopy. Over 8 different patients, 17 needle placement procedures have been conducted, where some adjustments were given to the latter trials at the aim of obtaining better results than that of the first trials. To explain the outcome of the first trials, it was hypothesized that the needle bent and strayed from its desired path, due to forces it undergoes while traversing the prostate tissue. An absolute positioning error ranging from 2.3 to 6.5 mm has been obtained. For the second set of the experiments, a thicker needle is employed and is supported by a custom-designed device. This aims at minimizing possible bending of the needle. It was noticed that the positioning accuracy enhanced, where the absolute positioning error dropped to 2.5 mm.

Most of the works, presented so far, that investigated the use of ultrasound imaging, and the 2D ultrasound imaging more particularly, in automatically guiding robotic tasks have however not directly used the observed image in the servoing loop. They instead employed position-based visual servoing, where the image is only used to obtain 3D positions of concerned features. It is well-known that the position-based visual servoing methods suffer from the relatively low accuracy in term of positioning errors. This is due to the fact that the control is performed on estimated locations (usually in the robot working frame). As such, the accuracy of the positioning consequently relies heavily on that of the estimation and that of the robot. In contrast, a control that is performed directly on the observed image, namely image-based visual servoing, would result in more accuracy. The reason is that an image provides homogeneous sensing of the actual features, whose measures the servoing is applied on; the accuracy in this case is of course affected by the image resolution. In the following section, we present works that used the 2D ultrasound images (or part of the information conveyed by the image) directly in the visual servoing scheme.

2.5.4 2D ultrasound-guided image-based visual servoing

The main challenge when dealing with 2D ultrasound images in robot servoing consists in the ability to control the out-of-plane motions. Indeed, as pointed out in Chapter 1, a 2D ultrasound image provides information only in its observation plane and none outside of this latter. This challenge corresponds mainly to a physical and mathematical modeling problem. More particularly, the difficulty consists in the ability to relate the differential changes of the visual features to displacements of the robotic system. Such relation, that is well-known by the term *Interaction Matrix*, is in fact crucial to be able to build an image-based visual servoing scheme [41]. A couple of works considered the interaction with geometrically known surgical instruments. These latter are geometrically represented and their 3D model related. From such models, the interaction matrix is then derived. A more complex modeling problem consists in the case where considering not only manufactured objects (as surgical instruments for example) but also soft tissue objects. The first works in this latter context considered however only the control of probe in-plane motions. Recently, a couple of works dealt with out-of-plane motions control. This latter is mainly the subject of this thesis. In contrast to the existing literature works, we model the exact form of the interaction matrix, and then address the problem of controlling both in-plane and out-of-plane motions.



(a) (b)

Figure 2.13: 2D ultrasound-based instrument guidance. (a) Sketch of the forceps (depicted in green) intersecting the probe observation plane (delineated with blue lines) in two points M_1 and M_2, whose coordinates are used in the servoing scheme - (b) Robot actuating the forceps instrument. (Photos: (a) taken from [79], and (b) from [81]).

Control of the interaction with geometrically-known surgical instruments

A 2D ultrasound-based servoing technique for automatic positioning of a surgical instrument for a beating heart intercardiac surgery procedure is presented in [81, 80, 79]. The instrument consists of surgical laparoscopic forceps actuated by a robot arm [see Fig. 2.13(b)]. An eye-to-hand 2D ultrasound probe is employed for the observation, and thus for providing both the surgeon and the robotic system with real-time images, in order to insure procedure monitoring. It observes both the forceps's pair of pincers and the heart. The objective is to automatically position the forceps instrument in such a way it intersects the ultrasound image plane at desired image positions, that were anteriorly indicated on the image by an operator. The 2D ultrasound cross-section image provides two image points that result from the intersection of the ultrasound planar beam with the forceps [see Fig. 2.13(a)]. These points are fed back to a visual servo scheme, that computes the commands to move the robot accordingly, in order that the observed points converge to the target ones. Previous to that, the points with the corresponding target ones are extracted to be transmitted in four independent features inputs to the servo scheme. Two configurations of the feedback visual features vector are proposed, depending on the choice of the elements forming this vector. In the first configuration, the feedback visual features vector corresponds to the four image coordinates of the two points. In the second configuration, the segment in the image formed by the two image points relates the feedback vector. Two elements of the vector correspond to the two image coordinates of a point lying in that segment, while the remaining two other elements correspond respectively to the segment's length and orientation with

respect to one of the image axes. *In-vivo* experiments have been conducted on a pig heart, where the system performed a task in a reported duration of about 1 min. The proposed technique deals with images of instruments of known geometry, where the forceps's pair of pincers have been respectively modeled with two 3-D straight lines.

In the context described above, where a motionless eye-to-hand 2D ultrasound probe is employed to guide automatic positioning of instrument carried by a robot arm, a Nonlinear Model Predictive Control scheme is proposed in [69]. The objective is to perform automatic positioning of the instrument tip while at the same time to respect some constraints, namely to keep the instrument in the probe observation plane and to take into account the robot mechanical joints limits. The first constraint if not satisfied would yield the instrument getting out of the observation plane, thus leading the feature points vanishing from the image. Since such features are required in the visual control scheme, the robot guidance would fail. As for the second constraint, if not satisfied the robot would get out of its workspace or would reach singularities. The robot thus would be mechanically trapped, and consequently would not be able to move according to the ordered servoing commands; i. e., leading to task failure.

So far, in the present chapter, positioning with respect to observed soft tissues has not yet been introduced. Dealing however with soft tissue ultrasound images in the servoing scheme allows direct interaction and positioning with the observed soft tissue, as can be seen in the following.

Control of the interaction with soft tissues: In-plane motions control

A robotically-assisted system for medical diagnostic ultrasound is presented in [1]. The system consists of a master hand controller, a slave robot manipulator that carries a 2D ultrasound transducer, and a monitoring host computer (see Fig. 2.14). The objective is to automatically assist the ultrasound clinician when performing the diagnostic. While the ultrasound transducer is remotely moved by the clinician through the master hand, the robotic system can automatically compensate for the unwanted motions in such a way the transducer keeps a certain view configuration with respect to the patient body. This is afforded by a servo scheme paradigm wherein the operator's motion commands and a visual servoing controller share the control of the robot holder motion. The primary envisaged use for the system is carotid artery examination. The task then consists in automatically keeping the center of one, or more, artery in the middle of the ultrasound image, while at the same time the transducer is being teleoperated over the patient's neck by the remote clinician.

Figure 2.14: Robotic system for medical diagnostic ultrasound [1]. (Photo: S. E. Salcudean's research group web page http://www.ece.ubc.ca).

The artery is thus kept in the middle of the image thanks to the visual servoing scheme, which automatically controls 3 DOFs of the robot holder in the probe observation plane. It controls the two translations along the image's two axes and the rotation around the axis orthogonal to the image plane, respectively. The remaining DOFs are being operated by the clinician through the master hand. The visual servoing is fed back with the center coordinates of each of the artery in the ultrasound image. Before this, image processing is applied on each of the acquired 2D ultrasound image to detect and track the boundary of each artery. The image coordinates of points lying on a boundary are used to compute the corresponding center coordinates in the image. Five detection and tracking techniques have been tested and compared. These techniques consists in the Cross Correlation algorithm [67], the Sequential Similarity Detection (SSD) algorithm [13], the Star algorithm [33], the Star-Kalman algorithm inspired from [5], and the Discrete Snake Model algorithm modified from [20]. They have been tested on successive 2D ultrasound images, captured at a rate of 30 frames/sec from an ultrasound phantom. In this latter, three plastic tubes are positioned along three different axes. During the acquisition, in-plane motions are performed by moving back-and-forth the ultrasound transducer along one axis of the image plane, with constant absolute velocity. According to the obtained results, the Star-Kalman and the SSD algorithms outperformed the other techniques, where the former algorithm showed to be more advantageous with less computational time. That conclusion is, however, inferred from trials where the image variations are due to motions of the transducer only along its image axis. Therefore, only plane motions have been performed, and consequently motions in the transducer's out-of-plane have not been considered. Indeed, out-of-plane motions (e. g., motions along the axis orthogonal to the image plane) lead to deformations in the ul-

trasound image itself (e. g., boundary shrinking/stretching) rather than it is simply shifted, as in the presented case. Thus, the techniques that better performed for motions within the image plane might, perhaps, present drawbacks or completely not apply in out-of-plane motions case, and vice versa.

The system was tested experimentally, where two features that represent the visual feedback correspond to the center coordinates of two pipes of the phantom. The system operated at a rate up to 30 Hz. Two main applications of the system have been considered. The first concerns a 3-D ultrasound imaging system, that can be used to make up a 3D image of a scanned region of interest (the artery in this case) from successive 2D centered ultrasound images acquired during the scan's sweep. That sweep is monitored by the visual servoing controller in such a way the artery remains centered in the image. Inputting those captured images whether to a Stradx tool [35] or to a Star-Kalman based reconstruction, a 3D image is outputted. The latter reconstruction algorithm showed to be more advantageous with shortened computational time, since only the coordinates of the contour points extracted from each acquired image are stored, rather than the full image when using the Stradx tool. As for the second application of the robotic system, it concerns tele-ultrasound exam. A clinician is located at a remote place, and can from there supervise the procedure which takes place in a different location. The clinician can visualize the procedure development thanks to the display, on different screens, of images about the operation room. These images are respectively provided by two observing cameras and the ultrasound transducer, both located in the operating room where the patient is under diagnostic assisted by the robot. By handling the master device, the technician's commands are sent to the carrying robot. Data transmission between the two sites is performed through an Internet connection.

A 2D ultrasound-guided robot for percutaneous needle placement for cholecystostomy treatment is presented in [38]. The robot possesses 2 active DOFs used for automatic needle insertion (see Fig. 2.15). The intraoperative 2D ultrasound images, of the gallbladder along with the needle, are directly used in a visual servo scheme that computes the control commands. The robot will thus position the needle accordingly, while in the same time compensating for possible target shifting. The latter can occur due, as instance, to patient's heart beating, breathing, or pain that could rise due to local anesthesia. Prior to insertion, the needle is mechanically constrained to lay in the same plane of that of the ultrasound beam. That configuration is kept throughout the procedure. This is achieved using 5 passive DOFs that the robot also possesses. Those DOFs furthermore allow to position the needle at the skin entry point right prior the insertion. The gallbladder is detected in the image using a motion optimized active contour model, while the needle direction is extracted using

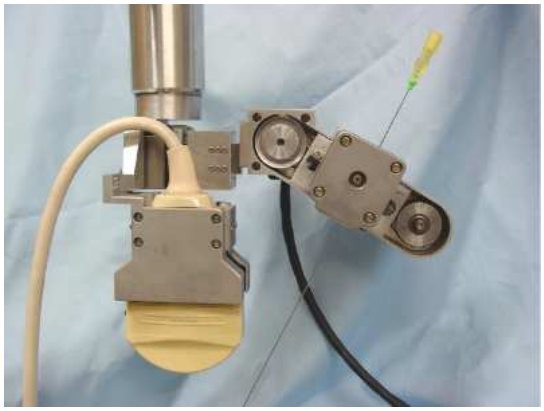
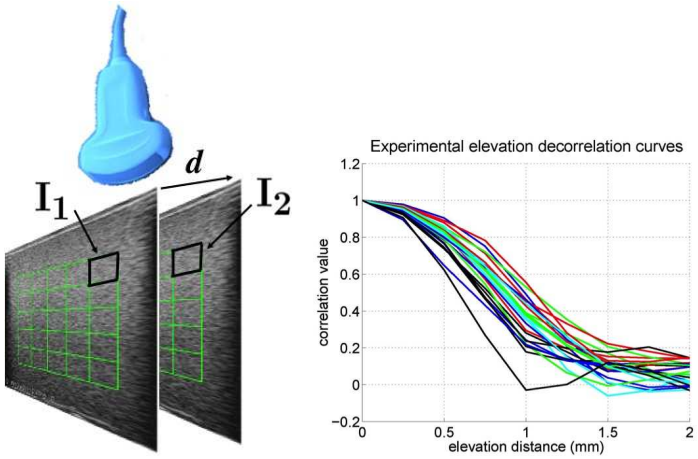


Figure 2.15: 2 DOFs robot for 2D ultrasound-based needle insertion. (Photo: taken from [38]).

the Hough transform [39]. The system performance was assessed through phantom experiments, video simulations, and animal experiments. The robotic system operated at a rate of about 3 Hz, at which the needle path planning is updated. It performed with gallbladder recognition error of less than 1.5 mm under ordinary breathing conditions, and with needle positioning error of about 2 mm in animal trials.

Control of the interaction with soft tissues: Both in-plane and out-of-plane motions control

An ultrasound visual servoing technique using the 2D ultrasound modality for soft tissue motion robotized tracking and stabilization is presented in [46]. It makes use of speckle information, contained in the B-scan images, in separately controlling the probe in-plane and out-of-plane motions in order to maintain the probe observation plane on a target B-scan ultrasound image. Although ultrasound speckle was considered in different works as noise to reduce, it is in fact not a random noise but coherent reflections of small cells contained in soft tissue. The B-scan observation plane is in reality of millimeter order thick and, as consequence, successive acquired B-scan images overlap in space thus resulting in correlation of speckle between each of them (see Fig. 2.16). Speckle information have been used to estimate multi-dimensional flow of 2D ultrasound images [12], and its correlation used for sensorless estimation of the 3D pose of freehand 2D ultrasound probes, as in [34]. In the latter work, the speckle correlation is approximated by an exponential function based on image intensity, in order to estimate the displacement between two plane of successive acquired B-scan images [see Fig. 2.16(b)].



(a) (b)

Figure 2.16: Speckle correlation between two successive B-scan image planes acquired by a 2D ultrasound probe (displayed in blue) [46]. (a) Two successive B-scan images, whose respective planes are spaced by a distance d, and where two corresponding patches I_1 and I_2 are shown on their respective grids (displayed in green) - (b) Correlation curves between the two B-scan planes considered for 25 patches. The curves are function of the distance d between the two planes. (These two figures have been kindly provided by Alexandre Krupa).

That principle is exploited in [46] to estimate the B-scan probe out-of-plane motions, that would bring the probe to its target plane from its current one. The objective is in fact to estimate the target image plane with respect to the observed one. The out-of-plane motions are related to translations along the axis orthogonal to the probe observation plane (image) and rotations around the two image's axes. To estimate those movements, different patches are attached to the ultrasound image, which are discriminated according to their respective allocated pixel coordinates. For each patch of the observed (current) image, its distance from its corresponding patch belonging in the target image is computed according to the decorrelation technique introduced above using their respective intensity information. Note that the target image has been previously saved as a pixel intensity array. Prior that the exponential function is applied, the intensity of the B-scan image is decompressed to be expressed on a linear scale [72]. This is performed since the outputted B-scan image's intensity is compressed according to a logarithmic scale whereas the original raw radio-frequency signal (RF) provided by the transducer is expressed on a linear scale. The estimated distances are used to geometrically represent the target image pose with respect to the observed one, by defining for each distance a 3D position with respect to a frame attached to the observed image. The patches of the target image are then fitted with a plane, defined by its normal

vector and its distance from the observed image's one. This plane is nothing but an estimation of the target image's one.

The four elements of the target plane, that are the three components of its normal vector and its distance from the current image's plane, are fed back to a 3D visual servoing scheme that then computes the velocity command [51] for the out-of-plane motions of the ultrasound probe. The in-plane motions are however separately controlled by a different 2D visual servoing scheme. The latter is fed back with a visual feature vector of three components. This vector relates the rigid in-plane motion from the plane of the observed image to that of the target. Two of its elements correspond to two translations respectively along each of the image's two axes, while the latter element corresponds to a rotation around the image's orthogonal axis (elevation axis). These three elements represent the differences of respectively the displacement and the rotation from the observed image to the target one. They are extracted using the image tracking technique [37]. It consists in minimizing an intensity function which is based on a motion model.

The approach has been tested in both simulations and experiments. The simulations consist in a scenario where a virtual 2D ultrasound probe interacts with a realistic ultrasound volume, made up from a set of parallel real 2D ultrasound images. The latter have been, at a previous time, successively captured at an equivalent distance interval during motions of a 2D ultrasound probe along its orthogonal axis. The motions have been performed with constant velocity. The experiments have been conducted using two different setups. In the first one, a 2D ultrasound probe was carried by a 2 DOFs robot, that provides two translations respectively along the horizontal probe axis and along the axis orthogonal to the probe. The second setup consists of a 6 DOFs medical robot carrying a 2D ultrasound probe. In both the simulations and the experiments, the robotic task was focused in tracking a target ultrasound B-scan image, since the proposed approach is devoted for tracking by allowing only slight displacements from the target image. This limitation however has been alleviated, where it is proposed to register up to a certain width the continually acquired images. This would allow to recover the path followed by the probe, by stacking the different displacements between each successive images. The approach is devoted solely for B-scan images, and requires a calibration step through which parameters involved in the correlation exponential function are estimated. In fact, those parameters vary depending on the imaged soft tissue. Note that the approach heavily relies on the estimated target plane, on which the probe's plane has to be automatically positioned. Consequently, any estimation errors will be, undoubtedly, reflected in positioning errors of the ultrasound probe leading to drifts from the actual target.

In the Lagadic group, IRISA/INRIA Rennes, wherein this PhD work has been conducted, a former preliminary work [4] dealt with the control of both in-plane and out-of-plane motions of a 2D ultrasound probe interacting with egg-shaped objects. The objective was to automatically position the probe with respect to such an object. It was attempted for use in a context where a robot arm actuates the probe, which continually provides 2D cross-section images of the observed object. These images are fed back to a visual servoing scheme, that subsequently computes the command velocity. The robot will then have to position the probe transducer, by moving according to the ordered velocity, in such a way that at the convergence the observed cross-section image corresponds to desired one. The probe observation plane intersects the object of interest, which results in a cross-section ultrasound image. Assuming the soft tissue being egg-shaped, the contour of the cross-section is fitted with a third order polynomial, whose coefficients are used as the feed-back visual features. The method has been tested is simulation, where the scenario consists of a mathematically modeled virtual 2D ultrasound probe and an egg-shaped object. Their respective poses (position and orientation) are assumed known with respect to a base frame. Those mathematical models of the object and the probe are used to simulate their interaction, and thus providing the contour of the cross-section image. The contour is characterized with a set of its points coordinates. The proposed approach, however, is dedicated to soft tissue with known geometry, namely egg shaped objects. It relies, moreover, on visual features that have no physical signification and are not robust to image noise. Extracting these features from the image can, sometimes, become challenging, and is prone to failures. This consequently can threaten the system stability. In robotics, in general, and in medical robotics, more particularly, the robustness is an important trait that has to be addressed, especially when dealing with the ultrasound modality, that inherently provides very noisy images. The work we present in this dissertation exploits instead visual information that are robust to image noise. Such information can moreover be readily extracted after the image would have been segmented. These features we select to feedback the visual servoing scheme consists in combination of *image moments*; the latter are presented in Chapter 3. Moreover, we develop the exact form of the interaction matrix related to these features. The formulae we develop is general in the sense that it can be applied to different shapes, say to whatever considered closed volumes. We in fact developed new theoretical foundations that yield us able to derive such a matrix. The corresponding modeling is presented also in Chapter 3. Another main contribution brought though this thesis is that we propose an efficient estimation method that endow the robotic system with the capability of interacting with objects without any prior knowledge of their shape, 3D parameters, nor location in the 3D space. This is presented in Chapter 4. Only the image, along with robot odometry, is used to compute the control law, as presented in Chapter 5.

2.6 Conclusion

We have provided through this chapter an overview about image-based medical robotic systems, and more particularly about ultrasound-guided ones. We started by giving a short introduction to robotics control and medical robotics. Different paradigms reflecting the manner a medical robot is commanded have been presented. We recall that this thesis is concerned with the self-guided paradigm, where the robot completely autonomously interacts with its environment thanks to closed-loop servoing techniques developed and presented in this document. The intervention of the operator only consists in indicating to the system the objectives of a required task, right prior that the robot is launched to perform the procedure.

It was highlighted that usually medical robotic systems use mainly visual sensing for monitoring the interaction with their respective environment. Examples of most investigated imaging modalities for guiding medical robotic systems have been introduced with some examples for illustration. These modalities range from, but are not limited to, optical, MRI, X-ray or CT, and ultrasound. They provide with valuable sensing allowing for interaction monitoring. Each of them provides, indeed, with particular information about its environment, that could be greatly relevant for certain range of medical applications. Optical imaging systems, as instance, provide images of open-space fields, and therefore find their use in minimally invasive surgery. They are however restrained to some applications like endoscopic surgery robotics, and recently in microsurgery robotics. This is due to the fact that they can not provide internal anatomical views, unless they are inserted inside the patient's body. This latter resolution is however not appropriate for many kinds of applications, because of the possible trauma and hemorrhage that could result, and since some body's regions are not readily accessible and viewed. Yet, internal images are in most of the cases required in medical robotics since medical robots are usually interacting with body's parts that are not naked-eye viewed. In contrast to optical imaging, MRI, CT, and ultrasound provide internal anatomical images without any dissection. X-ray, or CT, showed however to be invasive and harmful for the patient body. As for the MRI modality, even if it is considered noninvasive, the images are not provided at a sufficient rate to envisage real-time robotic applications. Concerning the ultrasound modality, it is considered thanks to its noninvasiveness as not harmful to the patient body, and more particularly 2D ultrasound modality can provide images with a relatively high streaming rate.

We introduced in this chapter works related to automatic guidance using ultrasound images, and classified them according to different classes. It is perhaps useful to summarize them. We distinguished: ultrasound-based simulation; 3D ultrasound-guided robotics; 2D ultrasound-based position-based visual servoing; and 2D ultrasound-based image-based vi-

sual servoing. Also, within the last class we distinguished the following categories: positioning of surgical instruments; positioning with respect to observed soft tissue where only probe in-plane motions are controlled; and positioning with respect to observed soft tissue where both probe in-plane and out-of-plane motions are controlled. This thesis falls within the latter category.

Chapter 3

Modeling

Building a visual servoing scheme requires the modeling of the interaction matrix that relates the time variation of the feedback visual features to the motions of the robot. Such interaction matrix is in fact crucial for computing the control law. In case of optical systems, like a perspective camera carried by a robot arm for example, the interaction matrix is generally already available thanks to the amount of works that have considered such a sensor (e. g., see [41] and [17, 18]). It is however not the case for robotic systems using 2D ultrasound imaging modality as source of visual information. This thesis concerns automatic guidance of a general robot arm from observed 2D ultrasound images. These images are provided by a 2D ultrasound probe carried at the robot end-effector. We need therefore to model the interaction matrix for the case of 2D ultrasound in order to allow the robot automatic interaction with its environment. One of the challenging issue, however, concerns the fact that a 2D ultrasound probe interacts with its environment by such a manner that was, so far, difficult to model. This is addressed in the present chapter. Firstly, we used the concept of image moments to construct the feedback visual features. This concept seems completely relevant when dealing with the ultrasound modality, as discussed in Section 3.1. Then, we propose new theoretical foundations that allow us to model the analytical form of the image point velocity as function of the robot velocity. This fundamental modeling is subsequently used to obtain the exact analytical form of the interaction matrix that relates the image moments time variations as function of the probe velocity. The modeling method we propose can be applied for general-shaped objects. We theoretically test and thus validate this general result on some simple shapes like spheres, cylinders, and 3-D straight line-shaped wires.

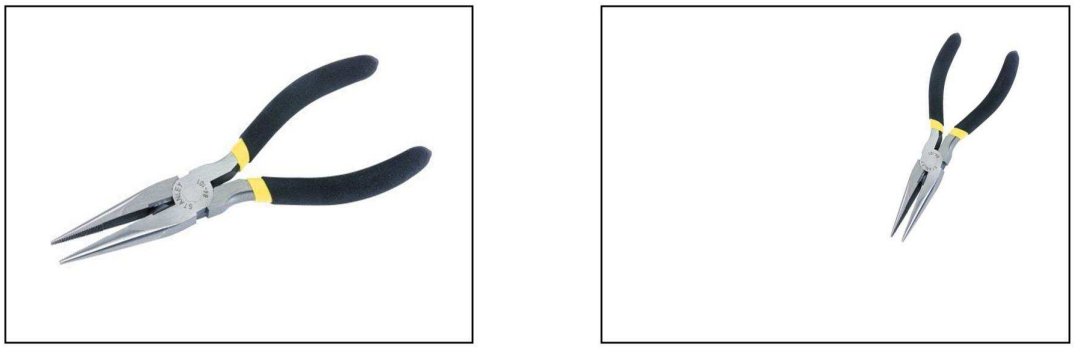
3.1 Image moments: a brief state-of-the-art

Image moments are mathematical entities whose inferred values can describe the configuration of sections in the image; by "configuration" we mean also the section's shape, though implicitly . Such configurations are mainly correlated to the section's geometry. They could be as instance the location, orientation, contrast, or size of a section in the image. After their original version being introduced in the field of mathematics, moments, or image moments as currently referred when dealing with images, were finally considered for pattern recognition field. Based on the theory of algebraic invariants, functions of moments that are insensitive to particular section's changes, such as translation and rotation in the image, and size are presented in [40]. Such functions are indeed of great interest for pattern recognition applications. The fact that image moments can describe section's configuration, they can therefore be used to discriminate between the different sections. Each section could be assigned with a particular value, more particularly numerical value. However, if a considered section is subject to configuration changes such those mentioned above (translation and rotation of the section in the image), its assigned value likely would vary. In the case such changes occur, the considered section can no longer be assigned with a certain value, and consequently can not be discriminated and thus recognized. But image moments that are invariants to such changes would keep their initial value, and consequently they can be used as pointer for a considered image section, still under the mentioned changes. This is illustrated in Fig. 3.1. Such image moments are called *moment invariants* [40]. We will see in chapter 5 that such invariance properties are of great interest for the selection of the feedback visual features, since these latter we propose are based on image moments.

Consider a section lying in a plane Π, that is defined by an orthogonal frame (u, v) (see Fig. 3.2). The two-dimensional moments of a density distribution function $\rho(x, y)$ and of $(i + j)$th order related to this section are defined in terms of the surface integral by [40]:

$$m_{ij} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^i \, y^j \, \rho(x, y) \, dx \, dy \tag{3.1}$$

where (x, y) represent the 2-D coordinates of a point $\mathbf{P}$ lying in the section. We can therefore note that moments are strongly correlated to the shape of the section, as can be deduced from the term product $x^i \, y^j$. In the case Π represents a plane of a 2D image, then we will refer to 2D image moments. The couple (x, y) will then represent the pixel coordinates of point $\mathbf{P}$. The function $\rho(x, y)$ could be related to the pixel intensity of the image, its color, or else.



(a) (b)

Figure 3.1: Illustration of moment invariants with two different images of a same pair of pliers. (a) Initial image whose boarders are delineated with a rectangle - (b) Final image of the pliers after configuration changes in the position, rotation, and scale. Ordinary image moments values of respectively the images (a) and (b) are different, whereas those of the invariant moments to position, rotation, and scale are the same. The latter values can thus be assigned to the pliers for prospective identification.

Successive research works have then followed by applying image moment invariants for pattern recognition applications. We can cite as example, the use of moment invariants for automatic recognition of aircraft shapes and types from images, as in [25] [8]. They have also been used for pose estimation of planar objects [59]. Along with the widespread of moments invariants for wide range of applications, theoretical studies with objectives of making these functions more powerful have also been reported. Moment invariants to image contrast changes that at the same time keep their initial insensitivity to image translation, rotation, and scale are presented in [50]. Objects which present symmetries might be difficult to identify with moments, since the latter could tend to vanish (i. e., to be null) as more as the symmetry appears in the image while the object configuration changes. To deal with such limitations, moment invariants to image translation, rotation, and scale, but dedicated for detecting objects that present N-fold rotation symmetry are presented in [32]. Another formulation of image moments quite inspired from the existing version is presented in [19]. It consists in defining a new version of image moments as function of only image coordinates of the points lying on the boundary of the considered section, instead of those of the whole points lying within the section image (such version's moments do not correspond to those of the old version when expressed on the contour using the Green's theorem). Such formulation is aimed at decreasing the computational time of image moments by considering fewer number of points involved in the computation than if the whole points of the

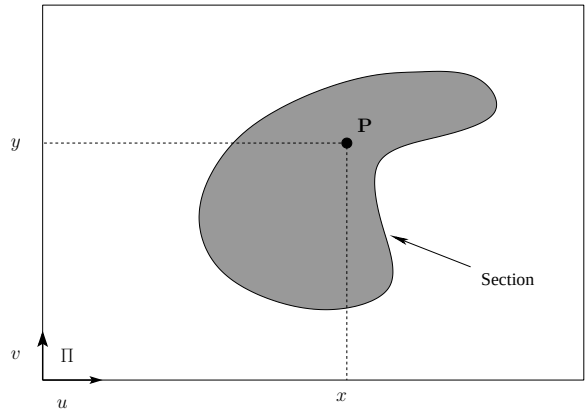


Figure 3.2: Section lying on a plane Π.

section are considered. This version's moments have also been made invariant to image translation, rotation, and scale changes. Adjusting the original formulation of moment invariants, introduced in [40], and generalizing them to the case of n-dimensional moment invariants through a generalized fundamental theorem is presented in [52]. There are also other interesting works about image moments devoted for the fields of pattern recognition and computer vision, but we settle for those cited above and in Section. 3.2. A survey about image moments is available in [66]. The objective of the present section concerns the introduction of image moments and the illustration of their usefulness, since the visual techniques we propose through this thesis exploit these information. Dealing with image moments for pattern recognition or computer vision is not the objective of the present thesis and is beyond its scope.

One of the trait of image moments is that they can generically represent an image section, without prior knowledge about this latter. Image moments can be readily computed from a segmented image. These features are relatively robust to image noise, compared for example to features composed of coordinates of points. Another but typical trait to image moments is that they do not require matching of points in the image but only a global segmentation. We will see that this trait is of tremendous interest for visual servoing based on 2D ultrasound. Indeed, this trait matches one of the key solutions that could enable addressing the modeling issue of 2D ultrasound. This will be recalled at an appropriate step of the modeling technique, in Section 3.5.2 more precisely. Moment invariants, more particularly, provide with information about the configuration of the section, with respect to the image, in a decoupled way. This latter property makes image moments relatively amenable in order to build independent visual features and thus to develop partially, or perhaps totally, decoupled visual servoing schemes in a natural way. These features are intuitive with geometrical meaning, where their low order completely and directly relay in-

formation about the size, the center of gravity, and the orientation of section in the image. Therefore, a set of features based on image moments can be directly related to a pose of a 3-D scene in the space (the case of planar scene being pointed out and reported in [50]).

All these traits make image moments a potential entity for deriving relevant visual features, that can be used as feedback in visual servoing schemes. Such features can endow the robotic system with the capability of automatically reaching configurations from which the robot can provide desired images of the scene, and thus positioning with respect to the latter. Such systems do not deal with coordinates, or the like, but directly with shapes of objects. However, the key solution related to the development of visual servoing schemes is the jacobian matrix that relates the differential variations of the selected set of visual features to the differential changes of the configuration of the robotic system [27] [41]. Such jacobian is well known by the term *interaction matrix* when the velocity space considered is $SE3$. It is, indeed, in most of the cases, challenging to model and obtain such matrix, especially when this concerns its analytical form.

We provide in what follows a brief state of the art about works that investigated the modeling of such matrix, in case of optical systems, in systems using camera as source of visual information more particularly. We emphasize that the modeling in case of optical systems quite differs from that of 2D ultrasound, as has been shown in the previous chapter. In this thesis we model and thus provide the exact analytical form of the interaction matrix that relates the differential changes of the image moments to the differential changes in the configuration of a general 6 DOFs robot arm that carries a 2D ultrasound probe at its end-effector.

3.2 Discussion with regards to image moments

Similarly to different other pattern recognition and computer vision features, image moments present also some inherent drawbacks. Moment invariants might suffer, in some configurations, from information suppression, loss, and redundancy [2], and from occlusion. The suppression effect is related to the case where the information of the section's central area are rejected. The information loss is related to the case where the information relayed by the image higher order harmonics are filtered. As for redundancy, it occurs when a selected set of moments-based features represents different sections.

The suppression effect can be noticed from the relationship (3.1). Since in this thesis we deal with moment invariants, we can directly consider for clarity that the section is centered in the image. We can first remark that the image moments function is strongly correlated to the section shape, and this is ensured through the image pixel coordinates x and y. More

the value of x is high more is that of x^i. The same applies for y. We can subsequently remark that more the point is far from the center more higher is the value x or y. The farthest points correspond, in a general sense, to those lying on the section's boundary. The relationship (3.1) mainly contains products of the power of x with that of y. Therefore, the points lying more farther from the section center have higher values and thus have more effect than those closer to the center. Consequently, the farther points have more weight on image moments's inferred value than those closer. This effect is more felt when moments of higher order are employed. Indeed, higher is the order larger is the difference between the values $x^i\, y^j$ of respectively the farther points and those closer. The suppression effect can be embodied in a concrete sense. If, for example, the image intensity function is introduced in the definition of image moments relationship, and if, as instance, the central area have large intensity information, this latter would partly, or almost totally, suppressed due to the effect of the farthest area, as we just described; this information would be swallowed up by that conveyed by the farthest regions. If, however, the section has no valuable intensity information in its center-closer area, there would be no information suppression that could be caused by the above discussed effect of moment invariants. Nevertheless, this drawback presented above is precluded in our case (in the servoing system we propose, more precisely). Indeed, only the shape of the section in the image is exploited in the control law, i. e., the image is first segmented and binarized. Thanks to the formulation we use in this thesis in the definition of image moments, these latter are no longer affected by information suppression. More precisely, we exploit solely the geometric shape of the section in the image. We do not consider, for example, image contrast information in the definition of moments. The geometry of the section is mainly represented by the boundary forms of the latter. There is therefore no information in the central part of the section that could relay valuable information with regards to the section shape. Owing to what has been discussed above, the section geometry information are consequently not subject to suppression effect since the valuable information are present in the farther points not in the closer to the section center. We exploit, to summarize, only the boundary of the section.

High-order moments are also well-known to be vulnerable to image noise, as can be clearly noticed from the fact that a moment's order corresponds to the power at which the coordinates are elevated. However, the visual information we present in this dissertation employ only up to the third order moments.

As for information loss, it is related to section's parts whose boundary presents curvatures with high frequency. This is illustrated by Fig. 3.3, on which we can distinguish the part that possesses high harmonics. Such part is vulnerable to the information loss effect. That is, the information relayed by such part would be filtered by moment invariants functions and thus lost. However, usual considered objects could unlikely possess parts with a certain level

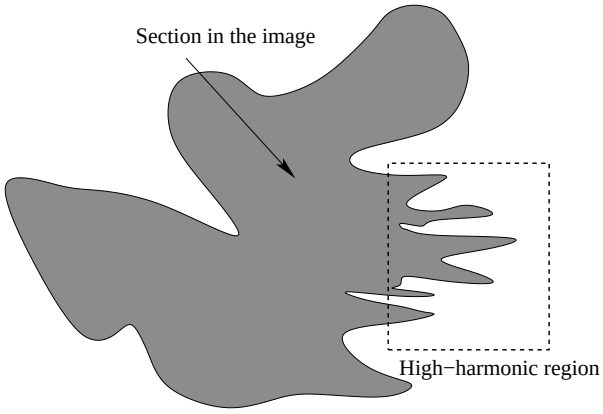


Figure 3.3: Image section subject to the effect of information loss inherent to image moments (typical example, grossly sketched). The section's part, roughly enclosed by a dashed rectangle, presents high-harmonic curvatures.

of high-harmonics susceptible to yield them vulnerable to such effect. Moreover, we employ only up to the third order moments, and thus we clearly do not deal with high harmonics and therefore we do not consider them in the control law. Note that the information loss is somewhat related, say similar, although inversely, to the characteristic of vulnerability to image noise.

Finally, the occlusion effect represents the case when a part, or whole, of the concerned section disappears from the image. The image moments represent mainly the shape of the observed section in the image. When an occlusion occurs, it is clear that the section would be warped in the image, since at least part of it would vanish from the image. In that case, the image moments values would consequently change since they represent another shape different from the original one. Therefore, the image moments initial values, representing the section in its whole form, could no longer represent that section. Nevertheless, we assume that the whole section can be imaged and no part of it would disappear. This assumption shows to be consistent since a 2D ultrasound probe can provide in-depth information, and thus the concerned section could be imaged even though there are other soft tissues lying between it and the probe transducer. Concerning the redundancy effect, we prefer to discuss it in Chapter 5, considering that doing so is more appropriate.

The above cited drawbacks could, in some cases, become favorable [2]; although these drawbacks are precluded in our case, as has been described. If the central part of the image section possesses highly-noisy information, the suppression effect would reject the carried noise. Similarly, the effect of information loss will filter and thus reject the noise carried by high-harmonics parts.

3.3 Image moments-based visual servoing with optical systems: state of the art

One important and crucial step in the modeling of the interaction matrix is already acquired when dealing with optical imaging robotic systems, whatever the kind of the visual features used as feedback information in the servoing scheme. Indeed, the jacobian matrix that relates the differential changes of the image points coordinates with respect to the variation of the configuration of the robotic system is, in most of the cases, say in all, available. After obtaining a second jacobian that this time relates the differential changes of the visual features to the differential changes of those image points coordinates, it would be easy to derive the global interaction matrix that relates the visual features to the robot configuration. If, as an example, the considered visual features are coordinates of the points in the image, the second jacobian matrix is nothing but the identity matrix.

This can be formulated and thus illustrated by the following relationships. Let vectors $\mathbf{s}$ and $\mathbf{x}$ be respectively the set of the visual features and the set of points's image coordinates, and let vector $\mathbf{q}$ be the configuration of a robotic system, whatever the imaging modality used as source of visual information. The jacobian matrix $\mathbf{L_x}$ relates the differential changes $\dot{\mathbf{x}}$ of $\mathbf{x}$ to the differential changes $\dot{\mathbf{q}}$ of $\mathbf{q}$ by: $\dot{\mathbf{x}} = \mathbf{L_x}\,\dot{\mathbf{q}}$. Such matrix is indeed available for the case of optical systems. The differential changes $\dot{\mathbf{s}}$ of $\mathbf{s}$ can be written as:

$$\begin{aligned} \dot{\mathbf{s}} &= \tfrac{\partial \mathbf{s}}{\partial \mathbf{x}}\,\dot{\mathbf{x}} \\ &= \tfrac{\partial \mathbf{s}}{\partial \mathbf{x}}\,\mathbf{L_x}\,\dot{\mathbf{q}} \;=\; \mathbf{L_s}\,\dot{\mathbf{q}} \end{aligned} \tag{3.2}$$

The entity $\mathbf{L_s} = \frac{\partial \mathbf{s}}{\partial \mathbf{x}}\,\mathbf{L_x}$ is the global jacobian matrix that relates $\mathbf{s}$ to $\mathbf{q}$, where $\frac{\partial \mathbf{s}}{\partial \mathbf{x}}$ represents the second jacobian matrix that relates $\mathbf{s}$ to $\mathbf{x}$. This latter matrix is the one equal to identity if the visual features are the coordinates of the points in the image (i. e., $\mathbf{s} = \mathbf{x}$). Therefore, in case using optical systems, modeling the interaction matrix generally comes to only obtain the matrix $\frac{\partial \mathbf{s}}{\partial \mathbf{x}}$, since the jacobian matrix $\mathbf{L_x}$ is in most of the cases already available.

A first work toward modeling the interaction matrix relating the image moments time variation in the case of perspective camera was attempted in [9]. Coarse approximations have however been assumed for that. A visual features vector composed of the area (size), the gravity center, and the orientation of the section in the image was considered to automatically control only 4 DOFs of a robot arm. Another work [83] used neural network to estimate

the interaction matrix. Finally, an exact form of such matrix was obtained, provided that the observed object 3-D model with respect the camera is well-known [15, 16]. Six visual features corresponding respectively to the area, the coordinates of the center of gravity, the orientation, and two other third order moments of the section in the image were selected to control the 6 DOFs of the robot. The corresponding visual servoing scheme has been validated from both simulations and experiments using planar objects. A combination of image moments yielding the visual servoing partially decoupled is presented in [76]. Six visual features have been proposed to control the 6 DOFs of a robot arm holding the camera. The method was first developed for the case a planar object is parallel to the image plane of a perspective camera and, then, generalized to the case where the planar object is not parallel. The commands generated and then sent by such decoupled visual servoing scheme allow the robot performing appropriate 3-D trajectories. The proposed visual servoing scheme is devoted to images wherein the section is represented either by continuous contours or by discrete points. It has been validated from both simulations and experiments where, once again, the observed objects are planar. Another advantage of obtaining a partially decoupled (or totally decoupled at the best) servoing schemes is that the computational time required to compute the pseudo inverse (or the inverse) of the interaction matrix can be shortened; even though this advantage could be considered at a relatively fewer interest. This can be ensured thanks to the properties related to sparse matrices, as the developed one in [76]. Note that dealing with the interaction matrix in terms of decoupling is equivalent as dealing with the decoupling of the control scheme, since this latter uses mainly the interaction matrix to compute the commands to the robot.

As has been discussed in Chapter 1 (in Section Contributions, more precisely), the modeling in case of optical systems quite differs from that of 2D ultrasound; the latter consists the field this thesis is addressing. The interaction matrix developed for optical systems, which has been introduced hereinbefore, does not apply in the present case. Even worse, the elemental jacobian matrix that relates the image points variation to the sensor velocity is not available in this case; we recall that we refer to the matrix $\mathbf{L_x}$ introduced by (3.2) (in case of optical systems, the jacobian is however generally available). Yet, such jacobian is crucial and required in order to develop the interaction matrix and thus to derive the visual servoing scheme. Through the works presented in this dissertation, we have finally been able to model such jacobian and then to obtain the interaction matrix that relates the image moments variations to the configuration of a general robot arm (and thus to the robotized sensor velocity) holding the 2D ultrasound probe, and thereby to derive a corresponding visual servoing scheme. The interaction matrix form we provide is analytical. This chapter provide theoretical foundations, which, based on it visual servoing methods can be derived. This is thoroughly presented in what follows.

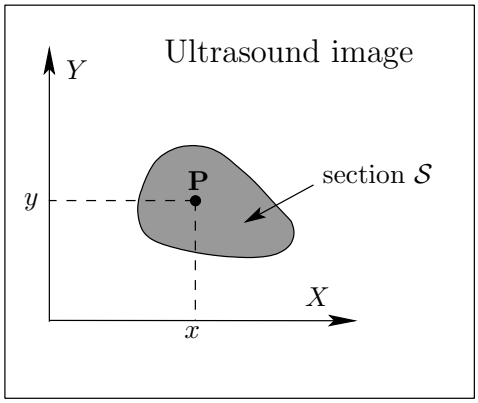


Figure 3.4: Representation of an ultrasound image. The image's boundary is represented by the outer rectangle, where $(X,\ Y)$ represent the 2D orthogonal frame attached to the image.

3.4 Modeling objectives

The scenario consists of a 2D ultrasound probe transducer actuated by a general 6 DOFs robot arm. This robot and thus the transducer are interacting with a soft tissue object. In a continuous streaming, 2D ultrasound images of the observed object are provided by the 2D ultrasound transducer. The robotic task consists in automatically positioning the transducer with respect to the object, using the observed ultrasound images. These latter in fact have to automatically guide the robot and thus monitor its motions in such a way the transducer, carried by the robot, automatically reach and stabilize at a desired configuration with respect to the object. Automatically achieving such task necessitates the development of a visual servoing technique which, at its turn, requires appropriate visual features to feed back the robot system and thus correct its motions. We recall that we propose to exploit image moments information, along with its derivative form that are the famous moment invariants. This has already been described in Section 3.1.

The control system paradigm with which we are concerned consists in servoing the robot with velocity commands. It could be considered as a relatively high-level control compared, for example, to torque control. The system we propose can, nevertheless, be connected to the robot low-level in order to envisage torque control. The robot is assumed, of course, that it already possesses its own low-level control system enabling it to move according to the ordered command velocity. The visual servoing scheme, thus, computes velocity commands which accordingly the robot will move. The modeling objective becomes, therefore,

to firstly provide the jacobian matrix $\mathbf{L_x}$, involved in (3.2), that relates the image points time variation to the sensor velocity and thus to the robot velocity. Then, we use $\mathbf{L_x}$ to model the interaction matrix $\mathbf{L_s}$ that relates the image moments time variation to the sensor velocity.

More precisely, let the $(i+j)$th order image moment m_{ij}, previously introduced by (3.1), now relate solely the shape of a section $\mathcal{S}$ in the image. This entity is thus defined by this double integration:

$$m_{ij} = \int\int_{\mathcal{S}} f(x,y)\, dx\, dy \tag{3.3}$$

with

$$f(x,y) = x^i\, y^j \tag{3.4}$$

where (x,y) are the image coordinates of point $\mathbf{P} = (x,y)$ belonging to section $\mathcal{S}$ (see Fig. 3.4). Note that since we consider only the shape of the section in the image in the definition of image moments, the function $\rho(x,y)$ involved in (3.1) is now $\rho(x,y) = 1$. Note that we assume that the ultrasound beam is a perfect plane, and that the whole actual cross-section lies in the imaged.

Consider a 6 DOFs robot arm that carries at its end-effector a 2D ultrasound probe transducer (see Fig. 3.5). A 3-D cartesian frame $\{R_s\}$ is attached the probe sensor. This body frame is defined by the three orthogonal axes X, Y, and Z. (X, Y) are defined in such a way they lie in the image plane, while Z axis is normal to the latter (see Fig. 3.5, 3.6, and 3.7). Let 6 dimension vector $\mathbf{v}$ represent the velocity of the transducer (probe) in the 3-D space. More precisely, it represents the body frame velocity of $\{R_s\}$ expressed in $\{R_s\}$. It is denoted by $\mathbf{v} = (\boldsymbol{v}, \boldsymbol{\omega})$, where $\boldsymbol{v} = (v_x,\ v_y,\ v_z)$ and $\boldsymbol{\omega} = (\omega_x,\ \omega_y,\ \omega_z)$ represent respectively the translational and the rotational velocity of the probe. The scalar components $(v_x,\ v_y,\ v_z)$ are respectively along axes X, Y, and Z of the probe, while the scalar components $(\omega_x,\ \omega_y,\ \omega_z)$ are respectively around X, Y, and Z. This is represented on both Fig. 3.5 and 3.7. Note that dealing with either the probe velocity or the robot one is equivalent, provided that the kinematic transformation (i. e., the homogeneous transformation matrix) from the robot end-effector to probe attached frame is known. Obtaining such matrix is referred to as *hand-eye calibration*. If such matrix is not enough accurate, then the corresponding errors would be considered as perturbations to the visual servoing scheme.

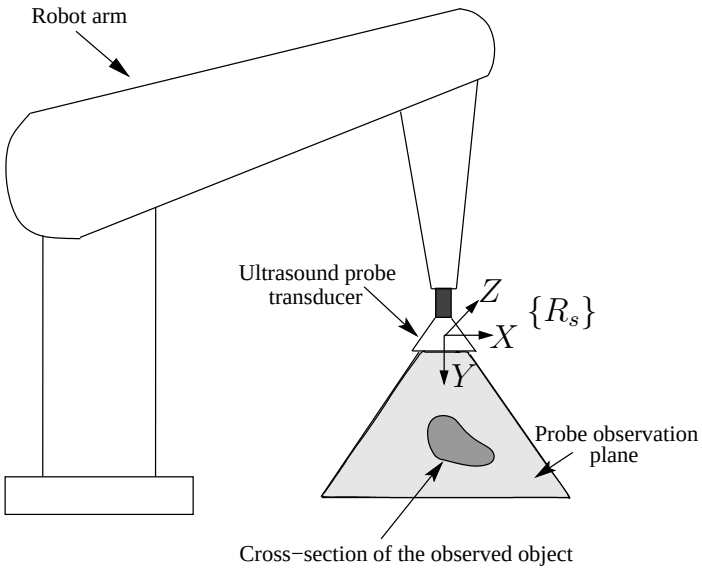


Figure 3.5: A 2D ultrasound probe carried by a robot arm. The probe is interacting with an object, where a cross-section resulting from the intersection of the probe observation plane with the object is shown. The frame (X, Y, Z) attached to the probe is also depicted. The vectors X and Y lies in the probe observation plane, whereas Z is orthogonal to it.

The modeling objective is finally to write the time variation $\dot{m}_{ij}$ of image moment m_{ij}, defined by (3.3) and (3.4), as function of the probe velocity in a linear form. This objective can be formulated as follows:

$$\dot{m}_{ij} = \mathbf{L}_{m_{ij}} \mathbf{v} \tag{3.5}$$

where $\mathbf{L}_{m_{ij}}$ is the interaction matrix related to m_{ij} denoted by:

$$\mathbf{L}_{m_{ij}} = \begin{bmatrix} m_{vx} & m_{vy} & m_{vz} & m_{\omega x} & m_{\omega y} & m_{\omega z} \end{bmatrix} \tag{3.6}$$

such that the six components of $\mathbf{L}_{m_{ij}}$ represent the scalars whose analytical form is what we want to obtain.

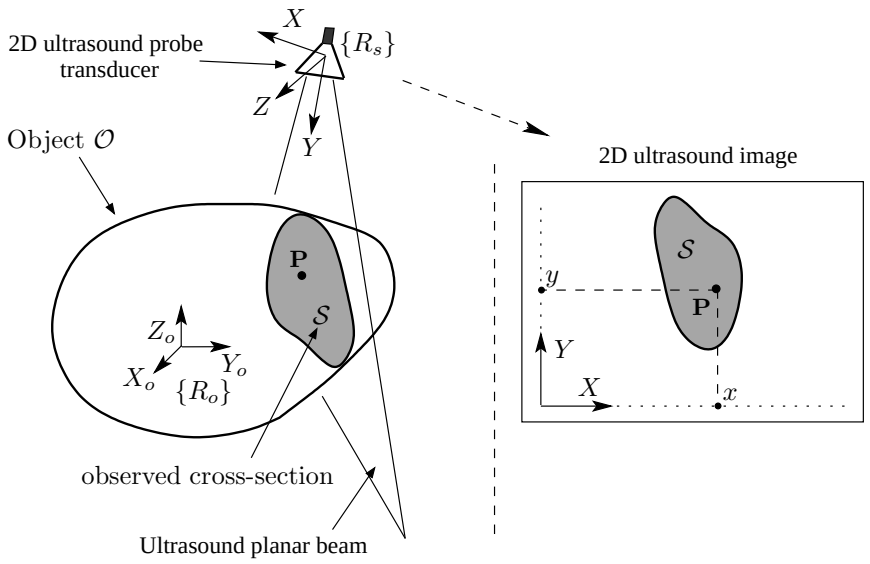


Figure 3.6: Interaction between a 2D ultrasound probe and an object. 3D cartesian frames $\{R_s\}$ and $\{R_o\}$ are attached respectively to the probe and to the object (left). A cross-section $\mathcal{S}$ results from this intersection, where a point $\mathbf{P}$ that belongs to it is shown. The ultrasound planar beam that observes this cross-section reflects it on a 2D ultrasound image (right). Both $\mathcal{S}$ and $\mathbf{P}$ are shown on that image, where the image coordinates $(x,\ y)$ of $\mathbf{P}$ in the 2D image frame (X, Y) are also depicted. Note that the two axes X and Y constituting the image frame (left) clearly correspond to those forming the probe frame $\{R_s\}$ (right).

The time variation $\dot{m}_{ij}$ of image moment m_{ij} can be expressed as function of the image point velocity $(\dot{x}, \dot{y})$, in form of a double integral over section $\mathcal{S}$ as follows [16]:

$$\dot{m}_{ij} = \int\int_{\mathcal{S}} \left[\frac{\partial f}{\partial x}\,\dot{x} + \frac{\partial f}{\partial y}\,\dot{y} + f(x,y)\,\left(\frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y} \right) \right] dx\,dy \tag{3.7}$$

that we prefer to write in the following form that can be readily used afterwards:

$$\dot{m}_{ij} = \int\int_{\mathcal{S}} \left[\frac{\partial}{\partial x}\,(\dot{x}\,f(x,y)) \;+\; \frac{\partial}{\partial y}\,(\dot{y}\,f(x,y)) \right] dx\,dy \tag{3.8}$$

The above relationship requires the analytical form of the image point velocity $(\dot{x}, \dot{y})$ as function of probe velocity $\mathbf{v}$, in order it can be expressed as function of this latter, and thus

to obtain the interaction matrix $\mathbf{L}_{m_{ij}}$. Therefore, we firstly need to model the analytical form of the jacobian matrix $\mathbf{L}_{\mathbf{x}}$, defined by (3.2). It relates the image point velocity $(\dot{x},\ \dot{y})$, of point $\mathbf{P} = (x,\ y)$, to the probe velocity $\mathbf{v}$. We present below new theoretical foundations to obtain $\mathbf{L}_{\mathbf{x}}$.

3.5 Image point velocity modeling

When a 2D ultrasound probe sweeps a region of a soft tissue, the variation of the section in the image strongly depends on the shape of that object. Therefore, the image velocity of points lying in the image section also heavily relies on the object shape. In contrast, this is not the case when dealing with optical imaging systems. When, for example, an eye-in-hand camera is observing an object while performing motions, the image points displacements and thus the image points velocity grossly are not affected by the object shape. The already existing interaction matrix that relates the image points velocity to the camera one does not hold in our case and, thus, can not be used.

To make the illustration of this difference more fair and rigorous, consider two different robotic systems consisting respectively of a 2D ultrasound probe carried by a robot arm, as in our case, and of a perspective camera also carried by another robot arm. Each system is observing and thus interacting with its corresponding object. For the 2D ultrasound probe, the intersection of the transducer observation planar beam with the object results in a cross-section which then is reflected in the ultrasound image (see Fig. 3.5). In case of the camera, however, the object surface encountering the image rays is projected and thus reflected in the camera image (see Fig. 1.4 of Chapter 1). Let $\mathbf{P}$ be a point lying in the cross-section of the object observed by the 2D ultrasound probe, and let $\mathbf{U}$ be another point lying on the second object surface observed under the field of view of the camera. It is clear that when the camera moves, point $\mathbf{U}$ remains at the same position. This fact is correct provided of course that the object is motionless, and that $\mathbf{U}$ is kept within the camera field of view. We can consequently consider $\mathbf{U}$ as *physically the same* point. It is quite not the case for point $\mathbf{P}$. Indeed, when the 2D ultrasound probe is moved and thus positioned at another cross-section, the points in the image are those who belong to this new cross-section which, physically and thus its 3D location, does not correspond to the initial one (see Fig. 3.8). Consequently, the 3-D location of the *new* point $\mathbf{P}$ is different from that previously captured at the initial probe position, even if these two points represent a same image point. That is, the two points $\mathbf{P}(t_0 + \Delta t)$, obtained at time $t_0 + \Delta t$ after the probe was moved, and $\mathbf{P}(t_0)$, obtained at the initial time t_0, are not *physically the same* (referring to this difference, between the two points, by the term "*not the same physical entities*" could also be employed).

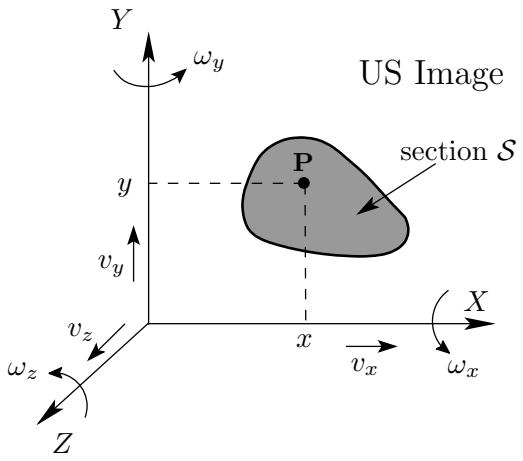


Figure 3.7: Representation of the probe velocity velocity vector on the ultrasound image

New technique to how to model the image point velocity as function of the probe one need consequently to be developed. Note that the statement we provide above is valid when the probe out-of-plane motions occur. If only the in-plane motions occur, it is clear that point $\mathbf{P}$ can be *physically the same*. However, we made a statement for a general case, where all the probe motions are involved, and not for the specific case of in-plane motions. Note also that the modeling we present in this chapter is valid whether only probe in-plane motions, only out-of-plane motions, or both motions are considered.

This manner, discussed above, that according to, a 2D ultrasound probe interacts with its environment yielded consequently the modeling of this interaction quite challenging. This was made worse, because of the strong dependence of the image points variations on the shape of the observed object.

Consider object (organ) $\mathcal{O}$ with which the probe is interacting. Let $\{R_o\}$ be a 3-D cartesian frame attached to this object (see Fig. 3.6). Let ${}^s\mathbf{R}_o$ be the rotation matrix representing the orientation of $\{R_o\}$ with respect to probe frame $\{R_s\}$, and ${}^s\mathbf{t}_o = (t_x,\ t_y,\ t_z)$ be the translation vector defining the origin of $\{R_o\}$ with respect to $\{R_s\}$. Consider now point $\mathbf{P}$ which, we recall, lies on cross-section $\mathcal{S}$ (see Fig. 3.5 and Fig. 3.6). This point lies in the 3D-space, where its coordinates with respect to object frame $\{R_o\}$ are denoted by vector position ${}^o\mathbf{P} = ({}^ox,\ {}^oy,\ {}^oz)$. The coordinates of $\mathbf{P}$ in probe frame $\{R_s\}$ are denoted by vector position ${}^s\mathbf{P}$, which represents nothing but the image coordinates of $\mathbf{P}$. It is thus given by ${}^s\mathbf{P} = (x,\ y,\ 0)$, (see Fig. 3.6 right). Its first two elements x and y represent respectively the abscissa and the ordinate of this point in the ultrasound image. Note that its third

element is equal to zero since this point lies within the probe observation plane and thus has no elevation in Z direction. The image coordinates of point $\mathbf{P}$ can thus be expressed as function to its 3-D coordinates in the object frame as follows (see Appendix A.4):

$$ {}^{s}\mathbf{P} = {}^{s}\mathbf{R}_o \, {}^{o}\mathbf{P} + {}^{s}\mathbf{t}_o \tag{3.9} $$

We recall that our modeling first objective is to write the image point velocity $(\dot{x}, \dot{y})$ as function of probe velocity $\mathbf{v}$. This image velocity is enclosed in the vector ${}^{s}\dot{\mathbf{P}}$, that is equal to ${}^{s}\dot{\mathbf{P}} = (\dot{x},\ \dot{y},\ 0)$. That is the reason why we derive with respect to time t vector ${}^{s}\mathbf{P}$ given by the relationship (3.9). This yields:

$$ {}^{s}\dot{\mathbf{P}} = {}^{s}\dot{\mathbf{R}}_o \, {}^{o}\mathbf{P} + {}^{s}\mathbf{R}_o \, {}^{o}\dot{\mathbf{P}} + {}^{s}\dot{\mathbf{t}}_o \tag{3.10} $$

where ${}^{s}\dot{\mathbf{R}}_o$, ${}^{s}\dot{\mathbf{t}}_o$, and ${}^{o}\dot{\mathbf{P}}$ represent the time variation of respectively rotation matrix ${}^{s}\mathbf{R}_o$, translation vector ${}^{s}\mathbf{t}_o$, and vector position ${}^{o}\mathbf{P}$. The above relationship requires however at least a brief interpretation before we should continue. The entity ${}^{s}\dot{\mathbf{P}}$ represents the velocity of point $\mathbf{P}$ in the image, while ${}^{o}\dot{\mathbf{P}}$ represents its velocity in the 3-D space. Point $\mathbf{P}$, as was introduced, is a moving "*particle*" that slides through the observed object according to the displacements of the probe planar beam (see Fig. 3.8). It is in fact not a concrete point, but a virtual one. If the 2D ultrasound probe is stabilized on a cross-section of the object, point $\mathbf{P}$ can therefore be attached to a corresponding physical point. Otherwise, it could be related to object's physical points only instantaneously. Virtual point $\mathbf{P}$ thus moves with ${}^{o}\dot{\mathbf{P}}$ velocity with respect to the observed object. As illustration, if a camera system is considered then the term ${}^{o}\dot{\mathbf{P}}$ would be null referring, as we have discussed above, to the fact that point $\mathbf{P}$ would be motionless in the 3-D space, provided of course that the object is motionless. It is quite different in the case of 2D ultrasound. In our case, indeed, $\mathbf{P}$ is moving in the 3D-space and consequently ${}^{o}\dot{\mathbf{P}} \neq \mathbf{0}$. Note that, for notational convenience, the term "$\mathbf{0}$" corresponds to the 3×1 null matrix $\mathbf{0}_{3\times1}$. It will be frequently encountered in the rest of this dissertation.

Till now, the probe velocity $\mathbf{v}$ has not yet appeared in the relationship (3.10). Since the objective is to write ${}^{s}\dot{\mathbf{P}}$ as function of $\mathbf{v}$, we will now make this latter appear. Let us therefore consider the following fundamental kinematic relationship:

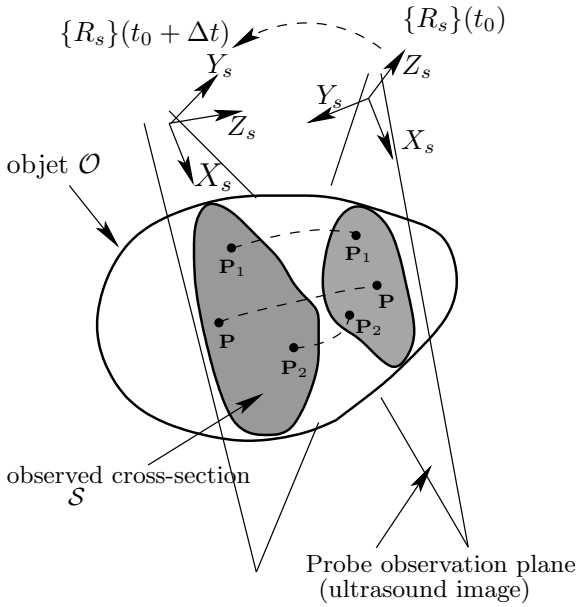


Figure 3.8: Three points P_1, P and P_2 lying in the ultrasound cross-section. They represent therefore the image points. They have been captured at a first time t_0 and at another time $t_0 + \Delta t$ after the probe had been moved from its initial location (pose). We can note that each point is not *physically the same* as its corresponding point lying in the other cross-section section (image), although they represent a same image point.

$$\left\{ \begin{array}{rcl} {}^s\dot{\mathbf{R}}_o & = & -\left[\boldsymbol{\omega}\right]_\times {}^s\mathbf{R}_o \\ {}^s\dot{\mathbf{t}}_o & = & -\boldsymbol{v} + \left[{}^s\mathbf{t}_o\right]_\times \boldsymbol{\omega} \end{array} \right. \tag{3.11}$$

where $[a]_\times$ denotes the skew symmetric matrix associated to vector a (see Appendix A.2). This above relationship relates time variation ${}^s\dot{\mathbf{R}}_o$ of rotation matrix ${}^s\mathbf{R}_o$ and time variation ${}^s\dot{\mathbf{t}}_o$ of translation vector ${}^s\mathbf{t}_o$ as function of probe velocity $\mathbf{v} = (\boldsymbol{v}, \boldsymbol{\omega})$. Thus replacing this relationship in (3.10), we have:

$${}^s\dot{\mathbf{P}} = -\left[\boldsymbol{\omega}\right]_\times {}^s\mathbf{R}_o \; {}^o\mathbf{P} - \boldsymbol{v} + \left[{}^s\mathbf{t}_o\right]_\times \boldsymbol{\omega} + {}^s\mathbf{R}_o \, {}^o\dot{\mathbf{P}} \tag{3.12}$$

Recalling the vector cross-product properties (see Appendix A.2), we then have:

$$ {}^s\dot{\mathbf{P}} = -\boldsymbol{v} - [\boldsymbol{\omega}]_\times \, {}^s\mathbf{R}_o \, {}^o\mathbf{P} - [\boldsymbol{\omega}]_\times \, {}^s\mathbf{t}_o + {}^s\mathbf{R}_o \, {}^o\dot{\mathbf{P}} \tag{3.13} $$

that can be written:

$$ {}^s\dot{\mathbf{P}} = -\boldsymbol{v} - [\boldsymbol{\omega}]_\times \, ({}^s\mathbf{R}_o \, {}^o\mathbf{P} + {}^s\mathbf{t}_o) + {}^s\mathbf{R}_o \, {}^o\dot{\mathbf{P}} \tag{3.14} $$

Recalling the expression of ${}^s\mathbf{P}$ given by (3.9), we finally obtain:

$$ {}^s\dot{\mathbf{P}} = -\boldsymbol{v} - [\boldsymbol{\omega}]_\times \, {}^s\mathbf{P} + {}^s\mathbf{R}_o \, {}^o\dot{\mathbf{P}} \tag{3.15} $$

that we prefer to write in the following appropriate form:

$$ {}^s\dot{\mathbf{P}} = -\boldsymbol{v} + [{}^s\mathbf{P}]_\times \, \boldsymbol{\omega} + {}^s\mathbf{R}_o \, {}^o\dot{\mathbf{P}} \tag{3.16} $$

which represents the expression of image velocity ${}^s\dot{\mathbf{P}} = (\dot{x}, \dot{y}, 0)$ of point $\mathbf{P}$ as function of probe velocity $\mathbf{v} = (\boldsymbol{v}, \boldsymbol{\omega})$, its image coordinates ${}^s\mathbf{P} = (x, y, 0)$, rotation matrix ${}^s\mathbf{R}_o$, and its velocity ${}^o\dot{\mathbf{P}}$ in the 3-D space.

We want in fact to obtain the image point velocity as function of the velocity of only the probe. It is however not the case for the relationship (3.16), where the velocity ${}^o\dot{\mathbf{P}}$ is also involved. Entity ${}^o\dot{\mathbf{P}}$ therefore needs to be replaced. Point $\mathbf{P}$ results from the intersection of the probe observation plane with the object. Its velocity ${}^o\dot{\mathbf{P}}$ represents the velocity of its displacements in the 3D space according to the displacements of the ultrasound planar beam. Point $\mathbf{P}$ always remains in the probe planar beam emitted by the probe even when this latter moves. Therefore ${}^o\dot{\mathbf{P}}$ is obviously related to the probe motions, and thus is inevitably constrained by probe velocity $\mathbf{v}$. Indeed, this is shown in what follows where we establish two constraints that point $\mathbf{P}$ can fulfill. Those two constraints then are used to replace ${}^o\dot{\mathbf{P}}$ as function of $\mathbf{v}$ in the relationship (3.16). A first key solution to obtain them consists in dealing with the surface of the observed object.

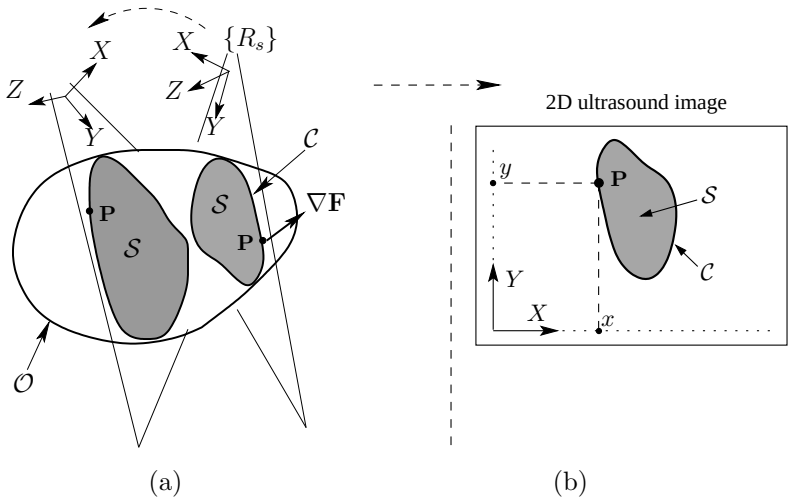


Figure 3.9: Contour points - (a) A point **P** that lies on the cross-section's contour $\mathcal{C}$ is depicted along with the observed object in the 3-D space. Both its two locations when the 2D ultrasound probe had been positioned at two different poses are shown. The normal vector $\nabla\mathbf{F}$ to the object surface at **P** is also shown - (b) A 2D ultrasound image provided by the probe transducer.

Let $\mathcal{OS}$ be the set of points that lie on the object surface. Let also $\mathcal{C}$ be the contour of cross-section $\mathcal{S}$ (see Fig. 3.9). It is therefore nothing but the contour in the image of $\mathcal{S}$. Term **P** now denotes a point that lies only on contour $\mathcal{C}$ ($\mathbf{P} \in \mathcal{C}$), and not in the interior of $\mathcal{S}$ as it was so far considered. Therefore, **P** lies on the object surface.

3.5.1 First constraint

The object surface can be defined by a scalar relationship of the form:

$$F({}^{o}\mathbf{P}) = F({}^{o}x,\, {}^{o}y,\, {}^{o}z) = 0 \tag{3.17}$$

where F is a scalar function that represents the shape of object $\mathcal{O}$. The above relationship states that any point that lies on the object surface, as the case for **P**, satisfies $F = 0$. We recall that ${}^{o}\mathbf{P} = ({}^{o}x,\, {}^{o}y,\, {}^{o}z)$ represent the 3-D coordinates of **P** in object frame $\{R_o\}$.

When the 2D ultrasound probe moves and thus sweeps the observed object, point **P** also moves accordingly in the 3-D space in such a way it always remains within the probe planar

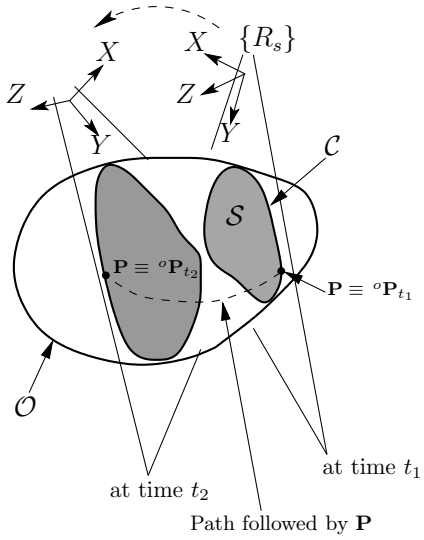


Figure 3.10: Point $\mathbf{P}$ sliding on the object surface when the probe plane is moving. Its 3-D position ${}^o\mathbf{P}_{t_1}$ when the probe is at an initial position at time t_1, and its 3-D position ${}^o\mathbf{P}_{t_2}$ when the probe is at another position at time t_2 are depicted. We can see that ${}^o\mathbf{P}_{t_1} \neq^o \mathbf{P}_{t_2}$. The path that $\mathbf{P}$ had followed is also depicted. Such path lies on the object surface.

beam. This is due to the fact that $\mathbf{P}$ results from the intersection of that planar beam with the object. Virtual point $\mathbf{P}$, as now defined, *always* lies on contour $\mathcal{C}$ of the image, and therefore it remains on object surface $\mathcal{OS}$, even with the displacements of the 2D ultrasound probe (i. e., $\forall$ probe positions, $\mathbf{P} \in \mathcal{OS}$), provided of course that the probe plane does not get out of the object. Consequently, $\mathbf{P}$ *always* satisfies the relation (3.17) throughout its motions. That is, when $\mathbf{P}$ has moved from an initial location ${}^o\mathbf{P}_{t_1}$, captured at time t_1, to another different location ${}^o\mathbf{P}_{t_2}$, captured at time t_2, $({}^o\mathbf{P}_{t_1} \neq {}^o\mathbf{P}_{t_2})$, function F is still equal to zero, i. e., $F({}^o\mathbf{P}_{t_1}) = F({}^o\mathbf{P}_{t_2}) = 0$, due to the fact that $\mathbf{P}$ is still on the object surface (see Fig. 3.10). Consequently, since $F({}^o\mathbf{P})$ remains constant, its time derivative is equal to zero. This can be formulated by:

$$\dot{F}({}^o x,\, {}^o y,\, {}^o z) = 0\ ,\ \forall \mathbf{P} \in \mathcal{OS} \tag{3.18}$$

Assuming object $\mathcal{O}$ is rigid, we can write:

$$\begin{aligned} \dot{F}({}^o x,\, {}^o y,\, {}^o z) &= \frac{\partial F}{\partial {}^o x}\, {}^o\dot{x} + \frac{\partial F}{\partial {}^o y}\, {}^o\dot{y} + \frac{\partial F}{\partial {}^o z}\, {}^o\dot{z} \\ &= {}^o\nabla\mathbf{F}^\top\, {}^o\dot{\mathbf{P}} \end{aligned} \tag{3.19}$$

where ${}^o\nabla\mathbf{F} = (\frac{\partial F}{\partial {}^o x},\ \frac{\partial F}{\partial {}^o y},\ \frac{\partial F}{\partial {}^o z})$ is the gradient vector of F. It is expressed in object frame $\{R_o\}$. It represents the normal vector to the object surface at point $\mathbf{P}$ (see Fig. 3.9). Since $\dot{F}({}^o\mathbf{P}) = 0$, we finally obtain:

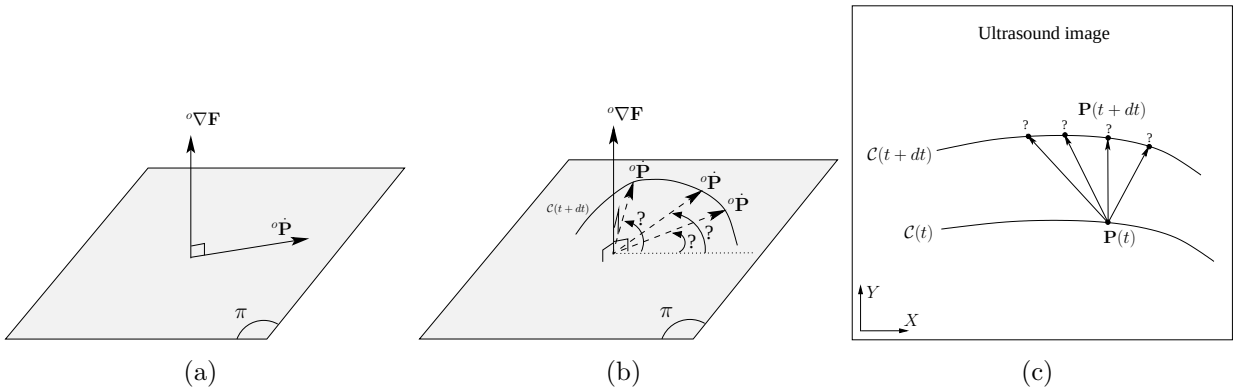
$${}^o\nabla\mathbf{F}^\top\, {}^o\dot{\mathbf{P}} = 0 \tag{3.20}$$

which represents the first constraint on velocity of point $\mathbf{P}$ in the 3-D space. This constraint states that velocity vector ${}^o\dot{\mathbf{P}}$ in the 3-D space of point $\mathbf{P}$ is orthogonal to normal vector $\nabla\mathbf{F}$.

Consider plane π to which $\nabla\mathbf{F}$ is orthogonal. The relationship (3.20) states, in fact, that vector ${}^o\dot{\mathbf{P}}$ lies on π (see Fig. 3.11(a)). There is however an infinity of possible orientations with which a vector can lie on a defined plane. This is, therefore, also the case for vector ${}^o\dot{\mathbf{P}}$ (see Fig. 3.11(b)). Yet, point $\mathbf{P}$ represents a "*particle*" whose velocity, namely ${}^o\dot{\mathbf{P}}$, should clearly have an orientation in the 3-D space and thus on π (since ${}^o\dot{\mathbf{P}}$ lies on π). This is described in the following, where we show that ${}^o\dot{\mathbf{P}}$ can satisfy a second constraint related to its orientation on π.

3.5.2 Second constraint

Entity ${}^o\dot{\mathbf{P}}$ is a velocity that represents the differential displacements of point $\mathbf{P}$, over a dt differential time span, from a 3-D space position ${}^o\mathbf{P}(t)$, at time t, to position ${}^o\mathbf{P}(t+dt)$, at time $t+dt$ (see Fig. 3.12). Note that in contrast to $\mathbf{P}$ that is a virtual point, both ${}^o\mathbf{P}(t+dt)$ and ${}^o\mathbf{P}(t)$ are *physical points*. Indeed, ${}^o\mathbf{P}$ represents the 3D coordinates of a point attached to the surface of the object, while $\mathbf{P}$ represents a particle that is not attached to the object surface but instead slides on it; at time t point $\mathbf{P}$ coincides with ${}^o\mathbf{P}(t)$, while at time $t+dt$ it instead coincides with point ${}^o\mathbf{P}(t+dt)$. – Note also that we made above a statement for the case of general probe motions where both in-plane and out-of-plane motions are involved. In the case only the in-plane motions are involved, point $\mathbf{P}$ can be attached to the object surface and, thus, can be considered as a *physical point*. – However, when the out-of-plane motions are involved, there is an infinity of points on the contour (object surface) with which $\mathbf{P}$ can coincide at time $t+dt$, since $\mathbf{P}$ is a virtual one (i. e., there is an infinity of



(a) (b) (c)

Figure 3.11: Orthogonality between vectors ${}^o\nabla\mathbf{F}$ and ${}^o\dot{\mathbf{P}}$ deduced from the first constraint, given by the relationship (3.20) - (a) Vector ${}^o\dot{\mathbf{P}}$ lies within plane π, represented by its normal ${}^o\nabla\mathbf{F}$ - (b) Vector ${}^o\dot{\mathbf{P}}$ can lie on π with an infinity of possible directions - (c) Evolution of image point $\mathbf{P}$ due to an out-of-plane motion.

possibilities for point a ${}^o\mathbf{P}(t+dt)$). This is represented with Fig. 3.11(c). We illustrate this in the following. Consider a soft tissue object whose surface is exactly fitted with a grid that encloses it. – Such grid is made, for example, from a material that yields it well visualized in the 2D ultrasound image. – The grid consists of lines that homogeneously travel along the object surface. The cross-section 2D ultrasound image thus shows mainly discrete points in the image, points that result from the intersection of the grid with the ultrasound beam. In fact, doing so, the problem is translated into a discrete one, in terms of the set of considered image points. When the probe moves, along its orthogonal axis, for example, the intersection points accordingly slide on the grid's lines. Let us therefore consider one point to make the illustration more fair. The velocity of the point in the image is ${}^s\dot{\mathbf{P}}$, while in the 3D space it is ${}^o\dot{\mathbf{P}}$. It is clear that the grid can fit the object with an infinity of configurations (as two examples of extreme configurations: the grid's lines might travel along a sagittal plane of the object or along its coronal plane). Therefore, there is an infinity of directions with which $\mathbf{P}$ might slide on the object surface, since the point slides on a grid's line, which can have an infinity of configurations and thus of directions; of course, the point must remain on the object surface (this is already satisfied thanks to the first constraint formulated by the relationship (3.20)). Yet, both the infinity of grids describe the same soft tissue object. Consequently, we can freely define a direction for the motions of $\mathbf{P}$ in the 3D space, and thus the direction of ${}^o\dot{\mathbf{P}}$ (on the tangent plane, of course). Defining a direction comes to set and perform a matching between the two points ${}^o\mathbf{P}(t)$ and ${}^o\mathbf{P}(t+dt)$. Since however, as described above, the second point ${}^o\mathbf{P}(t+dt)$ can have an infinity of locations on the contour, a correspondence between these two points could be performed only through a *virtual* matching. In other words, a correspondence rule (protocol) between the

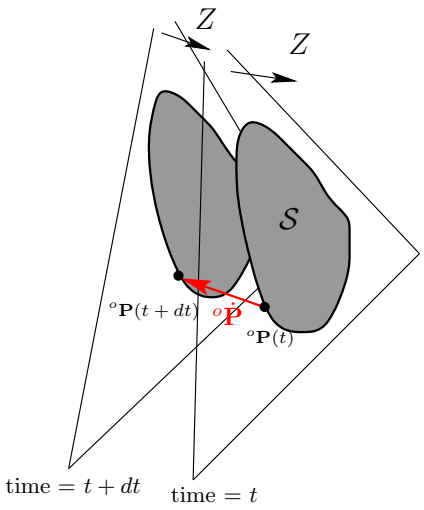


Figure 3.12: Velocity ${}^o\dot{\mathbf{P}}$ (in red), in the 3-D space, of a contour point $\mathbf{P}$.

points need to be set, with which the *virtual matching* has to comply. This is done in the following, where between the infinity of directions that ${}^o\dot{\mathbf{P}}$ might have, we choose one that seems quite tangible. To summarize, this comes to choose a point $\mathbf{P}(t+dt)$ to locate $\mathbf{P}(t)$ on the contour $\mathcal{C}(t+dt)$. Note that this way to proceed is valid since the point velocity we model will be used to determine the variation of image moment, that thanks to its integral effect requires only that the point have to be located on $\mathbf{C}(t+dt)$. In other words, choosing another location for $\mathbf{P}(t+dt)$ would modify the result of image point velocity, but would not change the result of image moment time variation.

Velocity ${}^o\dot{\mathbf{P}}$ is in fact generated by the probe out-of-plane motions. When the probe plane sweeps a surface of a considered object, point $\mathbf{P}$ moves accordingly in such a way it remains within the probe plane. Such sweep motions are mainly represented by the probe out-of-plane motions. If for example only the probe in-plane motions are performed, then velocity ${}^o\dot{\mathbf{P}}$ would be null. The out-of-plane motions lead the probe plane gets out of the initial plane. Such motions are generated by the velocities v_z, ω_x, and ω_y of the probe. Consider the Z axis of the probe frame $\{R_s\}$ (e. g., see Fig. 3.9 and Fig. 3.7). It represents in fact the orthogonal vector to the in-plane motions (v_x, v_y, ω_z). Since, as we highlighted above, vector ${}^o\dot{\mathbf{P}}$ would be null if only the in-plane motions are involved, its tangible direction seems therefore according to the probe Z axis, which is orthogonal to such motions. Let us make such statement more illustrated. Consider a cylindrical object which is orthogonal to the probe plane. When the probe moves along its orthogonal axis Z, it is clear that the most tangible direction that point $\mathbf{P}$ moves along is the Z direction (i. e., the direction of ${}^o\dot{\mathbf{P}}$ is Z). Consequently, we select the Z axis as the direction that, according to, the

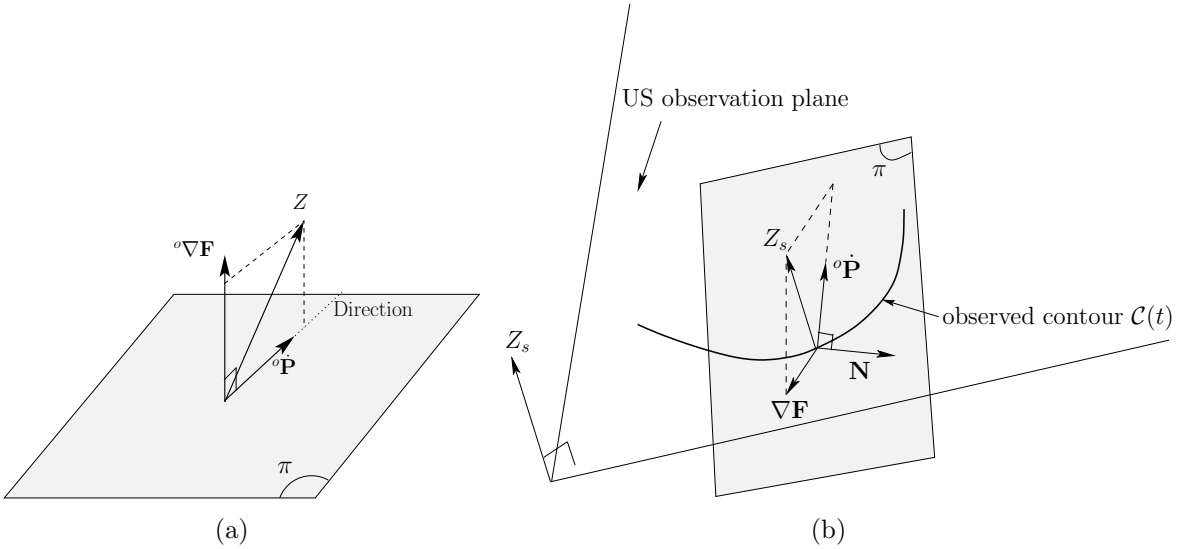


Figure 3.13: Direction of $^o\dot{\mathbf{P}}$ within the plane π. Note that both $^o\dot{\mathbf{P}}$ and $^o\dot{\mathbf{N}}$ lie within the plane π, which is not necessarily the case for the vector Z and the contour $\mathcal{C}$. This latter lies within the ultrasound image plane.

point $^o\mathbf{P}$ would move. – Note that by "according to the direction" we do not mean "along the direction", as it is more precisely described afterwards. – Such statement represents in fact a protocol for the *virtual* matching. Each virtual point would move in the 3-D space according to the direction of Z, i. e., the point $\mathbf{P}$ moves from the position $^o\mathbf{P}(t)$ according to the direction Z to reach the position $^o\mathbf{P}(t+dt)$. The matching objective is that all the points lying in contour $\mathcal{C}(t)$, at time t, match their respective corresponding points on the probe plane at time $t+dt$, in such a way that the whole of matched points can constitute contour $\mathcal{C}(t+dt)$. In other words, the objective is to model the configuration changes of contour C in the image as function of the probe velocity. We can recall (Section 3.1) that the visual features we use, namely the image moments, do not require matching of the points in the image but only matching of the contour. This is therefore (although already roughly highlighted) of great interest in our case, where only the contour can be matched but not the points that are instead virtually matched. Indeed, image moments do not require to specify which point corresponds to another one on the precedent image, but instead they only require that the new points (as $^o\mathbf{P}(t+dt)$) lie on the contour $\mathcal{C}(t+dt)$. That is the reason why, as introduced in this chapter, image moments represent relevant visual features when dealing with the 2D ultrasound imaging modality.

Hence, vector $^o\dot{\mathbf{P}}$ lies according to the direction of Z. However, the latter does not lie in plane π as it is the case for $^o\dot{\mathbf{P}}$. Therefore, the projection of Z on π represents the vector

whose direction is that of ${}^{o}\dot{\mathbf{P}}$ [see Fig. 3.13(a)]. – That is the reason why above we distinguished between the term "according to" and "in direction of" when referring to the relation between the vectors Z and ${}^{o}\dot{\mathbf{P}}$.

Let therefore vector $\mathbf{N}$ be normal to the plane formed by $\nabla\mathbf{F}$ and Z [see Fig. 3.13(b)]. This vector can be obtained from the following vector cross-product:

$$ {}^{o}\mathbf{N} = {}^{o}\mathbf{Z}_s \times {}^{o}\nabla\mathbf{F} \tag{3.21} $$

where ${}^{o}\mathbf{N}$ is the expression of $\mathbf{N}$ in object frame $\{R_o\}$. Since ${}^{o}\mathbf{N}$ is orthogonal to the plane formed by Z and $\nabla\mathbf{F}$ wherein vector ${}^{o}\dot{\mathbf{P}}$ is lying, the two vectors ${}^{o}\mathbf{N}$ and ${}^{o}\dot{\mathbf{P}}$ are consequently clearly orthogonal. This can be formulated by:

$$ {}^{o}\mathbf{N}^{\top}\, {}^{o}\dot{\mathbf{P}} = 0 \tag{3.22} $$

which represents the second constraint on the direction of vector velocity ${}^{o}\dot{\mathbf{P}}$.

3.5.3 Virtual point velocity

After having established two constraints given by the relationships (3.20) and (3.22), image point velocity $(\dot{x}, \dot{y})$ can finally be related as function of probe velocity $\mathbf{v}$. The previously obtained relationship of the image point velocity (3.16) is a system of three scalar equations with five scalar unknowns ${}^{s}\dot{\mathbf{P}} = (\dot{x}, \dot{y}, 0)$ and ${}^{o}\dot{\mathbf{P}} = ({}^{o}\dot{x},\ {}^{o}\dot{y},\ {}^{o}\dot{z})$. Thanks to the two constraints (3.20) and (3.22), which represent two scalar relationships, the whole number of equations is raised to five scalar relationships, thus equalizing the number of unknowns, and therefore yielding to a unique solution of this relationships system.

Firstly, going back to the relationship (3.16), it can be written:

$$ {}^{s}\mathbf{R}_o^{\top}\, {}^{s}\dot{\mathbf{P}} \;=\; -{}^{s}\mathbf{R}_o^{\top}\, \boldsymbol{v} \;+\; {}^{s}\mathbf{R}_o^{\top}\, [{}^{s}\mathbf{P}]_{\times}\, \boldsymbol{\omega} \;+\; {}^{o}\dot{\mathbf{P}} \tag{3.23} $$

Multiplying it once by ${}^{o}\nabla\mathbf{F}^{\top}$ and then by ${}^{o}\mathbf{N}^{\top}$, we obtain after recalling the constraints (3.20) and (3.22):

$$\left\{\begin{array}{l} {}^{o}\nabla\mathbf{F}^{\top}\,{}^{s}\mathbf{R}_{o}^{\top}\,{}^{s}\dot{\mathbf{P}} = -{}^{o}\nabla\mathbf{F}^{\top}\,{}^{s}\mathbf{R}_{o}^{\top}\,\boldsymbol{v} + {}^{o}\nabla\mathbf{F}^{\top}\,{}^{s}\mathbf{R}_{o}^{\top}\,[{}^{s}\mathbf{P}]_{\times}\,\boldsymbol{\omega} \\ {}^{o}\mathbf{N}^{\top}\,{}^{s}\mathbf{R}_{o}^{\top}\,{}^{s}\dot{\mathbf{P}} = -{}^{o}\mathbf{N}^{\top}\,{}^{s}\mathbf{R}_{o}^{\top}\,\boldsymbol{v} + {}^{o}\mathbf{N}^{\top}\,{}^{s}\mathbf{R}_{o}^{\top}\,[{}^{s}\mathbf{P}]_{\times}\,\boldsymbol{\omega} \end{array}\right. \tag{3.24}$$

Expressions ${}^{s}\nabla\mathbf{F}$ and ${}^{s}\mathbf{N}$ of respectively vectors $\nabla\mathbf{F}$ and $\mathbf{N}$ in probe frame $\{R_s\}$ can be obtained by:

$$\left\{\begin{array}{lcl} {}^{s}\nabla\mathbf{F} & = & {}^{s}\mathbf{R}_{o}\,{}^{o}\nabla\mathbf{F} \\ {}^{s}\mathbf{N} & = & {}^{s}\mathbf{R}_{o}\,{}^{o}\mathbf{N} = {}^{s}\mathbf{Z}_{s}\times{}^{s}\nabla\mathbf{F} \end{array}\right. \tag{3.25}$$

Replacing the above relationships in (3.24), we have:

$$\left\{\begin{array}{l} {}^{s}\nabla\mathbf{F}^{\top}\,{}^{s}\dot{\mathbf{P}} = -{}^{s}\nabla\mathbf{F}^{\top}\,\boldsymbol{v} + {}^{s}\nabla\mathbf{F}^{\top}\,[{}^{s}\mathbf{P}]_{\times}\,\boldsymbol{\omega} \\ {}^{s}\mathbf{N}^{\top}\,{}^{s}\dot{\mathbf{P}} = -{}^{s}\mathbf{N}^{\top}\,\boldsymbol{v} + {}^{s}\mathbf{N}^{\top}\,[{}^{s}\mathbf{P}]_{\times}\,\boldsymbol{\omega} \end{array}\right. \tag{3.26}$$

that represents a system of two scalar equations with two scalar unknowns $\dot{x}$ and $\dot{y}$, which yields to the unique following solution:

$$\left\{\begin{array}{lcl} \dot{x} & = & -v_x - K_x\,v_z - y\,K_x\,\omega_x + x\,K_x\,\omega_y + y\,\omega_z \\ \dot{y} & = & -v_y - K_y\,v_z - y\,K_y\,\omega_x + x\,K_y\,\omega_y - x\,\omega_z \end{array}\right. \tag{3.27}$$

with:

$$\left\{\begin{array}{lcl} K_x & = & f_x\,f_z\,/\,(f_x^2 + f_y^2) \\ K_y & = & f_y\,f_z\,/\,(f_x^2 + f_y^2) \end{array}\right. \tag{3.28}$$

such that ${}^{s}\nabla\mathbf{F} = (\,f_x,\ f_y,\ f_z\,)$.

The final obtained relationship (3.27) relates the image velocity of points lying on the image contour as function of the probe velocity.

The jacobian matrix $\mathbf{L_x}$ is finally derived from (3.27) as follows:

$$\mathbf{L_x} = \begin{bmatrix} -1 & 0 & -K_x & -y\,K_x & x\,K_x & y \\ 0 & -1 & -K_y & -y\,K_y & x\,K_y & -x \end{bmatrix} \tag{3.29}$$

We can note that the terms relating probe-in-plane motions ($v_x,\ v_y,\ \omega_z$) require only the image coordinates (x, y) of the considered point lying on the image contour, while the terms relating out-of-plane motions ($v_z,\ \omega_x,\ \omega_y$) require also the knowledge of normal vector ${}^s\nabla\mathbf{F}$ expressed in probe frame $\{R_s\}$. Note also that this jacobian is not affected by the amplitude of ${}^s\nabla\mathbf{F}$, but only its direction. This can be deduced since its the case for the two constraints (3.20) and (3.22) used to derive $\mathbf{L_x}$. Indeed, those two relationships employ only the direction of ${}^s\nabla\mathbf{F}$ to constrain ${}^o\dot{\mathbf{P}}$. An easy way to verify this, is that if the amplitude of ${}^s\nabla\mathbf{F}$ is varied, coefficients K_x and K_y remain unchanged.

This first result we obtained above is now exploited to develop the relationship of image moment time variation $\dot{m}_{ij}$ as function of probe velocity $\mathbf{v}$, as was introduced and described by (3.2). This is presented in the following section.

3.6 Image moments time variation modeling

The modeling objective is now to relate time variation $\dot{m}_{ij}$ of image moment m_{ij} as function of probe velocity $\mathbf{v}$ according to the linear relationship (3.5). To do that, we go back to relationship (3.8). However, that relationship is function of image velocity of the whole points lying in image section $\mathcal{S}$. It requires therefore the relationship of that image velocity. In the previous section, the image velocity we modeled is nevertheless that of points lying only on the image contour $\mathcal{C}$, and not that of the whole section points. Consequently, the relationship (3.8) can not be used as is. It requires instead to be formulated as function of the image velocity $(\dot{x}, \dot{y})$ of only the points lying on section's contour $\mathcal{C}$. This can be performed thanks to the Green's theorem [73].

The Green's theorem states indeed a relationship between a line integral around a simple

closed curve and a double integral over a planar region bounded by that curve. This is therefore equivalent to our case, where the planar region corresponds to image section $\mathcal{S}$ and its bounding curve to contour $\mathcal{C}$. Considering two scalar functions F_x and F_y, the Green's theorem is given by:

$$\oint_{\mathcal{C}} F_x \, dx + \oint_{\mathcal{C}} F_y \, dy = \iint_{\mathcal{S}} \left(\frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y} \right) dx \, dy \tag{3.30}$$

This formula can thus be used to express (3.8), which is formulated as a double integral over section $\mathcal{S}$, in a form of a line integral around contour $\mathcal{C}$. Identifying the second term of (3.8) to the second term of (3.30), we directly deduce $F_x = -\dot{y}\, f(x,y)$ and $F_y = \dot{x}\, f(x,y)$. Replacing this result in the first term of the formula (3.30), the image moment time variation is then expressed as a line integral around $\mathcal{C}$ as follows:

$$\dot{m}_{ij} = -\oint_{\mathcal{C}} [\, f(x,y)\, \dot{y}\,]\; dx + \oint_{\mathcal{C}} [\, f(x,y)\, \dot{x}\,]\; dy \tag{3.31}$$

which is function of image velocity $(\dot{x}, \dot{y})$ of points lying only on contour $\mathcal{C}$. We can therefore directly use the previous result of the image velocity of contour points (3.27) in the above relationship. Before doing that, let us express image moment m_{ij} also as a line integral around $\mathcal{C}$.

The Green's theorem given by the formula (3.30) in once again employed, but this time on (3.3). Identifying the second term of the former relationship to the latter one, we can find that $(F_x = -\frac{1}{j+1}\, x^i\, y^{j+1},\ F_y = 0)$ is one solution. Replacing this result in the first term of (3.30), we can thus formulate:

$$m_{ij} = \frac{-1}{j+1} \oint_{\mathcal{C}} x^i\, y^{j+1}\, dx \tag{3.32}$$

Similarly, we can also find that $(F_x = 0,\ F_y = \frac{1}{i+1}\, x^{i+1}\, y^j)$ is another solution. Replacing also this result in the first term of (3.30), we can furthermore formulate m_{ij} by:

$$m_{ij} = \frac{1}{i+1} \oint_{\mathcal{C}} x^{i+1}\, y^j\, dy \tag{3.33}$$

Finally, replacing the image velocity of contour points (3.27) in (3.31), then identifying image moments according to (3.32) and (3.33), we obtain the six elements of interaction matrix $\mathbf{L}_{m_{ij}}$, presented by (3.6), that relates image moment time variation $\dot{m}_{ij}$ as function of probe velocity $\mathbf{v}$. We obtain:

$$\left\{\begin{array}{lcl} m_{vx} & = & -i\, m_{i-1,j} \\ m_{vy} & = & -j\, m_{i,j-1} \\ m_{vz} & = & {}^{x}m_{ij} \; - \; {}^{y}m_{ij} \\ m_{\omega x} & = & {}^{x}m_{i,j+1} \; - \; {}^{y}m_{i,j+1} \\ m_{\omega y} & = & -{}^{x}m_{i+1,j} \; + \; {}^{y}m_{i+1,j} \\ m_{\omega z} & = & i\, m_{i-1,j+1} - j\, m_{i+1,j-1} \end{array}\right. \tag{3.34}$$

where:

$$\left\{\begin{array}{lcl} {}^{x}m_{ij} & = & \oint_{\mathcal{C}} x^i\, y^j\, K_y\, dx \\ {}^{y}m_{ij} & = & \oint_{\mathcal{C}} x^i\, y^j\, K_x\, dy \end{array}\right. \tag{3.35}$$

We thus have reached the modeling objective, consisting in relating image moment time variation $\dot{m}_{ij}$ as function of probe velocity $\mathbf{v}$ in a linear form (3.5) as presented in Section 3.4.

Similarly to the image point velocity (3.27), we note that elements $(m_{vx},\ m_{vy},\ m_{\omega z})$ of the interaction matrix related to the probe-in-plane motions (v_x, v_y, ω_z) require only information from the observed image, namely image moments. As for elements $(m_{vz},\ m_{\omega x},\ m_{\omega z})$ related to the out-of-plane motions $(v_z, \omega_x, \omega_y)$, however they furthermore necessitate the knowledge of normal vector ${}^{s}\nabla\mathbf{F}$ to the object surface at each of the contour points. The normal vector is in fact enclosed in coefficients K_x and K_y, of (3.28), involved in the above relationship. Note also that the interaction matrix is insensitive to ${}^{s}\nabla\mathbf{F}$'s amplitude, since it is as such for jacobian matrix $\mathbf{L}_{\mathbf{x}}$ used to derive it, as shown earlier.

3.7 Interpretation for simple shapes

In this section, the above developed theoretical foundations are applied in order to be validated on some simple shapes, like spheres and cylinders.

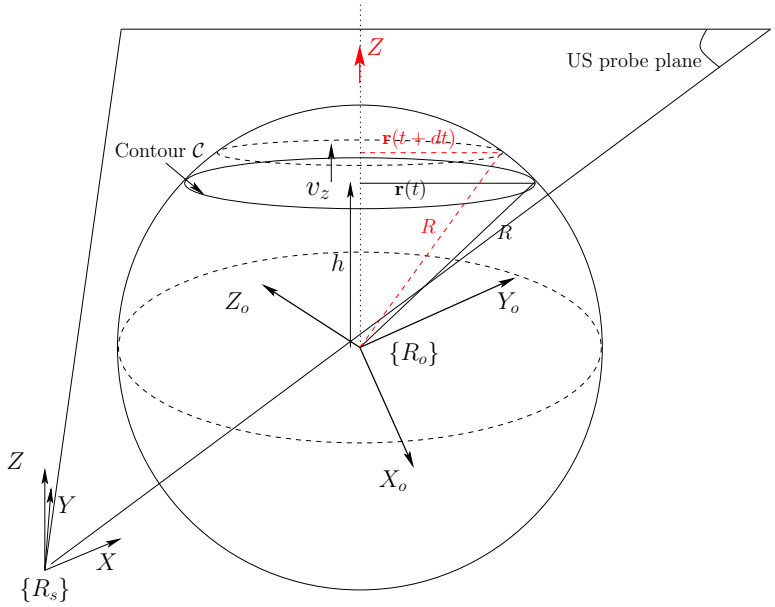


Figure 3.14: A 2D ultrasound probe interacting with a spherical object. When it performs a differential displacements v_z along its z axis, the initial value of the image section radius $r(t)$ changes to the value $r(t+dt)$, (depicted in red).

3.7.1 Spherical objects

Consider the case where a 2D ultrasound probe interacts with a spherical object. Let R be the radius of this sphere. Object frame $\{R_o\}$ is attached to the sphere center. Vector position ${}^s\mathbf{t}_o$ thus defines in this case the coordinates of the sphere center in the probe frame, and rotation matrix ${}^s\mathbf{R}_o$ describes the orientation of $\{R_o\}$ with respect to probe frame $\{R_s\}$ (see Fig. 3.14). We first want to derive the corresponding analytical form of interaction matrix $\mathbf{L}_{m_{ij}}$ by applying the general relationship (3.34) we obtained above. Before, we need to derive also the analytical form of coefficients K_x and K_y involved in the relationship (3.27) of the image point velocity as function of probe velocity $\mathbf{v}$, since these two coefficients are required in the interaction matrix formula (3.34). So, we use the general relationship of K_x and K_y given by (3.28), by applying it to this case.

Image point velocity

In this case, the relationship (3.17) is satisfied by any point $\mathbf{P}$ lying on the sphere surface. It is therefore given by:

$$F({}^o x, {}^o y, {}^o z) = ({}^o x/R)^2 + ({}^o y/R)^2 + ({}^o z/R)^2 - 1 = 0 \tag{3.36}$$

where we recall that ${}^o\mathbf{P} = ({}^o x,\ {}^o y,\ {}^o z)$ is a vector position defining the 3-D coordinates of

point $\mathbf{P}$ in object (sphere) frame $\{R_o\}$. It can be expressed as function of image coordinates ${}^s\mathbf{P} = (x, y, 0)$ of $\mathbf{P}$ as given by (3.9):

$$ {}^o\mathbf{P} = {}^s\mathbf{R}_o^\top \ ({}^s\mathbf{P} - {}^s\mathbf{t}_o) \tag{3.37} $$

Normal vector ${}^o\nabla\mathbf{F}$ to the sphere surface at $\mathbf{P}$ is the gradient vector of scalar function F, and thus is given by ${}^o\nabla\mathbf{F} = (\frac{\partial F}{\partial x},\ \frac{\partial F}{\partial y},\ \frac{\partial F}{\partial z})$. We obviously obtain ${}^o\nabla\mathbf{F} = \frac{2}{R^2}\,{}^o\mathbf{P}$. Replacing ${}^o\mathbf{P}$ with the above relationship, yields:

$$ {}^o\nabla\mathbf{F} = \frac{2}{R^2}\ {}^s\mathbf{R}_o^\top \ ({}^s\mathbf{P} - {}^s\mathbf{t}_o) \tag{3.38} $$

that we express in $\{R_s\}$ by multiplying with rotation matrix ${}^s\mathbf{R}_o$:

$$ {}^s\nabla\mathbf{F} = \frac{2}{R^2}\ {}^s\mathbf{R}_o\,{}^s\mathbf{R}_o^\top \ ({}^s\mathbf{P} - {}^s\mathbf{t}_o) \tag{3.39} $$

Since ${}^s\mathbf{R}_o\,{}^s\mathbf{R}_o^\top = \mathbf{I}$, we obtain after recalling that ${}^s\mathbf{P} = (x,\ y,\ 0)$ and that ${}^s\mathbf{t}_o = (t_x,\ t_y,\ t_z)$ (Section 3.5, pp. 63):

$$ \begin{array}{rcl} {}^s\nabla\mathbf{F} & = & \frac{2}{R^2}\ ({}^s\mathbf{P} - {}^s\mathbf{t}_o) \\ & = & \frac{2}{R^2}\ (x - t_x,\ y - t_y,\ -t_z)^\top \end{array} \tag{3.40} $$

Replacing this result in the relationship (3.28) of the coefficients K_x and K_y, yields:

$$ \left\{ \begin{array}{rcl} K_x & = & -t_z\,(x - t_x)/\big((x - t_x)^2 + (y - t_y)^2\big) \\ K_y & = & -t_z\,(y - t_y)/\big((x - t_x)^2 + (y - t_y)^2\big) \end{array} \right. \tag{3.41} $$

The above relationship can in fact be expressed in a more simple and appropriate form. We first formulate (3.36) as follows:

$$ F({}^ox,\ {}^oy,\ {}^oz) = {}^o\mathbf{P}^\top\,{}^o\mathbf{P} - R^2 = 0 \tag{3.42} $$

Substituting ${}^{o}\mathbf{P}$ according to (3.37) and recalling that ${}^{s}\mathbf{R}_{o}\,{}^{s}\mathbf{R}_{o}^{\top} = \mathbf{I}$, the equation (3.42) becomes:

$$({}^{s}\mathbf{P} - {}^{s}\mathbf{t}_{o})^{\top}\ ({}^{s}\mathbf{P} - {}^{s}\mathbf{t}_{o}) - R^{2} = 0 \tag{3.43}$$

then replacing ${}^{s}\mathbf{P}$ and ${}^{s}\mathbf{t}_{o}$ with their respective expressions yields:

$$(x - t_x)^2 + (y - t_y)^2 + t_z^2 - R^2 = 0 \tag{3.44}$$

The above relationship states that the intersection of a planar beam with a sphere is a disk (or a circle if the spherical object is hollow) of radius $r = \sqrt{R^2 - t_z^2}$ and of center coordinates (t_x, t_y) in the image.

Let a be the area of section $\mathcal{S}$ in the image. In this case it represents the area of the disk region (or of the region surrounded by the circle if the object is hollow). It is therefore given by $a = \pi\, r^2 = \pi\,(R^2 - t_z^2)$. Substituting $(x - t_x)^2 + (y - t_y)^2$ according to (3.44) in (3.41) and taking into account the expression of the area a, we finally obtain the expressions of K_x and K_y in the case of a spherical object as follows:

$$\begin{cases} K_x = -\pi\, t_z\,(x - t_x)/a \\ K_y = -\pi\, t_z\,(y - t_y)/a \end{cases} \tag{3.45}$$

We can note that the image point velocity, from its parameters K_x and K_y obtained above, does not require rotation matrix ${}^{s}\mathbf{R}_{o}$ between the object and the probe frames. This can be explained by the fact that a sphere does not possess any orientation in the 3-D space.

Interaction Matrix

The three elements $(m_{v_z},\ m_{\omega_x},\ m_{\omega_y})$, of the interaction matrix $\mathbf{L}_{mij}$, that relate image moment time variation $\dot{m}_{ij}$ as function of the probe out-of-plane motions $(v_z, \omega_x, \omega_y)$ are formulated by (3.34) for a general case. We want to obtain their specific and simple form in the case of a spherical object. We use the simple form of the coefficients K_x and K_y given by (3.45), derived for the sphere case. These two coefficients are involved in those three elements of the interaction matrix. The three other elements $(m_{v_x},\ m_{v_y},\ m_{\omega_z})$, of the

interaction matrix, that relate the in-plane motions are already given by (3.34) in a simple and appropriate form, since they are function of only the image moments. Replacing (3.45) in (3.35), we have:

$$\begin{cases} {}^{x}m_{ij} = \pi t_z \left[(j+1)m_{ij} - j\, t_y\, m_{i,j-1}\right]/a \\ {}^{y}m_{ij} = \pi t_z \left[-(i+1)m_{ij} + i\, t_x\, m_{i-1,j}\right]/a \end{cases} \tag{3.46}$$

and then obtain the term involved in the three elements $(m_{v_z},\ m_{\omega_x},\ m_{\omega_y})$ of $\mathbf{L}_{m_{ij}}$, as follows:

$${}^{x}m_{ij} - {}^{y}m_{ij} = \frac{\pi\, t_z}{a}\left[(i+j+2)\, m_{ij} - it_x\, m_{i-1,j} - jt_y\, m_{i,j-1}\right] \tag{3.47}$$

As introduced, image moments can describe the configuration of a section in the image. Since this case concerns an interaction with a spherical object, the observed section in the image is a disk or circle. Consequently, the configuration in the image of such section can be described just with three parameters that obviously are: the area a and the image coordinates (x_g, y_g) of the gravity center of the section. These three parameters can be expressed in terms of image moments as follows (e. g., [40]):

$$\begin{cases} a &= m_{00} \\ x_g &= m_{10}/m_{00} \\ y_g &= m_{01}/m_{00} \end{cases} \tag{3.48}$$

The interaction matrices that relate the time variation of a, x_g and y_g can be obtained by applying the relationship of image moment time variation $\dot{m}_{ij}$ given by (3.34), and then using the result (3.47) we obtained for the case of a spherical object. We denote, even in a general case, these three matrices by $\mathbf{L}_a$, $\mathbf{L}_{x_g}$, and $\mathbf{L}_{y_g}$ referring respectively to a, x_g, and y_g. They are obtained as follows:

$$\begin{array}{rcccccccc} \mathbf{L}_a &=& [& 0 & 0 & a_{vz} & a_{\omega x} & a_{\omega y} & 0 &] \\ \mathbf{L}_{x_g} &=& [& -1 & 0 & x_{g_{vz}} & x_{g_{\omega x}} & x_{g_{\omega y}} & y_g &] \\ \mathbf{L}_{y_g} &=& [& 0 & -1 & y_{g_{vz}} & y_{g_{\omega x}} & y_{g_{\omega y}} & -x_g &] \end{array} \tag{3.49}$$

with:

$$\left\{\begin{array}{lcl} a_{vz} & = & 2\,\pi\,t_z \\ a_{\omega x} & = & \pi\,t_z\,(\,3\,y_g - t_y) \\ a_{\omega y} & = & -\pi\,t_z\,(\,3\,x_g - t_x) \\ x_{g_{vz}} & = & \pi\,t_z/a\,(\,x_g - t_x) \\ x_{g_{\omega x}} & = & \pi\,t_z/a\,[\,4\,n_{11} - y_g\,(t_x + 3x_g)\,] \\ x_{g_{\omega y}} & = & \pi\,t_z/a\,[\,-4\,n_{20} + x_g\,(t_x + 3\,x_g)\,] \\ y_{g_{vz}} & = & \pi\,t_z/a\,(\,y_g - t_y) \\ y_{g_{\omega x}} & = & \pi\,t_z/a\,[\,4\,n_{02} - y_g\,(t_y + 3\,y_g)\,] \\ y_{g_{\omega y}} & = & \pi\,t_z/a\,[\,-4\,n_{11} + x_g\,(t_y + 3\,y_g)\,] \end{array}\right. \tag{3.50}$$

where:

$$n_{ij} = m_{ij}/a \tag{3.51}$$

The elements of these interaction matrices that are related to the out-of-plane motions can be expressed in more simple form. We can indeed first notice that since $(t_x,\ t_y)$ represent the center of the disk (or circle), as concluded from the relationship (3.44), they are therefore nothing but the gravity center $(x_g,\ y_g)$ of the section in the image, i. e., $(t_x = x_g,\ t_y = y_g)$. We show in Appendix B.2 that the entities n_{20}, n_{11}, and n_{02}, involved in (3.50), also can be expressed in a more simple form.

Finally, we obtain a more simple form of $\mathbf{L}_a$, $\mathbf{L}_{x_g}$ and $\mathbf{L}_{y_g}$ in case of a spherical object, by replacing (B.30), (B.25), and (B.20) in (3.50), and recalling that $(t_x = x_g,\ t_y = y_g)$, as follows:

$$\begin{array}{lcl} \mathbf{L}_a & = 2\,\pi\,t_z\,[& \begin{array}{cccccc} 0 & 0 & 1 & y_g & -x_g & 0 \end{array}\ ] \\ \mathbf{L}_{x_g} & = & [\ \begin{array}{cccccc} -1 & 0 & 0 & 0 & -t_z & y_g \end{array}\ ] \\ \mathbf{L}_{y_g} & = & [\ \begin{array}{cccccc} 0 & -1 & 0 & t_z & 0 & -x_g \end{array}\ ] \end{array} \tag{3.52}$$

These matrices depend only of the gravity center image coordinates (x_g, y_g), and elevation t_z of the sphere center with respect to the probe frame.

Verification

In case the probe performs a motion along its axis Z, we can calculate time variation $\dot{a}$ of area a with another method different from the general one of this thesis. This allows

us to check the identicalness of the two methods respective results, and thus to verify the correctness and the validity of the developed theoretical foundations of this thesis when applied to this case. The probe velocity is, in this case, $\mathbf{v} = (0,\ 0\ ,\ v_z,\ 0,\ 0,\ 0)$, since only a translational motion along Z is performed. The element of the interaction matrix (3.52) involved in such motions is a_{v_z}, obtained equal to $2\,\pi\,t_z$. Assuming such probe motions, this coefficient is calculated below with another method. Note however that the result that we obtain would be still valid for the case where all the probe motions are applied.

Let h be the elevation of the probe frame from the sphere origin (see Fig. 3.14). It is therefore nothing but $h = -t_z$, where we recall that ${}^s\mathbf{t}_o = (t_x,\ t_y,\ t_z)$ is the vector position defining the coordinates of the origin of frame $\{R_o\}$ in probe frame $\{R_s\}$. Since vector Z of $\{R_s\}$ is orthogonal to the probe observation plane, sphere radius R thus can be expressed as function of elevation h and radius r of the section in the image (see again Fig. 3.14) as follows:

$$R^2 = r^2 + h^2 \tag{3.53}$$

that we derivate with respect to time t, as follows:

$$R\dot{R} = r\,\dot{r} + h\,\dot{h} \tag{3.54}$$

The sphere radius is constant and thus its time derivative $\dot{R}$ is null, i. e., $\dot{R} = 0$. Since h represents the elevation it is clear that $\dot{h} = v_z$. We thus have from (3.54) after recalling that $h = -t_z$:

$$r\dot{r} = -h\,v_z = t_z\,v_z \tag{3.55}$$

Since the image cross-section is a disk (or a circle), area a of the region it covers is given by $a = \pi\,r^2$. Time derivating a yields:

$$\dot{a} = 2\pi\,r\,\dot{r} \tag{3.56}$$

Finally replacing $r\,\dot{r}$ with its expression (3.55), we obtain

$$\dot{a} = 2\pi\,t_z\,v_z \tag{3.57}$$

and thus:

$$a_{v_z} = 2\pi\,t_z \tag{3.58}$$

This result is identical to that previously obtained (3.52) with the general modeling approach. This consequently theoretically validates the general modeling technique of this

thesis, when applied to the case of spherical object, concerning the element involved in the probe motion along its Z axis.

3.7.2 Cylindrical objects

Consider now the case where a 2D ultrasound probe interacts with a cylindrical object (see Fig. 3.15). When the probe performs translational motions along its Z axis, that is $\mathbf{v} = (0,\ 0\ ,\ v_z,\ 0,\ 0,\ 0)$, image section area a clearly does not vary. This means that, even in a general case, the coefficient a_{v_z} that relates time variation $\dot{a}$ of a to probe velocity ($\dot{a} = a_{v_z}\, v_z$) is null ($a_{v_z} = 0$). We want to verify that, by using the general result (3.34) we obtained, and applying it on this case, we can indeed retrieve that expected result, that is $a_{v_z} = 0$. This is shown in what follows.

The coefficient a_{v_z} corresponds to the element m_{v_z} of the formula (3.34), for $i = j = 0$ since $a = m_{00}$. It is thus expressed as follows:

$$a_{v_z} = m_{v_z} = {}^{x}m_{00} - {}^{y}m_{00} \tag{3.59}$$

using (3.35) yields:

$$a_{v_z} = \oint_{\mathcal{C}} K_y\, dx - \oint_{\mathcal{C}} K_x\, dy \tag{3.60}$$

then substituting K_x and K_y with their respective expressions given by (3.28), we have:

$$a_{v_z} = \oint_{\mathcal{C}} \frac{f_y\, f_z}{f_x^2 + f_y^2}\, dx - \oint_{\mathcal{C}} \frac{f_x\, f_z}{f_x^2 + f_y^2}\, dy \tag{3.61}$$

where we recall that ${}^{s}\nabla\mathbf{F} = (f_x, f_y, f_z)$ represents the normal to the object surface (the cylindrical surface in this case) at point $\mathbf{P}$.

The previous relationship (3.17) represents a constraint satisfied by any point lying on the surface of a general object. When the object is cylinder-shaped, as in this case, it is formulated as follows:

$$F({}^{o}x, {}^{o}y, {}^{o}z) = ({}^{o}x/a_1)^2 + ({}^{o}y/a_2)^2 - 1 = 0 \tag{3.62}$$

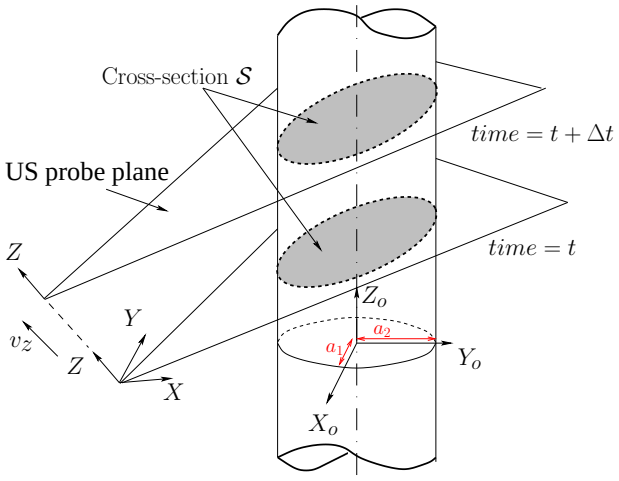


Figure 3.15: A 2D US probe interacting with a cylinder-shaped object. The probe performed an out-of-plane motion with velocity v_z during Δt time span.

where a_1 and a_2 represent the half length values of the cylindrical object main axes (see Fig. 3.15). We recall that vector position ${}^o\mathbf{P} = ({}^ox,\ {}^oy,\ {}^oz)$ represents the 3-D coordinates of point $\mathbf{P}$ in the object frame, and ${}^s\mathbf{P} = (x, y, 0)$ its image coordinates. From the above relationship, we can set the following change of coordinates:

$$\begin{cases} {}^ox &= a_1\, C\theta \\ {}^oy &= a_2\, S\theta \end{cases} \quad ;\ 0 \leqslant \theta < 2\pi \tag{3.63}$$

with $C\theta = \cos(\theta)$ and $S\theta = \sin(\theta)$, such that θ represents the angle in the image.

Normal vector ${}^o\nabla\mathbf{F}$ expressed in $\{R_o\}$ can be derived from function F, given by (3.62), as ${}^o\nabla\mathbf{F} = (\frac{\partial F}{\partial {}^ox},\ \frac{\partial F}{\partial {}^oy},\ \frac{\partial F}{\partial {}^oz},)$. We thus have:

$${}^o\nabla\mathbf{F} = \begin{bmatrix} 2\,{}^ox/a_1^2 \\ 2\,{}^oy/a_2^2 \\ 0 \end{bmatrix} \tag{3.64}$$

substituting ${}^ox$ and ${}^oy$ with their respective expressions (3.63) as function of angle θ, yields:

$${}^o\nabla\mathbf{F} = 2 \begin{bmatrix} C\theta/a_1 \\ S\theta/a_2 \\ 0 \end{bmatrix} \tag{3.65}$$

that can be expressed in probe frame $\{R_s\}$ by multiplying with rotation matrix ${}^s\mathbf{R}_o$:

$$ {}^{s}\nabla\mathbf{F} = 2\,{}^{s}\mathbf{R}_o \begin{bmatrix} C\theta/a_1 \\ S\theta/a_2 \\ 0 \end{bmatrix} \tag{3.66} $$

Coefficient a_{v_z}, which is formulated in (3.61) as a line integral around image contour $\mathcal{C}$, can be expressed as an integral over angle θ as follows:

$$ a_{v_z} = \int_0^{2\pi} \frac{f_z}{f_x^2 + f_y^2} \left(f_y \frac{dx}{d\theta} - f_x \frac{dy}{d\theta}\right) d\theta \tag{3.67} $$

From (3.63) and using the relationship (3.9), image coordinates ${}^{s}\mathbf{P} = (x, y, 0)$ can be expressed as function of angle θ. After denoting r_{kl} the elements of the rotation matrix such that $r_{kl} = {}^{s}\mathbf{R}_o(k, l)$, the derivative of x and y with respect to θ are:

$$ \begin{cases} dx/d\theta &= -a_1(r_{11} - r_{13}/r_{33}\, r_{31})S\theta \\ & \quad +a_2(r_{12} - r_{13}/r_{33}\, r_{32})C\theta \\ dy/d\theta &= -a_1(r_{21} - r_{23}/r_{33}\, r_{31})S\theta \\ & \quad +a_2(r_{22} - r_{23}/r_{33}\, r_{32})C\theta \end{cases} \tag{3.68} $$

replacing this in (3.67), we have:

$$ a_{v_z} = \int_0^{2\pi} \frac{(\epsilon_1\, C\theta^2 + \epsilon_2\, C\theta\, S\theta + \epsilon_3\, S\theta^2)(\epsilon_4\, C\theta + \epsilon_5\, S\theta)}{\epsilon_6\, C\theta^2 + \epsilon_7\, C\theta\, S\theta + \epsilon_8\, S\theta^2}\, d\theta \tag{3.69} $$

where $\epsilon_{k|_{k=1..8}}$ are 3D parameters such that $\epsilon_k = \epsilon_k({}^{s}\mathbf{R}_o,\, a_1,\, a_2)$. We then obtain:

$$ \begin{aligned} a_{v_z} \;=\; & \big[\, \sigma_0 \textstyle\sum_{r_i} \psi_i(r_i)\, \ln(\tan(\theta/2) - r_i) + \\ & \textstyle\sum_{\sigma_k} \sigma_k\, \tan(\theta/2)/[\tan(\theta/2)^2 + 1] \,\big]_0^{2\pi} \end{aligned} \tag{3.70} $$

where σ_k are also 3-D parameters, such that $\sigma_k = \sigma_k({}^{s}\mathbf{R}_o,\, a_1,\, a_2)$. The entity ψ_i is a scalar function of the scalar r_i, where $r_{i|_i}$ represent the roots of the polynomial $\epsilon_6\, r^4 - 2\,\epsilon_7\, r^3 + 2\,(-\epsilon_6 + 2\epsilon_8)r^2 + 2\,\epsilon_7\, r + \epsilon_6$. Consequently, r_i is function of only a_1, a_2 and ${}^{s}\mathbf{R}_o$, i. e., $r_i = r_i({}^{s}\mathbf{R}_o,\, a_1,\, a_2)$. Therefore, in contrast to the entities ψ_i, r_i, and σ_k, only $\tan(\theta)$ is function of the angle θ.

Finally, since $\tan(0) = \tan(\pi) = 0$, we easily obtain from (3.70) that $a_{v_z} = 0$. This result we obtained by applying the general relationship (3.34) on the case of a cylindrical object exactly corresponds to that expected above.

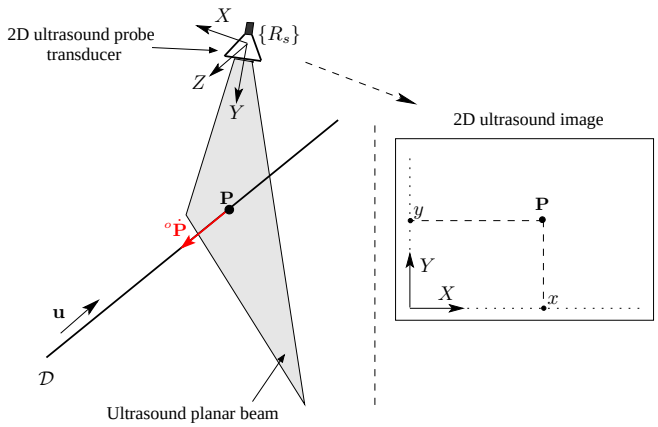


Figure 3.16: A 2D ultrasound probe interacting with a 3D straight line-shaped wire (left). The observed point $\mathbf{P}$ in the image is sketched on the right. It velocity ${}^{o}\dot{\mathbf{P}}$ in the 3-D space is shown in red. Note that its orientation is arbitrarily set. It could either as depicted or in the inverse direction, depending on the probe motions.

3.7.3 Interaction with a 3D straight line

The modeling technique we proposed in this thesis can also be applied to the case where a 2D ultrasound probe interacts with 3D straight line-shaped wire (see Fig. 3.16), although such a geometrical primitive does not correspond to a closed volume. The intersection of the probe plane with the wire results in a point $\mathbf{P}$ in the image, instead of a section in the image. Since we deal with only one image point the concept of image moments seems not relevant to this case. We consider therefore only the modeling of the image velocity ${}^{s}\dot{\mathbf{P}} = (\dot{x},\ \dot{y},\ 0)$ of $\mathbf{P}$.

Let 3-D vector $\mathbf{u}$ represent the orientation of that 3-D straight line, denoted $\mathcal{D}$. We take back the relationship (3.16) in order to model the image point velocity. Its is clear that point $\mathbf{P}$ slides on $\mathcal{D}$. Consequently, vector 3-D velocity ${}^{o}\dot{\mathbf{P}}$ of $\mathbf{P}$ and vector $\mathbf{u}$ of $\mathcal{D}$ are collinear. It is well known that the vector cross-product of two collinear vectors is null, and thus we have:

$$ {}^{o}\mathbf{u} \times {}^{o}\dot{\mathbf{P}} = 0 \tag{3.71} $$

where ${}^{o}\mathbf{u}$ is the expression of $\mathbf{u}$ in the object frame. The above constraint can be expressed in probe frame $\{R_s\}$ instead of $\{R_o\}$ by multiplying with ${}^{s}\mathbf{R}_{o}$. We thus have:

$$\begin{array}{rl} & ({}^s\mathbf{R}_o\,{}^o\mathbf{u}) \times ({}^s\mathbf{R}_o\,{}^o\dot{\mathbf{P}}) = 0 \\ \Leftrightarrow & {}^s\mathbf{u} \times ({}^s\mathbf{R}_o\,{}^o\dot{\mathbf{P}}) = 0 \\ \Leftrightarrow & [{}^s\mathbf{u}]_\times\ {}^s\mathbf{R}_o\,{}^o\dot{\mathbf{P}} = 0 \end{array} \tag{3.72}$$

Going back to the relationship (3.16) and multiplying it by $[{}^s\mathbf{u}]_\times$, we have:

$$[{}^s\mathbf{u}]_\times\ {}^s\dot{\mathbf{P}} = -[{}^s\mathbf{u}]_\times\ \boldsymbol{v} + [{}^s\mathbf{u}]_\times\ [{}^s\mathbf{P}]_\times\,\boldsymbol{\omega} + [{}^s\mathbf{u}]_\times\ {}^s\mathbf{R}_o\,{}^o\dot{\mathbf{P}} \tag{3.73}$$

taking then into account the constraint (3.72), yields:

$$[{}^s\mathbf{u}]_\times\ {}^s\dot{\mathbf{P}} = -[{}^s\mathbf{u}]_\times\ \boldsymbol{v} + [{}^s\mathbf{u}]_\times\ [{}^s\mathbf{P}]_\times\,\boldsymbol{\omega} \tag{3.74}$$

since $[{}^s\mathbf{u}]_\times$ is a skew symmetric matrix, its rank is equal to two ($\text{rank}([{}^s\mathbf{u}]_\times) = 2$), which represents the number of independent equations at the left of the above system. Therefore the number of independent equations is equal to the number of unknowns ${}^s\dot{\mathbf{P}} = (\dot{x}, \dot{y}, 0)$, which finally leads to the unique following solution $(\dot{x}, \dot{y})$:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{bmatrix} -1 & 0 & \frac{u_x}{u_z} \\ 0 & -1 & \frac{u_y}{u_z} \end{bmatrix} \boldsymbol{v} + \begin{bmatrix} \frac{u_x}{u_z}\,y & -\frac{u_x}{u_z}\,x & y \\ \frac{u_y}{u_z}\,y & -\frac{u_y}{u_z}\,x & -x \end{bmatrix} \boldsymbol{\omega} \tag{3.75}$$

This result we obtain is identical to that given in [44].

3.8 Conclusion

In this chapter we have modeled the exact analytical form (3.34) of the interaction matrix that relates the image moments time variation to the velocity of a 2D ultrasound probe carried by a general 6 DOFs robot arm. To do that, we have developed new theoretical foundations that analytically states the image points velocity as function of the probe velocity, as given by (3.27). We recall that the image velocity was modeled for the points lying only on the image contour, that we denoted $\mathcal{C}$, and not for all the points lying on the whole image section, that we denoted $\mathcal{S}$. Thanks to the Green's theorem, the image

moments time variation was formulated as function of only the image contour points, whose developed image velocity relationship was used to analytical derive that of image moments time variation. The obtained relationships represent the interaction matrix $\mathbf{L}_{m_{ij}}$. We noted that three elements of the interaction matrix that relate the probe-in-plane motions require information only from the observed image, that are image moments. In contrast, the remaining three elements that relate the probe-out-of-plane motions also require the knowledge of normal vector $\nabla\mathbf{F}$ to the object surface at each of the contour points. Finally, the modeling method we proposed is valid for general shaped objects.

We tested this general result in the case where a 2D ultrasound probe is interacting with a spherical object, a cylindrical object, or a 3-D straight line-shaped wire. We have obtained a simplified form (3.52) of the interaction matrix for the case of the spherical object. Moreover, we have theoretically validated the correctness of an element that relate the area time variation to probe velocity. This was achieved by calculating that element with another modeling approach, suitable for that case. Applying the general method on cylindrical objects, we have found that an element, of the interaction matrix, that relates the image area time variation to the probe velocity, is null. This theoretical result also validates the modeling approach of this thesis.

Chapter 4

Normal vector on-line estimation

In the previous chapter we have modeled the analytical form of the interaction matrix (3.34) that relates the image moments time variation to the probe velocity. It was noted that the normal vector to the surface of the observed object, at each point lying on the contour of the image section, appears in this matrix. This normal vector could be derived if a pre-operative 3-D model of the object is available. That would also necessitate a difficult step to localize the object frame with respect to the sensor frame (the probe frame in this case). In this chapter, we propose efficient methods to estimate on-line the normal vector, and thus bypass and overcome those limitations imposed by any pre-operative model. These methods can valuably endow the robotic system with the capability of automatically interacting with objects without any prior knowledge of their shape, 3-D parameters, nor their 3-D location (pose). They are discriminated according to the geometrical primitives considered to estimate the normal vector. We propose to separately use straight line, curved line, and quadric surface primitives.

Let point $\mathbf{P}$ lie in the 2D ultrasound image. More particularly, let this point belong to contour $\mathcal{C}$ of image section $\mathcal{S}$. Consequently, this point lies on the surface of observed object $\mathcal{O}$. The objective is in fact to obtain the vector normal to the object surface at point $\mathbf{P}$ (see Fig. 4.1). We already denoted this vector by $\nabla\mathbf{F}$ in the previous chapter.

4.1 On-line estimation methods based on lines

Let vector $\mathbf{d_i}$ be tangent to the object surface at considered point $\mathbf{P}$, such that it belongs to the probe beam (image), as can be seen on Fig. 4.1. Let vector $\mathbf{d_t}$ be also tangent to the object surface at $\mathbf{P}$ but, in contrast to $\mathbf{d_i}$, it does not lie in the probe observation plane. Therefore, performing the vector cross-product on these two vectors clearly gives vector $\nabla\mathbf{F}$

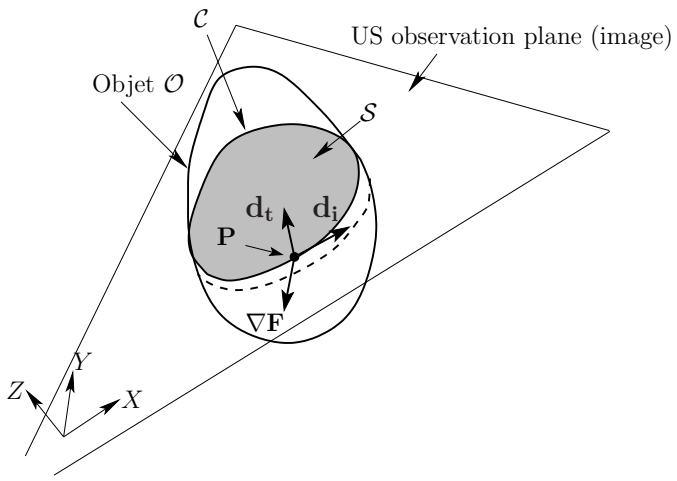


Figure 4.1: Normal vector to the object surface, along with two tangent vectors $\mathbf{d_i}$ and $\mathbf{d_t}$, at point $\mathbf{P}$.

(see Fig. 4.1). This is formulated as follows:

$$^s\nabla\mathbf{F} = {}^s\mathbf{d_i} \times {}^s\mathbf{d_t} \tag{4.1}$$

where $^s\nabla\mathbf{F}$, $^s\mathbf{d_i}$, and $^s\mathbf{d_t}$ are the expressions in probe frame $\{R_s\}$ of vectors $\nabla\mathbf{F}$, $\mathbf{d_i}$, and $\mathbf{d_t}$, respectively. Note that we are interested only in the direction of $^s\nabla\mathbf{F}$, not its amplitude. Indeed, the interaction matrix is not affected by the amplitude of $^s\nabla\mathbf{F}$ but only its direction, as already shown in Chapter 3. This said, we can set $^s\nabla\mathbf{F}$ to a unitary vector.

Since vector $^s\mathbf{d_i}$ lies in the probe observation plane and is, moreover, expressed in the probe frame, it can be extracted from the observed 2D ultrasound image. It is however not the case for vector $^s\mathbf{d_t}$. Indeed, this vector does not lie in the probe observation plane, and therefore can not be extracted from solely the observed image. We need therefore to estimate $^s\mathbf{d_t}$ in order to obtain $^s\nabla\mathbf{F}$. Such an estimation seems efficient since we do not have to estimate in whole the normal vector but only a part of it, which is vector $^s\mathbf{d_t}$, since its second part, vector $^s\mathbf{d_i}$, is already available. In this section, we present two methods to estimate $^s\mathbf{d_t}$. The principle consists in making use of the successive acquired 2D ultrasound images to estimate 3-D lines that are tangent to the surface of the observed object. The estimation is performed for each point of the image contour. It is subsequently used to extract an estimate of vector $^s\mathbf{d_t}$, tangent at each of those points. As presented in what follows, the first method is based on the estimation of 3-D straight lines, while the second method, even

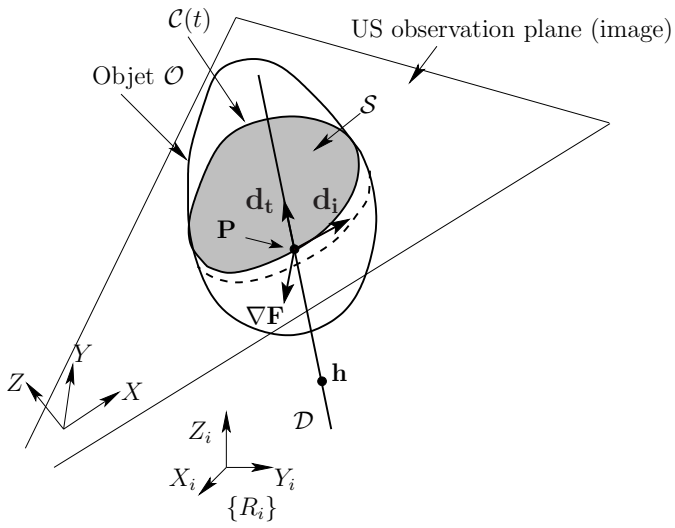


Figure 4.2: Straight line tangent to the object surface at point $\mathbf{P}$.

if it has the same concept as the first one, is based on estimating 3-D curved lines. Note that the methods are described for estimating ${}^s\mathbf{d_t}$ at one point $\mathbf{P}$, but the same principle is applied for all the other image contour points; since the interaction matrix requires ${}^s\nabla\mathbf{F}$ at each of the contour points.

4.1.1 Straight line-based estimation method

Let 3-D straight line $\mathcal{D}$ be tangent to the object surface at point $\mathbf{P}$ (see Fig. 4.2). It is assumed not lying within the probe observation plane. Since both $\mathcal{D}$ and $\mathbf{d_t}$ are tangent to the object surface and do not lie within the image plane, we can set that the direction of $\mathcal{D}$ in nothing but vector $\mathbf{d_t}$ we want to estimate, as shown in Fig. 4.2. We thus propose in this section to estimate $\mathcal{D}$, from which we then infer ${}^s\mathbf{d_t}$.

Consider that the probe is performing out-of-plane motions while at same time acquiring successive 2D ultrasound images of the considered object. From each of the acquired images, contour $\mathcal{C}$ is extracted. Such contour is subsampled in a set of L points $(\mathbf{P}_1, \cdots, \mathbf{P}_L)$ lying on it. We denote such set by $\mathbf{C_p}$. – In practice, we use around L =400 image points to characterize the contour. – Within the image, these points are arranged such that the first point $\mathbf{P}_1$ intersects the X axis of image frame centered on section $\mathcal{S}$. The remaining points are successively located by traveling around $\mathcal{C}$ in counterclockwise direction, as depicted in Fig. 4.3. We can consider $\mathbf{C_p}$ as a vector whose elements are those contour points. Point $\mathbf{P}$ corresponds to element $\mathbf{P}_i$ of $\mathbf{C_p}$. Let point $\mathbf{P}(t)$ lie on image contour $\mathcal{C}(t)$ at time t,

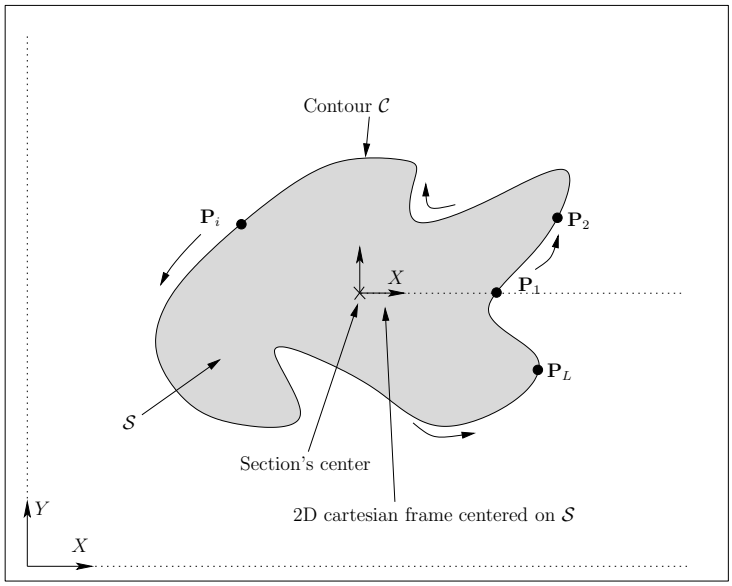


Figure 4.3: Arrangement of image contour points in a set $\mathbf{C_p} = [\mathbf{P}_1, \mathbf{P}_2, \cdots, \mathbf{P}_L]$. The first point $\mathbf{P}_1$ intersects the X axis of the frame centered on image section $\mathcal{S}$. The image 2D cartesian frame is indicated with its (X, Y) axes (bottom left).

and $\mathbf{P}(t+dt)$ lie on subsequent contour $\mathcal{C}(t+dt)$, extracted after the probe had performed a differential out-of-plane motion during a duration dt (see Fig. 4.4). Elements $\mathbf{P}(t)$ and $\mathbf{P}(t+dt)$ have the same index in their respective sets $\mathbf{C_p}(t)$ and $\mathbf{C_P}(t+dt)$ (i. e., $\mathbf{P}(t)$ corresponds to a point $\mathbf{P}_i$ of set $\mathbf{C_p}(t)$, and $\mathbf{P}(t+dt)$ to a point $\mathbf{P}_i$ of $\mathbf{C_p}(t+dt)$, where subscript i is their common index). Note that we assume that number L of points extracted from each of the successive images is constant all along the estimation. A straight line that passes through these two points $\mathbf{P}(t)$ and $\mathbf{P}(t+dt)$ is therefore tangent to the surface of the observed object. Such straight line thus corresponds to $\mathcal{D}$ that we want to estimate (see Fig. 4.4). Theoretically, two points are enough to estimate $\mathcal{D}$. This however is not the case in practice due to different factors. Indeed, due to measurement perturbations, as instance, the recorded points could be either too close to each other or, inversely, misaligned. Either configuration would lead to a wrong estimation of $\mathcal{D}$. That is the reason why we use in the estimation more points; respectively extracted from the successive images.

Considering successively extracted points from the succeeding acquired images, where these points have a same index in their respective set $\mathbf{C_P}$, the estimation principle then consists in fitting them with straight line $\mathcal{D}$. The latest extracted point corresponds to $\mathbf{P}$, at which $\mathcal{D}$ should be tangent to the object surface. To do so, the estimation is also performed by assigning different weights to the extracted points, such that the current (new) point $\mathbf{P}$ is

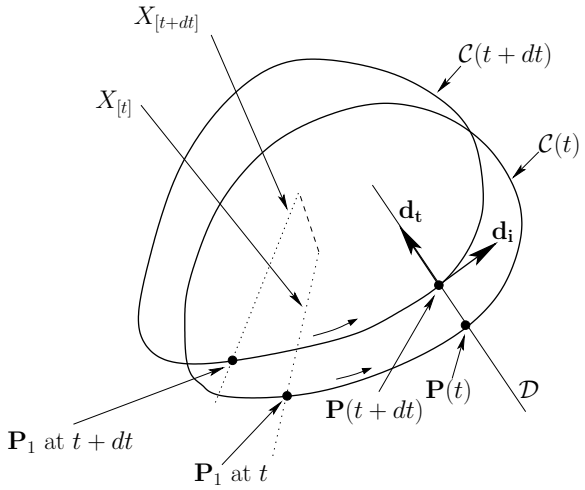


Figure 4.4: Image contour 3D evolution with a corresponding tangent straight line $\mathcal{D}$. In the 3-D space, the contour lies on the surface of the observed object. Contour $\mathcal{C}(t)$ is extracted from the ultrasound image at time t, while $\mathcal{C}(t+dt)$ is extracted at time $t+dt$ after the probe had performed an out-of-plane motion. Contour's first point $\mathbf{P_1}$ at time t and that at time $t+dt$ are indicated. They correspond to the intersection of X axis of the centered image frame with the contour, respectively at time t and at time $t+dt$. The X axis of time t is denoted $X_{[t]}$, while that of time $t+dt$ is denoted $X_{[t+dt]}$.

assigned with the highest weight; the values of the different assigned weights are arranged in a decreasing fashion. In fact, each new extracted point, along with its assigned highest weight, updates the estimation of $\mathcal{D}$ in such a way this latter adjusts its orientation to become tangent to the object surface at $\mathbf{P}$ (see Fig. 4.5). In other words, consider the two successive points $\mathbf{P}_{[k-1]}$ and $\mathbf{P}_{[k]}$ respectively acquired at the precedent sample time $k-1$ and the current one k (note that k refers to time t in the discrete domain). At time $k-1$, $\mathcal{D}$ is considered already estimated to be tangent to the surface at $\mathbf{P}_{[k-1]}$. The objective is to re-estimate D in such a way it becomes tangent at $\mathbf{P}_{[k]}$. The latter, along with its assigned highest weight, leads $\mathcal{D}$ adjusting its orientation to become tangent to the surface at $\mathbf{P}_{[k]}$.

Let $\{R_i\}$ be a 3-D cartesian frame in which $\mathcal{D}$ is estimated (see Fig. 4.2). It corresponds to initial probe frame $\{R_s(t_0)\}$. Let point $\mathbf{h}$ lie on $\mathcal{D}$ (see Fig. 4.2). Its expression in $\{R_i\}$ is ${}^i\mathbf{h}$. A point $\mathbf{P}$ that lies on $\mathcal{D}$ satisfies the following relationship:

$$\left({}^i\mathbf{h} - {}^i\mathbf{P}\right) \times {}^i\mathbf{d_t} = 0 \tag{4.2}$$

where ${}^i\mathbf{P} = ({}^ix,\ {}^iy,\ {}^iz)$ and ${}^i\mathbf{d_t} = (d_x,\ d_y,\ d_z)$ are the expressions of respectively $\mathbf{P}$ and $\mathbf{d_t}$ in frame $\{R_i\}$. 3-D coordinates vector ${}^i\mathbf{P}$ is obtained from image coordinates ${}^s\mathbf{P} = (x, y, 0)$ using the robot odometry, according to the classical relationship (A.9) by ${}^i\mathbf{P} = {}^s\mathbf{R}_i^\top\ ({}^s\mathbf{P} - {}^s\mathbf{t}_i)$. Rotation matrix ${}^s\mathbf{R}_i$ and translation vector ${}^s\mathbf{t}_i$ are obtained from the robot odometry. They define respectively the orientation and the origin of $\{R_i\}$ with respect to $\{R_s\}$. The above relationship can be formulated in its minimal form as follows:

$$\left\{ \begin{array}{rcl} {}^ix & = & \eta_1\, {}^iz + \eta_0 \\ {}^iy & = & \tau_1\, {}^iz + \tau_0 \end{array} \right. \tag{4.3}$$

where $\eta_1 = d_x/d_z$ and $\tau_1 = d_y/d_z$ are 3-D parameters representing the orientation of $\mathcal{D}$. The elements η_0 and τ_0 are also 3D parameters, but are moreover related to the location of $\mathcal{D}$ since they are function of both ${}^i\mathbf{d_t}$ and ${}^i\mathbf{P}_0$. Vector ${}^i\mathbf{d_t}$ can be expressed as:

$${}^i\mathbf{d_t} = d_z \begin{pmatrix} d_x/d_z \\ d_y/d_z \\ 1 \end{pmatrix} = d_z \begin{pmatrix} \eta_1 \\ \tau_1 \\ 1 \end{pmatrix} \tag{4.4}$$

The direction of a vector cross-product, as that of the relationship (4.1), is affected solely by the direction of the vectors and not their amplitude. Therefore, we only need to estimate the direction of $\mathbf{d_t}$. This comes to estimate parameters η_1 and τ_1 since they represent its orientation, as can be seen from the above relationship. The model used for the estimation is the relationship (4.3), where coordinates $({}^ix,\ {}^iy,\ {}^iz)$ are the input information while $\Theta = (\eta_1,\ \tau_1,\ \eta_0,\ \tau_0)$ is the vector to estimate.

The system (4.3) can be formulated as follows:

$$\mathbf{Y} = \mathbf{\Phi}^\top \Theta \tag{4.5}$$

where

$$\mathbf{Y} = ({}^ix,\ {}^iy) \ \text{ and } \mathbf{\Phi}^\top = \begin{pmatrix} {}^iz & 0 & 1 & 0 \\ 0 & {}^iz & 0 & 1 \end{pmatrix} \tag{4.6}$$

We propose to use a stabilized recursive least-squares algorithm [43]. The principle consists in finding an estimate $\hat{\Theta}$ of vector Θ that minimizes the following quadric sum $J(\hat{\Theta}_{[k]})$ of the residual errors:

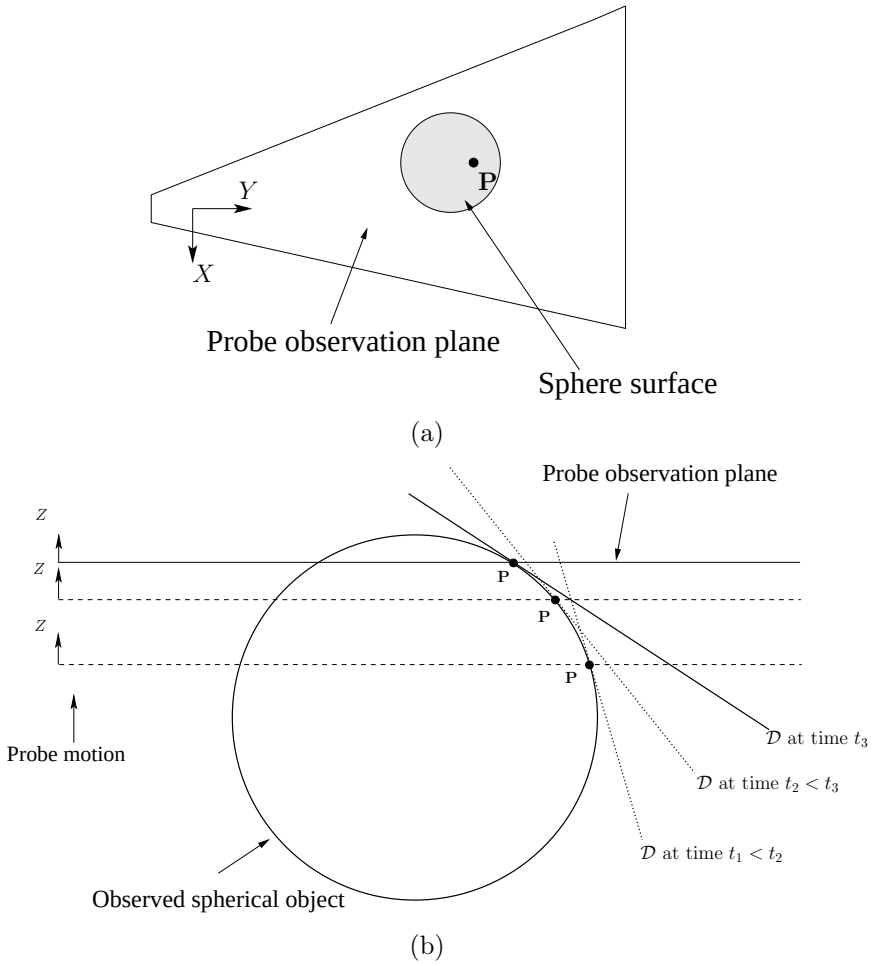


Figure 4.5: Evolution of estimated 3-D straight line for the case a 2D ultrasound probe is interacting with a spherical object. (a) Upper sight showing point **P** lying on the sphere surface, both being observed by the probe planar beam - (b) Transverse sight: the probe is performing an out-of-plane motion by moving along its Z axis, while at the same time straight line $\mathcal{D}$ continually adjusts its orientation to remain tangent to the sphere surface at **P**.

$$J(\hat{\Theta}_{[k]}) = \sum_{i=t_0}^{k} \beta^{(i-t_0)} \left(\mathbf{Y}_{[i]} - \mathbf{\Phi}_{[i]}^{\top} \hat{\Theta}_{[i]}\right)^{\top} \left(\mathbf{Y}_{[i]} - \mathbf{\Phi}_{[i]}^{\top} \hat{\Theta}_{[i]}\right) \tag{4.7}$$

where $\hat{\Theta}_{[i]}$ is the estimate and $(\mathbf{Y}_{[i]},\ \mathbf{\Phi}_{[i]})$ are the measures at sample time i. The scalar $\beta \in]0,\ 1]$ is a forgetting factor assigned to the estimation errors $\mathbf{Y}_{[i]} - \mathbf{\Phi}_{[i]}^{\top} \hat{\Theta}_{[i]}$. It is employed to give highest weights to the newly recorded measures. Vector $\hat{\Theta}_{[k]}$ minimizing J is expressed in a recursive form as function of the current measures $(\mathbf{Y}_{[k]},\ \mathbf{\Phi}_{[k]})$ and the precedent estimate $\hat{\Theta}_{[k-1]}$, as follows [43]:

$$\hat{\Theta}_{[k]} = \hat{\Theta}_{[k-1]} + \mathbf{F}_{[k]}\, \mathbf{\Phi}_{[k]} \left(\mathbf{Y}_{[k]} - \mathbf{\Phi}_{[k]}^{\top} \hat{\Theta}_{[k-1]}\right) \tag{4.8}$$

where $\mathbf{F}_{[k]}$ represents the covariance matrix at time k. It is given by the following relationship, also recursive:

$$\mathbf{F}_{[k]}^{-1} = \beta\, \mathbf{F}_{[k-1]}^{-1} + \mathbf{\Phi}_{[k]}\, \mathbf{\Phi}_{[k]}^{\top} + (1-\beta)\ \beta_0\, \mathbf{I}_4 \tag{4.9}$$

where $\mathbf{I}_4$ is the 4×4 identity matrix. Its dimension refers to the four parameters of Θ to estimate. The term $(1-\beta)\ \beta_0\, \mathbf{I}_4$ corresponds to a stabilization element. It is added in order to prevent the matrix $\mathbf{F}_{[t]}^{-1}$ becoming ill-conditioned. The latter might occur when there is not enough excitation in the input information $(\mathbf{Y},\ \mathbf{\Phi})$. This is mainly caused by lack of probe out-of-plane motions. The algorithm is initialized by setting $\mathbf{F}_{[t_0]} = f_0\, \mathbf{I}_4$, with $f_0 \in]0,\ 1/\beta_0]$, and $\Theta_{[t_0]} = \Theta_0$, where Θ_0 might be arbitrarily selected. However, in order to obtain initial estimate Θ_0 that is expected closer to the actual parameters Θ and thus yielding the estimation more faster, we use another different algorithm to estimate Θ_0 as presented in Section 4.3.

Then, estimate ${}^{i}\hat{\mathbf{d}}_{\mathbf{t}}$ of tangent vector ${}^{i}\mathbf{d}_{\mathbf{t}}$ can be derived after obtaining estimate $\hat{\Theta} = (\hat{\eta}_1,\ \hat{\tau}_1,\ \hat{\eta}_0,\ \hat{\tau}_0)$ and replacing the first two parameters $(\hat{\eta}_1,\ \hat{\tau}_1)$ in (4.4). We can obtain it as an unitary vector as follows:

$${}^{i}\hat{\mathbf{d}}_{\mathbf{t}} = (\hat{\eta}_1, \hat{\tau}_1, 1)/\|(\hat{\eta}_1,\ \hat{\tau}_1,\ 1)\| \tag{4.10}$$

Its expression ${}^s\hat{\mathbf{d}}_\mathbf{t}$ in the probe frame can be obtained using the rotation matrix ${}^s\mathbf{R}_i$. As already said, this matrix is obtained using the robot odometry. Thus, the estimate ${}^s\hat{\mathbf{d}}_\mathbf{t}$ of ${}^s\mathbf{d}_\mathbf{t}$ is obtained as follows:

$$ {}^s\hat{\mathbf{d}}_\mathbf{t} = {}^s\mathbf{R}_i \, {}^i\hat{\mathbf{d}}_\mathbf{t} \tag{4.11} $$

Finally, replacing this result in the relationship (4.1), normal vector ${}^s\nabla\mathbf{F}$ is estimated.

Recall that the estimation method we presented in this section is described to estimate the normal vector for only one point $\mathbf{P}$ lying on image contour $\mathcal{C}$. It is in fact applied for all the points extracted from the contour.

4.1.2 Curved line-based estimation method

Although the above presented method of using 3-D straight lines to estimate the tangent vector presents some advantages as the shortened processing time, since only four parameters are estimated, it however heavily relies on the assigned weights as means to adjust the orientation of the straight line, in such a way this latter becomes tangent to the object surface. To improve this, we present in this section an estimation method based on 3-D curved lines instead of straight ones (see Fig. 4.6). This has the advantage to deal more effectively with the curvature of the observed object, if curvature there is. Tangent vector $\mathbf{d_t}$ to the object surface can then be simply obtained as the tangent to the estimated curve.

Let $\mathcal{K}$ denote the tangent curve to estimate. Its analytical model, stating the constraint that any point $\mathbf{P}$ lying on it must satisfy, can be formulated as follows:

$$ \begin{cases} {}^ix &= \eta_2 \, {}^iz^2 + \eta_1 \, {}^iz + \eta_0 \\ {}^iy &= \tau_2 \, {}^iz^2 + \tau_1 \, {}^iz + \tau_0 \end{cases} \tag{4.12} $$

where ${}^i\mathbf{P} = ({}^ix,\ {}^iy,\ {}^iz)$ is the expression of point $\mathbf{P}$ in frame $\{R_i\}$. Elements $\eta_{p|_{p=\overline{0,2}}}$ and $\tau_{q|_{q=\overline{0,2}}}$ are 3-D parameters representing the shape of $\mathcal{K}$.

Since $\mathcal{K}$ is considered tangent to the object surface at $\mathbf{P}$, its tangent vector at that point is nothing but vector $\mathbf{d_t}$ we want to estimate. We can formulate such vector by:

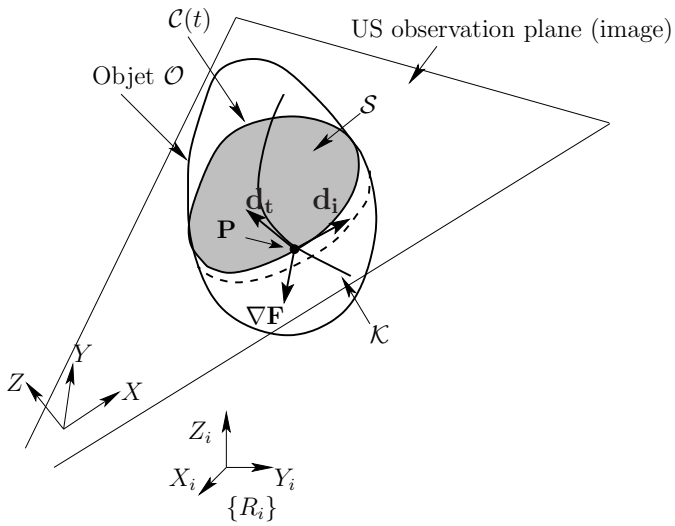


Figure 4.6: Curved line $\mathcal{K}$ tangent to the object surface at point $\mathbf{P}$.

$$ {}^i\mathbf{d_t} = \begin{bmatrix} \partial\,{}^i x/\partial\,{}^i z \\ \partial\,{}^i y/\partial\,{}^i z \\ \partial\,{}^i z/\partial\,{}^i z \end{bmatrix} = \begin{bmatrix} \partial\,{}^i x/\partial\,{}^i z \\ \partial\,{}^i y/\partial\,{}^i z \\ 1 \end{bmatrix} \tag{4.13} $$

Applying this on the relationship (4.12), the tangent vector is expressed as follows:

$$ {}^i\mathbf{d_t} = \begin{bmatrix} 2\,\eta_2\,{}^i z + \eta_1 \\ 2\,\tau_2\,{}^i z + \tau_1 \\ 1 \end{bmatrix} \tag{4.14} $$

Coordinate ${}^i z$ of $\mathbf{P}$ is considered available, after the point would have been extracted from the image and then expressed in frame $\{R_i\}$ thanks to the robot odometry. We therefore need to obtain an estimate of the parameters $(\eta_2, \tau_2, \eta_1, \tau_1)$ which then would yield that of $\mathbf{d_t}$. The model on which the estimation is based is that given by (4.12), which expresses the constraint satisfied by any point lying on $\mathcal{K}$. The input information feeding the estimation are coordinates $({}^i x,\ {}^i y,\ {}^i z)$, while the parameters vector to estimate is $\Theta = (\eta_2,\ \tau_2,\ \eta_1,\ \tau_1,\ \eta_0,\ \tau_0)$. The curve model (4.12) can then be re-formulated in an expression as that of (4.5), but with:

$$ \mathbf{Y} = \begin{bmatrix} {}^i x \\ {}^i y \end{bmatrix} \quad \text{and} \quad \mathbf{\Phi}^\top = \begin{bmatrix} {}^i z^2 & 0 & {}^i z & 0 & 1 & 0 \\ 0 & {}^i z^2 & 0 & {}^i z & 0 & 1 \end{bmatrix} \tag{4.15} $$

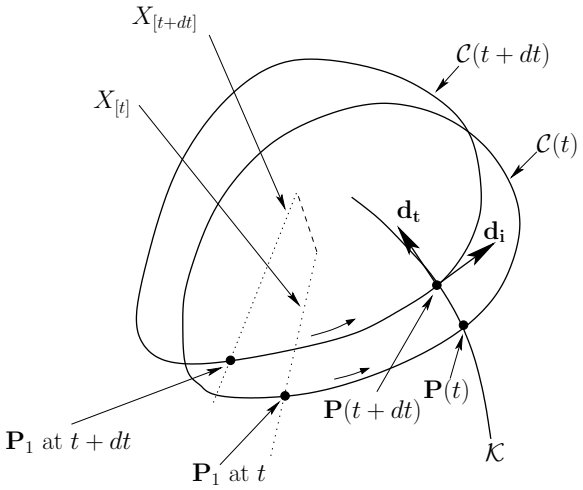


Figure 4.7: Contour 3-D evolution with its tangent curve.

We propose to use again the stabilized least-squares recursive algorithm [43] to perform the estimation. It has already been introduced in Section 4.1.1, where estimate $\hat{\Theta}_{[k]}$ of Θ at current sample time k is given by the recursive expression (4.8), but covariance matrix $\mathbf{F}_{[k]}$ at time k is now given by the following recursive expression:

$$\mathbf{F}_{[k]}^{-1} = \beta\,\mathbf{F}_{[k-1]}^{-1} + \mathbf{\Phi}_{[k]}\,\mathbf{\Phi}_{[k]}^{\top} + (1-\beta)\,\beta_0\,\mathbf{I}_6 \tag{4.16}$$

where the 6×6 identity matrix $\mathbf{I}_6$ is employed, instead of $\mathbf{I}_4$ used in (4.9), since in this case the size of parameters vector Θ becomes equal to six.

The estimation principle is similar to that of the straight line case. Each new extracted point updates the algorithm with its coordinates involved in the input variables $\mathbf{Y}$ and $\mathbf{\Phi}$. In this case, in fact, curve $\mathcal{K}$ has to fit points extracted from the successive acquired images (see Fig. 4.7). These points have a same index in their corresponding set $\mathbf{C_P}$, similarly as for the straight line estimation. A forgetting factor β is used to infer different weights assigned to these extracted points in such a way to take more into account the recently extracted points. This has the advantage to perform a local estimation, and thus yielding the estimation more accurate and robust, since $\mathcal{K}$ is restrained to fit only a local surface (see Fig. 4.8). Indeed, more β is smaller, for example, less previous points are taken into account in the estimation. The effect is like that of an estimation performed over a window of data information, thus allowing the curve more adapting to the object surface and sparing it the

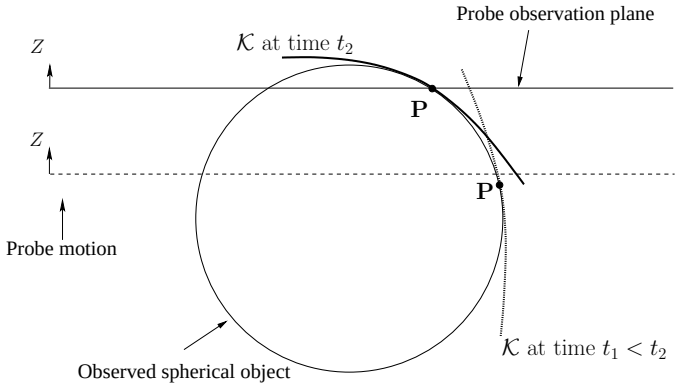


Figure 4.8: Transverse sight showing the evolution of the estimated 3-D curved line $\mathcal{K}$ at point $\mathbf{P}$, for the case a 2D ultrasound probe is interacting with a spherical object. The objective is to estimate $\mathcal{K}$ in such a way this latter would be tangent to the object surface at $\mathbf{P}$. The upper sight is similar to that depicted on Fig. 4.5(a).

effect of far points, that could compromise the estimation. Indeed, only one curve might not be sufficient to fit both those far points and recent ones.

Estimate $\hat{\Theta} = (\hat{\eta}_2,\ \hat{\tau}_2,\ \hat{\eta}_1,\ \hat{\tau}_1,\ \hat{\eta}_0,\ \hat{\tau}_0)$ of Θ being obtained, that of ${}^i\mathbf{d_t}$ can then be derived by replacing the result in (4.14), that we set as a unitary vector by:

$${}^i\hat{\mathbf{d}}_\mathbf{t} = \begin{bmatrix} 2\,\hat{\eta}_2\,{}^iz + \hat{\eta}_1 \\ 2\,\hat{\tau}_2\,{}^iz + \hat{\tau}_1 \\ 1 \end{bmatrix} \Big/ \left\| \begin{matrix} 2\,\hat{\eta}_2\,{}^iz + \hat{\eta}_1 \\ 2\,\hat{\tau}_2\,{}^iz + \hat{\tau}_1 \\ 1 \end{matrix} \right\| \tag{4.17}$$

whose expression in $\{R_s\}$ is derived by ${}^s\hat{\mathbf{d}}_\mathbf{t} = {}^s\mathbf{R}_i\,{}^i\hat{\mathbf{d}}_\mathbf{t}$. Finally, the estimate of normal vector $\nabla\mathbf{F}$ is obtained by substituting ${}^s\mathbf{d_t}$ with its estimate ${}^s\hat{\mathbf{d}}_\mathbf{t}$ in (4.1).

4.2 Quadric surface-based estimation method

In this section, we propose to estimate the local surface of the considered object and then use it to derive an estimate of $\nabla\mathbf{F}$.

The estimation consists in fitting a quadric surface to a cloud of points of the object considered local surface, i. e., fitting a quadric to local surface of the object (see Fig. 4.9). The points are obtained from the successive contours $\mathcal{C}$ extracted from succeeding acquired 2D

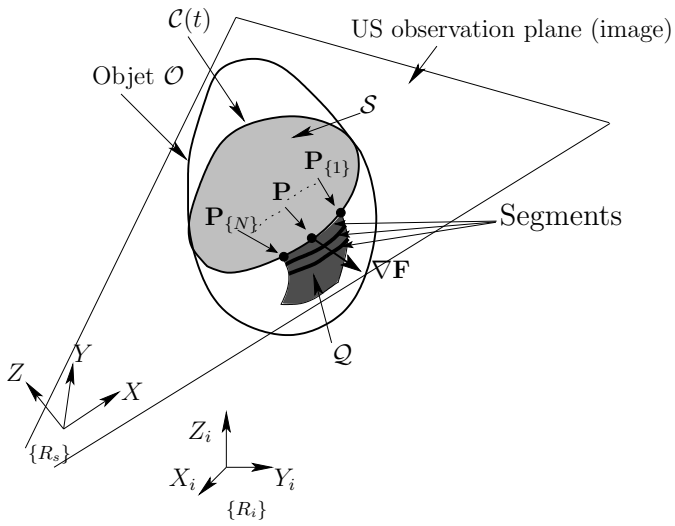


Figure 4.9: Quadric surface $\mathcal{Q}$ that, ideally, should exactly fit the local surface surrounding point $\mathbf{P}$. The current observed segment is also shown, where its starting point $\mathbf{P}_{\{1\}}$ and ending one $\mathbf{P}_{\{N\}}$ are indicated on it. The segment is centered on $\mathbf{P}$.

ultrasound images.

Let $\mathcal{Q}$ be a quadric surface (see Fig. 4.9). Any point ${}^i\mathbf{P} = ({}^ix,\ {}^iy, {}^iz)$ lying on $\mathcal{Q}$ satisfies the following constraint:

$$\begin{aligned} F({}^ix,\ {}^iy,\ {}^iz) \quad = \quad & \gamma_{20}\,{}^ix^2 + \gamma_{02}\,{}^iy^2 + \gamma_{11}\,{}^ix\,{}^iy \\ & + \gamma_{10}\,{}^ix + \gamma_{01}\,{}^iy + \gamma_{00}\,{}^iz - 1 = 0 \end{aligned} \tag{4.18}$$

where $\gamma_{pq|_{p,q=\overline{0,2}}}$ are 3-D parameters representing the shape of quadric surface $\mathcal{Q}$.

The objective is to estimate parameters γ_{pq} using the cloud of points lying on the local surface surrounding point $\mathbf{P}$ (see Fig. 4.9). Let $\mathbf{P}_{\{j\}|_{j=\overline{1,N}}}$ be points lying on contour $\mathcal{C}$, such that $\mathbf{P}_{\{j\}}$ is adjacent to $\mathbf{P}_{\{j+1\}}$ and that $\mathbf{P} = \mathbf{P}_{\{(N+1)/2\}}$ (see Fig. 4.9). The set of points $\mathbf{P}_{\{j\}|_{j=\overline{1,N}}}$ in fact defines a segment that is centered on $\mathbf{P}$. Note that these points, and thus the corresponding segment, are nothing but part of set $\mathbf{C_p}$, previously defined in Section 4.1.1 (see Fig. 4.3). Similarly to the principle used in the cases of the straight and the curved line, each two successive points $\mathbf{P}_{\{j\}[k]}$ and $\mathbf{P}_{\{j\}[k-1]}$, extracted respectively from the image acquired at time k and that acquired at precedent sample time $k-1$, have the same index in their corresponding vector $\mathbf{C_p}$. Within their respective segment, their position is indicated with subscript j. The estimation we propose uses the successively

acquired segments to estimate quadric $\mathcal{Q}$ that best fits the cloud of points extracted from those segments. The parameters vector to estimate is $\Theta = (\gamma_{20},\ \gamma_{02},\ \gamma_{11},\ \gamma_{10},\ \gamma_{01},\ \gamma_{00})$. Point $P_{\{j\}[k]}$ is assumed lying on $\mathcal{Q}$, then satisfies the constraint (4.18), which can be reformulated as follows:

$$\mathbf{Y}_j = \mathbf{\Phi}_j^\top \Theta \tag{4.19}$$

with :

$$\mathbf{Y}_j = 1 \text{ and } \mathbf{\Phi}_j^\top = \left[{}^i x_j^2,\ \ {}^i y_j^2,\ \ {}^i x_j \times {}^i y_j,\ \ {}^i x_j,\ \ {}^i y_j,\ \ {}^i z_j\right]^\top \tag{4.20}$$

where $\mathbf{P}_{\{j\}} = ({}^i x_j,\ {}^i y_j,\ {}^i z_j)$ is the expression of point $\mathbf{P}_{\{j\}}$ in frame $\{R_i\}$. Applying the formulation (4.20) on all points $\mathbf{P}_{\{j\}|_{j=\overline{1,N}}}$ of the contour segment, then stacking the obtained constraint relationships, yields:

$$\begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \vdots \\ \mathbf{Y}_N \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}_1^\top \\ \mathbf{\Phi}_2^\top \\ \vdots \\ \mathbf{\Phi}_N^\top \end{bmatrix} \Theta \tag{4.21}$$

that can be formulated $\mathbf{Y} = \mathbf{\Phi}^\top \Theta$ as (4.5), but with:

$$\mathbf{Y} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \text{ and } \mathbf{\Phi}^\top = \begin{bmatrix} \mathbf{\Phi}_1^\top \\ \mathbf{\Phi}_2^\top \\ \vdots \\ \mathbf{\Phi}_N^\top \end{bmatrix} \tag{4.22}$$

where $\mathbf{Y}$ and $\mathbf{\Phi}^\top$ are of dimension N and $N \times 6$, respectively. We recall that N corresponds to the width of a contour segment (i. e., number of points lying on the contour segment). The relationship (4.5) according to (4.22) states the constraint satisfied by the contour segment centered on $\mathbf{P}$. When the 2D ultrasound probe performs out-of-plane motions while at the same time acquiring successive 2D ultrasound images, successive segments are extracted. Those segments represent the cloud of points lying on the object local surface surrounding $\mathbf{P}$. We propose to use again the stabilized least-squares recursive algorithm, that gives the estimate $\hat{\Theta}$ of Θ by the recursive relationship (4.8), and its involved covariance matrix by

(4.16). Each current (observed) segment updates the estimation algorithm. The assigned forgetting factor β enables to take more into account the recently acquired segments and thus prevent the *"old"* segments compromising the estimation. To illustrate this, if for example the 2D ultrasound probe performed a complete scan of the observed object by sweeping it and at the same time acquiring its images successively, then a 3D volume of the object is made up, it is unlikely that one quadric might be sufficient to fit the whole surface of that constructed volume. That is the reason why an estimated local surface is expected to relatively fit well the object surface in the neighborhood of a considered point, at which vector $\nabla\mathbf{F}$ is expected to be normal.

$\nabla\mathbf{F}$ can be analytically expressed from the quadric surface relationship, using the following classical formula:

$$ {}^{i}\nabla\mathbf{F} = \begin{bmatrix} \partial F/\partial {}^{i}x \\ \partial F/\partial {}^{i}y \\ \partial F/\partial {}^{i}z \end{bmatrix} \tag{4.23} $$

where ${}^{i}\nabla\mathbf{F}$ is the expression of $\nabla\mathbf{F}$ in $\{R_i\}$. Thus applying (4.23) on (4.18) yields:

$$ {}^{i}\nabla\mathbf{F} = \begin{bmatrix} 2\,\gamma_{20}\,{}^{i}x + \gamma_{11}\,{}^{i}y + \gamma_{10} \\ 2\,\gamma_{02}\,{}^{i}y + \gamma_{11}\,{}^{i}x + \gamma_{01} \\ \gamma_{00} \end{bmatrix} \tag{4.24} $$

Replacing estimated parameters $\hat{\Theta}$ in the above relationship, estimate ${}^{i}\nabla\hat{\mathbf{F}}$ of ${}^{i}\nabla\mathbf{F}$ is obtained. Then, using rotation matrix ${}^{s}\mathbf{R}_i$ that defines the orientation of $\{R_i\}$ with respect to frame $\{R_s\}$, the desired estimate ${}^{s}\nabla\hat{\mathbf{F}}$ of normal vector ${}^{s}\nabla\mathbf{F}$ is finally obtained as follows:

$$ {}^{s}\nabla\hat{\mathbf{F}} = {}^{s}\mathbf{R}_i \; {}^{i}\nabla\hat{\mathbf{F}} \tag{4.25} $$

4.3 Sliding least squares estimation algorithm

We presented above three methods to estimate the normal vector. Both of these methods employ a recursive algorithm to perform the estimation online. Such algorithm requires an initial vector parameters Θ_0 to start the recursive estimation. If these initial parameters are

far from the actual ones, the recursive algorithm would take relatively large duration before estimate $\hat{\Theta}$ becomes closer to actual one Θ. This would undoubtedly be reflected on the visual servoing performances, where the commands are sent at a real-time streaming rate to the robot. Indeed, the control command depends on the normal vector, as we will see in Chapter 5. If this vector is not well estimated, for a large time duration, thus the command would be yielded erroneous. That is the reason why it is necessary to obtain a relatively good estimate in the first few iterations of the servoing, or before the servoing is launched. To do so, we propose to first perform an estimation directly on a window (set) of recorded measurements. We propose for that to use a Sliding Least Squares (SLS) algorithm [22]. We apply it only at the beginning for first iterations. Right after, the recursive algorithm will then take place during all the estimation. The SLS algorithm, in fact, behaves similarly to the Non-Recursive least squares one. Its particularity is that it tends to take into account in the estimation only the part of the measurement that conveys wealthy information.

Consider different measurements $\mathbf{Y}_{[i]}$ and $\mathbf{\Phi}_{[i]}^{\top}$ recorded and saved on a window of N_{LS} size ($i = k - N_{LS} + 1$ up to $i = k$). Their weighted correlations are calculated as follows (see [22]):

$$\mathbf{\Gamma} = \sum_{i=k-N_{LS}+1}^{k} \left(\frac{\beta^{(k-i)}}{m_{[i]}^2} \, \mathbf{\Phi}_{[i]} \, \mathbf{\Phi}_{[i]}^{\top} \right) \tag{4.26}$$

$$\mathbf{w} = \sum_{i=k-N_{LS}+1}^{k} \left(\frac{\beta^{(k-i)}}{m_{[i]}^2} \, \mathbf{\Phi}_{[i]} \mathbf{Y}_{[i]} \right) \tag{4.27}$$

where we recall that β is a forgetting factor assigned to the different measurements, in such a way to take more into account the fresh ones. We set the scalar $m_{[i]}$ as the max norm of the matrix $\mathbf{\Phi}_{[i]} \, \mathbf{\Phi}_{[i]}^{\top}$, that is $m_{[i]} = \| \, \mathbf{\Phi}_{[i]} \, \mathbf{\Phi}_{[i]}^{\top} \, \|_{\max}$. It is employed for normalization between the different measurements. The estimation objective is to obtain an estimate $\hat{\Theta}$ that best fits the model relationship (4.5), for whole of those registered measurements. If the algorithm would have consisted in a weighted non-recursive method, the estimate would be obtained as $\hat{\Theta} = \mathbf{\Gamma}^{-1} \, \mathbf{w}$. However, when the measurements do not convey enough wealthy information, matrix $\mathbf{\Gamma}$ tends to be ill-conditioned. The SLS algorithm instead deals with such eventuality. Its principle consists in processing the valuable parts of the information differently from the other part, that is suspected at the origin of the ill-conditioning. This latter part is detected using the eigenvalues. The discrimination is performed according to a threshold ϵ_0; an eigenvalue, or its normalized value, smaller than the threshold is considered as related to the singularity. More precisely, consider the eigenvalue decomposition of matrix $\mathbf{\Gamma}$, since the latter is symmetric according to (4.26), as follows:

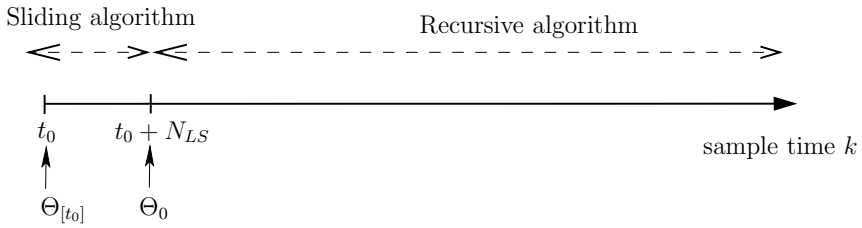


Figure 4.10: Estimation contrivance consisting in applying firstly the sliding algorithm for only the first N_{LS} iterations, and then the recursive algorithm solely throughout the estimation.

$$\mathbf{\Gamma} = \mathbf{Q}\,\Lambda\,\mathbf{Q}^\top \tag{4.28}$$

where diagonal matrix Λ contains eigenvalues $\lambda_{i_{|i=\overline{1,n}}}$. These latter are positive ($\lambda_{i_{|i}} \geqslant 0$), and are arranged in non-increasing fashion, that is, $\lambda_i \geqslant \lambda_{i+1}$. They should be normalized by λ_1. The $n \times n$ matrix $\mathbf{Q}$ is orthogonal ($\mathbf{Q}^{-1} = \mathbf{Q}^\top$), and is given as:

$$\mathbf{Q} = [\mathbf{q}_1\ \mathbf{q}_2 \cdots \mathbf{q}_n] \tag{4.29}$$

where $\mathbf{q}_{i|_i}$ is an $n \times 1$ eigenvector associated to value λ_i. According to the SLS algorithm, estimate $\hat{\Theta}_{[k]}$ at sample time k is thus given by:

$$\hat{\Theta}_{[k]} = \begin{cases} \mathbf{\Gamma}^{-1}\,\mathbf{w} & \text{if} \quad \lambda_n > \epsilon_0 \\ \left(\sum_{i=1}^{l} \frac{1}{\lambda_i}\,\mathbf{q}_i\,\mathbf{q}_i^\top\right)\mathbf{w} \;+\; \left(\sum_{i=l+1}^{N_{LS}} \mathbf{q}_i\,\mathbf{q}_i^\top\right)\hat{\Theta}_{[k-1]} & \text{if} \quad \lambda_l > \epsilon_0 \ \text{ and } \ \lambda_{l+1} \leqslant \epsilon_0 \\ \hat{\Theta}_{[k-1]} & \text{if} \quad \lambda_1 \leqslant \epsilon_0 \end{cases} \tag{4.30}$$

Note that when $\lambda_n > \epsilon_0$ all the other eigenvalues are also larger than ϵ_0, and when $\lambda_1 \leqslant \epsilon_0$ all the remaining eigenvalues are also not larger than ϵ_0.

We recall that our goal is to obtain an initial estimate $\hat{\Theta}$ that is closer to the actual one Θ. For that, we apply the SLS algorithm for only the first N_{LS} iterations to obtain an estimate $\hat{\Theta}_{[t0+N_{LS}-1]}$, that is expected closer to Θ. This estimate is then employed as the initial parameters vector Θ_0 for launching the recursive algorithm; this latter then is applied solely throughout the estimation. This is depicted on Fig. 4.10.

4.4 Simulation results

The methods we developed above are tested in simulations and their performances assessed. These simulations are classified in two distinct sets. In the first simulation trials, each method is applied to estimate its corresponding geometric primitives. These first trials allow to test that the stated primitives can be estimated using the developed methods, and thus to verify the validity of these latter. As for the second set's trials, they are conducted on a simulated ellipsoidal object. This latter is provided from a 3-D mathematical model we designed. Therefore, the surface normal vector can be mathematically derived and its numerical value inferred. Such value serves in fact as ground truth datum. Comparing that obtained value with the estimated ones (separately obtained with each of the three estimation methods), the validity of these methods in estimating the normal vector to the surface of the object, namely the ellipsoid, is verified.

The three estimation methods have been implemented in the C++ programming language. Some of the corresponding arithmetic and matrix operations, as addition and multiplication for example, are coded using the ViSP C++ library [53]. The simulations are performed using a PC computer running LINUX operating system.

4.4.1 Interaction with straight lines

We apply the straight line-based method in estimating simulated 3-D straight lines. To do so, the interaction of a virtual 2D ultrasound probe with three 3-D straight lines is simulated. This interaction is mathematically modeled, from which the intersection of the virtual probe image plane with those lines is derived. This intersection thus results in three image points, whose coordinates are obtained from the mathematical model. We assume the knowledge of the full mathematical model (direction and a belonging 3-D point) of each of those straight lines. We finally compare the actual 3-D parameters of the straight lines with those estimated.

The simulation is conducted by moving with constant velocity the virtual 2D ultrasound probe. This latter continuously acquires 2D cross-section images of those lines, while at the same time the estimation method is applied separately using each of the three image points. The image point 2D coordinates update the estimation algorithm, as described in Section 4.1.1, after they would have been expressed in the fame $\{R_i\}$ using the pose (position and orientation) of this latter with respect to probe frame $\{R_s\}$ (such pose is afforded by the mathematical model). The sampling time is set to 40 ms, and the probe constant velocity to $\mathbf{v} = (-0.07,\ -0.04,\ -0.03,\ 0,\ 0,\ 0)$ (m/s and rad/s). The estimation parameters involved in the recursive relationship (4.8) and (4.9) are set to $\beta = 0.8$, $f_0 = 1e5$,

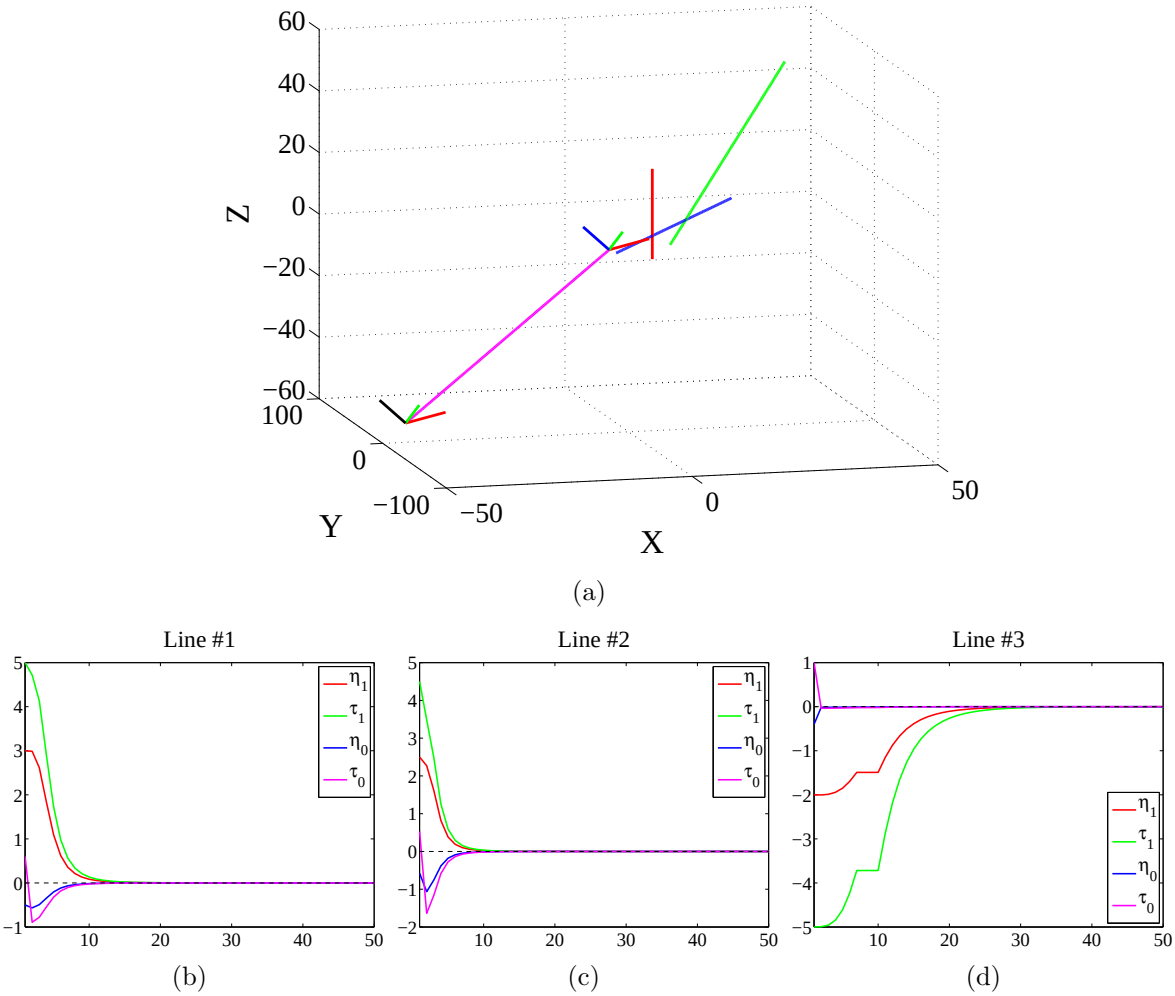


Figure 4.11: Estimation of three 3-D straight lines - (a) Interaction of a virtual 2D ultrasound probe with three 3-D straight lines. The probe has applied a motion with constant velocity and the resulting trajectory (cm, cm, cm) is plotted in magenta. The segments swept by the 2D ultrasound probe plane during that motion are also shown, where line #1 is depicted in red, line #2 in green, and line #3 in blue. Probe frame's X, Y, and Z axes at the initial pose are shown in red, green, and blue color respectively. Whereas, at the final pose the Z axis is depicted in black color - (b), (c), and (d) show the obtained 3-D parameters estimation errors of respectively line #1, line #2, and line #3 versus iteration number.

and $\beta_0 = \frac{1}{20 \cdot f_0}$. These parameters have been empirically tuned. The initial value Θ_0 of Θ is arbitrarily set to $\Theta_0 = [\ 3,\ 5,\ -0.4,\ 1]^\top$. In this simulation we do not employ the SLS algorithm to obtain Θ_0, but we use solely the recursive algorithm in order to first analyze its behavior especially with regards to convergence time. The corresponding simulation results are shown on Fig. 4.11. We can verify that both the three straight lines have been well estimated, as can be seen respectively on Fig. 4.11(b), 4.11(c), and 4.11(d). The errors between the 3-D parameters actual values Θ and those estimated $\hat{\Theta}$ converge to zero, for each line. We can notice that the convergence time related to line #3 is relatively higher than that obtained for the two other lines. This can be explained due to the orientation of this line that tends to be parallel to the probe observation plane, as can be seen on Fig. 4.11(a) (blue line). Indeed, according to the formulation (4.4), the third element d_z of the orientation vector is assumed not null, since otherwise parameters η_1 and τ_1 would equal to infinity (∞). This occurs when the straight line is parallel to the probe observation plane. Finally, as conclusion the obtained results validate the straight-line based method in estimating direction $\mathbf{d_t}$ of 3-D straight lines.

4.4.2 Interaction with curved lines

We apply the curved line-based method, presented in Section 4.1.2, on simulated 3-D curved line. Similarly to the previous section, the interaction of a virtual 2D ultrasound probe with a 3-D curve is simulated with a mathematical model we designed, where the curve relationship is of the form given by (4.12). The model provides the image points 2D coordinates resulting from the intersection of the probe image plane with the curve. The estimation is performed while the probe is moved with constant velocity $\mathbf{v} = (-0.07,\ -0.04,\ -0.03,\ 0,\ 0,\ 0)$(m/s and rad/s). The estimation algorithm is fed and thus updated with the image 2D coordinates of intersection point continually extracted from the cross-section image, as described by (4.15). Before these coordinates are used, they are expressed in frame $\{R_i\}$ using the pose (position and orientation) of the probe's attached frame $\{R_s\}$. The parameters of the recursive algorithm are empirically set to $\beta = 0.8$, $f_0 = 1e5$, and $\beta_0 = \frac{1}{20 \cdot f_0}$, as before. We recall that the algorithm is given by the relationships (4.8) and (4.16). The initial estimate is set to $\Theta_0 = (1,\ 1,\ 1,\ 1,\ 1,\ 1)$, while actual curve is of parameters $\Theta = (2,\ 1.5,\ 0.3,\ 0.5,\ 0.4,\ 0.2)$.

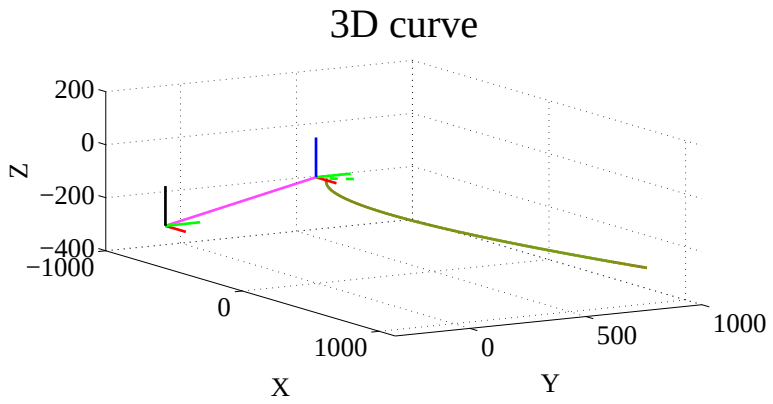
The corresponding simulation results are shown on Fig. 4.12. The estimated curve converges to the actual one, as can be seen on Fig. 4.12(a) where both curves are plotted. Indeed, the curves superimpose on each other. – At each iteration the estimated parameters vector $\hat{\Theta}$ is used to compute the 3-D coordinates of a point of the estimated curve. The whole of points obtained as such, all along the probe motion and the estimation, constitute the estimated curve that is plotted on Fig. 4.12(a). – Note, however, that even though the estimated curve

"*physically*" corresponds to the actual one, the estimated parameters $\hat{\Theta}$ do not correspond to actual one Θ. This may explained by the fact that the mathematical relation (4.12) between the points 3-D coordinates $({}^{i}x,\ {}^{i}y,\ {}^{i}z)$ of a curve and its corresponding parameters $(\eta_2,\ \tau_2,\ \eta_1,\ \tau_1,\ \eta_0,\ \tau_0)$ is not a one to one mapping. Nevertheless, this does not hinder our objective since the algorithm is able to well estimate the parameters that represent the actual curve, which is our goal. Indeed, from those estimated parameters the derived vector $\mathbf{d_t}$ would clearly be tangent to actual curve $\mathcal{K}$, as shown on Fig. 4.12(b) where we can see that the errors vector between actual tangent vector $\mathbf{d_t}$ and estimated one $\hat{\mathbf{d}}_\mathbf{t}$ converges to zero. Accordingly, the obtained result shows the validity of the curve-based method in estimating tangent vector $\mathbf{d_t}$ to 3-D curves, of shape represented by relationships of the form (4.12).

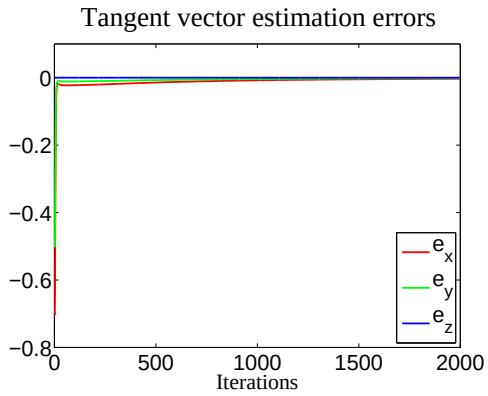
4.4.3 Interaction with quadric surfaces

The quadric surface-based estimation method, presented in Section 4.2, is now tested in simulation. The scenario consists in a 2D virtual probe interacting with a simulated 3-D surface. The interaction is again represented with a mathematical model we designed. To simulate the surface, we employed a relationship of the form given by (4.18). The interaction model provides the image coordinates of the points lying on contour $\mathcal{C}$ of image cross-section $\mathcal{S}$. These coordinates, after being expressed in initial probe frame $\{R_i\}$, are then used to compute the input variable $\mathbf{\Phi}$ according to (4.22); the input $\mathbf{Y}$ being already provided off-line before the estimation is launched. The two inputs continually feed and thus update the estimation algorithm, which estimate Θ according to the relationships (4.8) and (4.16). However, before this recursive algorithm is launched, a SLS algorithm of pre-defined window length is applied in order to firstly obtain estimates $\hat{\Theta}$, that are expected to be relatively closer to the actual parameters Θ. The recursive algorithm will then take place, instead of the sliding one. Note that this contrivance, which has already been introduced at the end of Section 4.3, will be most often applied for performing the estimation with either the straight line-, curved line-, or quadric surface-based estimation methods, as can be encountered in the remaining of the dissertation. Note also that in contrast to the two previous simulations where the recursive algorithm was used solely, the SLS is employed in this case since we noticed that it was quite difficult to estimate the surface using only the recursive algorithm.

The estimation is performed while the virtual probe moves with constant velocity along its orthogonal axis Z; the probe plane being horizontal to the plane (X_0, Y_0) of the base frame $\{R_0\}$. The algorithm parameters are empirically set to $\beta = 1.0$, f_0 =1e2, $\beta_0 = \frac{1}{20 \times f_0}$, $\epsilon_0 =$ 1e-20, $N = 21$, and $N_{LS} = 21$. We recall that N represents the number of points defining a segment. The initial estimated parameters are arbitrarily set to $\Theta_0 = (0,\ 0,\ 0,\ 0,\ 0,\ 0)$. The quadric surface actual parameters are $\Theta = (0.09,\ 0.07,\ 0.04,\ 0.02,\ 0.01,\ 0.05)$. We



(a)



(b)

Figure 4.12: Estimating 3-D curves with which a 2D virtual probe is interacting. (a) The estimation is performed while the probe performs motion with constant velocity, where its resulting path is plotted in magenta (cm, cm, cm). The X, Y, Z axes of the probe attached frame $\{R_s\}$ at the initial time are respectively depicted in red, green, and blue. At the final time they are whereas plotted respectively with red, green, and black (we recall that the X, Y axes are those representing the probe image plane). The actual curve is plotted with red, while the estimated one with green. Those curves superimpose on each other. (b) Tangent vector estimation errors vector $e = (e_x, e_y, e_z) = \mathbf{d_t} - \hat{\mathbf{d}}_\mathbf{t}$ versus iteration number.

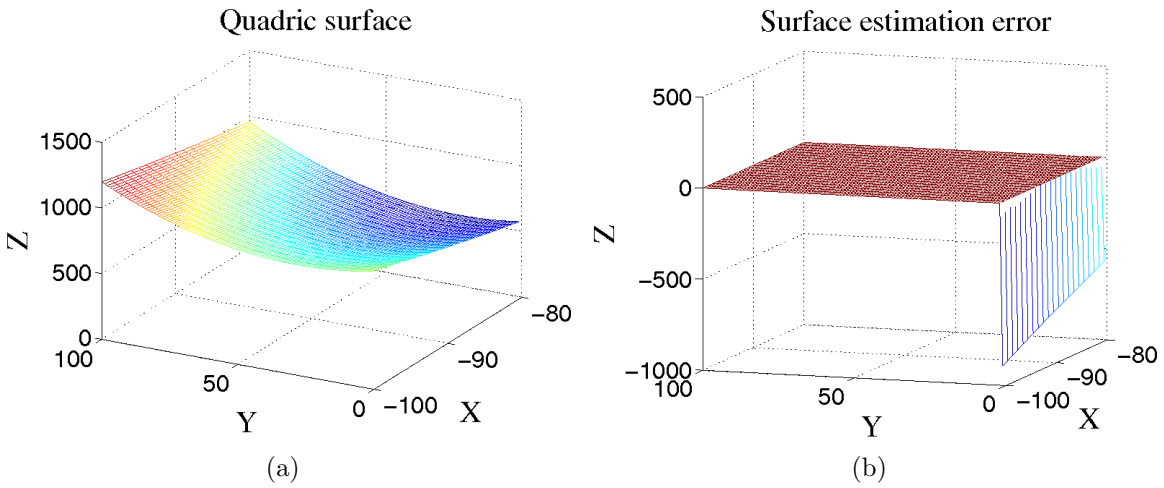


Figure 4.13: Interaction with a quadric surface plotted on (a) - (b) The obtained errors on estimating that surface using the quadric surface-based estimation method.

first test solely the SLS algorithm, presented in Section 4.3. The corresponding simulation results are shown on Fig. 4.13. We can see that the actual surface, shown on Fig. 4.13(a), has been well estimated as can be concluded from the estimation errors shown on Fig. 4.13(b). The latter figure indeed shows the error between elevation z of the actual surface and that of the estimated one, for each swept coordinates (x, y). Those errors are obtained of order ranging from 1e-5 to 1e-8 cm, and those related to the estimated parameters $\hat{\Theta}$ are of order ranging from 1e-8 to 1e-12 (expressed in their corresponding units).

We noticed that the recursive algorithm, if applied alone, had not performed well. In that case the obtained estimation errors between the actual surface elevations and those of the estimated one are of an order ranging from 1e-1 to 1e0 m. The estimation errors related to Θ are not satisfying too. But by applying the SLS algorithm for only one window at the beginning of the estimation then launching the recursive algorithm, the obtained errors on the surface estimation are considerably dropped. Indeed, their order is obtained ranging from 1e-7 to 1e-9 cm. As for the estimation errors on the parameters Θ, their order ranges from 1e-8 to 1e-13.

We presented above results obtained from the first set of trials. Those simulations have been performed to estimate 3-D primitives ranging from straight lines, curved lines, and quadric surface. The obtained results are satisfactory, as pointed out above. Those simulations aimed at verifying that tangent vector $\mathbf{d_t}$ can actually be estimated with the presented methods, when simple primitives are considered. In what follows, we consider more complex geometric primitives. The three estimation method are applied to estimate normal

vector $\nabla\mathbf{F}$ to the surface of a simulated object, namely an ellipsoid, and their corresponding performances are also assessed. These trials represent the second set of simulations we highlighted earlier. They are conducted in two main different conditions. The first one consists in the case of perfect context, where no noise is considered. The second condition, whereas, consists in the case where measurement noise is present. They are presented in what follows.

4.4.4 Ellipsoid objects: perfect and noisy cases

The interaction of a virtual 2D ultrasound probe with an ellipsoidal object is simulated by means of a 3-D mathematical model we designed. This model allows to extract contour $\mathcal{C}$ of cross-section $\mathcal{S}$ lying in the probe observation plane, as shown for example on Fig. 4.14(a), 4.14(b), and 4.14(c). In fact, the extraction consists in obtaining the 2D image coordinates of points lying on the contour. The simulations presented in the remaining of this chapter are conducted using 400 extracted points to characterize the image contour, at each iteration.

The probe is moved with constant velocity as shown on Fig. 4.14, while the image contour points coordinates are extracted at each iteration. During the motion, normal vector ${}^s\nabla\mathbf{F}$ to the ellipsoid is estimated, separately using the three estimation methods. The estimate ${}^s\nabla\hat{\mathbf{F}}$ is compared to the actual one ${}^s\nabla\mathbf{F}$, and the corresponding error is inferred. This allows us to verify if the normal vector has been well estimated. The actual normal vector is computed using again the mathematical model. Indeed, this latter encloses the ellipsoid 3-D model that is expressed as follows:

$$F = ({}^ox/a_1)^2 + ({}^oy/a_2)^2 + ({}^oz/a_3)^2 - 1 = 0 \tag{4.31}$$

where a_1, a_2 and a_3 are 3-D parameters that represent the ellipsoid shape (i. e., the ellipsoid half length values), whereas $({}^ox, {}^oy, {}^oz)$ are the 3-D coordinates of point ${}^o\mathbf{P}$, that lies on the ellipsoid surface. These coordinates are expressed in frame $\{R_o\}$ attached to the center of the ellipsoidal object. Using the above relationship, the actual normal vector can be calculated by applying (4.23) as follows:

$${}^o\nabla\mathbf{F} = \begin{pmatrix} 2\,{}^ox/a_1^2 \\ 2\,{}^oy/a_2^2 \\ 2\,{}^oz/a_3^2 \end{pmatrix} \tag{4.32}$$

which can be expressed in $\{R_s\}$ by ${}^s\nabla\mathbf{F} = {}^s\mathbf{R}_o\,{}^o\nabla\mathbf{F}$. The point 3-D coordinates are calculated from its image coordinates (x, y), using the relationship (3.37) presented in Chapter 3.

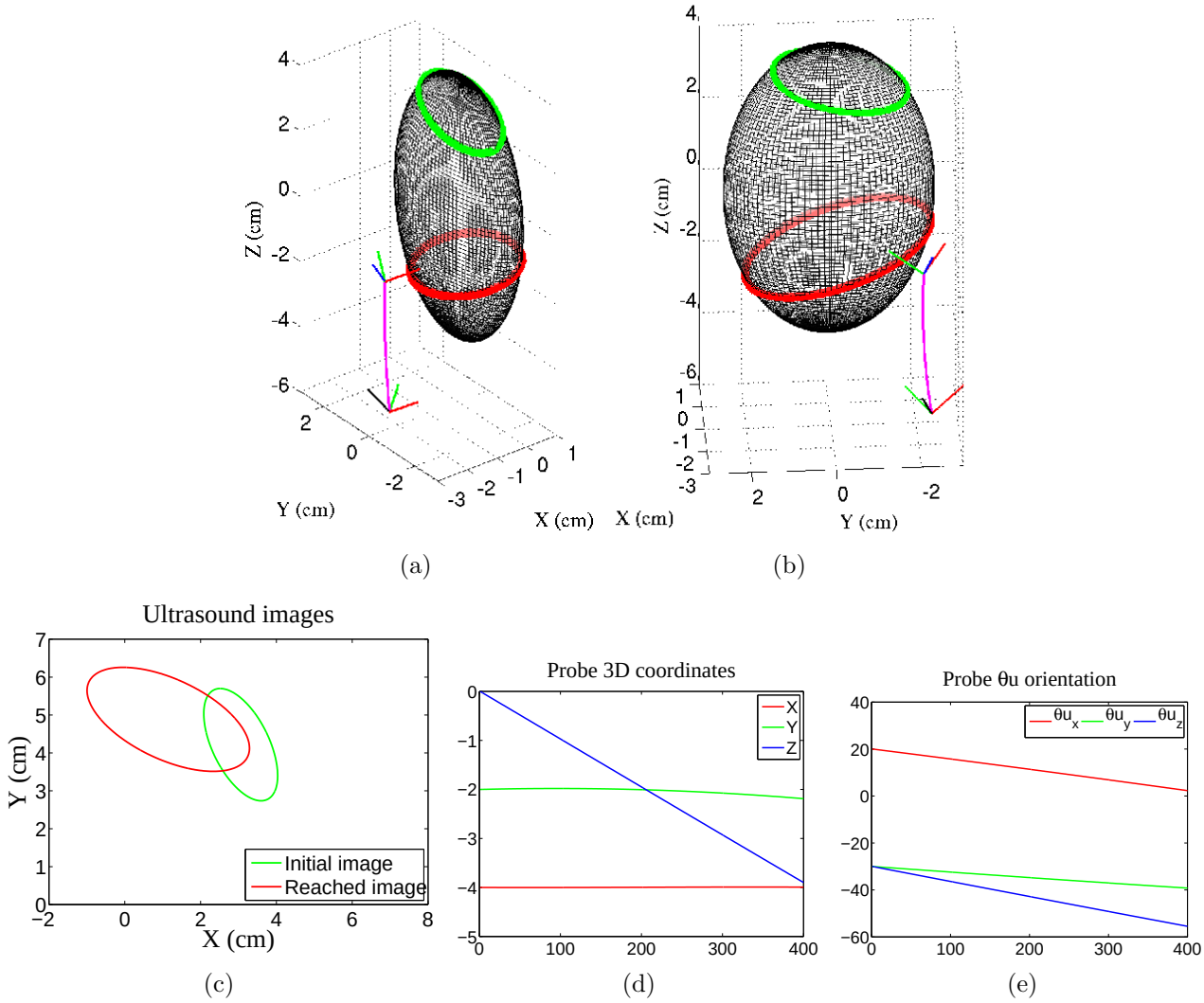
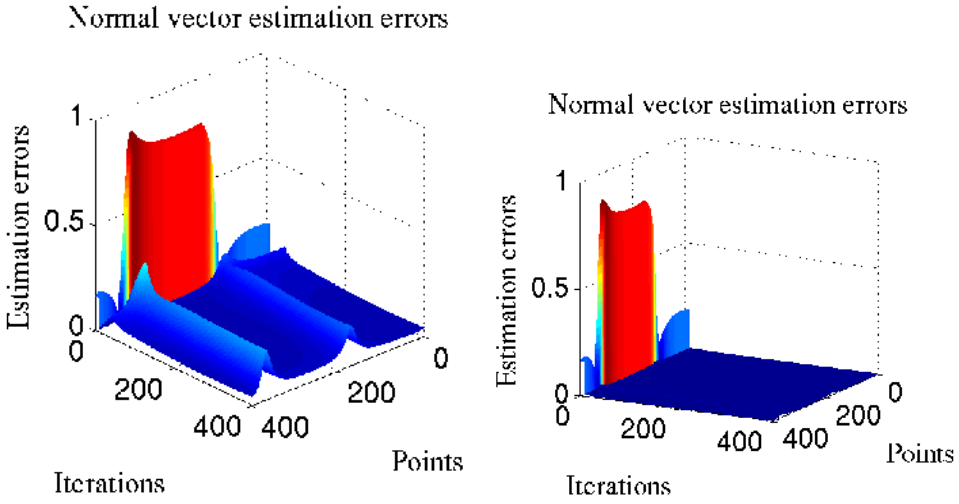


Figure 4.14: Simulation of a 2D ultrasound probe that interacts with an ellipsoidal object. It is afforded by the mathematical model. The probe performs a motion with constant velocity - (a) The frame $\{R_s\}$ of its initial and final poses is indicated. At the initial time the frame's (X, Y, Z) axes are depicted respectively with the (red, green, blue) lines. Whereas at the final pose, the probe Z axis is depicted with a black line. The probe path is plotted in magenta. The intersection of the probe observation plane with the ellipsoid results in a cross-section, whose contour at the initial and final probe poses is respectively depicted with green and red - (b) Another image of the interaction taken from a different sight angle - (c) The contour image at the initial and final poses is respectively indicated with green and red color - (d) Evolution of the probe 3-D coordinates (cm, cm, cm) during the motion versus iterations, while the $\theta\mathbf{u}$ orientations (deg, deg, deg) are shown in (e).



(a) (b)

Figure 4.15: Normal vector estimation errors e_f obtained using the straight line-based method. The probe is interacting with an ellipsoidal object - (a) Using the contrivance that consists in applying the recursive algorithm, right after the SLS one would have been applied for the first N_{LS} iterations. The estimator parameters are set to $\beta = 0.95$ and f_0 =1e8 - (b) Applying the SLS algorithm throughout the estimation. The estimator parameters are set to $\beta = 0.5$ and f_0 =1e8.

The involved rotation matrix ${}^s\mathbf{R}_o$ and the translation vector ${}^s\mathbf{t}_0$ are provided by the interaction mathematical model. This latter also provides the image coordinates $(x,\ y)$. The estimation error e_f consists in the square root of the vectorial error $\mathbf{e}_f = {}^s\nabla\mathbf{F} - {}^s\nabla\hat{\mathbf{F}}$. For each point, it is thus given by:

$$e_f = \|\mathbf{e}_f\|_2 = \sqrt{\mathbf{e}_f{}^\top \mathbf{e}_f} = \sqrt{e_{fx}^2 + e_{fy}^2 + e_{fz}^2} \tag{4.33}$$

where $\mathbf{e}_f = (e_{fx},\ e_{fy},\ e_{fz})$. Below, we first present results obtained in the ideal case, where no perturbation is introduced. We then consider the case of measurement noises.

Straight lines-based estimation

We apply the straight line-based method to estimate the normal to the surface of the ellipsoidal object.

We present results of two differently performed estimations. In the first simulation we em-

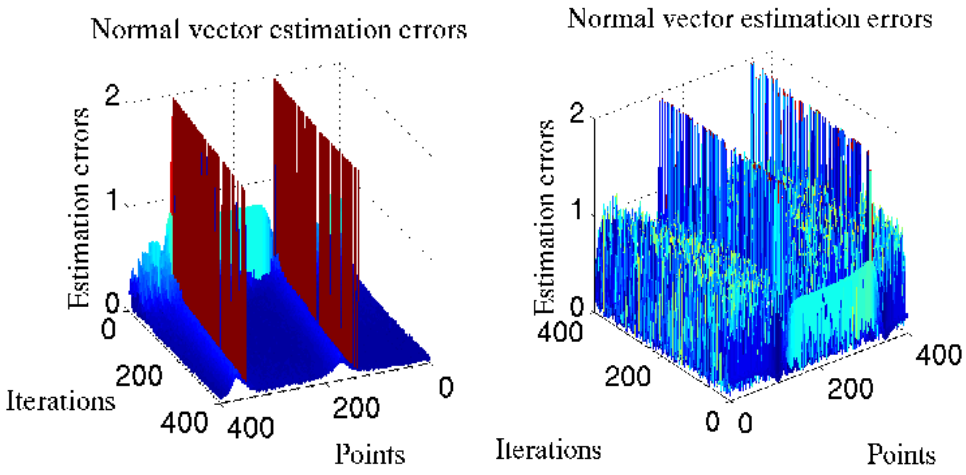


Figure 4.16: Normal vector estimation errors e_f obtained using the straight line-based method, in the presence of additive measurement noise of 0.3 mm amplitude. Simulation where a virtual 2D ultrasound probe is interacting with an ellipsoidal object - (a) Using the contrivance that consists in first employing the SLS algorithm, for only one window, and then applying the recursive method during all the estimation. The estimator parameters have been tuned to $\beta = 0.95$ and f_0 =1e8 - (b) Applying the SLS algorithm, alone, throughout the estimation. The estimator parameters have been tuned to $\beta = 0.5$ and f_0 =1e8.

ploy the contrivance that consists in applying firstly the SLS algorithm for only the first N_{LS} iterations and then in using the recursive one. As pointed out, this contrivance allows to obtain after the SLS algorithm being achieved (after the first N_{LS} iterations) an estimate $\hat{\Theta}$ that is expected to be closer to the actual one Θ. The estimation convergence time would be, as a result, considerably shortened. Note that this is of great interest in the context where the image is varying, and thus when the normal vector is also changing, as it is the case in the simulations we present and in general practical cases. As for the second simulation, the SLS algorithm is applied all along the virtual probe motion. Obtained results are shown on Fig. 4.15. We can see that the estimation errors are quite dropped to zero with the two algorithms (respectively performed in the first and the second simulation). However, we can notice two folds obtained with the recursive algorithm. They are grossly centered on two points of the image contour. The tangent vector at those two points likely tends to be parallel to the image plane, which could account for the lesser dropped estimation errors at those points and their close neighborhood. These results therefore suggest that the second algorithm outperforms the first one, as can be clearly seen on Fig. 4.15(b) where the obtained estimation errors are nearly null. That conclusion however showed to be not valid in other contexts. Indeed, in the presence of measurements noise that comparison's

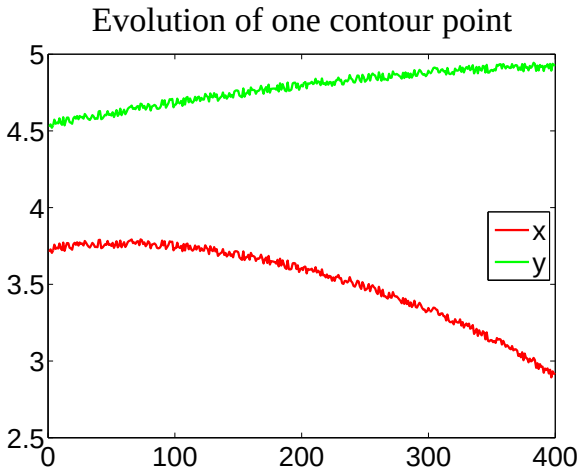


Figure 4.17: Image coordinates (x, y) in cm of one point lying on the contour, perturbed with an additive measurement noise of 0.3 mm amplitude. The two coordinates x and y are plotted with respect to the iteration number.

conclusion is dramatically reversed. In such noise context we noticed that the recursive algorithm has been able to estimate the normal vector, whereas it is not the case when using the SLS algorithm. This can be seen on Fig. 4.16, that shows results obtained from simulations conducted in the presence of measurement noise in the image. This considered perturbation consists in a 0.3 mm amplitude random white Gaussian noise. Its effects on the evolution of the image coordinates of one contour point, during the probe motion, is shown on Fig. 4.17. This noise is added to the original extracted image points coordinates (x, y) and their derivatives with respect to the angle in the image. We notice peaks at two points of the image contour, with both algorithms. Those peaks indicate that the estimation has not been well performed at those two contour points. They seem in fact as successors, although worse, of the two folds obtained in the ideal case. Nevertheless, it is unlikely that such two peaks might compromise the system performance. Indeed, the objective of estimating the normal vector is to use it in order to compute the control law. We do not use only a couple but at least hundreds of image points, and thus of estimated normal vectors to compute the control law; we recall that we use 400 points in the simulations we present in the present chapter. Consequently, the obtained two peaks constitute an error with a weight of only $2/400$, which is negligible. They are therefore considered as modeling errors, and thus can be reduced by the servoing scheme thanks to its closed loop.

Curved lines-based estimation

Similarly as described above, we now apply the curved line-based method to estimate the normal vector to the ellipsoidal object surface. The virtual probe is moving with constant velocity, where the resulting interaction with the ellipsoid is shown on Fig. 4.14. During that motion, the estimation is performed at each of the 400 image contour points. The estimates are then compared to the actual values of the normal vector, and the estimation error e_f given by (4.33) is inferred. The actual values are computed from the interaction mathematical model, according to the relationship (4.32). Again, the estimation is performed according to two different approaches. The former approach employs the contrivance consisting in applying firstly the SLS algorithm for only the first N_{LS} iterations and then applying the recursive method for the remaining of the estimation. The second approach is performed by applying solely the SLS algorithm throughout the estimation. Their corresponding results are then compared.

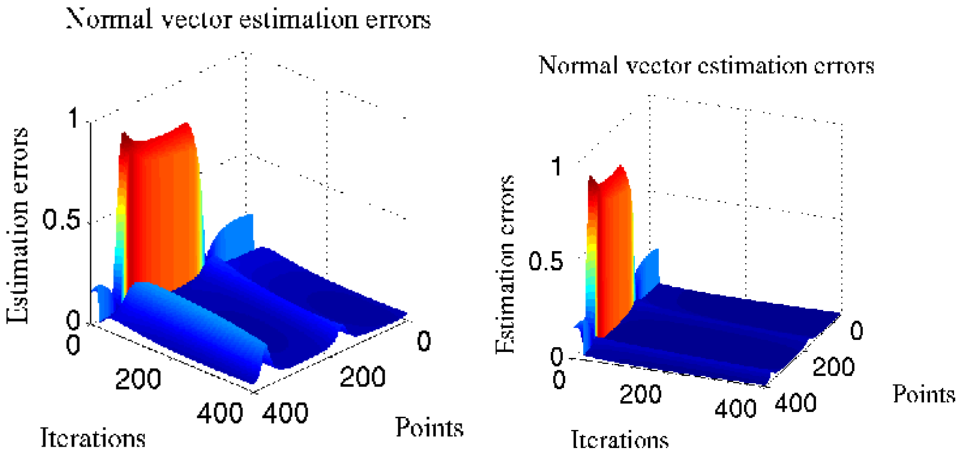
We first consider the ideal case where the system is not subject to perturbations. Corresponding simulation results are shown on Fig. 4.18. We can note that the sliding least squares estimation approach slightly outperformed the recursive one, as can be seen respectively on Fig. 4.18(b) and Fig. 4.18(a).

We now consider the case where a noise perturbs the image. This measurement noise is again set as random white Gaussian noise of 0.3 mm amplitude. Obtained simulation results are shown on Fig. 4.19. We can notice that, again, the estimation using the sliding algorithm slightly outperformed that using the recursive paradigm.

The performances of the straight line- and the curved line-based estimation methods will be compared to that of the quadric surface-based estimation, as it is presented in Section 4.5.

Quadric surface-based estimation

In the same scenario, we also applied the quadric surface-based method to estimate the normal vector to the ellipsoid surface. Similarly, the estimation is performed according to two approaches. We first consider the ideal case, and then the case where an additive measurement noise is considered. To fit the quadric surface, we tune N to $N = 21$ points (segment width) and the window size to $N_{LS} = 21$ iterations. The estimator parameters β_0 and ϵ_0 are tuned to $\beta_0 = \frac{1}{20 \times f_0}$ and ϵ_0 =1e-20. Corresponding obtained results are shown on Fig. 4.20. We can conclude from these results that the estimation employing the recursive method has outperformed that employing the SLS algorithm. The recursive algorithm has grossly well estimated the normal vector, but only in the ideal case. In the noisy case they are however



(a) (b)

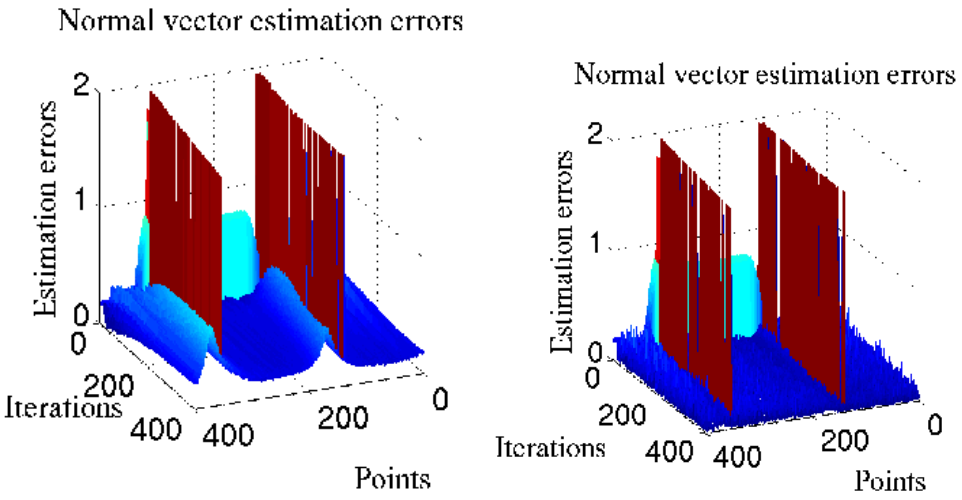
Figure 4.18: Normal vector estimation errors e_f obtained using the curved line-based method, in the ideal case. The scenario consists in an interaction of a virtual 2D ultrasound probe with an ellipsoidal object - (a) Using the contrivance that consists in applying the recursive method, right after the SLS algorithm would have been applied for only the first N_{LS} iterations. The estimator parameters have been tuned to $\beta = 0.9$ and $f_0 = 5{\times}1e3$ - (b) Using the SLS algorithm throughout the estimation. The estimator parameters have been tuned to $\beta = 0.9$ and $f_0 = 5{\times}1e3$.

both not satisfactory. A more detailed discussion is given in the following section.

4.5 Discussion

The obtained results suggest that the curved line-based method has outperformed the two other methods. Indeed, the curved line-based method has been able to provide a good estimate of the normal vector ${}^s\nabla\mathbf{F}$ both in ideal cases, where no perturbation is occurring, and in cases where measurements noises are present in the image. The straight line-based method has not performed as the curved-based one in the presence of measurement noise. As for the quadric-based one, the performances are even less better than that of both the two first methods, especially in the presence of measurement noise, where the results are not satisfactory. This method is furthermore computationally more expensive, since it uses a segment to update its estimate at each iteration, instead of using only one point as it is the case for the two first methods.

We presented results that as actual as possible reflect the performances of each of the three



(a) (b)

Figure 4.19: Normal vector estimation errors e_f obtained using the curved line-based method, in the presence of an additive measurement noise of 0.3 mm amplitude - (a) Using the contrivance that consists is applying the recursive method, right after the SLS algorithm would have been applied for only one window. The estimator parameters are set $\beta = 0.95$ and f_0 =1e2 - (b) Using the SLS algorithm throughout the estimation. The estimator parameters are set to $\beta = 0.95$ and f_0 =1e2.

estimation methods. Indeed, the performance depends on the tuned estimation parameters β, f_0, β_0, N_{LS}, and ϵ_0 (and also N for the quadric surface-based method). The parameters, as highlighted, have been empirically tuned in order to obtain as best as possible estimation results, separately for each of the three methods. The tuning has been performed at each time the simulation condition changed (perfect or noisy) and at each time a different estimation method is employed. To do so, we have performed many different trials where the estimator parameters are tuned according to the famous dichotomy manner. We thus have presented results that we consider have been tuned in order they have allowed each method to perform as best as possible. During those trials, we noticed that the performance of both the straight line- and the curved line-based method is only slightly affected by the variations of those parameters. Note that large variations of the parameters have been considered. As for the quadric-based method, we obtained another conclusion. Indeed, the performances of this method heavily rely on the values of the estimation parameters, and is quite affected by their variations. It is nevertheless important to note that the sliding estimation algorithm, we presented in Section 4.3, corresponds to a vectorial algorithm, that is, the input measure $\mathbf{Y}$ is a vector and not a scalar. Yet, the original formulation of this algorithm is stated for scalar inputs measures [22]. We have in fact tried to adapt

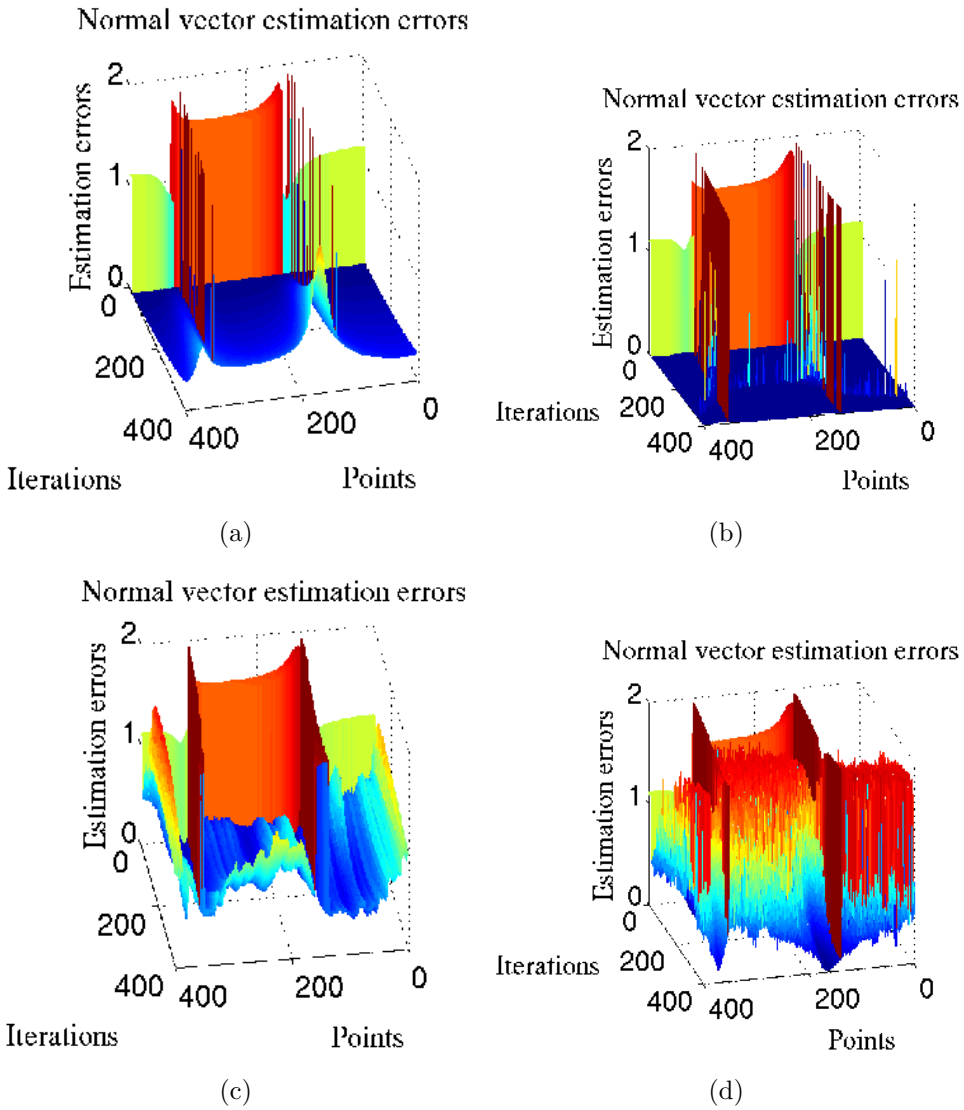


Figure 4.20: Normal vector estimation errors e_f obtained using the quadric surface-based estimation. The results obtained in the ideal case are shown on (a) and (b), while those obtained in the presence of perturbation are shown on (c) and (d) - (a) The estimation is performed by employing the recursive method, right after the SLS algorithm has been employed for only one window. The remaining estimator parameters are tuned to $\beta = 0.5$ and f_0 =1e8 - (b) Employing the SLS algorithm throughout the estimation. The remaining estimator parameters are tuned to $\beta = 1.0$ and f_0 =1e2 - (c) Employing the recursive algorithm after the SLS one. The estimator parameters are tuned to $\beta = 1.0$ and f_0 =1e2 - (d) Employing the SLS algorithm alone. The estimator parameters are tuned to $\beta = 1.0$ and f_0 =1e2.

the original algorithm to a vectorial case. It could be therefore possible that some modifications have not been rigorously taken into account. We performed other trials but using the original scalar sliding algorithm. To do so, the estimation model (4.3), firstly, has been decomposed in two scalar equations. Each equation has been considered as an estimation model (with η_1 and η_0 as parameters for the former scalar system to estimate, and τ_1 and τ_0 as parameters for the latter scalar system). Then, each of the obtained estimates are combined and the estimate of the normal vector is derived. The same approach is undertaken for the curve system (4.12). We obtained similar results, with both straight line- and curved line-based method (using the scalar formulation), to those previously obtained with the vectorial algorithm respectively, except that in the noisy case we obtained better result with scalar straight line-based estimation than previously. Nonetheless, the performances of this latter method are still lower than that of the curved line-based estimation. As for the quadric-based estimation method, the scalar algorithm, as is, seems not relevant, since this estimation method inherently uses a segment of points (and thus a vector of measures) to update the estimate. That is the reason why we presented in Section 4.3 a vectorial version of the sliding algorithm. The low outcome of the quadric-based estimation method could be explained by the fact that this method estimates in whole the normal vector and thus the estimation errors are expected to be larger than those obtained with the two first methods. Indeed, these latter methods estimate instead only a part ${}^s\mathbf{d_t}$ of the normal vector, while the second part is directly extracted from the observed image. Moreover, fitting a surface to a cloud of points seems more constrained than fitting a line to a set of points.

4.6 Conclusion

We proposed in this chapter three methods to estimate on-line the normal vector to the surface of an object with which a 2D ultrasound probe is interacting. We recall that such normal vector appears in the interaction matrix that relates the image moments time variation to the probe velocity, as developed and presented in the previous chapter. The estimation is performed without any prior knowledge of the shape, 3-D parameters, nor location in the 3-D space of the observed object. Doing so, we overcome the limitation and constraints imposed if the resolution of developing a pre-operative model of the observed object would be envisaged.

The three methods we proposed are based on respectively straight line, curved line, and quadric surface primitives. Their performances have been compared. They have been tested in different simulation trials, where satisfactory results have been obtained with the curved-line based estimation method. The straight line-based method showed to be relatively more sensitive to measurements noise. As for the quadric surface-based method, besides of being even more sensitive to the noise, it requires rigorous tuning of its estimation parameters.

Chapter 5

Visual Servoing

In the present chapter, we finally design novel ultrasound-based image moments-based visual servoing schemes. These latter will allow to automatically position a 2D ultrasound probe in order to reach and maintain a desired cross-section image. After having modeled interaction matrix $\mathbf{L}_{m_{ij}}$ that relates image moment time variation $\dot{m}_{ij}$ to probe velocity $\mathbf{v}$, in Chapter 3, and having developed techniques to estimate on-line normal vector ${}^{s}\nabla\mathbf{F}$ to the object surface, in Chapter 4, visual servoing schemes can now be designed whether we have a pre-operative 3-D model of the observed object or not. The section in the image can be described by a combinations set of image moments m_{ij} enclosed in a vector we denote $\mathbf{s}$, that we use as feedback visual features in the control scheme. The servoing objective, stated above, thus can be formulated as to automatically move the probe in order that the vector $\mathbf{s}$ becomes identical with the features vector $\mathbf{s}^*$ that describes the desired image section. Vector $\mathbf{s}^*$ is nothing but vector $\mathbf{s}$ computed on the desired image. As already introduced and discussed in Section 3.1, a set of combinations of image moments can be used to represent the image section. Thus, vector $\mathbf{s}^*$ represents the desired image section. Consequently, when $\mathbf{s}$ becomes equal to $\mathbf{s}^*$, it means that the observed image well and truly corresponds to the desired one. So, the servoing objective can be mathematically formulated as to move the robot in order that visual error $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$ converges to zero. To build the servoing scheme, we need to relate vector $\mathbf{s}$ as function of probe velocity $\mathbf{v}$. To do so, we use the modeling results we obtained, i. e., the relationships (3.34) and (3.35), since $\mathbf{s} = \mathbf{s}(m_{ij})$. We thus write time variation $\dot{\mathbf{s}}$ of $\mathbf{s}$ as function of $\mathbf{v}$ in the following linear form:

$$\dot{\mathbf{s}} = \mathbf{L}_{\mathbf{s}}\,\mathbf{v} \tag{5.1}$$

where $\mathbf{L}_{\mathbf{s}}$ is the interaction matrix related to $\mathbf{s}$. Such matrix, along with the visual features $\mathbf{s}$ it relates, in any visual servoing schemes, is crucial to design the control law and has

predominant effect on the robot behavior [82, 27, 71, 14]. We use a classical control law [26], as follows, such that the visual error $\mathbf{e}$ is expected to converge to zero exponentially (to smoothly stop at the desired image). When $\dim(\mathbf{s}) = 6$, that control law is:

$$\mathbf{v_c} = -\lambda \, \hat{\mathbf{L}}_\mathbf{s}^{-1} \, (\mathbf{s} - \mathbf{s}^*) \tag{5.2}$$

where $\mathbf{v_c}$ is the probe velocity command sent to the low-level robot controller, λ is a positive control gain, and $\hat{\mathbf{L}}_\mathbf{s}^{-1}$ is the inverse of the estimated interaction matrix $\mathbf{L_s}$. Such obtained control scheme is known to be locally asymptotically stable when a correct estimation $\hat{\mathbf{L}}_\mathbf{s}$ of $\mathbf{L_s}$ is used (that is, as soon as $\mathbf{L_s} \, \hat{\mathbf{L}}_\mathbf{s}^{-1} > 0$) [26]. The global convergence can not be ensured in our case with this control law. This is due to the fact that the object surface might have local minima (i. e., concave regions) in which the probe could be trapped.

When less than six visual features are enclosed in $\mathbf{s}$ (i. e., matrix $\mathbf{L_s}$ becomes not square), the pseudo inverse $\hat{\mathbf{L}}_\mathbf{s}^{+}$ of the estimated interaction matrix $\mathbf{L_s}$ is employed in (5.2), instead of the inverse $\hat{\mathbf{L}}_\mathbf{s}^{-1}$. This pseudo inverse is given by:

$$\hat{\mathbf{L}}_\mathbf{s}^{+} \; = \; \hat{\mathbf{L}}_\mathbf{s}^{\top} \, \left(\hat{\mathbf{L}}_\mathbf{s} \, \hat{\mathbf{L}}_\mathbf{s}^{\top} \right)^{-1} \tag{5.3}$$

Matrix $(\hat{\mathbf{L}}_\mathbf{s} \, \hat{\mathbf{L}}_\mathbf{s}^{\top})$ should be of full rank.

In case a pre-operative 3-D model of the observed object is used to obtain an approximate ${}^s\nabla\tilde{\mathbf{F}}$ of the normal vector in the control law, the servoing method is referred to as *model-based visual servoing* method. A corresponding visual servoing scheme is presented on Fig. 5.1. If otherwise, neither prior knowledge of the shape of the object, its 3-D parameters, nor its location is used, but instead the normal vector is on-line estimated with one of the methods developed and presented in Chapter 4, the servoing method is referred to as *model-free visual servoing* method. A corresponding visual servoing scheme is presented on Fig. 5.2. Note that the estimate of the normal vector is denoted ${}^s\nabla\hat{\mathbf{F}}$, while its approximate from a pre-operative 3-D model is denoted ${}^s\nabla\tilde{\mathbf{F}}$.

As for the selection of the visual features, if the observed object presents asymmetric parts we can define six independent visual features. The first three visual features can be defined to control the probe-in-plane motions, while the last three elements can be defined to control the probe out-of-plane motions. The whole of these six visual features thus can define the complete probe motion.

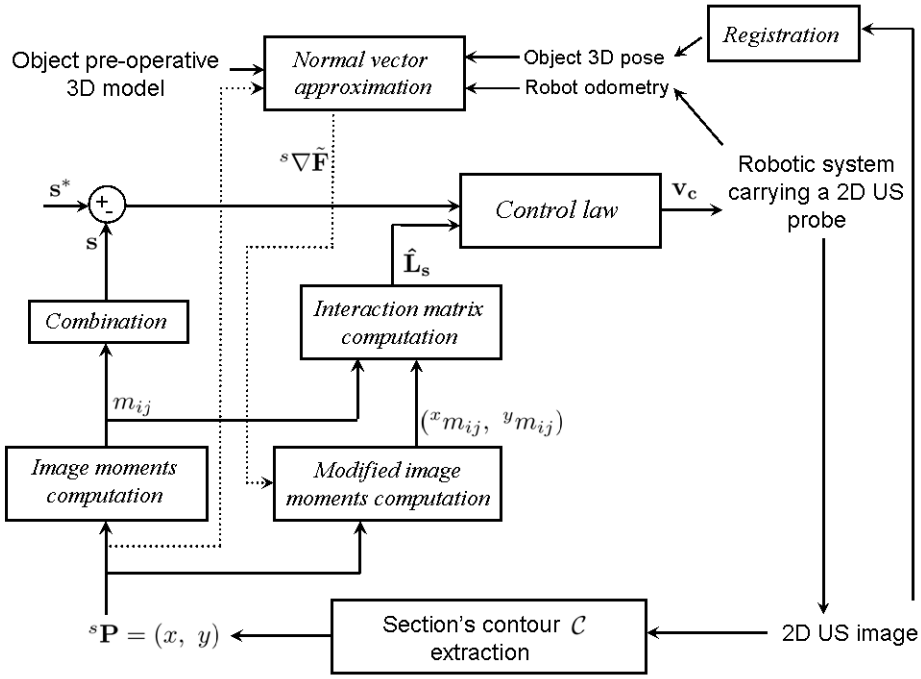


Figure 5.1: Model-based visual servoing scheme. Note that ${}^s\nabla\tilde{\mathbf{F}}$ is an approximate of ${}^s\nabla\mathbf{F}$.

5.1 Visual features selection

When the 2D ultrasound probe performs in-plane motions, section $\mathcal{S}$ only shifts and rotates in the image. Such configurations changes of the image section can be observed respectively by the coordinates of its gravity center and the orientation of its main angle in the image. 2D image coordinates $(x_g,\ y_g)$ of an object gravity center have already been introduced in Chapter 3 and are expressed in terms of image moments up to the first order by the relationship (3.48). We select them as the first two elements of $\mathbf{s}$. The third element consists in the main angle of the section with respect to image X axis (see Fig. 5.3). It is defined by:

$$\alpha = \frac{1}{2}\arctan\left(\frac{2\,\mu_{11}}{\mu_{20}+\mu_{02}}\right) \tag{5.4}$$

where μ_{ij} is the $(i+j)$th order central image moment. It is defined by the following double integral over image section $\mathcal{S}$:

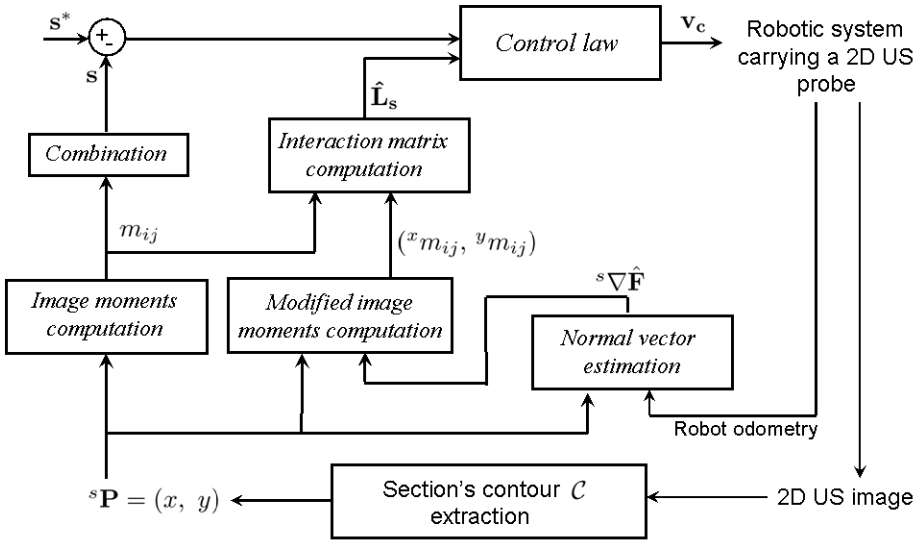


Figure 5.2: Model-free visual servoing scheme.

$$\mu_{ij} = \iint_{\mathcal{S}} (x - x_g)^i \, (y - y_g)^j \, dx \, dy \tag{5.5}$$

An $(i+j)$th order central image moment can thus be defined as function of image moments of up to the $(i+j)$th order. We provide the expressions of up to the third order central image moments, as follows:

$$\begin{cases} \mu_{20} &= m_{20} - m_{10}\, x_g \\ \mu_{11} &= m_{11} - y_g\, m_{10} \;=\; m_{11} - x_g\, m_{01} \\ \mu_{02} &= m_{02} - m_{01}\, y_g \end{cases} \tag{5.6}$$

and

$$\begin{cases} \mu_{30} &= m_{30} - 3\, m_{20}\, x_g + 2\, m_{10}\, x_g^2 \\ \mu_{03} &= m_{03} - 3\, m_{02}\, y_g + 2\, m_{01}\, y_g^2 \\ \mu_{21} &= m_{21} - 2\, m_{11}\, x_g - m_{20}\, y_g + 2\, m_{01}\, x_g^2 \\ \mu_{12} &= m_{12} - 2\, m_{11}\, y_g - m_{02}\, x_g + 2\, m_{10}\, y_g^2 \end{cases} \tag{5.7}$$

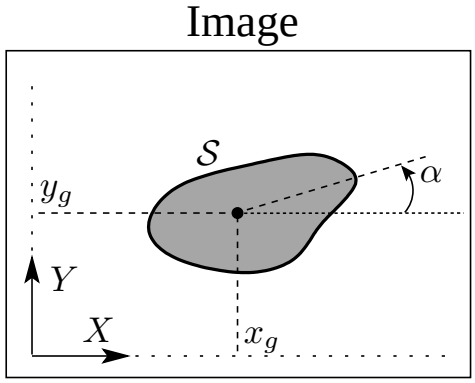


Figure 5.3: Sketch representing image coordinates (x_g, y_g) of gravity center of observed section $\mathcal{S}$, and main orientation α of the latter.

Consider now the probe out-of-plane motions. In the following, we describe how to obtain three independent visual features that can relate such motions. These features thus would represent the last three elements of $\mathbf{s}$. In contrast to in-plane motions, when out-of-plane motions occur the section in the image generally deforms. Its size varies and its shape changes. Therefore, the objective consists to derive three visual features that are respectively sensitive to such modifications of the section in the image, while at the same time they are insensitive to modifications due to probe in-plane motions, in order they are independent from the first three features of $\mathbf{s}$. Firstly, since the size variation can clearly be related to the area a of the section in the image, we can select the fourth element of the visual features vector as $\sqrt{a}$. Note that we applied the square root to a since the three element $(x_g, y_g, \sqrt{a})$, thus brought together, have a same unit, that is meter is this case. Secondly, as for the shape variations, they can be related by image moments from the second and higher orders. However, as highlighted above, the last three features should be insensitive to in-plane motions; area a obviously satisfies such condition. As for the prospective last two visual features, they can be obtained from moment invariants, introduced in Section 3.1. Indeed, image moments can be made invariant to image translation, rotation, and image scale changes. These traits are consequently of great interest in the present case. Let us first search for the fifth element of $\mathbf{s}$; the same manner to proceed is afterwards applied for the sixth element. A visual feature corresponding to a combination of moments of the second and higher orders that are invariant to scale change is expected independent from the image area, and thus also from $\sqrt{a}$. This can be explained by the fact that scale changes are mainly related to those of the image section area. As a result, a visual feature invariant to scale would grossly be insensitive to area a, and thus independent from it. Moreover, when this visual feature is also invariant to translation and rotation, it would be independent respectively from gravity center coordinates (x_g, y_g) and orientation α. In other words,

such feature would be independent also from the in-plane motions. To summarize, this feature would be therefore independent from the first four elements $(x_g, y_g, \alpha, \sqrt{a})$. We select this fifth feature from moment invariants of the second order. Finally, the sixth visual feature is similarly selected as an invariant image moment to translation, rotation, and scale, but is obtained from combination of third order image moments. Indeed, since this sixth feature would be obtained from third order image moments, wile the fifth feature is from the second order ones, the former feature is expected independent from the latter one. We thus can choose these last two features from such moment invariants, already provided in the literature. We employ features provided in [76]. We denote them respectively by ϕ_1 and ϕ_2. They are expressed in terms of image moments as follows:

$$\left\{ \begin{array}{rcl} \phi_1 & = & I_1/I_2 \\ \phi_2 & = & I_3/I_4 \end{array} \right. \tag{5.8}$$

where $I_1 = \mu_{11}^2 - \mu_{20}\,\mu_{02}$, $I_2 = 4\,\mu_{11}^2 + (\mu_{20} - \mu_{02})^2$, $I_3 = (\mu_{30} - 3\,\mu_{12})^2 + (3\,\mu_{21} - \mu_{03})^2$, and $I_4 = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2$.

The visual features vector $\mathbf{s}$ we propose is thus:

$$\mathbf{s} = (x_g,\ y_g,\ \alpha,\ \sqrt{a},\ \phi_1,\ \phi_2) \tag{5.9}$$

Time variation $\dot{\mathbf{s}}$ of $\mathbf{s}$ can now be analytically related to probe velocity $\mathbf{v}$, using interaction matrix $\mathbf{L}_{mij}$ given by (3.34). We obtain, after arranging the elements related to the probe in-plane motions (v_x, v_y, ω_z) and those to the out-of-plane motions $(v_z, \omega_x, \omega_y)$, as follows:

$$\dot{\mathbf{s}} = \begin{bmatrix} -1 & 0 & y_g & x_{g_{vz}} & x_{g_{\omega x}} & x_{g_{\omega y}} \\ 0 & -1 & -x_g & y_{g_{vz}} & y_{g_{\omega x}} & y_{g_{\omega y}} \\ 0 & 0 & -1 & \alpha_{vz} & \alpha_{\omega x} & \alpha_{\omega y} \\ 0 & 0 & 0 & \frac{a_{vz}}{2\sqrt{a}} & \frac{a_{\omega x}}{2\sqrt{a}} & \frac{a_{\omega y}}{2\sqrt{a}} \\ 0 & 0 & 0 & \phi_{1vz} & \phi_{1\omega x} & \phi_{1\omega y} \\ 0 & 0 & 0 & \phi_{2vz} & \phi_{2\omega x} & \phi_{2\omega y} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega_z \\ v_z \\ \omega_x \\ \omega_y \end{bmatrix} \tag{5.10}$$

The detailed form of some elements is not provided because of their tedious form. We can note that the selection of $\mathbf{s}$ given by (5.9) yields the visual servoing scheme partially decoupled. Indeed, we can first remark from (5.10) that the last three elements $(\sqrt{a}, \phi_1, \phi_2)$

of $\mathbf{s}$ are invariant to the in-plane motions. Moreover, the first elements (x_g, y_g, α) present a good decoupling property for the in-plane motions, owing to the triangular part they form.

Although the selection (5.9) yields good decoupling properties, as also shown from results of simulations we conducted, we however noticed from further simulations that element ϕ_1 is relatively less robust to image noise than, for example, the length of the image section main axis. We denote the latter feature by l_1. It is expressed in terms of image moments as follows [16]:

$$ {l_1}^2 = \frac{2}{a}\left(\mu_{02} + \mu_{20} + \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}\right) \tag{5.11} $$

Therefore, the fifth element ϕ_1 could be, in some cases, substituted by l_1. Such selection is of course subject to a trade-off between probe decoupling motions, obtained with the former feature, and more robustness to image noise with the latter.

The remainder of the chapter presents visual servoing results. It is organized as follows. In Section 5.2, we test both the model-based and model-free visual servoing methods in simulations where the probe interacts with an ellipsoidal object. We consider, for that, both ideal cases where no perturbation is present and the cases where additive measurements noise perturbs the image. In Section 5.3 and Section 5.4, we present results obtained from simulations respectively on realistic 3-D ultrasound object and on an asymmetric binary object. Finally, *ex-vivo* experimental results obtained with both a spherical object, an ultrasound phantom, a lamb kidney, and a gelatin-made soft tissue object relatively complex are reported in Section 5.5.

5.2 Simulation results with an ellipsoidal object

The scenario consists of a virtual 2D ultrasound probe that interacts with an ellipsoidal object. The virtual robotic task consists to automatically position the probe in such a way to reach a target image, starting from one totally different. To do so, the probe is servoed by the control scheme we developed in this thesis. The command velocity are generated with the control law (5.2). Note however that since the object is ellipsoidal, the observed cross-section is an ellipse in the image. Consequently, we can define only five independent visual features from the image. Thus, instead of using $\hat{\mathbf{L}}_{\mathbf{s}}^{-1}$, the pseudo inverse $\hat{\mathbf{L}}_{\mathbf{s}}^{+}$ given by (5.3) is employed in (5.2).

The interaction of the probe with the object is simulated using the mathematical model we developed and which has been introduced in Section 4.4.4. This model allows to position

and move the probe. It provides the observed image in form of contour points set, from which the visual features $\mathbf{s}$ are computed. In the following simulations, we use 400 image contour points to compute the visual features along with the interaction matrix. The interaction mathematical model also provides the pose (position and orientation) of the probe (i. e., of its attached frame, already denoted $\{R_s\}$) with respect to a base frame. With this model, we can test both the model-based and model-free visual servoing methods. Indeed, it can also provide a 3-D mathematical (pre-operative) model of the ellipsoid in form of 3-D parameters and pose with respect to probe frame $\{R_s\}$; the 3-D parameters consist in the ellipsoid half-length values (a_1, a_2, and a_3), as formulated by (4.31) in Section 4.4.4. With those data, normal vector ${}^s\nabla\mathbf{F}$ can be obtained according to (4.32), and then can be used to compute interaction matrix $\mathbf{L_s}$ involved in the control law.

In the following, we firstly present results from simulation performed using the model-based visual servoing method, where the object pre-operative model is used to compute the control law. Such results are essential to test the validity of the theoretical foundations of the interaction matrix modeling, developed and presented in Chapter 3. Indeed, a pre-operative model provides us with a ground truth. ${}^s\nabla\mathbf{F}$ can be exactly known, and consequently no modeling error can be introduced in the interaction matrix formula (3.34) and (3.35). If the interaction matrix is truly exact, the visual features errors should converge to zero exponentially and at the same time. If however they do no converge as so, this would mean that the modeling is not exact. Afterwards, we present results obtained using the model-free visual servoing method, based on the curved lines estimation technique presented in Section 4.1.2, since this technique has shown to be better than the straight lines and quadric surface estimation techniques according to the results reported in the previous chapter. Nevertheless, visual servoing results with these two techniques can be found in Appendix C.1. We recall that the mode-free visual servoing we propose uses only the image contour points and the robot odometry to estimate the normal vector, and thus to compute the control law.

The following simulations are conducted with an ellipsoidal object whose half length values are $(a_1,\ a_2,\ a_3) = (1,\ 2.5,\ 4)$ cm. The control gain λ is set to 0.7, and the sampling time to 40 ms.

5.2.1 Model-based visual servoing

The ellipsoidal object, with which the probe is interacting, is exactly known. Both its half length values (a_1, a_2, a_3) and its pose with respect to $\{R_s\}$ are used to compute the exact value of ${}^s\nabla\mathbf{F}$, as related by (4.32). We first select the feedback visual features as $\mathbf{s} = (x_g,\ y_g,\ \alpha, \sqrt{a}, l_1)$. The corresponding simulation results are shown on Fig. 5.4. The

feedback visual features errors $\mathbf{e}$ exponentially converge to zero [see Fig. 5.4(f)], and the reached section image corresponds to the desired one (see Fig. 5.4(e)), despite the large difference between this target image and the initial one. Moreover, the probe motions are correct and smooth as can be seen on Fig. 5.4(g) and Fig. 5.4(a). Both the translational and rotational motions are large, as can be seen respectively on Fig. 5.4(c) and Fig. 5.4(d). These results, consequently, validate the proposed model-based visual servoing method. More particularly, they validate the theoretical foundations along with the interaction matrix modeling we developed and presented in Chapter 3.

With the above selected visual features $\mathbf{s}$, we can notice that the rotational motions are slightly coupled as can be seen on Fig. 5.4(d). We can remark indeed that the rotational motions $\theta \mathrm{u}_y$ and $\theta \mathrm{u}_z$ intersect[1]. The origin of that can be explained by the fact that the two last elements $\sqrt{a}$ and l_1 of $\mathbf{s}$ are not totally independent. Indeed, both these two features are related to the size of the section in the image. Let us therefore select another visual feature instead of l_1 that would yield the probe motions more decoupled. Since $\sqrt{a}$ relates solely the size of the section in the image, a prospective visual feature would be nothing but combination of moments invariants to image scale, as has already been highlighted and explained above in Section 5.1. We have already proposed ϕ_1 as fifth visual feature. Nevertheless, this feature shows to be relatively more sensitive to image noise than l_1, as will be seen later from results we present in this section. In fact, since the feature l_1 showed to be relatively robust to image noise, as will be also seen afterwards, we want to derive another feature close to l_1. Thus, the obtained feature might well satisfy both decoupling and robustness properties. Using the invariants presented in [50], we can deduce, that for example, the feature $l_1/\sqrt{a}$ is invariant to both in-plane motions and image scale. In the following, we present successively results obtained with $l_1/\sqrt{a}$ and then with ϕ_1, as fifth visual feature instead of l_1, to subsequently compare the corresponding performance.

In the same scenario in which the precedent simulation has been conducted, we select now $l_1/\sqrt{a}$ as the fifth visual feature, that is, $\mathbf{s} = (x_g,\ y_g,\ \alpha,\ \sqrt{a},\ l_1/\sqrt{a})$. The corresponding simulation results are shown on Fig. 5.5. We can see that the task has been well performed, as in the precedent simulation, where the feedback visual features errors converge to zero exponentially and the reached image corresponds to the desired one. Nevertheless, we can note that indeed, as expected, the rotational motions are relatively decoupled, even though slightly, comparing to those obtained with l_1 as fifth feature, as can be seen respectively on on Fig. 5.5(c) and Fig. 5.4(d). We can note indeed that the rotational motions $\theta \mathrm{u}_y$ and $\theta \mathrm{u}_z$ do not intersect; although their plots are close to each other during the first iterations.

[1] $\theta\mathbf{u}$ representation is defined by a unitary vector $\mathbf{u}$, representing the rotation axis, and rotation angle θ around this axis.

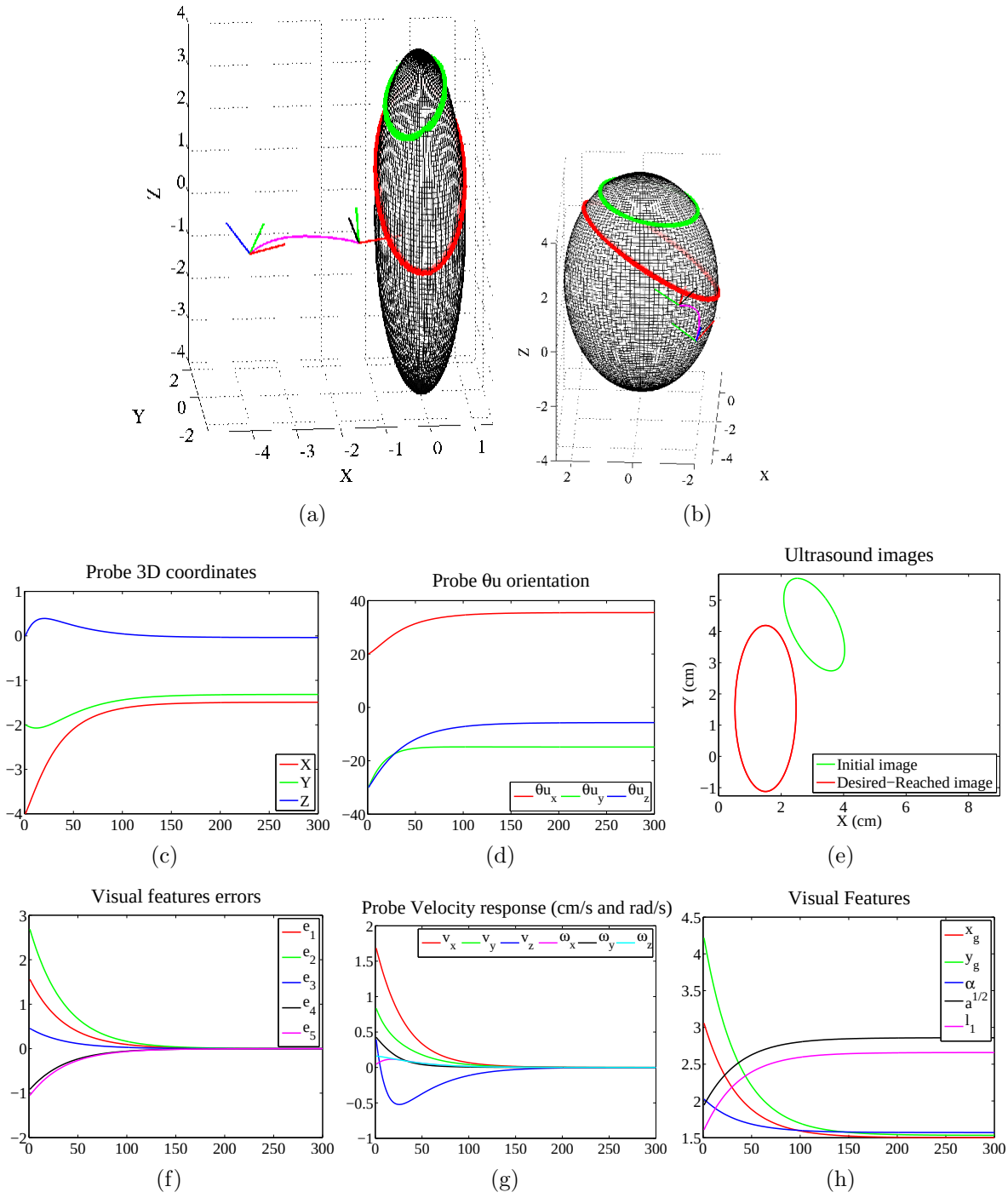


Figure 5.4: **Model-based** visual servoing on simulated ellipsoidal object. The visual features are $\mathbf{s} = (x_g,\ y_g,\ \alpha, \sqrt{a}, l_1)$. (a) and (b): The initial cross-section is plotted in green, while the reached one is plotted in red. The probe initial frame is depicted with the cartesian frame's $(X,\ Y,\ Z)$ axes respectively plotted with (red, green, blue) lines, while the final frame is plotted with (red, green, black) lines. The path performed by the probe is plotted in magenta. The visual features and their corresponding errors are in (cm, cm, rad, cm, cm). The abscissa of (c), (d), (f), (g), (h) corresponds to the number of iterations. It will be maintained as such for all the coming figures; this of course concerns the vectors that have been plotted on those indicated figures.

We finally test ϕ_1, that is, $\mathbf{s} = (x_g,\ y_g,\ \alpha,\ \sqrt{a},\ \phi_1)$. The corresponding simulation results are shown on Fig. 5.6. We can note that the decoupling performance on the rotational motions is better than those obtained either with l_1 or $l_1/\sqrt{a}$, as can be seen on Fig. 5.6(d). Indeed, we can note that the rotational motions $\theta \mathrm{u}_y$ and $\theta \mathrm{u}_z$ neither intersect nor are close to each other. The performance can also be clearly noticed from the plots of probe velocity shown on Fig. 5.6(g). Indeed, velocity component v_z converge with a considerably slight back-and-forth behavior during the first iterations, compared to the former obtained results with l_1 and $l_1/\sqrt{a}$.

The above three simulations have been conducted to compare the performance of the visual servoing schemes in terms of probe motions decoupling, more particularly this concerned the rotational motions. It is however important to compare their performances in terms of robustness to image noise, especially since the ultrasound images are inherently very noisy. To do so, we perform simulations with the three different visual servoing schemes, that is, the visual features vector's fifth element is respectively selected as l_1, $\frac{l_1}{\sqrt{a}}$, and ϕ_1 in the scenario where a measurement noise of 0.3 mm amplitude is present in the image. This noise is set as a random white Gaussian noise. The impact that such noise can have on the image coordinates of one point lying on contour $\mathcal{C}$ is shown on Fig. C.3(b). The corresponding simulations results are shown on Fig. 5.7. The obtained respective performances in terms of robustness to image noise are the inverse of those previously obtained in terms of motion decoupling. Indeed, we can see that feature l_1 is more robust to noise comparing to $\frac{l_1}{\sqrt{a}}$ and ϕ_1. The robustness is reflected on the performance of the visual servoing scheme in terms of probe behavior, as can be seen on the obtained velocity commands. This difference of robustness can be related to the denominators of these features. Feature ϕ_1 is less robust since its denominator is a second order moment; more the moment is of higher order less it is robust, as discussed in Chapter 3.

Finally, we can conclude that both of the simulation results we obtained and presented in this section validate the proposed model-based visual servoing method and its robustness to image noise. In the following, we test the model-free visual servoing method. In fact, from the interaction mathematical model used in the above simulations, we exploit this time only the image contour coordinates and the probe pose. We recall that the latter, in practice, can be obtained from the robot odometry. Thus, we do not use any prior knowledge of the shape, 3-D parameters, nor location (pose) of the object to compute the control law. The servoing method is tested both in a perfect case, where no noise is present, and in case where additive measurements noises are introduced.

Note however that since only five visual features are employed, although the reached image

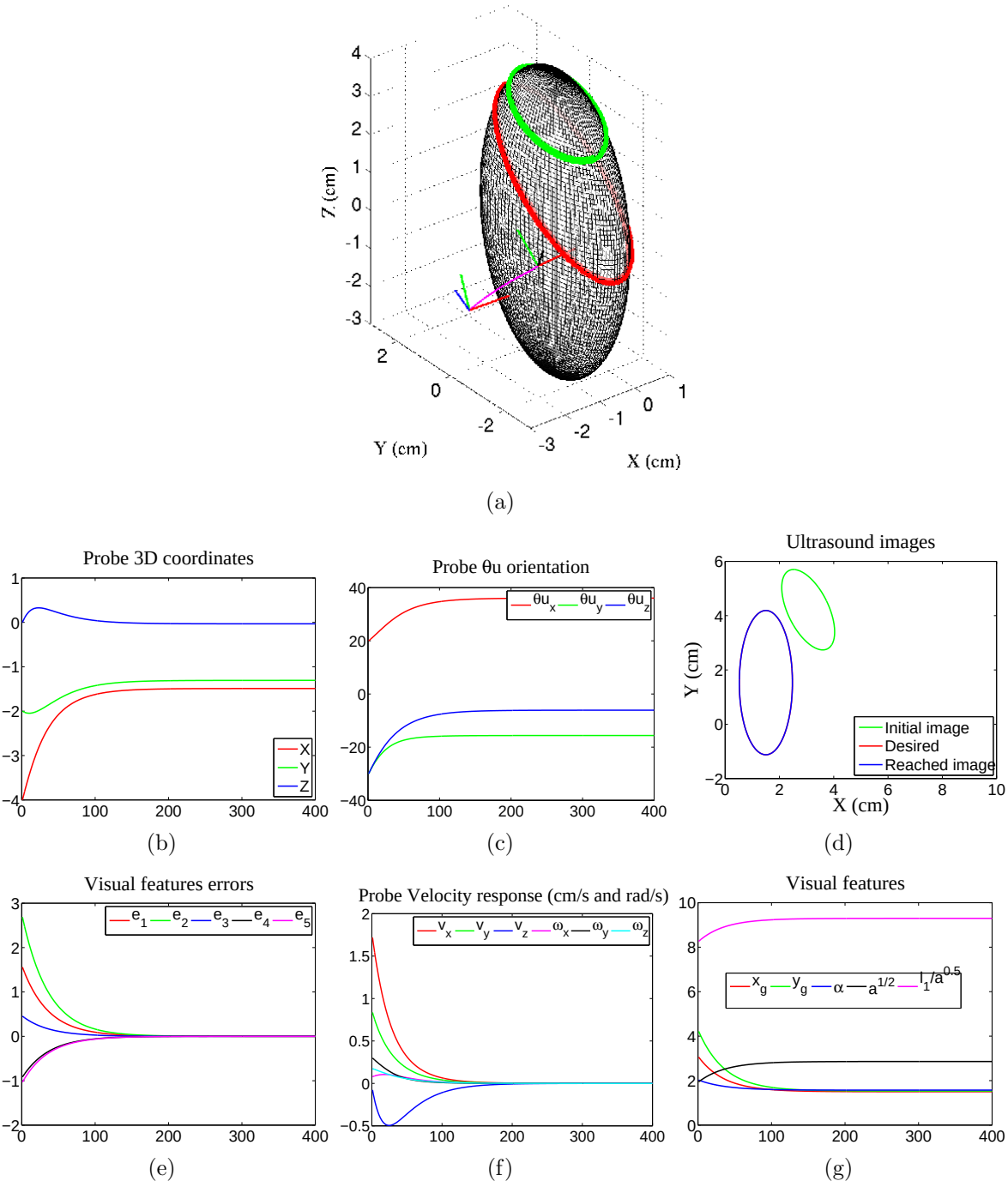


Figure 5.5: **Model-based** visual servoing on simulated ellipsoidal object. The visual features are $\mathbf{s} = (x_g,\ y_g,\ \alpha, \sqrt{a}, \frac{l_1}{\sqrt{a}})$. They are plotted in (cm, cm, rad, cm, unit/10), similarly as their corresponding errors.

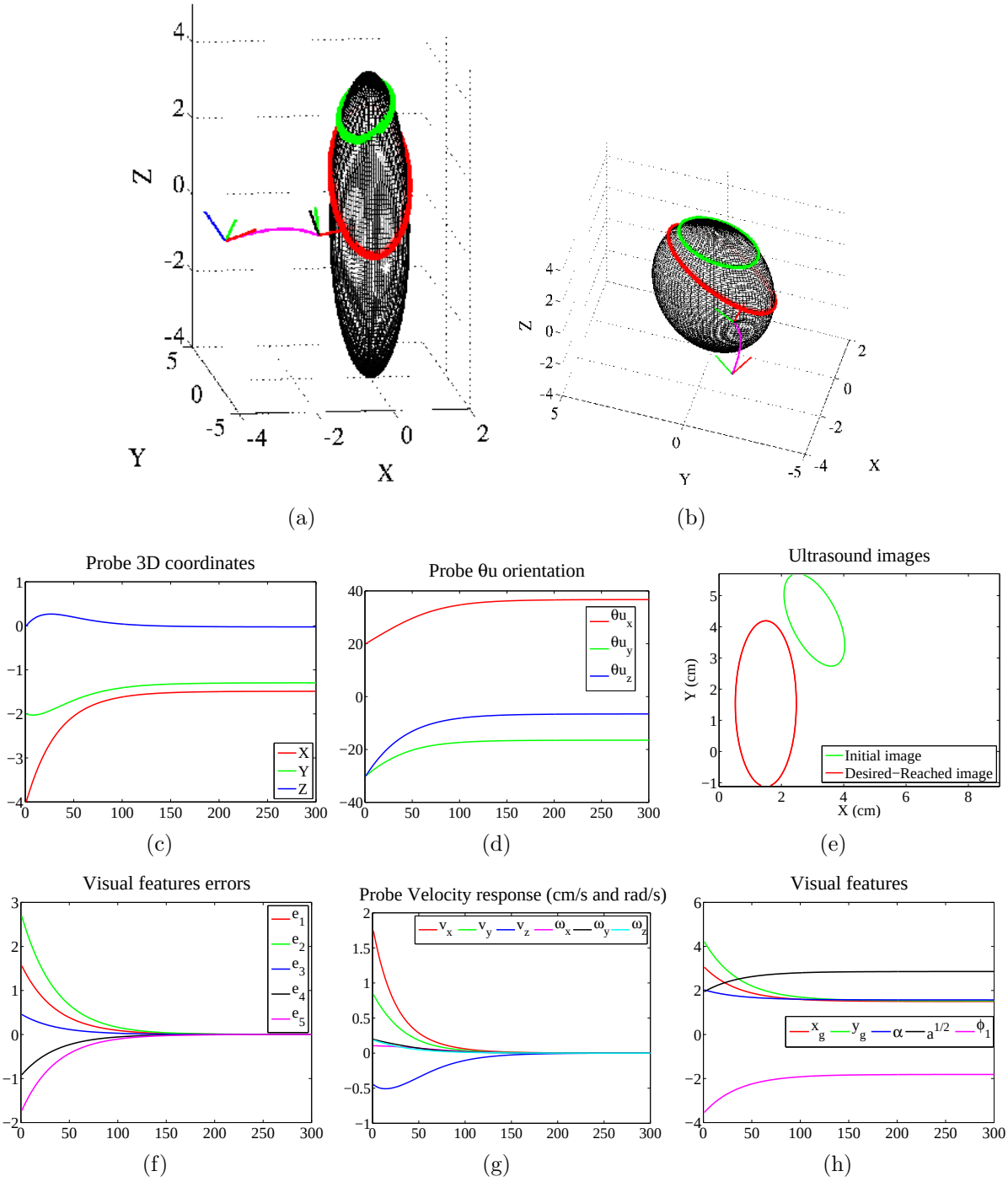


Figure 5.6: **Model-based** visual servoing on simulated ellipsoidal object. The visual features are $\mathbf{s} = (x_g,\ y_g,\ \alpha, \sqrt{a}, \phi_1)$. They are plotted in (cm, cm, rad, cm, unit/10), similarly as their corresponding errors.

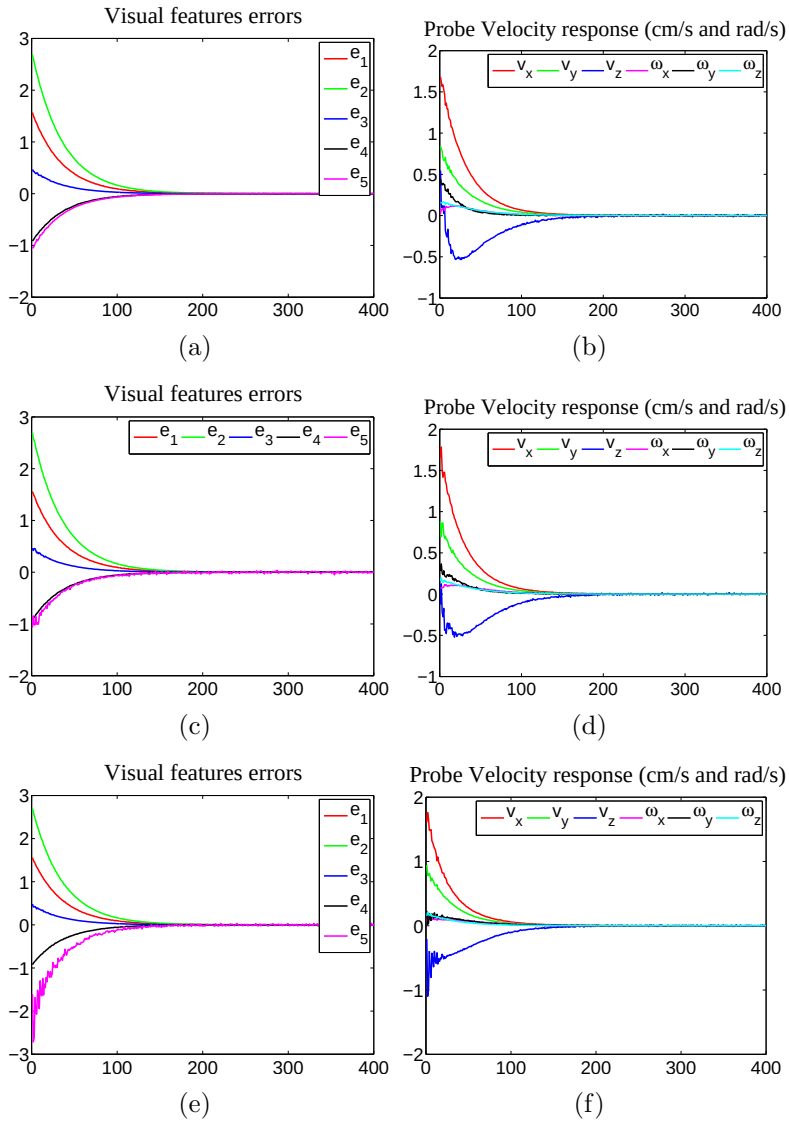


Figure 5.7: **Model-based** visual servoing on simulated ellipsoidal object, in the presence of additive measurement perturbations of 0.3 mm amplitude. The results obtained with l_1 as fifth feature are shown on (a) and (b) - Those obtained with $\frac{l_1}{\sqrt{a}}$ are shown on (c) and (d) - Those obtained with ϕ_1 are shown on (e) and (f).

corresponds to the desired one, the pose reached by the probe would unlikely correspond to that where the desired image had been captured. This can be explained by the fact that because of the object symmetry the probe can have an infinity of locations from which it can convey a same image. To control the 6 DOFs of the robotic system, and thus to automatically position the probe with respect to the observed object, at least six independent visual features are required. Of course, afterwards we present results obtained with an asymmetric object by controlling six visual features.

5.2.2 Model-free visual servoing using the curved line-based normal vector estimation

In the present section, we test the model-free servoing that uses the curved line-based normal vector on-line estimation method, described in Section 4.1.2.

The virtual probe is firstly moved in open-loop with constant velocity while at the same time the SLS algorithm, described in Section 4.3, is applied in order to obtain an initial estimate Θ_0. Note that this open-loop motion is applied for only the first N_{LS} iterations; in this case we set N_{LS} =20 iterations. Right after, the servoing is launched, where the recursive algorithm related by the relationships (4.8) and (4.16) takes place, instead of the SLS one, throughout the servoing. The estimator parameters are empirically tuned to $\beta = 0.9$, $f_0 = 5\times$1e3, $\beta_0 = \frac{1}{20\times f_0}$, and ϵ_0 =1e-10. The corresponding simulation results are shown on Fig. 5.8, while the estimated parameters are plotted in Fig. 5.9. We can see that the visual features errors exponentially converge to zero, and the reached image corresponds to the desired one. Also, correct and smooth probe behavior and motions have been obtained. These results thus validate the model-free visual servoing method that employs the curved line-based estimation. The plots of Fig. 5.9 in fact highlights the consistency of the estimated parameters between the whole points of the contour. Indeed, since the object surface is smooth (i. e., the partial derivatives of the surface are continuous), the variation of the normal vector when traveling along the object surface, and thus around contour $\mathcal{C}$ also, should be smooth, too; it is the case for the results we obtained. If it was not as such, this would mean that the normal vector is not well estimated. Doing so, that is, analyzing the consistency of the estimated parameters, could be therefore adopted as a first indicator about the estimation performance.

An additive measurement noise of 0.4 mm amplitude is now introduced in the image. The estimator parameters β and f_0 are now tuned to $\beta = 0.95$ and f_0 =1e2. Note that these parameters are adjusted to values slightly different from the previous ones, used above, only

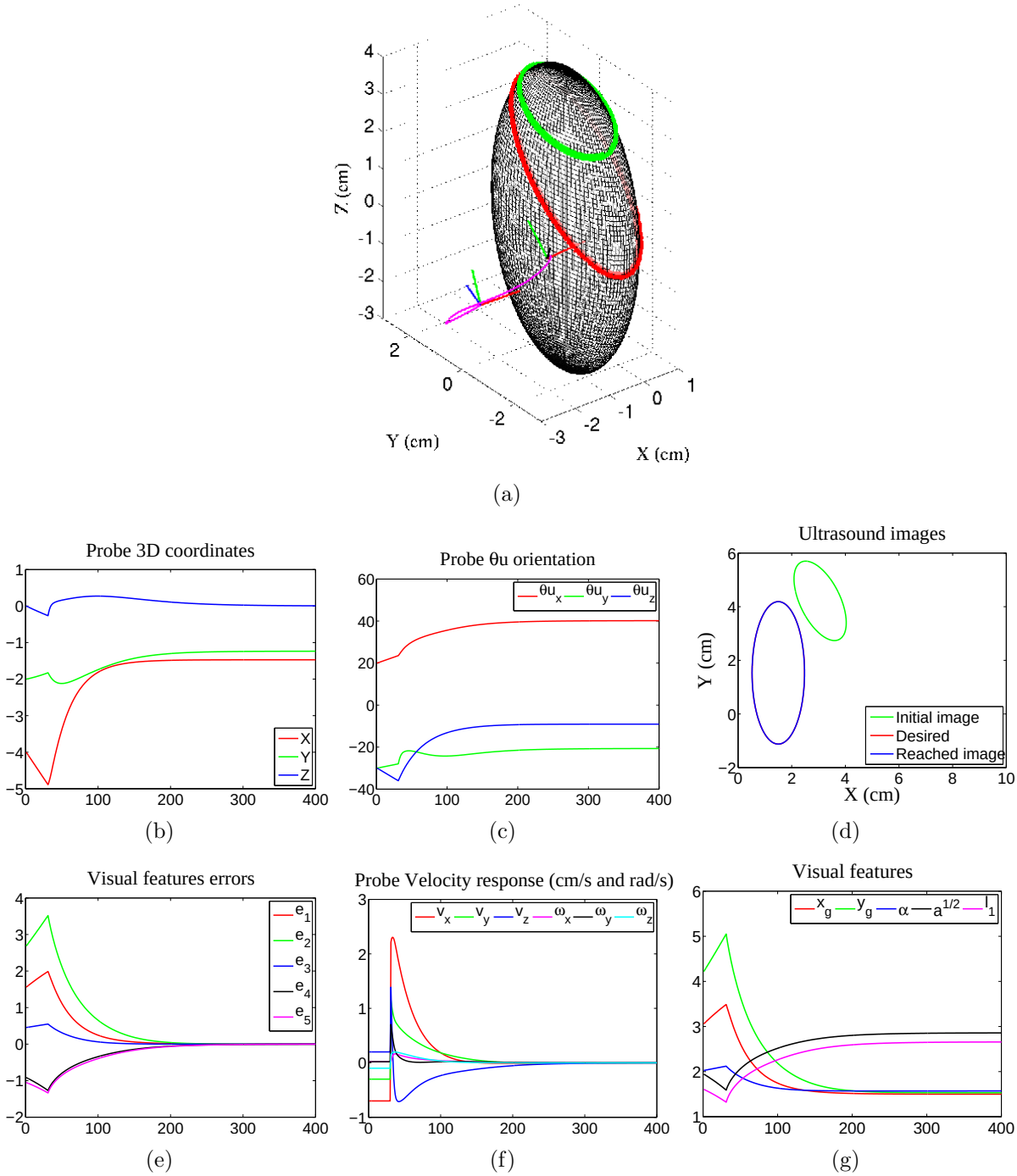


Figure 5.8: **Model-free** visual servoing using the **curved** line-based estimation method, in a perfect condition where no measurement noise is present. The visual features and their corresponding errors are in (cm, cm, rad, cm, cm).

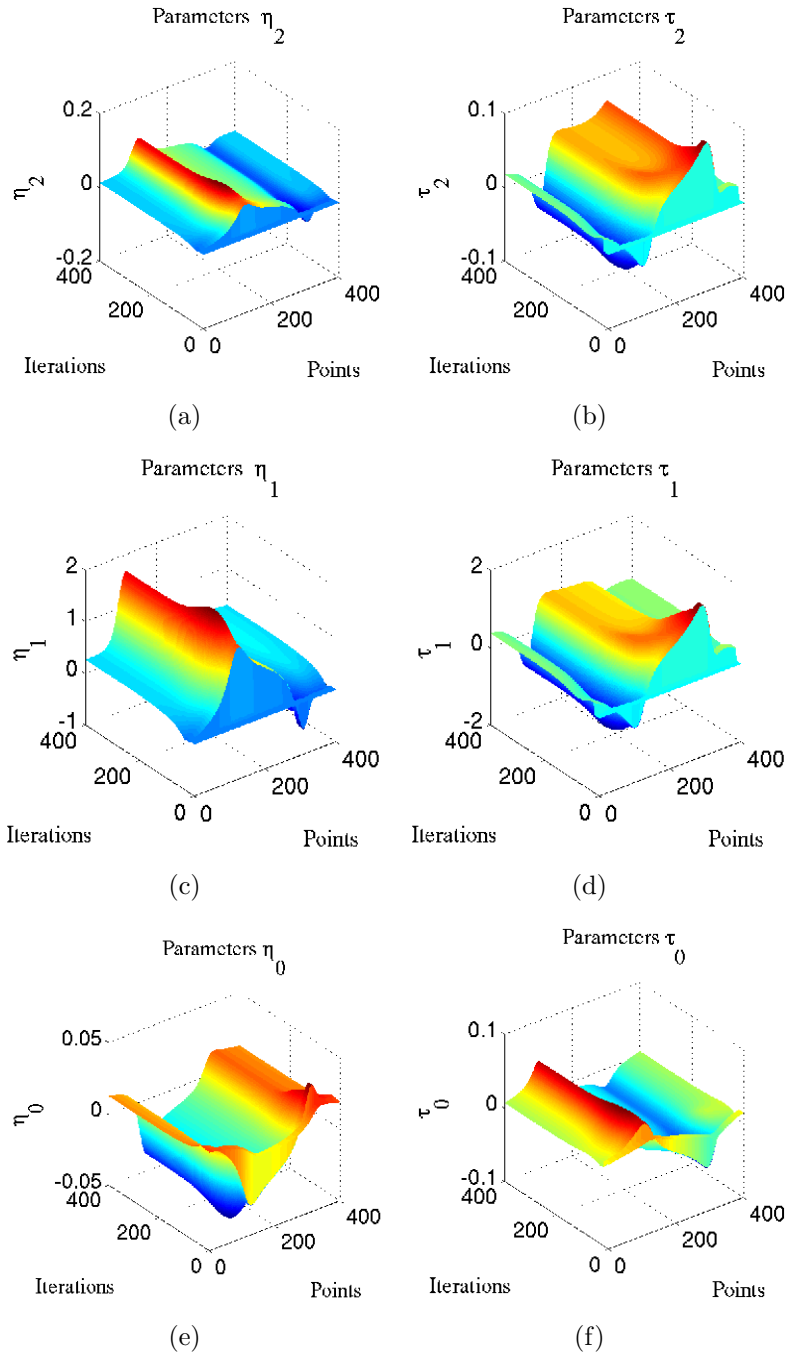


Figure 5.9: Estimated parameters $\hat{\Theta}$ corresponding to the results shown on Fig. 5.8.

at the aim to adapt the system to noises and, thus, it could behave better than if the previous parameters are used. The corresponding simulation results are shown on Fig. 5.10, and the estimated parameters on Fig. 5.11. We can see that the results are satisfactory, which validates the robustness of this model-free method with respect to measurement perturbations. Note that the system still well converged in perfect conditions with these values of the estimator parameters, but however the performance had slightly decreased. The simulations described below relate this. Note, nevertheless, that tangent vector ${}^s\mathbf{d_i}$, involved in the normal vector computation [relationship (4.1)], is in this simulation directly computed as pixel difference between the adjacent contour points; we recall that ${}^s\mathbf{d_i}$ corresponds to the vector tangent to contour $\mathcal{C}$ in the image. Performing directly a pixel difference is well-known to decrease the system stability. In practice and in more realistic simulations that we present afterwards, we do not compute ${}^s\mathbf{d_i}$ as such. We instead employ firstly an image processing algorithm to extract a contour of the section in the image. The extraction in fact consists to fit a parametric contour to the actual edge of the section in the image. Thus, the extracted contour would be filtered from eventual noises. We then compute ${}^s\mathbf{d_i}$ from that contour, thus mitigating the noise effect on the estimation. The system robustness therefore can only be expected better.

The results we showed are those we consider obtained using sufficiently well tuned estimator parameters. The tuning has been performed empirically, while making a compromise between estimation speed, accuracy, robustness to image noise; we modified the parameters according to a dichotomy manner. Nevertheless, we noticed that the system still converges and well behaves for different values of the parameters, and generally it was relatively easy to tune these latter. In fact the system performance is not dramatically compromised with parameters wise changes. To show this, we conducted different set of simulations, where in each set only one parameter is modified. In the first set we successively varied β. We present on Fig. 5.12 results separately obtained for $\beta = 1.0$, 0.5, and 0.04, while the remaining parameters are fixed throughout the tests to $f_0 = 5{\times}1e3$, $\beta_0 = \frac{1}{20\times f_0}$, and $\epsilon_0 =$1e-10. We can notice that when $\beta = 0.5$ the system performance is better. In the second set, the system is tested when starting with different values of initial estimate Θ_0. For that, we assigned different values to parameter ϵ_0, since the latter is involved in the SLS algorithm, that is employed to obtain Θ_0. The remaining parameters are fixed to $\beta = 0.9$, $f_0 = 5{\times}1e3$, and $\beta_0 = \frac{1}{20\times f_0}$. Results obtained for $\epsilon_0 =$1e-40, 1e-5, and 1.0 are shown on Fig. 5.13. We can conclude that the visual servoing system using the curved line-based estimation is quite tolerant to the values that the two parameters β and ϵ_0 might have. We also tested the system for different values of f_0 and N_{LS}. It was noticed that it grossly disclosed similar performances for different values of f_0, expect for very small ones, as 0.01 for example, where the convergence becomes relatively quite slow. The system also converged for different values of N_{LS}.

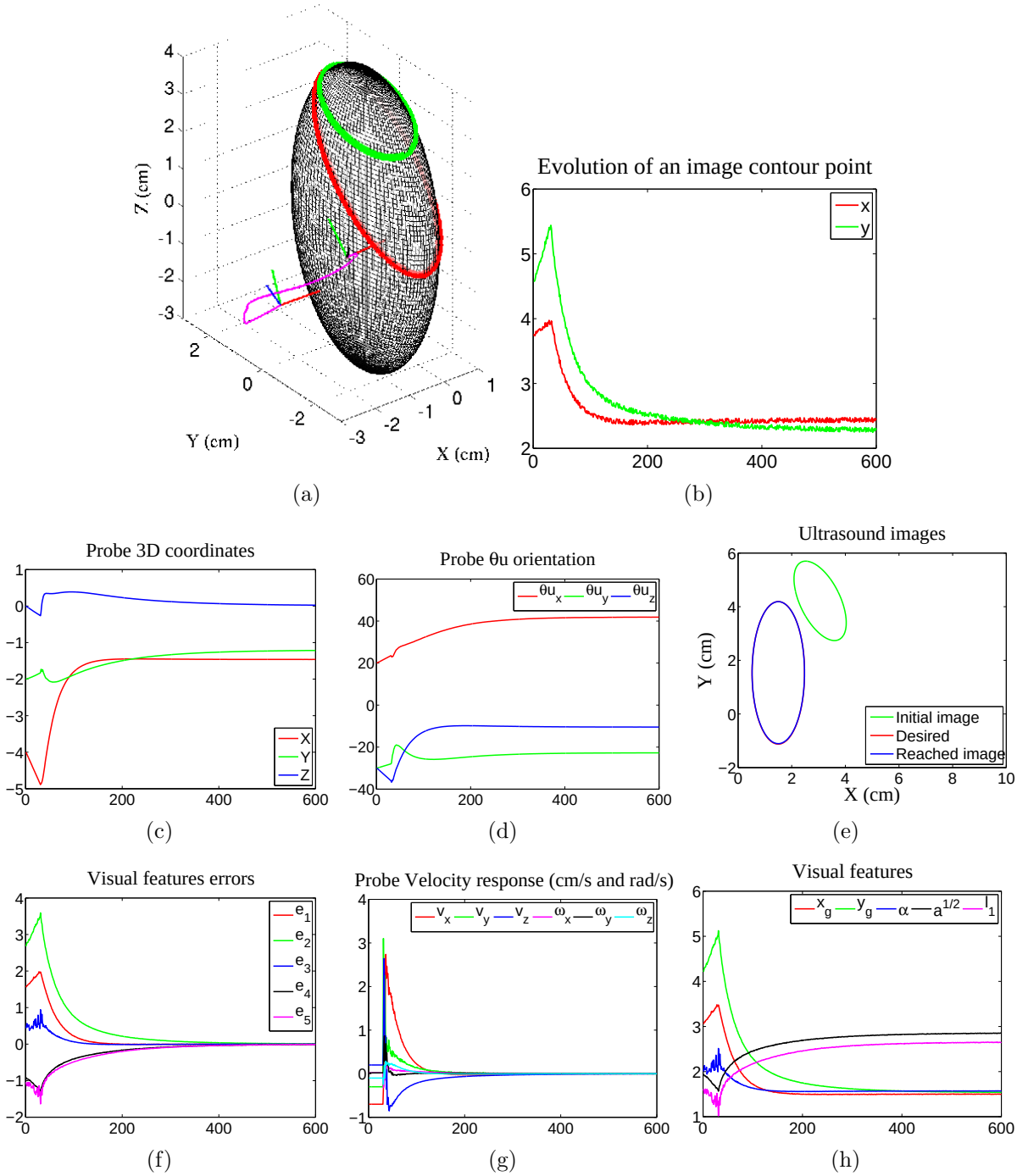


Figure 5.10: **Model-free** visual servoing using the **curved** line-based estimation method, in the presence of an additive measurement noise of 0.4 mm amplitude. The visual features and their corresponding errors are in (cm, cm, rad, cm, cm).

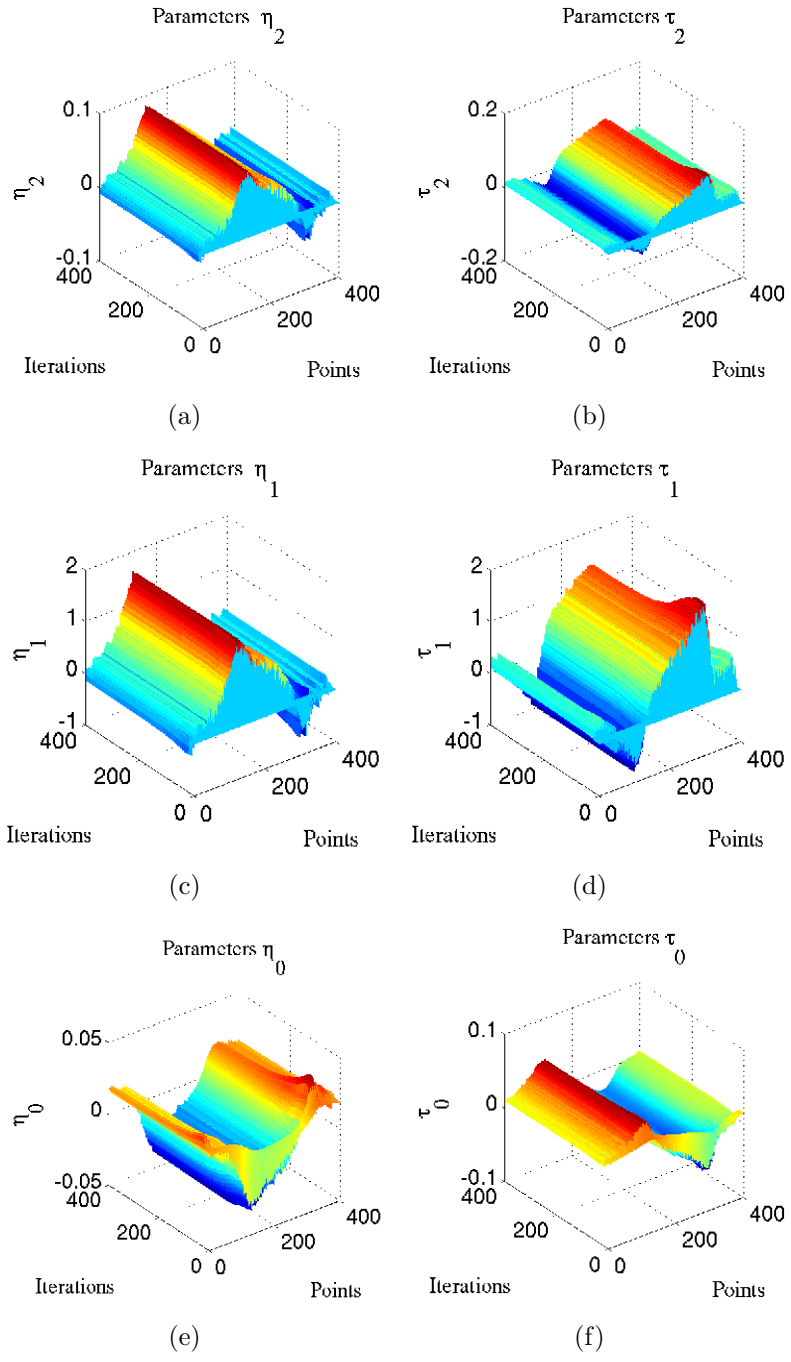


Figure 5.11: Estimated parameters $\hat{\Theta}$ corresponding to the results shown on Fig. 5.10.

We recall that results obtained with the straight line- and the quadric surface-based estimation methods are respectively presented in Section C.1.1 and C.1.2. We conclude that the quadric-based model-free servoing method considerably underperformed the two other methods. In addition, it was quite difficult to tune its estimation parameters. In fact, we are not surprised about this outcome, because of the low performances this estimation method had already shown in Section 4.4.4. The other drawback of this method, as highlighted in the previous chapter, consists in the fact that it is relatively computationally expensive. Indeed, this method uses at each iteration a segment of N points to estimate and thus to update normal vector ${}^{s}\nabla\mathbf{F}$, in contrast to the two other methods (respectively based on straight and curved line estimation) where only one point is used to update the estimation.

5.3 Simulation results with realistic ultrasound images

In the present section, the curved line-based model-free visual servoing method is tested on a realistic simulated object. The latter consists in an ultrasound image volume, made from a previously performed scan of an ultrasound phantom containing an egg-shaped object. The scan has been performed by acquiring 100 real B-scan images with a conventional 2D ultrasound probe, that swept the phantom by moving with constant velocity along its orthogonal Z axis. The images were successively captured at each 0.1 mm interval of the probe motion. Using a software presented in [45], the interaction of a virtual 2D ultrasound probe with the object volume is simulated. This software simulator is built from the Visualization Toolkit (VTK) software [70] system and ViSP [53]. It allows to move and position the probe, and can provide the corresponding realistic ultrasound image along with a 3D view of the interaction, as can be seen respectively on Fig. 5.14(a) and Fig. 5.14(b). In the following, we test the servoing method, by using this simulator. This allows us to verify its validity on realistic ultrasound images. The method uses only the observed image and the probe pose (robot odometry), also provided by the simulator, to compute the control law. The latter is then applied on the virtual probe that moves accordingly. However, we need to extract from the observed image the section contour, since it is used to compute the feedback visual features and the interaction matrix, both involved in the control law. 2D ultrasound images are, yet, very noisy and difficult to segment. Moreover, such extraction should be not time consuming, but it should instead be performed as fast as possible in a duration within the real-time servoing streaming rate. This latter constraint is more difficult to satisfy when this concerns robotic applications, because of the high streaming rate

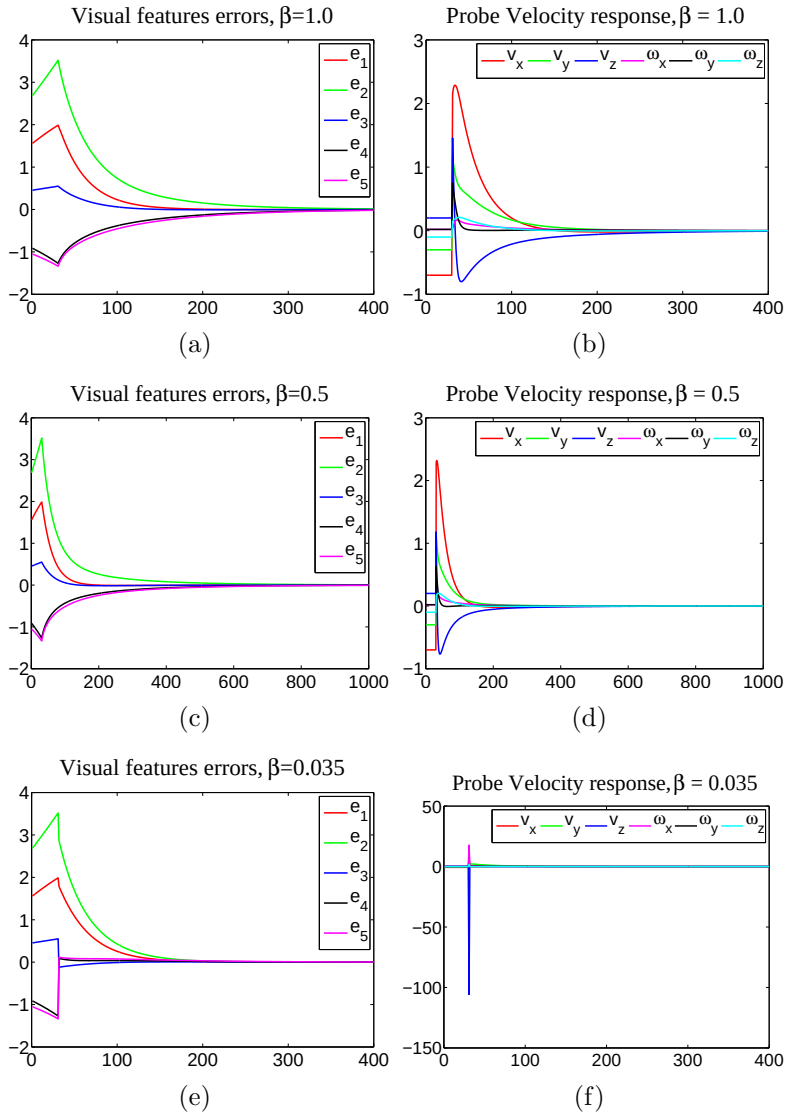


Figure 5.12: Results obtained by employing the **model-free** visual servoing using the **curved** line-based estimation for different values of the parameter β. The visual features errors are in (cm, cm , rad, cm, cm), and the probe velocity is in (cm/s and rad/s).

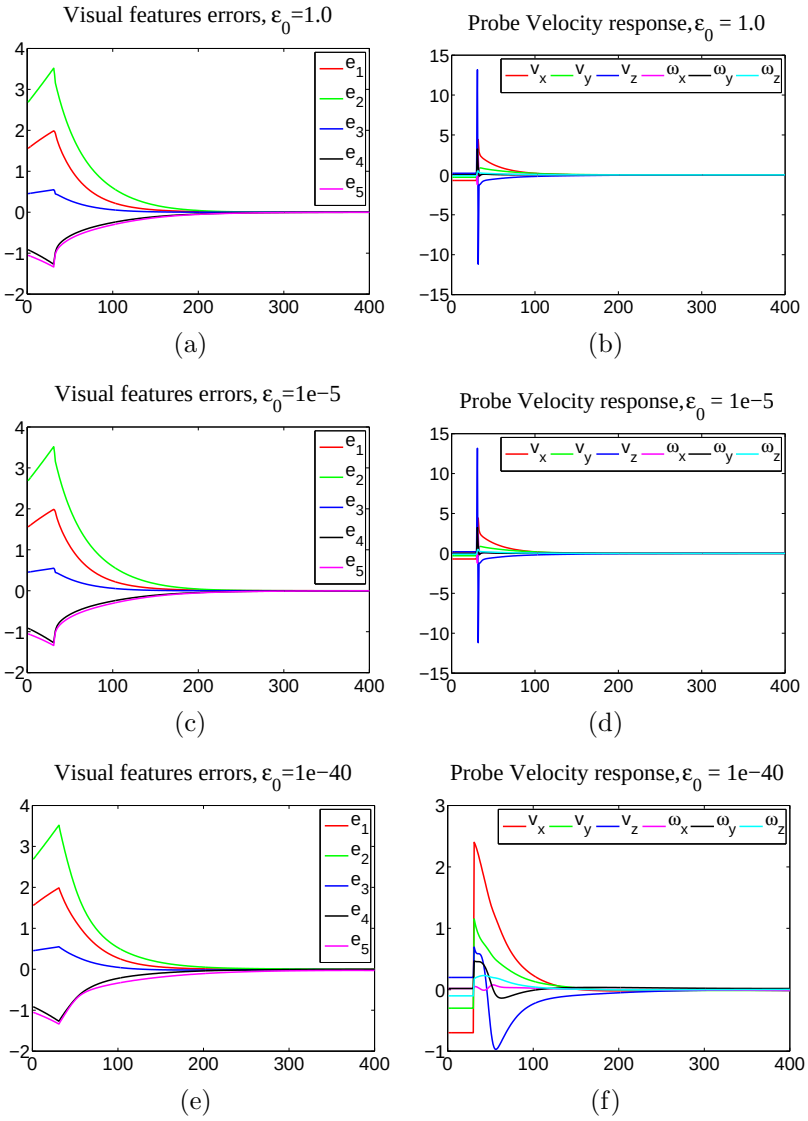
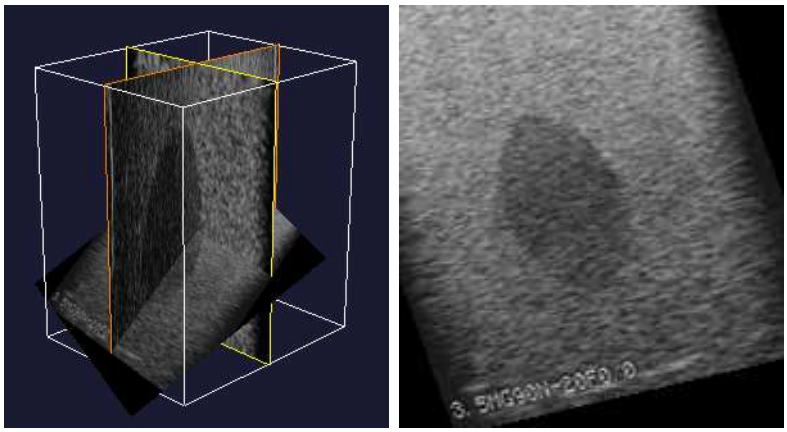


Figure 5.13: Results obtained by employing the **model-free** visual servoing using the **curved** line-based estimation for different values of the parameter ϵ_0. The visual features errors are in (cm, cm , rad, cm, cm), and the probe velocity is in (cm/s and rad/s).



(a) (b)

Figure 5.14: Simulating the interaction of a virtual 2D ultrasound probe with a real ultrasound 3D volume - (a) A 3D view of the probe observation plane intersecting (observing) the egg-shaped object - (b) Observed 2D ultrasound image.

at which the systems perform. If this constraint is not satisfied, the system performance would be totally compromised and, even more, its stability would be deeply threatened. We use the image processing algorithm presented in [21] to segment and track the section in the ultrasound image. This algorithm is based on a snake approach, and a polar parametrization to model the contour. It has shown to be relatively fast. Since image processing is beyond the scope of this thesis, it is not detailed in this document. Note that this algorithm is employed in all the simulations and experiments presented in the remainder of this chapter.

The segmentation provides coordinates of points lying on image section contour $\mathcal{C}$, as can be seen as instance on Fig. 5.15. These points, more precisely their image 2D coordinates, are then used to compute feedback visual features vector $\mathbf{s}$, on-line estimate normal vector ${}^{s}\nabla\mathbf{F}$, and finally compute the control law (5.2). Note however that, in this case, the observed section in the image is nearly an ellipse [see Fig. 5.14(b) for example]. Consequently, we can define only five independent visual features from the image. We select them, similarly as in Section 5.2.2 (and also Section C.1.1 and C.1.2), $\mathbf{s} = (x_g, y_g, \alpha, \sqrt{a}, l_1)$. Accordingly, pseudo inverse $\mathbf{L}_\mathbf{s}^+$ is employed in (5.2) instead on inverse $\mathbf{L}_\mathbf{s}^{-1}$, since matrix $\mathbf{L}_\mathbf{s}$ is not square in this case.

The task that has to be performed by the virtual probe consists to automatically reach a first desired image starting from one different, and then to reach a second target. This allows us to verify that the recursive algorithm can re-estimate ${}^{s}\nabla\mathbf{F}$, after the observed image has not conveyed wealthy information during a while. Indeed, when the first target image

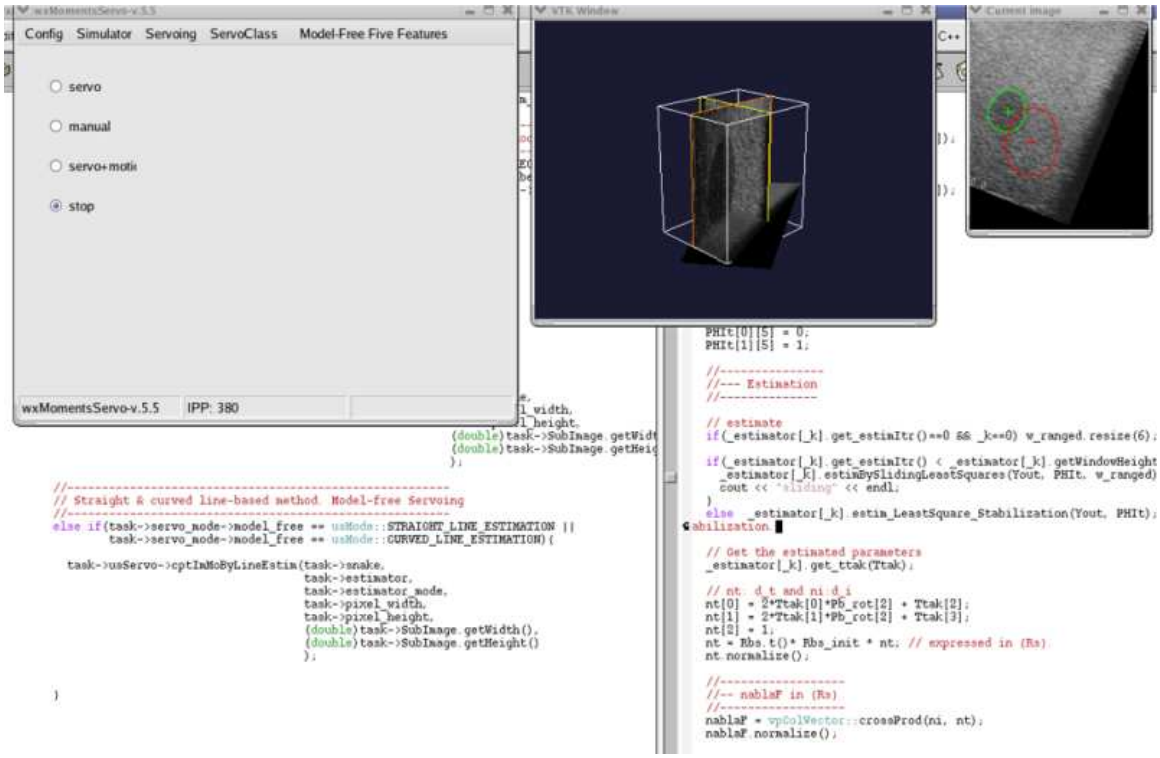


Figure 5.15: A Screenshot of the graphical human-machine interface (top left), along with a 3D view of the interaction between the virtual probe plane with a realistic object (top middle), and also along with the observed image whose section is contoured with green and where the contour of the target image section is displayed in red (right). Right the user would have pushed the button "servo" (round button at top left), the servoing would be launched.

would have been reached, the probe would stand roughly motionless until the second target would be sent to the controller. During that time span, the observed image is roughly the same and, as consequent, there would not be information to stimulate the recursive estimator. This might yield the covariance matrix $\mathbf{F}_{[k]}$ ill-conditioned, thus compromising the estimation. Moreover, the algorithm might be trapped and might not be pulsed even though new images would then convey wealthy information. However, thanks to stabilization term $(1-\beta)\,\beta_0\,\mathbf{I}$ introduced both in the recursive relationships (4.9) and (4.16), it is expected that covariance matrix $\mathbf{F}_{[k]}$ is prevented from becoming ill-conditioned when there are not enough probe motions.

The simulation scenario consists to first position the probe on an image [see the section image contoured in green, shown on Fig. 5.16(a)] totally different from both the two targets.

Then, it is moved in open-loop with constant velocity $\mathbf{v} = (\ -0.4,\ 0,\ -0.3,\ 0,\ 0,\ 0)$ (cm/s and rad/s) during the first N_{LS} iterations, where the SLS algorithm is being applied in order to obtain an initial estimate Θ_0. Right after, the servoing is launched where the recursive algorithm takes place instead of the SLS one. The recursive algorithm is solely applied throughout the servoing. The control gain is set to $\lambda = 0.7$. The first initial estimate, before that the SLS algorithm is applied, is arbitrarily set to $\Theta_{[t_0]} = \mathbf{0}$ ($\mathbf{0}_4$ when using the straight line-based estimation, and $\mathbf{0}_6$ when either using the curved line- or the quadric surface-based estimation). The estimator parameters are tuned to $\beta = 0.9$, $f_0 = 5{\times}1e3$, $\beta_0 = \frac{1}{20\times f_0}$, ϵ_0 =1e-10, and $N_{LS} = 20$ iterations. The corresponding results are shown on Fig. 5.16 and Fig. 5.17. We can see that the successive reached images correspond to the desired ones, and the visual errors converge to zero. These results thus show the validity of the curved line-base model-free visual servoing method on realistic ultrasound images. Moreover, they show its robustness as can be clearly seen how much the images are of low quality. Due to the fact that the snake shook when tracking the actual section contour, because of the very noisy images and since the section is low contrasted from the image background, the probe velocity has consequently not been smooth, as can be seen on Fig. 5.16(f). Using a more powerful contour detection would undoubtedly, and perhaps considerably, improve the system behavior.

Results obtained using the straight line- and the quadric surface-based estimation methods are respectively reported in Section C.2.1 and C.2.2. As expected (see for example Section C.1.2), the latter method again underperformed the two other methods.

The trials presented so far were able to use only five independent visual features in the visual servoing scheme. This was due to the fact that the section in the image was roughly ellipsoid, that is, symmetric. In such cases, however, although the desired section in the image is reached, the pose reached by the probe would unlikely correspond to the desired pose (i. e., pose where the target image had been captured). The reason is that when the image is symmetric (i. e., the object is symmetric) a desired image can correspond to an infinity of probe poses. In fact, to be able to reach a desired pose using the image, at least six independent visual features are required to control the 6 DOFs of the robotic system. In the next section, we perform simulations on a virtual object which is grossly non-symmetric, such that the six chosen visual features are all independent.

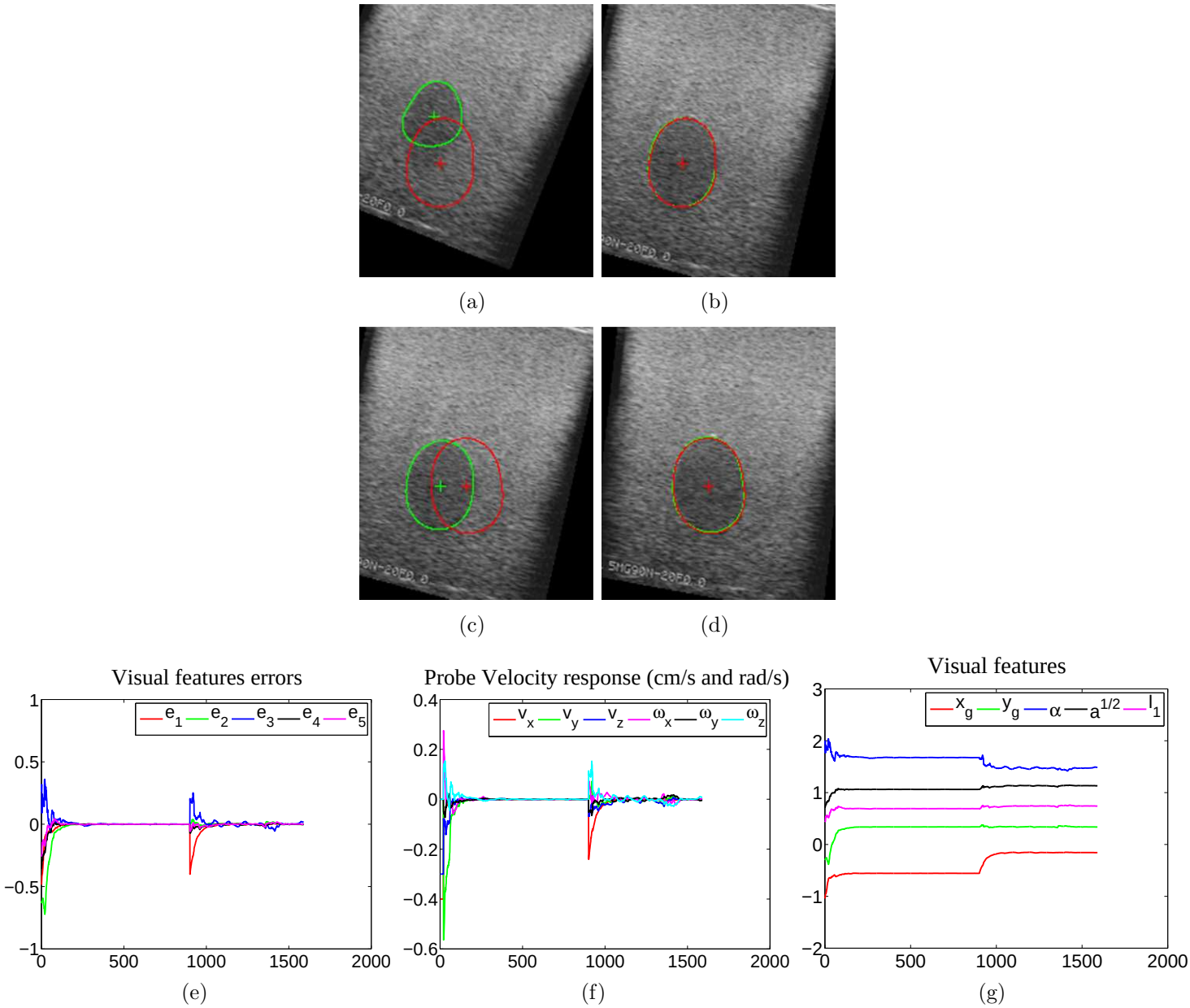


Figure 5.16: **Model-free** visual servoing using the **curved** line-based estimation method performed on a realistic ultrasound 3D volume. The visual features and their corresponding errors are in (cm, cm, rad, cm, cm).

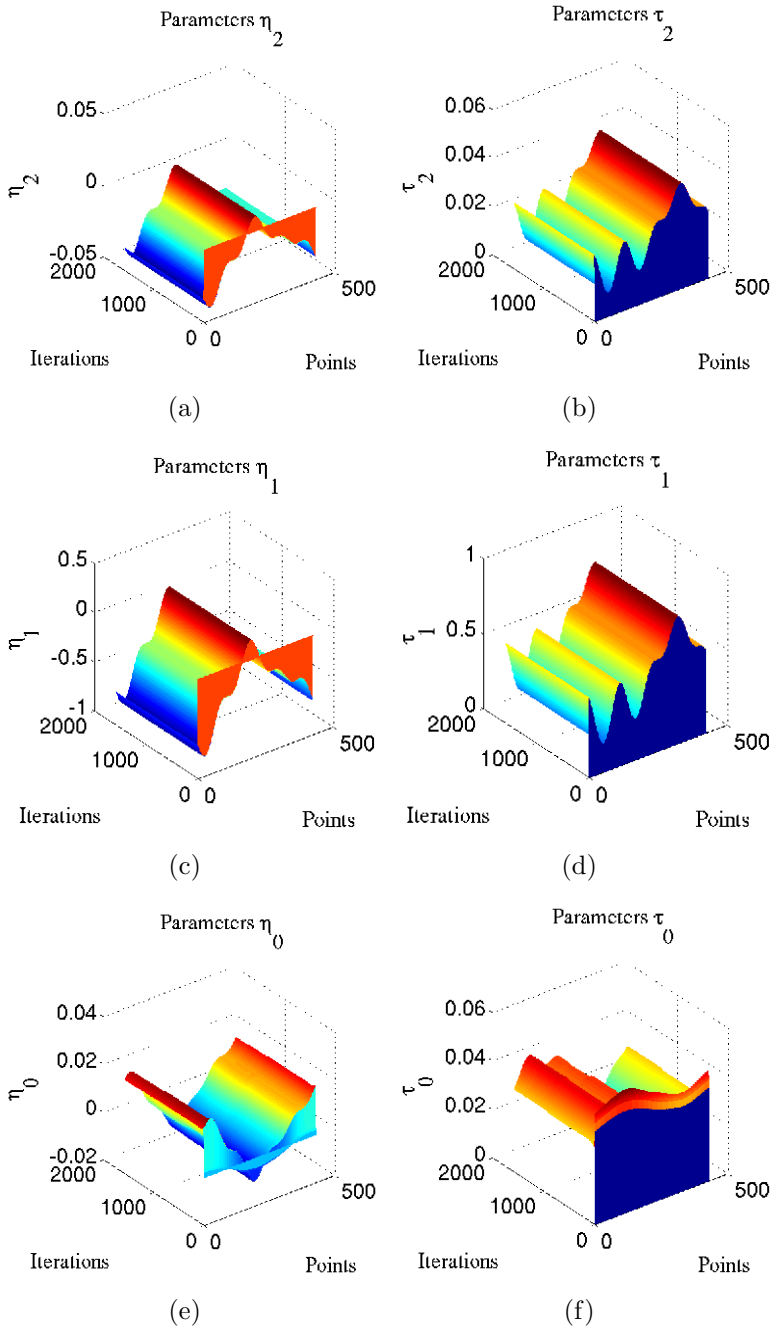


Figure 5.17: Estimated Parameters $\hat{\Theta}$ corresponding to the results shown on Fig. 5.16.

5.4 Simulation results with a binary object

The curved line-based model-free visual servoing method is now tested on a virtual binary object, which is grossly asymmetric. The selected features are $\mathbf{s} = (x_g,\ y_g,\ \alpha,\ \sqrt{a},\ l_1,\ \phi_2)$. Note that in this case $\mathbf{L_s}$ is a 6×6 matrix and, thus, a square matrix. Therefore, we can directly employ the control law (5.2) as is. The task now consists, besides of reaching the desired image, to also reach the pose where that image had been captured. We use for the simulations the same software described in Section 5.3, but, which is now loaded with slice images of the binary object. We similarly load 100 slices.

The scenario is similar to that described in Section 5.3. Two target images are successively sent to the visual servoing system, where the latter target is ordered after the former would have been automatically reached. At initial time t_0, the probe is positioned by the user at a pose different from those where the two target images had been captured. Then, it is moved in open-loop with constant velocity $\mathbf{v} = (0,\ -0.1,\ 0.12,\ 0,\ 0,\ 0)$ (cm/s and rad/s) for the first 100 iterations. During that time, the SLS algorithm is applied in order to obtain initial estimate Θ_0. Before the open-loop motion is performed, initial estimate $\hat{\Theta}_{[t_0]}$ is arbitrarily set $\hat{\Theta}_{[t_0]} = \mathbf{0}$. Note, however, that this open-loop motion yields the probe (and thus the actual image) more farther from both the first and second targets. At the end of this motion, the corresponding pose represents that from which the model-free visual servoing is launched. As for the detection and tracking of the contour, which consists to extract the 2D image coordinates of points lying on it, and which is required to compute the control law, we similarly use the snake detection algorithm also introduced in Section 5.3. The control gain is set to $\lambda = 0.2$. The estimator parameters are tuned to $\beta = 0.8$, f_0 =1e6, $\beta_0 = \frac{1}{20 \times f_0}$, ϵ_0 =1e-10, and $N_{LS} = 20$ iterations. The corresponding simulation results are shown on Fig. 5.18 and 5.19. They are quite satisfactory, since the visual features errors converge to zero, exponentially. The two poses reached by the probe correspond also to those where the first and second target images had been captured, respectively. The obtained positioning errors are (1.28×1e-3, -8.4×1e-4, -1.9×1e-4, 0.086, 0.378, 0.03)(cm and deg) for the former and (4.5×1e-4, 4.1×1e-6, -1.13×1e-5, -0.12, -0.22, 0.008)(cm and deg) for the latter automatic positioning. These results show the validity of the method in automatically positioning the probe with respect to an observed object. They also show the relevance of the selected six visual features to control the 6 DOFs of the system.

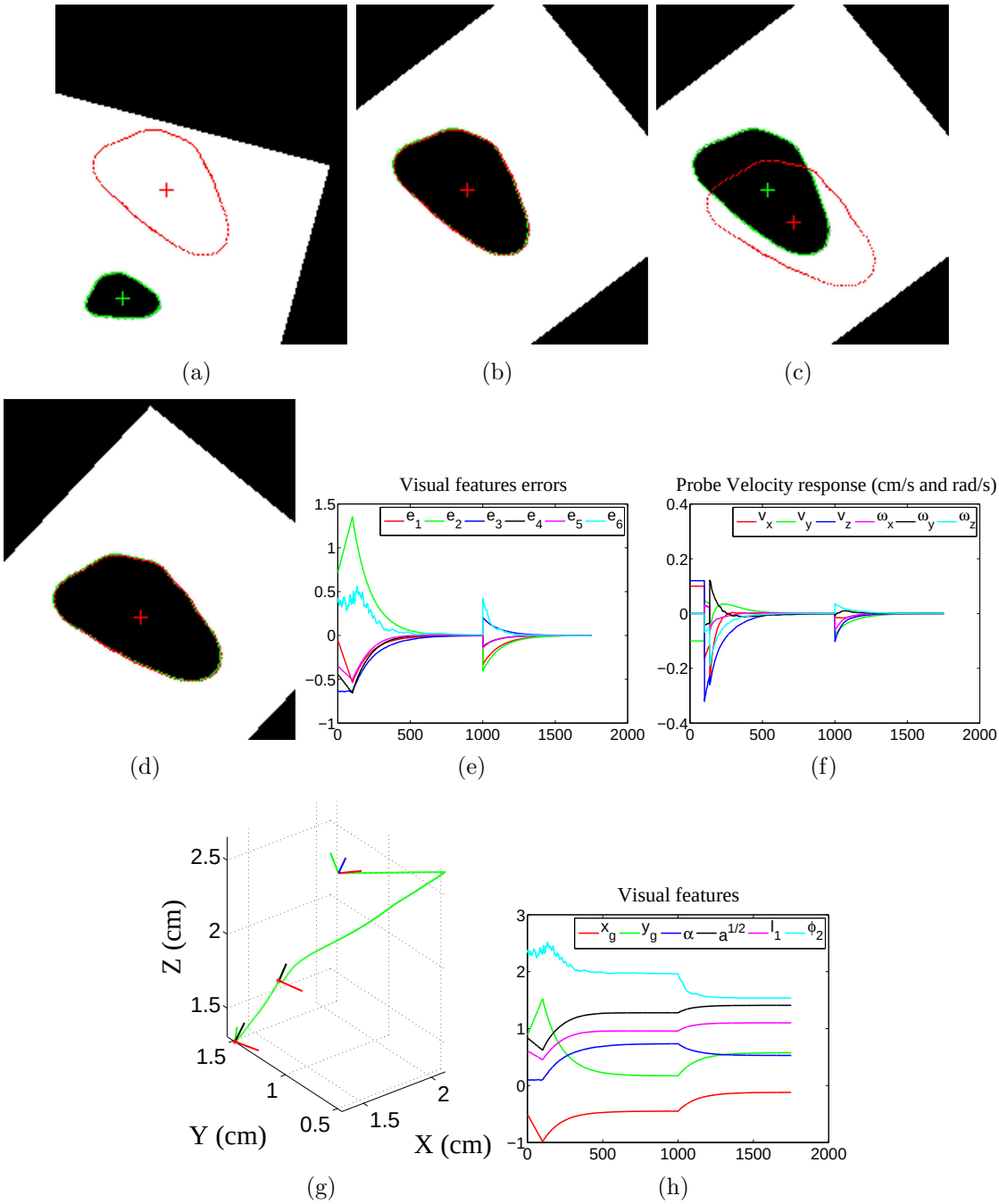


Figure 5.18: **Model-free** visual servoing that uses the **curved** line-based estimation, tested on a simulated binary object.

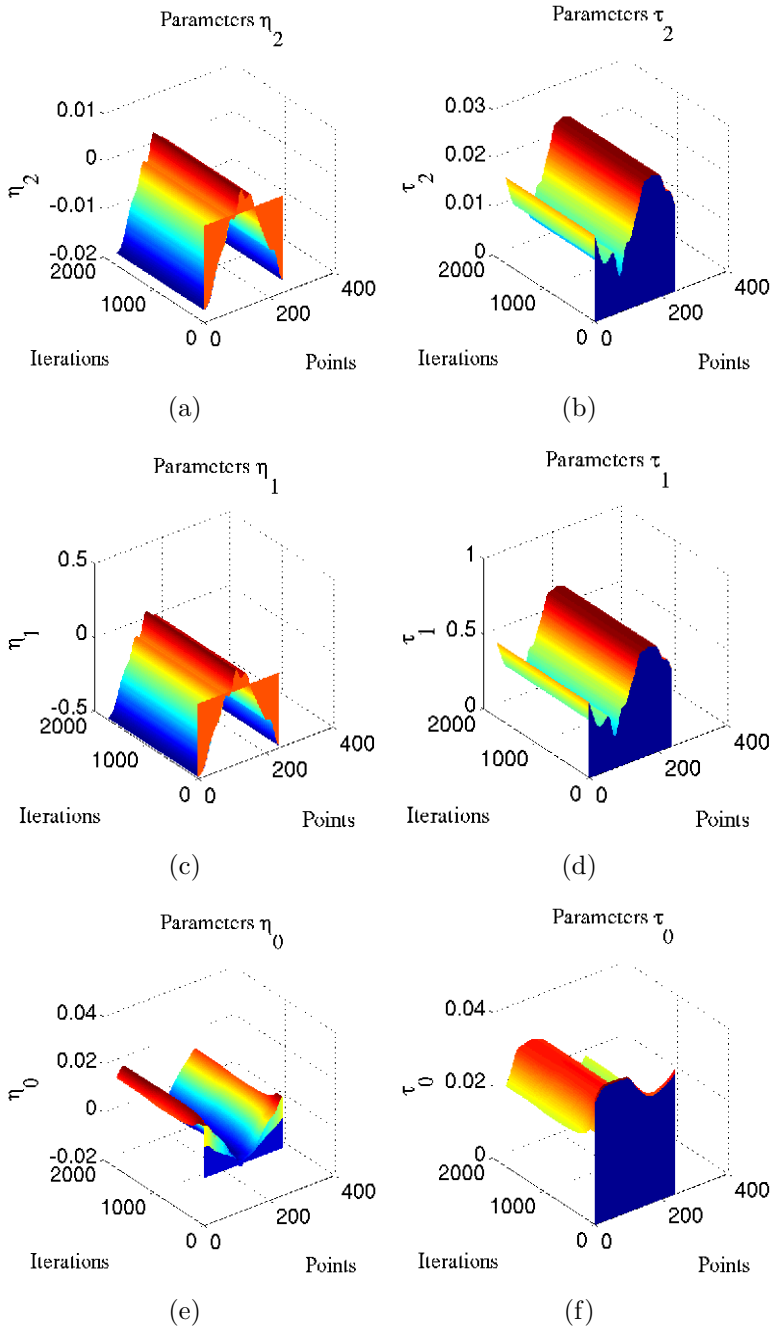


Figure 5.19: Estimated parameters $\hat{\Theta}$ corresponding to the results shown on Fig. 5.18.

Figure 5.20: Experimental setup - A 6 DOFs medical robot arm (right) actuating a 2D ultrasound probe (left), which is interacting with an object immersed in a water-filled tank. The observed image is displayed on the imaging system screen (middle).

5.5 Experimental results

In the following, we finally present experimental results obtained with the model-free visual servoing that uses the line-based estimation. We employ 6 DOFs anthropomorphic robot arms. All the experiments have been conducted with a medical robot arm similar to the Hippocrate robotic system [65], except the last one where a new acquired robot has been employed as presented in Section 5.5.5. The robot carries at its end-effector a 5-2 MHz 2D broadband US transducer (see Fig. 5.20 for example). The latter acquires the images at a streaming rate of 25 frames/s. A block diagram shown on Fig. 5.21 illustrates the different steps involved in the servoing along with the corresponding data flow. The servoing method has been implemented in the C++ programming language under LINUX operating system, and the control law is computed using an ordinary personal computer. We consider first a simple case of a spherical object with which the probe is interacting, the case of a relatively symmetric object enclosed in an ultrasound phantom, and then a more complex case of non-symmetrical soft tissue object. Both the spherical and soft tissue objects are separately immersed in a water-filled tank. The latter experiment allows us to experimentally test the automatic positioning with respect to an observed object and, thus, the validity of the model-free visual servoing method in controlling the 6 DOFs of the robotic system. We conclude these tests by carrying out an experiment where we take back the ultrasound

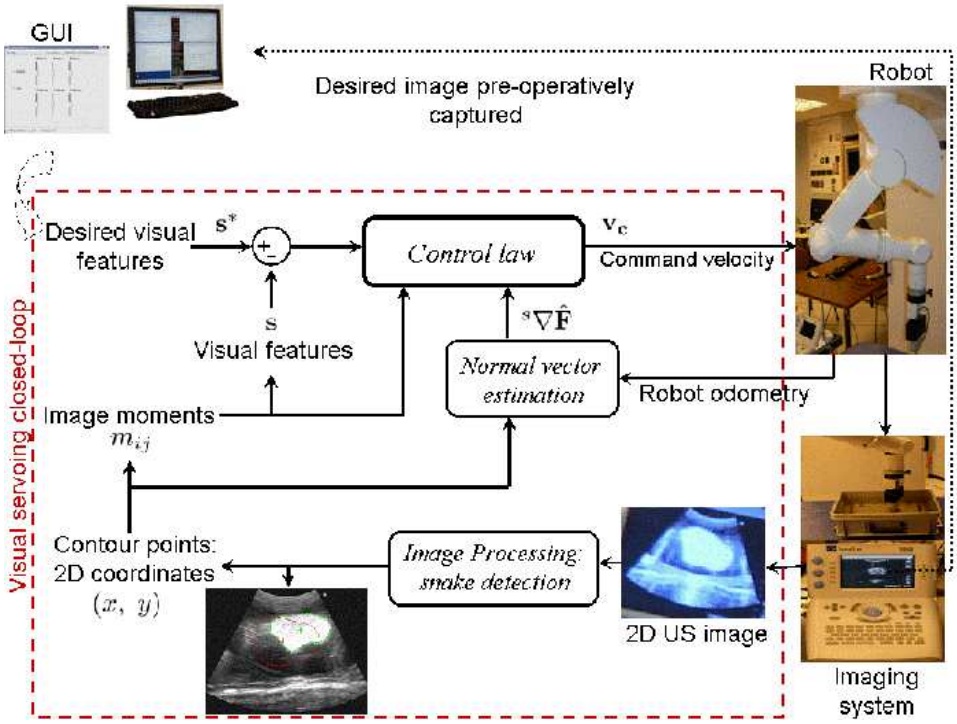


Figure 5.21: Architecture of the model-free servoing method, where the different involved steps along with the corresponding data flow, up to hardware setups, are presented.

phantom. In the latter, in fact, two relatively-symmetric objects can be observed in a same acquired image. The robotic task of this experiment consists in tracking both the two sections, instead of only one. We will show that by doing so the probe can be positioned and thus stabilized with respect to the two objects, although the symmetry of each one. Therefore, we provide a solution to address the problem of symmetry, pointed out in this document.

5.5.1 Experimental results with a spherical object

The robotic system is interacting with a ping-pong ball of 4 cm diameter. Note that we do not use any prior knowledge about the ball in the servoing. No information about its diameter nor location is exploited. Since the observed image is a sphere, we can define only three independent visual features, as has been described in Section 3.7.1. Therefore, the feedback visual features vector we select is $\mathbf{s} = (x_g, y_g, \sqrt{a})$, where its elements have already

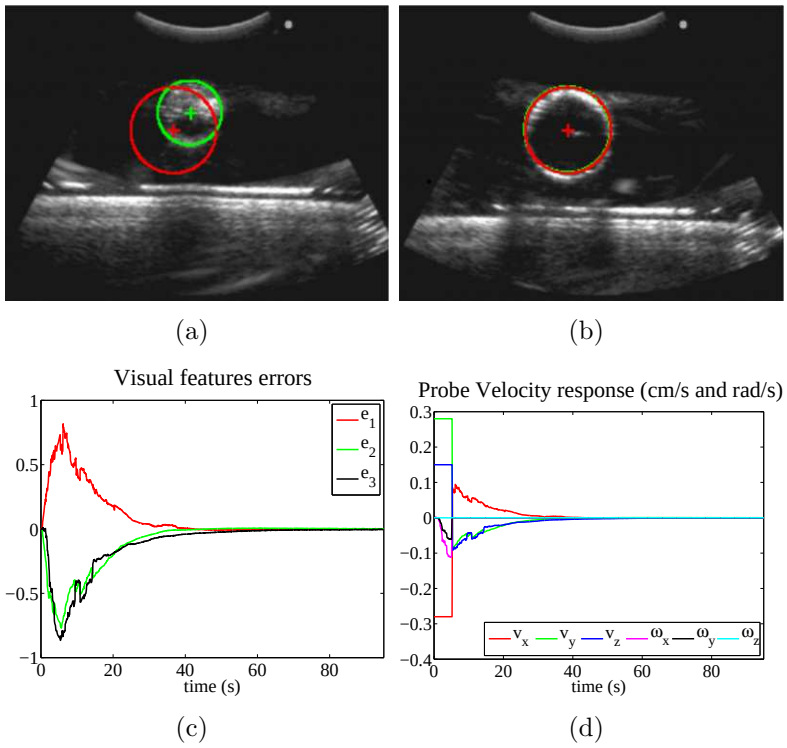


Figure 5.22: Experiment using the **model-free** visual servoing that uses the **curved** line-based estimation, where the probe interacts with a spherical object -(a) Initial image captured right before launching the servoing, where the actual section is contoured with green. The contour of the desired image section is displayed with red and superimposed on the initial image - (b) Target image automatically reached after visual servoing - (c) Visual features errors in (cm ,cm, cm) (d) Probe velocity applied on the probe.

been defined in terms of image moments by the relationship (3.48). The robotic task consists in first learning a desired image section, then moving away the probe transducer from that target by applying open-loop motion with constant velocity. During that motion, the SLS algorithm presented in Section 4.3 is employed for only the first $N_{LS} = 60$ iterations, in order to obtain initial estimate Θ_0. Right after, when the probe reaches a distant location, the servoing is launched where the recursive least squares estimation algorithm presented in Section 4.1.2 is employed throughout the trial. The control gain is set to $\lambda = 0.1$. As for the estimator parameters, they are tuned to $\beta = 0.8$, f_0 =1e6, $\beta_0 = \frac{1}{(20 \times f_0)}$, and ϵ_0 = 1e-10. Note that in this experiment, we have employed the straight line-based estimation method. Corresponding experimental results are shown on Fig. 5.22. The visual features errors converge to zero, roughly exponentially, as can be seen on Fig. 5.22(c), and the reached image section corresponds to the desired one as can be seen on Fig. 5.22(b). The robot behavior

Figure 5.23: The probe transducer interacting with an ultrasound phantom.

is correct, where smooth motions have been applied as can be seen on Fig. 5.22(d). These results thus give a first experimental validation of the model-free visual servoing method based on line estimation.

5.5.2 Exprimental results with an ultrasound phantom

The model-free visual servoing method based on straight line estimation is tested on an ultrasound phantom (see Fig. 5.23). In this case the ultrasound transducer is in contact with the phantom and applies a 2 N force on it. For that, the velocity v_z of the probe is constrained by force control. We noticed however that feature l_1 was coupled with the area, likely due to the relatively-symmetric shape in the image of the phantom object. We thus removed that feature from the visual features vector, which is now $\mathbf{s} = (x_g, y_g, \alpha, \sqrt{a})$. The estimator parameters are tuned to $\beta = 0.95$, f_0 =1e8, and $\beta_0 = \frac{1}{20 \times f_0}$. The robotic task consists to automatically reach two successive target images; the second target is sent to the controller after the first one would have been reached. Corresponding results are shown on Fig. 5.24. The visual features errors converge to zero roughly exponentially [see Fig. 5.24(e)]. Both the two target images have been reached as can be seen respectively on Fig. 5.24(b) and 5.24(d). The motions of the probe are also correct as can be noticed from the applied probe velocity, shown on Fig. 5.24(f).

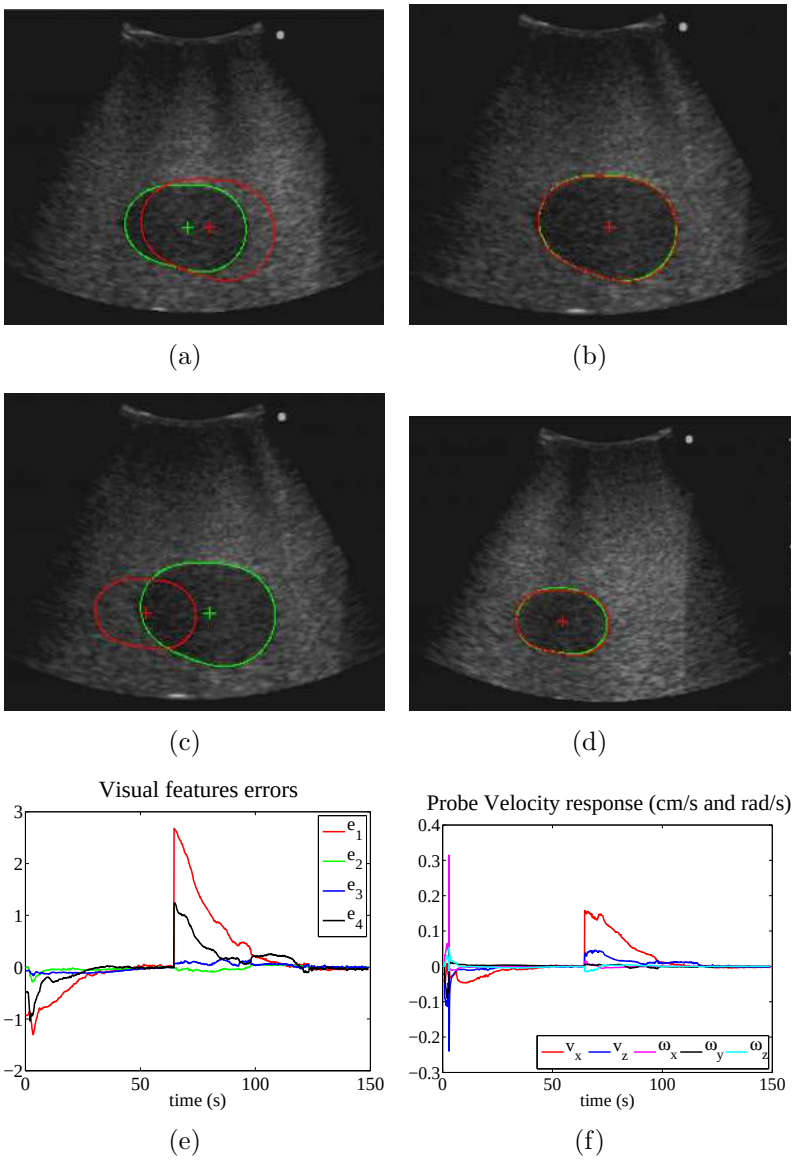


Figure 5.24: Experimental results with an ultrasound phantom using the **model-free** visual servoing method based on **straight** line estimation (the current contour is in green and the desired one in red): (a) Initial and first target image - (b) First target reached after visual servoing - (c) A second target image is sent to the robot - (d) The second target image is reached after visual servoing - (e) Visual error time response (cm, cm, rad, cm)- (f) Control velocity applied to the probe.

5.5.3 *Ex-vivo* experimental results with a lamb kidney

We test the model-free visual servoing method based on straight line estimation on a motionless lamb kidney immersed in the water-filled tank. Similarly, the robotic task consists to automatically reach two successive target images. The feedback visual features vector is $\mathbf{s} = (x_g, y_g, \alpha, \sqrt{a}, l_1)$. We have not used six visual features because of the symmetry of the section in the image. The estimator parameters are tuned to $\beta = 0.8$, f_0 =1e5, and $f_0 = \frac{1}{20 \times f_0}$. Corresponding results are shown on Fig. 5.25. The visual features errors converge to zero [see Fig 5.25(e)]. Both to the two reached images correspond to the desired ones, as can be seen respectively on Fig. 5.25(b) and 5.25(d). The robot behavior is correct as can be noticed from the relatively smooth applied probe velocity shown on Fig. 5.25(f). These results therefore experimentally validate the model-free servoing method on real soft tissue.

Note that in the experiments presented above, less than six visual features have been used. As such, the pose reached by the probe would unlikely correspond to that where the desired image had been captured, as already highlighted in Section 5.2.1. In the following, we present experimental results obtained with six visual features at least.

5.5.4 Experimental results with a motionless soft tissue

We test the servoing method on a grossly asymmetric gelatin-made soft tissue object. Such asymmetry yields the six visual features independent, which allows to control the 6 DOFs of the robotic system and, thus, to automatically position the probe with respect to the object. In other words, the probe should automatically recover the pose with respect to the object where the desired image had been captured. The feedback visual features are $\mathbf{s} = (x_g,\ y_g,\ \alpha,\ \sqrt{a},\ \phi_1,\ \phi_2)$. Note that we used the curved line-based estimation method. As before, the robotic task consists in first acquiring a desired image, then moving away the probe from the corresponding location where this image had been captured. The motion is performed during 70 iterations in open-loop with constant velocity. During this moving away, the SLS algorithm is applied for the first $N_{LS} = 60$ iterations. This allows to obtain initial estimate Θ_0. Right after the recursive algorithm takes place, instead of the SLS one, and then is solely applied throughout the trial. The control gain is set to $\lambda = 0.05$, and the estimator parameters are tuned to $\beta = 0.8$, $f0$ =1e6, $\beta_0 = \frac{1}{20 \times f_0}$ and ϵ_0 =1e-10. The corresponding experimental results are shown on Fig. 5.26. The six visual errors converge to zero, roughly exponentially, as can be seen on Fig. 5.26(c), and the reached image section corresponds to the desired one as can bee seen on Fig. 5.26(b). Moreover, the probe automatically came back quite near the pose where the desired image had been captured

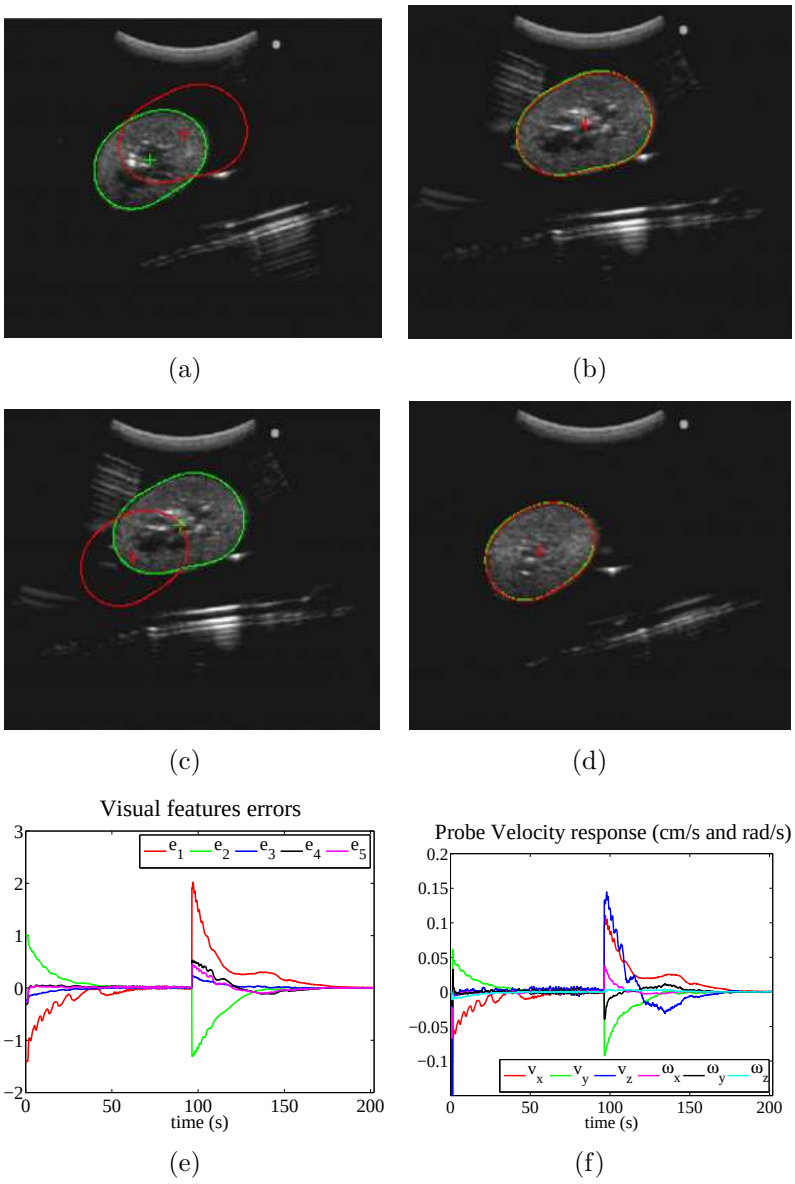


Figure 5.25: Experimental results with a lamb kidney using the **model-free** visual servoing method based on **straight** line estimation (the current contour is in green and the desired one in red): (a) Initial and first target image - (b) First target reached after visual servoing - (c) A second target image is sent to the robot - (d) The second target image is reached after visual servoing - (e) Visual error time response (cm, cm, rad, cm, cm)- (f) Control velocity applied to the probe.

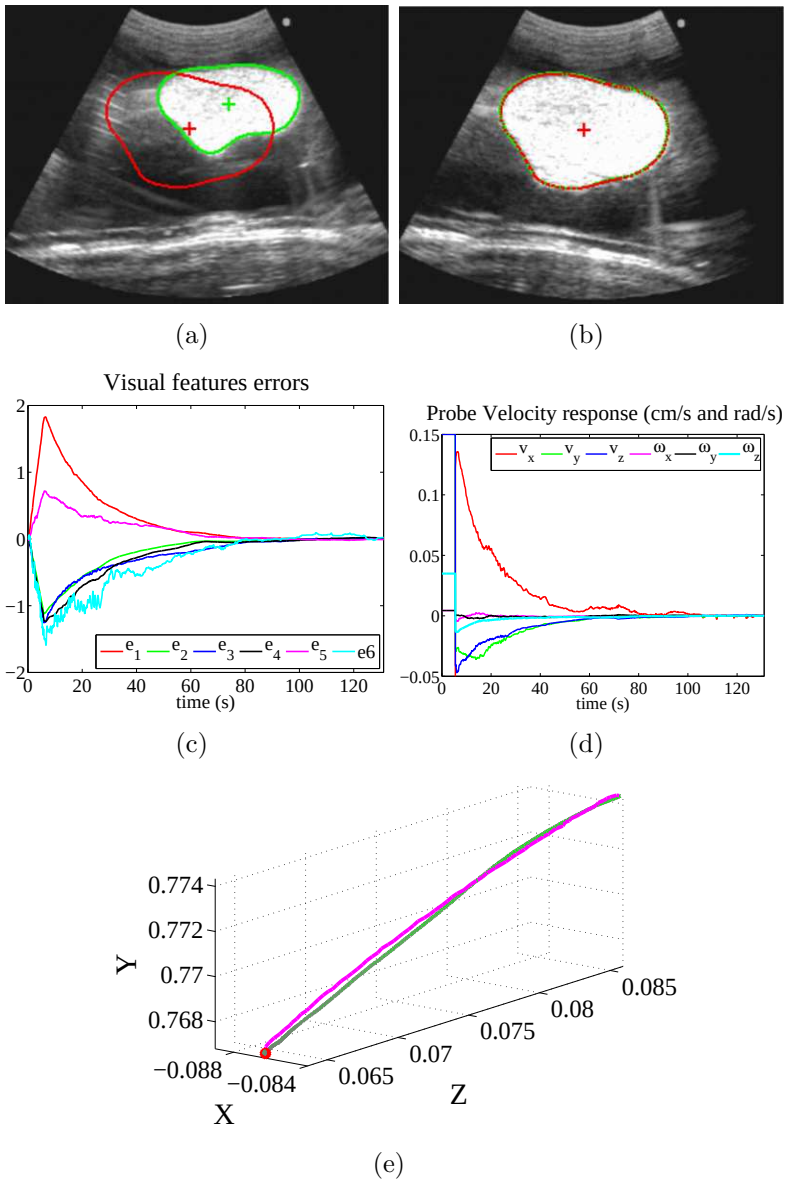


Figure 5.26: Experimental results obtained with the **model-free** visual servoing method based on **curved** line estimation, where the probe is interacting with a soft tissue object that possesses asymmetric regions - (a) Initial image captured right before launching the servoing, where the actual section is contoured with green. The contour of the desired image section is displayed with red and superimposed on the initial image (b) Desired image reached after visual servoing - (c) Visual features errors in (cm, cm, rad, cm, unit, 10×unit) - (d) Probe Velocity - (e) Trajectory performed by the probe, where that obtained during the open-loop motion is plotted in magenta and that obtained during the servoing is plotted with green. The position where the desired image had been captured in indicated with the read stared point.

[see Fig. 5.26(e)]. The corresponding obtained positioning errors are (0.4, 0.6, -0.2) mm and (0.05, -0.7, -0.8) degree respectively for the position and the θu rotation[2]. The robot behavior is correct, where smooth motions have been performed as can be seen on Fig. 5.26(d), despite the noisy images. Thus, these results experimentally validate the servoing method for both reaching a desired ultrasound image and recovering the location where that image had been captured.

5.5.5 Tracking two targets

In case the observed object is not asymmetric, it is still possible to stabilize the probe with respect to it. We propose two solutions for that. They are described in Chapter 6. Let us consider here the second solution, that consists to rather consider a couple of targets instead of only one, as was so far considered in this work. As observed object, we take back the ultrasound phantom used in the experiment reported in Section 5.5.2. We have seen that when considering only one target section image, it is unlikely that the probe retrieves the pose where that target image is captured and thus it would not be possible to stabilize the probe with respect to the phantom (object). However, in this experiment we consider two target sections, as can be seen on Fig. 5.27(i). From each observed section, five visual features are computed. As a result, the system is fed back with 10 visual information. Note however that velocity component v_z is servoed by force control in order that the probe exerts a couple of newton force along its Y axis (see Fig. 3.5 and 3.7 for the probe axes configuration). The phantom is put on a manually-driven tray. The task consists to track the two target section images when the phantom is arbitrarily and manually moved. Note that in contrast to the above presented results, we employed in this experiment a new acquired 6 DOFs anthropomorphic robot arm. Corresponding experimental results[3] are shown on Fig. 5.27. We can see that the robotized probe automatically tracks the moving ultrasound phantom, and stabilizes with respect to it. The observed image sections superimpose on the target ones [see Fig. 5.27(i)]. Note that since we used a basic control law, the system response is relatively slow and presents delays. Employing a tracking-dedicated control law, the system reactivity would increase. We have not estimated and thus not predicted the phantom motions to then eventually forward the information in the robot motions control. In this work, only the observed image along with the robot odometry is used to compute the commands to control the robot. However, estimating the phantom movements, as using for example a Kalman filter, the results are expected to be better. Note also that we have been

[2] $\theta\mathbf{u}$ representation is defined by a unitary vector $\mathbf{u}$, representing the rotation axis, and rotation angle θ around this axis.

[3] The corresponding video can be found at http://www.irisa.fr/lagadic/team/old/Rafik.Mebarki-eng.html.

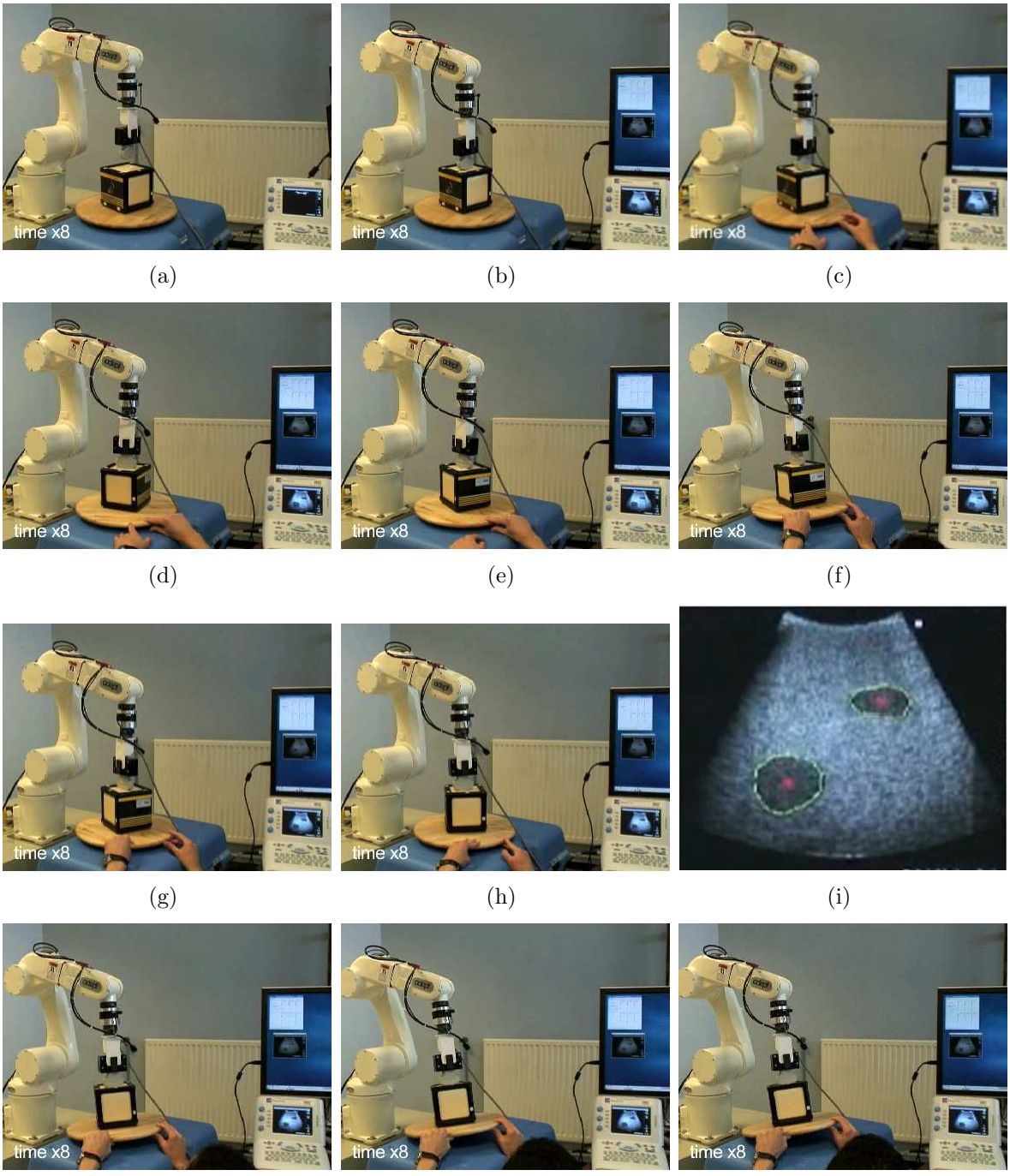


Figure 5.27: Tracking two target sections: sequences taken during the tracking - (i) Observed 2D ultrasound image. The two observed cross-sections are contoured with green, while the contours of their respective targets are in red. The contour of each observed cross-section superimposes on its corresponding target.

constrained by the computational time, since the image processing takes a large amount of resources. To cope with that, we have used only 50 image points to characterize the contour of each observed section. As so, the contour is not enough sub-sampled, which could compromise the accuracy of both the normal vector estimation and the computed command velocity. We noticed shakiness of the snake during the phantom displacements.

5.6 Conclusion

We presented new visual servoing methods to automatically position a robotized 2D ultrasound probe in order to reach and maintain desired cross-section images. Firstly, we presented simulation results that have shown the validity of the model-based visual servoing method, where the object 3-D model is required. The latter constraint, as emphasized, considerably limits visual servoing based on ultrasound images. Thanks, however, to the normal vector estimation methods, we developed model-free visual servoing methods that overcome that constraint. Indeed, these methods do not require any prior knowledge of the shape of the observed object, its 3D parameters, nor its location in the 3D space. They instead on-line estimate normal vector ${}^{s}\nabla\mathbf{F}$ to then employ it in the control law. We presented three different model-free servoing methods, according to the geometrical primitive they use for the estimation. We distinguished servoing methods that use respectively straight line-, curved line-, and quadric surface-based estimation method. In this chapter, we reported simulation results obtained with method based on curve estimation, while those obtained with the straight line- and the quadric surface-based methods are presented in Appendix C. The results showed the validity of the two methods based on straight and curved line primitives. They suggested that these two methods outperform the quadric surface-based method. The latter one, indeed, showed to be considerably less robust to image noise, and has failed for important probe displacements. For small displacements, the probe velocity was nevertheless too shaky. Such performances were in fact expected from the simulations presented in Chapter 4. In those simulations, we noticed that the curved line-based estimation showed to be more effective. Then, we reported experimental results where we have tested the model-free visual servoing based on line estimation. They have been obtained with both a spherical object, an ultrasound phantom, a lamb kidney, and a relatively complex soft tissue object. The probe automatically reached the desired cross-section images. Moreover, it automatically comes back quite near to the pose where the desired image is captured on the gelatin-made soft tissue object. Finally, considering two target sections simultaneously, on the the ultrasound phantom, the latter has been automatically tracked by the robotized probe. All those results thus experimentally validated the model-free visual servoing we propose in this dissertation. Consequently, and more precisely, they validate both the the-

oretical foundations developed in Section 3, the normal vector on-line estimation method presented in Chapter 4, and the selection of feedback visual features $\mathbf{s}$ in the present chapter.

Chapter 6

Conclusions

The research work presented in this dissertation lies mainly within the field of image-based visual servoing. It investigated the exploitation of 2D ultrasound images for automatic guidance and thus positioning of robotized 2D ultrasound probes with respect to observed soft tissues. The scenario consists of a 2D ultrasound probe carried and thus actuated by the end-effector of a general medical robot arm. The latter is servoed in velocity thanks to the visual servoing schemes we developed and presented in this document. The control law of the visual servoing scheme indeed computes the velocity that the robot has to achieve in order to reach the desired ultrasound image. As highlighted, the control law of an image-based visual servoing scheme requires the interaction matrix related to the feedback visual features. The interaction matrix, in fact, relates the differential changes of the visual features to differential displacements (configuration changes) of the robot. However, the analytical form of such matrix was not available for 2D ultrasound imaging systems, due to the fact that the latter interact with their environment with a manner that was, so far (before our works), challenging to model. These systems completely differ from optical systems, for instance, whose use in robotic automatic guidance is the subject of extensive investigations in the field of visual servoing. In particular, for optical imaging systems, as perspective cameras for example, the interaction matrix related to differential changes of the image points coordinates is already available. From that matrix, that related to different visual features can be derived. It was not the case for 2D ultrasound imaging systems. Another main challenge when dealing with these systems consists in the fact that the image feature variations strongly depend on the 3-D shape of the object with which the probe is interacting. The challenge corresponds mainly to a mathematical modeling problem. A couple of investigation works, that have been presented in Chapter 2, provided the interaction matrix for only a simple 3-D geometrical primitive, namely 3-D straight line. The work presented in this dissertation addressed all those cited challenges. We developed, indeed, general methods that endow the the robotic system with the capability of dealing with objects of whatever shapes, in order to automatically position the probe with respect to

them. Doing so required mainly to develop new theoretical foundations in term of modeling techniques. Our contributions can be summarized as follows:

(a) We have proposed to use visual features based on image moments as feedback for visual servoing schemes, to automatically control the robot from the observed 2D ultrasound images. This direction seems judicious since the image moments show to be relevant in case of 2D ultrasound images. Indeed, computing the image moments needs only a global segmentation of the section in the image, and thus does not require matching of points in the image except for the section in the image. This is of great interest when dealing with 2D ultrasound since, as described in the present document, the points of the image do not match to those of the precedent image. This is explained by the fact that the observed points are not the same, in contrast to optical systems for example. A preliminary exploration work [54] validated the relevance of our choice for image moments. However the interaction matrix related to image moments was approximated. Moreover, the considered observed object is assumed grossly ellipsoidal, and its 3-D parameters are assumed roughly known. This has been addressed and presented in this dissertation, where the exact form of interaction matrix $\mathbf{L}_{m_{ij}}$, related to image moment m_{ij}, has been modeled;

(b) To obtain the interaction matrix, its exact form, more precisely, we first highlighted that a key solution would be to consider only the image velocity of the points lying on the contour of the section in the image (the contour and the section in the image have been respectively denoted by $\mathcal{C}$ and $\mathcal{S}$). The image moments time variation, thanks to the Green's theorem, can be formulated as function of the velocity of those contour points. The objective then consisted to obtain such image velocity;

(c) The image contour points indeed correspond to points sliding on the surface of the observed object. We have shown that such points can satisfy two constraints, that consist in the relationships (3.20) and (3.22). Each constraint corresponds to a scalar mathematical relationship. Using these two constraints, we then have been able to model an exact form of the image velocity of the contour points. The formulae is given by the relationship (3.27) according to (3.28). Such image velocity, denoted $(\dot{x}, \dot{y})$, is expressed as function of velocity $\mathbf{v}$ of the robot end-effector (or of frame $\{R_s\}$ attached to the robotized probe);

(d) Using the image velocity relationship, we finally derived the exact form of interaction matrix $\mathbf{L}_{m_{ij}}$, as given by the relationship (3.34) according to (3.35). It was noticed that the interaction matrix requires the knowledge of the image coordinates of the points lying on the image contour, and also vector ${}^{s}\nabla\mathbf{F}$ normal to the surface of the observed object at each of the considered contour points. The obtained results

have been verified on simple 3-D geometrical primitives, like spheres and cylinders, for certain configurations. We then have designed a visual servoing scheme where the feedback visual features are combinations of image moments. Six relevant independent visual features have been proposed to control the 6 DOFs of the robotic system. A classical control law has been employed in the servoing scheme. The control law requires the interaction matrix, or its estimate, at each iteration. If the matrix is exact, the visual features errors are expected to converge to zero exponentially. This latter characteristic has been exploited to verify again the exactitude of the interaction matrix. To do so, we performed simulations where the scenario consisted of a virtual 2D ultrasound probe that interacts with an ellipsoidal object. This object was assumed exactly known. Its half length values an its pose are used to compute the actual values of the image coordinates and the normal vector, that are used to compute the control law. We have noticed that, indeed, as expected, the feedback visual features errors converge to zero exponentially [e. g. Fig. 5.4(f)]. This validates, once again, the correctness of the developed interaction matrix;

(e) Another problematic, as pointed out above, consisted in the fact that the variations of the image information depend strongly on the 3-D shape of the observed object. This can be noticed from the involvement of ${}^{s}\nabla\mathbf{F}$ in elements K_x and K_y, given by (3.28), that are required in the expression of the image point velocity and, hence, of interaction matrix $\mathbf{L}_{m_{ij}}$. Computing this normal vector would have suggested the use of a 3-D pre-operative model of the observed object. Such resolution however would have greatly hindered the visual servoing, where the 3-D model has to be registered to the object at each iteration; besides that the accuracy of the extracted normal vector would be directly and heavily based on that of the registered 3-D model. Our work overcame such limitations, where we proposed model-free visual servoing methods that do not require any prior information about the shape, 3-D parameters, nor 3-D location (position and orientation) of the observed object. To do so, we have developed estimation methods to on-line estimate the normal vector. We proposed three estimation techniques:

- straight line-based estimation;
- curved line-based estimation;
- and quadric surface-based estimation.

Even though that opting for quadric surface primitives for the estimation seems the more natural direction that one could take, we have noticed from different performed simulations that the quadric surface-based estimation considerably underperformed the first two methods, that rather well performed in different conditions. In fact, we expected such difference of outcomes. This can be explained by the fact that the

two first methods do not estimate in whole the normal vector but only a part of it. Indeed, these two techniques decompose a normal vector into two tangent vectors. The former tangent vector can be extracted from the image, while only the latter needs to be estimated. Doing so, we spare the obtained normal vector value the effect of a part that would add errors if the normal vector would have to be estimated in whole; thus only the errors on the estimation of the second tangent vector have an impact on the normal vector. Moreover, fitting a line to a set of successive points seems less constrained than fitting a surface to a cloud of points. Experiments have been conducted, where we tested the model-free visual servoing methods that use the line-based estimation. The corresponding results have experimentally validated the methods.

Thus, the previous cited challenges that were faced and that hindered robotics automatic guidance from 2D ultrasound images are now addressed thanks to the theoretical foundations and the methods we have presented in this document. We have provided through this thesis basics on which new investigation and thus developments can now be undertaken. Nevertheless, some of the proposed methods could be improved. It was proposed in this dissertation to employ a stabilized recursive least squares algorithm to perform the estimation of ${}^{s}\nabla\mathbf{F}$. It would be interesting instead to test a Kalman filter (KF), or an Extended Kalman filter (EKF), in order to verify which algorithm gives the best outcome in terms of estimation accuracy, speed, and robustness. Let us point out that in [86] it was concluded that an EKF estimator outperformed a least squares one in terms of accuracy and speed in predicting periodic motions; mimicking mitral valve motions for heart surgery. The 3-D ultrasound imaging was employed in that work. Another point is that we have performed simulations and experiments mainly on motionless observed objects. Dealing with moving objects could be considered with the developed methods as is. This in fact could be technically addressed by making the robotic system performing at high sampling streaming rates. Indeed, acquiring the images and then ordering the command velocity at a sufficiently high streaming rate, such that the motions of the object between two samples could be neglected, the estimation algorithm would be insensitive to the object motions. The modeled interaction matrix is also concerned if moving objects are considered, but this again could be similarly addressed. However, if the motions of the observed object become faster with regards to the streaming rate, such that its displacements between two acquired images can not be neglected, the proposed methods might fail. That is the reason why this should be further investigated. Nevertheless, we think that the concepts we have proposed and used in the modeling of the interaction matrix and in the estimation of the normal vector can be taken back and adapted for the case of moving objects.

We have proposed six visual features to control the 6 DOFs of the robotic system and thus to automatically position the probe at a desired cross-section of the observed object. The 2D ultrasound probe can automatically come back to the pose (position and orientation) where a desired image is captured. This can be achieved provided that object is asymmetric. If it is not the case, a desired image can corresponds to an infinity of cross-sections (slices), and consequently the probe might fail to automatically retrieve the corresponding pose. Nevertheless, such issue might be addressed by employing not only one 2D ultrasound probe but, instead, a couple of probes; as example to illustrate, two orthogonal probes can be employed. Both probes should of course be actuated by the same robotic system. Each acquired image from each of the probes would provide different section and also would target a different cross-section. The task would then consist to reach both desired cross-sections. In fact, the whole information provided by all the probes should be enough to extract at least six independent visual information. This can be afforded by means of, for example, a *selection matrix* using the *task function approach* [68]. When all the probes would reach their respective target cross-sections, the considered probe would thus clearly be positioned at the desired cross-section that we are interested in; the other probes with the imaged cross-sections are only considered to add visual information, no more. A second solution, which is a dual solution of the above-mentioned one, would be to rather consider different target sections in the same image, instead of only one section. Indeed, we have shown in Section 5.5.5 that by considering two target sections the robotized probe has been able to stabilize with respect to a moving ultrasound volume, the 3-D phantom in this case.

Another issue is that if the shape (closed surface) of the object possesses local minima, the visual servoing method might be trapped by these latter, in case the probe trajectory would encounter them. A resolution that could be proposed consists in using path of images that would successively guide the probe up to the desired image of the target cross-section that we are interested in. Such resolution could be also used to guide the probe from relatively far locations. As for the selection of the six visual features, we suggested to use ϕ_1 [given by the relationship (5.8)] as fifth feature, if the image noise is not considerably high to a certain extent that would compromise the system stability. If it is not the case (i. e., the noise is high) we recommended to instead use l_1 [relationship (5.11)] as fifth feature. This latter indeed showed to me more robust to image noise. The advantage of ϕ_1 is that it has rather showed to yield the probe motions more decoupled. The choice between ϕ_1 and l_1 is thus subject to a compromise between motion decoupling and robustness. However, it would be nice to find, or to investigate for, a feature that could well satisfy both the two traits: decoupling and robustness. The same applies for the sixth feature, where we proposed ϕ_2.

In a practical scenario, the probe is in contact with the patient skin. Therefore, the interaction forces need to be controlled. In some of the experiments we have conducted, where

the probe was in contact with the soft surface of the ultrasound phantom, we have constrained probe velocity component v_y with a proportional force controller, in such a way the probe could exert a couple of newtons force along its Y axis. However, such an approach is rudimentary, since one DOF of the system is no longer used and thus lost by the visual servo controller to compensate for all in-plane and out-of-plane motions. As such, some motions could no longer be compensated to keep the target in the image. Moreover, in case the probe is oriented with respect to the contact surface, the force along Y axis would not correspond to the amount of exerted forces on that surface. Therefore, controlling v_z would no longer allow to control all the contact forces. That is the reason why we propose to investigate for a more sophisticated approach. However, we provide a direction for this. A system where visual servoing and force control share the command of the robot motions should be considered. The *task function approach* again could be useful, where the control law can be computed based on the priority given to the functions to achieve: vision or force. To do so, we propose to consider at least two modes. The first mode would correspond to the case where the exerted forces are below a pre-fixed threshold and thus are considered no dangerous for the patient body, while the second mode would consist in the case where these forces are above the threshold. In the first mode, the priority should be given to the visual servoing rather than to the force control. As for the second mode, the priority should be inverted, that is, giving more importance to control the forces than keeping the target in the image. The system of course should switch to either modes depending on the amount of exerted forces with respect to the threshold.

Dealing with deformable objects (mimicking soft tissue deformations) should also be investigated in the future works. Doing so seems to be a strong challenge. A preliminary key solution would be to order the robotic system with variable desired image, and not with static one as it is the case for most of visual servoing schemes. In case the object deforms periodically, the section in the image would also periodically vary. Consequently, the variation of the cross-section and thus of the section in the image could be predicted. The objective would be then to send to the visual servoing schemes the predicted images of the desired cross-section. If the images are well predicted and synchronized with respect to the object deformations, we expect that the 2D US probe could be automatically positioned at the desired cross-section of the observed object. It would be likely assumed that the object deforms homogeneously, such that shear deformations would not be considered.

Finally, the methods developed through this thesis have brought basics on which, we expect, new techniques could now be developed. These theoretical foundations could also be combined with other different techniques dedicated for robotics control. Although the methods we developed focused on robotic guidance using 2D ultrasound images, they might be extended to MRI and X-ray. These two latter modalities provide indeed, like ultrasound,

full information in their observation plane, and thus both of these three modalities interact with their environments by the same manner. Therefore the modeling methods developed in this thesis can apply to the two latter modalities. Ultimately, the imaging modalities discussed in this thesis might be complementary and thus exploited in a synergistic manner.

Appendix A

Some fundamentals in coordinate transformations

A.1 Scalar product

Let vectors a and b be of same dimension n defined respectively by $a = (a_1, a_2, ..., a_n)$ and $b = (b_1, b_2, ..., b_n)$. The scalar product $a \cdot b$ of a and b is defined by:

$$a \cdot b = b \cdot a = a^\top b = b^\top a = \sum_{k=1}^{k=n} a_k \, b_k \tag{A.1}$$

If a and b are orthogonal, we have:

$$a \cdot b = 0 \tag{A.2}$$

A.2 Skew-symmetric matrix

The skew-symmetric matrix $[a]_\times$ associated to vector $a = (a_x, a_y, a_z)$ is given by:

$$[a]_\times = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \tag{A.3}$$

The following property can be deduced:

$$[a]_\times^\top = -[a]_\times \tag{A.4}$$

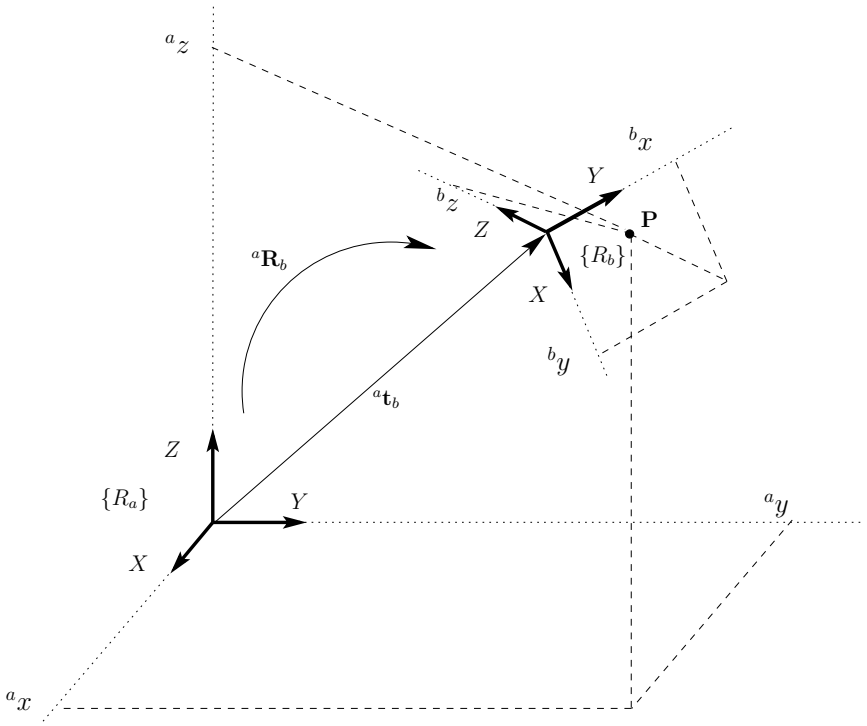


Figure A.1: Points projection

A.3 Vector cross-product

Let a and b be vectors. Their cross-product can be written as:

$$a \times b = [a]_\times \, b \tag{A.5}$$

with the following property:

$$a \times b = -\, b \times a \tag{A.6}$$

The resulted vector is orthogonal to the plane formed by a and b. This can be written by the following scalar product:

$$(a \times b) \cdot a = (a \times b) \cdot b = 0 \tag{A.7}$$

If a and b are parallel, we therefore have:

$$a \times b = 0 \tag{A.8}$$

A.4 Points Projection

Let $\mathbf{P}$ a point of the 3-D space, where $\{R_a\}$ and $\{R_b\}$ are 3-D cartesian frames (see Fig A.1). The coordinates of $\mathbf{P}$ in the frame $\{R_a\}$ are given by the vector position ${}^a\mathbf{P} = ({}^a x,\ {}^a y,\ {}^a z)$.

Let also, ${}^a\mathbf{R}_b$ be the rotation matrix defining the orientation of the frame $\{R_b\}$ with respect to the frame $\{R_a\}$, and ${}^a\mathbf{t}_b$ be the vector position defining the origin of $\{R_b\}$ in the frame $\{R_a\}$. Therefore the 3-D coordinates ${}^b\mathbf{P} = ({}^b x,\ {}^b y,\ {}^b z)$ of $\mathbf{P}$ in the frame $\{R_b\}$ can be obtained as follows:

$$ {}^b\mathbf{P} = {}^a\mathbf{R}_b^\top \ ({}^a\mathbf{P} - {}^a\mathbf{t}_b) \tag{A.9} $$

A.5 Rotation matrix properties

A rotation matrix is an orthogonal matrix. Considering a rotation matrix $\mathbf{R}$, the orthogonality is expressed as follows:

$$ \mathbf{R}^\top = \mathbf{R}^{-1} \tag{A.10} $$

that can be also written by:

$$ \mathbf{R}^\top \mathbf{R} = \mathbf{R}\,\mathbf{R}^\top = \mathbf{I}_3 \tag{A.11} $$

A rotation matrix possesses the following property:

$$ {}^a\mathbf{R}_b = {}^b\mathbf{R}_a^{-1} = {}^b\mathbf{R}_a^\top \tag{A.12} $$

Appendix B

Calculus

B.1 Integral of trigonometric functions

We provide in this section calculus results of some trigonometric functions integration that has been used in Section 3.7.1.
Consider a real scalar θ. We obviously have:

$$\int_0^{2\pi} \sin\theta \, d\theta = 0 \tag{B.1}$$

$$\int_0^{2\pi} \cos\theta \, d\theta = 0 \tag{B.2}$$

Consider now the complex entity $e^{i\theta}$, of the scalar θ, and its conjugate $e^{-i\theta}$, where i is the imaginary unit such that $i^2 = -1$. Theses two complex entities can be written:

$$\begin{cases} e^{i\theta} &= \cos\theta + i\sin\theta \\ e^{-i\theta} &= \cos\theta - i\sin\theta \end{cases} \tag{B.3}$$

From which it can be deduced:

$$\begin{cases} \cos\theta &= \frac{1}{2}\left(e^{i\theta} + e^{-i\theta}\right) \\ \sin\theta &= \frac{1}{2i}\left(e^{i\theta} - e^{-i\theta}\right) \end{cases} \tag{B.4}$$

The above relationship is used to calculate first the integral of the function $\sin^2\theta$. The same approach can then be followed to calculate the integral of the remaining functions, presented in the following, that we need in Section B.2. We have from (B.4):

$$\sin^2\theta = -\frac{1}{4}\left(e^{2i\theta} + e^{-2i\theta} - 2\right) \tag{B.5}$$

Integrating the above relationship gives:

$$\int_0^{2\pi} \sin^2\theta\, d\theta = -\frac{1}{4}\left[\frac{1}{2i}\left(e^{2i\theta} - e^{-2i\theta}\right) - 2\theta\right]_0^{2\pi} \tag{B.6}$$

which yields:

$$\int_0^{2\pi} \sin^2\theta\, d\theta = \pi \tag{B.7}$$

Similarly following the above approach, we have:

$$\int_0^{2\pi} \cos^2\theta\, \sin\theta\, d\theta = 0 \tag{B.8}$$

$$\int_0^{2\pi} \cos\theta\, \sin^2\theta\, d\theta = 0 \tag{B.9}$$

$$\int_0^{2\pi} \cos^2\theta\, \sin^2\theta\, d\theta = \frac{\pi}{4} \tag{B.10}$$

$$\int_0^{2\pi} \sin^3\theta\, d\theta = 0 \tag{B.11}$$

$$\int_0^{2\pi} \cos\theta\, \sin^3\theta\, d\theta = 0 \tag{B.12}$$

$$\int_0^{2\pi} \sin^4\theta\, d\theta = \frac{3\pi}{4} \tag{B.13}$$

B.2 Calculus of n_{ij}, spherical case

In this section, we express in a simple and appropriate form the elements n_{20}, n_{11}, and n_{02} given by (3.51). From the obtained equation (3.44) that states the relationship of points lying on the image contour in case of sphere shaped object, we can set the following change of coordinates:

$$\begin{cases} x &= t_x + r\,C\theta \\ y &= t_y + r\,S\theta \end{cases} \quad 0 \leqslant \theta < 2\pi \tag{B.14}$$

with $C\theta = \cos(\theta)$ and $S\theta = \sin(\theta)$, where θ represents the angle in the image.

Since $t_x = x_g$ and $t_y = y_g$, the above relationship system becomes:

$$\begin{cases} x &= x_g + r\,C\theta \\ y &= y_g + r\,S\theta \end{cases} \tag{B.15}$$

The image moment m_{ij} can be formulated as a line integral around the image contour $\mathcal{C}$, as given by (3.32). We use this relationship to calculate the second order image moments m_{20}, m_{11} and m_{02}.

Applying (3.32), the moment m_{20} is thus expressed as:

$$m_{20} = -\oint_{\mathcal{C}} x^2\,y\,dx \tag{B.16}$$

$$= -\int_0^{2\,\pi} x^2\,y\,\frac{dx}{d\theta}\,d\theta \tag{B.17}$$

substituting x and y with their corresponding expressions given by (B.15), we have:

$$m_{20} = r^3\,y_g \int_0^{2\pi} C\theta^2\,S\theta + r^2\,x_g^2 \int_0^{2\pi} S\theta^2 + 2r^3\,x_g \int_0^{2\pi} C\theta S\theta^2 + r^4 C\theta^2 S\theta^2 \tag{B.18}$$

then using the result of some trigonometric integrations provided in Appendix B.1, then recalling that the area a of the image section is $a = \pi r^2$, we obtain

$$m_{20} = a(x_g^2 + r^2/4) \tag{B.19}$$

replacing, finally, this in (3.51), $n_{20} = m_{20}/a$, yields:

$$n_{20} = (x_g^2 + r^2/4) \tag{B.20}$$

Similarly for the moment m_{11}, it can be expressed as follows:

$$m_{11} = -\frac{1}{2}\oint_{\mathcal{C}} x\, y^2\, dx \tag{B.21}$$

$$= \int_0^{2\pi} x\, y^2\, \frac{dx}{d\theta} \tag{B.22}$$

substituting x and y with their respective expressions given by (B.15) yields:

$$\begin{aligned} m_{11} \;=\; & \tfrac{1}{2}\, r^4 \int_0^{2\pi} C\theta S\theta^3 d\theta \;+\; \tfrac{1}{2}\, r^3\, x_g \int_0^{2\pi} S\theta^3 d\theta \;+ \\ & r^3\, y_g \int_0^{2\pi} C\theta S^2\theta\, d\theta \;+\; r^2\, x_g\, y_g \int_0^{2\pi} S\theta^2 d\theta \;+ \\ & \tfrac{1}{2}\, r^2\, y_g^2 \int_0^{2\pi} C\theta\, S\theta\, d\theta \;+\; \tfrac{1}{2}\, r\, x_g\, y_g^2 \int_0^{2\pi} S\theta\, d\theta \end{aligned} \tag{B.23}$$

after using calculus results of Appendix B.1, we obtain m_{11} as follows:

$$m_{11} = a\, x_g\, y_g \tag{B.24}$$

which yields since $n_{11} = m_{11}/a$:

$$n_{11} = x_g\, y_g \tag{B.25}$$

We follow the same steps for m_{02}. It can be expressed by:

$$m_{02} = -\frac{1}{3}\oint_{\mathcal{C}} y^3\, dx \tag{B.26}$$

$$= -\frac{1}{3}\int_0^{2\pi} y^3\, \frac{dx}{d\theta}\, d\theta \tag{B.27}$$

substituting y with its expression given by (B.15), we have:

$$m_{02} = \frac{1}{3}\, r^4 \int_0^{2\pi} S\theta^4 d\theta + r^3\, y_g \int_0^{2\pi} S\theta^3 d\theta + r^2\, y_g^2 \int_0^{2\pi} S\theta^2 d\theta \tag{B.28}$$

we find:

$$m_{02} = a(y_g^2 + r^2/4) \tag{B.29}$$

and thus:

$$n_{02} = (y_g^2 + r^2/4) \tag{B.30}$$

Appendix C

Supplementary simulation results of model-free visual servoing

C.1 Model-free servoing on the ellipsoid

C.1.1 Using the straight line-based method

We use the straight line-based technique, described in Section 4.1.1 to on-line estimate the normal vector to the object surface, namely the surface of the ellipsoidal object in this case. The estimation is performed at each of the 400 contour points. When the servoing is applied, the new acquired image with its extracted contour points update the estimation. The new computed value of the normal vector then is used to compute the control law. An open-loop motion with constant velocity is applied to the probe before the servoing is launched. During that motion, the SLS algorithm described in Section 4.3 is firstly applied. This allows us to obtain an initial estimate Θ_0, which is expected more closer to the actual one Θ. This aims to spare the robotized probe possible backlash, that might result from wrong estimation of the normal vector in the control law. Right after the SLS algorithm has been performed for the first iterations, the recursive algorithm formulated by the relationships (4.8) and (4.9) is applied throughout the servoing.

The estimator parameters have been empirically tuned to $\beta = 0.95$, f_0 =1e8, $\beta_0 = \frac{1}{20 \times f_0}$, $N_{LS} = 10$, and ϵ_0 =1e-10. The corresponding simulation results are shown on Fig. C.1. The visual features errors exponentially converge to zero as can be seen on Fig. C.1(e), and the reached image corresponds to the desired one despite the large difference with the initial one as can be seen on Fig. C.1(d). The system behavior is quite correct [see Fig. C.1(f)], and the probe motions are smooth [see Fig. C.1(a), C.1(b), and C.1(c)]. These results are similar to those obtained with the model-based servoing. Consequently, they validate the model-free visual servoing method that is based on straight line estimation. The estimated

parameters $\hat{\Theta}$ are shown on Fig. C.2.

We now consider the case where an additive measurement noise perturbs the image. Similarly to the simulations conducted for the model-based servoing, the noise consists in a random white Gaussian signal of 0.4 mm amplitude. The corresponding simulation results are shown on Fig. C.3. The results are satisfactory, where the visual features errors exponentially converge to zero and the reached image corresponds to the desired one. The probe behavior is correct as can be seen on Fig. C.3(g), despite the effect of the noise on the image as can be seen on Fig. C.3(b). The estimated parameters are plotted on Fig. C.4. Other simulations have been conducted to test up to what noise amplitude the servoing system can still perform. We have noticed that the system did not converge when the measurement noise is over 0.5 cm amplitude. However, note again that a pixel difference is performed to compute $\mathbf{d_i}$, as it is the case for the simulations presented in Section 5.2.2.

We test the system for different values of the estimator parameters. We consider two sets of simulations. In the first one, parameter β is varied while the remaining parameters are fixed to f_0 =1e8, $\beta_0 = \frac{1}{20\times f_0}$, ϵ_0 =1e-10, $N_{LS} = 10$. We show on Fig. C.5 results obtained for β set to 1.0, 0.5, and 0.04. We noticed that the system well behaved for β's values ranging from 0.5 to 1.0. It diverged only for values below 0.04.

In the second set we consider the case where ϵ_0 is varied and the remaining parameters are fixed throughout the tests to $\beta = 0.95$, f_0 =1e8, $\beta_0 = \frac{1}{20\times f_0}$, $N_{LS} = 10$. We show on Fig. C.6 results obtained for ϵ_0 equal to 1e-40 and 1e-7. The results for ϵ_0 =1e-10 have already been reported earlier in this appendix. We can notice that the servoing system is quite tolerable to the values that ϵ_0, and thus Θ_0, might have.

The system has also been tested for different values of N_{LS} and of f_0. It was noticed that it similarly behaves for N_{LS} values ranging for example, from 5 to 30, and for f_0 values above 1. For very small values of this latter, for example below 0.001, the convergence is relatively slow. We thus can conclude that the model-fee servoing using the straight line-based method is quite tolerant also to the values that f_0 and N_{LS} might have.

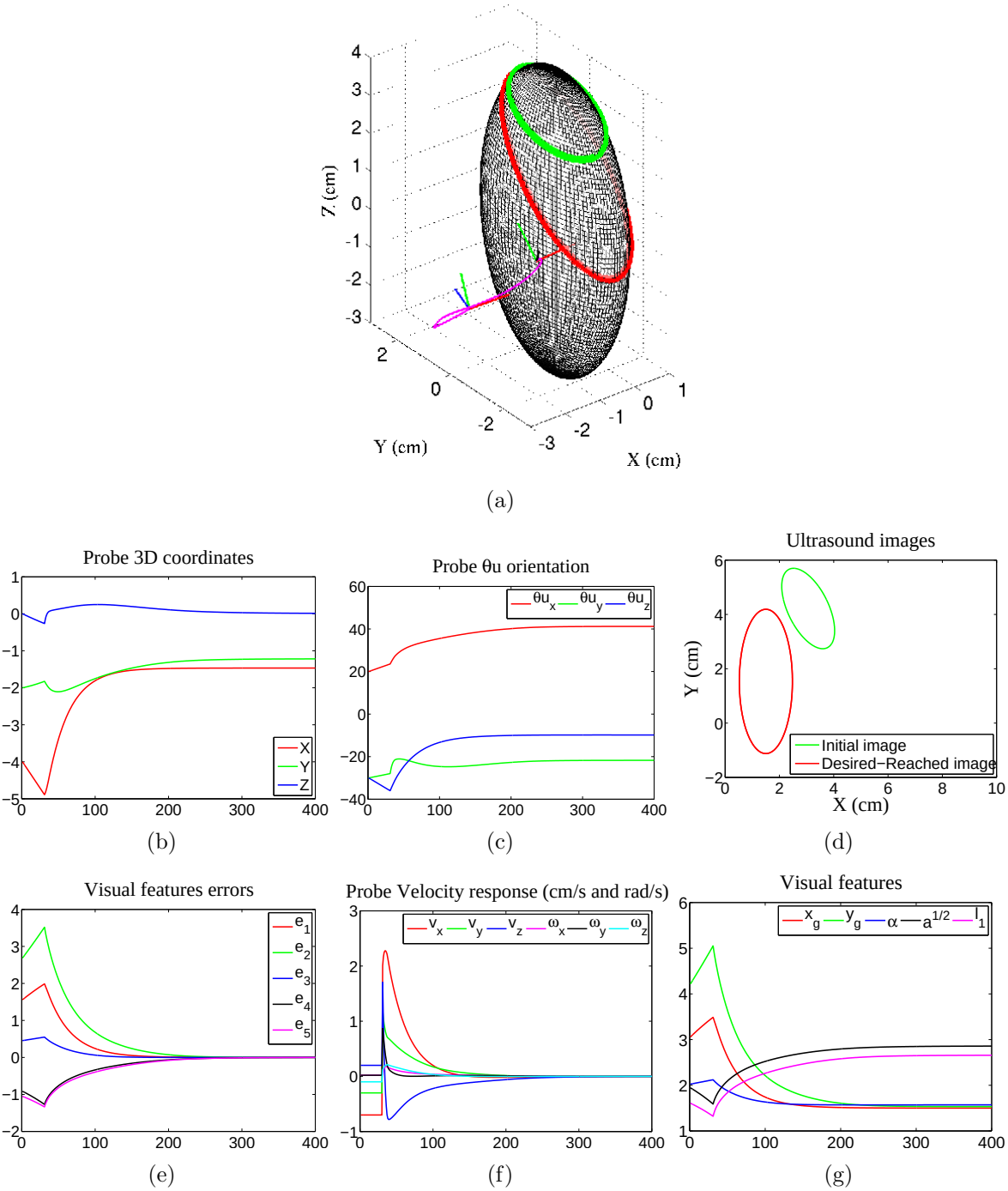


Figure C.1: **Model-free** visual servoing using the **straight line-based estimation** method, in a perfect case where no perturbation is present. The visual features and their corresponding errors are in (cm, cm, rad, cm, cm).

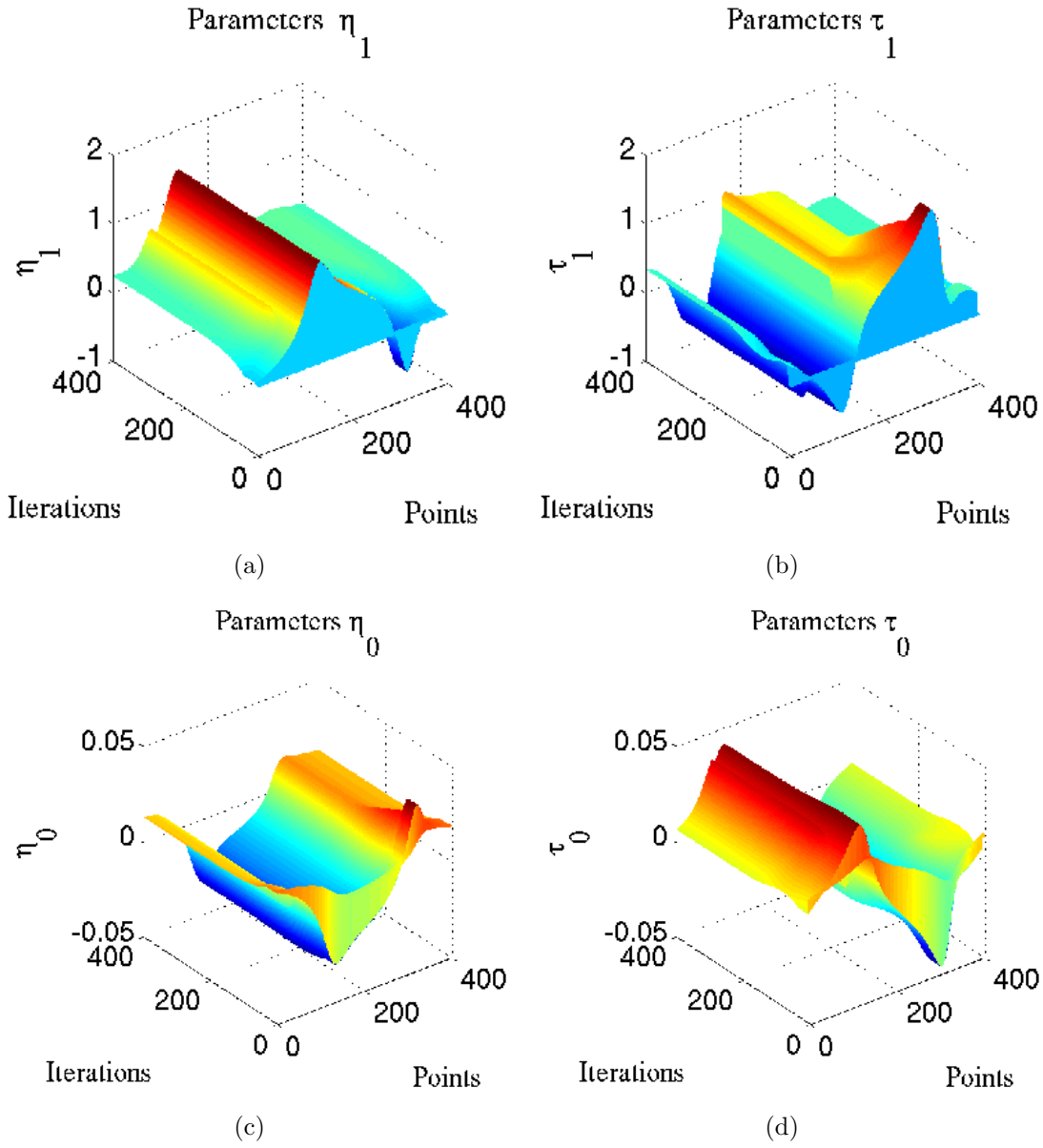


Figure C.2: Estimated parameters $\hat{\Theta}$ corresponding to the experiment whose results are shown on Fig. C.1.

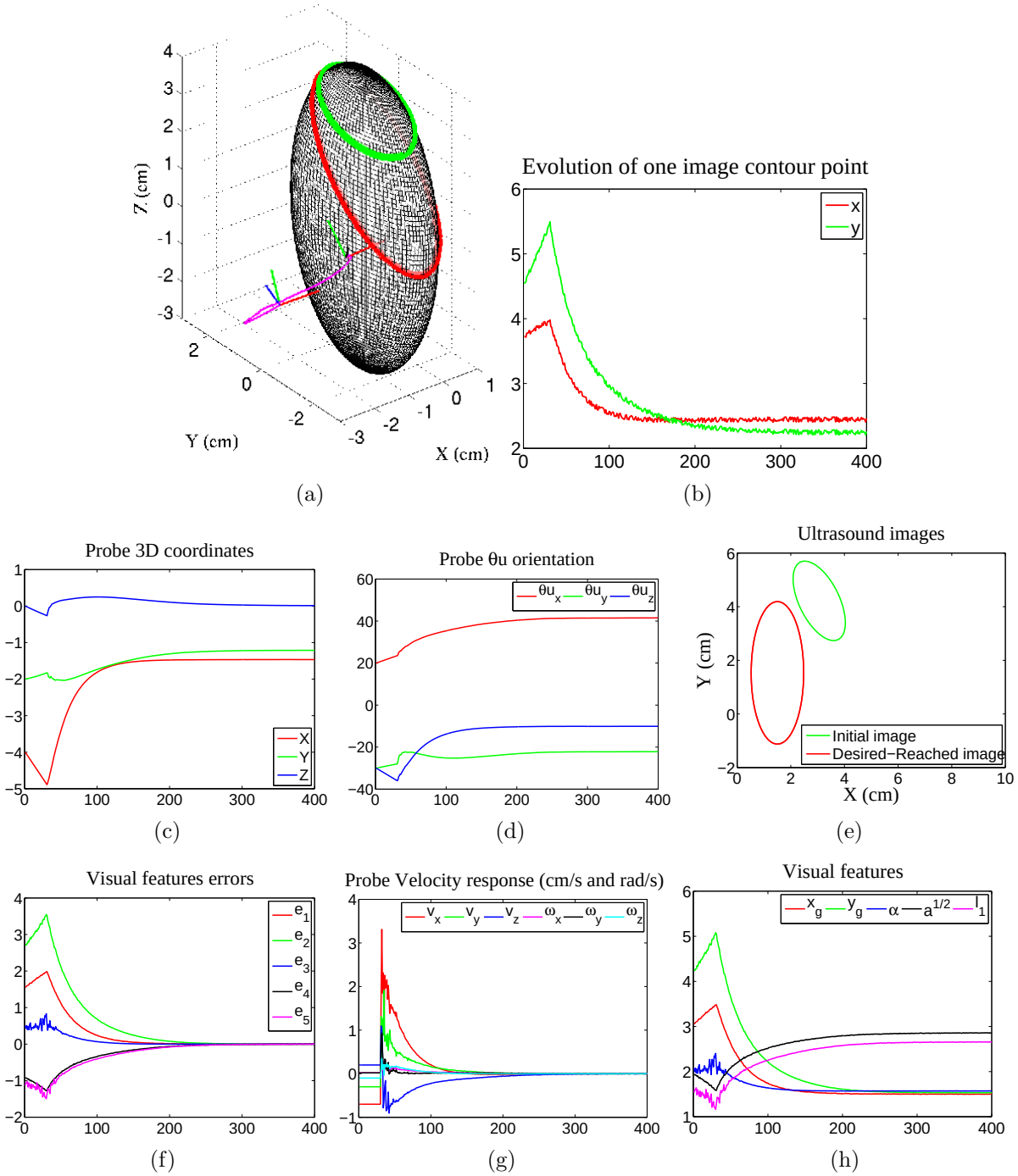


Figure C.3: **Model-free** visual servoing using the straight line-based estimation technique, in the presence of an additive measurement noise of 0.4 cm amplitude. The visual features and their corresponding errors are in (cm, cm, rad, cm, cm).

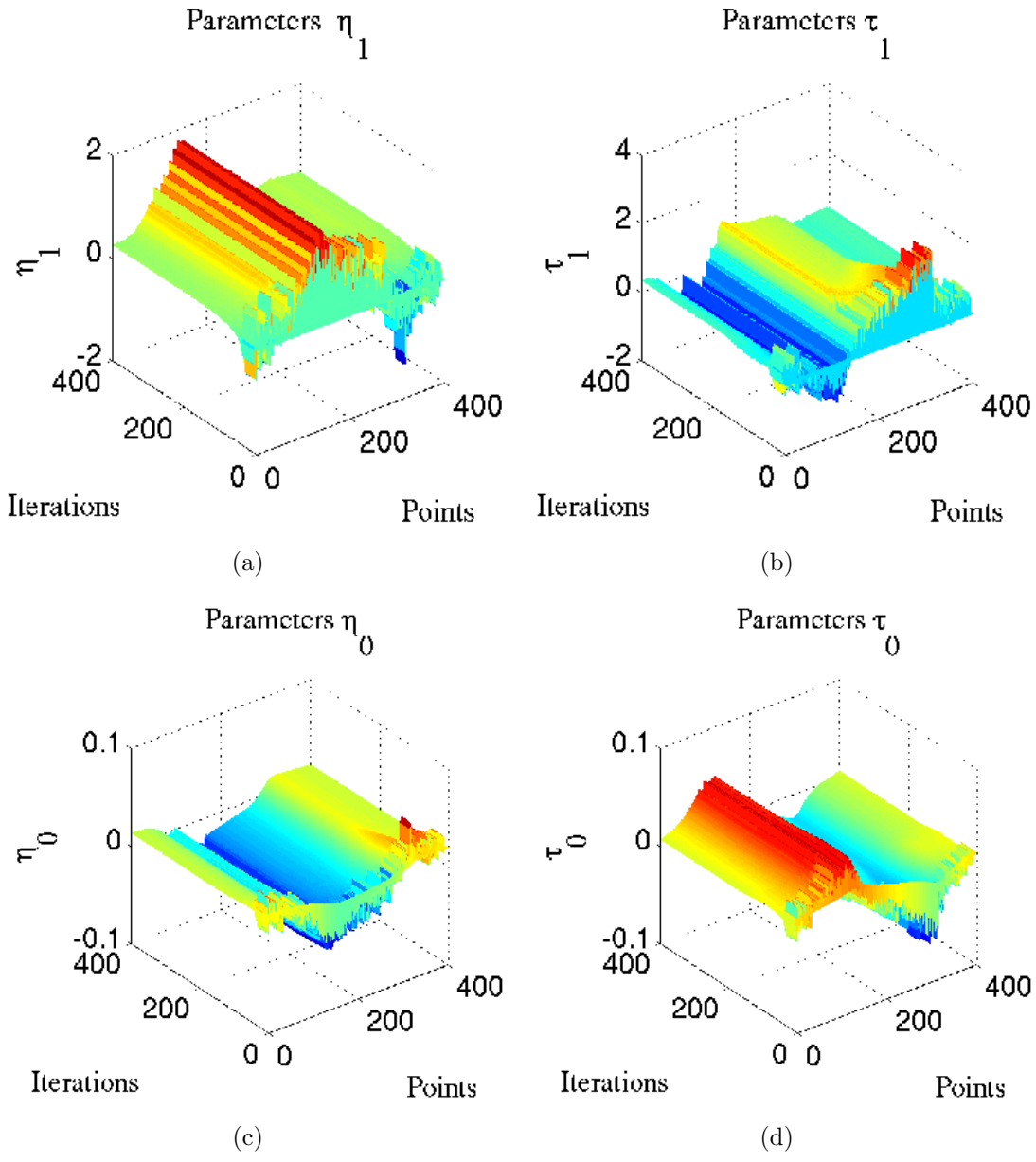


Figure C.4: Estimated parameters $\hat{\Theta}$ corresponding to the results shown on Fig. C.3.

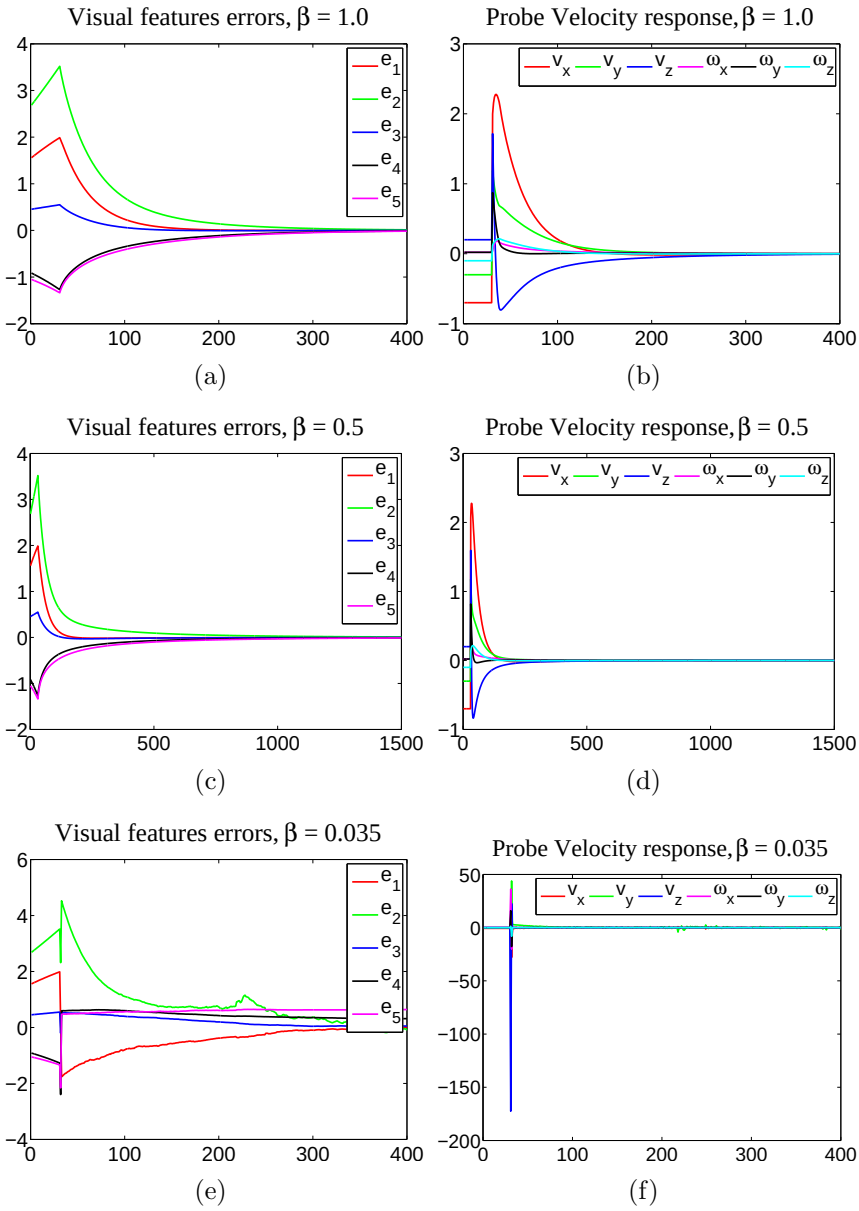


Figure C.5: Results obtained by employing the **model-free** visual servoing using the **straight line-based estimation** for different values of the parameter β. The visual features errors are in (cm, cm , rad, cm, cm), and the probe velocity is in (cm/s and rad/s).

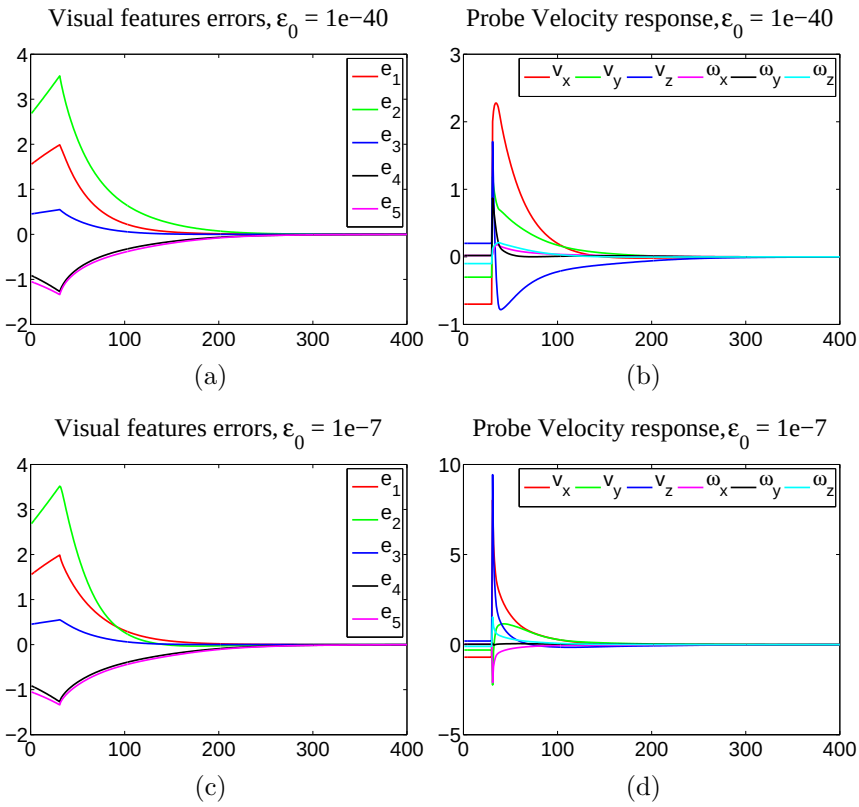


Figure C.6: Results obtained by employing the **model-free** visual servoing using the **straight line**-based estimation for different values of the parameter ϵ_0. The visual features errors are in (cm, cm , rad, cm, cm), and the probe velocity is in (cm/s and rad/s).

C.1.2 Using the quadric surface-based method

The model-free visual servoing method that uses the quadric surface-based estimation, presented in Section 4.2, is finally tested. The simulation scenario is the same as before. The SLS algorithm is applied for only the first N_{LS} iterations in order to obtain an initial estimate Θ_0. Then, the servoing is launched where the recursive algorithm, formulated by the relationships (4.8) and (4.16), is employed, instead of the SLS one, throughout the servoing.

We firstly test the method in a perfect condition where no measurement noise is present. The estimator parameters are tuned to $\beta = 1.0$, f_0 =1e2, $N_{LS} = 21$, and ϵ_0 =1e-20. As for length N of the contour segments, that update the algorithm at each iteration, as described in Section 4.2, it is tuned to $N = 21$. The corresponding simulation results are shown on Fig. C.7, while the estimated parameters are plotted in Fig. C.8. The results are also satisfactory, which validates the model-free visual servoing method based on quadric surface estimation, for a perfect condition.

The method is now tested when measurement perturbations are present in the image. The noise also consists in a random white Gaussian signal of 0.4 cm amplitude. The corresponding simulation results are shown on Fig. C.9 and Fig. C.10. We can see that they are clearly not satisfactory. The probe velocity is shaky, as can be seen on Fig. C.9(g), which results also in a shaky probe path as can be seen on Fig. C.9(a). Thus, we can conclude that the model-free method based on quadric surface estimation is not robust. Moreover, it was not easy to tune the estimator parameters, compared to the two previously tested model-free servoing methods.

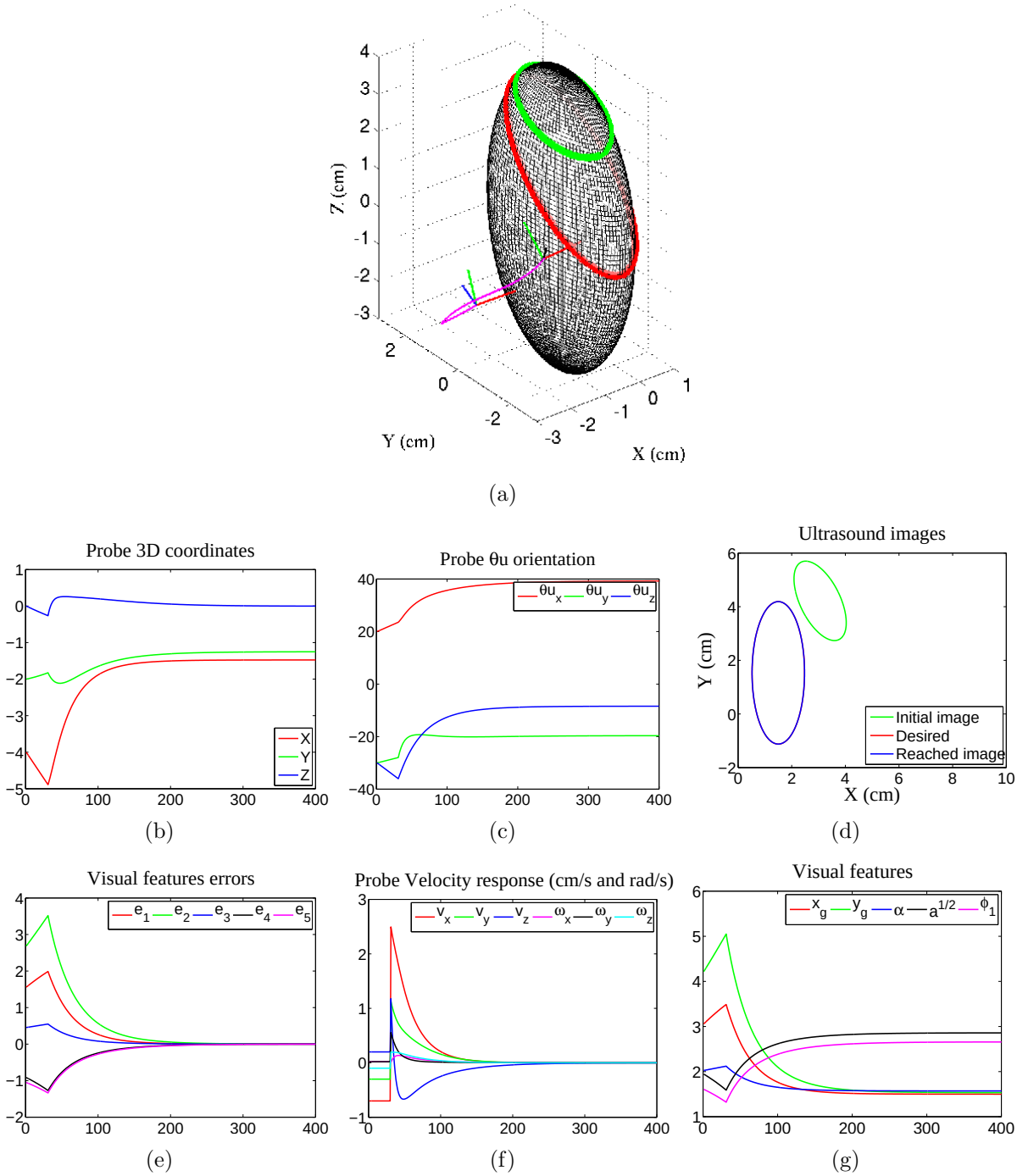


Figure C.7: **Model-free** visual servoing using the **quadric** surface-based estimation method, in a perfect case where no measurement noise is present. The visual features and their corresponding errors are in (cm, cm, rad, cm, cm).

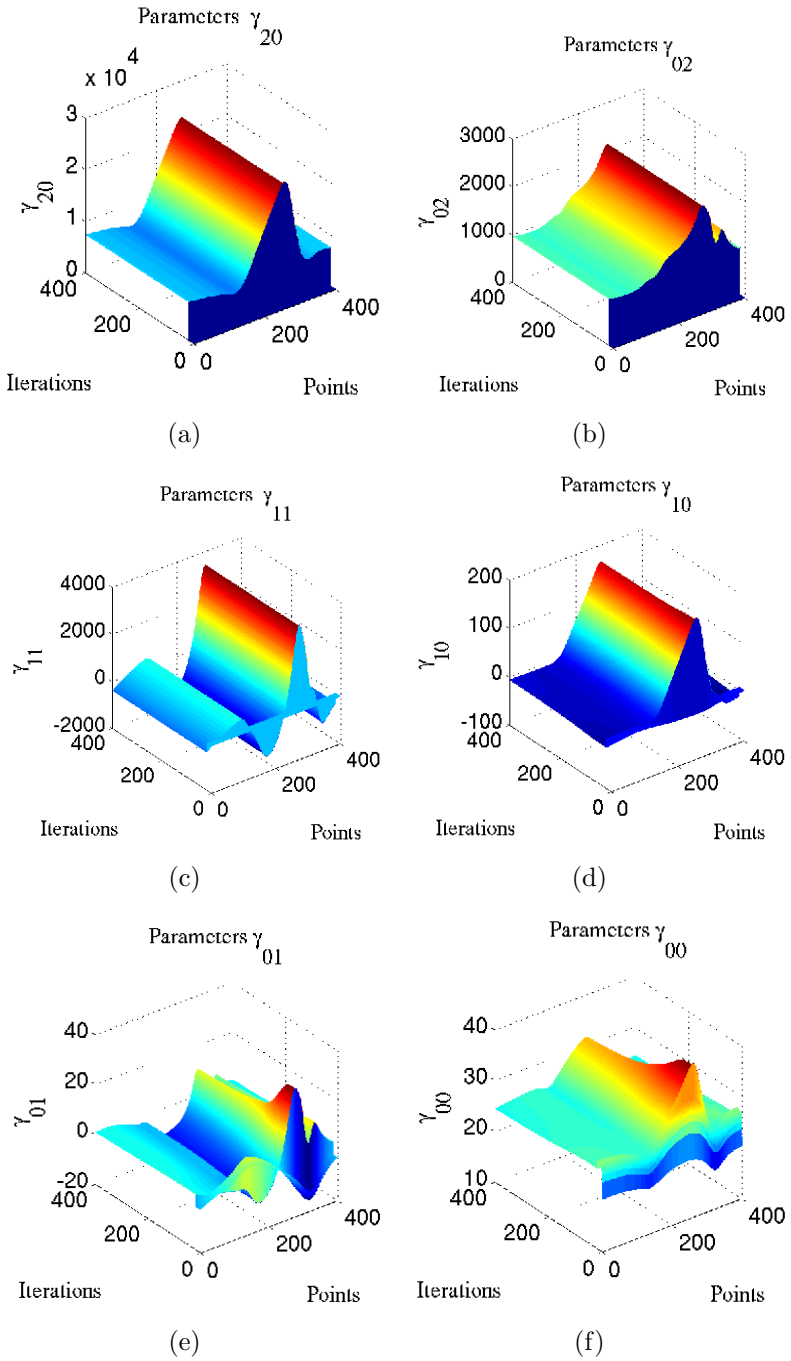


Figure C.8: Estimated parameters $\hat{\Theta}$ corresponding to the results shown on Fig. C.7.

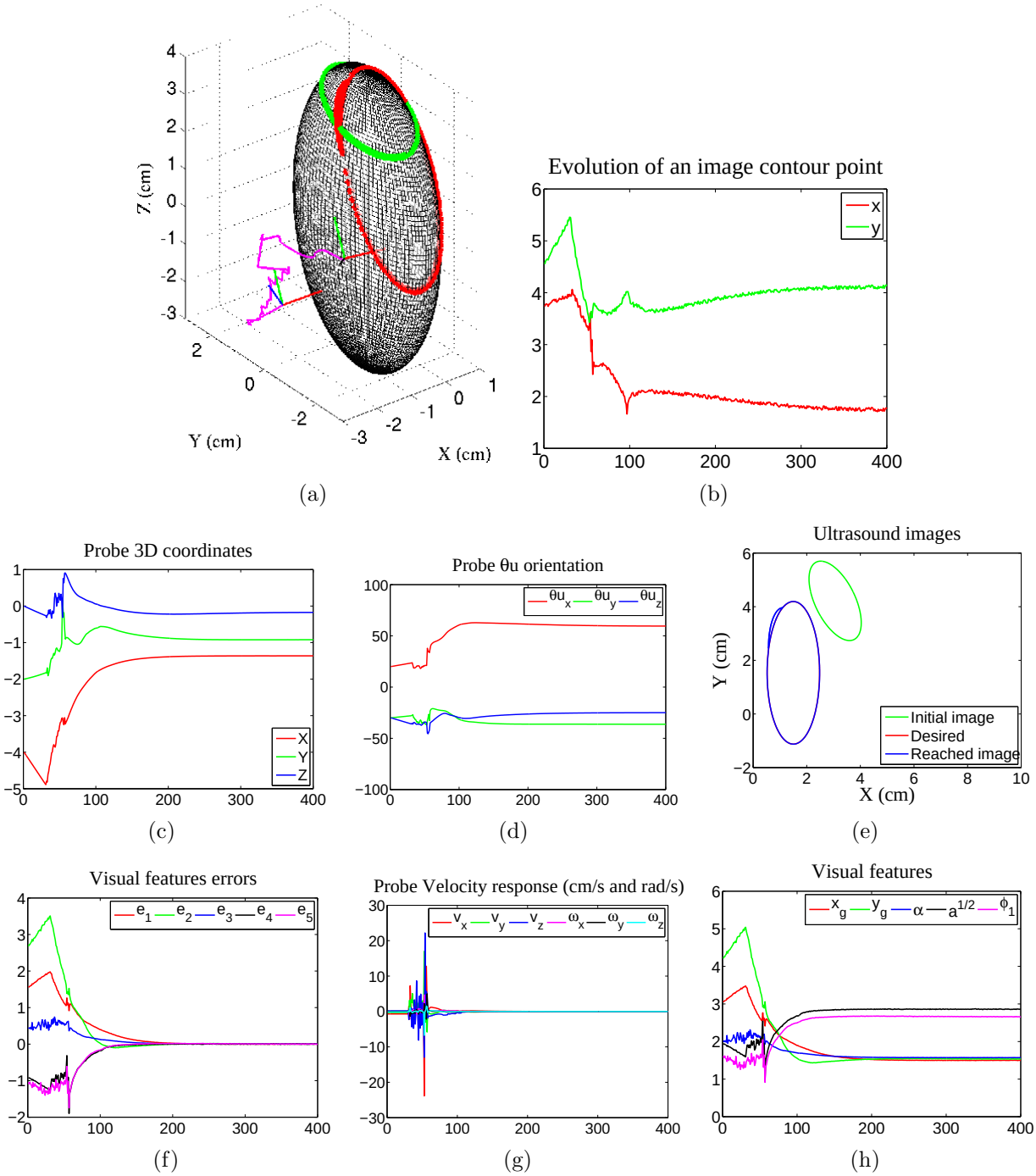


Figure C.9: **Model-free** visual servoing using the **quadric** surface-based estimation method, when measurement noises of 0.4 mm amplitude are introduced in the image. The visual features and their corresponding errors are in (cm, cm, rad, cm, cm).

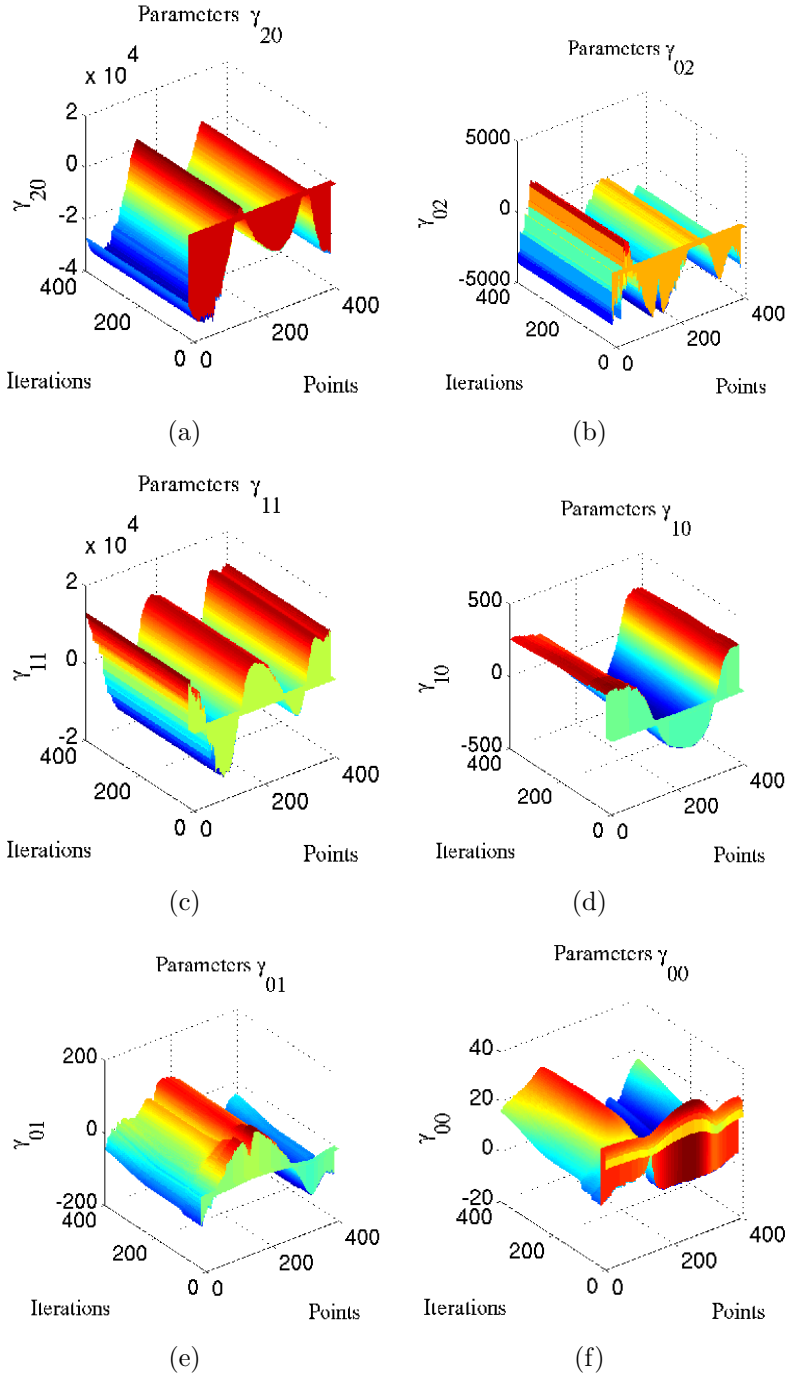


Figure C.10: Estimated parameters $\hat{\Theta}$ corresponding to results shown on Fig. C.9.

C.2 Simulations with realistic ultrasound images

In this section we test the straight line- and the quadric surface-based model-free visual servoing methods using realistic ultrasound images. The images are afforded with the simulator described in Section 5.3. The task to be achieved by the virtual probe is also presented in that section.

C.2.1 Straight line-based estimation

The estimator parameters are tuned to $\beta = 0.8$, f_0 =1e4, $\beta_0 = \frac{1}{20 \times f_0}$, ϵ_0 =1e-10, and $N_{LS} = 10$ iterations. The corresponding results are shown on Fig. C.11 and Fig. C.12. The visual features errors converge to zero as can be seen on Fig. C.11(e), and both two target image sections have been reached as can be seen respectively on Fig. C.11(b) and Fig. C.11(d). These results show the validity of the model-free method that uses the straight line-based estimation.

C.2.2 Quadric surface-based estimation

Finally, we test the servoing method that uses the quadric surface-based estimation. The estimator parameters are tuned to $\beta = 1.0$, f_0 =1e5, $\beta_0 = \frac{1}{20 \times f_0}$, ϵ_0 =1e-20, $N_{LS} = 17$ iterations, and $N = 17$ points. Corresponding results are shown on Fig. C.13 and Fig. C.14. In contrast to those previously obtained with the two other servoing methods, these results are however not satisfactory, where both the visual features do not converge smoothly, as can be seen on Fig, C.13(e), and the probe velocity is very shaky, as can be seen on Fig. C.13(f). Moreover, it was relatively tedious to tune the estimator parameters, where we noticed that with this method the system is highly sensitive to their variation. The obtained outcome was in fact expected, because of the low performances this servoing method had previously shown in the simulation presented in Section C.1.2. Those performances indeed seem to agree with the present ones.

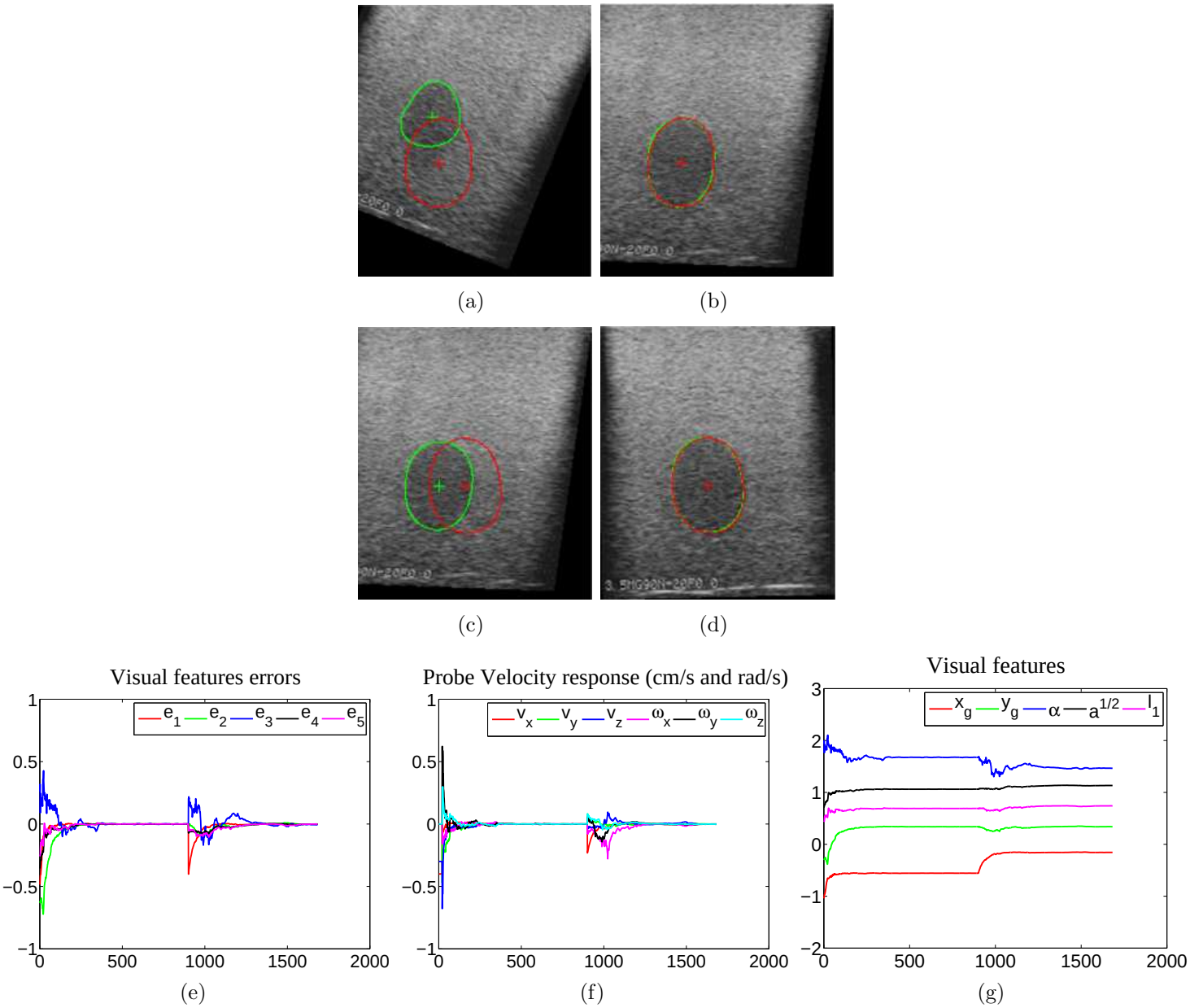


Figure C.11: **Model-free** visual servoing using the **straight** line-based estimation method performed on a realistic ultrasound 3D volume - (a) Initial image, whose section is contoured with green, captured right before the servoing is launched. The contour, of the first target image section, is displayed in red and is superimposed on the image - (b) The first target is automatically reached, where the observed (green) and the desired contour (red) now become superimposed - (d) The second target (red) is ordered - (d) The second target is automatically reached. The visual features and their corresponding errors are in (cm, cm, rad, cm, cm).

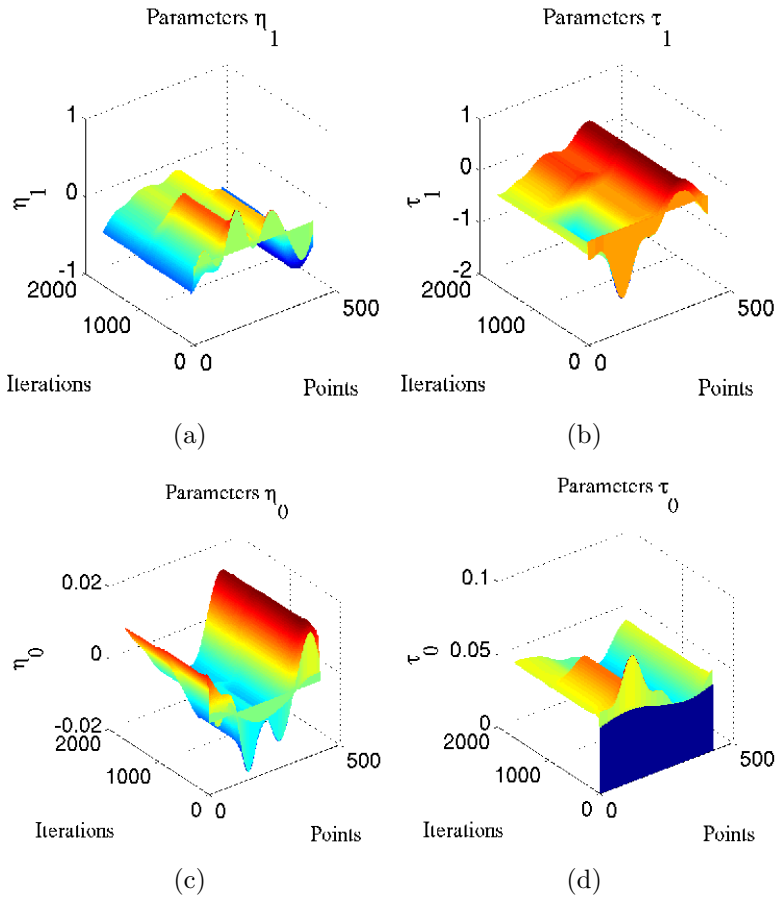


Figure C.12: Estimated Parameters $\hat{\Theta}$ corresponding to the results shown on Fig. C.11.

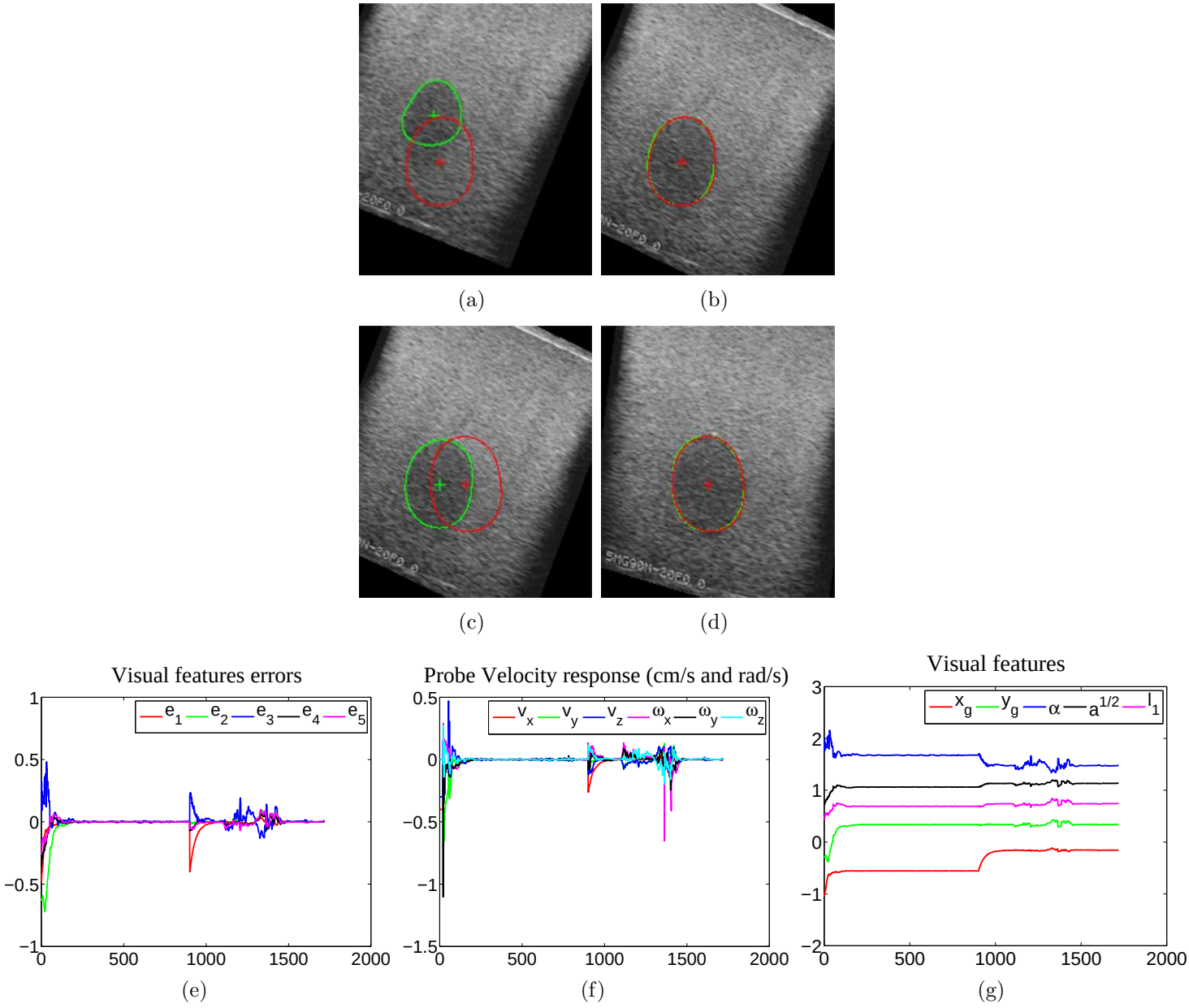


Figure C.13: **Model-free** visual servoing using the **quadric** surface-based estimation method performed on a realistic ultrasound 3D volume. The visual features and their corresponding errors are in (cm, cm, rad, cm, cm).

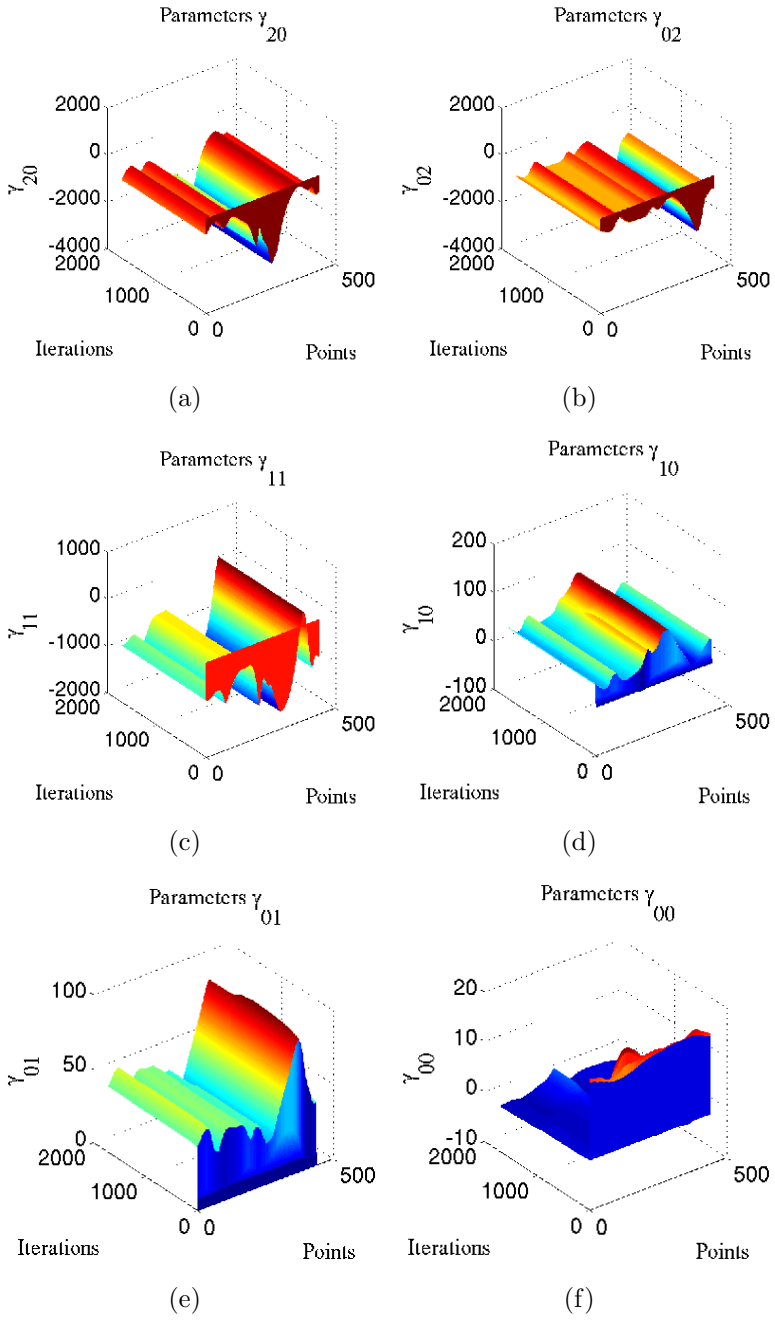


Figure C.14: Estimated Parameters $\hat{\Theta}$ corresponding to the results shown on Fig. C.13.

C.3 Simulations with the binary volume

Following the results shown in Section 5.4, the straight line- and the quadric-based model-free servoing methods are tested on a binary object.

C.3.1 Straight line-based estimation

The estimator parameters are tuned to $\beta = 0.8$, f_0 =1e6, $\beta_0 = \frac{1}{20 \times f_0}$, ϵ_0 =1e-10, and $N_{LS} = 10$ iterations. The corresponding simulation results are shown on Fig. C.15 and Fig. C.16. The visual features errors converge to zero, roughly exponentially, as can be seen on Fig. C.15(e), and the reached images correspond to the desired ones as can be seen respectively on Fig. C.15(b) and C.15(d), despite the large initial differences. The system behavior is quite correct as can bee seen on Fig. C.15(f), and the path performed by the probe is thus also quite smooth as can be seen on Fig. C.15(g). More, the objective has been achieved, since the two reached poses correspond respectively to the poses were the first and second target images had been captured. We obtained positioning errors of (4.4×1e-4, 9.1×1e-4, 2.7×1e-4, 0.0469, 0.0745, -0.0573)(cm and deg) for the former and (0, 4.2x1e-4, -2.09x1e-3, -0.2865, -0.3323, -0.0017)(cm and deg) for the latter positioning.

C.3.2 Quadric surface-based estimation

As for the model-free method based on quadric surface estimation, it has considerably underperformed the straight and the curved lines-based estimation methods. In the same simulation conditions this method has completely diverged. For small displacements from the target image, it nevertheless converged the system to the desired target as can be seen on Fig. C.17. The estimator parameters were tuned to $\beta = 0.7$, f_0 =1e2, $\beta_0 = \frac{1}{20 \times f_0}$, ϵ_0 =1e-20, $N_{LS} = 41$ iterations, and $N = 41$ points. However, the probe velocity is too shaky, as can be clearly seen on Fig. C.17(e), although the images are not noisy but perfectly contrasted (image section in black and the background in white). This shakiness is thus due to the quadric-based method itself and not to image noise. Indeed, in this same simulation condition the two other methods performed quite well, as can be seen on Fig. C.19.

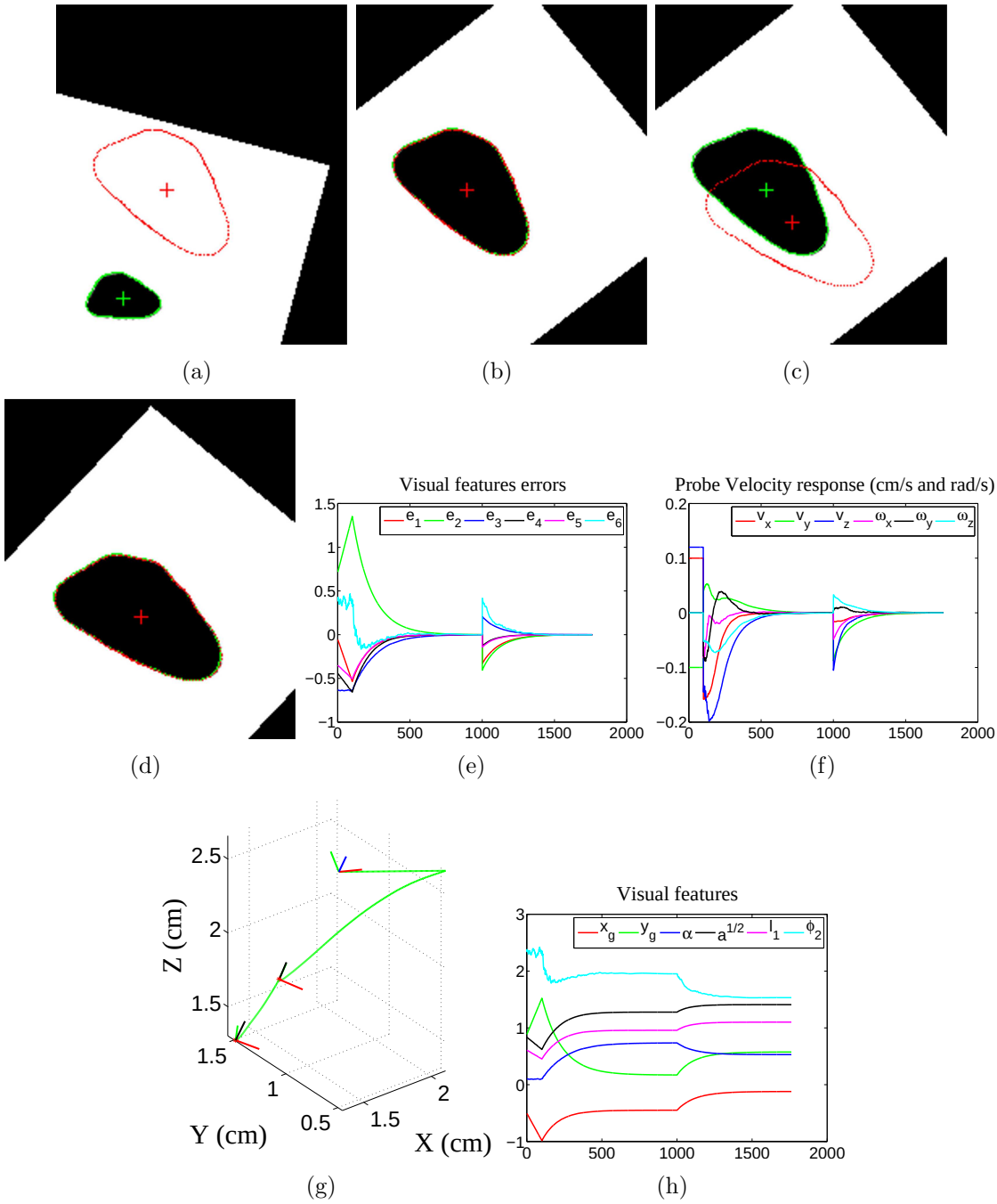


Figure C.15: **Model-free** visual servoing that uses the **straight** line-based estimation, tested on a simulated binary object (a) Initial image acquired right before launching the servoing, where the actual section is contoured with green. The contour of the target image section is displayed in red and superimposed on the image - (b) The first target is reached - (c) The second target is ordered - (d) The second target image is reached - (g) The probe initial pose frame is indicated by its cartesian frame whose (X, Y, Z) axes are respectively represented by the red, green and blue segments. The probe path is plotted in green. At the poses of the first and second target, the Z axis is represented with a black segment - (h) and (e) The visual features and their corresponding errors are in (cm ,cm, rad, cm, cm, 10×unit).

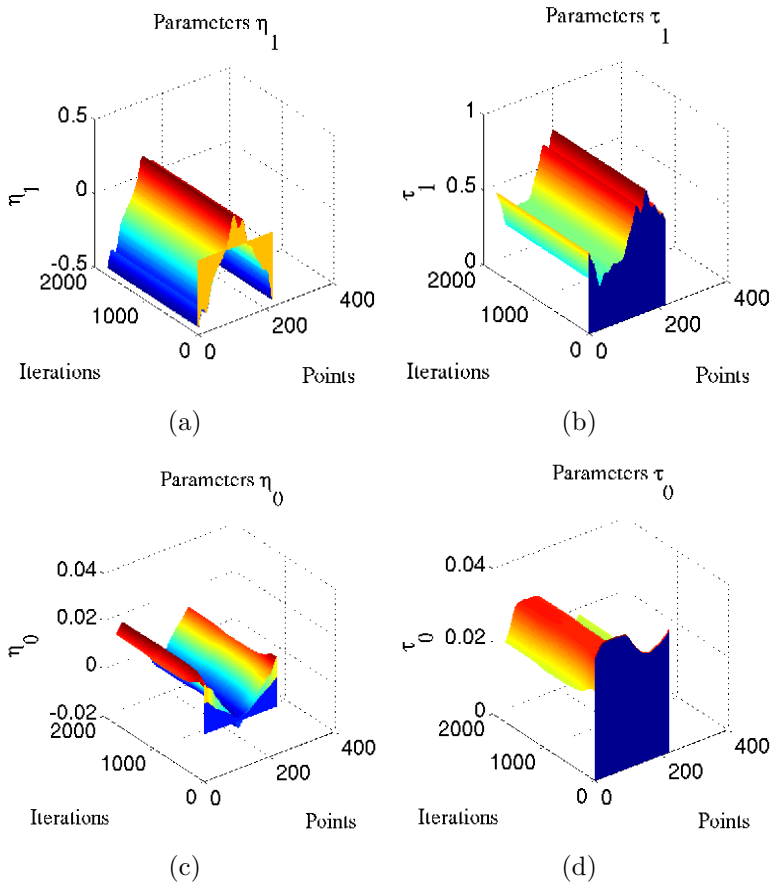


Figure C.16: Estimated parameters $\hat{\Theta}$ corresponding to the results shown on Fig. C.15.

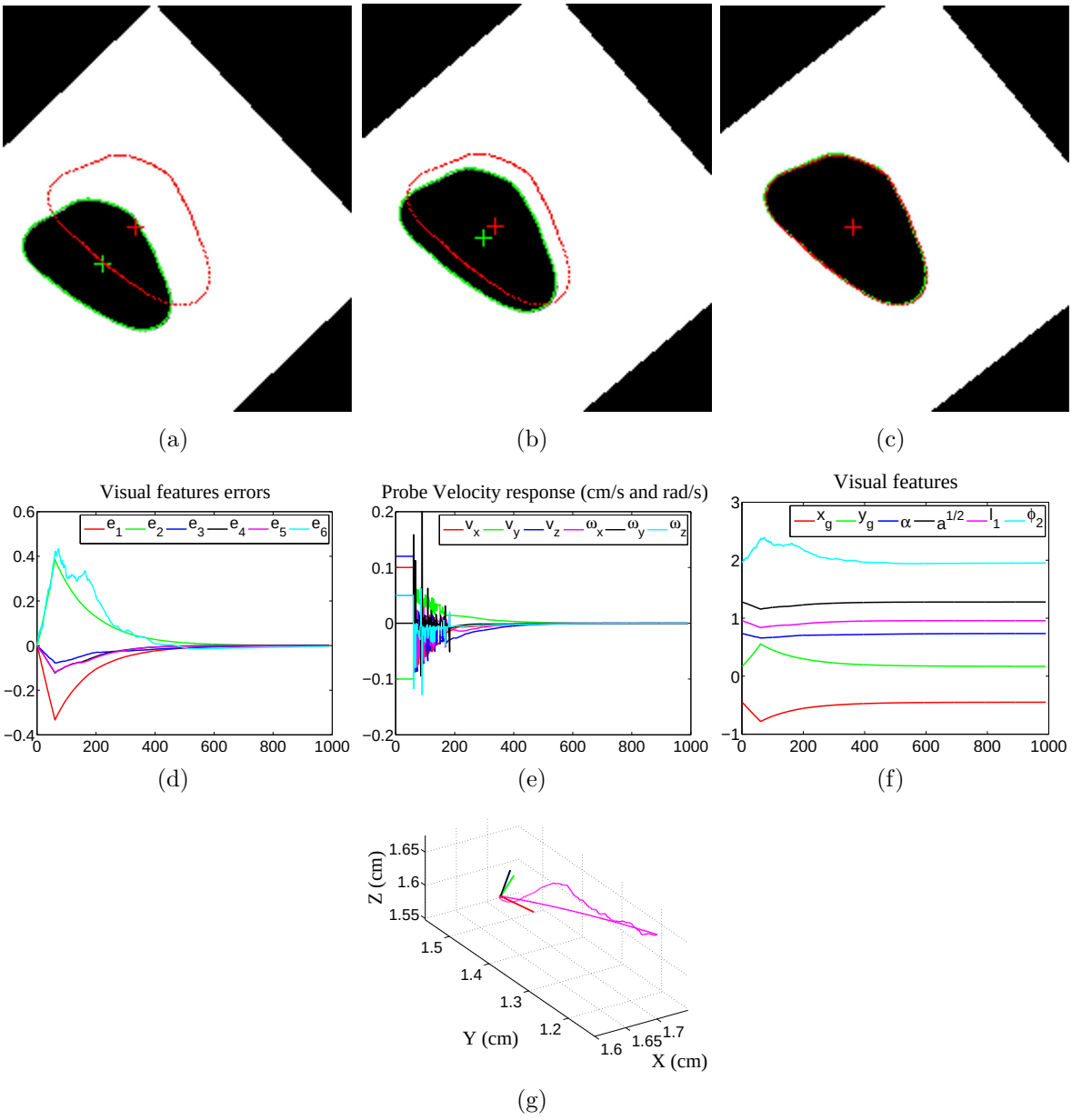


Figure C.17: **Model-free** visual servoing that uses the **quadric surface**-based estimation, tested on a simulated binary object for relatively small displacements.

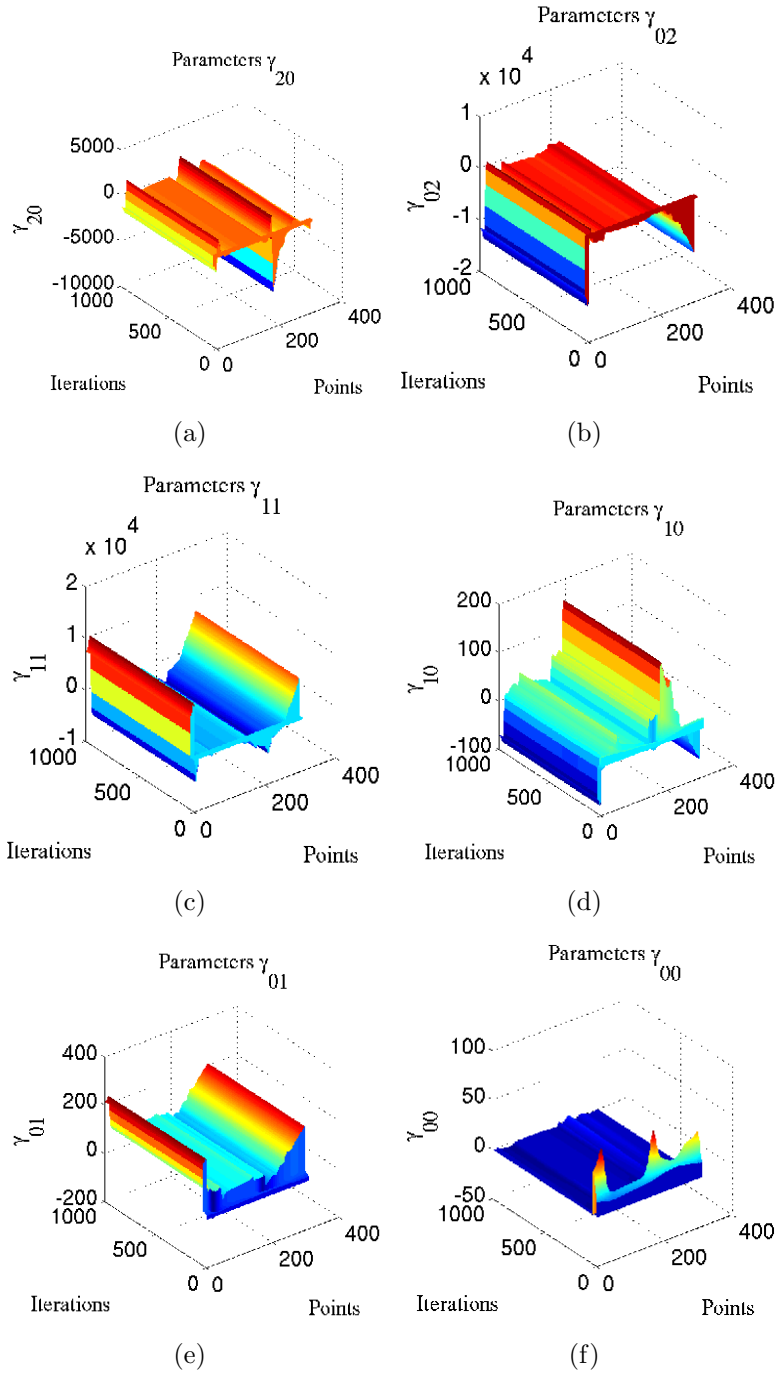


Figure C.18: Estimated parameters $\hat{\Theta}$ corresponding to the results shown on Fig. C.17.

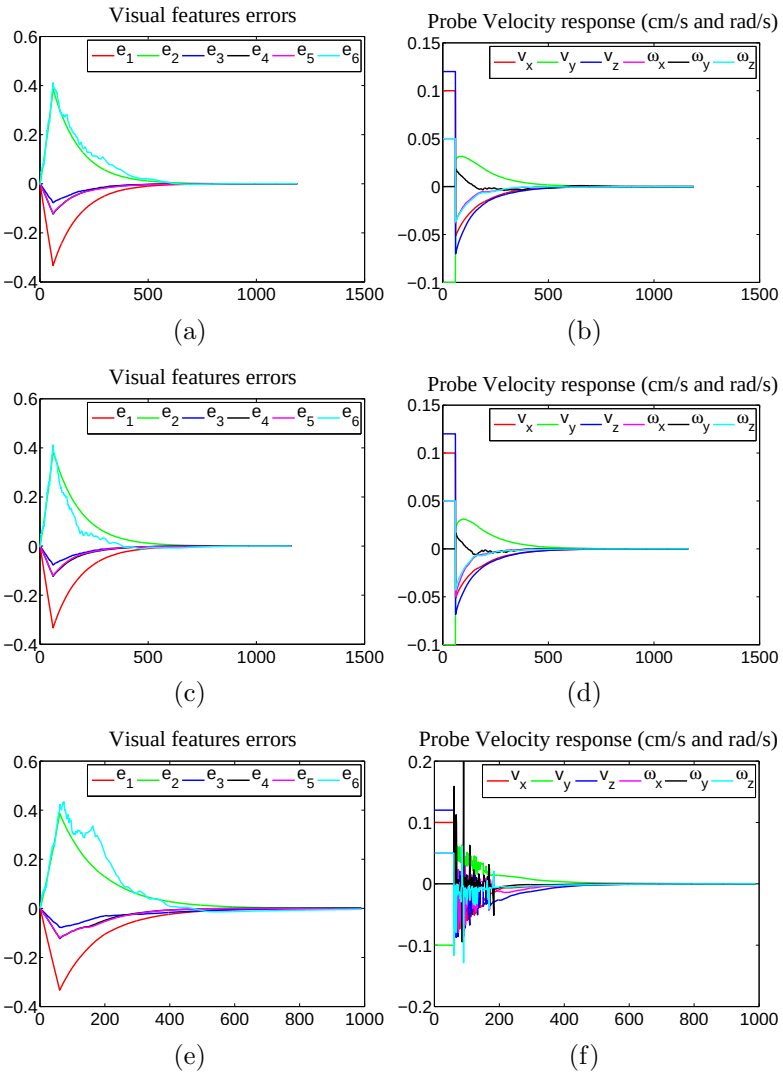


Figure C.19: Comparison of the performances obtained with the **model-free** servoing method that uses the **quadric** surface-based estimation (Fig. C.17, that are also reported here on the two figures at the bottom) to those obtained with the two methods that use respectively the **straight** line- (top) and the **curved** line-based estimation (middle).

Bibliography

[1] P. Abolmaesumi, S. E. Salcudean, W-H. Zhu, and M. R. Sirouspour. Image-guided control of a robot for medical ultrasound. *IEEE Trans. on Robotics*, 18(1):11–23, February 2002.

[2] Y. S. Abu-Mostafa and D. Psaltis. Recognitive aspects of moment invariants. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, PAMI-6(6):698–706, November 1984.

[3] M. Avriel. *Nonlinear programming: analysis and methods*. Dover Publications, 2003.

[4] W. Bachta and A. Krupa. Towards ultrasound image-based visual servoing. In *IEEE Int. Conf. on Robotics and Automation, ICRA'06*, pages 4112–4117, Orlando, Florida, May 2006.

[5] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic, New York, 1988.

[6] H. Bassan, T. Hayes, R. V. Patel, and M. Moallem. A novel manipulator for 3d ultrasound guided percutaneous needle insertion. In *IEEE Int. Conf. on Robotics and Automation, ICRA'07*, pages 617–622, Roma, Italy, April 2007.

[7] O. Bebek and M. Cavusoglu. Intelligent control algorithms for robotic assisted beating heart surgery. *In IEEE Trans. on Robotics*, 23(3):468–480, June 2007.

[8] S. O. Belkasim, M. Shridhar, and M. Ahmadi. Pattern recognition with moment invariants: a comparative study and new results. *Pattern Recogn.*, 24(12):1117–1138, 1991.

[9] Z. Bien, W. Jang, and J. Park. Characterization an use of feature-jacobian matrix for visual servoing. In K. Hashimoto, editor, *Visual Servoing*, pages 317–363. World Scientific, Singapore, 1993.

[10] E. M. Boctor, G. Fisher, M. A. Choti, G. Fichtinger, and G. Taylor. A dual-armed robotic system for intraoperative ultrasound guided hepatic ablative therapy: a prospective study. In *IEEE Int. Conf. on Robotics and Automation, ICRA'04*, pages 2517–2522, New Orleans, LA, April 2004.

[11] E. M. Boctor, R. J. Webster, M. A. Choti, R. H. Taylor, and G. Fichtinger. Robotically assisted ablative treatment guided by freehand 3d ultrasound. In *Int. Congress Series*, volume 1268, pages 503–508, 2004.

[12] L. N. Bohs, B. J. Geiman, M. E. Anderson, S. C. Gebhart, and G. E. Trahey. Speckle tracking for multi-dimensional flow estimation. *Ultrasonics*, 28(1), 2000.

[13] L. G. Brown. A survey of image registration techniques. *Computing Surveys*, 24(4):325–376, 1992.

[14] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In D. Kriegman, G . Hager, and A. S. Morse, editors, *The Confluence of Vision and Control*, number 237 in Lecture Notes in Control and Informations Systems, pages 66–78. Springer-Verlag, 1998.

[15] F. Chaumette. A first step toward visual servoing using image moments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'2002*, pages 378–383, Lausanne, Switzerland, October 2002.

[16] F. Chaumette. Image moments: A general and useful set of features for visual servoing. *IEEE Trans. on Robotics*, 20(4):713–723, August 2004.

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

[18] F. Chaumette and S. Hutchinson. Visual servo control, part ii: Advanced approaches. *IEEE Robotics and Automation Magazine*, 14(1):109–118, March 2007.

[19] C. C. Chen. Improved moment invariants for shape discrimination. *Pattern Recognition*, 26(5):683–686, 1993.

[20] C. M. Chen, H. H. S. Lu, and Y. C. Lin. A new ultrasound image segmentation algorithm based on an early vision model and discrete snake model. *SPIE*, 3338:959–970, 1998.

[21] C. Collewet. Polar snakes: A fast and robust parametric active contour model. In *IEEE Int. Conf. on Image Processing, ICIP'09*, Cairo, Egypt, November 2009.

[22] M. de Mathelin and R. Lozano. Robust adaptive identification of slowly time-varying parameters with bounded disturbances. *Automatica*, 35:1291–1305, July 1999.

[23] E. Dehghan, X. Wen, R. Zahiri-Azar, M. Marchal, and SE. Salcudean. Needle-tissue interaction modeling using ultrasound-based motion estimation: Phantom study. *Computer Aided Surgery*, 13(5):265–280, September 2008.

[24] C. Doignon, B. Maurin, B. Bayle, and M. de Mathelin. A visual 3d-tracking and positioning technique for stereotaxy with ct scanners. *Robotics and Autonomous Systems*, 56:385–395, 2008.

[25] S. A. Dudani, K. J. Breeding, and R. B. McGhee. Aircraft identification by moment invariants. *IEEE Transactions on Computers*, 26(1):39–45, January 1977.

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

[27] J.T. Feddema, C.S.G. Lee, and 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.

[28] O. Felfoul, E. Aboussouan, A. Chanu, and S. Martel. Real-time positioning and tracking technique for endovascular untethered microrobots propelled by mri gradients. In *IEEE Int. Conf. on Robotics and Automation, ICRA'09*, pages 2693–2698, Kobe, Japan, May 2009.

[29] A. Fenster, K. J. M. Surry, G. R. Mills, and D. B. Downey. 3d ultrasound guided breast biopsy system. In *Ultrasonics*, volume 42, pages 769–774. Elsevier B.V., 2004.

[30] G. S. Fischer, I. Iordachita, C. Csoma, J. Tokuda, P. W. Mewes, C. M. Tempany, N. Hata, and G. Fichtinger. Pneumatically operated mri-compatible needle placement robot for prostate interventions. In *IEEE Int. Conf. on Robotics and Automation, ICRA'08*, pages 2489–2495, Pasadena, CA, USA, May 2008.

[31] Ioana Fleming, Marcin Balicki, John Koo, Iulian Iordachita, Ben Mitchell, James Handa, Gregory D. Hager, and Russell H. Taylor. Cooperative robot assistant for retinal microsurgery. In *MICCAI (2)*, pages 543–550, 2008.

[32] J. Flusser and T. Suk. Rotation moment invariants for recognition of symmetric objects. *IEEE Trans. on Image Processing*, 15(12):3784, 3790 2006.

[33] N. Friedland and D. Adam. Automatic ventricular cavity boundary detection from sequential ultrasound images using simulated annealing. *IEEE Trans. on Medical Imaging*, 8(4):344–353, December 1989.

[34] A. H. Gee, R. J. Housden, P. Hassenpflug, G. M. Treece, and R. W. Prager. Sensorless freehand 3d ultrasound in real tissue: speckle decorrelation without fully developed speckle. *Medical Image Analysis*, 10(2):137–149, 2006.

[35] A. H. Gee and R. W. Prager. Sequential 3d diagnostic ultrasound using the stradx system. In C. Tayor and A. Colchester, editors, *Medical Image Computing and Computer Assisted Intervention, MICCAI'99*, pages 716–725, New York, 1999. Springer.

[36] R. Ginhoux, J. Gangloff, M. de Mathelin, L. Soler, M. A. Sanchez, and J. Marescaux. Active filtering of physiological motion in robotized surgery using predictive control. *In IEEE Trans. on Robotics*, 21(1):27–79, 2006.

[37] G. D. Hager and N. Belhumer. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, 1998.

[38] J. Hong, T. Dohi, M. Hashizume, K. Konishi, and N. Hata. An ultrasound-driven needle insertion robot for percutaneous cholecystostomy. *Phys. Med. Biol.*, 49(3):441–454, 2004.

[39] P. V. C. Hough. A method and means for recognizing complex patterns. In *US patent 3,069,654*. 1962.

[40] M. K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, February 1962.

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

[42] M. Jakopec, S. J. Harris, F. R. Y. Baena, P. Gomes, J. Cobb, and B. L. Davies. The first clinical application of hands-on robotic knee surgery system. *Comput. Aided Surgery*, 6:329–339, 2001.

[43] G. Kreisselmeier. Stabilized least-squares type adaptive identifiers. *IEEE Transactions on Automatic Control*, 35(3):306–309, March 1990.

[44] A. Krupa and F. Chaumette. Control of an ultrasound probe by adaptive visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'05*, volume 2, pages 2007–2012, Edmonton, Canada, August 2005.

[45] A. Krupa, G. Fichtinger, and G. D. Hager. Full motion tracking in ultrasound using image speckle information and visual servoing. In *IEEE int. Conf. on Robotics and Automation ICRA'07*, pages 2458–2464, Rome, Italy, May 2007.

[46] A. Krupa, G. Fichtinger, and G.D. Hager. Real-time motion stabilization with b-mode ultrasound using image speckle information and visual servoing. *The International Journal of Robotics Research, IJRR*, 28(10):1334–1354, 2009.

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

[48] D. Lee, N. Koizumi, K. Ota, S. Yoshizawa, A. Ito, Y. Kaneko, Y. Matsumoto, and M. Mitsuish. Ultrasound-based visual servoing system for lithotripsy. In *In Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS 2007*, pages 877–882, San Diego, CA, USA, Oct. 29 - Nov. 2, 2007.

[49] Mitchell J. H. Lum, Diana C. W. Friedman, Hawkeye H. I. King, Regina Donlin, Ganesh Sankaranarayanan, Timoty J. Broderick, Mika N. Sinanan, Jacob Rosen, and Blake Hannaford. Teleoperation of a surgical robot via airborne wireless radio and transatlantic internet links. In *FSR*, pages 305–314, 2007.

[50] S. Maitra. Moment invariants. In *Proceedings of the IEEE*, volume 67, pages 697–699, April 1979.

[51] E. Malis. Improving vision-based control using efficient second-order minimization techniques. In *Int. Conf. on Robotics and Automation*, pages 1843–1848, New Orleans, LA, April 2004.

[52] A. G. Mamistvalov. N-dimensional moment invariants and conceptual mathematical theory of recognition n-dimensional solids. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):819–831, August 1998.

[53] E. Marchand, F. Spindler, and 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):40–52, December 2005.

[54] R. Mebarki, A. Krupa, and F. Chaumette. Image moments-based ultrasound visual servoing. In *IEEE Int. Conf. on Robotics and Automation, ICRA'08*, pages 113–119, Pasadena, California, May 2008.

[55] R. Mebarki, A. Krupa, and F. Chaumette. Modeling and 3d local estimation for in-plane and out-of-plane motion guidance by 2d ultrasound-based visual servoing. In *IEEE Int. Conf. on Robotics and Automation, ICRA'09*, Kobe, Japan, May 2009.

[56] R. Mebarki, A. Krupa, and F. Chaumette. 2d ultrasound probe complete guidance by visual servoing using image moments. *IEEE Trans. on Robotics*, 26(2):296–306, April 2010.

[57] R. Mebarki, A. Krupa, and C. Collewet. Automatic guidance of an ultrasound probe by visual servoing based on b-mode image moments. In D. Metaxas, L. Axel, G. Fichtinger, and G. Szekely, editors, *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention, MICCAI'08*, volume 5242 of *Lecture Notes in Computer Science, LNCS*, pages 339–346, New York, September 2008. Springer.

[58] G. Megali, O. Tonet, C. Stefanini, M. Boccadoro, V. Papaspyropoulos, L. Angelini, and P. Dario. A computer-assisted robotic ultrasound-guided biopsy system for video-assisted surgery. In W. Niessen and M. Viergever, editors, *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention, MICCAI'01*, volume 2208 of *Lecture Notes in Computer Science, LNCS*, pages 343–350. Springer, October 14-17 2001.

[59] R. Mukundan and K. R. Ramakrishnan. An iterative solution for object pose parameters using image moments. *Pattern Recognition*, 17(12):1279–1284, October 1996.

[60] Florent Nageotte, Philippe Zanne, Christophe Doignon, and Michel de Mathelin. Stitching planning in laparoscopic surgery: Towards robot-assisted suturing. *I. J. Robotic Res.*, 28(10):1303–1321, 2009.

[61] A. Nishikawa, T. Hosoi, K Koara, A. Negoro, D.and Hikita, H.and Miyazaki F. Asano, S.and Kakutani, M. Sekimoto, M. Yasui, Y. Miyake, and M. Takiguchi, S.and Monden. Face mouse: A novel human-machine interface for controlling the position of a laparoscope. *IEEE Transactions on Robotics and Automation*, 19(5):825–841, October 2003.

[62] P. M. Novotny, J. A. Stoll, P. E. Dupont, and R. D. Howe. Real-time visual servoing of a robot using three-dimensional ultrasound. In *IEEE Int. Conf. on Robotics and Automation, ICRA'07*, pages 2655–2660, Roma, Italy, April 2007.

[63] P. J. Parker and B. D. O. Anderson. Frequency tracking of nonsinusoidal periodic signals in noise. *Signal Processing*, 20:127–152, 1990.

[64] L. Phee, D. Xiao, J. Yuen, C. F. Chan, H. Ho, C. H. Thng, C. Cheng, and W. S. Ng. Ultrasound guided robotic system for transperineal biopsy of the prostate. In *IEEE Int. Conf. on Robotics and Automation, ICRA'05*, pages 1315–1320, Barcelona, Spain, April 2005.

[65] F. Pierrot, E. Dombre, E. Degoulange, L. Urbain, P. Caron, S. Boudet, J. Gariepy, and J. Megnien. Hippocrate: A safe robot arm for medical applications with force feedback. *Medical Image Analysis (MedIA)*, 3(3):285–300, 1999.

[66] R. J. Prokop and A. P. Reeves. A survey of moment-based techniques for unoccluded object representation and recognition. *CVGIP: Graph. Models Image Process.*, 54(5):438–460, 1992.

[67] S. E. Salcudean, G. Bell, S. Bachmann, W. H. Zhu, P. Abolmaesumi, and P. D. Lawrence. Robot-assisted diagnostic ultrasound – design and feasibility experiments. In *Medical Image Computing and Computer Assisted Intervention, MICCAI'99*, pages 1062–1071, New York, 1999. Springer.

[68] C. Samson, M. L. Borgne, and B. Espiau. Robot control : The task function approach. *Clarendon Press*, 1991.

[69] M. Sauvée, P. Poignet, and E. Dombre. Ultrasound image-based visual servoing of a surgical instrument through nonlinear model predictive control. *International Journal of Robotics Research, IJRR*, 27(1):25–40, 2008.

[70] W. Schroeder, K. Martin, and B. Lorensen. The visualization toolkit an object oriented approach to 3d graphics. In *3rd Edition, ISBN, 1-930934-12-2 Kitware.* Inc. Publishers.

[71] R. Sharma and S. Hutchinson. Optimizing hand/eye configuration for visual-servo systems. In *IEEE Int. Conf. on Robotics and Automation, ICRA'95*, pages 172–177, 1995.

[72] W. L. Smith and Fenster A. Optimum scan spacing for three-dimensional ultrasound by speckle statistics. *Ultrasound in Medicine and Biology*, 26(4):551–562, 2000.

[73] J. Stewart. Calculus, 2nd ed. *Pacific Grove, CA: Brooks/Cole*, 1991.

[74] J. A. Stoll, P. E. Dupont, and R. D. Howe. Ultrasound-based servoing of manipulators for telesurgery. In *Telemanipulator and Telepresence Technologies*, volume 4570, MA, USA, October 2001.

[75] J. A. Stoll, P. M. Novotny, R. D. Howe, and P. E. Dupont. Real-time 3d ultrasound-based servoing for a surgical instrument. In *IEEE Int. Conf. on Robotics and Automation, ICRA'06*, pages 613–618, Orlando, Florida, USA, May 2006.

[76] O. Tahri and 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.

[77] R. H. Taylor, B. D. Mittelstadt, H. A. Paul, W. Hanson, P. Kazanzides, J. F. Zuhars, B. Williamson, B. L. Musits, E. Glassman, and W. L. Bargar. An image-directed robotic system for precise orthopaedic surgery. *In IEEE Trans. on Robotics and Automation*, 10(3):261–275, June 1994.

[78] RH. Taylor and D. Stoianovici. Medical robotics in computer-integrated surgery. *IEEE Trans. on Robotics and Automation*, 19(5):765–781, October 2003.

[79] M. A. Vitrani, H. Mitterhofer, G. Morel, and N. Bonnet. Robust ultrasound-based visual servoing for beating heart intracardiac surgery. In *IEEE Int. Conf. on Robotics and Automation, ICRA'07*, pages 2021–3027, Roma, Italy, April 2007.

[80] M. A. Vitrani, G. Morel, N. Bonnet, and M. Karouia. A robust ultrasound-based visual servoing approach for automatic guidance of a surgical instrument with in vivo experiments. In *IEEE Int. Conf. on Biomedical Robotics and Biomechatronics, BIOROB'06*, Pisa, Italy, February 2006.

[81] M. A. Vitrani, G. Morel, and T. Ortmaier. Automatic guidance of a surgical instrument with ultrasound based visual servoing. In *IEEE Int. Conf. on Robotics an Automation, ICRA'05*, pages 508–513, Barcelona, Spain, April 2005.

[82] L.E. Weiss, A.C. Sanderson, and C.P. Neuman. Dynamic sensor-based control of robots with visual feedback. *IEEE Journal of Robotics and Automation*, 3(5):404–417, October 1987.

[83] G. Wells, C. Venaille, and C. Torras. Vision-based robot positioning using neural networks. *Image and Vision Computing*, 14(10):715–732, December 1996.

[84] P. N. T. Wells. Current status and future technical advances of ultrasonics imaging. *IEEE Engineering in Medicine and Biology*, pages 14–20, September-October 2000.

[85] F. Widjaja, C. Y. Shee, W. T. Latt, W. L. Au, P. Poignet, and W. T. Ang. Kalman filtering of accelerometer and electromyography (emg) data in pathological tremor sensing system. In *IEEE Int. Conf. on Robotics and Automation, ICRA'08*, pages 3250 – 3255, Pasadena, CA, May 2008.

[86] S. G. Yuen, P. M. Novotny, and R. D. Howe. Quasiperiodic predictive filtering for robot-assisted beating heart surgery. In *IEEE Int. Conf. on Robotics and Automation, ICRA'08*, pages 3875–3880, Pasadena, CA, USA, May 2008.

[87] R. Zahiri-Azar and SE. Salcudean. Motion estimation in ultrasound images using time domain cross-correlation with prior estimates. *IEEE Trans Biomed Eng*, pages 1990–2000, 2006.

[88] R. Zahiri-Azar and SE. Salcudean. Real-time estimation of lateral motion using time domain cross-correlation with prior estimates. In *Proceedings of the IEEE Int. Ultrasonics Symposium*, pages 1209–1212, Vancouver, British Columbia, Canada, October 2006.

[89] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method*, volume 1. Butterworth-Heinemann, 5th edition, 2000.

Résumé

Cette thèse présente de nouvelles méthodes permettant de guider automatiquement une sonde échographique actionnée par un robot médical. Nous proposons d'utiliser des informations visuelles basées sur des moments 2D extraits de l'image échographique. Cela nécessite l'obtention de la matrice d'interaction qui lie ces informations à la vitesse de la sonde (ou bien du robot porteur de cette dernière). Nous présentons de nouvelles bases théoriques qui ont permis de développer l'expression analytique exacte de cette matrice. Nous avons également élaboré six informations visuelles indépendantes permettant de contrôler les 6 degrés de liberté de la sonde et ainsi pouvoir positionner automatiquement la sonde par rapport à l'objet observé. Le système robotique est aussi capable d'interagir avec des objets dont aucune connaissance préalable sur leur forme, paramètres 3D, ni position dans l'espace 3D est disponible. Pour cela, nous avons développé une méthode efficace qui estime en-ligne les paramètres 3D intervenant dans la matrice d'interaction.

Enfin, nous présentons des résultats de simulations et d'expérimentations obtenus respectivement à partir d'objet 3D simulés, et d'une plate-forme expérimentale constitué un robot médical à 6 degrés de liberté portant une sonde qui interagit avec des objets immergés dans une bassine d'eau. Ces résultats ont validé les méthodes développées dans cette thèse et ont montré leur robustesse au bruit des images échographiques.

Mots clés : Robotique médicale, asservissement visuel, imagerie échographique, modélisation, commande sans modèle.

Abstract

This dissertation presents a new 2D ultrasound-based visual servoing method. The main goal is to automatically guide a robotized 2D ultrasound probe held by a medical robot in order to reach a desired cross-section ultrasound image of an object of interest. This method allows to control both the in-plane and out-of-plane motions of a 2D ultrasound probe. It makes direct use of the 2D ultrasound image in the visual servo scheme, where the feedback visual features are combinations of image moments. To build the servo scheme, we develop the analytical form of the interaction matrix that relates the image moments time variation to the probe velocity. That modeling is theoretically verified on simple shapes like spherical and cylindrical objects. In order to be able to automatically position the 2D ultrasound probe with respect to an observed object, we propose six relevant independent visual features to control the 6 degrees of freedom of the robotic system. Then, the system is endowed with the capability of automatically interacting with objects without any prior information about their shape, 3D parameters, nor 3D location. To do so, we develop on-line estimation methods that identify the parameters involved in the built visual servo scheme.

We conducted both simulation and experimental trials respectively on simulated volumetric objects, and on both objects and soft tissues immersed in a water-filled tank. Successful results have been obtained, which show the validity of the developed methods and their robustness to different errors and perturbations especially those inherent to the ultrasound modality.

Keywords: Medical robotics, visual servoing, 2D ultrasound imaging, kinematics modeling, model-free servoing.