TESI DI DOTTORATO

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

DIPARTIMENTO DI INGEGNERIA ELETTRICA
E DELLE TECNOLOGIE DELL'INFORMAZIONE

DOTTORATO DI RICERCA IN
INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

VISION-BASED AUTONOMOUS CONTROL IN ROBOTIC SURGERY

ROCCO MOCCIA

Il Coordinatore del Corso di Dottorato
Ch.mo Prof. Daniele RICCIO

Il Tutore
Ch.mo Prof. Bruno SICILIANO

Il co-Tutore
Ch.mo Prof. Fanny FICUCIELLO

A. A. 2020–2021

"...to all the honest Workers in every sector.
I hope that the Robots could guide you
to freedom and social justice."

Contents

List of Acronyms

MIRS	Minimally Invasive Robotic Surgery
MIS	Minimally Invasive Surgery
CBFs	Control Barrier Functions
dVRK	da Vinci Research Kit robot
VF	Virtual Fixture
LS	Laparoscopic surgery
RAMS	Robot-Assisted MicroSurgery
FRVFs	Forbidden-Region Virtual Fixtures
GVFs	Guidance Virtual Fixtures
QP	Quadratic Programming
CNN	Convolutional Neural Network
ROS	Robot Operating System
PSM	Patient Side Manipulator
ECM	Endoscope Camera Manipulator
MTM	Master Tool Manipulator
DoF	Degree of Freedom
SUJ	Setup Joint
RCM	Remote Center of Motion
DH	Denavit-Hartenberg
CRC	Colorectal cancer
EKF	Extended Kalman Filter
DQ	Dual Quaternion

List of Figures

List of Tables

Chapter 1

Introduction

Robotic Surgery has completely changed surgical procedures. Enhanced dexterity, ergonomics, motion scaling, and tremor filtering, are well-known advantages introduced with respect to classical laparoscopy. In the past decade, robots played a fundamental role in MIRS in which the da Vinci robotic system (Intuitive Surgical Inc., Sunnyvale, CA) is the most widely used system for robot-assisted laparoscopic procedures. Robots also have great potentiality in Microsurgical Robotic applications, where human limits are crucial and surgical sub-millimetric gestures could have enormous benefits with motion scaling and tremor compensation. Nevertheless, surgical robots still lack advanced assistive control features that could notably support surgeon's activity and perform surgical tasks in autonomy for a high quality of intervention. In this scenario, images are the main feedback the surgeon can use to correctly operate in the surgical site. Therefore, in view of the increasing autonomy in surgical robotics, vision-based techniques take a leading role in surgical scenarios. Moreover, a large number of surgical tasks could benefit from the application of advanced control techniques, allowing the surgeon to work under less stressful conditions and performing the surgical procedures with more accuracy and safety.

1.1 Thesis Contribution & Overview

The thesis starts from the topics of development of control strategies and enhanced visual perception, analyzing the most advanced state-of-art approaches in surgical robotics. Then, the work addresses the problem to provide surgical robots the ability to perform complex tasks helping the surgeon to skillfully

manipulate the system and accomplish the above requirements. The works presented in this thesis propose high-quality solutions to enhance specific tasks or entire surgical procedures. The thesis introduces innovative control strategies and advanced computer vision techniques to provide adequate information for control. Each approach follows the steps:

- definition of the specific surgical problem;
- description of the state-of-the-art solutions;
- definition of advanced control strategy to solve the problem;
- enhancement of the robot perception phase using computer vision techniques;
- experimental validation of the method on surgical robots.

An increase in safety and a reduction in mental workload is achieved through the introduction of active constraints, that can prevent the surgical tool from crossing a forbidden region and similarly generate constrained motion to guide the surgeon on a specific path, or to accomplish robotic autonomous tasks. This leads to the development of a vision-based method for robot-aided dissection procedure allowing the control algorithm to autonomously adapt to environmental changes during the surgical intervention through the use of stereo images of the surgical site. Computer vision is exploited to define a surgical tools collision avoidance method that uses FRVFs by rendering a repulsive force to the surgeon. Advanced control techniques based on an optimization approach are developed, allowing multiple tasks execution with task definition encoded utilizing CBFs and enhancing haptic-guided teleoperation system during suturing procedures. The proposed methods are tested on a different robotic platform involving dVRK and a new microsurgical robotic platform. Finally, the integration of new sensors and instruments in surgical robots are considered, including a multi-functional tool for dexterous tissues manipulation and different visual sensing technologies.

1.2 Thesis Structure

This Section gives a brief overview of the thesis. Chapter 1 contains the introduction of the presented thesis, underlying the robotic surgery background and the obtained achievements with the paper production based on the work developed during the Ph.D. studies. Chapter 2 presents an overview of the robotic use in surgical applications, emphasizing the benefits and safety given to MIS

and Microsurgery. Moreover, a review of the state-of-the-art techniques in advanced robot control and computer vision for robotic perception is provided in this chapter. Ultimately, the robotic platforms and the simulation environments used in this thesis are also described.

The scientific contribution is presented in the next chapters. A novel robotic solution for polyp dissection surgical procedures is proposed in Chapter 3. Colorectal polyp dissection is a delicate surgical procedure, which requires very accurate detection of the region of interest and precise cutting with adequate safety margins. The proposed method assists the surgeon during polyp dissection utilizing haptic shared control technique and basic computer vision approach for polyp identification.

A preliminary study on a vision-based method for suturing needle tracking is illustrated in Chapter 4. Suturing is a complex and tiring procedure and joint limits and singularities are common problems in this robot-assisted procedure, forcing the surgeon to continuously re-grasp the needle causing high operator cognitive load and fatigue. The chapter proposes a vision-based tracking of suturing needle to define the grasping pose that optimizes the cost of robot joint limits and singularities. Then, a haptic shared control approach is developed to guide the surgeon towards an optimal needle grasping configuration, allowing executing continuous suturing trajectories avoiding tedious interruptions.

A surgical tool collision avoidance method is described in Chapter 5. During the execution of robotic-aided surgical procedures, the tools can collide and create serious damage to human tissues. The proposed method helps the surgeon to avoid surgical tool clashing, by rendering a repulsive force. This work uses a marker-less surgical tool tracking, based on a Machine Learning approach that couples vision and robot kinematics information.

Chapter 6 introduces a framework that allows the execution of multiple surgical tasks simultaneously. A generic surgical procedure is divided into a series of sub-tasks that can be executed at the same time with different priorities. The method uses an optimization-based approach with tasks definition and execution achieved through a novel control technique, known as CBFs. The framework strongly enhances teleoperation control input integrated with a haptic feedback method on the robot master side, and it can also be adopted for autonomous surgical task execution. Moreover, this chapter also gives a brief introduction of future works. Ultimately, Chapter 7 presents an overview of the entire thesis.

1.3 Thesis Publications

The works presented in this thesis is has been published in the following articles:

- **R. Moccia** and F. Ficuciello "Multiple Tasks Execution using Control Barrier Functions in Surgical Robotics", *in submission for IEEE Transactions on Robotics*, September 2021.
- **R. Moccia**, C. Iacono, B. Siciliano, F. Ficuciello, "Vision-based Dynamic Virtual Fixtures for Tools Collision Avoidance in Robotic Surgery", *IEEE Robotics and Automation Letters and presented at IEEE International Conference on Robotics and Automation 2020*, vol. 5, no. 2, pp. 1650-1655, June 2020.
- H. Liu, M. Selvaggio, P. Ferrentino, **R. Moccia**, S. Pirozzi, U. Bracale, F. Ficuciello, "The MUSHA Hand II: A Multi-Functional Hand for Robot-Assisted Laparoscopic Surgery", *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 1, pp. 393-404, February 2020.
- D.E. Canbay, P. Ferrentino, H. Liu, **R. Moccia**, S. Pirozzi, B.Siciliano, F. Ficuciello, "Calibration of tactile/force sensors for grasping with the PRISMA Hand II", *Proc. IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Deft, Netherlands, 2021.
- **R. Moccia**, M. Selvaggio, L. Villani, B. Siciliano, F. Ficuciello, "Vision-based Virtual Fixtures Generation for Robotic-Assisted Polyp Dissection Procedures", *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Macau, China, pp. 7928-7933, November 2019.
- M. Selvaggio, A. M. Ghalamzan E., **R. Moccia**, F. Ficuciello and B. Siciliano, "Haptic-Guided Shared Control for Needle Grasping Optimization in Minimally Invasive Robotic Surgery", *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Macau, China, pp. 7734-7939, November 2019.
- C. Iacono, **R. Moccia**, B. Siciliano, F. Ficuciello, "Forbidden Region Virtual Fixtures for Surgical Tools Collision Avoidance", *Proc. Institute for Robotics and Intelligent Machine the Conference*, Rome, Italy, October 18-20, 2020.

- C. Iacono, **R. Moccia**, B. Siciliano, F. Ficuciello, "Vision-Based Dynamic Virtual Fixtures for Tools Collision Avoidance in MIRS", *10th Joint Workshop on New Technologies for Computer/Robot Assisted Surgery*, Barcelona, Spain, September 28-30, 2020.

- **R. Moccia**, M. Selvaggio, L. Villani, B. Siciliano, F. Ficuciello, "Vision-based Virtual Fixtures Generation for Robotic-Assisted Polyp Dissection Procedures", *The 12th International Workshop on Human-Friendly Robotic*, Reggio Emilia, Italy, October 24-25, 2019.

- M. Selvaggio, A. M. Ghalamzan E., **R. Moccia**, F. Ficuciello and B. Siciliano, "Haptic-guided shared control for needle grasping optimization in minimally invasive robotic surgery", *Proc. Institute for Robotics and Intelligent Machine the Conference*, Rome, Italy, October 18-20, 2019.

- **R. Moccia**, M. Selvaggio, F. Ficuciello, "Haptic-guided shared control for needle grasping optimization in minimally invasive robotic surgery", *Hamlyn Symposium Workshop: From BCI to human robot augmentation*, London, England, June 23-26, 2019.

- M. Selvaggio, A. M. Ghalamzan E., **R. Moccia**, F. Ficuciello, B. Siciliano, "Haptic-guided needle grasping in minimally invasive robotic surgery", *IEEE ICRA Workshop - Next Generation Surgery: Seamless Integration of Robotics, Machine Learning and Knowledge Representation within the Operating Rooms*, Montreal, Canada, May 20-24, 2019.

- **R. Moccia**, M. Selvaggio, B. Siciliano, A. Arezzo, F. Ficuciello, "Vision-based virtual fixtures generation for MIRS dissection tasks", *9th Joint Workshop on New Technologies for Computer/Robot Assisted Surgery*, Genova, Italy, March 21-22, 2019.

Chapter 2

State-of-art in Robotic Surgery

This chapter presents an overview of the robotic methodologies developed since their introduction in surgical applications. The goal is to provide basic information on novel surgical robotic platforms, advanced control algorithms, and vision-assisted applications. Section 2.1 introduces the use of robots in surgical procedures, underlining the benefits and safety given to MIS and Microsurgery. Section 2.2 provides an overview of the advanced control techniques developed for surgical applications empathizing the use of Active Constraints (or VFs), *i.e* an overlay that can guide the surgeon during the procedure or avoid conditions that can compromise safety. Section 2.3 contains a description of the computer-vision techniques used in the surgical scenario. Ultimately, Section 2.4 describes the robotic platforms and the simulation environments used in this thesis.

2.1 Robotics in Surgical Procedures

The first method that comes to mind thinking about surgery, is traditional open surgery, where a surgeon operates with handheld instruments through a large cut (incision). For this reason since its introduction, MIS had a significant impact on surgical procedures. This kind of method includes LS, a surgical technique in which short cannulae, containing surgical instruments, are inserted into the abdomen through small (typically less than 1 cm) incisions. LS is most commonly used in gynecology, gastroenterology, and urology, showing excellent results in difficult treatments of gastrointestinal diseases such as rectal cancer [4] and in particular in colorectal polypectomy [5]. The application of this technique offers several advantages to the patient, like decreased post-

operative pain, shorter hospitalization, reduced risk of complication, low blood loss and transfusions, and minimal scarring. Nonetheless, LS still presents important limitations, mainly due to reduced visual and haptic perceptions, that can compromise the quality of intervention and create significant challenges on this kind of surgery. Reduced depth perception induced by the use of 2D monitors can lead to weak hand-eye coordination. At the same time, one of the major issues in LS is tissue damage mainly caused by the use of excessive force involuntary applied by the surgeon.

As previously exposed, MIS encompasses all surgical procedures that reduce the size of the incision in patient's body. Among those, imaging techniques are crucial, starting from a clear view of the surgical site usually guaranteed by the use of an endoscope. Several surgical specialties require imaging capabilities that cannot be satisfied by the use of an endoscope and required optical magnification ensured by the use of an operating microscope. Starting from this assumption, microsurgery is currently used in many surgical procedures, such as general surgery, ophthalmology, orthopedic surgery, gynecological surgery, otolaryngology, neurosurgery, oral and maxillofacial surgery, plastic surgery, podiatric surgery, and pediatric surgery. Specifically, microsurgery is usually adopted to perform reconstructive procedures, that are impossible to realize with traditional surgery. In particular, microvascular anastomosis, the suturing of blood vessels and nerves whose diameter is less than 3 mm (typically 1 mm), plays a fundamental role in these types of procedures, allowing transfer of tissue from one part to another and re-attachment of severed parts [6]. In general, microsurgery operates on very small structures with miniaturized instruments, that are small enough to accurately grab and manipulate tissue. The biggest limitation in microsurgery is represented by the absence of movement scaling, limiting the microsurgical technique by the motor abilities of the human body. Moreover, physiological tremor is present in all human beings, which is normally damped holding the instrument as close to the tip as possible, but any situation demanding an increased undamped instrument will result in higher tremor amplitude. As in MIS, the use of excessive force applied by the surgeon can deform the object in the instrument's jaw, creating tissue damage.

Since the early 2000s, the use of medical robots completely changed surgery, significantly improving surgeon's technical capabilities and transcending human sensory-motor limits. Robotic systems are adopted in many surgical fields, as shown in Fig. 2.1, with high adoption on gastrenterology and urological operations [1]. The robots improve the accuracy of tissue manipu-

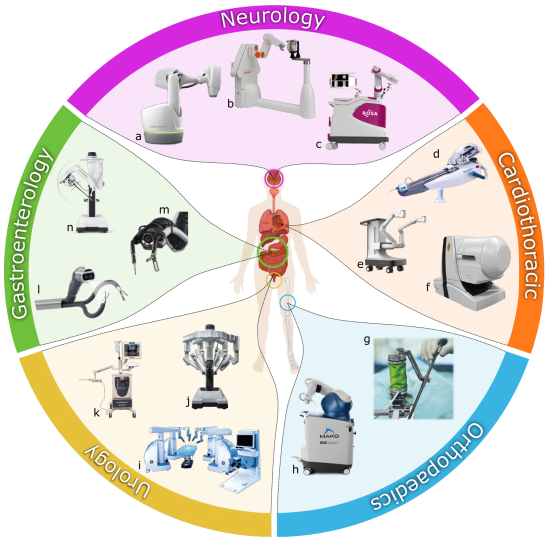


Figure 2.1: Overview of medical robots used for clinical applications, from "Autonomy in Surgical Robotics" [1]: (a) CyberKnife M6, Accuray; (b) neuromate, Renishaw; (c) ROSA ONE, Zimmer Biomet; (d) Magellan, Hansen Medical; (e) Monarch, Auris Health; (f) Niobe, Stereotaxis; (g) Renaissance, Mazor Robotics; (h) Mako, Stryker; (i) Senhance, TransEnterix; (j) da Vinci Xi, Intuitive Surgical; (k); AquaBeam, PROCEPT BioRobotics; (l); SPORT, Titan Medical; (m) Flex Robotic System, Medrobotics; and (n) da Vinci SP, Intuitive Surgical.

lation tasks and the quality of surgical techniques, thus improving results and reducing the need for revision surgery. The benefits introduced by medical robots, compared to traditional LS, include a high level of dexterity, motion scaling, elevated precision, high-definition and magnified 3D images, which allow for increased accuracy and vision inside the body.

There are different surgical robotic platforms commonly used to substitute laparoscopic instruments in many clinical procedures for MIRS. The da Vinci®robotic system is the most widely used robotic system for robot-assisted laparoscopic procedures. The da Vinci®platform on the market are: (i) the da Vinci®Xi that is the most advanced platform from Intuitive Surgical, shown in Fig. 2.2; (ii) the da Vinci®X which has the same arm architecture of da Vinci®Xi with modular components; (iii) da Vinci®SP that is a surgical platform designed for single-port access with a single-arm delivering three multi-jointed instruments. The da Vinci®robotic platform is a teleoperated robot, composed by a surgical console, a patient-side cart and a vision cart. The sur-

Figure 2.2: da Vinci Xi Surgical System.

gical console is located outside the sterile site, and is controlled by the surgeon through two master controllers and pedals. A 3D endoscope offers a view, processed by the vision cart, of the surgical site that is provided to the surgeon through a stereo visor in which the tips of the instruments are aligned with the surgeon's hands that grasp the master controllers. Ultimately, the patient-side is the operative part of the system that is physically in contact with the patient. It consists of four arms holding instruments and endoscope, in which each arm has the EndoWrist®tools with a diameter of 8 mm and about 60 cm long. The Endowrist gives freedom of movement in seven axis and rotation of $\sim 360^{\circ}$, and a vast range of robotic instruments can be used including needle holders, electrocautery forceps, cold forceps, grasping forceps and bipolar dissectors.

Microsurgical procedures imply steps that are entirely different than the corresponding steps in laparoscopy. Microvascular anastomosis requires a specific sequence of actions to correctly operate and an accurate knowledge of anatomy is required. The procedure implies the manipulation of small blood vessels, which need to be clamped in their extremities to stop the blood flow and different sizes of vessels diameters need to be considered. Robotic surgery must consider these different procedures to properly develop the motion control system of the robot since the typical movements performed by the robotic tools of da Vinci®Robot could not be replicated in microsurgery. Nowadays, the creation of RAMS platforms is still under development; however, there are novel systems suitable for teleoperated open microsurgery. A clinically available surgical robot is MUSA™ (Microsure, Eindhoven, Nederland), assisting open microsurgery. The system presents the traditional benefits of teleoperated robots as miniaturizing hand movements and reducing tremors. One of the advanced surgical robot for complex microsurgical proce-

(a) (b)

Figure 2.3: (a) Symani®Surgical System; (b) NanoWrist®robotic microinstruments.

dures is Symani®Surgical System (Medical Microinstruments (MMI) S.p.A, Italy), shown in Fig. 2.3a. Symani®is a teleoperated robotic system composed of a mobile platform consisting of an articulated structure with two robotic arms, that is positioned on the anatomical region. The system features a $7-20$X motion scaling with tremor filtration and allows manipulating a NanoWrist®robotic microinstrument, a miniaturized tool with 3 mm shaft that represents the most innovative part of the system itself (Fig. 2.3b). The master console consists of a surgical chair, two stick manipulators enabled by a foot pedal, and a 3D visualizer.

Despite the improvements, the presented robotic systems still show some limitations that have not yet been overcome. A recent study classifies surgical robots into six stages based on the level of autonomy [2] (Fig. 2.4):

- **Level 0**: no autonomy. The robot is teleoperated and responds to the surgeon who performs all tasks from monitoring to decision making.
- **Level 1**: robot assistance. The robot mechanically or virtually interacts with the surgeon to guide him/her during a task. This category includes haptic guidance and active constraints (or VFs) support;
- **Level 2**: task autonomy. The robot performs specific tasks in an autonomous mode based on the surgeon's specifications that maintains a discrete control rather than a continuous control on the system;

Figure 2.4: Levels of autonomy for Surgical Robots, from "Medical robotics-Regulatory, ethical, and legal considerations for increasing levels of autonomy" [2].

- **Level 3**: conditional autonomy. The robot can autonomously plan and execute specific task strategies the surgeon can select and approve;
- **Level 4**: high autonomy. The robot plans, executes, and makes decisions, while the surgeon is only responsible for supervising the surgical operation;
- **Level 5**: full autonomy. The robot performs the surgical procedure and no human is needed.

A vast amount of literature is available at Level 0, teleoperated platforms previously exposed in this section belong to this level, where no decision making is conferred to the machines and enhancements in the process automation are limited to tremor filtering, motion scaling, and stereoscopic vision. Generally, platforms at this level are not equipped with any system to give the surgeon the sense of touch or any force feedback. Level 1 and Level 2 represent a crucial step in technology development, including important surgical assistance while the human is still required to be constantly in control of the system, as needed in the unstructured clinical environment. On the contrary, Level 3

and Level 4 exhibit higher autonomy and limit the user intervention, which requires a highly structured environment and consequently advanced perception capabilities. For these reasons, no robotic platform has reached a level of full automation as expressed in Level 5. The work developed in this thesis tried to propose novel control strategies and computer vision algorithms to reduce the surgeon's physical-mental workload and enhance surgical operation quality, starting from Level 0 and moving toward Level 5. In the next sections, an overview of the related works in robotic surgery is given. In detail, the most advanced control algorithms developed for minimally invasive robotic surgery and microsurgery, related to the different levels of autonomy, are summarized in Sect. 2.2. In Sect. 2.3 visual perception for surgical environment characterization is reviewed. Section 2.4 presents the robotic hardware and software used to test the advanced control method developed in this thesis.

2.2 Control Algorithms in Robotic Surgery

As well-known from literature, the main characteristic of a robotic system is the ability to couple complex information detected from the environment to action in order to achieve a specific task [7]. Surgical procedures are mainly composed by very difficult tasks in a highly unstructured environment, that makes even basic process a complicated assignment. This aspect influences the choice of the level of autonomy for the development of technical innovations in surgical robotics.

The clinical site represents a good example of a safety-critical environment, pushing for the development of a telerobotic platform that prevents the user to physically interact with the environment itself and introduces an enhancement to transcend human limits. As presented in Sect. 2.1, the first robotic platforms introduced in surgery are represented by teleoperated robots. This type of device allows the development of different control architectures, related to the level of autonomy, grouped into three categories: direct control, supervisory control, shared control [8]. Direct control belongs to robotic systems at Level 0 of autonomy, which are directly controlled by the surgeon through a master interface, implying that the only implemented control algorithms regard the tremor filtering and motion scaling, which represent a great improvement compared to open surgery [9, 10]. In supervisory control, user commands and feedback occur at a higher level and the connection is more relaxed and the slave has to rely on stronger local autonomy to refine and execute tasks. The control architecture may incorporate sensory feedback to the

user in a bilateral configuration. If the slave motion is controlled by a combination of direct user commands and local sensory feedback or autonomy, the architecture is denoted as shared control [11], relying on Level 1 and Level 2 of autonomy. First robotic surgery applications of shared control techniques have been proposed to compensate for beating heart movements, overlaying sensed heart motion onto surgeon commands allowing them to operate on the stabilized patient [12].

A special application of shared control is the use of VFs, a virtual overlay that can guide the surgeon during the procedure or avoid conditions that can compromise safety. VFs can be classified into two categories: FRVFs, suitable for simulating barriers around forbidden regions, and GVFs, showing attractive behavior pulling the robot end-effector towards the desired path. The VFs use on teleoperated robots is often coupled to haptic feedback or guidance, in which the fixtures provide active assistance to the user by force rendering at the master-side. The first author to have introduced VFs was Rosenberg in his works [13] and an extensive review about it can be found in [14]. Moreover, these shared control techniques represent a suitable instrument for high-quality surgical applications [15]. Recently, many authors have considered VFs use in shared control teleoperation, and multiple works were presented to introduce active constraint in surgical robotics for task and safety accomplishments. In [16], authors explored the application of VFs for MIS procedures, where the tool is inserted in the patient through a fulcrum. Here, a key issue is to render to the surgeon forces that are virtually applied at the instrument distal tip, while the robot is physically attached to the instrument proximal handle. Selvaggio et al. propose on-line VFs generation and adaptation guiding the surgeon during procedures [17]. A large number of works used VFs implementations to solve a specific sub-task, i.e to avoid surgical tool collision, simultaneously offering assistance for high precision movements in geometrically complex environments [18, 19]. Dynamic active constraints are modeled in real-time to autonomously adapt to tissue deformation and anatomic structure in [20]. This method is particularly suitable for hyper-redundant and flexible robots that are gradually introduced in robotic surgery.

Advanced control laws developed on surgical robots increasingly take advantage of the application of constrained optimization. The creation of VFs can be achieved using QP with specific non-linear constraints definition. The authors in [21], introduced a new teleoperated system with a redundant slave robot for robotic surgery, that guides the surgeon towards target anatomy by providing force feedback based on VFs exploiting a closed-form inverse

kinematics to solve redundancy based on an optimization approach. Adorno in [22], proposed a novel unified framework for teleoperated robot control using QP programming and dynamic VFs, generated by adding linear constraints. An extension of this work provides dynamic constraints, obtained by adding a fixed residual term [23]. Despite all the proposed techniques, a complex design of the master interface makes force feedback particularly challenging. Moreover, most of these works encounter particular difficulties in VF geometrical definition caused by the complex surgical environments, limiting the interactions between the robot tool tip and the environment itself. The work based on QP problems often used non-linear constraint, without considering closed-loop stability. Recently, many works enriched Level 3 and Level 4 of autonomy, proposing innovative technological systems that will allow a single surgeon to execute MIRS without the need of an expert assistant surgeon. In particular, authors in [24] proposed a teleoperated multirobots platform, which intends to be operated by the assistant surgeon during MIRS and to cooperate with the da Vinci surgical system. In [25], the work presents a method for online trajectory generation that can compute collision-free motions for an autonomous assistive robotic manipulator.

Suturing procedures require a specific mention, representing a critical task that mostly relied on the surgeon's ability. Suturing is usually divided into multiple sub-tasks like needle grasping, approaching the skin, making stable contact, and task performance. Multiple papers were presented enhancing suturing procedure, being part of Level 3 and Level 4 of autonomy. A significant improvement compared to direct teleoperation is introduced in [26], but the method is only utilized for grasping and manipulating objects whereas they do not account for the shape of the object. In [27], Chen et al. have described the implementation of the VF on the first generation of da Vinci®robot including a user study to compare the performance of VFs assistance and freehand teleoperation in both needle passing and knot tying suturing sub-tasks. Liu et al. in [28] presented an offline optimization-based solution to needle grasping and selection of entry ports of robotic instruments. However, this method cannot be utilized for the haptic shared-control system during a suturing task due to the high computation time. In [29], the authors proposed a novel mechanical needle guide and a framework for optimizing needle size, trajectory, and control parameters using sequential convex programming. Fontanelli et al. in [30] made a comparison between multiple assistive methods for suturing in MIRS. As exposed in this section, the trend of advanced control methods starts from Level 0 of autonomy with the introduction of teleoperation robots. It continues

with the creation of shared control techniques simplifying surgeon's work and introducing autonomous sub-tasks and goes towards completely autonomous task execution in robotic surgery. The work proposed in this thesis tried to introduce advanced control methods, relying on shared control paradigms and laying the foundations for fully autonomous surgical procedures.

2.3 Vision Assisted Control in Robotic Surgery

Perception is an essential component of a robotic system. Vision plays a fundamental role to obtain geometrical and qualitative information on the environment where the robot operates, without physical interaction. A vision system implies the use of a combination of camera hardware and software algorithm computation that allows robots to process the data obtained from the environment.

The first use of camera and vision in robot control was to enhance trajectory planning in robot grasping problems, where a simple motion controller executes the planned path. In this approach, visual measurements are used in open-loop making the system sensitive to uncertainties due to poor positioning accuracy of the manipulator or the fact that the target object is moving. Successively in vision-based control, the visual measurements are used in feedback control to compute an appropriate error vector defined between the current end-effector pose and the pose of the target object. The crucial characteristic of vision-based control is represented by the controlled variables, which are not directly measured by a sensor but through complex elaborations. In particular, computer vision allows extracting useful information from camera images using image processing, employed by the control system at different levels for task planning and feedback control [7]. Moreover, advanced functions required in robotics use Machine Learning techniques, focusing on pattern recognition in acquired data.

In surgery, images are the main feedback the surgeon can use to correctly operate in the surgical site. A vast amount of works is present in literature for computer vision and image understanding techniques applied to robotic surgical images [31]. Traditional teleoperated robots were equipped with 3D visualizers, but the correct visualization of the surgical site was their only assignment. More recently, image-based approaches were introduced to develop assistive techniques. Shared control methods could notably improve their effectiveness considering a vision-based approach, that guarantees the ability to adapt the VF geometry to a changing environment. In [32, 33], a control law

for position-based VFs is shown, where the VFs are generated in real-time from microscope video. In MIRS and RAMS, a reliable 3D reconstruction of the surgical scene is strictly needed to properly develop motion control of the surgical robot. Authors in [34] created a shared control strategy that exploits VFs to maneuver the pick-up probe for optimal prostate image acquisition. In a surgical scenario, soft tissue deformation is an important aspect to be considered. 3D tracking of tissue deformation is proposed in [35], based on laparoscopic and endoscopic images. As previously exposed, Machine Learning techniques are used more often to solve complex problems in surgical robotics. In [36], a Reinforcement Learning framework is described and used to prototype and implement state-of-art Reinforcement Learning algorithms on surgical robotics problems using da Vinci®robotic system. Deformation tracking through vision information represents a relevant method also for contact force estimation, which is a crucial aspect of haptic guidance in robot-assisted surgery. Authors in [37] propose a force estimation model based on CNN, using spatiotemporal information present in video sequences and the temporal structure of tool data.

Another core topic is represented by surgical tool tracking, which is commonly executed using kinematic data from the robot and could be an effective method for a control system. Surgical tool tracking can be performed using CAD model and robot kinematics [38], Bayesian state estimation [39], or adopting Deep Learning techniques for tool detection and real-time segmentation [40]. Surgical gesture classification also becomes an important instrument to enhance the robot's capability to elaborate decisions, using endoscopic or depth images [41, 42], or exploiting recurrent neural networks trained with kinematic information [43]. Vision-based control can significantly improve the quality of the suturing procedure. In particular, some papers tried to propose a solution for the needle insertion problem, identifying the entry point [44, 45] and for the knot tying problem [46]. In [47], a vision-based method to solve for needle grasping is introduced, allowing the robot to automatically grasp the suturing needle based on visual tracking of simple markers positioned on it. Regrasping the needle also represents a crucial time-consuming process in suturing and authors in [48] present rapid trajectory generation for bi-manual needle re-grasping via Reinforcement Learning. In [49], an autonomous laparoscopic robotic suturing system is shown, using point clouds to autonomously plan the needle path. Ultimately, computer vision and Machine Learning algorithms can also exploit different types of preoperative imaging, like Magnetic Resonance Imaging (MRI), Computed Tomography (CT), and Ultrasound imaging

Figure 2.5: The da Vinci Research Kit robot (dVRK) in ICAROS Center, Università degli studi di Napoli, Federico II.

(US). In particular autonomous organs and tumors, segmentation can be performed using different images techniques including MRI and CT and optimization method or deep learning models [50, 51]. Furthermore, surgical robotics could also benefit from the introduction of new visual sensing technologies: TeraHertz, Narrow band Imaging, and Hyperspectral Imaging.

The work proposed in this thesis tried to introduce computer vision algorithms, adopting traditional and Machine Learning approaches, that represent a valid support for novel control methods in surgical robotics.

2.4 Hardware and Software

The dVRK robotic platform, placed in ICAROS Center, Università degli studi di Napoli Federico II, has been used as hardware in the work reported in this thesis (Fig. 2.5), alongside simulations, to develop and test the proposed control algorithms. The dVRK is an open-source mechatronic system, constituted by the first-generation of the da Vinci®robotic system equipped with electronics, firmware, and software developed by Johns Hopkins University LCSR and Worcester Polytechnic Institute AIM Lab, on purpose to create an open control architecture [52, 53]. The open controller provides a full ROS-based control of all arms [54], allowing position, velocity, and current control and thus opens the way for developing and testing advanced control techniques. A complete kinematic and dynamic description of the dVRK robot is provided in [55, 56]. The robotic system is composed of two PSMs and an ECM commanded by two MTMs. The MTM is an 8 DoFs manipulator where only the first 7 DoFs are considered, the 8th joint is not actuated and it is only used to command the opening and closing of the gripper. The slave manipulators (PSMs and

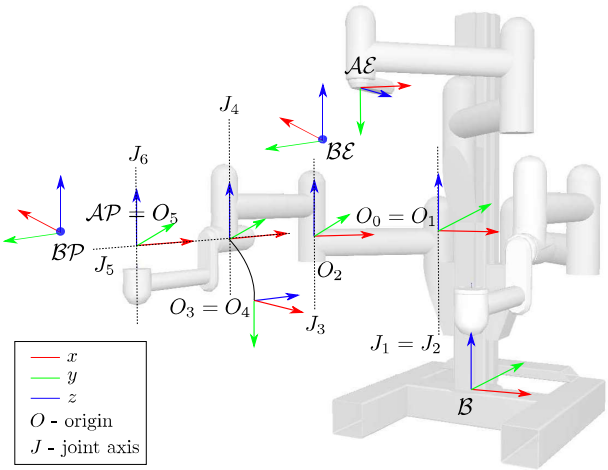


Figure 2.6: SUJ arm kinematics.

ECM) are mounted on a Setup Joint (SUJ) which is an articulated structure composed by not actuated arms but it is possible to control breaks in each joint and read the angular position using potentiometers. The SUJ-PSM arms have 6 DoFs with $\boldsymbol{q_s} = [\boldsymbol{q_{s1}}, \ldots, \boldsymbol{q_{s6}}]$ as the generalized joint coordinates and kinematic description shown in Fig. 2.6. The SUJ-PSM end-effector pose, described by the homogenous trasformation $\boldsymbol{T}^{\mathcal{B}}_{\mathcal{AP}}$ between end-effector frame $\mathcal{AP} : \{\boldsymbol{O}_{ap} - \boldsymbol{x}_{ap}, \boldsymbol{y}_{ap}, \boldsymbol{z}_{ap}\}$ and base frame $\mathcal{B} : \{\boldsymbol{O}_b - \boldsymbol{x}_b, \boldsymbol{y}_b, \boldsymbol{z}_b\}$, can be computed using the Denavit-Hartenberg (DH) convention (Table 2.1, where $a_2 = 0.58\ m$, $a_3 = 0.56\ m$ and $d_4 = 0.425\ m$) to the kinematic chain. In the same way, the SUJ-ECM end-effector pose, described by the homogenous trasformation $\boldsymbol{T}^{\mathcal{B}}_{\mathcal{AE}}$ between end-effector frame $\mathcal{AE} : \{\boldsymbol{O}_{ae} - \boldsymbol{x}_{ae}, \boldsymbol{y}_{ae}, \boldsymbol{z}_{ae}\}$ and base frame $\mathcal{B} : \{\boldsymbol{O}_b - \boldsymbol{x}_b, \boldsymbol{y}_b, \boldsymbol{z}_b\}$, can be computed considering only the first four rows of Table 2.1. Two additional constant transformation $\boldsymbol{T}^{\mathcal{AP}}_{\mathcal{BP}}$ and $\boldsymbol{T}^{\mathcal{AE}}_{\mathcal{BE}}$ are considered to compute the transformation between $\mathcal{AP}$,$\mathcal{AE}$ and $\mathcal{BP}$ and $\mathcal{BE}$ as the base frame of PSM and ECM respectively. The PSM is a 7 DoFs actuated arm moving around a fixed RCM that is a fixed point with respect to joint, corresponding to the insertion point inside patient's body during operations. The first 6 DoFs are defined with the following sequence RRPRRR, where R and P identify revolute and prismatic joints respectively, while the 7th DoF corresponds to the opening and closing of the gripper. The frame $\mathcal{BP}$ is chosen as coincident with the RCM point with $\boldsymbol{q_p} = [\boldsymbol{q_{p1}}, \ldots, \boldsymbol{q_{p6}}]$ as

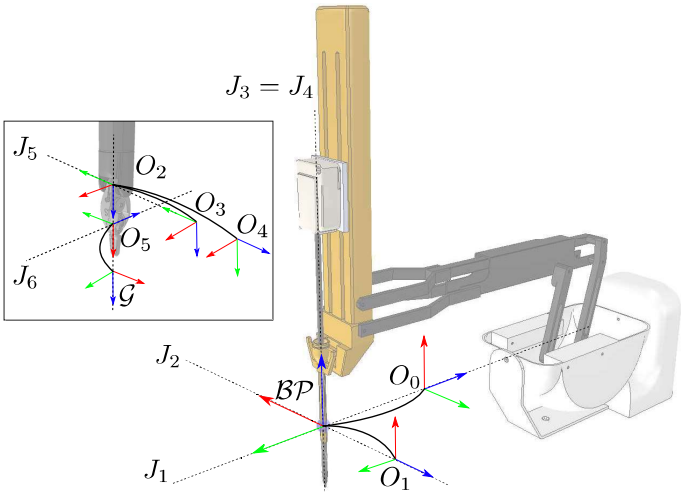


Figure 2.7: PSM arm kinematics.

the generalized joint coordinates and kinematic description shown in Fig. 2.7. The PSM end-effector pose, described by the homogenous trasformation $\boldsymbol{T}_{\mathcal{G}}^{\mathcal{BP}}$ between end-effector frame $\mathcal{G} : \{\boldsymbol{O}_g - \boldsymbol{x}_g, \boldsymbol{y}_g, \boldsymbol{z}_g\}$ positioned at the center of the gripper, and RCM frame $\mathcal{BP} : \{\boldsymbol{O}_{bp} - \boldsymbol{x}_{bp}, \boldsymbol{y}_{bp}, \boldsymbol{z}_{bp}\}$, can be computed using the DH convention (Table 2.2, where $a_5 = 0.0091m$) to the kinematic chain. The ECM is a 4 DoFs actuated arm moving about a RCM with the following sequence RRPR. The frame $\mathcal{BE}$ is chosen as coincident with the RCM point with $\boldsymbol{q_e} = [\boldsymbol{q_{e1}}, \ldots, \boldsymbol{q_{e6}}]$ as the generalized joint coordinates and kinematic description shown in Fig. 2.8. The ECM end-effector pose, described by the homogenous trasformation $\boldsymbol{T}_{\mathcal{C}}^{\mathcal{BE}}$ between end-effector camera frame $\mathcal{C} : \{\boldsymbol{O}_c - \boldsymbol{x}_c, \boldsymbol{y}_c, \boldsymbol{z}_c\}$ and RCM frame $\mathcal{BE} : \{\boldsymbol{O}_{be} - \boldsymbol{x}_{be}, \boldsymbol{y}_{be}, \boldsymbol{z}_{be}\}$, can be computed using the DH convention (Table 2.3, where $d_4 = 0.007m$) to the

Table 2.1: DH parameters of SUJ arm

link	joint	a_i	α_i	d_i	θ_i
1	P	0	0	q_{s1}	–
2	R	a_2	0	–	q_{s2}
3	R	a_3	0	–	q_{s3}
4	R	0	$-\pi/2$	–	q_{s4}
5	R	0	$\pi/2$	$-d_4$	q_{s5}
6	R	0	0	–	q_{s6}

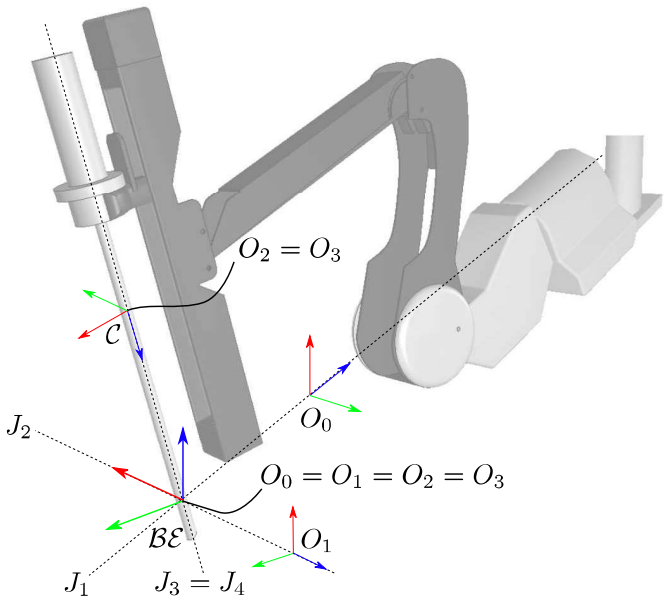


Figure 2.8: ECM arm kinematics.

kinematic chain. Moreover, joint position and velocity limits are reported in Table 2.4. The work developed in this thesis uses a dynamic model identification of MTM and PSM arms using recursive Newton-Euler approach, provided by Fontanelli et al. in [55]. The method allows computing the joint torques considering inertia, Coriolis, centrifugal, and gravity contribution, while the contributions due to joint friction and to elastic torques acting on some of the joints are added separately. In some works developed in this thesis, the MTM

Table 2.2: DH parameters of PSM arm

link	joint	a_i	α_i	d_i	θ_i
1	R	0	$-\pi/2$	–	q_{p1}
2	R	0	$-\pi/2$	–	q_{p2}
3	P	0	0	q_{p3}	–
4	R	0	$\pi/2$	–	q_{p4}
5	R	a_5	$-\pi/2$	–	q_{p5}
6	R	0	$-\pi/2$	–	q_{p6}

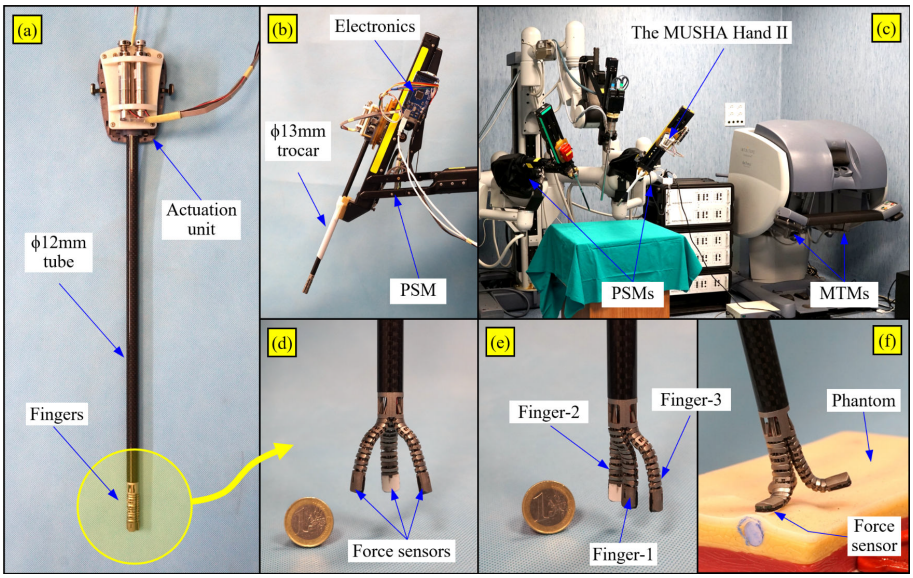


Figure 2.9: MUSHA Hand II: has ⌀12 mm cylindrical form (a); is mounted on the PSM of the dVRK system (b) (c); in grasper mode (d); in fan retractor mode (e); in palpation mode (f).

manipulator is used as an impedance controlled interface developed using the robot dynamic parameters. Finally, the integration of advanced surgical instruments on da Vinci®robotic platform [57, 58] was carried out. A side activity of the work in this thesis regards the design and development of a new surgical instrument, called the MUSHA Hand II (Fig. 2.9). The tool is a multifunctional hand with force sensors for robot-assisted LS, with three snake-like underactuated fingers equipped with a three-axis force sensor, to provide force information. The hand can be configured to either grasper mode, retractor mode, or palpation mode for different tasks. Underactuated finger design enhances the adaptivity in grasping and the compliance in interaction with the environment.

Table 2.3: DH parameters of ECM arm

link	joint	a_i	α_i	d_i	θ_i
1	R	0	$-\pi/2$	$-$	q_{e1}
2	R	0	$-\pi/2$	$-$	q_{e2}
3	P	0	0	q_{e3}	$-$
4	R	0	0	d_4	q_{e4}

The hand is compatible with the latest surgical robots, as da Vinci Surgical Systems, enhancing the functionality of the robotic surgical platform and overcoming the limits on force sensing introduced in robot-assisted LS [59, 60]. The functionality of the hand has been demonstrated through multiple experiments on dVRK as a robotic testbed.

The second robotic platform that has been used as hardware in the work reported in this thesis, is Symani®Surgical System described in Section 2.1. The patient side is composed of an articulated structure with a not actuated arm controllable through control breaks in each joint, allowing positioning of the patient side console in the surgical site. The structure is correspondent to a SCARA robot with 4 joints and the following sequence RRRP. A rigid bar is placed at its extremities equipped with two 7 DoFs manipulators arms with the first 6 DoFs has the following sequence PPPRRR and the 7th DoF corresponds to the opening and closing of the gripper. Each arm has a micro-base frame positioned on the rigid bar, chosen as coincident with the common joint centers of the first three prismatic joints and related to a common macro-base positioned at the center of the rigid bar, through a fixed transformation. Table 2.5 (where $a = 0.15$, $b = 0.00225$ and $c = 0.003$) shows the DH parameters of the arm kinematic chain, obtained considering $\boldsymbol{q_a} = [\boldsymbol{q_{a1}}, \ldots, \boldsymbol{q_{a6}}]$ as the generalized joint coordinates and adding three fixed dummy joints. Moreover, joint position limits are reported in Table 2.6.

The described robotic systems are created in a simulation environment utilizing CoppeliaSim (formerly V-REP) [61], an integrated development environment based on a distributed control architecture in which each object/model can be individually controlled via an embedded script, a plugin, a ROS or BlueZero node, a remote API client, or a custom solution. Simulation scripts can be written in C/C++, Python, Java, Lua, Matlab, or Octave. The robot model is controlled on CoppeliaSim simulation scene using ROS framework or directly in CoppeliaSim scene using embedded scripts. Each simulation scene streams topic to send the robot's joints state and spatial position of simulated

Table 2.4: Joint limits for PSM arm

	J_1	J_2	J_3	J_4	J_5	J_6
$q_{mim}[deg-m]$	-60	-45	0.05	-180	-90	-90
$q_{max}[deg-m]$	60	45	0.18	180	90	90
$\dot{q}_{mim}[rad/s-m]$	-2	-2	-0.4	-6	-5	-5
$\dot{q}_{max}[rad/s-m/s]$	2	2	0.4	6	5	5

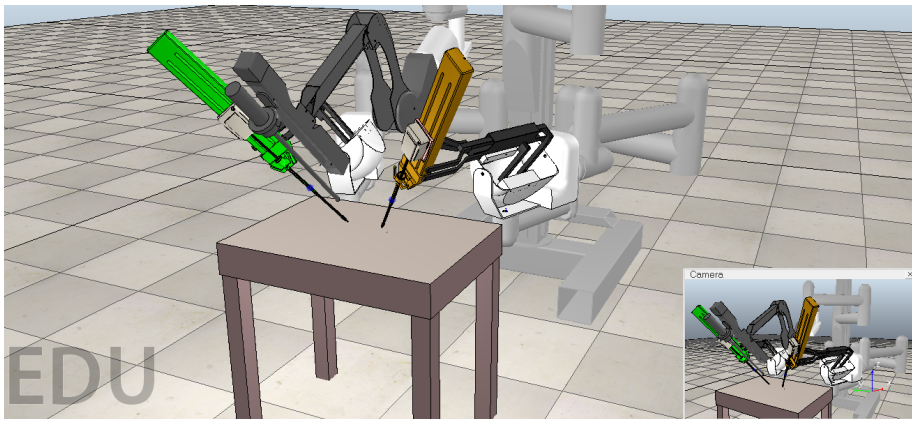


Figure 2.10: CoppeliaSim simulator scene with dVRK robot model.

objects. In particular, the dVRK simulator is presented in [56], [62] and [63], presenting the kinematic model, the control architecture and example simulation scenes. The complete dVRK simulator is open-source and available at `https://github.com/unina-icaros/dvrk-vrep.git` (Fig. 2.10). A complete simulator of the patient-side manipulators of Symani®is created, using the Universal Robot Data Format (URDF) to represent the kinematic structure and visual appearance of the robot, as shown in Fig. 2.11.

Table 2.5: DH parameters of Symani®patient-side arm

link	joint	a_i	α_i	d_i	θ_i
$1'$	Fixed	0	$\pi/2$	–	$\pi/2$
1	P	0	$\pi/2$	q_{a1}	$\pi/2$
2	P	0	$\pi/2$	q_{a2}	$\pi/2$
3	P	0	$\pi/3$	q_{a3}	–
$3'$	Fixed	0	$\pi/2$	–	$-\pi/6$
4	R	0	$-\pi/2$	a	q_{a4}
5	R	0	$\pi/2$	–	q_{a5}
$5'$	Fixed	0	$\pi/2$	b	$\pi/2$
6	R	c	0	–	$q_{a6} + \pi/2$

Table 2.6: Joint limits for of Symani®patient-side arm

	J_1	J_2	J_3	J_4	J_5	J_6
$q_{mim}[deg-m]$	-0.1	-0.1	-0.1	-180	-90	-90
$q_{max}[deg-m]$	0.1	0.1	0.1	180	90	90

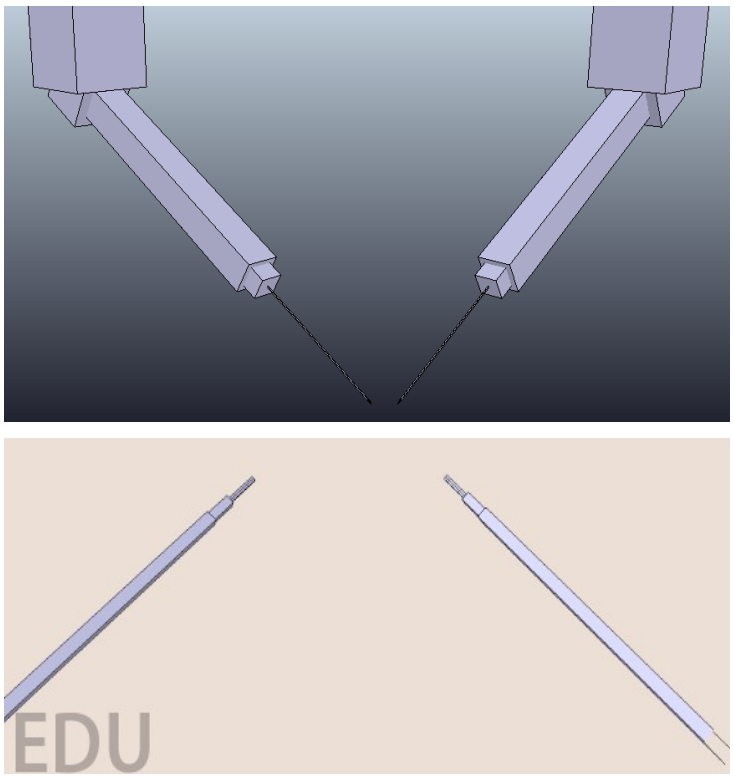


Figure 2.11: CoppeliaSim simulator scene with patient-side robot model of Symani®Surgical System.

Chapter 3

Vision-based Virtual Fixtures Generation for Polyp Dissection Procedures

This chapter presents a novel solution for polyp dissection surgical procedure that requires very accurate detection of the region of interest and high-precision cutting with adequate safety margins. In this case, the dissection procedure represents the specific surgical problem to enhance through the use of shared control approach, whose geometry is defined using basic computer vision techniques. In particular, this chapter proposes a method to constrain the robot to follow an accurate dissection path based on VFs. The VFs are created via specific control points obtained directly from images of the surgical scene and are updated by the vision algorithm. The VFs constraints can autonomously adapt themselves to environmental changes during the surgical intervention. The entire pipeline is validated through experiments on the dVRK robot.

3.1 Introduction

Nowadays, CRC is one of the major health problems. The majority of CRCs arises from adenomas or "polyp" growths on the inner surface of the colon. Endoscopic detection and removal of colorectal polyps significantly reduce the incidence and mortality of CRC, justifying the development of efficient polyp dissection procedures, which require precise movements, high dexterity, and enhanced surgeon's skills for region identification and accurate path defi-

nition [5]. In particular, sessile polyps are flat serrated growths adhered to the colon surface, difficult to be detected and removed.

As exposed in Section 2.1, the benefits and safety of MIRS in gastric surgery have been evaluated, showing the enhancement of the quality in surgical procedures [64]. In the case of robotic polypectomy, the surgical operation is performed following precise steps:

- colonoscopy for polyp detection;
- safety margins definition around the polyp;
- path planning for cutting execution.

Traditionally, the polyp detection is performed by the surgeon based only on his/her experience in the identification of specific surgical features (colors and textures), allowing the definition of the region of intervention. Moreover, the safety margins' definition is executed by the surgeon that marks selected points in telemanipulation modality by producing cautery spots around the polyp. At the conclusion, the surgeon performs a first cutting operation considering the defined margins, then he/she lifts the surface of the polyp to execute another cutting operation on the underlying tissue while keeping the focus on predefined margins. Furthermore, change in polyp's shape and rigid displacements, due to the patient's movements during the task, may compromise the correct execution of the procedure, modifying the predetermined safety margins. In this scenario, images are the main feedback the surgeon can use to correctly operate in the surgical site. Therefore, given the increasing autonomy in surgical robotics, vision-based techniques play an important role and can arise by extending computer vision algorithms to surgical scenarios [31]. Moreover, a large number of surgical tasks could benefit from the application of advanced shared-control techniques [14]. In particular, VFs are commonly recognized as a powerful method to improve surgeon's performances, increasing accuracy and precision [15] as exposed in Section 2.2.

3.1.1 Related Works

Related works can be divided into two main topics: vision algorithms for polyp detection and VFs techniques applied to surgical robotics.

- **Vision algorithms** Automatic polyp detection is a hot research topic. Bernal et al. [65] compared the performances of different polyp detection methods. Some detection methods exploit classic image processing techniques to obtain polyp boundaries. Hwang [66] used speeded up robust features (SURF)

and quantized them with K-means clustering to represent the images as a histogram of visual words. The features are then, classified with an SVM classifier. Zhu et al. [67] developed a computer-aided detection of colonic polyps based on polyp curvature estimation. Other authors exploit texture and color information as region descriptors in the image. Karkanis et al. [68] proposed an approach based on wavelet decomposition, while Hwang and Celebi [69] used watershed segmentation with initial markers selected using Gabor texture features and K-means clustering. Recent developments in deep learning, e.g., the use of CNNs have made significant advances in this field. Most of all, the differences among the methods are based on the selection of the specific network architecture or the data-set for training. Particularly, Riberio et al. [70] used CNNs for automated classification of colonic mucosa for colon polyp staging.

Remarkably, the majority of these methods are not currently used in clinic procedures. There are several reasons behind this, for example, some of them are not suitable for real-time constraints; some are built on a theoretical model of the polyp; CNNs performance is strongly related to the quality and amount of realistic images used to train the network; in conclusion, the clinical environment, with the presence of smoke, blood and specular highlights, makes the detection process a complicated task. For these reasons, more feasible and reliable methods for polyp detection need to be developed and implemented to ensure the correct execution of the polyp resection pipeline.

- **Virtual Fixtures** One of the major obstacles in vision-based VFs generation in surgical procedures is the ability to adapt the VFs geometry to a changing environment. More recently, VFs generation is supported by the introduction of vision-based techniques. Bettini et al. [71] proposed a VFs application in vitreoretinal eye surgery, using computer vision for providing a reference trajectory to the VFs control algorithm. Rydén et al. [72] showed a method for creating FRVFs protecting an object from unwanted contact using point cloud streamed by an RGB-D camera. Yamamoto et al.[73] developed an interface for teleoperated MIRS providing vision-based FRVFs and augmented visual feedback. Very few papers make a significant consideration of adaptive VFs, where the constraint geometry autonomously moves as a result of environmental changes.

3.1.2 Contribution

The work presented in this chapter proposes a vision-based method for robot-aided polyp dissection using the dVRK robot [74, 17]. In this case, the authors chose to reproduce a simple surgical scenario to create a concrete base for the development of dynamic GVFs. For these reasons, the approach exploits basic computer vision concepts, a simplified setup with a colored object, and impedance control to enforce GVFs constraints. The VFs path is adapted to the change in polyp's shape and environment displacement, that may occur during the dissection procedure, by updating new control points from the vision algorithm. The goal of the work is to define a functioning pipeline to assist the surgeon in the dissection task, by enhancing the quality of intervention. The method presented in this chapter could represent a preliminary system for the application of autonomous adapting GVFs, using a simple perception phase that can support an advanced control strategy. An experimental setup recreates patient's anatomy using phantoms and the proposed approach includes vision algorithms for detection and segmentation of the polyp and path planning of the cutting task. Then, a VF, i.e. a constraint that restricts the motion of the robot's tip along the path through a haptic guidance force rendered to the surgeon, is generated. The proposed pipeline is articulated as follows:

1. pre-operative calibration;
2. stereo endoscopic images acquisition, with the dVRK endoscope in fixed position;
3. detection and segmentation of the polyp, allowing the computation of control points, defining the polyp contours adjusted with a safety margin;
4. point 3D reconstruction and path planning;
5. VF generation on the pre-planned path, by rendering a force to the user when the the robot exceeds the path.

3.2 System Description

Figure 3.1 shows the overview of the system, composed by the dVRK robot and an experimental setup intended to replicate a surgical scene, adopting a plastic phantom to reproduce patient's tissue and a blue object representing the polyp.

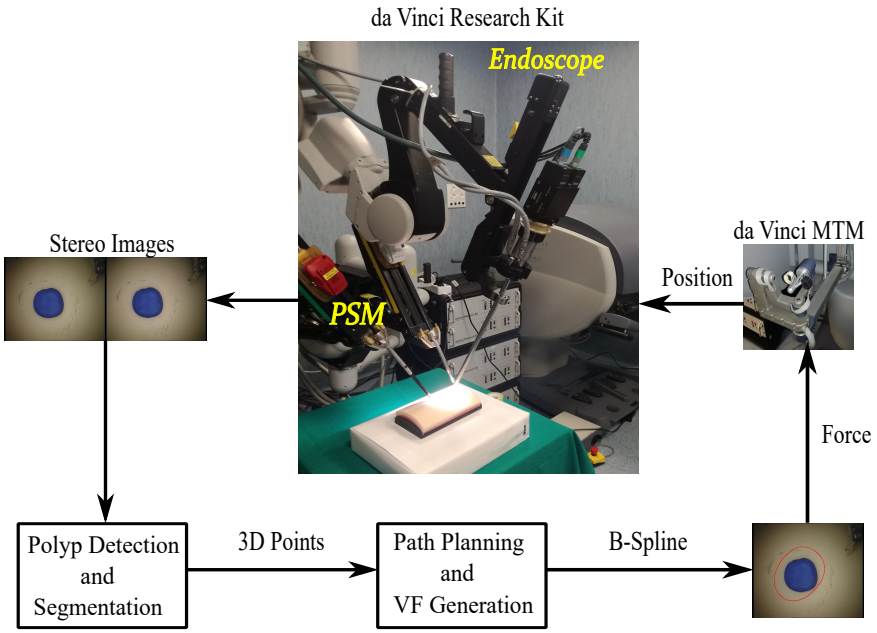


Figure 3.1: Overview of the system. The approach starts from stereo endoscopic images processing, including detection and segmentation of the region of interest, and leads to the definition of accurate points needed in path planning and VF generation, executed through haptic guidance forces rendered to the user.

3.2.1 dVRK Robot

The dVRK robot is used in teleoperation mode, with the two PSMs commanded by two MTMs, using the open controller developed by [52] and described in Section 2.4. One of the MTMs is controlled through an impedance controller, which requires the measurement of external forces to the user. Considering an n DoFs manipulator and a task space vector $\boldsymbol{x} \in \mathbb{R}^r$ with $r \leq n$, the impedance dynamics is achieved through control:

$$M\ddot{\tilde{\boldsymbol{x}}} + D\dot{\tilde{\boldsymbol{x}}} = f_h + f_{VF}(\cdot), \tag{3.1}$$

where $\tilde{\boldsymbol{x}} = \boldsymbol{x}_d - \boldsymbol{x}$, with $\boldsymbol{x}_d$ as the desired value of the robot task space variable, $M \in \mathbb{R}^{r\times r}$ and $D \in \mathbb{R}^{r\times r}$ are the inertia and damping matrices, designed to be fixed, diagonal and positive definite, $f_h \in \mathbb{R}^r$ is the external forces applied by the user and $f_{VF}(\cdot)$ is the additional force due to the VF. This dynamics is

obtained by setting the torque control input $\boldsymbol{\tau} \in \mathbb{R}^n$ of the master robot as

$$\boldsymbol{\tau} = B(\boldsymbol{q})\boldsymbol{v} + N(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{J}^T(\boldsymbol{q}) f_h, \tag{3.2}$$

$$\boldsymbol{v} = \boldsymbol{J}_A^{-1}(\boldsymbol{q}) M^{-1} (M\ddot{\boldsymbol{x}}_d + D\dot{\tilde{\boldsymbol{x}}} - M\dot{\boldsymbol{J}}_A(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - f_{h,A}), \tag{3.3}$$

where $B(\boldsymbol{q}) \in \mathbb{R}^{r \times n}$ is the joint space inertia matrix, $\boldsymbol{J}(\boldsymbol{q})$, $\boldsymbol{J}_A(\boldsymbol{q}) \in \mathbb{R}^{r \times n}$ are the geometric and the analytic Jacobians respectively, and

$$N(\boldsymbol{q}, \dot{\boldsymbol{q}}) = C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + g(\boldsymbol{q}) + h(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{3.4}$$

are terms for Coriolis and centrifugal contributions $C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$, gravity $g(\boldsymbol{q})$, friction and disturbance torques $h(\boldsymbol{q}, \dot{\boldsymbol{q}})$ [7, 17]. The force estimation is performed by resorting to a nonlinear observer [17, 75, 76]. The system includes an ECM, consisting of a stereo camera with a 5 mm baseline. The dVRK dynamic model was computed and identified in [55].

3.2.2 Calibration and Reference Frame Definition

The proposed method requires pre-operative calibration. Figure 3.2 shows the reference frames used in the Chapter. The generated VF is expressed in this inertial reference frame $\mathcal{F}_p : (\boldsymbol{O}_p - \boldsymbol{x}_p \boldsymbol{y}_b \boldsymbol{z}_p)$, with the origin positioned in the PSM's RCM. The origin of the reference frame $\mathcal{F}_t : (\boldsymbol{O}_t - \boldsymbol{x}_t \boldsymbol{y}_t \boldsymbol{z}_t)$ is positioned in the PSM tool tip. The current position of the tool tip in Cartesian space, computed through the direct kinematics of the dVRK, provides the coordinates of point $\boldsymbol{O}_t$ in the frame $\mathcal{F}_p$. A Zhang's stereo camera calibration is performed [77] to estimate the transformation between the two endoscopic cameras and to define the camera reference frame $\mathcal{F}_c : (\boldsymbol{O}_c - \boldsymbol{x}_c \boldsymbol{y}_c \boldsymbol{z}_c)$. Then, by positioning the tool tip in ten different points, the transformation $\boldsymbol{T}_c^b$ between $\mathcal{F}_p$ and $\mathcal{F}_c$ is computed adopting an absolute orientation formulation [78]. It is important to underline that this kind of calibration remains an open problem in robotics, and this solution ensures an error lower than 1 cm.

3.2.3 Vision Algorithm: Segmentation and 3D Reconstruction

The system takes stereo endoscopic images as input and a pre-processing step is performed. A watershed transformation is applied on the left gray-scale image. It operates on the image as a topographic map, with the brightness of each point representing its height, and finds the lines that run along the tops of ridges. This method allows defining the object region in the image, which is later used as a seed point defining a bounding box around it. Relying on the

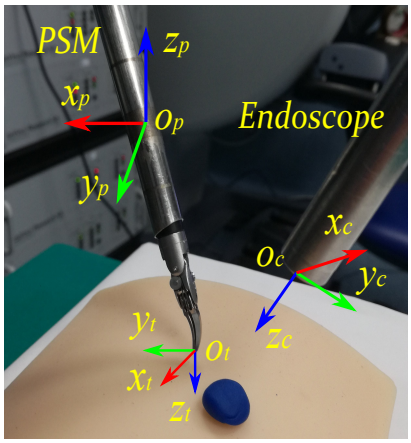


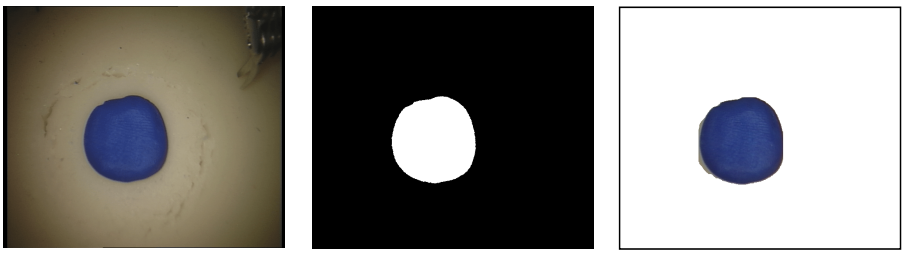
Figure 3.2: Reference frames definition. $\mathcal{F}_c : (\boldsymbol{O}_c - \boldsymbol{x}_c\boldsymbol{y}_c\boldsymbol{z}_c) =$ endoscope reference frame; $\mathcal{F}_p : (\boldsymbol{O}_p - \boldsymbol{x}_p\boldsymbol{y}_b\boldsymbol{z}_p) =$ inertial reference frame; $\mathcal{F}_t : (\boldsymbol{O}_t - \boldsymbol{x}_t\boldsymbol{y}_t\boldsymbol{z}_t) =$ tool reference frame.

defined region, the widespread *GrabCut* segmentation method [79] is applied. This technique is based on graph cuts, addressing the visual segmentation task as an energy minimization problem, based on foreground (polyp) and background models. Giving the input image I, $\alpha = (\alpha)_{i=1}^{P}$ is the set of unknown binary label of the pixels ($\alpha_i = 0$ for the background pixels, $\alpha_i = 1$ for foreground), with P as the number of pixel. The algorithm estimates the values α minimizing the energy function:

$$E(\alpha) = E_{data}(\alpha) + \gamma E_{smooth}(\alpha) \tag{3.5}$$

where $E_{data}(\alpha) = \sum_i U_i(\alpha_i)$ and $U_i(\alpha_i)$ defines the probability for a pixel to belong to the foreground or background [80]. As in [80], a modification of this algorithm is adopted, solving the minimization problem by a graph cuts minimization algorithm and defining the statistical models for the data energy function as a Gaussian Mixture Model based on color distribution.

The background and foreground layers are defined by a bounding box around the seed point created by the watershed transformation around the object, allowing isolating the object inside the image (Fig. 3.3). At this point, the homographic transformation H between the original left and right images is computed, using Scale-Invariant Feature Transform (SIFT) for features detection [81] and Fast Library for Approximate Nearest Neighbors (FLANN) for matching [82]. The left segmented image allows detecting contour points



(a) (b) (c)

Figure 3.3: Segmentation method: (a) Original frame; (b) Binary mask; (c) Segmented Image.

of the object on the left image plane, computing the Hull convex approximation. The corresponding contour points on the right image plane are identified applying the previously computed H transformation. In conclusion, the 3D position of the contour's points is reconstructed using a triangulation method with direct linear transform, expressed in the camera frame $\mathcal{F}_c$.

3.2.4 Path Planning and VF Generation

The contour points are mapped in the inertial reference frame $\mathcal{F}_p$ through the transformation $\boldsymbol{T}_c^b$. In particular, the points' coordinates are adjusted with a safety margin that allows performing the cutting in safe conditions. Once these 3D points are determined, they are used to build the VF geometry. As in [17], the path for cutting is formulated through a parametric curve. In this case, a closed B-Spline curve is adopted, defined in its 1-dimensional form by:

$$\Gamma(\boldsymbol{s}) = \sum_{i=0}^{n} N_{i,k}\left(\boldsymbol{s}\right) \boldsymbol{p}_i, \tag{3.6}$$

where $\Gamma(\boldsymbol{s})$ denotes the curve, k its order, $\boldsymbol{s} \in [0, 1]$ is the normalized curve parameter and $N_{i,k}$ are its basis function. The 3D points identified by our vision algorithm are used as controls points of the curve $(\boldsymbol{p}_i)$. Then, the VF is defined as the surface created by sweeping the polyp contours, determined by the B-Spline curve, along the axis that is perpendicular to the tissue's plane, which is set as coincident to the axis perpendicular to the camera. A simple constraint enforcement method is selected, consisting in the application of a spring-damper like force imposed on the path:

$$f_{VF} = K_p(\boldsymbol{x}_d - \boldsymbol{x}) - K_d\dot{\boldsymbol{x}}, \tag{3.7}$$

where K_p and K_d are properly designed diagonal and positive definite matrices and $\boldsymbol{x}_d$ is the set point belonging to the constraint geometry having minimum distance from the current position $\boldsymbol{x}$.

For the defined B-Spline, the Newton-Raphson (NR) method is used to find the nearest point on the curve $\boldsymbol{x}_d$ starting from the current robot position. This represents a general method for iteratively finding the extrema of a given function. As explained in [17], the desired point on the curve is the minimum distance point $\boldsymbol{x}_d = \Gamma(\bar{\boldsymbol{s}})$, obtained by finding the correspondent spline parameter $\bar{\boldsymbol{s}}$. The customary NR update law is:

$$\boldsymbol{s}_{k+1} = \boldsymbol{s}_k + \frac{\delta(\boldsymbol{x}, \boldsymbol{s}_k)}{\delta'(\boldsymbol{x}, \boldsymbol{s}_k)}, \tag{3.8}$$

where $\delta(\boldsymbol{x}, \boldsymbol{s}) : \mathbb{R}^r \times \mathbb{R} \to \mathbb{R}$ is the distance function the point $\boldsymbol{x}$ and the spline $\Gamma(\boldsymbol{s})$, defined by:

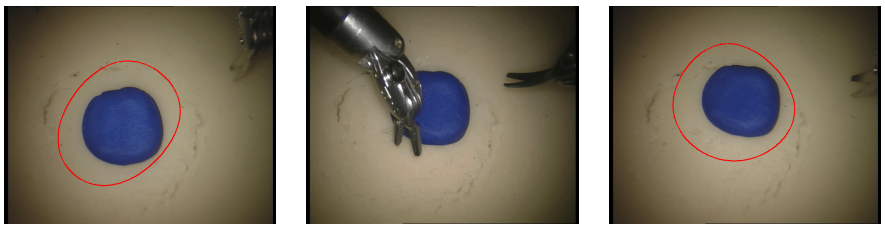
$$\delta(\boldsymbol{x}, \boldsymbol{s}) = \sqrt{(\boldsymbol{x} - \Gamma(\boldsymbol{s}))^T (\boldsymbol{x} - \Gamma(\boldsymbol{s}))} \tag{3.9}$$

$\delta'(\boldsymbol{x}, \boldsymbol{s}_k)$ is the derivative at $\boldsymbol{s}_k$ of $\delta(\boldsymbol{x}, \boldsymbol{s})$ with respect to the curve parameter $\boldsymbol{s}$. Finally, the VF is mapped to the MTM robot, considering the z-axis value defined by the tissue's normal and displaying the attractive force f through impedance control of the MTM, realized thanks to the dVRK dynamic model with the parameters identified in [55].

3.3 Experimental Validation

The proposed vision-based assistive control is evaluated by executing multiple dissection tasks. In all the sessions, a blue object with 1 cm diameter is considered to reproduce the polyp attached to a silicon rubber phantom commonly used by the surgeons for training. The safety margin is set at 1 cm from the polyp's edge, as usual in surgical dissection operations.

Three different non-expert users perform the dissection procedure by commanding the PSM of the dVRK through the MTM manipulator and following the cutting path defined by the vision algorithm. The dissection task is repeated twice, in the first session the VF is activated while in the second session it is deactivated. In the first session, whenever the user exceeds the defined path, the attractive force pulls him/her towards the planned path. During the second session, without the VF generation, the cutting path is simply defined



(a) (b) (c)

Figure 3.4: (a) VF path (red line); (b) The polyp's shape and position during the task changes due to dVRK tool interaction; (c) VF path (red line) adapted after the change in polyp position.

by a circle, with 1.5 cm diameter, centered on the polyp. The user can perform the dissection task, trying to follow the imposed circle, projected on the endoscopic image, without any haptic constraint.

To evaluate the accuracy and precision in the dissection task, the mean absolute error is computed between the current position of the PSM and the desired position defined as the minimum distance point on the B-Spline, as defined in Section 3.2.4. To evaluate the performance of the 3D reconstruction, the current position of the PSM tool's tip is recorded through the direct kinematics of the dVRK and the vision algorithm proposed, positioning the tip in 50 different workspace positions. Then, a comparison is made between the means of the z-axis values through a statistical unpaired t-test with a significance level $\alpha = 0.05$.

The B-Splines projected on the left camera image are shown in Fig. 3.4a. It is possible to notice that the vision algorithm adapts the B-Spline, at a frequency of 25 Hz, after a change in polyp position and shape, that occurred after the interaction with the dVRK tool during the task (Fig. 3.4b). This allows keeping the VF constraint on the specific path during the entire dissection procedure, autonomously moving as a result of environmental changes (Fig. 3.4c).

3.3.1 Results

Figure 3.5 shows the PSM position and B-Spline during a dissection task in the xy plane of the PSM reference frame; Figure 3.6 shows the PSM position and the circle during a dissection task in free motion. The dissection tasks with VF generation presents a mean absolute error equal to 2.1 mm along

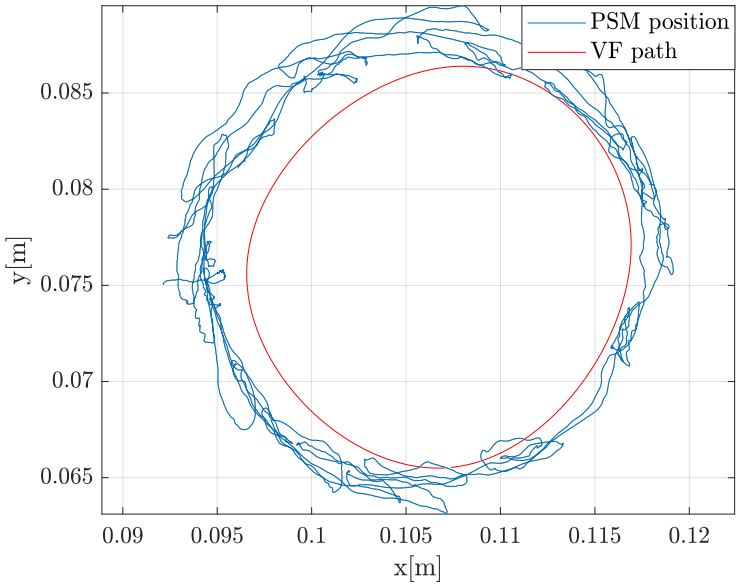


Figure 3.5: VF path (red line) and PSM position during the dissection task.

x-axis and 1.7 mm along y-axis. Conversely, the dissection tasks without VF generation has a mean absolute error equal to 13 mm along x-axis and 31.6 mm along y-axis. The results of the t-test for the 3D reconstruction evaluation proved to be statistically significant with a probability level of $p = 0.9831$ ($\alpha = 0.05$). Figure 3.8a shows the PSM position and the desired position (minimum distance point on the B-Spline) along x-axis. The related estimated force norm along x-axis is represented in Figure 3.8b. Moreover, Figure 3.8c reports the corresponding PSM position and desired position along y-axis, and, finally, Figure 3.8d shows the estimated force norm along y-axis. Figure 3.7 contains the estimated haptic guidance forces rendered to the user through the master side (MTM) during the tasks. Figure 3.8 and 3.7 are referred to the same dissection task.

3.3.2 Discussion

As it is possible to notice in Fig. 3.5, the user follows the determined path during the procedure, while in Fig. 3.6 a more irregular path is performed by the user in free motion without the VF assistance. The mean absolute error

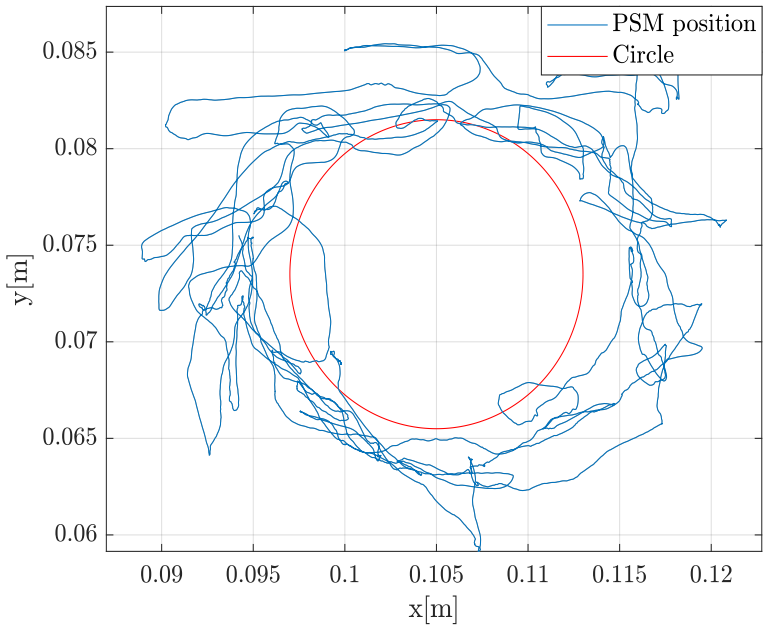


Figure 3.6: PSM position and the visual guidance circular line (red line).

values present a significant reduction ($\sim$ 10 mm along x-axis and $\sim$ 30 mm along y-axis) in the VF-based dissection task. This suggests that the introduction of the VF improves the accuracy of the procedure, helping the user performing a more precise cutting path. It may be noted that in Fig. 3.7 the maximum reached force is about 4 N, allowing the surgeon to perform the task accurately, without experiencing excessive force. The results of the 3D reconstruction t-test show the absence of statistically significant differences between z-axis values computed through the vision algorithm and the robot's kinematics. In Fig. 3.8, the peaks in force norms occur when the PSM position distances the desired position. Thus, the user continuously experiences guidance forces when the robot position is not on the VF path.

3.4 Conclusion and Future Works

The work presented in this chapter introduces an experimentally validated vision-based method for VF generation in MIRS polyp dissection tasks. The

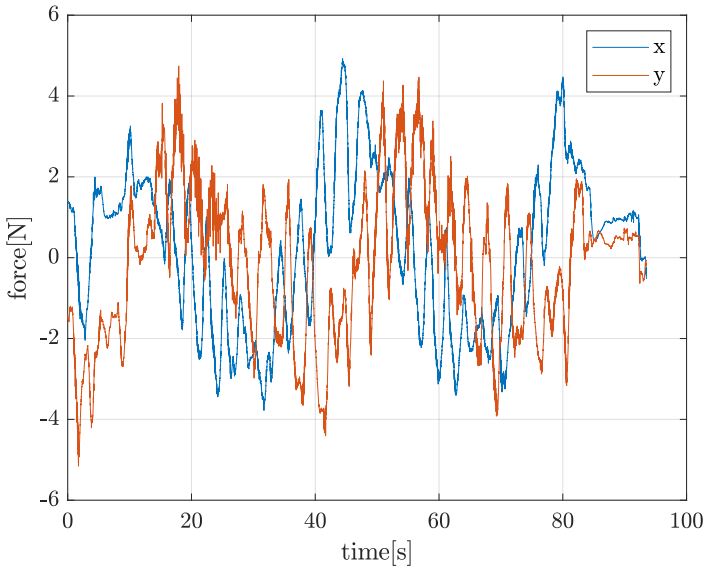
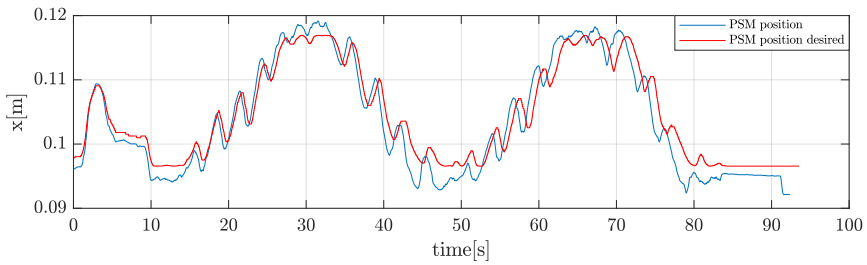
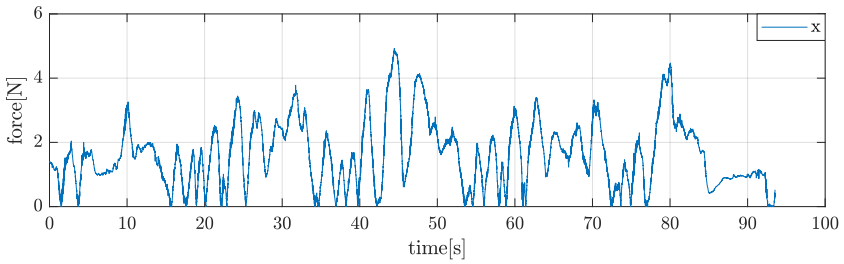


Figure 3.7: Estimated Haptic guidance forces displayed to the user through the MTM.

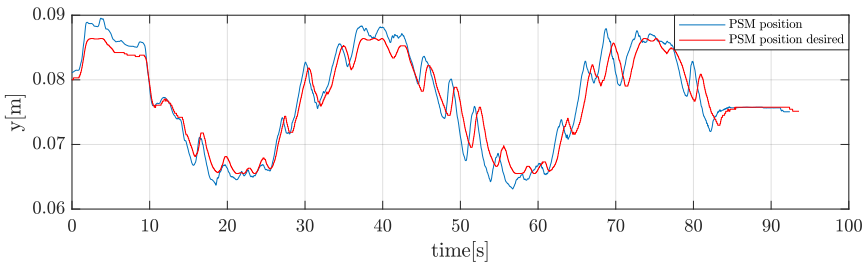
vision algorithm allows creating a constraint path for cutting, through VF generation. The VF path is updated by the vision algorithm, allowing considering environment displacement during the dissection task. The proposed strategies are evaluated through dissection experiments on dVRK performed by non-expert users, showing good results in improving the accuracy and precision of intervention and thus suggesting the feasibility of the proposed pipeline. The goal of future works is to consider more advanced computer vision techniques, allowing the proposed method to be extended to realistic surgical scenarios with regards to polyp detection, to integrate tissue's deformation, and to enforce the method also in presence of occlusions. Also, an accurate study on medical procedures will be considered for a correct definition of safety margins for cutting. In conclusion, a validation involving surgeons could be conducted to prove the effectiveness of the pipeline.



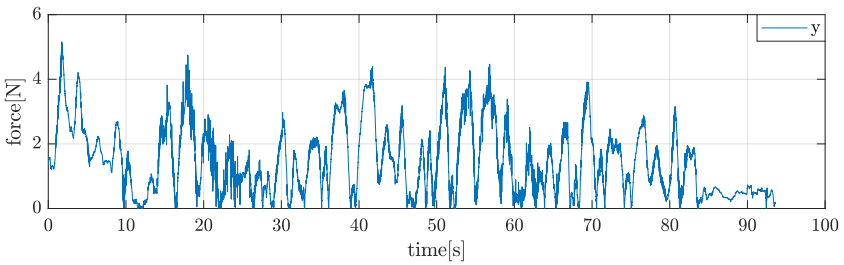
(a)



(b)



(c)



(d)

Figure 3.8: Dissection experiment. Duration: 100 seconds. Time histories of: (a) PSM position and the desired position (minimum distance point on the B-Spline) along the x-axis; (b) The related estimated force norm along x-axis; (c) PSM position and the desired position (minimum distance point on the B-Spline) along the y-axis; (d) The related estimated force norm along y-axis.

Chapter 4

Vision-based Virtual Fixtures for Suturing Needle Grasping Optimization

This chapter addresses a frequent problem in surgery, suturing. Here, a vision-based method for suturing needle tracking is proposed, defining the needle grasping pose and avoiding encountering kinematic constraints on the dVRK. The central purpose is to define a suitable tracking method for needle pose estimation, allowing defining a system to guide the surgeon towards needle grasping configurations that minimize the possibility of encountering joint limits and singularities along with a suturing task.

4.1 Introduction

In MIRS procedures, suturing represents a very difficult procedure, requiring the surgeon to continuously change the orientation of the needle to find the appropriate pose required to correctly perform the suturing task [83]. Difficult conditions are also imposed by the robot itself, for example, joint limits and singularities could increase the surgeon's cognitive workload and causes degeneration in performances. As previously exposed, haptic shared control techniques represent a powerful method to improve surgeon performances in suturing tasks. In particular, a grasping optimization approach using VFs constraints could bring a significant enhancement to the quality of intervention. A crucial step for the application of such methods is represented by the ac-

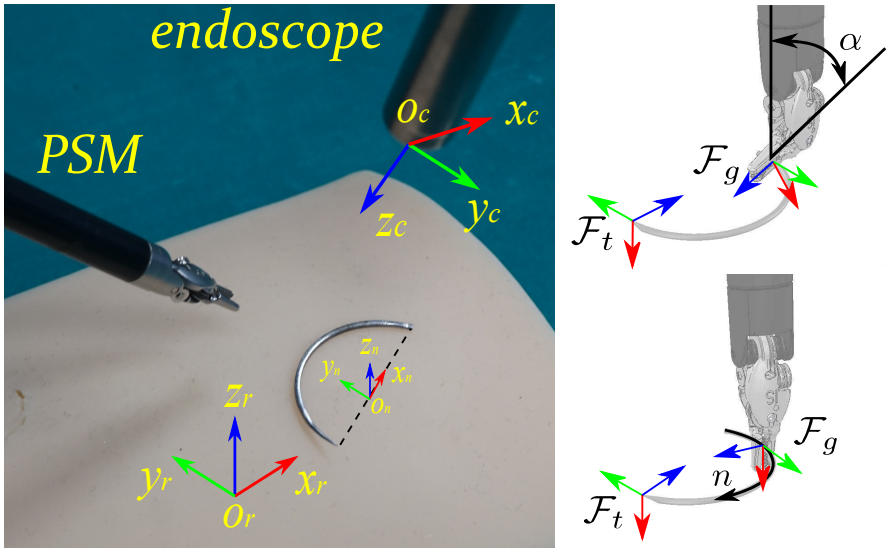


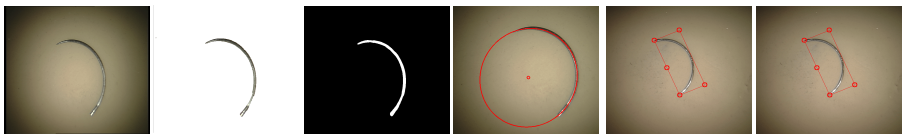
Figure 4.1: Experimental setup with reference frames and grasp parametrization of the needle.

curate detection and tracking of the needle. Vision-based techniques exploit color-based segmentation using markers on the needle, generally not suitable for realistic applications. In [84], the 3D pose of the needle is computed using geometric information, but relying on multiple observations of the needle.

The work presented in this chapter proposes a vision-based tracking of suturing needle, exploiting basic tracking-by-detection techniques, defining the chosen grasping pose for immediate starting of the suturing procedure. The method is tested using dVRK robot in a haptic shared control application, guiding the surgeon during reach-to-grasp the needle in suturing task, optimizing the cost of robot joint limits and task-oriented manipulability.

4.2 Methods

The experimental setup is similar to the one exposed in Section 3.2.2 (Fig. 4.1), composed by the PSM arm of the dVRK robot whose inertial reference frame is represented by $\mathcal{F}_r : (\boldsymbol{O}_r - \boldsymbol{x}_r\boldsymbol{y}_r\boldsymbol{z}_r)$. Zhang stereo camera calibration is performed to estimate the transformation between the two endoscopic cameras and to define the camera reference frame $\mathcal{F}_c : (\boldsymbol{O}_c - \boldsymbol{x}_c\boldsymbol{y}_c\boldsymbol{z}_c)$. While, $\mathcal{F}_n : (\boldsymbol{O}_n - \boldsymbol{x}_n\boldsymbol{y}_n\boldsymbol{z}_n)$ is the needle center frame, $\mathcal{F}_t : (\boldsymbol{O}_t - \boldsymbol{x}_t\boldsymbol{y}_t\boldsymbol{z}_t)$ is the frame



(a) (b) (c) (d) (e) (f)

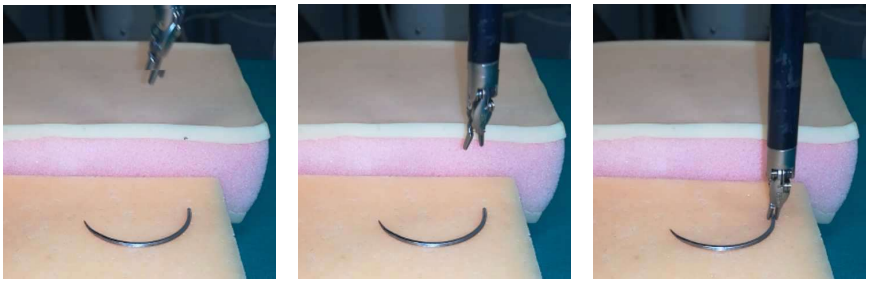
Figure 4.2: System pipeline: (a) Original frame; (b) Segmented Image; (c) Binary mask; (d) Ellipse fitting; (e) Minimum fitting rectangle; (f) Reference frame generation.

attached to the needle tip and $\mathcal{F}_g : (\boldsymbol{O}_g - \boldsymbol{x}_g \boldsymbol{y}_g \boldsymbol{z}_g)$ represents a frame attached to the robot end-effector corresponding with a generic desired grasping pose. The tracking algorithm allows defining the transformation $\boldsymbol{T}_n^c$, mapping $\mathcal{F}_n$ in $\mathcal{F}_c$.

The system takes stereo endoscopic images, as input and the use of robot kinematics allows restricting the image to a region of interest in which the needle is present, thus speeding up computations, and increasing robustness for visual occlusions. A pre-processing step is performed, exploiting edges and color and then, the widespread *GrabCut* segmentation method is used, defining a binary mask of the needle. This allows computing the minimum rectangle area that contains the needle and its ellipse-shaped projection, using the least-square fitting method. This leads to the definition of the central point of the needle $\boldsymbol{O}_n$ as the center of the fitted ellipse. In conclusion, the needle pose is estimated, given the 3D coordinates of five specific points on it and their correspondent image projection coordinates on the minimum rectangle, solving the Perspective-n-Points (PnP) problem with direct linear transform and RANdom SAmple Consensus (RANSAC) methods. Figure 4.2f shows the entire system pipeline.

Once the reference frame is defined, a parametrization of the needle grasping $\boldsymbol{z} = [\boldsymbol{n}, \boldsymbol{\alpha}]$ is adopted, identifying the curvilinear abscissa $\boldsymbol{n}$ and the angle $\boldsymbol{\alpha}$ around tangent, as shown in Fig. 4.1. The system computes the linear and angular velocities of the PSM arm in $\mathcal{F}_r$ and the joint coordinate vector $\boldsymbol{q}$, combining the differential forward kinematics of the PSM. These parameters are used to optimize the grasping pose, defining a cost function according to the joint limits. With $\boldsymbol{s}$ as a parameter of the trajectory of the needle tip in $\mathcal{F}_t$, the function is expressed as:

$$\mathcal{H}(\boldsymbol{z}) = \int_0^{\boldsymbol{s}^*} h(\hat{\boldsymbol{q}}_g(\boldsymbol{s}, \boldsymbol{z}))d\boldsymbol{s}. \tag{4.1}$$



(a) (b) (c)

Figure 4.3: Grasping a needle using PSM: (a) Initial pose; (b) Approaching; (c) Grasping.

The optimal grasping pose is obtained by finding the parameter vector $\boldsymbol{z}$ that minimizes the cost function $\mathcal{H}(\boldsymbol{z})$, solving through the gradient descent iterative method. The Cartesian pose for the PSM is calculated from the optimal grasping parameter $\boldsymbol{z}^*$ given the needle kinematics and its global pose. Finally, the corresponding desired MTM pose is determined from the optimal desired pose of the PSM, and a haptic cue is displayed on the MTM through impedance control, guiding the user toward optimal grasping configuration.

4.3 Results

The experiments consist in suturing tasks using the dVRK robot. To evaluate the accuracy of the tracking method, the 3D coordinates of the first corner of a chessboard, positioned on the origin of $\mathcal{F}_n$ from the vision algorithm, are calculated. A mean absolute error of 1.4 mm is obtained between the corner and origin coordinates. During the experiments the dVRK is set in teleoperation mode and a single PSM arm is commanded by one MTM, with impedance control implemented thanks to the dVRK dynamic model identified in [55]. The obtained grasping pose is used to generate force cues and inform the user during the reach-to-grasp phase. The surgeon feels haptic cues (Fig. 4.4), during the experiment shown in Fig. 4.3. The force cues decrease by the closeness to the optimal grasping pose and correspondingly, post-grasp movements during the suturing task execution are free from joint limits and singularities.

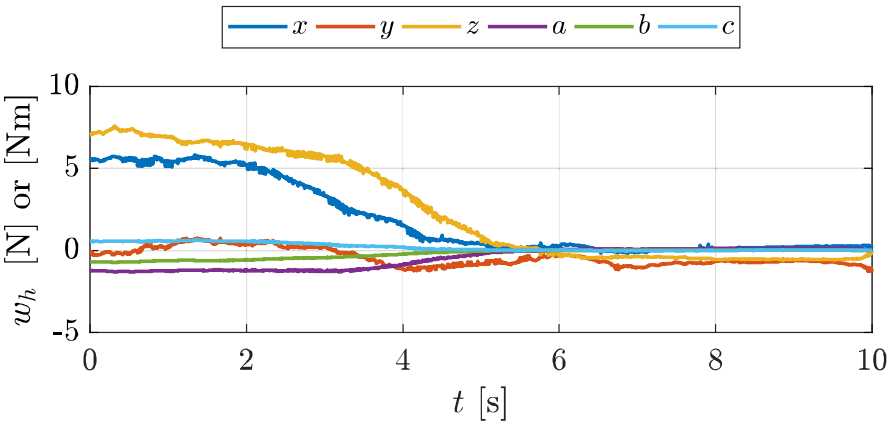


Figure 4.4: Haptic guidance force felt by the user during experiment.

4.4 Conclusion

The work presented in this chapter proposes a vision-based tracking of suturing needle. The method finds application in haptic shared control technique using dVRK, minimizing the possibility of encountering joint limits and singularities during the suturing task. The optimal grasp pose is used to compute force cues that guide the user's hand via a MTM. The mean absolute error suggests the feasibility of the tracking method, while the effectiveness of the shared control for needle grasping is illustrated using experiments performed on dVRK robot.

The goal for future works is to consider kinematic information from the robot and fusing it with visual information using Kalman or Particle filter as in [85], exploiting the proposed vision-based tracking. Moreover, quantifying performance improvements via a proper human subject test is another future extension of this work.

Chapter 5

Vision-based Dynamic Virtual Fixtures for Tools Collision Avoidance

This chapter addresses the problem of surgical tool clashing during procedures. In robot-aided surgery, during the execution of typical bimanual procedures such as dissection, surgical tools can collide and create serious damage to the robot or tissues. The procedures performed by da Vinci-like surgical robots are teleoperated, potential collisions between surgical tools are a very sensitive issue declared by surgeons. Shared control techniques based on VF can be an effective way to help the surgeon prevent tools collision.

The work in this chapter presents a surgical tools collision avoidance method that uses FRVFs. Tool clashing is avoided by rendering a repulsive force to the surgeon. To ensure the correct definition of the VF, a marker-less tool tracking method, using deep neural network architecture for tool segmentation, is adopted. The use of direct kinematics for tools collision avoidance is affected by tools position error introduced by robot component elasticity during tools interact with the environment. On the other hand, kinematics information can help in case of occlusions of the camera. Therefore, this work proposes an EKF for pose estimation that ensures a more robust application of VF on the tool, coupling vision and kinematics information. The entire pipeline is tested in different tasks using the dVRK system.

5.1 Introduction

During the execution of a surgical procedure, two or more tools can come dangerously close to each other. The surgeon has a very limited vision on the surgical site which reduces dexterity and increases the cognitive workload, making the task most difficult to be performed. The view the surgeon has could be insufficient to avoid the collision, thus this issue can cause tools or tissues damage. Experienced surgeons develop strong capabilities to compensate for the lack of haptic information, recreating the perception of haptic feedback from visual cues of the surgical scene [86]. Recent studies demonstrate different performances in MIRS procedures between experienced and novice surgeons, suggesting that haptic feedback affects performances differently based on the operator's level of experience with the robot [87]. Haptic feedback could significantly affect the performances of novice surgeons, reducing training duration and improving the effectiveness of the procedures.

A large number of surgical tasks can benefit from the introduction of collision avoidance techniques. During robotic polypectomy, as explained in Chapter 3, one surgical tool has to cut around the polyp while another tool keeps raised the surface of the polyp [88]. In this procedure, the surgeon performs a first cutting operation, then lifts the surface of the polyp and executes another cutting task. Automatic robotic assistance to avoid collision between the surgical tools can alleviate the surgeon's workload during the execution of this task and can allow the surgeon to focus on following the polyp margins. In procedures requiring tissues removal with the use of electrocautery, the direct coupling that occurs with a conductor, such as another tool, could burn non-targeted tissue [19]. The collision between surgical tools in MIRS can be avoided with the application of advanced shared control techniques. In particular, VFs can impose collision avoidance by rendering haptic cues to the surgeon. FRVFs restrict the motion of the robot's tool tip through a repulsive force rendered to the surgeon. The dVRK allows testing new control methods and it is already used to test VF-based methods [88] [17]. Since dVRK robot joints are driven through cables that introduce elasticity, backlash, and nonlinear friction [55], tools position information obtained through direct kinematics is affected by errors and thus requires correction. Therefore, to ensure a correct application of the VF, a method for surgical tool tracking is strictly needed.

5.1.1 Related Works

As exposed in Chapter 2, since its introduction, shared control techniques have had great success in surgical applications and for collision and obstacle avoidance. Li et al. presented an online collision avoidance method for real-time interactive control of a surgical robot in geometrically complex environments, such as the sinus cavities [18]. Ren et al. [89] proposed dynamic active constraints using medical images. The system builds potential fields to reduce the contact force between the tool tips. Xia et al. [90] reduced the proportional gain in an admittance control law according to the distance between the tool tip and the nearest obstacle. This allowed the system to smoothly avoid collisions. The method proposed in [72] uses depth information for FRVFs creation and it is generally applicable only for collision avoidance on the tool tips. Banach et al. proposed a Forbidden Region Active Constraints strategy to avoid surgical tool clashing and, at the same time, the collision with patient anatomy using elastoplastic frictional force control model [19]. In this work, the current poses of the tools are tracked in real-time from the robot's kinematics of the dVRK. Thus, a non-compensated position error could compromise the effectiveness of the method.

In the literature, most of the tracking methods used to correct the surgical tool position error are realized using the sensors of the robotic system or using external sensors integration, but still obtaining limited accuracy. A significant improvement is introduced by image-based approaches, detecting the tool's position and orientation in the camera reference frame. In [91], the authors presented a survey about vision-based and marker-less surgical tool tracking. The works can be classified based on the segmentation and tracking methods [92], most of them exploiting Random Forest (RF) or CNN techniques. In [93], the authors combine a region-based segmentation technique with point-based pose estimation, using prior knowledge of the instrument shape through classification with a Random Forest (RF), besides temporal motion is incorporated with a Kalman filter. Du et al. [94] proposed a 2D tracker based on a Generalized Hough Transform using SIFT features which can both handle complex environmental changes and recover from tracking failure. They extended the work in [95], presenting a 2D pose estimation framework for articulated endoscopic surgical instruments, which involves a fully convolutional detection-regression network (FCN) and a multi-instrument parsing component.

5.1.2 Contribution

This chapter proposes a surgical tool collision-avoidance method in MIRS. The goal is to improve safety in surgical procedures, enhancing especially novice surgeons' abilities. The method is tested on the dVRK robotic system. FRVFs are used to avoid surgical tool clashing, by rendering a repulsive force to the surgeon which is inversely proportional to the distance between tools. The method includes a marker-less surgical tool tracking technique using an EKF that couples vision and kinematics information to enhance the robustness of VFs application. Visual information allows overcoming the large position error, that occurs on the dVRK kinematics, especially when the surgical tools interact with the environment. While kinematics data reinforce the method in the presence of visual occlusions. To validate the method, an extensive study involving human subjects is conducted on two groups of surgeons, namely experts and novice surgeons, each group is constituted by 6 subjects. The goal is to demonstrate significant improvement in performances caused by the introduction of force cues. The pipeline of the method is articulated as follows:

1. pre-operative calibration and stereo endoscopic images acquisition;
2. tool segmentation and tool tip pose estimation from vision algorithm;
3. kinematic and vision data fusion with EKF;
4. VFs generation and force rendering.

5.2 System Description

5.2.1 dVRK Robot

This work considers both PSMs commanded by two MTMs of the MTMs and the endoscopic images obtained by the ECM. To generate force cues, as in Chapter 3 and Chapter 4 the MTMs are controlled through an impedance controller. Figure 5.1 shows the reference frames definition. The base frame, $\mathcal{F}_{b_1} : (\boldsymbol{O}_{b_1} - \boldsymbol{x}_{b_1}\boldsymbol{y}_{b_1}\boldsymbol{z}_{b_1})$, is positioned at the PSM1 RCM. Likewise, $\mathcal{F}_{b_2} : (\boldsymbol{O}_{b_2} - \boldsymbol{x}_{b_2}\boldsymbol{y}_{b_2}\boldsymbol{z}_{b_2})$ is the base frame centred in the PSM2 RCM. All the measurements in this work will be expressed referring to the base frame $\mathcal{F}_{b_2}$ of the PSM2. The frames $\mathcal{F}_{g_1} : (\boldsymbol{O}_{g_1} - \boldsymbol{x}_{g_1}\boldsymbol{y}_{g_1}\boldsymbol{z}_{g_1})$ and $\mathcal{F}_{g_2} : (\boldsymbol{O}_{g_2} - \boldsymbol{x}_{g_2}\boldsymbol{y}_{g_2}\boldsymbol{z}_{g_2})$ are the grippers frames. The direct kinematics of the dVRK allows computing the current position of the tools in the Cartesian space, providing the coordinates of $\boldsymbol{O}_{g_1}$ and $\boldsymbol{O}_{g_2}$ in $\mathcal{F}_{b_1}$ and $\mathcal{F}_{b_2}$ respectively. The reference frames

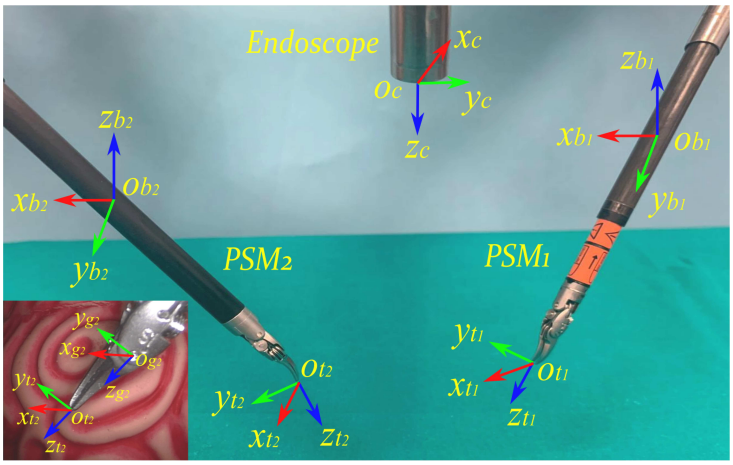


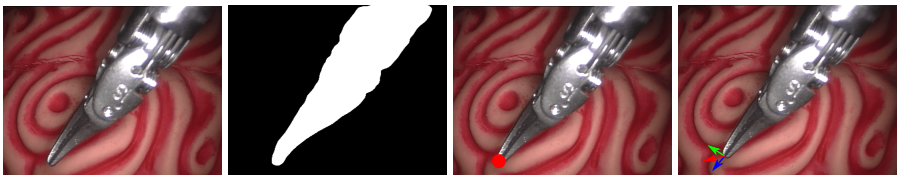
Figure 5.1: Experimental setup and frames definition. Gripper frame: $\mathcal{F}_{g_2} : (\boldsymbol{O}_{g_2} - \boldsymbol{x}_{g_2}\boldsymbol{y}_{g_2}\boldsymbol{z}_{g_2})$ and Tool tip frame $\mathcal{F}_{t_2} : (\boldsymbol{O}_{t_2} - \boldsymbol{x}_{t_2}\boldsymbol{y}_{t_2}\boldsymbol{z}_{t_2})$.

$\mathcal{F}_{t_1} : (\boldsymbol{O}_{t_1} - \boldsymbol{x}_{t_1}\boldsymbol{y}_{t_1}\boldsymbol{z}_{t_1})$ and $\mathcal{F}_{t_2} : (\boldsymbol{O}_{t_2} - \boldsymbol{x}_{t_2}\boldsymbol{y}_{t_2}\boldsymbol{z}_{t_2})$ have their origins in the PSM1 and PSM2 tool tips, respectively. For each PSM, the frames $\mathcal{F}_g$ and $\mathcal{F}_t$ have the same orientation and the origin of $\mathcal{F}_t$ is translated of 1 cm along the z-axis of $\mathcal{F}_g$. As in Chapter 3, Zhang stereo camera calibration is performed to define the camera reference frame $\mathcal{F}_c : (\boldsymbol{O}_c - \boldsymbol{x}_c\boldsymbol{y}_c\boldsymbol{z}_c)$ and a hand-eye calibration is performed to find the transformation $\boldsymbol{T}_c^{b_2}$ between $\mathcal{F}_{b_2}$ and $\mathcal{F}_c$. During the calibration process, the tool is placed in ten fixed positions and the transformation is computed adopting an absolute orientation formulation [78]. A hand-eye calibration is performed to find the transformation $\boldsymbol{T}_{b_1}^{b_2}$ between the fixed frames of each robotic arm.

5.2.2 Tool Segmentation and 3D Reconstruction

The method directly uses laparoscopic images to track the surgical instrument. A deep learning solution for instrument semantic segmentation is employed. It is based on U-Net architecture, which is a fully CNN, composed of a contracting path to capture context and by an expanding path that enables precise localization [96]. The system adopts the U-Net modification proposed in [97], called TernausNet that uses pre-trained VGG16 networks as an encoder. The network is trained using the dataset provided for MICCAI 2017 Endoscopic Vision Sub-Challenge: Robotic Instrument Segmentation [2] con-

[2] http://endovissub2017-roboticinstrumentsegmentation.grand-challenge.org/



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

Figure 5.2: Segmentation method: (a) Original frame; (b) Binary mask; (c) Point identification in the image plane; (d) Reference frame definition.

sisting of 8×225-frame sequences of high-resolution stereo camera images acquired from a da Vinci®Xi surgical system during several different porcine procedures, with a 2 Hz frame rate. The output of the model is an image in which each pixel is the probability value of belonging to the instrument or background area. Then the binary segmentation is obtained, in which all the instrument pixel values are set as 255 and all the background pixel values are set as 0. The homographic transformation $\boldsymbol{H}$ between the original left and right images are computed, using Scale-Invariant Feature Transform (SIFT) for features detection and Fast Library for Approximate Nearest Neighbors (FLANN) for matching, as in [88]. To detect the tool tip on the image plane, the search area range is reduced by re-projecting the tip kinematic position on the image plane and by constructing a rectangle centered on the projected point. Then, the 3D position of the PSM2 tip, expressed in the camera frame $\mathcal{F}_c$, is reconstructed by using a triangulation method with direct linear transform. The tool orientation is computed solving the PnP problem, which allows computing the orientation of the object from a set of n correspondences between 3D points and their 2D projections [98]. In this case, the line of symmetry of the tool is computed, allowing the identification of four specific points on the line in the image plane and their correspondent 3D coordinates thanks to the knowledge of the tool's geometry. Finally, using transformation $\boldsymbol{T}_c^{b_2}$, the tool tip position and orientation of PSM2 is found, expressed in the base frame $\mathcal{F}_{b_2}$. Figure 5.2 shows the results of the segmentation method.

5.2.3 Surgical Tool Tracking

For the estimation and tracking of the instrument pose, the EKF is used. Kalman filtering allows combining visual information from the endoscope with the robot kinematics [85]. The entire formulation is referred to PSM2 and the subscript 2 is omitted in this subsection.

The filter provides an estimate of the tool tip pose $\boldsymbol{\zeta} = [\boldsymbol{p}_t, \boldsymbol{q}_t]^T$, being $\boldsymbol{p}_t$ the true tool position, and $\boldsymbol{q}_t = [\boldsymbol{\eta}_t, \boldsymbol{\epsilon}_t]^T$ its quaternion-based true orientation in the base frame $\mathcal{F}_b$. The prediction step provides a preliminary estimation of the instrument pose through the linear and angular velocities of the gripper provided by the manipulator kinematics. Then, the vision-based estimated pose is used in the filter correction step. The process dynamics for the state vector $\boldsymbol{\zeta}$ is given by:

$$\begin{cases} \dot{\boldsymbol{p}}_t = \boldsymbol{v}_g + \boldsymbol{S}\left(\boldsymbol{\omega}_g\right) \boldsymbol{r}_{gt} + \boldsymbol{n}_p \\ \dot{\boldsymbol{q}}_t = \dfrac{1}{2} \boldsymbol{\Omega}\left(\boldsymbol{\omega}_g\right) \boldsymbol{q}_t + \boldsymbol{n}_q \end{cases} \tag{5.1}$$

where $[\boldsymbol{v}_g, \boldsymbol{\omega}_g]^T$ are the linear and angular velocity of the gripper in $\mathcal{F}_b$, $\boldsymbol{S}(\cdot)$ is the skew-symmetric operator, $\boldsymbol{r}_{gt}$ is the position vector of the tool tip respect to the gripper, $\boldsymbol{n} = [\boldsymbol{n}_p, \boldsymbol{n}_q]^T \sim \mathcal{N}\left(0, \boldsymbol{N}\right)$ is the process noise and

$$\boldsymbol{\Omega}\left(\boldsymbol{\omega}\right) = \begin{bmatrix} 0 & -\boldsymbol{\omega}^T \\ \boldsymbol{\omega} & \boldsymbol{S}\left(\boldsymbol{\omega}\right) \end{bmatrix}. \tag{5.2}$$

The error state vector is defined as $\widetilde{\boldsymbol{\zeta}} = [\widetilde{\boldsymbol{p}}, \delta\widetilde{\boldsymbol{\theta}}]^T$. The orientation error $\delta\widetilde{\boldsymbol{\theta}}$ is the 3×1 small-angle approximation vector of the quaternion orientation error. The vision algorithm computes the 3D pose of the tool tip, so the measurement model is given by:

$$\boldsymbol{y} = \boldsymbol{\zeta} + \boldsymbol{m} \tag{5.3}$$

where $\boldsymbol{m} \sim \mathcal{N}\left(0, \boldsymbol{M}\right)$ is the measurement noise. Then:

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{S}\left(\boldsymbol{\omega}_g\right) & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \boldsymbol{S}\left(\boldsymbol{\omega}_g\right) \end{bmatrix}; \qquad \boldsymbol{H} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \boldsymbol{I}_3 \end{bmatrix} \tag{5.4}$$

where $\boldsymbol{F}$ and $\boldsymbol{H}$ are respectively the control and measurement matrix used in the EKF implementation. The output of the EKF consists in the current pose of frame $\mathcal{F}_t$ of PSM2 with respect to the base frame $\mathcal{F}_b$.

5.2.4 Virtual Fixtures Generation

The collision avoidance between the two tools is ensured by the application of a FRVFs. To this purpose, the VF is defined as the swept surface along the tool axis, the forbidden region is around the PSM2. The VF has a cylindrical shape with a radius that is double the tool radius.

Assuming that the last two joints are kept still, the cylinder axis direction corresponds to $\boldsymbol{z}_{t_2}$ axis of $\mathcal{F}_{t_2}$ tracked by the EKF. The current pose of PSM1 is tracked in $\mathcal{F}_{b_1}$ using the dVRK kinematics and then mapped in $\mathcal{F}_{b_2}$ through the transformation matrix $\boldsymbol{T}_{b_1}^{b_2}$. The minimum distance between the PSM1 tool tip position $\boldsymbol{x}$ and the cylindrical FRVFs corresponds to the length of the line segment which joins perpendicularly the point to the axis minus the radius of the cylinder. A constraint enforcement method is defined, consisting in the application of a spring-damper like force:

$$\boldsymbol{f}_{VF}(\tilde{\boldsymbol{x}}, \dot{\tilde{\boldsymbol{x}}}) = -\boldsymbol{K}_{VF}\tilde{\boldsymbol{x}} - \boldsymbol{D}_{VF}\dot{\tilde{\boldsymbol{x}}} \tag{5.5}$$

where $\tilde{\boldsymbol{x}} = \boldsymbol{x}_d - \boldsymbol{x}$ is the displacement between the point $\boldsymbol{x}_d$, belonging to the constraint geometry having minimum distance from $\boldsymbol{x}$. The matrices $\boldsymbol{K}_{VF}$ and $\boldsymbol{D}_{VF}$ are properly designed diagonal and positive definite. The external force is not directly measurable, it is estimated by resorting to a non-linear dynamic observer [17], [75], and [76]. Finally, the force imposed by the VF is mapped on the MTM, so that it exhibits a repulsive behavior and pulls the robot end-effector away from the forbidden region. The MTM impedance controller exhibits a closed-loop behavior that can be described by

$$\boldsymbol{M}\ddot{\tilde{\boldsymbol{x}}} + \hat{\boldsymbol{D}}\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_{VF}\tilde{\boldsymbol{x}} = \boldsymbol{f}_h \tag{5.6}$$

where $\hat{\boldsymbol{D}} = \boldsymbol{D} + \boldsymbol{D}_{\boldsymbol{VF}}$ contains the damping assigned both by the impedance control and the constraint enforcement method.

5.3 Experimental Evaluation

The experimental validation is performed on dVRK robot, which is controlled at the MTM by an impedance controller, with $m_{ii} = 1.5$ and $d_{ii} = 0$, being the (i, i) entries of the matrices $\boldsymbol{M}$ and $\boldsymbol{D}$, respectively. The $\boldsymbol{D}_{VF}$ has been adapted according to the stiffness variation such that $d_{VF,ii} = 2\sqrt{m_{ii}k_{VF,ii}}$ where $d_{VF,ii}$ and $k_{VF,ii}$ are the diagonal values of the matrices $\boldsymbol{D}_{VF}$ and $\boldsymbol{K}_{VF}$, respectively and $k_{VF} = 8$ N/m, as in [17]. The dVRK dynamic parameters are identified in [55]. During the experiments, two Endowrist®da Vinci tools are used: curved scissors and Prograsp™ forceps. The kinematics data from the dVRK are acquired at 200 Hz, while the vision-based system estimated the tool position at the camera frame rate of 25 Hz. The EKF approach allows overcoming this limitation, providing tool pose at 200 Hz. The tool segmentation is performed using GPU implementation on an NVIDIA®GTX 1080 Ti to speed up computation.

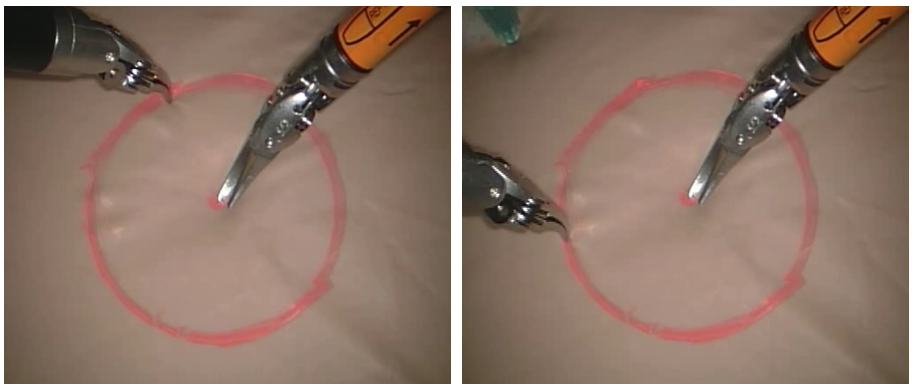


Figure 5.3: Two different frames of the second evaluation experiment; PSM1 tool holds the center of the circle; PSM2 moves following the circle.

5.3.1 Tracking Method Evaluation

The proposed tracking method is preliminarily evaluated on a simple task, executed with the dVRK robot. The task is planned *ad hoc* to reduce the variability introduced by the telemanipulation, and, thus, to obtain a reference target to measure the error. Two specific points, placed on a phantom tissue at a distance of 15 mm, are recorded offline from kinematic data, by holding the tool steady in the given positions. In this condition, the position error introduced by the kinematics, in the two selected points, is minimized since the tool is fixed and the interaction force with the phantom goes towards zero. After that, a linear path is defined analytically between the two points to serve as the ground truth for the evaluation. The experiment conducted to evaluate our tracking method, consists of moving the tool, in teleoperation mode, along the defined linear path drawn on the phantom. The task is performed slowly, with a duration of 12 seconds, to minimize the error along the linear path introduced by telemanipulation. During the task execution, the surgical tool is tracked using the EKF method. Then, the estimated pose is compared to the target linear path, obtaining mean absolute errors of 0.126 ± 0.08 mm along the x-axis, and 0.02 ± 0.01 mm along the y-axis. The results demonstrate the accuracy of our tracking method. Furthermore, we computed the error obtained just using kinematics information, obtaining mean absolute errors of 0.135 ± 0.08 mm along the x-axis, and 0.02 ± 0.01 mm along the y-axis. As expected, the pose error is similar to the one obtained with our tracking method because of the absence of interaction with the environment.

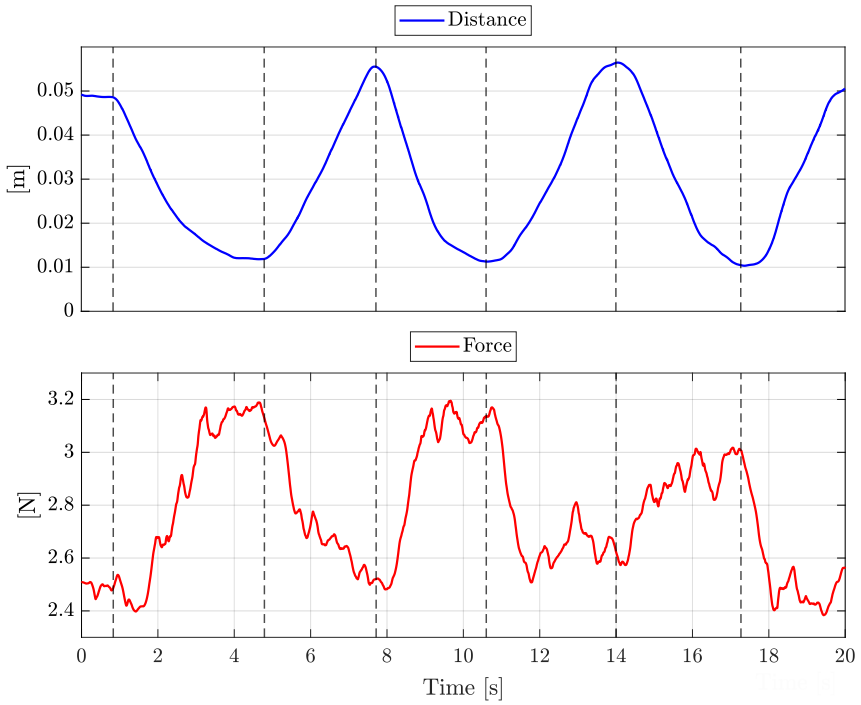


Figure 5.4: First evaluation experiment. Duration: 20 seconds. Time histories of: (Blue) Distance between surgical tools; (Red) Related estimated force norm.

5.3.2 Collision Avoidance Evaluation

The collision avoidance strategy is evaluated in two different tasks. During the first evaluation test, the PSM1 tool is fixed and the PSM2 is moved by the user in teleoperation mode towards PSM1. The collision avoidance strategy is applied during the entire duration of the test. Figure 5.4 shows the distance between the two surgical tools, computed considering the proposed tracking method, and the related haptic force norm rendered to the user through the master side (MTM) during the task. The maximum reached force is 3.2 N.

The second evaluation test consists of a human subject study to show significant differences in performance caused by the introduction of force feedback. The study involves 12 subjects divided into two groups, 6 experienced and 6 novice surgeons, based on self-evaluation about their experience in the use of da Vinci®Robotic system for minimally invasive surgical procedures. The study is articulated in two experiments using the dVRK robot in teleop-

eration mode. Taking inspiration from [19], the test simulates burning tissue with an electrocautery device. During each test, the subject keeps the PSM1 centered in the middle of a circle with a diameter of 20 mm. Meanwhile, the PSM2 has to follow the circular path for 270° from a definite starting point, as shown in Fig. 5.3. In the first experiment, the subjects perform the test 5 times moving the surgical tool in free motion. In the second experiment referred to as VF constraint tasks, they perform the same task 5 times with the proposed collision avoidance constraint applied. Each task has an average execution time of 10 seconds. Each subject is asked to try the test in advance, to become familiar with the task itself and with the dVRK platform. The minimum distance between the tools is considered as a performance parameter and it is computed using the proposed tracking method, in the VF constraint test the maximum force felt during the task is also computed.

Figure 5.5 and Fig. 5.6 show the mean values of the minimum distance between tools for novice and expert subjects during free motion and VF constraint tasks. The error bars represent the standard error of the means. To demonstrate the statistical relevance of the results, a comparison is made between the mean values of minimum distance, through a statistical unpaired t-test, with a significance level $\alpha = 0.05$. As presented in Table 5.1, the test shows statistically significant differences between the means for all subjects in the novice group. Moreover, it presents an increase in the minimum distance values of $\sim 10\%$ in collision tests for free-hand tests. The estimated force norm, rendered to the novice users through the master side (MTM), during the collision avoidance tasks, has a mean value of 3.1822 ± 0.5368 N. The expert group presents a mean force norm of 3.0493 ± 0.3629 N.

Table 5.1: Maximum force and t-test results on minimum distance for novice and expert users.

Novice	test	p	F_M [N]	Expert	test	p	F_M [N]
1	1	0.0044	2.4416	1	0	0.1352	3.4527
2	1	0.0127	3.0749	2	0	0.0856	2.8175
3	1	0.0030	3.3411	3	0	0.8286	3.5239
4	1	0.0219	2.8188	4	0	0.8757	2.6180
5	1	0.0206	3.9998	5	0	0.1140	3.0035
6	1	0.0012	3.4170	6	0	1	2.8800

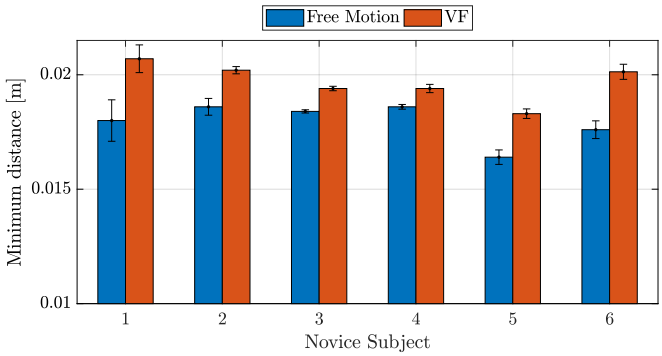


Figure 5.5: Novice subjects: Mean values of minimum distance between tools with standard error bars.

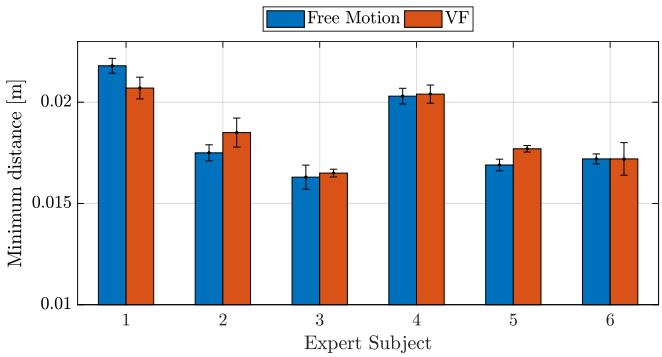


Figure 5.6: Expert subjects: Mean values of minimum distance between tools with standard error bars.

5.3.3 Discussion

A real comparison between the EKF-based tracking method and the kinematic measurements cannot be reliable without the use of an external sensor providing the ground truth of the tool pose. Indeed, we point out that the method used in Sect. 5.3.1 to evaluate the tracking algorithm is affected by the error due to the low resolution of the camera, and by the variability given by telemanipulation, even if it is minimized thanks to the ad hoc designed experiment. The experiments minimize the interaction with the environment introducing low position error in kinematic data, but ensuring the correct execution of the tool's movement following the defined path. In future works, we aim to sig-

nificantly improve the validation method through a more advanced calibration technique, and we aim to compare the tracking method with the kinematics measurements using an external camera to measure the true tool pose that can be considered as more appropriate ground truth for tracking evaluation. Figure 5.4 shows the repulsive force felt at the MTM when the distance between PSMs decreases. The method to generate the force is designed to have small forces values such as to be slightly perceived by the surgeon. This is because the purpose is not to interfere with the surgeon's actions, but to serve as an alarm to remind the presence of the other instrument in the proximity. Indeed, during the experiments, the maximum value of the force norm is 3.2 N.

The human subject study has shown a statistically significant difference regarding the mean of the minimum distance between the tools for the novice subjects. The VF test outperformed the free-hand test and this result suggests that feeling a haptic force during the task allows maintaining a safe distance between the surgical tools. On the contrary, in the free-hand test, the subject has no force feedback during the task and could dangerously reduce the distance between the tools. The maximum reached force is lower than 4 N and it does not create variation in the task performance.

As concern the expert subjects, the test does not show a statistically significant difference in the VF constraint task concerning the free motion task. Nevertheless, they were asked to compile the NASA-TLX questionnaire [99] to assess the perceived workload. The results of the questionnaires shown in Fig. 5.7 assess that the VF constrained task is not mentally, physically, and temporally demanding and the force feedback does not affect negatively the performances. On the contrary, it represents a comfortable reminder of the collision risk, that diminishes the user's mental workload. Similar results were obtained for novice surgeons.

5.4 Conclusion

The work presented in this chapter introduces a method based on haptic guidance and VFs that allows avoiding surgical tools collision in MIRS. A markerless algorithm allows estimating the PSM position and orientation, using kinematic and visual information. The PSM estimated pose is used to generate a FRVFs, that aims to avoid collision between the two instruments through a repulsive force felt at the MTM during the surgical task execution. The proposed strategies are evaluated through multiple experiments on dVRK, showing good results in improving novice surgeon's performance. Furthermore, the use of

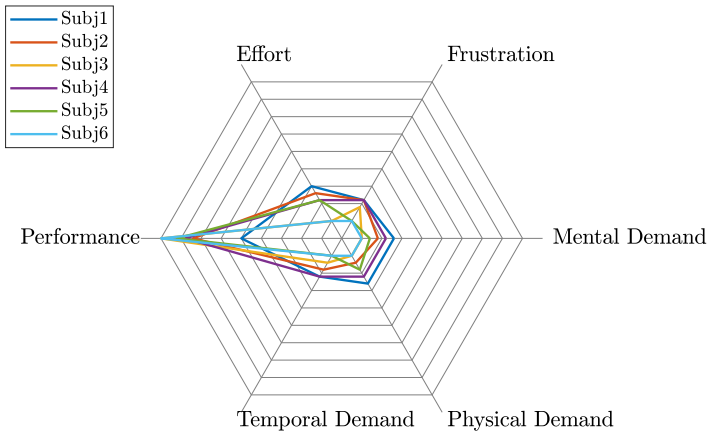


Figure 5.7: Radar graph of the TLX results on expert users during VF tasks.

VF allows also expert surgeon to better focus on the task, as far as the haptic force are small enough to suggest that the tools are dangerously close without affecting the performance. Therefore, the method can be considered effective both in a training stage of novice surgeons, as well as when the level of expertise increases. The goal for future works is to improve the accuracy of the tool pose estimation. For this purpose, a more advanced method for hand-eye calibration and 3D reconstruction will be considered.

Chapter 6

Multiple Tasks Execution using Control Barrier Functions in Surgical Robotics

Robot-aided surgical procedures involve a sequence of complex tasks, requiring dexterous movements and safety requirements that still depend on the surgeon's skills. Executing different tasks simultaneously could give the robot the ability to efficiently accomplish numerous assignments in safety conditions.

The present chapter introduces a multiple tasks execution framework for surgical robots, based on an optimization approach with tasks definition and execution achieved by means of CBFs. The proposed method allows easier execution of complex surgical tasks involving guidance and forbidden constraints executed at the same time and with different priorities, while DQ algebra allows efficient geometrical representation of the surgical environments. An experimental session is considered to test the entire method through different experiments in simulation and real environments.

6.1 Introduction

MIRS and RAMS require the surgeon to work with very high precision in a safety-critical environment and the impact of medical robots in surgical procedures is well known. Although, surgical robots cannot perform complex tasks relying on human operators to skillfully manipulate the robotic system and accomplish the above requirements. The ability to perform multiple tasks

is a crucial robot capability that could join task accomplishment and safety condition requirements. Surgical tasks are always composed of a series of subtasks that need to be executed with different priorities, reflecting environmental changes in the surgical scene. For example, MIRS dissection procedures require high-precision cutting with safety margins, avoiding tissue damages and possible collision between surgical tools [88]. While in RAMS, microvascular anastomosis requires a specific sequence of actions, involving the manipulation of small blood vessels, needle insertion and knot tying.

Frequently, an increase in safety and a reduction in mental workload has been achieved through the introduction of active constraints, that can prevent the surgical tool from crossing a forbidden region [100], or similarly generate constrained motion to guide the surgeon on a specific path ([17, 88]) or to accomplish robotic autonomous tasks. A large number of works show good results in the creation of active constraints but the complex surgical task remains not easy to implement, in particular the one that considers guidance and forbidden constraint at the same time. Recently, CBFs are successfully used to ensure the safety of robots in dynamic environments [101]. Originally formulated through the definition of a control input solving a simple convex optimization problem, CBFs could be extended to multiple task accomplishment.

In this scenario, a general method for multiple tasks execution could notably improve the quality of intervention in complex surgical tasks performed by the robot autonomously or allow a more efficient way for robot teleoperation.

6.1.1 Related Works

Multiple works were presented to introduce active constraint in surgical robotics for task and safety accomplishment, as presented in Chapter 2. The most powerful method is to couple shared control techniques like VFs to haptic guidance, rendering force feedback to the surgeon when the robot reaches a defined target, as exposed in the previous chapters. Selvaggio et al. propose an online VF generation and adaptation guiding the surgeon during procedures [17], and the same approach is extended for surgical tool collision avoidance in [100]. Nonetheless, a complex design of the master interface makes force feedback particularly challenging. Moreover, most of these work encounter particular difficulties in VF geometrical definition caused by the complex surgical environments, limiting the interactions between the robot tool tip and the environment itself. Constrained optimization is the most suit-

able approach to design motion control laws on the surgical robot. Some works consider VFs based on quadratic programming, defining non-linear constraint or linear approximation without considering closed-loop stability. The approach proposed in [22] represents a novel unified method for robot control under teleoperation. The framework uses QP programming for closed-loop inverse kinematics and dynamic VFs are generated by adding linear constraints to the QP problem using sample geometrical primitives modeled via DQ algebra. Constraint velocity compensation is given considering a fixed and not controllable term called residual [23] and complex surgical tasks are particularly challenging with the proposed approach [102].

CBFs have been successfully used for robots control in multiple domains [101]. Other chapters in this thesis exploit the use of VFs to implement various types of tasks in a teleoperated environment, like collision avoidance or suggestion of a preferred path. Nonetheless, this approach does not consider the system dynamics, that can inevitably affect the constraint definition, not guaranteeing adherence. A modern formulation for dynamical systems safety is presented in [103] involving the forward invariance property of a subset of the state space of the system. Additionally, it has been shown that they can be also employed to achieve set stability in [104]. Finally, Notomista et al. presented a set-theoretic approach to multi-task execution and prioritization [105]. The paper introduces an important method extending set-based tasks, encoding them using CBFs, and executing them by means of a constrained optimization problem. Moreover, this approach could be extended to more complex tasks as required in surgical applications, adding more articulated geometrical information as the one obtained from the surgical scene.

6.1.2 Contribution

This chapter proposes a multiple tasks execution framework for different surgical robots. The method uses an optimization-based approach with tasks definition and execution achieved utilizing CBFs. The technique allows the accomplishment of complex surgical tasks by defining constraints that guide the robot's motion or prevent it from entering a forbidden region. Different assignments can be created involving joint limits and obstacle avoidance, creating forbidden areas to prevent damages to robot structure and the surrounding tissues, and defining specific paths the robot has to follow. Moreover, the method allows the constraint to evolve to reflect environmental changes and endogenous changes of the system. DQ algebra is exploited creating a more efficient geometrical representation of the surgical scene. The same framework can be

adopted both for autonomous task execution as to enhance teleoperation control input. In this case, the method can be integrated with a haptic feedback approach on the robot master side. Ultimately, task prioritization allows the possibility of autonomously switching the priority of tasks during their execution.

6.2 Dual Quaternion Algebra

Like in [22, 23, 26, 106], this work uses DQ algebra which presents several advantages, like no representational singularities, a more compact definition. Moreover, this approach is useful for describing geometrical primitives, like lines and planes. Quaternion concept was first introduced by Hamilton [107] with the following representation $\mathbb{Q} = \{\eta, \epsilon\}$:

$$\mathbb{Q} \triangleq \{\eta + \hat{i}\epsilon_x + \hat{j}\epsilon_y + \hat{k}\epsilon_z : \eta, \epsilon_x, \epsilon_y, \epsilon_z \in \mathbb{R}\}, \tag{6.1}$$

with i, j, k as imaginary units and a unit-quaternion is constrained by the condition: $\eta^2 + \epsilon_x^2 + \epsilon_y^2 + \epsilon_z^2 = 1$. A rotation can be represented by a single unit norm quaternion $\boldsymbol{r} \in \mathbb{S}^3$ as:

$$\boldsymbol{r} = \cos(\frac{\phi}{2}) + v\sin(\frac{\phi}{2}), \tag{6.2}$$

where $\phi \in \mathbb{R}$, where $\mathbb{S}^3 \triangleq \{\boldsymbol{r} \in \mathbb{Q} : ||\boldsymbol{r}|| = 1\}$, $\phi \in \mathbb{R}$ and v are the axis and angle of rotation. A dual number is defined as: $a + \varepsilon b$, where $a, b \in \mathbb{R}$ and ε is a dual unit with $\varepsilon^2 = 0$ but $\varepsilon \neq 0$. Similarly, a dual quaternion is a dual entity with quaternion components and its set is defined as:

$$\mathcal{Q} \triangleq \{\boldsymbol{b} + \varepsilon\boldsymbol{b}' : \boldsymbol{b}, \boldsymbol{b}' \in \mathbb{Q}, \varepsilon^2 = 0, \varepsilon \neq 0\}, \tag{6.3}$$

where, $\mathbb{Q}$ is a single quaternion defined in set (6.1). Giving a DQ $\underline{\boldsymbol{b}} \in \mathcal{Q}$ as $\underline{\boldsymbol{b}} = \eta_1 + \hat{i}\epsilon_{x1} + \hat{j}\epsilon_{y1} + \hat{k}\epsilon_{z1} + \varepsilon\left(\eta_2 + \hat{i}\epsilon_{x2} + \hat{j}\epsilon_{y2} + \hat{k}\epsilon_{z2}\right)$, the operators $\mathcal{P}$ (primary part) and $\mathcal{D}$ (dual part) are defined as:

$$\mathcal{P} \triangleq \eta_1 + \hat{i}\epsilon_{x1} + \hat{j}\epsilon_{y1} + \hat{k}\epsilon_{z1},$$
$$\mathcal{D} \triangleq \eta_2 + \hat{i}\epsilon_{x2} + \hat{j}\epsilon_{y2} + \hat{k}\epsilon_{z2},$$

and the real and imaginary parts are:

$$\mathrm{Re}(\underline{\boldsymbol{b}}) \triangleq \eta_1 + \varepsilon\eta_2,$$
$$\mathrm{Im}(\underline{\boldsymbol{b}}) \triangleq \hat{i}\epsilon_{x1} + \hat{j}\epsilon_{y1} + \hat{k}\epsilon_{z1} + \varepsilon\left(\hat{i}\epsilon_{x2} + \hat{j}\epsilon_{y2} + \hat{k}\epsilon_{z2}\right).$$

DQ algebra allows defining a compact representation of the rigid motions with a single 8-dimensional vector. A rigid rotation is described by a quaternion (6.2) and a position (x, y, z) in $\mathbb{R}^3$ can be expressed by a quaternion $\boldsymbol{p} = \left(x\hat{i} + y\hat{j} + z\hat{k}\right) \in \mathbb{H}_p$, since the set $\mathbb{Q}_p \triangleq \{\boldsymbol{p} \in \mathbb{Q} : \mathrm{Re}\,(\mathbf{p}) = 0\}$ is isomorphic in $\mathbb{R}^3$. Finally a tridimensional pose of a rigid body can be expressed as a unit dual quaternion $\underline{\boldsymbol{x}} \in \underline{\mathcal{S}}$:

$$\underline{\boldsymbol{x}} = \boldsymbol{r} + \varepsilon \frac{1}{2} \boldsymbol{p}\boldsymbol{r}, \tag{6.4}$$

where, $\underline{\mathcal{S}} \triangleq \{\underline{\boldsymbol{x}} \in \mathbb{Q} : ||\underline{\boldsymbol{x}}|| = 1\}$ with $\underline{\boldsymbol{x}} \in \mathcal{Q}$, $\boldsymbol{p} \in \mathbb{Q}_p$ and $\boldsymbol{r} \in \mathbb{S}^3$ are the position and orientation respectively. As in [22], in this chapter the operator vec_4 and vec_8 are used to map single quaternion in $\mathbb{R}^4$ and dual quaternion in $\mathbb{R}^8$ respectively.

6.2.1 Kinematic Model in DQ

Considering a n DoFs manipulator with a base reference frame defined as $\mathcal{F}_{base}$, the first robot joint corresponds to $\mathcal{F}_0$. The robot has n links and $i = 0, \ldots, n-1$ joints, the dual quaternion transformation between two consecutive joints is $\underline{\boldsymbol{x}}_{i+1}^{i}(\boldsymbol{q}_i)$, where $\boldsymbol{q} \in \mathbb{R}^n$ is the generalized joint coordinates and each transformation depends on the correspondent joint value. Two additional transformations between the word reference frame and the base frame $\underline{\boldsymbol{x}}_0^{base}$ and between the last joint and the robot end-effector $\underline{\boldsymbol{x}}_{ee}^{n}$, can also be considered. Thus, forward kinematics is computed using post-multiplication as:

$$\underline{\boldsymbol{x}}_{ee}^{base}(\boldsymbol{q}) = \underline{\boldsymbol{x}}_0^{base}\underline{\boldsymbol{x}}_1^0(\boldsymbol{q}_0)\,\underline{\boldsymbol{x}}_2^1(\boldsymbol{q}_1)\ldots\underline{\boldsymbol{x}}_n^{n-1}(\boldsymbol{q}_{n-1})\,\underline{\boldsymbol{x}}_{ee}^{n}. \tag{6.5}$$

Each DQ transformation $\underline{\boldsymbol{x}}_{i+1}^{i}(\boldsymbol{q}_i)$ between two consecutive links is defined multiplying four specific transformation, one for each DH parameter. $\underline{\boldsymbol{x}}_{ee}^{base}$ is the DQ representing position and orientation of robot end-effector with the respect of the base reference frame.

6.2.2 Differential Kinematics and Jacobian in DQ

The differential kinematics is defined as:

$$\mathrm{vec}_8\dot{\underline{\boldsymbol{x}}}(\boldsymbol{q}) = \boldsymbol{J}_{\underline{\boldsymbol{x}}}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{6.6}$$

where $\dot{\underline{\boldsymbol{x}}}(\boldsymbol{q}) \in \mathbb{R}^m$ is the DQ vector of task space velocities that depends on robot coordinates $\boldsymbol{q} \in \mathbb{R}^n$ and $\boldsymbol{J}_{\underline{\boldsymbol{x}}}(\boldsymbol{q}) \in \mathbb{R}^{m\times n}$ is the DQ Jacobian. Considering the forward kinematics in (6.5), the partial derivative with respect to joint

value $\boldsymbol{q}_i$ can be obtained using the chain rule as in [106]:

$$\frac{\delta(\underline{\boldsymbol{x}}_{ee}^{base}(\boldsymbol{q}))}{\delta \boldsymbol{q_i}} = \frac{\underline{\boldsymbol{x}}_0^{base} \prod_{i=0}^{n-1} \underline{\boldsymbol{x}}_{i+1}^{i}(\boldsymbol{q_i})\, \underline{\boldsymbol{x}}_{ee}^{n}}{\delta \boldsymbol{q_i}} =$$

$$= \underline{\boldsymbol{x}}_{i-1}^{base} \frac{\delta(\underline{\boldsymbol{x}}_{i+1}^{i}(\boldsymbol{q_i}))}{\delta \boldsymbol{q_i}} \underline{\boldsymbol{x}}_{ee}^{i+1}, \tag{6.7}$$

Therefore, the Jacobian for a given kinematic chain is a $8 \times n$ matrix:

$$\boldsymbol{J}_{\underline{\boldsymbol{x}}}(\boldsymbol{q}) = \left[\text{vec}_8 \left(\underline{\boldsymbol{x}}_0^{base} \frac{\delta(\underline{\boldsymbol{x}}_1^0(\boldsymbol{q_0}))}{\delta \boldsymbol{q_0}} \underline{\boldsymbol{x}}_{ee}^{1} \right), \ldots, \right.$$

$$\left. , \ldots, \text{vec}_8 \left(\underline{\boldsymbol{x}}_{n-1}^{base} \frac{\delta(\underline{\boldsymbol{x}}_n^{n-1}(\boldsymbol{q}_{n-1}))}{\delta \boldsymbol{q}_{n-1}} \underline{\boldsymbol{x}}_{ee}^{n} \right) \right]. \tag{6.8}$$

6.3 Multiple Tasks Execution

In this section, the objective is to define a task-execution method that allows the robot to accomplish different tasks, involving both the position and orientation of the robot. Some of them are defined in joint space, i.e avoiding joint limits or simply assume a specific joints configuration. Other tasks involve the definition of a point or path in Cartesian space the robot has to follow, or definite orientation the robot end-effector has to reach. More complex tasks concern collision and obstacle avoidance, creating forbidden areas to prevent damages to robot structure or the surrounding tissues. In particular, surgical procedures often require a combination of multiple tasks to accomplish constrained or guided assignment in joint or Cartesian space.

6.3.1 Tasks Definition

Consider a generic task variable $\boldsymbol{\sigma}(\boldsymbol{t}) \in \mathbb{R}^m$ for a n DoFs manipulator illustrated in Section 6.2.1, that related to robot joint variable as $\boldsymbol{\sigma}(\boldsymbol{t}) = f(\boldsymbol{q}(\boldsymbol{t}))$. The task is considered accomplished if the value $\boldsymbol{\sigma}(\boldsymbol{t})$ is controlled to reach a desired value $\boldsymbol{\sigma_d}(\boldsymbol{t})$. The relationship between task variable velocities and robot joint velocities is defined by (6.6) as: $\text{vec}_8 \dot{\boldsymbol{\sigma}} = \boldsymbol{J_\sigma} \dot{\boldsymbol{q}}$. The task can be achieved by integrating the locally inverse mapping and solving iteratively the closed-loop kinematics as in [7]:

$$\dot{\boldsymbol{q_d}} = \boldsymbol{J}_{\boldsymbol{\sigma}}^{*}(\dot{\boldsymbol{\sigma_d}} + K_p \text{vec}_8(\boldsymbol{e_\sigma})), \tag{6.9}$$

where, $\dot{\boldsymbol{q}}_{\boldsymbol{d}}$ the corresponding joint references for $\boldsymbol{\sigma_d}$, $\boldsymbol{J}_{\boldsymbol{\sigma}}^{*}$ is the Jacobian pseudo-inverse that satisfies the four Moore-Penrose conditions, $K_p \in \mathbb{R}^{m \times m}$ is a positive-definite gain matrix and $\boldsymbol{e_\sigma} = \boldsymbol{\sigma} - \boldsymbol{\sigma_d}$ is the task error whose dynamics is reduced. In case of robotic system redundancy a null projector operator is add to (6.9): $(\boldsymbol{I_n} - \boldsymbol{J}_{\boldsymbol{\sigma}}^{*}\boldsymbol{J_\sigma})\dot{\boldsymbol{q}}_{\boldsymbol{0}}$, where $\boldsymbol{I_n}$ is an $n \times n$ identity matrix and $\boldsymbol{q_0} \in \mathbb{R}^n$ a vector of arbitrary joint velocities.

This approach could be extended to accomplish multiple tasks at the same time and the concept of set-based task execution is introduced in [108], extending multiple tasks inverse kinematic control. The objective is to keep the task variable in a specific set $\mathcal{C}$ rather than leading it to the desired value. This work uses extended set-based tasks introduced in [105], in which a specific set-based task characterized by a set $\mathcal{C}$ is expressed as the super-level set of a smooth function $\boldsymbol{h} : \mathbb{R}^n \to \mathbb{R}$ as:

$$\mathcal{C} = \{\boldsymbol{\sigma} \in \mathcal{T} : \boldsymbol{h}(\boldsymbol{\sigma}, \boldsymbol{t}) \geq 0\}. \tag{6.10}$$

where, $\mathcal{T}$ is the task space. Based on this definition, an extended set-based task can be declared as keeping a desired set invariant, i.e. ensuring that the state of the robot never leaves the set and is asymptotically stable. Considering the desired function to track $\boldsymbol{\sigma_d}$ related to joint velocities by the relationship (6.6), the function $\boldsymbol{h}$ in (6.10) can be defined as:

$$\boldsymbol{h}(\boldsymbol{\sigma_d}, \boldsymbol{t}) = -\frac{1}{2}||\boldsymbol{e_\sigma}||^2, \text{ where } \boldsymbol{e_\sigma} = \boldsymbol{\sigma} - \boldsymbol{\sigma_d}(\boldsymbol{t}), \tag{6.11}$$

$\boldsymbol{\sigma_d}$ is a continuous function of time and $\boldsymbol{h}$ is a continuously differentiable with respect to both $\boldsymbol{\sigma_d}$ and $\boldsymbol{t}$, rendering $\mathcal{C}$ asymptotically stable and forward invariant. Therefore, the set $\mathcal{C} = \{\boldsymbol{\sigma} \in \mathcal{T} : \boldsymbol{\sigma} = \boldsymbol{\sigma_d}(\boldsymbol{t})\}$ implying that its invariance corresponds to the task $\boldsymbol{\sigma_d}$ being accomplished, as in [105].

6.3.2 Tasks Execution

The task execution is accomplished by turning forward invariance of subsets of the state space of the robots into a control input constraint by using CBFs. The CBFs proprieties of a smooth function $\boldsymbol{h} : \mathbb{R}^n \to \mathbb{R}$ ensure the forward invariance of its the super-level set and the accomplishment of the related task [101]. The robot is modeled as a non-linear affine control system accessible through a task variable $\boldsymbol{\sigma}$ as:

$$\begin{cases} \dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u} \\ \boldsymbol{\sigma} = k(\boldsymbol{x}), \end{cases} \tag{6.12}$$

where $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ is the system state, $\boldsymbol{u} \in \mathcal{U} \subseteq \mathbb{R}^p$ is the control input in a set of admissible control values $\mathcal{U}$, $\boldsymbol{\sigma} \in \mathcal{T}$ is the system output, f and g are Lipschitz-continuous vector fields, and $k : \mathbb{R}^n \to \mathcal{T}$ is a smooth map representing the task forward kinematics. The kinematic model of robotic manipulator is considered, assuming that its joint velocities are controllable. In this case, the state $\boldsymbol{x}$ is the vector of joint coordinates $\boldsymbol{q}$ and the input $\boldsymbol{u}$ is the vector of joint velocities $\dot{\boldsymbol{q}}$. From definition of CBFs in [101], a continuously differentiable function $\boldsymbol{h} : \mathcal{X} \subseteq \mathbb{R}_{\geq 0} \to \mathbb{R}$ with a super-level set $\mathcal{C} \subset \mathcal{X} \subset \mathcal{T}$ is defined as a time-varying control barrier function if there exists a Lipschitz continuous extended class $\mathcal{K}_\infty$ function γ for all $\delta \in \mathcal{X}$, such that

$$\sup_{\boldsymbol{u} \in \mathcal{U}} \left\{ \frac{\delta \boldsymbol{h}}{\delta \boldsymbol{t}} + \frac{\delta \boldsymbol{h}}{\delta \boldsymbol{\sigma}} \frac{\delta \boldsymbol{\sigma}}{\delta \boldsymbol{x}} f(\boldsymbol{x}) + \frac{\delta \boldsymbol{h}}{\delta \boldsymbol{\sigma}} \frac{\delta \boldsymbol{\sigma}}{\delta \boldsymbol{x}} g(\boldsymbol{x}) \boldsymbol{u} \right\} \geq -\gamma(\boldsymbol{h}(\boldsymbol{\sigma})). \tag{6.13}$$

Considering the function $\boldsymbol{h}$ as defined in (6.11) and control input set a $\boldsymbol{u} = \dot{\boldsymbol{q}}$, the condition in (6.13) can be expressed as:

$$\sup_{\dot{\boldsymbol{q}} \in \mathcal{U}} \left\{ \boldsymbol{e}_{\boldsymbol{\sigma}}^T \dot{\boldsymbol{\sigma}}_d + \boldsymbol{e}_{\boldsymbol{\sigma}}^T \boldsymbol{J}_{\boldsymbol{\sigma}} \dot{\boldsymbol{q}} \right\} \geq -\gamma(\boldsymbol{h}(\boldsymbol{\sigma}_{\boldsymbol{d}})). \tag{6.14}$$

According to (6.14), describes that CBFs ensure forward invariance and asymptotic stability of the set $\mathcal{C}$ by enforcing the constraints on the joint velocities [105]. If $\boldsymbol{h}$ is a CBF on $\mathcal{X}$, then any Lipschitz continuous controller $\dot{\boldsymbol{q}} \in \mathcal{V}(\boldsymbol{q}, \boldsymbol{t})$, where $\mathcal{V}(\boldsymbol{q}, \boldsymbol{t}) = \{\boldsymbol{e}_{\boldsymbol{\sigma}}^T \dot{\boldsymbol{\sigma}}_d + \boldsymbol{e}_{\boldsymbol{\sigma}}^T \boldsymbol{J}_{\boldsymbol{\sigma}} \dot{\boldsymbol{q}} + \gamma(\boldsymbol{h}(\boldsymbol{\sigma}_{\boldsymbol{d}}) \geq 0\}$, for the robotic system, renders the set $\mathcal{C}$ forward invariant. Additionally, the set $\mathcal{C}$ is asymptotically stable in $\mathcal{X}$.

Ultimately, the minimum control input subject to a set of constraints that enforce the execution of tasks that is obtained by solving the following optimization problem formulated to synthesize a velocity controller $\dot{\boldsymbol{q}}$:

$$\begin{aligned} \underset{\dot{\boldsymbol{q}}}{\text{minimize}} \quad & \|\dot{\boldsymbol{q}}\|^2 \\ \text{subject to} \quad & \boldsymbol{e}_{\boldsymbol{\sigma}}^T(\boldsymbol{t})\, \boldsymbol{J}_{\boldsymbol{\sigma}}(\boldsymbol{q}) \dot{\boldsymbol{q}} \geq \\ & \quad -\gamma(\boldsymbol{h}(\boldsymbol{\sigma}_{\boldsymbol{d}}, \boldsymbol{t})) - \boldsymbol{e}_{\boldsymbol{\sigma}}^T(\boldsymbol{t}) \dot{\boldsymbol{\sigma}}_{\boldsymbol{d}}(\boldsymbol{t}), \end{aligned} \tag{6.15}$$

The presented formulation allows the robot to execute different tasks autonomously in a surgical environment by computing the h-function $\boldsymbol{h}(\boldsymbol{\sigma}_{\boldsymbol{d}}, \boldsymbol{t})$ depending on the error $\boldsymbol{e}_{\boldsymbol{\sigma}}$ and the dual quaternion Jacobian $\boldsymbol{J}_{\boldsymbol{\sigma}}$ related to the specific desired task variable. The same framework can be adopted even if the robot already has a nominal control input, which is usually represented

by teleoperation control input in surgical robotics. In this case, the goal is to modify the nominal controller in minimal way so as to satisfy the defined constraint [101]. The cost function in the optimization problem becomes the difference between the input control $\dot{\boldsymbol{q}}$ and the nominal control $\dot{\boldsymbol{q}}_n$, as $\|\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_n\|^2$.

6.3.3 Multiple Tasks Execution and Prioritization

The execution of multiple tasks can be obtained by adding more constraints to the optimization problem. Considering a set of different tasks T_m with $m = 1, \ldots, M$, that need to be executed, the optimization problem is enriched by adding M constraints enforcing the execution of tasks:

$$\begin{aligned} \underset{\dot{\boldsymbol{q}}}{\text{minimize}} \quad & \|\dot{\boldsymbol{q}}\|^2 \\ \text{subject to} \quad & \boldsymbol{A}(\boldsymbol{h}(\boldsymbol{\sigma}, \boldsymbol{t}))\dot{\boldsymbol{q}} \geq \boldsymbol{b}(\boldsymbol{h}(\boldsymbol{\sigma}, \boldsymbol{t})), \end{aligned} \tag{6.16}$$

where, $\boldsymbol{A}(\boldsymbol{h}(\boldsymbol{\sigma}, \boldsymbol{t})) \in \mathbb{R}^{m \times n}$ and $\boldsymbol{b}(\boldsymbol{h}(\boldsymbol{\sigma}, \boldsymbol{t})) \in \mathbb{R}^{m \times 1}$ defined as:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{e}_{\boldsymbol{\sigma 1}}^T \boldsymbol{J}_{\boldsymbol{\sigma 1}} \\ \vdots \\ \boldsymbol{e}_{\boldsymbol{\sigma M}}^T \boldsymbol{J}_{\boldsymbol{\sigma M}} \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} -\gamma \boldsymbol{h}_{\boldsymbol{1}} - \boldsymbol{e}_{\boldsymbol{\sigma 1}}^T \dot{\boldsymbol{\sigma}}_{\boldsymbol{d1}} \\ \vdots \\ -\gamma \boldsymbol{h}_{\boldsymbol{M}} - \boldsymbol{e}_{\boldsymbol{\sigma M}}^T \dot{\boldsymbol{\sigma}}_{\boldsymbol{dM}} \end{bmatrix}, \tag{6.17}$$

The multi-task prioritization proposed in [105] is adopted, that ensures simultaneously multiple tasks execution and introduces the possibility give different task priorities. Task prioritization is established introducing a slack variable $\boldsymbol{\delta}_m$ to each task, which denotes the extension quantity to which the constraint corresponding to task T_m can be relaxed. Considering two different tasks T_m and T_n, an higher priority to T_m than T_n is given by using an additional constraint defined as $\boldsymbol{\delta}_m \leq \boldsymbol{\delta}_n / k$ where $k > 1$. Prioritization between tasks are encoded by adding a single linear constraint $\boldsymbol{K}\boldsymbol{\delta} \geq 0$ enforced, where $\boldsymbol{K}(\boldsymbol{t})$ is a prioritization matrix depending on k which specify the prioritization stack among the tasks. This formulation ensures that $\boldsymbol{K}(\boldsymbol{t})$ is Lipschitz continuous in time, then the controller $\dot{\boldsymbol{q}}^*(\boldsymbol{t})$ is Lipschitz continuous in time, as solution of the following optimization problem:

$$\begin{aligned} \underset{\dot{\boldsymbol{q}}}{\text{minimize}} \quad & \|\dot{\boldsymbol{q}}\|^2 + l\|\boldsymbol{\delta}\| \\ \text{subject to} \quad & \boldsymbol{A}(\boldsymbol{h}(\boldsymbol{\sigma}, \boldsymbol{t}))\dot{\boldsymbol{q}} \geq \boldsymbol{b}(\boldsymbol{h}(\boldsymbol{\sigma}, \boldsymbol{t})) \\ & \boldsymbol{K}(\boldsymbol{t})\boldsymbol{\delta} \geq 0, \end{aligned} \tag{6.18}$$

where, $l > 0$ is a scalar factor and $\boldsymbol{\delta} = [\boldsymbol{\delta}_1, \ldots, \boldsymbol{\sigma}_M]^T$ is the vector of slack variables corresponding to each task.

In this chapter, a time-varying prioritization matrix $\boldsymbol{K}(\boldsymbol{t})$ is adopted. Unlike [105], the priority between tasks is switched automatically based on the specific task to accomplish. The robot changes the prioritization matrix according to the value of function $\boldsymbol{h}$. For example, the robot has to accomplish two different tasks, T_1 reach a specific point in workspace and simultaneously T_2 avoid an obstacle. If the distance between the robot and target point is higher than the distance between robot and obstacle, a higher priority is provided to T_2 rather than T_1 allowing the robot to first avoid the obstacle and then reach the target point. Priorities switching may cause issues due to discontinuities in the controller that may appear during the transient. The proposed formulation guarantees sufficient conditions to ensure continuity in control input during priority switching, as demonstrated in [105].

6.4 Experimental Validation

The experimental section involves results on dVRK robot and Symani®Surgical System. The session includes a set of tests that replicate the realistic tasks required in MIRS and RAMS. The tests are performed utilizing the CoppeliaSim physics simulators of those systems and replicated using real dVRK robotic platform. Kinematic descriptions of the two robotic systems are provided in Section 2.4, considering only a single PSM or robotic arm of Symani®Surgical System. In all tests, the task spaces are joint space and Cartesian space.

Let consider $\underline{\dot{\boldsymbol{x}}} \in \mathbb{R}^8$ as the DQ vector representing end-effector position and orientation in Cartesian space, and $\boldsymbol{q} \in \mathbb{R}^6$ as the vector of the robot generalized coordinates in which the last degree of freedom, corresponding to the opening and closing motion of the gripper, is not considered. Kinematic model and DQ algebra are defined using DQ Robotic C++ library in [102] and the optimization problem is solved using IBM ILOG CPLEX. The dVRK robot is controlled using the open-source control software developed in [52, 53], that is based on ROS software layer to interface to the dVRK. While for simulation tests, the robot models are controlled on the CoppeliaSim scene using embedded scripts and ROS topics. In detail, Section 6.4.1 describes the proposed evaluation tests and the corresponding implemented tasks, while the results of the simulation and real tests are exposed in Section 6.4.2, Section 6.4.3 and Section 6.4.4 respectively.

6.4.1 Evaluation Tests Definition

Different tests are created to replicate basic surgical movements. The validation tests are divided into multiple subtasks, that are implemented as follows:

- **Joint Limits Avoidance**

  Joint limits avoidance is accomplished by forcing joint variables to stay within an upper and a lower bound. The h-function for each joint is:

  $$\boldsymbol{h}_{lim,i} = \frac{(\boldsymbol{q_{max,i}} - \boldsymbol{q_i})(\boldsymbol{q_i} - \boldsymbol{q_{min,i}})}{(\boldsymbol{q_{max,i}} - \boldsymbol{q_{min,i}})^2}, \tag{6.19}$$

  where, $\boldsymbol{q_i}$ is the i-th joint variable, $\boldsymbol{q_{max,i}}$ and $\boldsymbol{q_{min,i}}$ are the upper and lower bound of the i-th joint. In this case $\boldsymbol{J_\sigma}(\boldsymbol{q})$ is $\delta \boldsymbol{h}/\delta \boldsymbol{q}$.

- **Reach Target Position**

  Following the formulation in (6.11) and in (6.15), the position control task can be defined through the following h-function:

  $$\boldsymbol{h}_{pos} = -\frac{1}{2}||\text{vec}_4(\boldsymbol{p_e}) - \text{vec}_4(\boldsymbol{p_d})(\boldsymbol{t})||^2, \tag{6.20}$$

  where, $\boldsymbol{p_e}$ and $\boldsymbol{p_d}(\boldsymbol{t})$ are robot end-effector and target position in Cartesian space. While Jacobian parameter $\boldsymbol{J_\sigma}(\boldsymbol{q})$ is set as the translational part of the robot Jacobian $\boldsymbol{J_{pos}}$.

- **Reach Target Orientation**

  Orientation control task is defined considering the unit quaternion $\boldsymbol{r_e} = \{\eta_e(\boldsymbol{q}), \epsilon_e(\boldsymbol{q})\}$ and $\boldsymbol{r_d} = \{\eta_d, \epsilon_d\}$ associated to robot end-effector and target rotation in Cartesian space. Therefore, the orientation error can be described in term of quaternion as $\Delta \boldsymbol{r} = \{\Delta\eta, \Delta\epsilon\} = \boldsymbol{r_d} * \boldsymbol{r_e}^{-1}$. Hence, the orientation task is expressed as following:

  $$\boldsymbol{h}_{ori} = -\frac{1}{2}||\eta_e(\boldsymbol{q})\epsilon_d - \eta_d\epsilon_e(\boldsymbol{q}) - \boldsymbol{S}(\epsilon_d)\epsilon_e(\boldsymbol{q})||^2, \tag{6.21}$$

  where $\boldsymbol{S}(\cdot)$ is the skew-symmetric operator. While Jacobian parameter $\boldsymbol{J_\sigma}(\boldsymbol{q})$ is set as the rotational part of the robot geometric Jacobian $\boldsymbol{J_{rot}}$. The orientation error could be expressed also using other end-effector orientation representation, like Euler angles or angle and axis. It is worth remarking that, unlike Euler angles or axis and angle representation, a single rotation by the angle $-\phi$ about the axis $-v$ is defined by the

same quaternion as that associated with the rotation by angle ϕ about the axis v. Thus, the unit quaternion representation allows solving any non-uniqueness and singularity problems.

- **Obstacle Avoidance**

  Obstacle avoidance is obtained preventing the tool tip from colliding with an obstacle that defined as a cylindrical shaped object. As in [23, 26], the central axis of the obstacle is described as a Plücker line $\underline{\boldsymbol{l}} \in \mathbb{Q}_p$, given by a unit DQ: $\underline{\boldsymbol{l}} = \boldsymbol{l} + \varepsilon \boldsymbol{m}$, where $\boldsymbol{l}$ is the unit norm line direction and $\boldsymbol{m} = \boldsymbol{p}_l \times \boldsymbol{l}$ is the line moment with $\boldsymbol{p}_l$ as the object point on the line. The obstacle avoidance task is defined as:

$$\boldsymbol{h}_{obs} = \frac{1}{2}||\text{vec}_4(\boldsymbol{p_e} \times \boldsymbol{l} - \boldsymbol{m})^T||^2, \tag{6.22}$$

  where, $\boldsymbol{p_e}$ is the robot end-effector position in Cartesian space. The Jacobian parameter is obtained computing the time derivative of $\boldsymbol{h}_l$ depending on joint velocities:

$$\boldsymbol{J}_{obs} = \text{vec}_4(\boldsymbol{p_e} \times \boldsymbol{l} - \boldsymbol{m})^T \boldsymbol{S}(\boldsymbol{l})^T \boldsymbol{J_{pos}}, \tag{6.23}$$

  where, $\boldsymbol{S}(\boldsymbol{l})$ is the skew-symmetric operator of $\boldsymbol{l}$.

- **Forbidden Region Constraint**

  Forbidden regions, in which the robot is not allowed to go, are created considering a plane as boundary. As in [23, 26], plane representation in DQ space is given by $\underline{\boldsymbol{\pi}} \triangleq \boldsymbol{n}_\pi + \varepsilon \boldsymbol{d}_\pi$, where $\boldsymbol{n}_\pi \in \mathbb{Q}_p$ is the normal to the plane and $\boldsymbol{d}_\pi \in \mathbb{R}$ is the euclidean distance between an arbitrary point $\boldsymbol{p}_\pi$ and the origin of a reference frame on the plane. The constraint is expressed by the CBF function:

$$\boldsymbol{h}_p = \frac{1}{2}\left\| -\frac{(\boldsymbol{p_e}\boldsymbol{n}_\pi) + (\boldsymbol{n}_\pi \boldsymbol{p_e})}{2} - \boldsymbol{d}_\pi \right\|^2, \tag{6.24}$$

  where, $\boldsymbol{p_e}$ is the robot end-effector position in Cartesian space. The Jacobian parameter is obtained computing the time derivative of $\boldsymbol{h}_p$ depending on joint velocities:

$$\boldsymbol{J}_p = (\text{vec}_4 \boldsymbol{n}_\pi)^T \boldsymbol{J_{pos}}, \tag{6.25}$$

  where, $\boldsymbol{J_{pos}}$ is the translational part of the robot Jacobian.

The described tasks are considered to create three different evaluation tests, in which joint limits avoidance is always executed with the highest priority considering it as a safety-critical task:

Test 1 **Reach Target**: the robot tool tip is commanded to reach a specific target in position and orientation, avoiding joint limits during the movement. In this test, position and orientation tasks have the same priority.

Test 2 **Reach Target in Position with Plane Constraint**: the robot tool tip has to reach a specific target in position, avoiding joint limits and preventing it from crossing the lower bound of the workspace delimited by a plane. In this case, the position task has a lower priority compared to forbidden region avoidance. Therefore, the prioritization matrix is set as $\boldsymbol{K}(\boldsymbol{t}) = [0 \;\; -1 \;\; 1/k]$, with $k > 1$.

Test 3 **Reach Target in Position with Obstacle Avoidance**: the robot is commanded to reach a specific target in position, avoiding joint limits and preventing the robot from colliding with an obstacle. Position task and obstacle avoidance have different priority. In particular, if $d_{target} > d_l$, where d_{target} and d_l are the distance between the robot end-effector and target and the distance from the obstacle respectively, the prioritization matrix is defined as $\boldsymbol{K}(\boldsymbol{t}) = [0 \;\; -1 \;\; 1/k]$, with $k > 1$. Otherwise, when $d_{target} < d_l$, the matrix becomes $\boldsymbol{K}(\boldsymbol{t}) = [0 \;\; 1/k \;\; -1]$.

6.4.2 Simulation Test Results on dVRK Robot

Figure 6.1 shows consecutive frames of the "Reach Target" Test on dVRK in simulation. During this test, the target point is positioned ~ 0.03 m from the robot and its reference frame is rotated by 90° around z-axis with respect to the end-effector frame. The CBFs values are represented by $\boldsymbol{h}_{pos}$ and $\boldsymbol{h}_{ori}$. During the test, the Euclidean and the angular distance between the robot end-effector and the target reference frame are measured. In Fig. 6.1, $\boldsymbol{h}_{pos}$ and $\boldsymbol{h}_{ori}$ are driven to 0 executing the position and orientation tasks during the considered test, as confirmed by the computed distances from the target.

Figure 6.2 presents the "Reach Target with Plane Constraint" Test. The target point is placed ~ 0.02 m beyond the plane, which identifies a region that is ~ 0.08 m distant from the robot. The trends of CBFs values during task show that $\boldsymbol{h}_{pos}$ reaches a constant value, stopping at a distance of ~ 0.02 m from the target, while $\boldsymbol{h}_p$ goes to 0 showing that the robot does not reach the target point that is placed in the forbidden region.

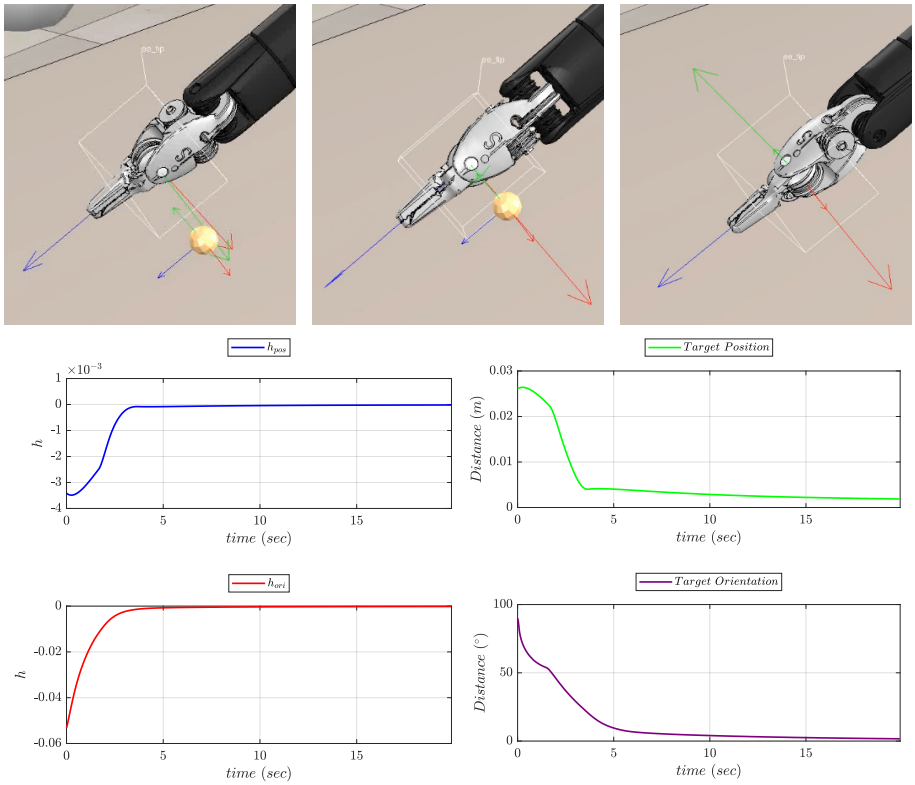


Figure 6.1: dVRK "Reach Target" Test in simulation: Blue line represents the CBF-function of position task, the red line is the CBF-function of orientation task, green line expresses the distance between the robot end-effector and target position, while the purple line is the angular distance between end-effector reference frame orientation and target reference frame orientation.

The "Reach Target with Obstacle Avoidance" Test is proposed in Fig. 6.3. In this case, the target point is located at a distance of ~ 0.04 from the obstacle. The CBF value $\boldsymbol{h}_{pos}$ goes to 0 executing the position task as justified by the decreasing distance from the target, while $\boldsymbol{h}_{obs}$ and the correspondent distance from the obstacle decreases as the robot moves closer to the object and then increases assuming a constant value when the robot approaches the target point.

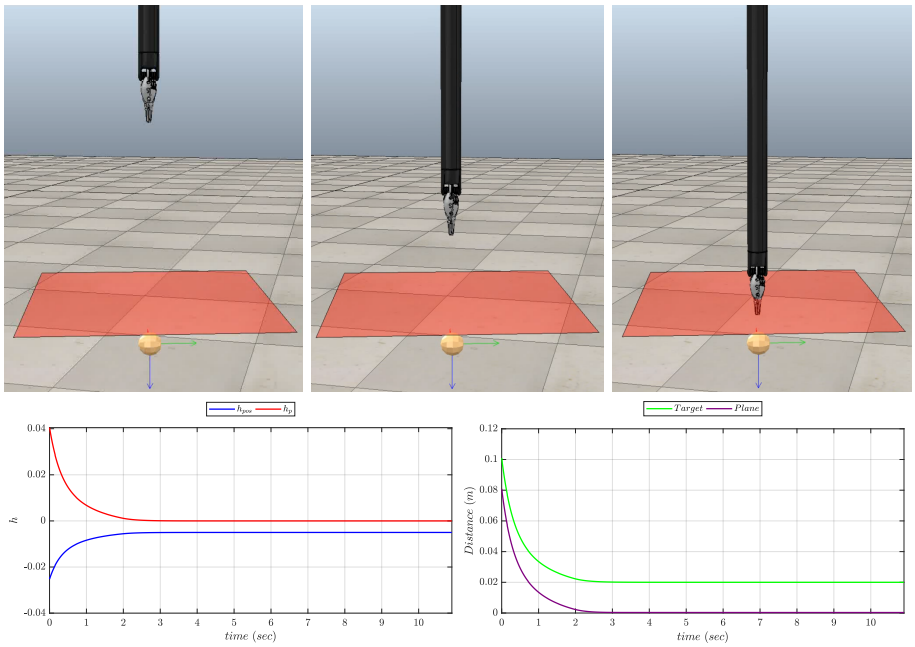


Figure 6.2: dVRK CBFs values in "Reach Target with Plane Constraint" Test in simulation. The blue line represents the CBF-function of the position task, while the red line is the CBF-function of the forbidden region avoidance task, the green line expresses the distance between the robot end-effector and target position, while the purple line is the distance between the end-effector and the plane.

6.4.3 Simulation Test Results on Symani®Surgical System

A "Reach Target" Test is also performed on Symani®Surgical System in the simulation environment, as shown in Fig. 6.4. Similarly to dVRK tests, different values are computed during the test, including CBFs functions $\boldsymbol{h}_{pos}$ and $\boldsymbol{h}_{ori}$, the Euclidean and angular distance between the robot end-effector and the target reference frame. Also in this case, the CBFs values $\boldsymbol{h}_{pos}$ and $\boldsymbol{h}_{ori}$ are guided to 0 executing the position and orientation tasks. The task accomplishment is justified by the distances trends from the target during the tests.

6.4.4 Test Results on Real dVRK Robot

The same tests presented in Section 6.4.2 are also performed on dVRK robot trying to replicate a real surgical scene. Figure 6.5 illustrates the "Reach Target Test" on real dVRK. Similar to the simulation tests, the target point is posi-

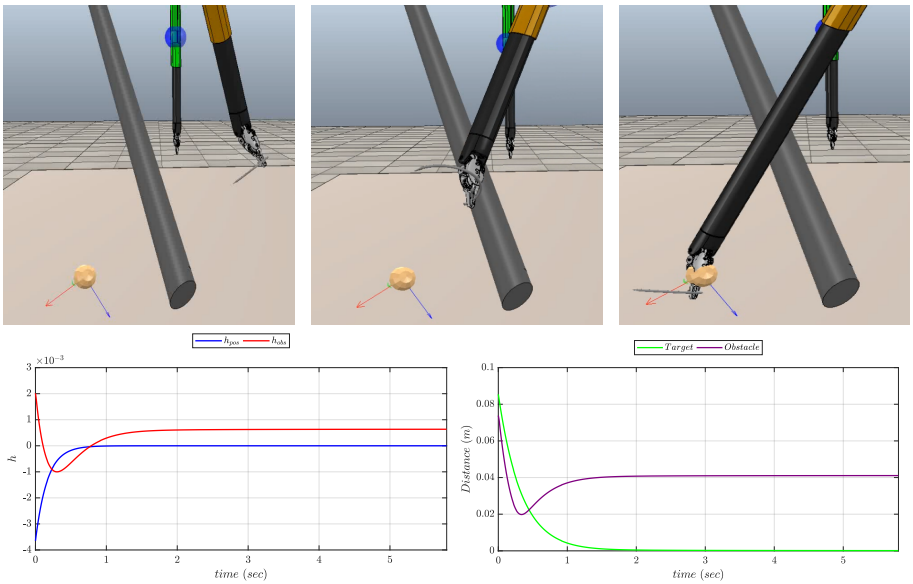


Figure 6.3: dVRK CBFs values of "Reach Target with Obstacle Avoidance" Test in simulation. The blue line represents the CBF-function of position task, while the red line is the CBF-function of obstacle avoidance task, the green line is the distance between the robot end-effector and target position, while the purple line is the distance between the end-effector and the obstacle axis.

tioned ~ 0.1 m from the robot and its reference frame is rotated by $90°$ around z-axis with respect to end-effector frame. The CBFs values ($\boldsymbol{h}_{pos}$, $\boldsymbol{h}_{ori}$), the Euclidean and angular distance between the robot end-effector and the target are measured. The CBFs values follow the same trends, rapidly going to 0 and accomplishing the position and orientation tasks, as shown by the correspondent distances from the target in Fig. 6.5.

The "Reach Target with Plane Constraint" Test is proposed in Fig. 6.6. In this case, the forbidden region is represented by a plane positioned at 0.08 m from the robot end-effector and the target point is on a plastic phantom, approximately 0.02 m beyond the plane. The $\boldsymbol{h}_{pos}$ function reaches a constant value, stopping at a distance of ~ 0.02 m from the target, while $\boldsymbol{h}_p$ goes to 0 showing that the robot does not reach the target point that is placed in the forbidden region.

Ultimately, Fig. 6.7 shows the "Reach Target with Obstacle Avoidance" Test on dVRK. In the test, the target point is located on a plastic phantom at a distance of ~ 0.02 from the obstacle, consisting of a cylinder that could rep-

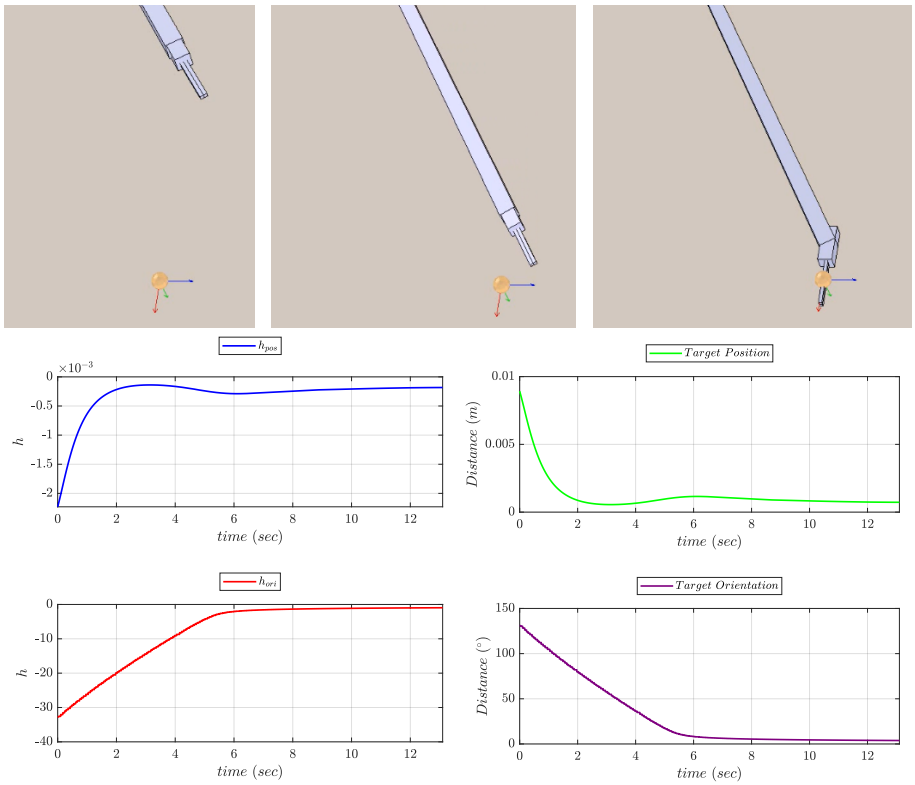


Figure 6.4: Symani CBFs values of "Reach target" Test in simulation. The blue line represents the CBF-function of the position task, while the red line is the CBF-function of the obstacle avoidance task, the green line is the distance between the robot end-effector and target position, while the purple line is the distance between the end-effector and the obstacle axis.

resent a soft tissue the robot is not allowed to touch during the procedure, for example, a blood vessel. Like in the other tests, CBF position value $\boldsymbol{h}_{pos}$ goes to 0 executing the task eliminating its distance from the target. Correspondingly, $\boldsymbol{h}_{obs}$ decreases as the robot moves closer to the object and the obstacle distance then increases assuming a constant value, showing that the robot tool tip correctly reaches the target point avoiding the obstacle.

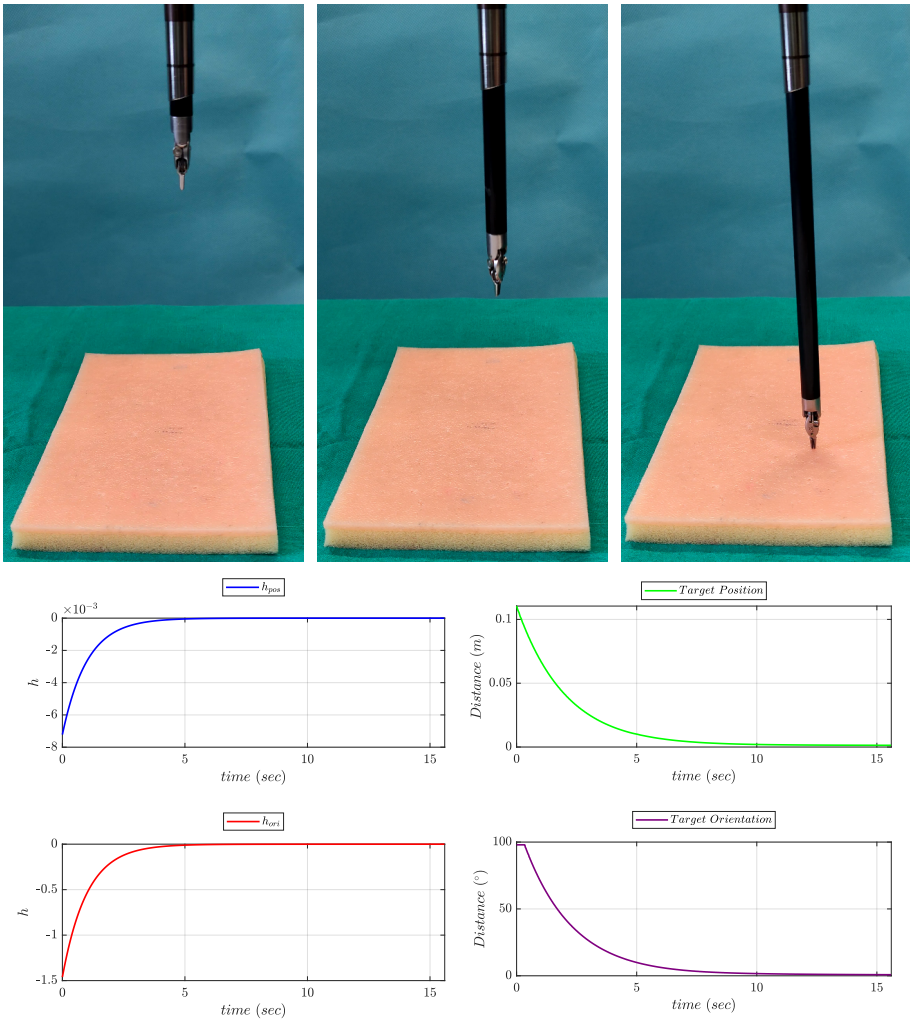


Figure 6.5: "Reach target" Test: Blue line represents the CBF-function of position task, the red line is the CBF-function of orientation task, the green line expresses the distance between the robot end-effector and target position, while the purple line is the angular distance between end-effector reference frame orientation and target reference frame orientation

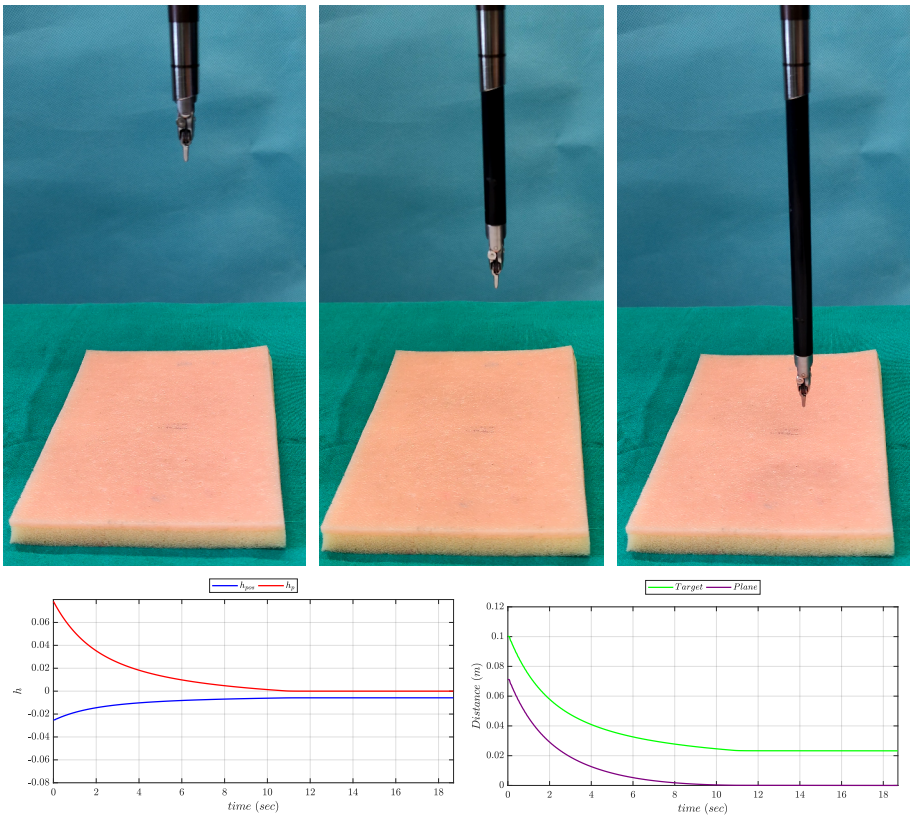


Figure 6.6: dVRK CBFs values in "Reach Target with Plane Constraint" Test. The blue line represents the h-function of position task, while the red line is the h-function of forbidden region avoidance task, the green line expresses the distance between the robot end-effector and target position, while the purple line is the distance between the end-effector and the plane.

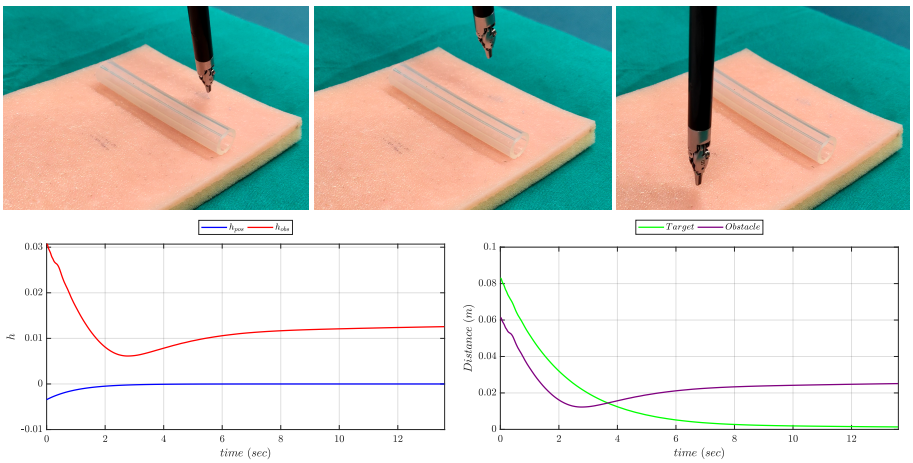


Figure 6.7: dVRK CBFs values of "Reach Target with Obstacle Avoidance" Test in simulation. The blue line represents the CBF-function of position task, while the red line is the CBF-function of obstacle avoidance task, the green line is the distance between the robot end-effector and target position, while the purple line is the distance between the end-effector and the obstacle axis.

6.5 Conclusion

The chapter presents a technique for multiple tasks execution on surgical robots that ensures an easier accomplishment of difficult surgical procedures, involving both guided and forbidden tasks. The proposed method uses optimization-based control and tasks definition and execution achieved using CBFs. DQ algebra allows a simple representation of the surgical scene with time-varying constraints that reflect environment change. Task prioritization ensures the possibility of autonomously switching the priority of the defined tasks during execution. The strategies are evaluated through multiple validation tests performed using two different robotic platforms, as dVRK robot for MIRS and Symani®Surgical System for microsurgery. The tests are achieved using CoppeliaSim physics simulators of the considered robotic systems and on real dVRK robot. The obtained results demonstrate the effectiveness of the method, showing the correct execution of the proposed tests.

The goal for future works is to use the proposed method to enhance teleoperation mode by coupling it with haptic feedback to the surgeon. The framework could be extended considering more complex tasks to execute, involving force contact between the robot and the tissues, showing the benefits of mul-

tiple tasks execution in a real surgical scenario. A preliminary extension is explained in Section 6.6. The advantage of multiple tasks definition as CBFs constraints could be a powerful method not only for a teleoperated robot but also towards autonomous task execution in robotic surgery.

6.6 Future Works

Feedback on interactions between tools and tissues still remains a crucial aspect for efficacy and safety in MIRS. The use of excessive force involuntary applied by the surgeon could create serious injuries to human tissues or damage the robot's tools. For this reason, the design of a robotic-assisted task that also involves physical interaction between the robot and environment becomes a necessary step to ensure safety and reactivity to the varying conditions of the surgical workspace. The multiple task execution framework based on CBFs could be adopted to create novel solutions for the creation of autonomous procedures in surgery. In this context, autonomous tissue scanning using da Vinci surgical robot with hybrid vision-force control is being developed in collaboration with University of Leeds, integrating different visual sensing technologies like ultra-sound images. A crucial step in the development of this project is represented by the formulation of an effective method for sensing the interaction force between the robot tools and the patient tissues. As exposed in Chapter 2, several works are available in the literature, most of all based on force measurements with no changes on robot structure, e.g using visual information or joint torques measurements. An accurate force estimation can be obtained using a sensor device presented in [3], integrated into da Vinci®robot instruments. This sensor is particularly suitable for this purpose since it does not require the modification of the surgical instruments and it is fully adaptable to different robot platforms. The proposed sensing device allows measuring only the interaction force components that lie in the plane orthogonal to the instrument's shaft, as shown in Fig. 6.8.

Accordingly, integration for the measurement of force component along insertion axis (z_e in Fig. 6.8) is needed. The method presented in [75, 76], already tested on da Vinci Robot MTM arm in [88, 100], could perform the required force estimation. This method could introduce a useful approach for sensor-less force estimation, only requiring joint positions and the applied torques without using acceleration measures that could add noisy terms to the estimated force signal. The method starts from the dynamic model of a n DoFs

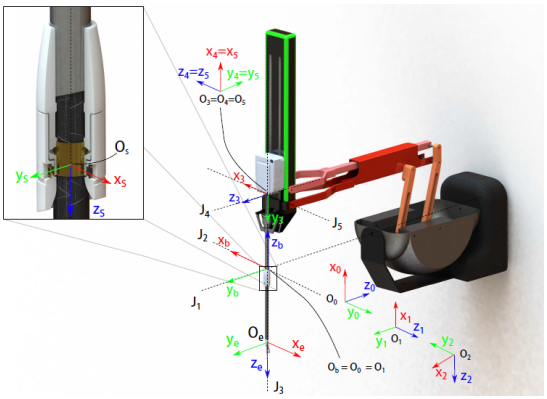


Figure 6.8: The dVRK robot PSM arm with force sensor [3].

manipulator:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} + \boldsymbol{\tau_{ext}}, \tag{6.26}$$

where $\boldsymbol{M}(\boldsymbol{q})$ is the inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ defines Coriolis and centrifugal elements, $\boldsymbol{g}$ is the gravity term, $\boldsymbol{\tau}$ is the control torque and $\boldsymbol{\tau_{ext}}$ describes the external torque due to contact forces applied on the robot. In this case, the contact point is assumed as coincident with the PSM end-effector and there is negligible friction at contact, i.e. the force is directed along the normal to the contact surface. In these assumptions, the external torque is computed as:

$$\boldsymbol{\tau_{ext}} = \boldsymbol{J_c^T}(\boldsymbol{q})\boldsymbol{F_c}, \tag{6.27}$$

where $\boldsymbol{F_c} \in \mathbb{R}^3$ represents the contact force and $\boldsymbol{J_c}$ is the Jacobian matrix related the velocity of contact point and joint velocity $\dot{\boldsymbol{q}}$. The method is based on robot actuator fault detection and isolation technique, that uses the generalized momentum of the robot $\boldsymbol{p} = \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}$ to compute a vector called residual, defined as:

$$\boldsymbol{r} = \boldsymbol{K_I}\left(\boldsymbol{p} - \int_0^t (\boldsymbol{\tau} + \boldsymbol{C}^T(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{r})\, ds\right), \tag{6.28}$$

where $\boldsymbol{K_I} \geq 0$ is a diagonal gain matrix. The dynamic evolution of $\boldsymbol{r}$ defined as $\dot{\boldsymbol{r}} = \boldsymbol{K_I}(\boldsymbol{\tau_{ext}} - \boldsymbol{r})$, becomes for sufficiently large gain:

$$\boldsymbol{r} \simeq \boldsymbol{\tau_{ext}}. \tag{6.29}$$

Thus, the external force could be estimated by:

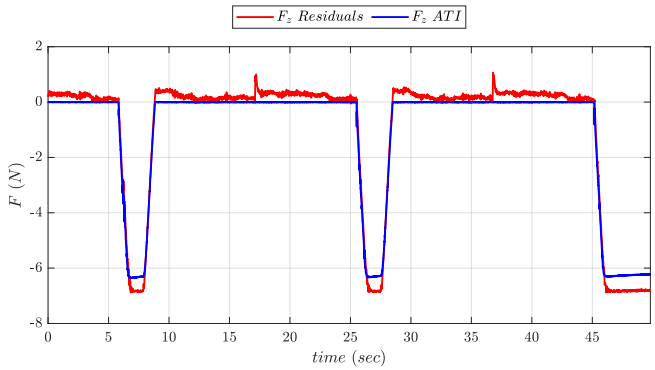


Figure 6.9: Estimated force along the insertion axis. The blue line is the force measured on the ATI sensor, and the red line represents the estimated component.

$$\hat{\boldsymbol{F}}_{\boldsymbol{c}} = \left(\boldsymbol{J}_{\boldsymbol{c}}^{T}(\boldsymbol{q})\right)^{*}\boldsymbol{r}, \tag{6.30}$$

where $\left(\boldsymbol{J}_{\boldsymbol{c}}^{T}(\boldsymbol{q})\right)^{*}$ is the pseudo-inverse matrix of the transposed Jacobian. Therefore, this approach provides a useful force estimation practice based on the computation of a single residual vector, which requires only the robot dynamic model. Hence, the correct estimation of the force contact on the insertion axis is related to the dynamic parameters identification of the da Vinci®robot. In this work, an approach similar to [55], presented in Chapter 2, is considered. In particular, the friction contribution is:

$$\boldsymbol{\tau_f} = \boldsymbol{F_v}\dot{\boldsymbol{q}} + \boldsymbol{F_c}\, sgn(\dot{\boldsymbol{q}}), \tag{6.31}$$

where $\dot{\boldsymbol{q}}$ is the vector of joints velocities, $\boldsymbol{F_v}$ and $\boldsymbol{F_c}$ are diagonal matrices representing the viscous and Coulomb friction constants. An alternatives to Coulomb friction with Stribeck effect model $\boldsymbol{F_c}\, sgn(\dot{\boldsymbol{q}})$ is also implemented, using hyperbolic tangent model $\boldsymbol{F_c}\, tanh(\dot{\boldsymbol{q}})$.

The force estimation method is evaluated through simple tests, in which the PSM arm is controlled to autonomously interact with a phantom tissues. The proposed method estimates the force on z-axis, comparing it with the force measured using a commercial force-torque sensor ATI Mini 45 positioned on the phantom. During the test, only the prismatic joint of PSM arm is considered and the robot moves vertically to reach several times the tissue. Figure 6.9 presents the estimated and the measured force trends, showing that the exposed method allows a sufficiently accurate estimation with a mean absolute error of 0.2660 ± 0.0024 N.

Chapter 7

Conclusions

This thesis addressed the problem of creating reliable solutions to enhance the quality of intervention in surgical robotic procedures. Modern teleoperated robots notably improve surgeon abilities compared to traditional open surgery and LS, but critical tasks mostly rely on surgeon's ability. These robotic systems are directly controlled by the surgeon and basic improvements mainly regard tremor filtering and motion scaling. For these reasons, more advanced control algorithms and methods show great potential to be implemented on surgical robotic platforms, enhancing teleoperation in a shared control paradigm or creating autonomous tasks. Moreover, the perception phase could also be reinforced using new computer vision techniques to obtain a more accurate geometrical description of the surgical site.

After an introduction of the surgical robotics scenario and review of the state-of-the-art methods given in Chapter 2, aiming to draw attention to the description of novel surgical robotic platforms, autonomous and shared control algorithms, and vision assisted applications, contributions of the thesis were presented. The first two scientific contributions concern assistive methods based on haptic-guided shared control applications on specific surgical procedures. The first work is presented in Chapter 3, focusing on the dissection and cutting task of precancerous growths, called polyps. In this case, vision and control enhancements were aimed at redefining each step of a specific surgical procedure to assist the surgeon with a functioning pipeline. The work proposed a shared control method based on GVFs in which computer vision algorithms are employed to detect the region of intervention. An overview of the traditional dissection procedures was provided, the control architecture design was explained in detail and experimental tests were reported to show

the performance improvements compared to the non-robotic procedure. Chapter 4 concerns another assistive method for suturing that represents the most difficult task in surgery. Like the previous work, the design regards a haptic-guided shared control application, focusing on the detection part proposing a tracking method of suturing needle. The grasping pose was obtained by a standard gradient descent optimization routine, that yielded neither joint limits nor singularities during post-grasp suturing trajectory execution. Despite positive results that justify the development of those methods, some limitations are still present. In particular, the use of basic computer vision concepts represents only a first step for a correct region of interest identification. More advanced vision techniques, e. g. based on Machine Learning approach, could allow the application of this method to the real surgical site. Moreover, the assistance provided by GVFs could become customized to be suitable for a different level of assistance; for example, a novice surgeon could prefer harder guidance instead of a softer one that is more desirable for the expert surgeons.

The work presented in Chapter 5 introduced a method that can help the surgeon in a generic sub-task that frequently occurs in all the surgical procedures. Collision avoidance of instruments tools could prevent serious damages and notably reduce the surgeon workload, which is relieved from continuously paying attention to the tool distance to avoid clashing. The work presented in this chapter adopted a marker-less surgical tool tracking based on EKF filter and the detection part was executed by a deep learning algorithm for instrument semantic segmentation, based on a fully CNN. Differently from the other proposed strategies, this approach was validated through experiments that involved novice and expert surgeons, showing good results in improving performances. In all the presented works, the robotic system is controlled by the surgeon while receiving haptic cues that help to better perform the task. The human factor is a central aspect to assess the feasibility and repeatability of the proposed methods.

Unlike the previous contributions, the achievements in Chapter 6 represented a generic framework to enhance teleoperation in multiple tasks, and at the same time was a first step to go towards autonomous task execution in robotic surgery. A series of basic actions composed a single surgical gesture, executed simultaneously by the framework with different priorities. Nonetheless, all the considered sub-tasks were defined in joint or Cartesian space and described accomplishments that did not involve any contact force exchanged between the robot and the surrounding environment. A novel method for force estimation is strictly needed to create more complex tasks to accomplish.

Bibliography

[1] A. Attanasio, B. Scaglioni, E. De Momi, P. Fiorini, and P. Valdastri, "Autonomy in Surgical Robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 16, no. 16, pp. 4.1–4.29, 2021.

[2] G.-Z. Yang, J. Cambias, K. Cleary, E. Daimler, and J. Drake, et al, "Medical robotics regulatory, etichal, and legal considerations for increasing levels of autonomy," *Science Robotics*, vol. 2, no. 4, March 2017.

[3] G. A. Fontanelli, L. R. Buonocore, F. Ficuciello, L. Villani, and B. Siciliano, "A Novel Force Sensing Integrated into the Trocar for Minimally Invasive Robotic Surgery," *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 131–136, 2017.

[4] A. Arezzo, R. Passera, G. Scozzari, M. Verra, and M. Morino, "Laparoscopy for rectal cancer reduces short-term mortality and morbidity: results of a systematic review and meta-analysis," *Surgical endoscopy*, vol. 27, no. 5, pp. 1485–1502, May 2013.

[5] M. Ferlitsch, A. Moss, and A. Arezzo et al, "Colorectal polypectomy and endoscopic mucosal resection (EMR): European Society of Gastrointestinal Endoscopy (ESG) Clinical Guideline," *Endoscopy*, vol. 49, no. 3, pp. 270–297, 2017.

[6] S. Tyagi and S. Kumar, "Microsurgery: an important tool for reconstructive surgery - a review," *International Journal of Pharma and Bio Sciences*, vol. 1, no. 3, pp. 1–11, 2010.

[7] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. New York, NY, USA: Springer-Verlag, 2009.

[8] G. Niemeyer, C. Preusche, S. Stramigioli, and D. Lee, *Telerobotics*. Springer International Publishing, 2016, pp. 1085–1108.

[9] R. Taylor, J. Funda, B. Eldridge, S. Gomory, K. Gruben, D. LaRose, M. Talamini, L. Kavoussi, and J. Anderson, "A telerobotic assistant for laparoscopic surgery," *IEEE Engineering in Medicine and Biology Magazine*, vol. 14, pp. 279–288, 1995.

[10] S. Charles, H. Das, T. Ohm, C. Boswell, G. Rodriguez, R. Steele, and D. Istrate, "Dexterity-enhanced telerobotic microsurgery," *Proc. International Conference on Advanced Robotics*, pp. 5–10, 1997.

[11] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. New York, NY, USA: Springer-Verlag, 2007.

[12] T. Ortmaier, M. Groger, D. H. Boehm, V. Falk, and G. Hirzinger, "Motion estimation in beating heart surgery," *IEEE Transactions on Biomedical Engineering*, vol. 52, pp. 1729–1740, October 2005.

[13] L. B. Rosenberg, "Virtual fixtures: Perceptual tools for telerobotic manipulation," *Proc. IEEE Virtual Reality Annual International Symposium*, pp. 76–82, 1993.

[14] S. A. Bowyer, B. L. Davies, and F. R. Baena, "Active constraints/Virtual fixtures: A survay," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 138–157, 2014.

[15] N. Enayati, E. D. Momi, and G. Ferrigno, "Haptics in robot-assisted surgery: Challenges and benefits," *IEEE Reviews in Biomedical Engineering*, vol. 9, pp. 49–65, 2016.

[16] M. A. Vitrani, C. Poquet, and G. Morel, "Applying virtual fixtures to the distal end of a minimally invasive surgery instrument," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 114–123, 2017.

[17] M. Selvaggio, G. A. Fontanelli, F. Ficuciello, L. Villani, and B. Siciliano, "Passive Virtual Fixtures adaptation in minimally invasive robotic surgery," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3129–3136, 2018.

[18] M. Li, M. Ishii, and R. H. Taylor, "Spatial Motion Constraints Using Virtual Fixtures Generated by Anatomy," *IEEE Transaction on Robotics*, vol. 23, no. 1, pp. 4–19, 2007.

[19] A. Banach, K. Leibrandt, M. Grammatikopoulou, and G.-Z. Yang, "Active Constraints for Tool-Shaft Collision Avoidance in Minimally Invasive Surgery," *Proc. IEEE International Conference on Robotics and Automation*, pp. 1556–1562, 2019.

[20] K.-W. Kwok, G. P. Mylonas, L. W. Sun, M. Lerotic, J. Clark, T. Athanasiou, A. Darzi, and G.-Z. Yang, "Passive Virtual Fixtures adaptation in minimally invasive robotic surgery," *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 410–417, 2018.

[21] E. Lopez, L. Zollo, and E. Guglielmelli, "Teleoperated control based on virtual fixtures for a redundant surgical system," *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 450–455, 2013.

[22] B. V. Adorno, "A Unified Framework for the Teleoperation of Surgical Robots in Constrained Workspaces," *Proc. IEEE International Conference on Robotics and Automation*, pp. 2721–2727, 2019.

[23] M. M. Marinho, B. V. Adorno, k. Harada, and M. Mitsuishi, "Dynamic Active Constraints for Surgical Robots using Vector Field Inequalities," *IEEE Transactions on Robotics*, vol. 35, no. 5, pp. 1166–1185, 2019.

[24] A. Leporini, E. Oleari, C. Landolfo, A. Sanna, A. Larcher, G. Gandaglia, N. Fossati, F. Muttin, U. Capitanio, F. Montorsi, A. Salonia, M. Minelli, F. Ferraguti, C. Secchi, S. Farsoni, A. Sozzi, M. Bonfè, N. Sayols, A. Hernansanz, A. Casals, S. Hertle, F. Cuzzolin, A. Dennison, A. Melzer, G. Kronreif, S. Siracusano, F. Falezza, F. Setti, and R. Muradore, "Technical and Functional Validation of a Teleoperated Multirobots Platform for Minimally Invasive Surgery," *IEEE Transactions on Medical Robotics and Bionics*, vol. 2, no. 2, pp. 148–156, 2020.

[25] A. Sozzi, M. Bonfè, S. Farsoni, G. DeRossi, and R. Muradore, "Dynamic Motion Planning for Autonomous Assistive Surgical Robots," *Electronics 2019*, vol. 8, no. 9, pp. 957–981, 2020.

[26] M. M. Marinho, B. V. Adorno, k. Harada, and M. Mitsuishi, "Active constraints using vector field inequalities for surgical robots," *Proc. IEEE International Conference on Robotics and Automation*, pp. 5364–5371, 2018.

[27] Z. Chen, A. Malpani, A. Chalasani, S. Duguet, S. Vedula, P. Kazanzides, and R. H. Taylor, "Virtual fixture assistance for needle passing and knot tying," *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2343–2350, October 2016.

[28] T. Liu and M. C. Cavusoglu, "Needle grasp and entry port selection for automatic execution of suturing tasks in robotic minimally invasive surgery," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 552–563, 2016.

[29] S. Sen, A. Garg, D. V. Gealy, S. McKinley, Y. Jen, and K. Goldberg, "Automating multi-throw multilateral surgical suturing with a mechanical needle guide and sequential convex optimization," pp. 4178–4185, 2016.

[30] G. A. Fontanelli, Y. G.-Z., and B. Siciliano, "A Comparison of Assistive Methods for Suturing in MIRS,," *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4389–4395, 2018.

[31] D. Stoyanov, "Surgical Vision," *Annals of Biomedical Engineering*, vol. 40, no. 2, pp. 332–345, February 2012.

[32] B. C. Becker, R. A. MacLachlan, G. D. Hager, and C. N. Reviere, "Handheld micromanipulation with vision-based virtual fixtures," *Proc. IEEE International Conference on Robotics and Automation*, pp. 4127–4132, 2011.

[33] B. C. Becker, R. A. MacLachlan, L. A. Lobes, G. D. Hager, and C. N. Reviere, "Vision-based control of a handheld surgical micromanipulator with virtual fixtures," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 674–683, 2013.

[34] H. Moradi, S. Tang, and S. E. Salcudean, "Toward Robot-Assisted Photoacoustic Imaging: Implementation Using the da Vinci Research Kit and Virtual Fixtures," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1807–1814, 2019.

[35] P. Mountney, D. Stoyanov, and G.-Z. Yang, "Three-Dimensional Tissue Deformation Recovery and Tracking," *IEEE Signal Processing Magazine*, vol. 27, no. 4, pp. 14–24, July 2010.

[36] F. Richter, R. K. Orosco, and M. C. Yip, "Open-Sourced Reinforcement Learning Environments for Surgical Robotics," *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019.

[37] A. Marbana, V. Srinivasan, W. Samek, J. Fernandez, and A. Casals, "A Recurrent Convolutional Neural Network Approach for Sensorless Force Estimation in Robotic Surgery," *Proc. IEEE International Conference on Image Processing Theory, Tools and Applications*, pp. 1–6, 2014.

[38] M. Ye, L. Zhang, S. Giannarou, and G.-Z. Yang, "Real-time 3d tracking of articulated tools for robotic surgery," *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2016.

[39] R. Hao, R. O. Ozguner, and M. C. Cavusoglu, "Vision-Based Surgical Tool Pose Estimation for the da Vinci Robotic Surgical System," *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1298–1305, 2018.

[40] E. Colleoni, S. Moccia, X. Du, E. De Momi, and D. Stoyanov, "Deep Learning Based Robotic Tool Detection and Articulation Estimation with spatio-Temporal Layers," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2714–2721, 2019.

[41] B. B. Haro, L. Zappella, and R. Vidal, "Surgical gesture classification from video data," *Medical Image Computing and Computer-Assisted Intervention MICCAI 2012*, pp. 34–41, 2012.

[42] T. Beyl, P. Nicolai, M. D. Comparetti, J. Raczkowsky, E. De Momi, and H. Wörn, "Time-of-flight-assisted kinect camera-based people detection for intuitive human robot cooperation in the surgical operating room," *International journal of computer assisted radiology and surgery*, vol. 11, pp. 1329–1345, 2016.

[43] E. Mazomenos, D. Watson, R. Kotorov, and D. Stoyanov, "Gesture classification in robotic surgery using recurrent neural networks with kinematic information," *Proc. 8th Joint Workshop on New Technologies for Computer/Robotic Assisted Surgery*, 2018.

[44] S. A. Pedram, P. Ferguson, J. Ma, E. Dutson, and R. J., "Autonomous suturing via surgical robot: an algorithm for optimal selection of needle diameter, shape, and path," *Proc. IEEE International Conference on Robotics and Automation*, pp. 2391–2398, 2017.

[45] C. Staub, T. Osa, A. Knoll, and B. R., "Automation of tissue piercing using circular needles and vision guidance for computer aided laparoscopic surgery," *Proc. IEEE International Conference on Robotics and Automation*, pp. 4585–4590, 2010.

[46] H. Mayer, F. Gomez, D. Wierstra, I. Nagy, A. Knoll, and S. J., "Automation of tissue piercing using circular needles and vision guidance for computer aided laparoscopic surgery," *Advanced Robotics*, vol. 22, pp. 1521–1537, 2008.

[47] C. D'Ettore, G. Dwyer, F. Chadebecq, F. Vasconcelos, E. De Momi, and D. Stoyanov, "Automated Pick-up of Suturing Needles for Robotic Surgical Assistance," *Proc. IEEE International Conference on Robotics and Automation*, pp. 1370–1377, 2018.

[48] Z. Y. Chiu, F. Richter, E. K. Funk, R. K. Orosco, and M. C. Yip, "Bimanual Regrasping for Suture Needles using Reinforcement Learning for Rapid Motion Planning," *arXiv:2011.04813*, 2020.

[49] H. Saeidi, H. N. D. Le, J. D. Opfermann, S. Leonard, A. Kim, M. H. Hsieh, J. U. Kang, and A. Krieger, "Autonomous laparoscopic robotic suturing with a novel actuated suturing tool and 3d endoscope," *Proc. IEEE International Conference on Robotics and Automation*, pp. 1541–1547, 2019.

[50] W. Qiu, J. Yuan, E. Ukwatta, Y. Sun, M. Rajchl, and A. Fenster, "Prostate segmentation: an efficient convex optimization approach with axial symmetry using 3-D TRUS and MR images," *IEEE Transactions on Medical Imaging*, vol. 33, pp. 947–960, 2014.

[51] X. Li, H. Chen, X. Qi, Q. Dou, C. W. Fu, and P. Heng, "H-Dense UNet: hybrid densely connected UNet for liver and tumor segmentation from CT volumes," *IEEE Transactions on Medical Imaging*, vol. 37, pp. 2663–2674, 2018.

[52] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da Vinci®Surgical System," *Proc. IEEE International Conference on Robotics and Automation*, pp. 6434–6439, 2014.

[53] dVRK Wiki. [Online]. Available: http://research.intusurg.com/dvrkwiki

[54] Robot opearting system (ROS) wiki Wiki. [Online]. Available: https://www.ros.org/

[55] G. A. Fontanelli, F. Ficuciello, L. Villani, and B. Siciliano, "Modelling and identification of the da Vinci Research Kit robotic arms," *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1464–1469, 2017.

[56] G. A. Fontanelli, M. Selvaggio, M. Ferro, F. Ficuciello, M. Vendittelli, and B. Siciliano, "A V-REP Simulator for the da Vinci Research Kit robotic Platform," *Proc. in 7th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, pp. 1056–1061, 2018.

[57] M. Selvaggio, G. A. Fontanelli, V. R. Marrazzo, U. Bracale, A. Irace, G. Breglio, L. Villani, B. Siciliano, and F. Ficuciello, "The MUSHA underactuated hand for robot-aided minimally invasive surgery," *The International Journal of Medical Robotics and Computer Assisted Surgery*, p. e1981, 2018.

[58] H. Liu, M. Selvaggio, P. Ferrentino, R. Moccia, S. Pirozzi, U. Bracale, and F. Ficuciello, "The MUSHA hand II: a multi-functional hand for robot-assisted laparoscopic surgery," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 1, pp. 393–404, February 2020.

[59] S. Saini, F. Ficuciello, H. Liu, and U. Bracale, "Elemento terminale per dispositivi di presa per interventi chirurgici, in particolare interventi a minima invasività," Patent, Patent granted n. 102019000001187, application submission date Jan 2019.

[60] H. Liu, F. Ficuciello, S. Saini, G. A. Fontanelli, M. Selvaggio, and U. Bracale, "Dispositivo di ausilio per interventi chirurgici, in particolare interventi a minima invasività," Patent, Patent granted n. 102019000001185, application submission date Jan 2019.

[61] E. Rohmer, S. P. N. Singh, and M. Freese, "CoppeliaSim (formerly V-REP): a versatile and scalable robot simulation framework, `www.coppeliarobotics.com`," *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1321–1326, November 2013.

[62] M. Ferro, D. Brunori, F. Magistri, L. Saiella, M. Selvaggio, and G. A. Fontanelli, "A portable da vinci simulator in virtual reality," *Proc. in 3rd IEEE International Conference on Robotic Computing*, pp. 447–448, 2019.

[63] G. A. Fontanelli, M. Selvaggio, M. Ferro, F. Ficuciello, M. Vendittelli, and B. Siciliano, "Portable dVRK: an augmented V-REP simulator of da Vinci Research Kit," *Acta Polytechnica Hungarica*, vol. 16, no. 8, pp. 79–98, 2019.

[64] A. Patriti, G. Ceccarelli, R. Bellochi, A. Bartoli, A. Spaziani, L. Di Zitti, and L. Casciola, "Robot-assisted laparoscopic total and partial gastric resection with D2 lymph node dissetion for adenocarcinoma," *Surgical Endoscopy*, vol. 22, no. 12, pp. 2753–2760, 2008.

[65] J. Bernal and D. Stoyanov et al., "Comparative validation of polyp detection methods in video colonoscopy: Results from the MICCAI 2015 endoscopic vision challenge," *IEEE Transactions On Medical Imaging*, vol. 36, no. 6, pp. 1231–1249, 2017.

[66] S. Hwang, "Bag-of-visual-words approach to abnormal image detection in wireless capsule endoscopy videos," in *Advances in Visual Computing*. Germany: Springer: Berlin/Heidelberg, 2011, pp. 320–327.

[67] H. Zhu, Y. Fan, and Z. Liang, "Improved curvature estimation for shape analysis in computer-aided detection of colonic polyps," *Proc. International Workshop on Computational Challenges and Clinical Opportunities in Virtual Colonoscopy and Abdominal Imaging*, pp. 9–14, 2010.

[68] S. A. Karkanis, D. K. Iakovidis, D. E. Maroulis, D. A. Karras, and M. Tzivras, "Computer-aided tumor detection in endoscopic video using color wavelet features," *IEEE Transactions on Information Technology in Biomedicine*, vol. 7, no. 3, pp. 141–152, 2003.

[69] S. Hwang and M. E. Celebi, "Polyp detection in wireless capsule endoscopy videos based on image segmentation and geometric feature,"

Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 678–681, 2010.

[70] E. Riberio, A. Uhl, and M. Hafner, "Colonic polyp classification with convolutional neural networks," *Proc. IEEE 29th International Symposium on Computer-Based Medical Systems (CBMS)*, pp. 253–258, 2016.

[71] A. Bettini, P. Marayong, S. Lang, A. M. Okamura, and G. D. Hager, "Vision-assisted control for manipulation using virtual fixtures," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 953–966, 2004.

[72] F. Rydén and H. J. Chizeck, "Forbidden-region virtual fixtures from streaming point clouds: Remotely touching and protecting a beating heart," *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3308–3313, 2012.

[73] T. Yamamoto, N. Abolhassani, S. Jung, A. M. Okamura, and T. N. Judkins, "Augmented reality and haptic interfaces for robot-assisted surgery," *International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 8, no. 1, pp. 45–56, 2012.

[74] R. Moccia, M. Selvaggio, B. Siciliano, A. Arezzo, and F. Ficuciello, "Vision-based Virtual Fixtures generation for MIRS dissection tasks," *9th Joint Workshop on New Technologies for Computer/Robot Assisted Surgery*, 2019.

[75] A. De Luca and R. Mattone, "Actuator failure detection and isolation using generalized momenta," *Proc. IEEE International Conference on Robotics and Automation*, pp. 634–639, 2003.

[76] A. De Luca and R. Mattone, "Sensorless robot collision detection and hybrid force/motion control," *Proc. IEEE International Conference on Robotics and Automation*, pp. 999–1004, 2005.

[77] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," *Proc. 7th IEEE International Conference on Computer Vision*, vol. 1, pp. 666–673, 1999.

[78] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A*, vol. 4, no. 4, pp. 629–642, 1987.

[79] C. Rother, V. Kolmogorov, and A. Blake, "Grubcut: Interactive foregroud extraction using graph cuts," *ACM Transactions on Graphics*, vol. 4, pp. 309–314, 2004.

[80] A. Petit, V. Lippiello, and B. Siciliano, "Real-time tracking of 3d elastic objects with RGB-D sensor," *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3914–3921, 2015.

[81] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[82] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," *Proc. International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, vol. 2, pp. 331–340, 2009.

[83] G. A. Fontanelli, M. Selvaggio, L. R. Buonocore, F. Ficuciello, L. Villani, and B. Siciliano, "A new laparoscopic tool with in-hand rolling capabilities for needle reorientation," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2354–2361, 2018.

[84] F. Zhong, D. Navarro-Alarcon, Z. Wang, Y. Liu, T. Zhang, H. M. Yip, and H. Wang, "Adaptive 3D pose computation of suturing needle using constraints from static monocular image feedback," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5521–5526, 2016.

[85] M. Ferro, G. A. Fontanelli, F. Ficuciello, B. Siciliano, and M. Vendittelli, "Vision-based suturing needle tracking with extended kalman filter," *7th Joint Workshop on New Technologies for Computer/Robot Assisted Surgery*, 2017.

[86] M. E. Hagen, J. J. Meehan, I. Inan, and P. Morel, "Visual clues act as a substitute for haptic feedback in robotic surgery," *Surgical Endoscopy*, vol. 22, no. 6, pp. 1505–1508, 2008.

[87] J. K. Koehn and K. J. Kuchenbecker, "Surgeons and non-surgeons prefer haptic feedback of instrument vibrations during robotic surgery," *Surgical Endoscopy*, vol. 29, no. 10, pp. 2970–2983, 2015.

[88] R. Moccia, M. Selvaggio, L. Villani, B. Siciliano, and F. Ficuciello, "Vision-based Virtual Fixtures Generation for Robotic-Assisted Polyp Dissection Procedures," *Proc. IEEE International Conference on Intelligent Robots and Systems*, pp. 7928–7933, 2019.

[89] J. Ren, R. V. Patel, K. A. McIsaac, G. Guiraudon, and T. M. Peters, "Dynamic 3-D Virtual Fixtures for Minimally Invasive Beating Heart Procedures," *IEEE Transactions on Medical Imaging*, vol. 27, no. 8, pp. 1061–1070, 2008.

[90] T. Xia, C. Baird, G. Jallo, K. Hayes, N. Nakajima, N. Hata, and P. Kazanzides, "An integrated system for planning, navigation and robotic assistance for skull base surgery," *International Journal of Medical Robot*, vol. 4, no. 4, pp. 321–330, 2008.

[91] D. Bouget, M. Allan, D. Stoyanov, and P. Jannin, "Vision-based and marker-less surgical tool detection and tracking: a review of the literature," *Medical Image Analysis*, vol. 35, pp. 633–654, 2017.

[92] S. Bodenstedt and *et al.*, "Comparative evaluation of instrument segmentation and tracking methods in minimally invasive surgery," *Medical Image Analysis, arXiv:1805.02475*, 2018.

[93] S. Allan, M. Thompson, M. J. Clarkson, S. Ourselin, D. J. Hawkes, J. Kelly, and D. Stoyanov, "2D-3D Pose Tracking of Rigid Instruments in Minimally Ivasive Surgery," *Information Processing in Computer-Assisted Interventions*, vol. 8498, pp. 1–10, 2014.

[94] X. Du, M. Allan, A. Dore, S. Ourselin, D. Hawkes, J. D. Kelly, and D. Stoyanov, "Combined 2D and 3D tracking of surgical instruments for minimally invasive and robotic-assisted surgery," *International Journal of Computer Assisted Radiology and Surgery*, vol. 11, no. 6, pp. 1109–1119, 2016.

[95] X. Du, T. Kurmann, P.-L. Chang, S. Allan, M. Ourseln, R. Sznitman, J. D. Kelly, and D. Stoyanov, "Articulated Multi-Instrument 2-d Pose Estimation Using Fully Convolutional Networks," *IEEE Transaction on Medical Imaging*, vol. 37, no. 5, pp. 1276–1287, 2018.

[96] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *Proc. International Con-*

ference on Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015, pp. 234–241, 2015.

[97] A. A. Shvets, A. Rakhlin, A. A. Kalinin, and V. I. Iglovikov, "Automatic Instrument Segmentation in Robot-Assisted Surgery using Deep Learning," *Proc. 17th IEEE International Conference on Machine Learning and Applications*, pp. 624–628, 2018.

[98] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An Accurate O(n) Solution to the PnP Problem," *International Journal of Computer Vision*, vol. 81, no. 2, p. 155, 2008.

[99] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," *Advances in Psychology*, vol. 52, pp. 139–183, 1988.

[100] R. Moccia, C. Iacono, B. Siciliano, and F. Ficuciello, "Vision-Based Dynamic Virtual Fixtures for Tools Collision Avoidance in Robotic Surgery," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1650–1655, June 2020.

[101] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control Barrier Functions: Theory and Applications," *Proc. 18th European Control Conference (ECC)*, pp. 3420–3431, 2019.

[102] B. V. Adorno and M. M. Marinho, "DQ Robotics: a Library for Robot Modeling and Control," *IEEE Robotics & Automation Magazine*, pp. 1–15, 2020.

[103] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," *Proc. IEEE Conference on Decision and Control*, pp. 6271–6278, 2014.

[104] X. D. Xu, J. W. Tabuada, P. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *Proc. IEEE Conference on Decision and Control*, vol. 48, no. 27, pp. 54–61, 2015.

[105] G. Notomista, S. Mayya, M. Selvaggio, M. Santos, and C. Secchi, "A Set-Theoretic Approach to Multi-Task Execution and Prioritization," *Proc. IEEE International Conference on Robotics and Automation*, 2020.

[106] B. V. Adorno, "Two-arm Manipulation, From Manipulators to Enhanced Human-Robot Collaboration," *PhD Dissertation, Université Montpellier 2*, 2011.

[107] W. R. Hamilton, *Elements of Quaternions*. London, UK: Longmans, Green and Company, 1866.

[108] S. Moe, G. Antonelli, R. Teel, K. Y. Petterson, and J. Schrimpf, "Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: General formulation, stability analysis, and experimental results," *Frontiers in Robotics and AI*, vol. 3, no. 16, pp. 1–16, 2016.