TESI DI DOTTORATO

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

DIPARTIMENTO DI INGEGNERIA ELETTRONICA
E DELLE TECNOLOGIE DELL' INFORMAZIONE

DOTTORATO DI RICERCA IN
INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

SHARED CONTROL TELEROBOTIC METHODS FOR INDUSTRIAL AND SURGICAL ROBOTIC SYSTEMS

MARIO SELVAGGIO

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

A. A. 2018–2019

to my parents

Acknowledgments

First and foremost, I want to express my deepest and most sincere gratitude to my advisor Prof. Bruno Siciliano for mentoring the research activities which lead to this dissertation and, more in general, for his outstanding and invaluable support during these last three years. Most of all, I would like to thank him for his unconditional trust and freedom that allowed me to pursue any research direction I found exciting and relevant. I have known him since I was a grad student, and I have always appreciated his scientific rigor, as well as his kindness and dedication. When I joined the Ph.D. school and his group, I still remember one of his most valuable advice for being successful in research: that of always keeping curiosity alive.

I have also to thank all the other professors of the PRISMA Lab, in particular Prof. Luigi Villani and Prof. Fanny Ficuciello whose support and advise were essential for the development of the work behind this thesis. A special thank goes to my colleague Giuseppe Andrea Fontanelli who always provided ground for deeper reflections and considerations about research in general.

During the last three years, I have had the amazing opportunity of spending two abroad periods in two different countries. They have had a deep and impactful influence on my research, especially related to experimental activity, work organization, and collaboration with colleagues favoring both my personal and professional growth.

I would like to thank Dr. Paolo Robuffo Giordano and Dr. Claudio Pacchierotti for giving me the opportunity to work in the Rainbow Team at IRISA/INRIA Rennes and for mentoring me during the time I spent in their institution. It has been amazing working with you and establishing this collaboration which is one of the most exciting I have ever had.

My deepest gratitude goes to Prof. Elliot Wright Hawkes, who hosted and advised me during my stay at the Department of Mechanical Engineering of the University of California Santa Barbara. I feel extremely honored to have worked with you during my career, it was a stroke of incredible luck.

Also, I wish to thank all my teammates, the people I met during my stay in France and in California, during summer schools and in robotics conferences and workshops I participated to (mentioning them all would be a very long list!).

A special thank goes to my (friends and) colleagues Stanislao Grazioso and Gennaro Notomista with whom I share, among so many other things, the passion towards robotics research. I would have never imagined, when we were just graduate students deciding what to study next, that our friendship and collaboration would have lasted for so long.

I gratefully acknowledge the contributions of all the co-authors of my research papers, both those used for this thesis, and those related to complementary research themes. Thanks for every single minute you spend contributing to our works.

The final and most important expression of gratitude is for my parents, my brother and my close friends who constantly supported and motivated me, especially during the most stressful periods of this recent chapter of my life.

Mario Selvaggio
Napoli, 13.03.2020

Contents

List of Figures

List of Tables

Chapter 1

Introduction

One of the earliest manifestations of robotics can be traced back to the 1950s, when Goertz built the first pair of mechanically linked manipulators for humans to handle radioactive material [6]. The mechanical connection between the two systems allowed the operator to naturally transmit her/his motions while receiving forces and vibrations through the connecting structure. Due to the limited distance required by mechanically connected systems, Goertz recognized the value of electrically coupled manipulators and laid the foundations of modern *telerobotics* and bilateral force-reflecting positional servos [7]. However, some years were needed to stably restore the desirable sense of touch in telerobotic systems, the main problem being the time delay introduced by electrical communication over long distances.

Despite this challenge, remotely operated robots have been used since then in many hazardous and safety-critical environments while their 'intelligence' has remained quite limited. On one hand, the reason behind this is the human-in-the-loop that provides the necessary 'computation' to effectively carry out the task. On the other hand, autonomously operating robots have not yet reached the level of reliability required by some safety-critical applications. This is the case, for instance, of nuclear environments or surgical settings where failures are not tolerated at all.

While full autonomy is still far from satisfying safety requirements in such robotic applications (especially in unknown environments), huge progress has been made in this field during the last decades. Inspired by these, new approaches fusing human intelligence with the precision and efficiency of autonomous robotic systems in the form of *shared-control* architectures have risen. Early forms of shared control were reported by Sheridan *et al.* in the

1960s who formalized a strategy for human operators to perform delayed remote manipulation tasks [8, 9]. Since then, the field has evolved slowly over the years and was established as an independent multi-disciplinary research field in the 1990s.

Despite the large amount of literature about shared control methods, some questions are still to be answered, for instance: Which requirements must a shared control architecture satisfy in order to be *efficient* in terms of task execution? What sensory feedback modalities can be exploited to let the system be *intuitive* for the user? How to guarantee the *safety* of the interacting system not degrading performances? The motivation behind the work presented in this thesis stems from the willingness to answer these questions. As such, the main goal is to develop shared-control methods suitable for safe, intuitive and efficient remote manipulation in hazardous and safety-critical environments.

1.1 Thesis Overview

This thesis presents the development of advanced shared-control methods for remote teleoperation of robotic systems. The proposed methods aim at enhancing the interaction between the human operator and the semi-autonomous robotic system with the ultimate goal of making it safer, more efficient and comfortable for the user.

Autonomous robots have rapidly evolved as a result of advances in hardware capabilities (*e.g.*, more efficient sensors, computational power, etc.) and software architectures (*e.g.*, vision-based/force control, machine learning, artificial intelligence, etc.). However, fully autonomous robots are still missing the robustness and the dependability needed to be reliably adopted in unknown/unstructured environments. As such, safety-critical tasks cannot be accomplished without the superior scene understanding and decision making capabilities of humans. In this scenario, autonomous control can be adopted in many different ways to assist the user instead of providing the robotic system with full autonomy. This is motivated by three main aspects: (i) increasing the efficiency by speeding up the task execution, (ii) ensuring the safety of the system interacting with humans, and (iii) improving the overall operator's experience by decreasing her/his physical and cognitive workload.

To this end, the work presented in this thesis aims at developing effective semi-autonomous architectures that can assist the human operator in remotely controlling a robotic system which can be cumbersome to manually/directly control. Additionally, it explores different forms of informative feedback tech-

niques, mostly in the form of haptic guidance, that help the user in efficiently accomplishing the task. The safety of semi-autonomous systems is guaranteed by exploiting passivity-based control techniques, opportunely tailored for the specific architecture.

1.2 Thesis Structure

A brief outline of the thesis is given in the following.

Chapter 2 contains the basic knowledge about the control methodologies used in telerobotics and a general overview of the robotic systems used throughout this thesis as hardware testbeds. A review of the state-of-the-art techniques and shared-control architectures used in robotic teleoperation is given in Chapter 3.

The original contributions of this work, in the context of shared-control methodologies for remote telemanipulation tasks, are grouped into two parts according to the underlying robotic application: Part I contains shared-control methods suitable for remote operations in hazardous (nuclear) industrial environments, while Part II contains shared-control methods developed to operate at different scales in surgical settings.

1.3 Outline of Part 1

In Part I of the thesis, two main contributions are presented.

A shared-control architecture for a dual-arm robotic system is described in Chapter 4. The overall system is partially controlled by a human operator through a haptic device, while some degrees of freedom are autonomously regulated using a vision-based and collision avoidance control architecture. While remotely controlling the system, the user receives haptic cues that help in maintaining her/him motions compatible with the system constraints (collisions, joint limits, and singularities). The control architecture is explained in details, passivity is analyzed through port-Hamiltonian modeling of the system, experiments and a human-subject study are reported to show the performance improvements during a grasping task.

A task-prioritized shared-control architecture for redundant robots is proposed in Chapter 5. This extends the previous work to simultaneous accomplishment of n-tasks exploiting the robotic system redundancy. In this case, the user commands constitute one of the tasks in a task-prioritized control architecture, while the others are autonomously regulated to achieve a pre-programmed

behavior (supposed to help the operator in accomplishing the task). From a stability point of view, it is shown that conflicting tasks may generate energy in the system, thus proving that the overall bilateral teleoperation architecture may not be passive. An energy-tanks passivity-based controller is developed to render the system stable and thus safe for the user.

1.4 Outline of Part 2

In Part II of the thesis, three other contributions are presented.

An interactive virtual fixtures generation and adaptation method is proposed in Chapter 6. The problem of high task dependency of virtual fixtures is here tackled through the proposed methods. To give the user the possibility of interactively generating virtual fixtures for interaction tasks, a penalized spline regression algorithm is used to fit a set of recorded interaction points. After the generation, virtual fixtures can be semi-autonomously adapted and/or activated/deactivated on demand. These mechanisms result in a robotic system exhibiting variable impedance at the master side that may generate energy. A passivity-based controller exploiting the energy tanks concept is developed to keep the system safe.

A shared-control approach suitable for assisted needle grasping is proposed in Chapter 7. The problem of needle re-grasping in the middle of the suturing is addressed. To reduce the occurrence of this situation, an optimal needle grasping pose, which minimizes the possibility of coming across system constraints along the post-grasping trajectory, is calculated. The user is guided towards the optimal needle grasping pose through assistive haptic forces. Simulated and real experiments are proposed to assess the applicability of the developed method.

In Chapter 8, a novel laparoscopic tool is proposed that allows in-hand needle re-orientation. With respect to a standard tool, an additional degree of freedom introduces redundancy in the system. This can be autonomously exploited giving rise to a shared control architecture that increases the dexterity and helps avoiding the system constraints. Experiments are proposed to validate this idea.

Thesis Publications

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

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

- 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," *7th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, Enschede, 2018, pp. 1056–1061.

- M. Selvaggio, F. Abi-Farraj, C. Pacchierotti, P. Robuffo Giordano, and B. Siciliano, "Haptic-based shared-control methods for a dual-arm system," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4249–4256, 2018

- 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

- M. Selvaggio, P. Robuffo Giordano, F. Ficuciello, and B. Siciliano, "Passive task-prioritized shared-control teleoperation with haptic guidance," in *2019 IEEE International Conference on Robotics and Automation*, Montreal, Canada, May 2019, pp. 430–436

- G. A. Fontanelli, M. Selvaggio, M. Ferro, F. Ficuciello, M. Vendittelli, B. Siciliano, "Portable dVRK: an augmented V-REP simulator of the da Vinci Research Kit", *Acta Polytechnica Hungarica*, Special Issue on Platforms for Medical Robotics Research, vol. 16, no. 8, pp. 79–98

- M. Selvaggio, G. A. Fontanelli, F. Ficuciello, L. Villani, B. Siciliano, "Task classification of robotic surgical reconstructive procedures using force measurements", *7th Joint Workshop on New Technologies for Computer/Robot Assisted Surgery*

- M. Selvaggio, L. Villani, B. Siciliano, F. Ficuciello, Physics-based task classification of da Vinci robot surgical procedures, *6th National Congress of Bioengineering*

- M. Selvaggio, G. A. Fontanelli, F. Ficuciello, L. Villani, B. Siciliano, "Enhancing dexterity with a 7-DoF laparoscopic suturing tool", *Hamlyn Symposium on Medical Robotics*

- M. Selvaggio, G. A. Fontanelli, F. Ficuciello, L. Villani, B. Siciliano, "A virtual fixture adaptation strategy for MIRS dissection tasks" *8th Workshop on New Technologies for Computer/Robot Assisted Surgery*

- M. Ferro, D. Brunori, F. Magistri, L. Saiella, M. Selvaggio, G. A. Fontanelli, "A portable da Vinci simulator in virtual reality", *3rd IEEE International Conference on Robotic Computing*, Naples, Italy, 2019, pp. 447–448.

- R. Moccia, M. Selvaggio, F. Ficuciello, "Suturing needle tracking for grasping optimization in minimally invasive surgery", *Hamlyn Symposium on Medical Robotics*

- 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*

- 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*

- R. Moccia, M. Selvaggio, L. Villani, B. Siciliano, F. Ficuciello, "Vision-based virtual fixtures generation for robotic-assisted polyp dissection procedures", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, The Venetian Macao, China, Nov. 2019, pp. 3617–3623

- M. Selvaggio, A. M. Ghalamzan E., R. Moccia, F. Ficuciello, B. Siciliano, "Haptic-guided shared control for needle grasping optimization in minimally invasive robotic surgery", *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems*, The Venetian Macao, China, Nov. 2019, pp. 7934–7939

Chapter 2

Preliminaries

The work presented in this thesis builds upon some consolidated robotic methodologies developed during the last decades. In this chapter, an overview of these methods and theories is given to provide the reader with the necessary technical background to understand the successive chapters. Section 2.1 summarizes the telerobotics paradigm and give some basic definitions that are commonly employed in this field. Section 2.2 provides an overview of the haptics field with particular emphasis on Virtual Fixtures (VFs), *i.e.* virtual constraints used to implement haptic guidance in telerobotic systems. Section 2.3 contains the main definitions and theorems of the passivity-based control approach and its realization through the *energy tanks* concept. Section 2.4 gives a brief overview of the task-prioritized control architecture for multi-task execution with redundant robots. Ultimately, Section 2.5 describes the hardware platforms and the simulation environments used in the work developed for this thesis.

2.1 Telerobotics

Telerobotics, literally meaning robotics at a distance (*tele*, which is derived from the Greek, means distant), represents one of the earliest robotics application [10]. Motivated by the need to accomplish tasks in hazardous or different scale environments, researchers started to devise machines to be operated from remote sides. Initially, these machines were simply articulated mechanisms that physically implemented high-level planning or cognitive decisions made by the human operator. Practically, they were robots whose connection between the perception and action was represented by the user.

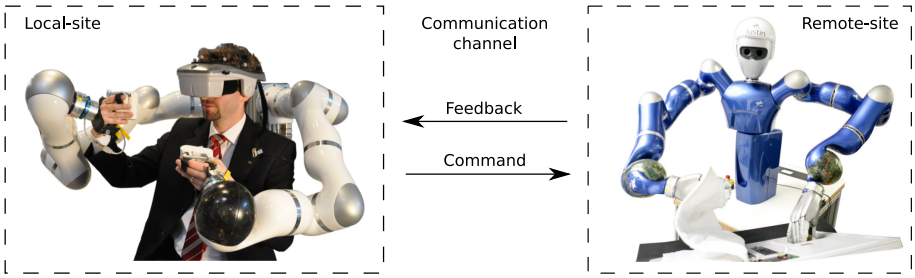


Figure 2.1: Overview of a telerobotic system (adapted from [1]).

Technically speaking, a telerobotic framework is composed of two robotic systems placed on two different sides and connected through a communication channel (Fig. 2.1). The *master* (local) side is constituted by the human operator who operates the whole system through the master robotic interface, which is used to translate actions in commands and provides different forms of sensory feedback to the user. The *slave* (remote) side is constituted by the robotic system interacting and performing the task into the environment. As explained above, this separation between the two sides is usually required by environmental constraints, such as hazards or different scales. A *masted-slave* system may encompass kinematically equal or dissimilar robots. In the case of kinematically equal robots, these can be connected at the joint level, *i.e.*, each joint at the slave exactly reproduces the movement of the corresponding joint at the master. In this case, the two sides have the same workspace and do not allow *clutching* (temporary disconnection that allows shifting the two robots configuration). Kinematically dissimilar robots are, instead, commonly coupled at their tip (in the Cartesian space). In this case, inverse kinematics techniques are used to find the joint space configuration corresponding to the desired tip pose of the robotic system at the slave side [11].

The simplest telerobotic control architecture is named *direct control* in which the slave robot is programmed to follow the motions of the master, which is unilaterally positioned by the user. At the other extreme, the *supervisory control* architecture requires very high-level user's commands while the slave robot is endowed with substantial intelligence to fulfill its function semi-autonomously. Between the two cases, lie a variety of *shared control* architectures, where some degree of autonomy or automated help is available to assist the user in accomplishing the task.

Human operators always require some form of feedback from the slave side to operate the system. Besides the essential visual displays, the most common form of feedback consists in forces rendered to the user through the

master robot. When force feedback is present, the communication becomes fully bidirectional and such telerobotic systems are denoted as *bilateral*. Feedback signals can be directly measured by the slave robot or calculated by an intelligent algorithm. In the first case, usually, the term *telepresence* is used to indicate the degree of immersiveness of the operator in the remote environment. In the second case, intelligent algorithms may be employed to calculate and display to the operator some useful information that might help in accomplishing the task. A classical example is constituted by virtual fixtures (presented in Sect. 2.2) or virtual (augmented) reality techniques (an example application is given in Sect. 2.5).

In this thesis, this last design paradigm is highly employed to develop shared-control telerobotic systems that are, at the same time, easy, intuitive and safe for the human operator.

2.2 Haptics and Virtual Fixtures

The word *haptics*, believed to be derived from the Greek word *haptesthai*, means related to the sense of touch [12]. In the neuroscience and psychology research, haptics is the study of human sensing through touch, specifically via *kinesthetic* (force) and *cutaneous* (tactile) receptors, associated with perception and manipulation. In the robotics and virtual reality literature, haptics is broadly defined as real and simulated touch interactions between robots, humans, and real or simulated environments, in various combinations. To improve the human operator performance in teleoperation scenarios, haptic interfaces seek to re-create the compelling sensation that the operator is directly touching a real environment.

Although sensory feedback from the slave side might be successfully employed to make the operator realistically feel present in the remote environment, abstract perceptual information might be beneficial to carry out the task more successfully. In this respect, the most common methodology is based on the use of *Virtual Fixtures* (VFs), which generate the abstract perceptual overlay that guides the operator. This stems from the idea that "*drawing a line with a ruler is usually straighter and faster and rely less on visual information*" [13]. Besides enhancing the task performance, the ruler concept can be generalized and applied to protect against dangerous or destructive failures. In this way, the operator is relieved both mentally and physically in operating the system. Technically speaking, VFs (often also denoted as *active constraints*) are characterized by the geometry and by a constrain enforcement

method [14, 15]. The geometry can be any $d-$dimensional ($d = 1, 2, 3$) form such as points, curves, surfaces, volumes. As for the constraint enforcement, is possible to distinguish *Guidance Virtual Fixtures* (GVF) and *Forbidden-Regions Virtual Fixtures* (FRVF). The former help the user to follow a predetermined path (attractive behavior), while, the latter act as barriers (repulsive behavior) to enforce constraints on region of the workspace the user is forbidden to enter.

In this thesis, both GVF and FRVF are employed to provide the users with haptic guidance which enhances their performances and/or prevents coming across dangerous constraints. In Chapter 4 and 5, FRVF are employed to avoid encountering mechanical constraints of the robotic slave manipulator(s), during telerobotic maintenance tasks. In Chapter 6 and 7, GVF are used to perform precise surgical dissection tasks and to aid the needle grasping process, respectively, during robotic surgical procedures. However, it is worth to remark that the term *haptic guidance* is used to denote computed forces. This is done to make a clear distinction from the *haptic feedback* case in which forces measured at the slave side are conveyed to the operator.

2.3 Passivity Theory

Passivity-based control techniques became very popular in the last decades to ensure the safe behavior of a telerobotic system. The passivity paradigm is appealing since it allows an intuitive energetic characterization of the state of the system [16]. Practically, a passive system either accumulates or dissipates the energy supplied to it through its interface. This is more rigorously explained by the following definition

Definition 1 (Passivity) *A dynamical system*

$$\begin{cases} \dot{\boldsymbol{x}} = & f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u} \\ \boldsymbol{y} = & p(\boldsymbol{x}) \end{cases} \tag{2.1}$$

where $\boldsymbol{x} \in \mathcal{X}$ is the state, $\boldsymbol{u} \in \mathcal{U}$ is the input, and $\boldsymbol{y} \in \mathcal{Y}$ is the output of the system, is passive if there exists a continuously differentiable, positive semidefinite (lower bounded) function $V : \mathcal{X} \to \mathbb{R}^+$ such that

$$V(\boldsymbol{x}(t)) - V(\boldsymbol{x}_0) \leq \int_0^t \boldsymbol{y}(\tau)^{\mathrm{T}} \boldsymbol{u}(\tau)\, \mathrm{d}\tau, \tag{2.2}$$

or, equivalently, in derivative form

$$\dot{V}(\boldsymbol{x}(t)) \leq \boldsymbol{y}^{\mathrm{T}} \boldsymbol{u}. \tag{2.3}$$

However, being a sufficient condition, sometimes enforcing passivity may result in designing overly conservative systems, while, on the contrary, this provides robustness guarantee with respect to uncertain and unmodelled dynamics. To overcome this limitation, a new control paradigm, denoted as *energy-tanks passivity-based control*, was more recently introduced [17]: it consists in endowing the system with an additional virtual energy storing element (the tank) that is used to implement the control action without violating the passivity property. This mechanism allows the system to behave less conservatively while assuring a passive and thus a safe behavior. Generally, from a mathematical point of view, the energy tank is associated with a storage function

$$T(z) = \frac{1}{2}z^2, \tag{2.4}$$

and its dynamics is chosen as

$$\dot{z} = \frac{\phi}{z}L_f V - \frac{\gamma}{z}w, \tag{2.5}$$

where $z \in R$ is the tank state, ϕ and γ are variables used to enforce an upper bound on the energy stored, $w = \boldsymbol{y}^{\mathrm{T}}\boldsymbol{u}$ denotes the power associated with the control action $\boldsymbol{u}$ to be applied to the system in (2.1). The symbol $L_f V$ denotes the Lie derivative of the function $V(\boldsymbol{x})$ along $f(\boldsymbol{x})$ direction, and, for physical systems, this represents the energy dissipated by the system. In summary, the tank accumulates the dissipated energy and keeps track of the energy potentially consumed by the control action $\boldsymbol{u}$. The storage function of the augmented system can be written as

$$G(\boldsymbol{x}, z) = T(z) + V(\boldsymbol{x}). \tag{2.6}$$

If $\boldsymbol{y} = \boldsymbol{g}^{\mathrm{T}}\frac{\partial V}{\partial \boldsymbol{x}}$, it can be easily shown that the extended system (2.1)–(2.5) is passive with respect to the storage function (2.6) and input-output pair $(\boldsymbol{u}, \boldsymbol{y})$ with a proper choice of the control variables ϕ and γ. Opportunely choosing these control variables allows enforcing an upper bound of the stored energy. When the tank is depleted, only the actions $\boldsymbol{u}$ that make the tank energy to increase ($w < 0$) are implementable on the system. A possible solution is to implement $\alpha(z, \boldsymbol{u})\boldsymbol{u}$ instead of $\boldsymbol{u}$ with $\alpha(z, \boldsymbol{u})$ satisfying

$$\begin{cases} \alpha(z, \boldsymbol{u}) & = 0 \quad \text{if} \quad T(z) = 0 \quad \& \quad w > 0 \\ \alpha(z, \boldsymbol{u}) & = 1 \quad \text{otherwise} \end{cases}. \tag{2.7}$$

This passivity enforcing mechanism has been opportunely adapted and used throughout the work developed for this thesis.

Another appealing method to analyze the system behavior is to look at its *port-Hamiltonian* model. Formally, given a system, the associated port-Hamiltonian model can be expressed in the form

$$\begin{cases} \dot{\boldsymbol{x}} = [J(\boldsymbol{x}) - R(\boldsymbol{x})] \dfrac{\partial \mathcal{V}}{\partial \boldsymbol{x}} + g(\boldsymbol{x})\, \boldsymbol{u} \\ \boldsymbol{y} = g^{\mathrm{T}}(\boldsymbol{x}) \dfrac{\partial \mathcal{V}}{\partial x} \end{cases} \tag{2.8}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ represents the system state, $\mathcal{V}(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ is the Hamiltonian function, namely the sum of the kinetic and potential energies of the system, $J(\boldsymbol{x}) = -J(\boldsymbol{x})^{\mathrm{T}}$ represents the internal interconnection, $R(\boldsymbol{x}) \geq 0$ the internal dissipation, $g(\boldsymbol{x})$ the input matrix, $\boldsymbol{u}$ the system input, and $\boldsymbol{y}$ the system output. It is easy to show that for a port-Hamiltonian system (PHS) the following inequality holds

$$\boldsymbol{u}^{\mathrm{T}} \boldsymbol{y} = \dot{\mathcal{V}}(\boldsymbol{x}) - \frac{\partial^{\mathrm{T}} \mathcal{V}}{\partial \boldsymbol{x}} R(\boldsymbol{x}) \frac{\partial \mathcal{V}}{\partial \boldsymbol{x}} \leq \dot{\mathcal{V}}(\boldsymbol{x}). \tag{2.9}$$

Equation (2.9) establishes the inherent passivity condition of a PHS with respect to the input-output pair $(\boldsymbol{u}, \boldsymbol{y})$ with storage function $\mathcal{V}(\boldsymbol{x})$. Practically, when the evolution of the system can be described through a port-Hamiltonian model, the system is inherently passive with respect to the function $\mathcal{V}$ taken as a storage function.

2.4 Task Priorities

Consider a serial n-Degree-of-Freedom (DoF) manipulator. The robot configuration can be expressed by the vector of generalized coordinates $\boldsymbol{q} \in \mathbb{R}^n$. Consider then the operational (or task) space with coordinates denoted by $\boldsymbol{x} \in \mathbb{R}^m$. The robot is *redundant*, when $(n > m)$ is satisfied, *i.e.* the robot has more DoFs than the one necessary to perform the task.

A *task* can be expressed by a generic function of the system state $\boldsymbol{\sigma}(\boldsymbol{q}) : \mathbb{R}^n \to \mathbb{R}^m$ such that $\dot{\boldsymbol{\sigma}} = \boldsymbol{J}(\boldsymbol{q})\, \dot{\boldsymbol{q}}$, with $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{m \times n}$ being the task Jacobian, which maps velocities from joint space to the task space. Accomplishing a task means to bring $\boldsymbol{\sigma}(\boldsymbol{q})$ to a desired value $\boldsymbol{\sigma}^*(\boldsymbol{q})$. *Set-based tasks* are tasks satisfied in a set of valid values. More specifically, the desired values for $\boldsymbol{\sigma}(\boldsymbol{q})$ belong to a set, *i.e.* $\boldsymbol{\sigma}^* \in \mathcal{D}$. This generalizes *equality tasks* which have a specific valid value, *i.e.* $\boldsymbol{\sigma}^* = \boldsymbol{\sigma}$. Joint limits and obstacle avoidance are classical examples of set-based tasks.

When the robot is an industrial manipulator, with low-level dynamic control loop faster than the kinematic one, joint velocity can be directly commanded, *i.e.* $\dot{\boldsymbol{q}} = \boldsymbol{u}_c$. Defining the task error as follows $\tilde{\boldsymbol{\sigma}}_i = \boldsymbol{\sigma}_i^* - \boldsymbol{\sigma}_i$, and choosing the control input as $\boldsymbol{u}_c = \boldsymbol{J}^\dagger \boldsymbol{\Lambda} \tilde{\boldsymbol{\sigma}}$, we have

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^\dagger \boldsymbol{\Lambda} \tilde{\boldsymbol{\sigma}} \tag{2.10}$$

that lets $\tilde{\boldsymbol{\sigma}} \to \boldsymbol{0}$ exponentially with minimum norm joint velocities. In (2.10), the symbol $\boldsymbol{J}^\dagger \in \mathbb{R}^{n \times r}$ denotes the Jacobian right Moore-Penrose pseudoinverse.

When multiple tasks need to be satisfied at the same time, a *task-prioritized* control architecture can be adopted. This allows accomplishing the primary task while trying to fulfill a number of hierarchically organized secondary tasks. In this case, the velocity control input chosen as $\boldsymbol{u}_c = \sum_{i \in T} \boldsymbol{P}_{i-1} \boldsymbol{J}_i^\dagger \boldsymbol{\Lambda}_i \tilde{\boldsymbol{\sigma}}_i$ yields

$$\dot{\boldsymbol{q}} = \sum_{i \in T} \boldsymbol{P}_{i-1} \boldsymbol{J}_i^\dagger \boldsymbol{\Lambda}_i \tilde{\boldsymbol{\sigma}}_i \tag{2.11}$$

where T is the *stack-of-tasks*, *i.e.* an ordered set of $r \in \mathbb{N}$ tasks, where the i-th element has task-space dimension r_i, and $\boldsymbol{\Lambda}_i \in \mathbb{R}^{r_i \times r_i}$ is the diagonal and positive-definite $i-$th task gain matrix, $\dot{\boldsymbol{\sigma}}_i$ is the i-th task space velocity, such that $\dot{\boldsymbol{\sigma}}_i = \boldsymbol{J}_i \dot{\boldsymbol{q}}$ where $\boldsymbol{J}_i \in \mathbb{R}^{r_i \times n}$ denotes the i-th task Jacobian matrix and $\boldsymbol{J}_i^\dagger \in \mathbb{R}^{n \times r_i}$ its right Moore-Penrose pseudoinverse, *i.e.* $\boldsymbol{J}_i^\dagger = \boldsymbol{J}_i^{\mathrm{T}} (\boldsymbol{J}_i \boldsymbol{J}_i^{\mathrm{T}})^{-1}$, and $\boldsymbol{P}_i \in \mathbb{R}^{n \times n}$ is the $i-$th null-space projector (with $\boldsymbol{P}_0 = \boldsymbol{I}_n$ being the $n \times n$ identity matrix).

The i-th null space projector $\boldsymbol{P}_i$ in (2.11) can be computed either by using the *augmented Jacobian* of all higher priority tasks, *i.e.*

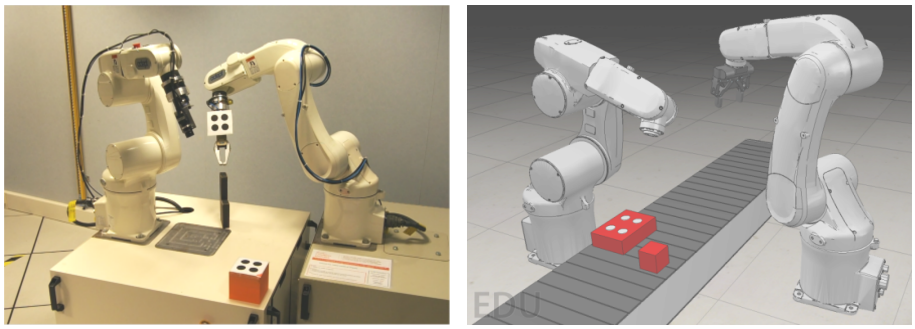
$$\bar{\boldsymbol{J}}_i = \begin{bmatrix} \boldsymbol{J}_1 \\ \vdots \\ \boldsymbol{J}_i \end{bmatrix} \quad \boldsymbol{P}_i = \left(\boldsymbol{I}_n - \bar{\boldsymbol{J}}_i^\dagger \bar{\boldsymbol{J}}_i \right), \tag{2.12}$$

or the *successive projection* of Jacobians, i.e

$$\boldsymbol{P}_i = \left(\boldsymbol{I}_n - \boldsymbol{J}_i^\dagger \boldsymbol{J}_i \right), \quad \boldsymbol{P}_i = \boldsymbol{P}_1 \dots \boldsymbol{P}_{i-1}. \tag{2.13}$$

Clearly, the i-th task is executed as long as the null-space of the $i-1$-th Jacobian matrix is not empty, *i.e.* $\mathcal{N}(\boldsymbol{J}_{i-1}) \neq \varnothing$ and $\boldsymbol{P}_{i-1} \neq \boldsymbol{O}$. When $\mathcal{N}(\boldsymbol{J}_i) = \varnothing$ the corresponding null-space projector becomes $\boldsymbol{P}_i = \boldsymbol{O}_{n \times n}$.

Stability properties of the system in (2.11) depend upon the tasks orthogonality/independence and the chosen null-space projection method, *i.e.* (2.12)



(a) (b)

Figure 2.2: IRISA/INRIA Rennes dual-arm robotic system: (a) real and (b) simulated environments.

or (2.13). More precisely, two tasks i and j are said to be *orthogonal* when $\boldsymbol{J}_{s,i}\boldsymbol{J}_{s,j}^{\dagger} = \boldsymbol{O}_{r_i \times r_i}$ while they are said to be *independent* when $\rho\left(\boldsymbol{J}_{s,i}^{\dagger}\right) + \rho\left(\boldsymbol{J}_{s,j}^{\dagger}\right) = \rho\left(\left[\boldsymbol{J}_{s,i}^{\dagger}\ \boldsymbol{J}_{s,j}^{\dagger}\right]\right)$ where $\rho\left(\cdot\right)$ denotes the rank function. Alternatively, it is possible to replace the Jacobian pseudoinverse in (2.12) or (2.13) by its transpose obtaining similar convergence results. A more comprehensive overview of task-prioritized control methods, as well as their stability properties, is given in [18].

When a task-prioritized robotic system interacts with the environment (or a user), the controlled systems in (2.11) becomes

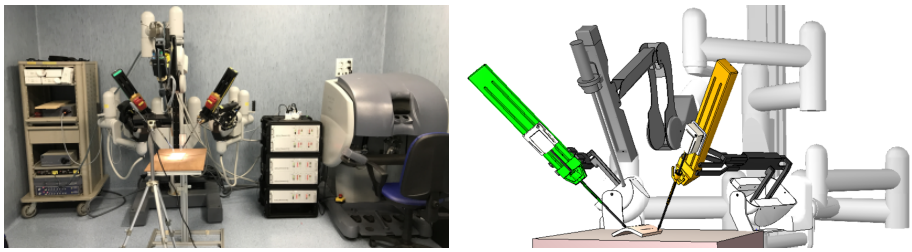
$$\dot{\boldsymbol{q}} = \sum_{i \in T} \boldsymbol{P}_{i-1}\boldsymbol{J}_i^{\dagger}\boldsymbol{\Lambda}_i\tilde{\boldsymbol{\sigma}}_i - \dot{\boldsymbol{q}}_e \tag{2.14}$$

where $\dot{\boldsymbol{q}}_e$ is the exogenous velocity input due to the interaction, for instance calculated using an admittance control scheme [19].

2.5 Hardware and Software

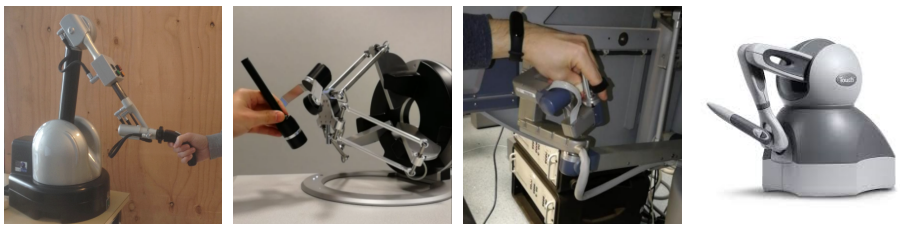
Two robotic systems have been used as hardware testbeds in the work reported in this thesis; correspondingly, simulated scenarios have been realized to develop and test the proposed shared-control algorithms.

The first system, emulating industrial scenarios, is a dual-arm robotic system located at IRISA/INRIA Rennes, France (see Fig. 2.2a). The system is composed of two Adept Viper 6-DoF anthropomorphic robotic manipulators, namely the Adept Viper 650 and the Adept Viper 850, endowed with a camera



(a) (b)

Figure 2.3: ICAROS Da Vinci Research Kit: (a) real and (b) simulated environments.



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

Figure 2.4: Master robotic platforms used as haptic interfaces: (a) Haption Virtuose 6D; (b) Force Dimension Omega 6; (c) dVRK Master Tool Manipulator; (d) Geomagic touch.

and a ROBOTIQ 2-finger gripper, respectively. The master-slave telerobotic system is completed by a haptic device that differs depending on the application scenario. More in details, the Haption Virtuose 6D Desktop interface (see Fig. 2.4a) has been used in the work described in Chapter 4 while the Force Dimension OMEGA-6 (see Fig. 2.4b) has been used in Chapter 5. Both haptic devices, used as master robots, provide gravity compensation and a Cartesian space control interface, in the sense that is possible to read the position and set forces at the tip for haptic rendering. Thus, no kinematic/dynamic models are needed for control purposes. The kinematic model of the two Adept manipulators (see Fig. 2.5) has been used to develop the shared-control algorithms presented in Part I and to build the corresponding simulated environment (see Fig. 2.2b). Denoting with $\boldsymbol{q} = [q_1, \ldots, q_6]^{\mathrm{T}}$ the vector of the Viper arm generalized coordinates, the homogeneous transformation matrix[1]

[1]Hereafter, we use the matrix notation $\boldsymbol{T}_b^a$, where the superscript a denotes the frame in

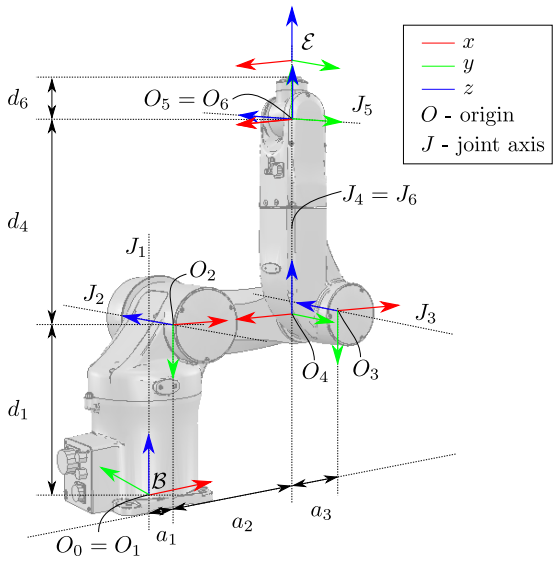


Figure 2.5: Adept Viper DH frame representation.

$\boldsymbol{T}_{\mathcal{E}}^{\mathcal{B}}(\boldsymbol{q}) \in SE(3)$, representing the pose of the Adept Viper end-effector frame $\mathcal{E} : \{\boldsymbol{O}_e; \boldsymbol{x}_e, \boldsymbol{y}_e, \boldsymbol{z}_e\}$ with respect to the base frame $\mathcal{B} : \{\boldsymbol{O}_b; \boldsymbol{x}_b, \boldsymbol{y}_b, \boldsymbol{z}_b\}$, can be easily computed applying the standard Denavit-Hartenberg (DH) convention to the kinematic chain $\{J_1, \ldots, J_6\}$ of Fig. 2.5. Table 2.1 and 2.2 contain the DH parameters of the Viper 650 and the Viper 850, respectively.

The second system is the da Vinci Research Kit (dVRK) of the University of Naples Federico II, located at the ICAROS Center (see Fig. 2.3a). The dVRK is an open-source robotic platform obtained from the first-generation

which vector components are expressed, the subscript b the current frame (*e.g.*, $\boldsymbol{T}_{\mathcal{E}}^{\mathcal{B}}$ denotes the pose of the end-effector frame expressed in the base frame).

Table 2.1: DH parameters of the Adept Viper 650

link	joint	$a_i[m]$	$\alpha_i[rad]$	$d_i[m]$	$\theta_i[rad]$
1	R	0.075	$-\pi/2$	0.335	q_1
2	R	0.270	0	0	q_2
3	R	0.90	$-\pi/2$	0	$q_3 + \pi$
4	R	0	$\pi/2$	0.295	q_4
5	R	0	$-\pi/2$	0	q_5
6	R	0	0	0.08	q_6

Intuitive Surgical System, endowed with open controllers and software developed at Johns Hopkins University LCSR and Worcester Polytechnic Institute AIM Lab [20]. The full dVRK consists of two/three Patient Side Manipulators (PSMs) and one Endoscopic Camera Manipulator (ECM) constituting the slave side, and two Master Tool Manipulators (MTMs) (see Fig. 2.4c), acting as master robots. For the work developed in this thesis, the MTMs use an impedance control interface developed using the robot dynamic parameters identified in [21]. The kinematic model of the slave robots, in the form of DH parameters, is given in Tables 2.3, 2.4 and 2.5. The PSMs are mounted on a Setup Joint (SUJ) that allows manual spatial positioning of PSMs and ECM bases. The SUJ is an articulated structure composed by three or, in the newest versions, four arms. The two PSMs are located at the end of two 6-DoF arms (indicated hereafter as SUJ-PSMs) while the ECM is located at the end of a 4-DoF arm (SUJ-ECM). All the robotic arms in the SUJ are not actuated by motors but it is possible to control breaks in each joint and read the angular position using potentiometers [22]. Denoting with $\boldsymbol{q}_{sp} = [q_{sp,1}, \ldots, q_{sp,6}]$ the vector of the SUJ-PSMs arms generalized coordinates, the homogeneous transformation matrix $\boldsymbol{T}^{\mathcal{B}}_{\mathcal{AP}}(\boldsymbol{q}_{sp}) \in SE(3)$, representing the pose of the SUJ-PSMs end-effector frame $\mathcal{AP} : \{\boldsymbol{O}_{ap}; \boldsymbol{x}_{ap}, \boldsymbol{y}_{ap}, \boldsymbol{z}_{ap}\}$ with respect to the base frame $\mathcal{B} : \{\boldsymbol{O}_b; \boldsymbol{x}_b, \boldsymbol{y}_b, \boldsymbol{z}_b\}$, can be easily computed applying the standard DH convention to the kinematic chain $\{J_1, \ldots, J_6\}$ of Fig. 2.6 (see Table 2.3 where $a_2 = 0.58$ m, $a_3 = 0.56$ m and $d_4 = 0.425$ m). Moreover, denoting with $\boldsymbol{q}_{se} = [q_{se,1}, \ldots, q_{se,4}]$ the vector of the SUJ-ECM arm generalized coordinates, the pose of the SUJ-ECM end-effector frame $\mathcal{AE} : \{\boldsymbol{O}_{ae}; \boldsymbol{x}_{ae}, \boldsymbol{y}_{ae}, \boldsymbol{z}_{ae}\}$ with respect to the base frame $\mathcal{B} : \{\boldsymbol{O}_b; \boldsymbol{x}_b, \boldsymbol{y}_b, \boldsymbol{z}_b\}$, defined by the homogeneous transformation matrix $\boldsymbol{T}^{\mathcal{B}}_{\mathcal{AE}}(\boldsymbol{q}_{se}) \in SE(3)$, can be computed considering only the first four rows of Table 2.3. Notice that, two constant homogeneous transformation matrices $\boldsymbol{T}^{\mathcal{AP}}_{\mathcal{BP}} \in SE(3)$ and $\boldsymbol{T}^{\mathcal{AE}}_{\mathcal{BE}} \in SE(3)$ must be considered to complete the kinematics description, providing the transfor-

Table 2.2: DH parameters of the Adept Viper 850

link	joint	$a_i[m]$	$\alpha_i[rad]$	$d_i[m]$	$\theta_i[rad]$
1	R	0.075	$-\pi/2$	0.335	q_1
2	R	0.365	0	0	q_2
3	R	0.90	$-\pi/2$	0	$q_3 + \pi$
4	R	0	$\pi/2$	0.405	q_4
5	R	0	$-\pi/2$	0	q_5
6	R	0	0	0.101	q_6

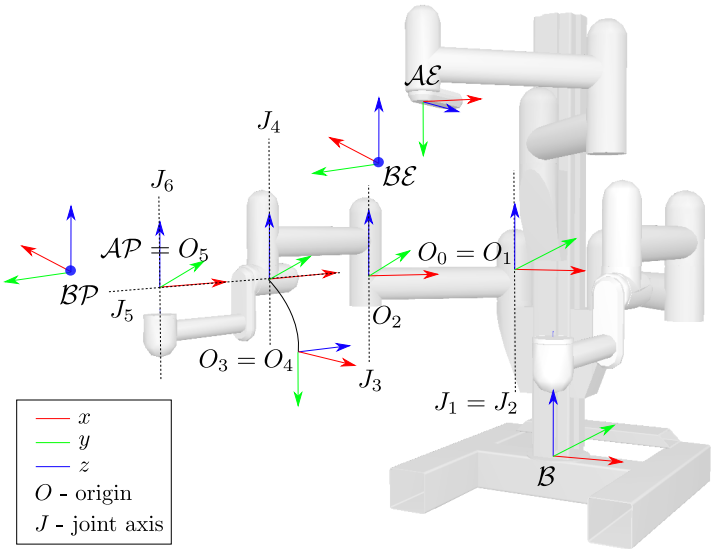


Figure 2.6: SUJ kinematic description.

mation between $\mathcal{AP}$ and $\mathcal{AE}$ (respectively the last SUJ-PSM and SUJ-ECM frames) and the base frames $\mathcal{BP}$ and $\mathcal{BE}$ of the PSMs and of the ECM, respectively (see Fig. 2.6).

Each PSM is a 7-DoF actuated arm, which moves a surgical instrument about a Remote Center of Motion (RCM), i.e., a fixed fulcrum point that is invariant with respect to the configuration of the PSM joints [23, 21]. The first 6 DoFs correspond to Revolute (R) or Prismatic (P) joints, combined in a RRPRRR sequence. The last DoF corresponds to the opening and closing motion of the gripper. The homogeneous transformation matrix $\boldsymbol{T}_{\mathcal{G}}^{\mathcal{BP}}(\boldsymbol{q}_p) \in SE(3)$ (where $\boldsymbol{q}_p = [q_{p,1}, \ldots, q_{p,6}]$ is the vector of the PSM generalized co-

Table 2.3: DH parameters of the SUJ

link	joint	$a_i[m]$	$\alpha_i[rad]$	$d_i[m]$	$\theta_i[rad]$
1	P	0	0	$q_{se,1}$	–
2	R	a_2	0	–	$q_{se,2}$
3	R	a_3	0	–	$q_{se,3}$
4	R	0	$-\pi/2$	–	$q_{se,4}$
5	R	0	$\pi/2$	$-d_4$	$q_{se,5}$
6	R	0	0	–	$q_{se,6}$

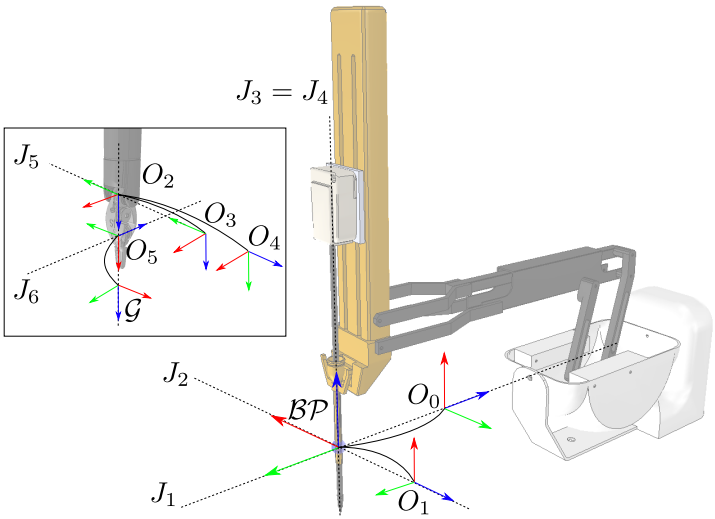


Figure 2.7: PSM kinematic description.

ordinates), representing the pose of the gripper frame $\mathcal{G} : \{\boldsymbol{O}_g; \boldsymbol{x}_g, \boldsymbol{y}_g, \boldsymbol{z}_g\}$ with respect to the base frame $\mathcal{BP} : \{\boldsymbol{O}_{bp}; \boldsymbol{x}_{bp}, \boldsymbol{y}_{bp}, \boldsymbol{z}_{bp}\}$, can be easily computed by choosing the origin of frame $\mathcal{BP}$ in the RCM point and applying the standard DH convention to the kinematic chain $\{J_1, \ldots, J_6\}$ of Fig. 2.7 (see Table 2.4, where $a_5 = 0.0091$ m).

The ECM is a 4-DoF actuated arm, which moves the endoscopic camera about the RCM through revolute and prismatic joints, combined in a RRPR sequence. The homogeneous transformation matrix $\boldsymbol{T}_{\mathcal{C}}^{\mathcal{BC}}(\boldsymbol{q}_e) \in SE(3)$ (where $\boldsymbol{q}_e = [q_{e,1}, \ldots, q_{e,4}]$), representing the pose of the camera frame $\mathcal{C} = \{\boldsymbol{O}_c; \boldsymbol{x}_c, \boldsymbol{y}_c, \boldsymbol{z}_c\}$ with respect to the base frame $\mathcal{BC} = \{\boldsymbol{O}_{bc}; \boldsymbol{x}_{bc}, \boldsymbol{y}_{bc}, \boldsymbol{z}_{bc}\}$, can be easily computed by choosing the origin of frame $\mathcal{CB}$ in the RCM point and applying the standard DH convention to the kinematic chain $\{J_1, \ldots, J_4\}$ of

Table 2.4: DH parameters of the PSM

link	joint	$a_i[m]$	$\alpha_i[rad]$	$d_i[m]$	$\theta_i[rad]$
1	R	0	$-\pi/2$	–	$q_{p,1}$
2	R	0	$-\pi/2$	–	$q_{p,2}$
3	P	0	0	$q_{p,3}$	–
4	R	0	$\pi/2$	–	$q_{p,4}$
5	R	a_5	$-\pi/2$	–	$q_{p,5}$
6	R	0	$-\pi/2$	–	$q_{p,6}$

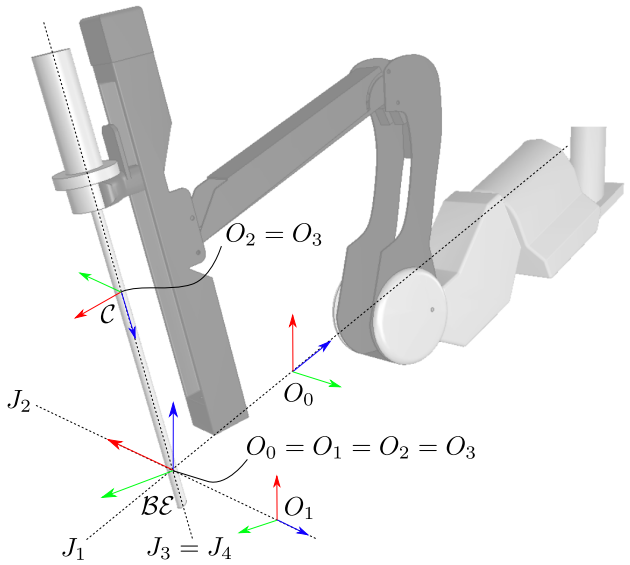


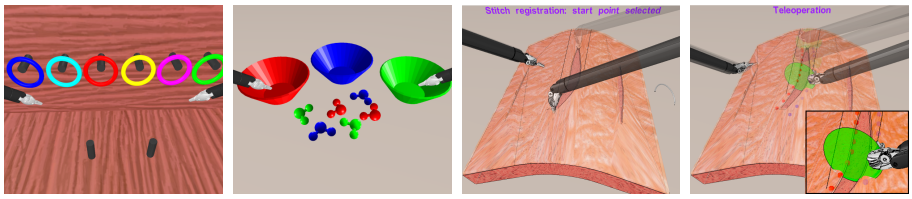
Figure 2.8: ECM kinematic description.

Fig. 2.8 (parameters are given in Table 2.5, where $d_4 = 0.007$m).

Both systems have been re-created in simulation using V-Rep (now Coppelia Sim) [24]. V-Rep is a robotic simulator based on a distributed control architecture. 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 programs can be written in C/C++, Python, Java, Lua, Matlab or Octave. The development of the simulated environment for the dVRK has been published in [5, 25, 26]. More in details, in [5] the dVRK simulator (illustrated in Fig. 2.3b), was preliminarily presented. The simulated setup includes the kinematic model of the SUJ, PSMs, ECM, the cameras and the interface with the ROS framework. In addition to this, example simulation scenes were proposed to demonstrate the potential bene-

Table 2.5: DH parameters of the ECM

link	joint	$a_i[m]$	$\alpha_i[rad]$	$d_i[m]$	$\theta_i[rad]$
1	R	0	$-\pi/2$	–	$q_{e,1}$
2	R	0	$-\pi/2$	–	$q_{e,2}$
3	P	0	0	$q_{e,3}$	–
4	R	0	0	d_4	$q_{e,4}$



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

Figure 2.9: Training and suturing scene setup. Two examples of training tasks: (a) peg on board; (b) pick and place. Augmented reality suturing scene: (c) wound registration; (d) stitches planning and execution.

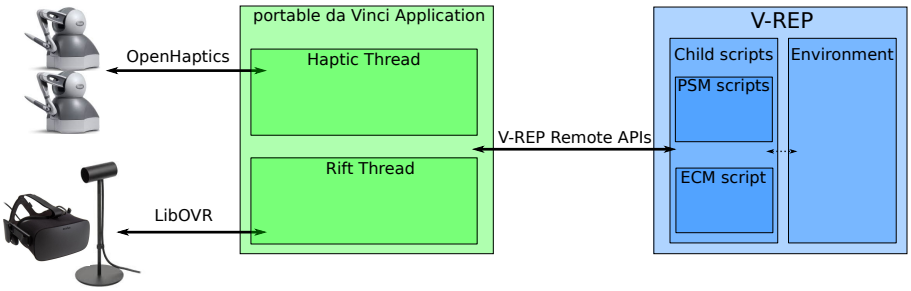
Figure 2.10: The presented *portable* da Vinci simulated system.

fits introduced by the simulator in robotic surgery research (see Fig. 2.9). Besides training and suturing environments, as an example of novel control strategies development and test, a needle tracking [27] and a visual servoing [28] environments were implemented as well. The complete simulator, together with the developed application scenes, is open-source and is available at https://github.com/unina-icaros/dvrk-vrep.git.

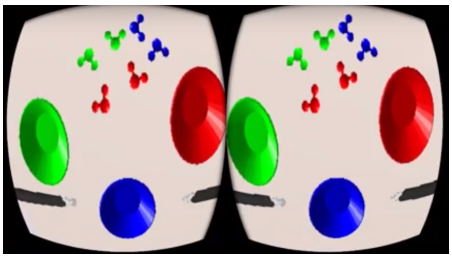
Extended functionalities were provided to the open-source da Vinci simulator in [25]. In particular, a low-cost version of the Master surgeon console was created to teleoperate the simulated robot and immerse the user in the virtual environment. For the purpose, a pair of Geomagic Touch[2] haptic devices and an Oculus Rift[3] virtual reality headset have been employed.

The pair of Geomagic Touch haptic interfaces (see Fig. 2.4d) have been used to emulate the pair of MTMs (see Fig. 2.4c), while the binocular vision

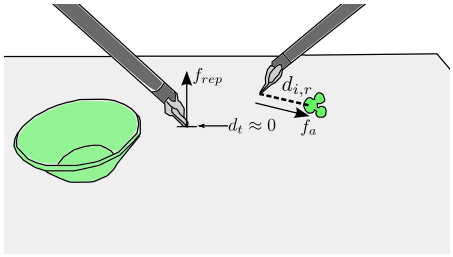
[2] https://www.3dsystems.com/haptics-devices/touch
[3] https://www.oculus.com/



(a) Module and device communication scheme of the portable da Vinci application.



(b) Oculus view of the training scene.



(c) Representation of the functionalities implemented in the training scenario.

Figure 2.11: Portable da Vinci (a) software architecture, (b) oculus view, (c) guidance/feedback functionalities.

is replicated through the use of the Oculus Rift. Aside from teleoperating the PSMs through the motion of the Geomagic stylus, this setup allows the user to freely move the Head-Mounted Display (HMD) and to directly control the ECM through the head movements, with a resulting fully immersive experience that also extends the capabilities of the real robotic system, as the camera view changes in a more intuitive way. As a result, the da Vinci Master console is emulated in a low-cost and easy-to-access fashion, thus realizing a *portable* da Vinci simulated system (see Fig. 2.10). For this purpose, pose and velocity measurements of the Oculus HMD are acquired through the Constellation positional tracking system, and used to build, through geometric considerations, the 6D velocity vector ${}^{\mathcal{C}}\boldsymbol{v}_{\mathcal{C}}$ of the endoscopic camera in its frame $\mathcal{C}$. The corresponding 6×4 Jacobian matrix $\boldsymbol{J}_{\mathcal{C}}$, reconstructed from the direct kinematics, is used to map the camera motion to the ECM 4D joint velocity vector $\dot{\boldsymbol{q}}_e$. However, since the ECM is a 4-DoF RRPR manipulator, only 4 out of the 6 Cartesian velocity space dimensions can be assigned. Therefore, alternate control of (i) the camera orientation, through the three revolute joints of the

arm and of (ii) the position along the longitudinal axis of the arm, through the prismatic joint of the arm, is implemented.

Analogously, pose and velocity measurements of the Geomagic stylus, held by the user, are retrieved from the joint encoders of the Geomagic device, to build the 6D velocity vector ${}^{\mathcal{G}}\boldsymbol{v}_{\mathcal{G}}$ of the PSM gripper expressed in its frame $\mathcal{G}$, through twist matrix transformations. Each Geomagic Touch device is a 6-DoF haptic interface equipped with joint encoders that measure the full pose of the Haptic Interface Point (HIP) of the stylus held by the user (see Fig. 2.4d). The device also provides force feedback along 3 DoFs to reproduce physical contacts with virtual objects. The corresponding Jacobian matrix $\boldsymbol{J}_{\mathcal{G}}$ is computed in the gripper frame $\mathcal{F}_{\mathcal{G}}$, reconstructed from the direct kinematics. Mapping the motion of the gripper to the corresponding PSM joint velocity vector $\dot{\boldsymbol{q}}_p$ is done through Jacobian matrix inversion.

To cope with the geometrical heterogeneity between master and slave robots (see Sect. 2.1), one of the stylus buttons is used to implement a *clutch*-based mechanism, to enable/disable the teleoperation of the slave with the master device upon the explicit command of the user. This allows relocating the stylus of the Geomagic in a more favorable configuration when the workspace limits of the device are reached. The complete software architecture is shown in Fig. 2.11a. To show the effectiveness of the simulator, a *pick-and-place* simulated training scenario is developed, considering a set of small rigid objects to be grasped and placed in three different cups on a table, while observing the scene from the endoscopic camera (Fig. 2.11b). Haptic feedback and guidance techniques are implemented to provide the user with environmental contact awareness and to easy the reach-to-grasp phase 2.11c.

This simulator has been also presented to the Maker Faire 2018 of Rome, and has been tested by several non-expert users. A comprehensive video, showing users performing the simulation described above, along with the source code of the simulator, can be found at the following link: `http://www.diag.uniroma1.it/~labrob/research/portableDaVinci.html`.

Finally, in [26], integration of advanced surgical instruments into the developed dVRK simulator was carried out. More specifically, the MUSHA hand [29, 30] and a novel needle holder with in hand rolling capabilities [31] (presented in Chapter 8) have been integrated. Moreover, simulation of deformable tissue interaction have been realized using the external Bullet dynamic simulation engine[4].

[4] `https://pybullet.org/wordpress/`

Chapter 3

State-of-art

Every robotic system, even the most autonomous, is operated by humans and/or used to fulfill a certain human need. Hence, the interaction between humans and robots, albeit minimal, is unavoidable. The field of study dedicated to understanding, designing, and evaluating robotic systems for use by or with humans is Human-Robot Interaction (HRI) [32]. A general overview of this field, with particular emphasis on safety and dependability, is given in [33].

Interaction, by definition, requires communication between the machine and its users. Different forms of communication are usable depending on the underlying applications, which may require from social to physical interaction. When dealing with shared-control robots, one is more concerned about *physical interaction* between a human and a robot. Physical interaction may take several forms and it is denoted as *proximate* when the human and the robot are in close proximity to each other, otherwise it is denoted as *remote*.

In shared control applications, *autonomy* constitutes a mean rather than the goal and its level varies widely from one application to another. Different descriptions of the levels of autonomy were proposed in the literature. The oldest and most widely adopted classification of autonomy levels in robotics is by Ferrell *et al.* whose scale ranges from the robotic system being fully commanded by an operator to being completely autonomous [34]. While for autonomous vehicles [35] levels of automation for on-road vehicles are well-defined [36], yet no such definitions exist for assistive robots. A taxonomy has recently been published with a special emphasis on the surgical robotics scenario [37]. Surgical robotic systems constitute the most prominent example of shared-control robots that operate in safety-critical environments and may

exhibit different levels of autonomy to actively assist the surgeon. Six levels of autonomy have been identified in the above-mentioned taxonomy, *i.e.*

- **Level 0**: No autonomy. This level includes teleoperated robots that faithfully respond to and follow the user's command.
- **Level 1**: Robot assistance. The robot provides some mechanical or virtual guidance during a task while the human has continuous control of the system. Examples include surgical robots with virtual fixtures (or active constraints).
- **Level 2**: Task autonomy. The robot is autonomous for specific tasks initiated by a human. The difference from Level 1 is that the operator has discrete, rather than continuous, control of the system. An example is surgical suturing – the surgeon indicates where a running suture should be placed, and the robot performs the task autonomously while the surgeon monitors and intervenes as needed [38].
- **Level 3**: Conditional autonomy. A system generates task strategies but relies on the human to select from among different strategies or to approve an autonomously selected strategy. This type of robot can perform a task without close oversight.
- **Level 4**: High autonomy. The robot can make decisions but under the supervision of a qualified user. A surgical analogy would be a robotic resident, who performs the surgery under the supervision of an attending surgeon.
- **Level 5**: Full autonomy. At this level, no human is needed.

The choice of the autonomy level depends on the task and on the environment. Safety-critical tasks in unstructured environments require the human to be constantly in control of the system (Level 0 to 2). On the contrary, in a highly-structured environment, the robot may exhibit higher autonomy thus limiting the user interventions (Level 3 to 6).

In some robotics applications, a seamless adaptation of the robot autonomy level is desirable. For instance, in human-robot collaborative scenarios, the robot may vary its impedance accounting for the human operator motion intentions estimation [39]. A review of intent detection, arbitration, and feedback in shared-control scenarios is presented in [40]. Autonomy can be adapted by resorting to a principled human-robot mutual adaptation formalisms [41, 42, 43],

or by developing mathematical and game-theoretic models of the collaboration [44, 45, 46, 47].

In the next sections, an overview of the related works in different areas involved in remote human-robot interaction is given. In detail, the most widely adopted teleoperation control architectures are summarized in Sect. 3.1. In Sect. 3.2 haptic feedback and guidance methods are reviewed. Stability and passivity enforcing controllers developed in the past are given in Sect. 3.3. Finally, the limitations of the currently adopted methods are analyzed in Sect. 3.4.

3.1 Teleoperation Control Architectures

In remote human-robot interaction applications, commonly denoted as telerobotics, classically the control architectures corresponding to different levels of autonomy have been grouped into three classes: *direct control*, *shared-control* and *supervisory control* [10].

Direct control implies no intelligence or autonomy in the system so that all slave DoFs are directly controlled by the user via the master interface. The first direct control telerobotic system was built by Goertz in the 1950s and was constituted by a pair of mechanically linked manipulators [6]. Electrically coupled manipulators laid the foundations of modern telerobotics and bilateral force-reflecting positional servos [7]. The Central Research Laboratory model M2 of 1982 was the first telerobotic system which realized force feedback while separating master and slave electronics. After that, several systems were developed mostly driven by nuclear [48] and space applications [49]. Throughout the 1980s and 1990s, as nuclear power activities began to decline, interests expanded into new areas including minimally invasive surgery [50, 51, 52, 53, 54], and undersea operations [55, 56, 57, 58, 59]. At the beginning of the 1960s the effects of time delay on bilateral teleoperation started to become a topic of research [8, 60]. To cope with this problem the concept of supervisory control was introduced [34] combining local sensory feedback, predictive displays, and teleoperation [61] and inspired the next years of development in the field.

In *supervisory control*, the user commands and feedback occur at a higher level [62, 63]. The connection is looser and the slave has to rely on a stronger local autonomy to refine and execute tasks. In general, supervisory control techniques focus on shifting more and more autonomy and intelligence to the robotic system. Supervisory control has been used in space applications with

large communication delays giving rise to the telesensor programming approach, characterized as a task-level programming technique, and simulation-based teaching by showing approaches [64, 65]. In essence, the operators interact with a complex simulation of the robot and remote environment, in which they can test and adjust tasks. The tasks, consisting of robot and environment signals and configuration parameters, are then uploaded to the remote site where the robot performs.

Shared control is comprehensive of all the intermediate levels in which the slave motion is controlled by a combination of direct user commands and local sensory feedback or autonomy [66]. It tries to combine the basic reliability and sense of presence achievable by direct control with the intelligence and possible safety guarantees of autonomous control [67, 68]. This may occur in various forms. For example, the slave robot may need to correct motion commands, regulate subsets of joints or subtasks, or overlay additional commands. One may also want the slave to assume control of subtasks, such as maintaining a grasp over long periods of time [69]. In robotic surgery applications, shared control has been proposed to compensate for beating heart movements. The sensed heart motion is overlaid on the user commands, so the surgeon can operate on a virtually stabilized patient [70]. In [71], a formal analysis of human-robot cooperative load transportation is presented. Three different possibilities for the assignment of the task effort are proposed.

Shared control architectures have been developed for teleoperation of multi-robot systems where the level of complexity demands designing intuitive and effective interaction interfaces. In [72] a survey about advances in human-robot team interaction with special attention devoted to control sharing methodologies is presented. Control sharing design, human behavior modeling, level of autonomy and human-machine interfaces are identified. Several architectures have been proposed in the past following this paradigm [73, 74, 75, 76, 77, 78, 79].

More recently, machine learning and intention estimation-based methods have allowed the development of new forms of control sharing [80]. Rakita *et al.* proposed a control method that moves a bimanual robotic system to mimic the operators arm movements, providing on-the-fly assistance to help the user complete tasks more easily [81]. Learning from demonstration can be used to learn models of the task and to provide contextualized assistance to the operator in a structured manner [82, 83, 84]. This mechanism can be used to online determine the level of autonomy by evaluating the confidence of the automation and of the user [85, 86].

The communication between the human operator and the telerobotic system is a crucial aspect when designing a shared control system. Communication can occur at several levels. The most used form of communication in telerobotics is haptics where the exchange of forces can provide the user with useful information about the status of the robotic system. Besides informing about the system status, haptic guidance can be used to create expert-in-the-loop telerobotic surgical training systems [87]. Shared control using haptic communication has largely demonstrated to be effective for performance enhancement and training in virtual environments [88]. However, it must be properly designed and tuned to be effectively exploited [89, 90, 91].

3.2 Haptic Feedback and Guidance

In telerobotics, haptic feedback is provided to the human operator through *kinesthetic* and *cutaneous* stimuli. Kinesthetic stimuli provide humans with information about the applied forces and torques at the slave side sensed through appropriate sensors. Cutaneous stimuli are felt by mechanoreceptors in the skin and provide the user with information about the local properties of objects such as shape, edges, and texture [92, 93]. A wide variety of haptic devices have been proposed over the years for providing the user with the desired haptic experience of the remote environment. They range from the more conventional kinesthetic and cutaneous devices [94] to the less popular touchable [95] or even ultrasound [96] variations. In addition to information feedback, virtual stimuli, in the form of haptic guidance, are renderable through haptic interfaces.

A particular implementation of haptic guidance-based shared-control architectures is the use of VFs [13, 15, 14]. Virtual elements, such as virtual surfaces, virtual velocity fields, guide tubes, or other appropriate objects, are superimposed to provide contextualized assistance for the user. These fixtures can help the operator to perform tasks by restricting the movement into predefined regions and/or along desired paths. Control is thus shared at the master side, taking advantage of pre-knowledge of the system or of the task to modify the user's commands and/or to combine them with autonomously generated signals [88].

Technically, a VF is described by the geometry and the constraint enforcement method (as explained in Sect. 2.2). The design of effective VFs requires information about the task and the environment in which the task must be performed. This information can be retrieved through vision-based

techniques, force sensors or using a combination of both [97, 98, 99, 100, 101, 102, 103, 104, 105, 106]. One of the greatest limitation of VFs is that they are highly task-dependent and typically inflexible to changes in the environment. This motivated research in the direction of VFs adaption mechanisms [107, 108, 109]. A novel dynamic virtual fixture to enhance the surgical operation accuracy of admittance-type medical robotics in the deforming environment is proposed in [110]. Additionally, virtual constraints have usually been rendered through spring-like forces that accumulate potential energy when the constraint is violated. To prevent the sudden release of such energy, some studies have focused on the development of frictional constraints [111, 112, 113, 109]. Other approaches exist to virtual fixtures rendering, such as hierarchical optimization [114].

In robotic surgery, a robot embedding VFs control (ACROBOT) has been used to accurately machine knee bones to allow the fitting of prosthetic knee implants [115]. On this line, several other works tried to extend the use of this guidance modality to more sophisticated tasks/environments, such as percutaneous needle insertion or suturing, by integrating collision avoidance, anatomy-based constraints and joint limits [116, 117, 118, 119]. The implementation of the virtual fixture assistance on a dVRK and the results of a user study to compare the performance of VFs assistance and freehand teleoperation in both needle passing and knot tying suturing sub-tasks is given in [120]. VFs have been used to construct expertise-oriented training platforms for robotic-assisted minimally invasive robotic surgery [121, 122] or to create virtually stabilized environments for robotic beating heart surgery [123, 124]. Several recent studies highlighted the challenges and the benefits of the use of haptics in robotic surgery [125, 126, 127, 128] and its implications on human factors relevant to cooperative scenarios [129].

Current work is going in the direction of human-robot cooperative control combining features such as active constraints, machine learning and automated movements [84, 130, 131, 132, 133, 134, 135]. The development of advanced sensors that can retrieve environmental information, such as anatomical constraints, has opened the possibility of online interactive VFs generation and control [105, 136, 137, 138, 139].

3.3 Stability and Passivity-based Control

The main issue when designing bilateral teleoperation architectures is the stability of the closed-loop system. In designing the bilateral controller, a classic

engineering trade-off between transparency and stability is unavoidable, since transparency must often be reduced in order to guarantee stable operation in the wide range of environment impedances [140, 141]. Enforcing stability has actually motivated the development of many different control schemes over the past decades, especially to overcome the time delay problem [142]. A thorough survey of control solutions proposed in the literature to counteract these issues is available in [143].

Among others, the passivity-based control paradigm has been widely adopted to design stable teleoperators [144]. Passivity-based control has been used in various forms to guarantee the safe behavior of telerobotic systems. Example of applications include multi-robot systems with multi-task specifications controlled by a human operator [145], variable impedance controllers [146], contact adaptation in unknown environments [147]. Scattering-based controllers render passive the communication channel by emulating the behavior of an electrical lossless transmission line [148, 149]. Several variations of this scheme have been proposed in the literature, *e.g.* adding energy balance monitoring [150]. Damping injection is an essential component in passivity-based control of manipulators that allows obtaining asymptotic stability [151]. However, enforcing a passive behavior may result in over-conservative control design since the closed-loop performance depends on the knowledge of model parameters, whose values are needed in order to find the added damping value.

To overcome this limitation, the energy tanks-based control approach (see Sect. 2.3) has been proposed and adopted in several applications [152, 78, 153, 154]. The goal is to achieve maximal system performance and simultaneously show passive behavior regardless of the operating conditions and uncertainties. To achieve this, the amount of dissipated energy accumulated over time is stored and reused to implement potential passivity violating actions. Based on this concept, the two-layer approach [155] guarantees stable behavior of bilateral teleoperation systems splitting the control architecture into two hierarchical layers: the top layer is used to implement a strategy that addresses the desired transparency, and the lower layer ensures that no virtual energy is generated.

The Passive-Set-Position-Modulation (PSPM) framework is another notable approach that enables to connect the robot position to a sequence of slowly updating/sparse set-position signal via the simple spring coupling with damping injection, while enforcing passivity of the closed-loop robotic system [156]. The PSPM modulates the original set-position signal in such a way

that the modulated signal is as close to the original signal as possible (*i.e.*, maximum information recovery for better performance), yet only to the extent permissible by the available energy in the system (*i.e.*, passivity constraints).

Another approach to enforcing a passive behavior of the teleoperation system while keeping desired system performances is the time domain passivity control approach [157, 158, 159, 160]. The method injects variable damping without any knowledge of model information to reduce conservatism. This idea has been extended via dissipative energy redirection to increase assistive control fidelity [161].

3.4 Limitations

Despite the large number of solutions proposed in the past, to develop a natural and efficient, yet safe and dependable, human-robot interface for shared-control systems is nontrivial.

The key problem in the design of shared-control architectures is the high task dependence: shared-control strategies are usually specific to the application domain and require an accurate model of the task to be performed. Being model-based techniques, changes in the task to be performed require substantial strategy modifications leading to interruptions and tedious setting up times. As such, the development of general sensor-based control methods and algorithms, combining several features, or allowing human-driven strategy correction is highly desired. For this reason, the design of shared control strategies is still an active research field whose aim is the development of the next generation collaborative robotic systems.

Moreover, the challenge of combining the strengths of a human operator (perception, situational awareness, experience, judgment) with the strengths of automation (accuracy, speed, inexhaustibility) requires new interfaces with intuitive communication between humans and robots. Haptic guidance constitutes one of the most promising communication means, however it must be properly designed and tuned to be effectively exploited. In the past, researchers have rarely addressed the evaluation of the user's comfort when operating haptic-based shared-control robots. In this view, how to design an intuitive and efficient assistive haptic interface is still an open question in many shared control applications.

As advanced shared control techniques are being developed, these require novel control methodologies that assure a safe behavior of the system. For instance, the availability of datasets and the leveraging of machine learning tech-

niques will enable, in the near future, enhanced flexibility of shared-control systems that will be capable of providing contextual or personalized assistance and seamless adaption of the autonomy level. However, this desirable trend raises new challenges for safety and stability certification of shared-control robotic systems, thus requiring new advanced control methods to safely implement the continuously evolving division of roles.

Part I

Shared Control Methods for Industrial Scenarios

Chapter 4

Shared Control Architectures for Dual-Arm Robotic Systems

In this chapter, novel haptic guidance methods for a dual-arm telerobotic manipulation system are proposed. These methods inform the user about several different constraints, such as collisions, joint limits, and singularities of the teleoperated system. Haptic guidance is implemented within a shared-control framework for autonomous orientation control and collision avoidance meant to further simplify the execution of grasping tasks. The stability of the overall system in various control modalities is analyzed via passivity arguments. In addition, a human subject study is carried out both in simulated and real scenarios to assess the effectiveness and applicability of the developed control approaches. Results show that the proposed haptic-enabled shared-control methods significantly improve the performance of grasping tasks with respect to the use of classic teleoperation with neither haptic guidance nor shared control.

4.1 Introduction

Over the last few years remote telemanipulation has shown significant advancements and promising results have been achieved in several fields such as minimally-invasive robotic surgery [162], sort and segregation of nuclear waste [2, 163], telemaintenance [164], and micromanipulation [165, 166]. Indeed, being able to remotely, intuitively, and easily manipulate objects in difficult or hazardous scenarios brings a wide range of benefits. In this respect, the nuclear industry is one of the most exciting fields of application for robotic

telemanipulation, where several tasks, such as sort and segregation of nuclear waste, can clearly benefit from advanced telemanipulation techniques. Current telerobotic systems designed for such tasks provide teleoperation capabilities through extremely primitive master consoles (*e.g.*, passive joystick or teach pendants), making these operations prohibitively slow to process large amounts of material in a reasonable time. Besides being time demanding, these tasks usually require highly-skilled human operators. Indeed, steering a remote manipulator toward a desired grasping pose is a quite complex task for an operator directly controlling the 6-DoF pose of a manipulator end-effector. This is due to (i) the complexity of regulating both the position and the orientation at the same time and (ii) the presence of several constraints (*e.g.*, collisions, joint limits, singularities) that further limit the operator's maneuvering dexterity (but of which the operator has no direct or intuitive awareness). To overcome these limitations, the user needs to constantly pay close attention to the status of the robotic system, which can be sometimes difficult and cognitively demanding (*e.g.*, for singularity or joints limit avoidance). A possible way to reduce the operator cognitive and physical workload is to exploit the sensory information collected at the slave side to design novel haptic-guided and shared control-enabled teleoperation systems.

Kinematic limitations avoidance has recently been applied to collaborative robots to improve the performance and the intuitiveness of physical human-robot interaction [167]. In robotic teleoperation this approach can be implemented through haptic guidance methods: instead of providing a high-fidelity haptic feedback, reflecting the actual physical contacts between a slave manipulator and the environment, the human operator is provided with an interface which informs her/him about proximity to constraints. Proposed solutions to this problem are based on potential fields [168], virtual spring-damper systems [169], and/or virtual fixtures [100, 101]. When the slave side is a complex robotic system (*e.g.*, dual-arm, highly redundant) or environmental obstacles are present, (self-)collision avoidance methods need to be considered. In the past, several collision avoidance algorithms have been developed to implement reactive control strategies or to plan collision-free paths for redundant robots [170, 171, 172, 173]. In this regard, self-collisions and joint limits have been used as criteria to find optimal inverse kinematic solutions for robotic manipulators [174, 175]. The idea of combining kinematic constraints and self-collision avoidance to create safe and intuitive haptic-guided telerobotic system has been preliminarily introduced in [176].

On the other hand, several shared-control techniques for telerobotic sys-

tems have been proposed in the past [177, 178, 179]. For instance, in [2], a sensory-based shared-control architecture for remote manipulation is presented: a visual-based autonomous algorithm regulates a subset of the gripper's DoFs to ease the approach toward an object to grasp. At the same time, the operator has control over the remaining null-space motions w.r.t. the primary (autonomous) task and is informed about possible constraints of the single-slave robotic system through haptic cues. However, all these previous works consider one slave arm operating in a static obstacle-free environment.

In this work, the case of a dual-arm robotic system is considered: one robot is (partially) teleoperated by a human, while the other autonomously executes a task in the same workspace. Similar scenarios are considered in [180, 80], where an operator teleoperates a robotic arm while a camera, mounted on a second manipulator, keeps the scene visible. In [180], an occlusion-avoidance algorithm controls the manipulator with the camera and keeps a target object in the field of view. No haptic guidance is considered. In [80], users are provided with haptic feedback about proximity to kinematic constraints of the manipulator they are controlling. However, the two manipulators are far enough to prevent any interference between them.

In this work, two manipulators (one of which is controlled by the operator) have independent and sometimes conflicting tasks with potential overlapping. Moreover, they share a limited workspace, risking dangerous collisions. The primary goal of this paper is to define and assess a suitable haptic shared-control interface for this system which facilitates the operator's task and decrease her/his mental workload in avoiding system constraints. To this end, the the approaches presented in [2] and in [176] are here adapted and improved, combining their features, extending their capabilities, and providing a novel, extensive human subject evaluation and results analysis. Specifically,

- the shared-control approach [2] is improved by (i) dividing the approach-to-grasp phase into multiple parts, according to the gripper distance from the target object, and (ii) devising different, optimized strategies for each of these parts;
- the self-collision-aware approach presented in [176] is improved by (i) also considering collisions with the surrounding environment and (ii) replacing the convex meshes collision model of the robot with discretized sphere volumes;
- the proposed, new above-mentioned techniques are combined to create a novel shared-control framework able to manage a dual-arm system,

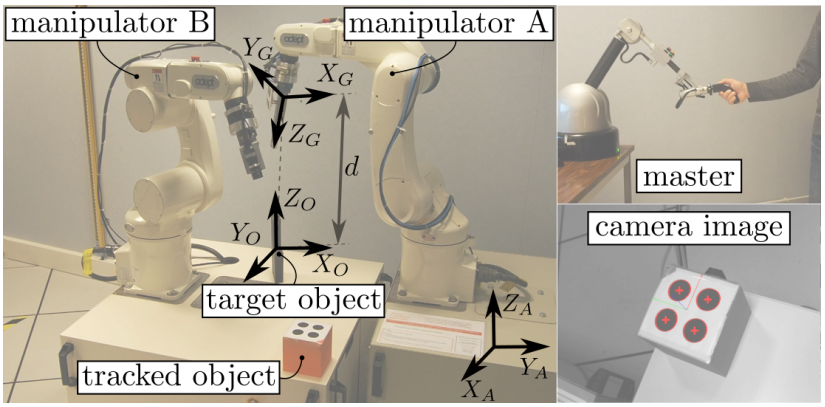


Figure 4.1: System architecture and main reference frames attached to the gripper and the object to be grasped.

where one robot is (partially) controlled by the operator and the other one autonomously performs a task in the same environment;

- a port-Hamiltonian model of the overall system subject to various control modalities is derived, proving the passivity of the system w.r.t. the operator actions;

- an extensive human subject evaluation in simulated and real environments is carried out, enrolling a total of 20 participants.

4.2 System Architecture

The considered telemanipulation system consists of two serial manipulators sharing a workspace in an industrial setting. One of the manipulators is performing an independent fully-autonomous task (in this case, a visual inspection with an onboard camera), while the second manipulator is (partially) teleoperated by a human operator whose aim is to grasp a target object (see Fig. 4.1). The operator faces a number of constraints while controlling such a system: kinematic/workspace constraints for the manipulator she/he is directly controlling, as well as possible interferences with the task performed by the fully-autonomous arm. As explained, the purpose here is to detail and assess a suitable shared-control interface to facilitate the operator's task and decrease her/his mental workload.

4.2.1 System Modeling

With reference to Fig. 4.1, the slave system consists of two 6-DoF velocity-controlled manipulator arms, A and B, with joint configurations $\boldsymbol{q}_A \in \mathbb{R}^6$ and $\boldsymbol{q}_B \in \mathbb{R}^6$, respectively. Manipulator A is (partially) controlled by a human operator, while manipulator B autonomously performs a visual task with a potential overlap with the workspace of manipulator A.

Let

$$\boldsymbol{u}_A = \begin{bmatrix} \boldsymbol{v}_A \\ \boldsymbol{\omega}_A \end{bmatrix} = \boldsymbol{J}_A \dot{\boldsymbol{q}}_A, \qquad \boldsymbol{u}_B = \begin{bmatrix} \boldsymbol{v}_B \\ \boldsymbol{\omega}_B \end{bmatrix} = \boldsymbol{J}_B \dot{\boldsymbol{q}}_B \tag{4.1}$$

be the Cartesian linear/angular velocities of the gripper (for manipulator A) and of the camera (for manipulator B) in their respective frames, and $(\boldsymbol{J}_A, \boldsymbol{J}_B)$ the associated geometric Jacobians. In the following, we will equivalently consider joint velocities $(\dot{\boldsymbol{q}}_A, \dot{\boldsymbol{q}}_B)$ or Cartesian velocities $(\boldsymbol{u}_A, \boldsymbol{u}_B)$ as available control inputs, depending on the context.

The master device is, instead, modeled as a generic, gravity pre-compensated, mechanical system

$$\boldsymbol{M}_m(\boldsymbol{x}_m)\dot{\boldsymbol{u}}_m + \boldsymbol{C}_m(\boldsymbol{x}_m, \boldsymbol{u}_m)\boldsymbol{u}_m + \boldsymbol{B}_m\boldsymbol{u}_m = \boldsymbol{f}_m + \boldsymbol{f}_h, \tag{4.2}$$

where $\boldsymbol{x}_m \in \mathbb{R}^6$ represents the device pose in Cartesian space and $\boldsymbol{u}_m \in \mathbb{R}^6$ the associated linear/angular velocity, $\boldsymbol{M}_m(\boldsymbol{x}_m) \in \mathbb{R}^{6\times6}$ is the positive-definite and symmetric inertia matrix, $\boldsymbol{C}_m(\boldsymbol{x}_m, \boldsymbol{u}_m) \in \mathbb{R}^{6\times6}$ consists of the Coriolis/centrifugal terms, $\boldsymbol{B}_m$ accounts for the friction term, and $\boldsymbol{f}_m, \boldsymbol{f}_h \in \mathbb{R}^6$ are the control and human forces, respectively. The control forces $\boldsymbol{f}_m$ will be exploited to provide force cues to the user meant to inform of the feasibility of her/his commands against the constraints of the slave system (see Sect. 4.2.4).

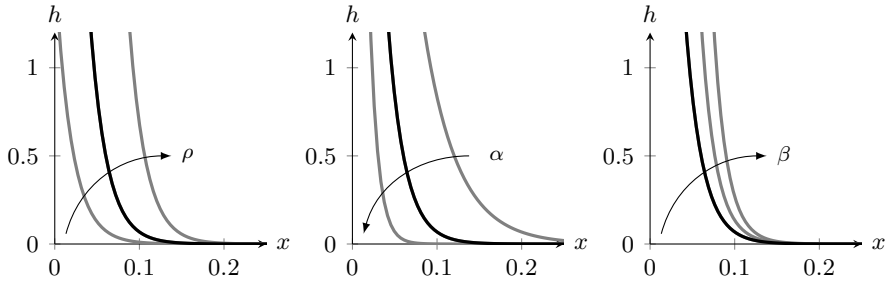
4.2.2 Constraints on the Slave Side

Constraints at the slave side are encoded in a suitable cost function $\mathcal{H}(\boldsymbol{q}_A, \boldsymbol{q}_B)$, whose gradient w.r.t. the joint configuration vectors will be exploited to generate force cues $\boldsymbol{f}_m$ provided to the operator and to implement the reactive behavior of manipulator B (see Sect. 4.2.3). As many constraints are considered, $\mathcal{H}(\boldsymbol{q}_A, \boldsymbol{q}_B)$ would be a sum of many cost functions h. The particular choice of the cost function h is inspired by the work in [174]. A function whose derivative is arbitrarily small when its variable is large and grows to infinity when its variable approaches zero is sought. Among the many

possibilities, the chosen function is in the form

$$h(x) = \rho e^{-\alpha x} x^{-\beta}. \tag{4.3}$$

which allows finer tuning and more flexibility with respect to the commonly-employed polynomial or exponential functions, combining their advantages. By using this function one can separately control the rate of decay and the amplitude near the constraint (in this case $x = 0$) as well as the point at which the cost function becomes flat (*i.e.*, the function derivative vanishes) by opportunely selecting the parameters (α, β, ρ). In Fig. 4.2, the qualitative behavior of the function for several values of the parameters (ρ, α, β) is shown.



(a) (b) (c)

Figure 4.2: Plot of the cost functions and their trends by varying the ρ (4.2a), α (4.2b) and β (4.2c) parameters. The black line represents the solution for $\rho = 10$, $\alpha = 20$, $\beta = 0.01$.

Joint Limits

As customary, the fact that both manipulator arms are subject to joint limits is considered. Among many possibilities, the following cost function to encode proximity to joint limits is adopted

$$h_J(\boldsymbol{q}) = \rho_J \sum_{i=1}^{6} \Big(e^{-\alpha_J (q_{\text{i,max}} - q_i)} \left(q_{\text{i,max}} - q_i\right)^{-\beta_J} + e^{-\alpha_J (q_i - q_{\text{i,min}})} \left(q_i - q_{\text{i,min}}\right)^{-\beta_J} \Big), \tag{4.4}$$

where $\boldsymbol{q}$ is the configuration vector of one of one slave arm, $(q_{\text{i,min}},\ q_{\text{i,max}})$ the i-th min/max joint limits, and $(\rho_J,\ \alpha_J,\ \beta_J)$ are scalar positive constants. The function $h_J(\boldsymbol{q})$ has a minimum at the center of the joint range and grows to infinity as one of the joint approaches its limits.

Singularities

As second constraint, singularities are considered. This constraint is encoded by the cost function [174],

$$h_S(\boldsymbol{q}) = \rho_S e^{-\alpha_S \det(\boldsymbol{\Gamma J}(\boldsymbol{q}))} (\det(\boldsymbol{\Gamma J}(\boldsymbol{q})))^{-\boldsymbol{\beta}_S}, \tag{4.5}$$

where, again, $(\rho_S,\ \alpha_S,\ \beta_S)$ are scalar positive constants, and $\boldsymbol{q}$ and $\boldsymbol{J}$ stand for the joint vector and Jacobian of the manipulator A or B, and $\boldsymbol{\Gamma}$ is a diagonal and positive-definite weight matrix that accounts for non-homogeneous physical dimensions of the Jacobian entries, namely, rotations and translations. The function $h_S(\boldsymbol{q})$ grows to infinity as $\det(\boldsymbol{J}(\boldsymbol{q})) \to 0$ and it vanishes exponentially as $\det(\boldsymbol{J}(\boldsymbol{q})) \to \infty$.

Collisions

As final constraint, collisions between the two slave manipulators are considered. Any cost function encoding the collision avoidance constraint needs to exploit some measure of the minimum distance between the two arms. In order to facilitate the computation of the minimum distance between two bodies, we take inspiration from the line sphere-swept model [181] (see Fig. 4.3a). However, as well known, the minimum distance between two (also convex) bounding volumes can have a discontinuous gradient when the pair of closest points on the two shapes suddenly jumps due to small perturbations of the shape locations. Therefore, the proposed solution consists in discretizing the bounding volumes with a finite number of spheres, and exploiting all the possible inter-sphere distances to build the cost function associated to the collision avoidance (see Fig. 4.3b).

An illustrative example of the resulting collision model is given in Fig. 4.3e, where one can appreciate how the adopted discretized-sphere model represents a fairly good approximation of the manipulator original and convex mesh models (Figs. 4.3c- 4.3d).

Let then $d_{ij}(\boldsymbol{q}_A,\ \boldsymbol{q}_B)$ represent the minimum distance between the i-th sphere on manipulator A and the j-th sphere on manipulator B. The collision avoidance cost function is defined as

$$h_C(\boldsymbol{q}_A,\ \boldsymbol{q}_B) = \rho_C \sum_{i=1}^{S_A} \sum_{j=1}^{S_B} e^{-\alpha_C d_{ij}} d_{ij}^{-\beta_C}, \tag{4.6}$$

where S_A and S_B are the number of spheres used for discretizing the bodies of manipulators A and B, and $(\rho_C,\ \alpha_C,\ \beta_C)$ are scalar positive constants. One

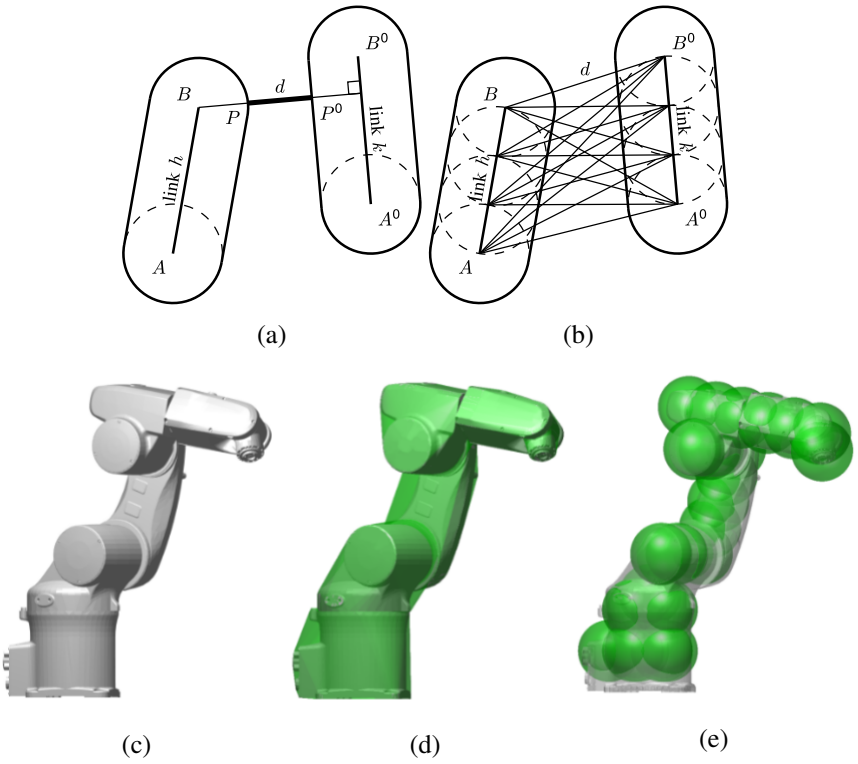


Figure 4.3: Minimum distance between two line sphere-swept bounding volumes (a) and distances among discrete sphere-swept bounding volumes (b). Solid (c), convex (d), and discrete-sphere (e) collision models of the robot.

can easily verify that $h_C(\boldsymbol{q}_A, \boldsymbol{q}_B) \to \infty$ as any of the inter-sphere distances vanishes while $h_C(\boldsymbol{q}_A, \boldsymbol{q}_B) \to 0$ as all the inter-sphere distances become large enough.

Total Cost Function

Finally, the total cost function accounting for all the above mentioned constraints is

$$\mathcal{H}(\boldsymbol{q}_A, \boldsymbol{q}_B) = h_J(\boldsymbol{q}_A) + h_J(\boldsymbol{q}_B) + h_S(\boldsymbol{q}_A) + h_S(\boldsymbol{q}_B) + h_C(\boldsymbol{q}_A, \boldsymbol{q}_B). \quad (4.7)$$

As explained, the gradient of $\mathcal{H}(\boldsymbol{q}_A, \boldsymbol{q}_B)$ w.r.t. the joint vector $\boldsymbol{q}_A$ can be used for cueing the operator about the feasibility of her/his commands against the constraints of the slave side, while the gradient of $\mathcal{H}(\boldsymbol{q}_A, \boldsymbol{q}_B)$ w.r.t. the

joint vector $\boldsymbol{q}_B$ can be used for implementing a reactive behavior in manipulator B for avoiding possible collisions with manipulator A (see Sect. 4.2.3). Note that these two actions (force cues and reactive behavior) are potentially coupled because of the mixed term $h_C(\boldsymbol{q}_A, \boldsymbol{q}_B)$.

4.2.3 Slave Control

The control architecture of manipulator B is here detailed. Manipulator B performs an autonomous visual task to keep an object of interest in visibility during the operation.

Let $\boldsymbol{s} \in \mathbb{R}^2$ represent the image plane location of a representative point on the target object (see Fig. 4.1), $\boldsymbol{s}_d$ a desired value for $\boldsymbol{s}$, and $\boldsymbol{L}_s \in \mathbb{R}^{2\times 6}$ the associated point feature interaction matrix [182], such that $\dot{\boldsymbol{s}} = \boldsymbol{L}_s \boldsymbol{J}_B \dot{\boldsymbol{q}}_B$. Manipulator B is then controlled by employing the usual projected gradient control [183] (hereafter we use $\mathcal{H}$ to indicate $\mathcal{H}(\boldsymbol{q}_A, \boldsymbol{q}_B)$)

$$\dot{\boldsymbol{q}}_B = k_{B1} \underbrace{(\boldsymbol{L}_s \boldsymbol{J}_B)^\dagger (\boldsymbol{s}_d - \boldsymbol{s})}_{\substack{\text{primary task} \\ \text{(visual servoing)}}} - k_{B2} \underbrace{(\boldsymbol{I} - (\boldsymbol{L}_s \boldsymbol{J}_B)^\dagger (\boldsymbol{L}_s \boldsymbol{J}_B)) \nabla_{\boldsymbol{q}_B} \mathcal{H}}_{\substack{\text{secondary task} \\ \text{(minimize } \mathcal{H}\text{)}}}, \tag{4.8}$$

with $k_{B1} > 0$ and $k_{B2} \geq 0$. The primary task consists in the regulation of $\boldsymbol{s}$ toward $\boldsymbol{s}_d$. Its null-space is then exploited for the secondary task of minimizing the constraint cost function $\mathcal{H}$ in (4.7). This null-space action will keep manipulator B away from its singularities and joint limits, as well as maintain a safe distance from manipulator A. This action can be deactivated by setting $k_{B2} = 0$, asking manipulator B to carry out only the primary visual servoing task.

For manipulator A, instead, two possible control modalities are considered, *i.e.*, *Full Teleoperation* and *Shared Control*.

Full Teleoperation

In this modality, the user is given full control over the 6-DoF pose of the gripper on manipulator A. This is achieved by simply setting

$$\boldsymbol{u}_A = k_A \boldsymbol{u}_m, \quad k_A > 0 \tag{4.9}$$

thereby implementing a classic velocity-velocity coupling between master and slave gripper linear/angular velocities.

Shared Control

The shared-control modality is meant to facilitate the user's operation in approaching the object to be grasped by letting an autonomous algorithm control a suitable subset of the gripper pose. This modality is split into two phases depending on the distance between the gripper and the object. With reference to Fig. 4.1, consider the frames $\mathcal{F}_O : \{\boldsymbol{O}_O; \boldsymbol{X}_O, \boldsymbol{Y}_O, \boldsymbol{Z}_O\}$ attached to the object to be grasped and $\mathcal{F}_G : \{\boldsymbol{O}_G; \boldsymbol{X}_G, \boldsymbol{Y}_G, \boldsymbol{Z}_G\}$ attached to the gripper of manipulator A, let ${}^G\boldsymbol{P}_O$ represent the object position in the gripper frame and $d = \|{}^G\boldsymbol{P}_O\|$ the gripper/object distance. When $d < d_{th}$ (threshold value), the shared-control strategy described in [2] is adopted, whose aim is to autonomously keep the pointing direction toward the object

$$\boldsymbol{p} = \frac{{}^G\boldsymbol{P}_O}{\|{}^G\boldsymbol{P}_O\|} \in \mathbb{S}^2 \tag{4.10}$$

always aligned with the gripper approaching direction $\boldsymbol{p}_d = \boldsymbol{Z}_G$, while the operator is in control of the remaining DoFs in the null-space of this primary task. By doing so, the gripper is always oriented toward the object, thus helping the user in completing the task.

This behavior is obtained as follows: defining $\boldsymbol{P}_p = \boldsymbol{I} - \boldsymbol{p}\boldsymbol{p}^T$, $[\cdot]_\times$ the usual skew-symmetric operator, and

$$\boldsymbol{L}_p = \begin{bmatrix} -\frac{1}{d}\boldsymbol{P}_p & [\boldsymbol{p}]_\times \end{bmatrix} \in \mathbb{R}^{3\times 6} \tag{4.11}$$

the interaction matrix associated to the task (4.10) (such that $\dot{\boldsymbol{p}} = \boldsymbol{L}_p\boldsymbol{u}_A$), the following controller is implemented on manipulator A

$$\boldsymbol{u}_A = k_{A1}\boldsymbol{L}_p^\dagger(\boldsymbol{p}_d - \boldsymbol{p}) + k_{A2}\boldsymbol{N}\boldsymbol{\Lambda}\boldsymbol{u}_m \tag{4.12}$$

with $k_{A1} \geq 0$ and $k_{A2} > 0$. Here, $\boldsymbol{N} = [\boldsymbol{n}_1 \dots \boldsymbol{n}_4] \in \mathbb{R}^{6\times 4}$ collects the four null-space directions

$$\boldsymbol{n}_1 = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{O} \end{bmatrix}, \boldsymbol{n}_2 = \begin{bmatrix} \boldsymbol{O} \\ \boldsymbol{p} \end{bmatrix}, \boldsymbol{n}_3 = \begin{bmatrix} -[\boldsymbol{p}]_\times \boldsymbol{e}_y \\ -\boldsymbol{P}_p\boldsymbol{e}_y/d \end{bmatrix}, \boldsymbol{n}_4 = \begin{bmatrix} [\boldsymbol{p}]_\times \boldsymbol{e}_x \\ \boldsymbol{P}_p\boldsymbol{e}_x/d \end{bmatrix}, \tag{4.13}$$

with $\boldsymbol{e}_x = [1\ 0\ 0]^T$ and $\boldsymbol{e}_y = [0\ 1\ 0]^T$, which represent a basis of the four-dimensional null-space of the primary task, and $\boldsymbol{\Lambda} \in \mathbb{R}^{4\times 6}$ is a selection matrix for extracting four components[1] of $\boldsymbol{u}_m$ used by the operator for commanding a 'velocity' along the four motion directions spanned by $\boldsymbol{N}$. Figure 4.4 provides a visual illustration of the four null-space motion directions associated to (4.13), while further details can be found in [2].

[1] In this case, it is assumed that the motion of the master device is blocked (via software) along the two non-selected components.

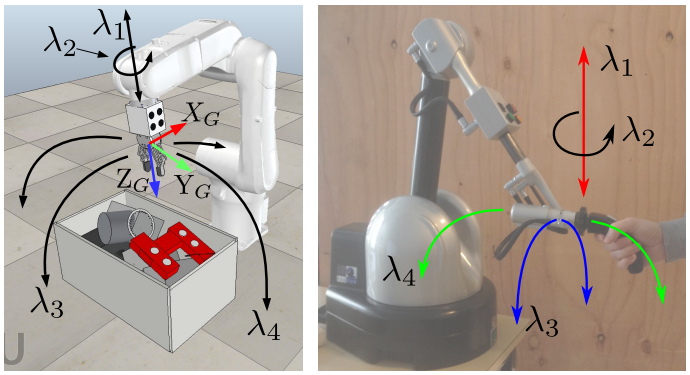


Figure 4.4: A visualization of the four null-space directions defined in (4.13).

On the other hand, when $d \geq d_{th}$, the gripper is not constrained to be oriented toward the object, since, when far from the target object, it would unnecessarily limit the slave maneuverability. Therefore, when $d \geq d_{th}$, the shared-control algorithm switches to a modality in which the operator has control over the *translational* motion of the slave gripper, while the gripper orientation is autonomously controlled so as to minimize the cost function $\mathcal{H}$ and, thus, stay away as much as possible from the system constraints. This is obtained by replacing (4.12) with

$$\boldsymbol{u}_A = k_{A1}\boldsymbol{S}\boldsymbol{u}_m - k_{A2}\boldsymbol{Z}\boldsymbol{J}_A^{-T}\nabla_{\boldsymbol{q}_A}\mathcal{H}, \tag{4.14}$$

where

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I}_3 \end{bmatrix}, \quad \boldsymbol{S} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} \in \mathbb{R}^{6\times 6} \tag{4.15}$$

are selection matrices. The controller (4.14) is indeed able to minimize $\mathcal{H}$ when $\boldsymbol{S}\boldsymbol{u}_m = \boldsymbol{0}$ (*i.e.*, in free translational motion) by acting on the angular velocity of manipulator A. This can be easily shown as follows: by considering variations w.r.t. $\boldsymbol{q}_A$ ($\boldsymbol{q}_B$ is taken care of by (4.8)), one has $\dot{\mathcal{H}} = \frac{\partial^T \mathcal{H}}{\partial \boldsymbol{q}_A}\dot{\boldsymbol{q}}_A = \frac{\partial^T \mathcal{H}}{\partial \boldsymbol{q}_A}\boldsymbol{J}_A^{\dagger}\boldsymbol{u}_A$. By applying (4.14) when $\boldsymbol{v}_m = \boldsymbol{O}$, one has

$$\dot{\mathcal{H}} = -k_{A2}\frac{\partial^T \mathcal{H}}{\partial \boldsymbol{q}_A}\boldsymbol{J}_A^{\dagger}\boldsymbol{Z}(\boldsymbol{J}_A^{\dagger})^T\frac{\partial \mathcal{H}}{\partial \boldsymbol{q}_A},$$

which is a negative semi-definite quadratic form, therefore resulting in $\dot{\mathcal{H}} \leq 0$ as claimed.

4.2.4 Haptic Guidance

As mentioned before, the control forces $\boldsymbol{f}_m$ on the master side (4.2) are exploited to inform the operator about the feasibility of her/his commands against the system constraints, encoded in the cost function $\mathcal{H}$. This is obtained as follows, depending on the control modality considered for manipulator A:

Full Teleoperation

In this case, manipulator A is controlled by (4.9), and the force cues are generated as

$$\boldsymbol{f}_m = -k_M \boldsymbol{J}_A^{-T} \nabla_{\boldsymbol{q}_A} \mathcal{H}, \qquad k_M > 0, \tag{4.16}$$

the rationale being that (4.16) provides the force that, when applied to all the master device DoFs, would make the master move so as to minimize $\mathcal{H}$ at the slave side. Therefore, cues (4.16) provide the human operator with information about where to move in order to stay away from the constraints.

Shared Control

When manipulator A is close enough to the object ($d < d_{th}$), its behavior is controlled by (4.12). In this case, the force cues are generated as

$$\boldsymbol{f}_m = -k_M \boldsymbol{\Lambda}^T \boldsymbol{N}^T \boldsymbol{J}_A^{-T} \nabla_{\boldsymbol{q}_A} \mathcal{H}, \qquad k_M > 0. \tag{4.17}$$

Similarly to the previous case, cues (4.17) represent the forces that should be applied to the free directions of the master device for letting the manipulator A minimize $\mathcal{H}$ along the null-space directions spanned by $\boldsymbol{N}$ (where the operator can act). When manipulator A is instead far from the object ($d \geq d_{th}$), its behavior is regulated by (4.14), and the force cues are

$$\boldsymbol{f}_m = -k_M \boldsymbol{S} \boldsymbol{J}_A^{-T} \nabla_{\boldsymbol{q}_A} \mathcal{H}, \quad k_M > 0, \tag{4.18}$$

with again the idea of providing a force feedback that, when applied to the free master DoF (*i.e.*, the translational ones in this case), would make manipulator A move so as to minimize $\mathcal{H}$.

4.3 Passivity Analysis

Enforcing passivity is very important when dealing with teleoperation systems in order to guarantee a stable closed-loop behavior. Passivity has been extensively studied and applied to haptic feedback teleoperation systems, especially

when time delays are present [143]. This section aims to analyze the passivity of the considered teleoperation system subject to the control modalities introduced in Sects. 4.2.3 and 4.2.4.

By considering the two velocity-controlled robots as simple integrators, the total energy of the system can be written as

$$\mathcal{V}\left(\boldsymbol{p}_m, \boldsymbol{q}_A, \boldsymbol{q}_B\right) = \frac{1}{2}\boldsymbol{p}_m^{\mathrm{T}}\boldsymbol{M}_m^{-1}\boldsymbol{p}_m + \mathcal{H}(\boldsymbol{q}_A, \boldsymbol{q}_B) \tag{4.19}$$

where $\boldsymbol{p}_m = \boldsymbol{M}_m\boldsymbol{u}_m$ is the haptic device momentum, $\boldsymbol{q}_A$, $\boldsymbol{q}_B \in \mathbb{R}^6$ are the generalized coordinates of the two slave manipulators, respectively, and $\mathcal{H}(\boldsymbol{q}_A, \boldsymbol{q}_B)$, introduced in (4.7), is the potential energy associated with the system constraints.

Passivity can thus be analyzed by showing that the closed-loop system equations can be put in port-Hamiltonian form (see Sect. 2.3). In the following, is shown that the three control modalities described in Sect. 4.2 lead to a closed-loop PHS formulation. For the analysis, we consider the primary task in (4.8) and (4.12) to have reached a steady state ($\boldsymbol{s} \to \boldsymbol{s}_d$ and $\boldsymbol{p} \to \boldsymbol{p}_d$).

Full teleoperation

Setting $k_M = k_A = k$, the closed-loop system can be written as follows

$$\begin{bmatrix}\dot{\boldsymbol{p}}_m\\ \dot{\boldsymbol{q}}_A\\ \dot{\boldsymbol{q}}_B\end{bmatrix} = \left(\begin{bmatrix}0 & -k\boldsymbol{J}_A^{-T} & 0\\ k\boldsymbol{J}_A^{-1} & 0 & 0\\ 0 & 0 & 0\end{bmatrix} + \right.$$
$$\left. - \begin{bmatrix}\boldsymbol{B}_m & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & \boldsymbol{P}\end{bmatrix}\right)\begin{bmatrix}\nabla_{\boldsymbol{p}_m}\mathcal{V}\\ \nabla_{\boldsymbol{q}_A}\mathcal{V}\\ \nabla_{\boldsymbol{q}_B}\mathcal{V}\end{bmatrix} + \begin{bmatrix}\mathbf{1}\\ 0\\ 0\end{bmatrix}\boldsymbol{f}_h. \tag{4.20}$$

Shared control ($d < d_{th}$)

Setting $k_M = k_{A2} = k$, the closed-loop system can be written as follows

$$\begin{bmatrix}\dot{\boldsymbol{p}}_m\\ \dot{\boldsymbol{q}}_A\\ \dot{\boldsymbol{q}}_B\end{bmatrix} = \left(\begin{bmatrix}0 & -k\boldsymbol{\Lambda}^T\boldsymbol{N}^T\boldsymbol{J}_A^{-T} & 0\\ k\boldsymbol{J}_A^{-1}\boldsymbol{N}\boldsymbol{\Lambda} & 0 & 0\\ 0 & 0 & 0\end{bmatrix} + \right.$$
$$\left. - \begin{bmatrix}\boldsymbol{B}_m & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & \boldsymbol{P}\end{bmatrix}\right)\begin{bmatrix}\nabla_{\boldsymbol{p}_m}\mathcal{V}\\ \nabla_{\boldsymbol{q}_A}\mathcal{V}\\ \nabla_{\boldsymbol{q}_B}\mathcal{V}\end{bmatrix} + \begin{bmatrix}\mathbf{1}\\ 0\\ 0\end{bmatrix}\boldsymbol{f}_h. \tag{4.21}$$

Shared control ($d \geq d_{th}$)

Setting $k_M = k_{A1} = k$, the closed-loop system can be written as follows

$$\begin{bmatrix} \dot{\boldsymbol{p}}_m \\ \dot{\boldsymbol{q}}_A \\ \dot{\boldsymbol{q}}_B \end{bmatrix} = \left(\begin{bmatrix} 0 & -k\boldsymbol{S}\boldsymbol{J}_A^{-T} & 0 \\ k\boldsymbol{J}_A^{-1}\boldsymbol{S}^{\mathrm{T}} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \right.$$
$$\left. - \begin{bmatrix} \boldsymbol{B}_m & 0 & 0 \\ 0 & k_{A2}\boldsymbol{J}_A^{-1}\boldsymbol{Z}\boldsymbol{J}_A^{-T} & 0 \\ 0 & 0 & \boldsymbol{P} \end{bmatrix} \right) \begin{bmatrix} \nabla_{\boldsymbol{p}_m}\mathcal{V} \\ \nabla_{\boldsymbol{q}_A}\mathcal{V} \\ \nabla_{\boldsymbol{q}_B}\mathcal{V} \end{bmatrix} + \begin{bmatrix} \mathbf{1} \\ 0 \\ 0 \end{bmatrix} \boldsymbol{f}_h, \tag{4.22}$$

where $\boldsymbol{P} = \left(\boldsymbol{I} - (\boldsymbol{L}_s\boldsymbol{J}_B)^{\dagger}(\boldsymbol{L}_s\boldsymbol{J}_B)\right)$ is a null-space projector such that $\boldsymbol{P} = \boldsymbol{P}^T \geq 0$.

Therefore, since in all cases the closed-loop systems can be put in a PHS form (being the resulting interconnection and dissipation matrices always skew-symmetric and positive semi-definite, respectively), one can conclude passivity of the three modalities w.r.t. the pair $(\boldsymbol{u}_m, \boldsymbol{f}_h)$ with energy function $\mathcal{V}(\boldsymbol{p}_m, \boldsymbol{q}_A, \boldsymbol{q}_B)$ as sought[2].

4.4 Experiments in Simulation

4.4.1 Experimental Setup, Task, and Participants

The experimental setup is shown in Fig. 4.5. The slave side is simulated using V-REP [24], and it is composed of two velocity-controlled manipulators: an Adept Viper 850, controlled by the human operator (manipulator A), and an Adept Viper 650, controlled by an autonomous algorithm (manipulator B) (see Sect. 2.5 for their kinematic model). The master side is composed of an Haption Virtuose 6D Desktop interface, which is used to control the Viper 850 robot and provides kinesthetic haptic feedback. Manipulator A is endowed with a ROBOTIQ 2-finger gripper, while manipulator B is equipped with a vision sensor. The remote environment is composed of two objects, a cube and a parallelepiped, placed on a conveyor belt. As detailed in Sect. 4.2, the autonomous manipulator B robot is in charge of tracking the parallelepiped using

[2]Note that the presented analysis does not account for the controller switching in the shared-control modality when close/far from the target object. If this switch needs to be taken into account because of non-negligible effects on the total energy, one could employ the energy tank machinery for passifying potential instabilities due to the switching mechanism [77]. However, in this case, it was empirically possible to see that the switching has a negligible effect on the system stability during the experiments.

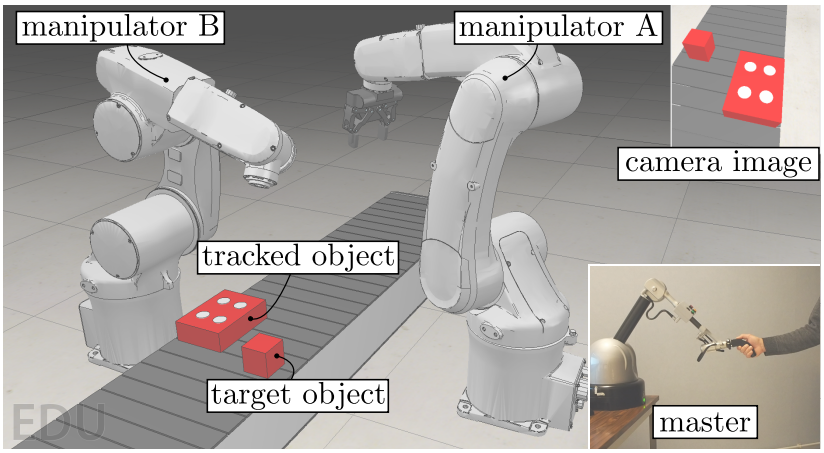


Figure 4.5: Experimental setup used for user study: slave side composed of simulated robots and vision; master side comprising the haptic interface.

standard visual servoing techniques [28]. On the other hand, participants are required to control the motion of manipulator A to grasp the cube and lift it from the ground, avoiding collisions with the fully-autonomous robot.

The task starts when the manipulator moves for the very first time, and it is considered successfully completed when the object is lifted from the ground. All the control policies described are implemented in MATLAB/SIMULINK and interfaced with V-REP using ROS through the matlab_ros_bridge (https://github.com/lagadic/matlab_ros_bridge). The control loop runs at 100 Hz. A video of the experiment is available as supplemental material and at https://youtu.be/uWzVEGW2i5c.

Fifteen right-handed subjects (average age 25.9, 11 males, 4 females) participated in the study. Four of them had previous experience with haptic interfaces. Each subject spent about ten minutes practicing the control of the telemanipulation system before starting the experiment. Participants were briefed about all the tasks and afterward signed an informed consent, including the declaration of having no conflict of interest. All of them were able to give the consent autonomously. The study was done in accordance with the 1964 Helsinki declaration and its later amendments.

4.4.2 Experimental Conditions

Two different levels of human involvement in the control of manipulator A (teleoperation vs. shared control, T vs. S), two haptic feedback modalities (haptic feedback about the distance from the second robot and workspace/joints constraints vs. no haptic feedback, H vs. $\not{\text{H}}$), and two control policies for manipulator B (reactive vs. non-reactive, R vs. $\not{\text{R}}$) are considered.

Human Involvement (T vs. S)

In condition T, subjects are able to control all the 6 DoFs of the manipulator A through the grounded master interface (see Sect. 4.2.3). Conversely, in condition S, the orientation of the gripper is controlled by an autonomous algorithm. When the gripper is close to the object to grasp, it is automatically oriented towards it; otherwise, the gripper is oriented to stay as much as possible away from the system constraints (see Sect. 4.2.3).

Haptic Feedback (H vs. $\not{\text{H}}$)

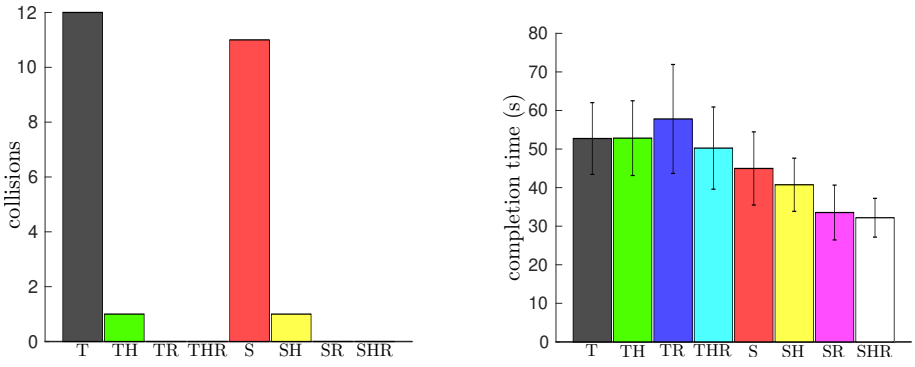
When haptic feedback is activated (H), subjects receive haptic stimuli about the feasibility of their commands against system constraints, such as joint limits, singular configurations, and collisions with manipulator B (see Sect. 4.2.4). Conversely, in condition $\not{\text{H}}$, subjects do not receive any haptic feedback.

Control of the Fully-Autonomous Manipulator (R vs. $\not{\text{R}}$)

Manipulator B always performs an autonomous visual task to keep an object of interest visible.

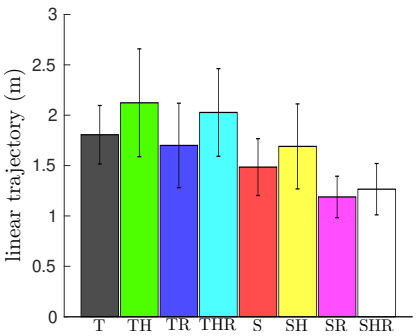
In condition R, it also uses the null-space of the above-mentioned primary visual task to minimize the constraint cost function $\mathcal{H}(\boldsymbol{q}_A, \boldsymbol{q}_B)$ (see Sect. 4.2.3). Notably, this secondary action will enable manipulator B to keep a safe distance from manipulator A, re-actively moving away when the latter comes too close. Conversely, in $\not{\text{R}}$, manipulator B only focuses on the primary visual task, regardless what the other manipulator does. Considering all the possible combinations, we ended up with eight different experimental conditions: THR, TH$\not{\text{R}}$, T$\not{\text{H}}$R, T$\not{\text{H}}\not{\text{R}}$, SHR, SH$\not{\text{R}}$, S$\not{\text{H}}R, S\not{\text{H}}\not{\text{R}}$. For brevity, from now on the $\not{\text{H}}$ and $\not{\text{R}}$ variables will be omitted (*e.g.*, S$\not{\text{H}}\not{\text{R}}$ is denoted by S).

The cost functions introduced in Sect. 4.2 include several parameters, which let control the curvature, rate of increase, and proximity to the limits

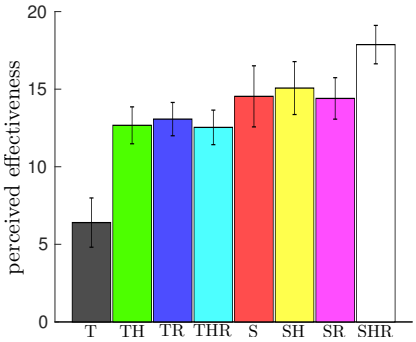


(a) Total number of collisions.

(b) Completion time (s).



(c) Linear trajectory (m).



(d) Perceived effectiveness.

Figure 4.6: Experimental evaluation. Mean and 95% confidence interval of (a) total number of collisions, (b) completion time, (c) linear motion, and (d) perceived effectiveness of the eight feedback conditions are plotted.

of these functions (see Fig. 4.2). The choice of these parameters is challenging, system-dependent, and very delicate, as it must ensure a smooth force feedback. To choose the right parameters for our system and target application, we asked 2 expert operators to repeatedly carry out the task, changing the parameters at runtime (*i.e.*, $\rho_{\mathrm{X}}, \alpha_{\mathrm{X}}, \beta_{\mathrm{X}}, k_{A1}, k_{A2}, k_{B1}, k_{B2}, k_M$) to make the teleoperation as intuitive, safe, and comfortable as possible. Finally, we asked them to find a consensus on the parameters' values and we used those in our implementation.

4.4.3 Results

Average task success rate across conditions was $92.4 \pm 6.3\%$. A Friedman test showed no statistically significant difference between the means of the eight feedback conditions. Figure 4.6a shows the total number of collisions occurred during the experiment between the two manipulators.

To compare other metrics, we ran three-way repeated-measures ANOVA tests on the data. Human involvement in the control (shared control vs. teleoperation, S vs. T), presence of haptic feedback (haptic feedback vs. no haptic feedback, H vs. $\not{\text{H}}$), and behavior of the fully-autonomous robot (reactive vs. non-reactive, R vs. $\not{\text{R}}$) were treated as within-subject factors. All data passed the Shapiro-Wilk normality test. Sphericity was assumed for all variables, since they all have two levels of repeated measures. Interaction effects between the factors were not statistically significant. Figure 4.6b shows the completion time.

The three-way ANOVA test revealed a statistically significant change in the metric for the human involvement in the control (F(1, 14) = 52.165, $p < 0.001$, shared control was better) and the control behavior for manipulator B (F(1, 14) = 6.400, $p = 0.024$, reactive mode was better) variables. Figure 4.6c shows the linear motion covered by the robotic gripper during the task. The three-way ANOVA test revealed a statistically significant change in the metric for the human involvement in the control (F(1, 14) = 13.599, $p = 0.002$, shared control was better) and the control behavior for the second robot (F(1, 14) = 6.567, $p = 0.023$, reactive mode was better) variables. Immediately after the experiment, subjects were also asked to report the effectiveness of each feedback condition in completing the given task using bipolar Likert-type twenty-two-point scales. Figure 4.6d shows the perceived effectiveness for the eight experimental conditions. The three-way ANOVA test revealed a statistically significant change in the metric for the human involvement in the control (F(1, 14) = 34.700, $p < 0.001$), the presence of haptic feedback (F(1, 14) = 33.217, $p < 0.001$, shared control was better), and the control behavior for the second robot (F(1, 14) = 25.305, $p < 0.001$, reactive mode was better) variables.

Finally, all fifteen subjects found conditions using the shared-control approach to be the most effective at completing the grasping task. Ten subjects chose SHR as the most effective, three SH, and two SR.

4.5 Experiment in a Real Environment

An experiment in the real environment is also carried out. The setup and task are similar to the simulated scenario of Sect. 4.4. The setup is shown in Fig. 4.1. Five right-handed subjects participated in the study. Three of them had previous experience with haptic interfaces. In this real-world case, control modalities TR, THR, SR, and SHR are implemented, which led to no collisions in Sect. 4.4. Of course, in this real-world case, is not possible to consider conditions which may lead to a collision between the two manipulators. All subjects successfully completed the task in all conditions and no collisions occurred. All subjects chose SHR to be the most intuitive and effective condition. A video of the experiment in the SHR condition (shared control, haptic feedback, and reactive control of manipulator B) is available as supplemental material and at https://youtu.be/uWzVEGW2i5c. A sequence of frames taken from the video is shown in Fig. 4.7, where it is possible to recognize all the relevant phases of the grasping task. In snapshots (a)-(b), it is possible to see the manipulator B reactively avoiding collisions while keeping track of the desired object. In snapshots (c)-(d), the shared-control algorithm is active and automatically orients the gripper of manipulator A toward the object to grasp. From the bottom graphs, we can see that the first half of the experiment is characterized by persistent haptic forces and by a significant reactive velocity of manipulator B. In the second half, the risk of collisions is lower and the operator can safely approach the object, aided by the action of the shared-control algorithm.

4.6 Discussion and Conclusion

A shared-control framework with haptic feedback, which enables the intuitive and effective teleoperation of a dual-arm robotic system, is presented in this work. The proposed architecture is tested in simulated and real environments using two 6-DoF robotic manipulators. The human operator controls one arm (manipulator A), equipped with a gripper, through a 6-DoF grounded haptic interface. Haptic feedback provides the operator with information about joint and workspace limits as well as about the presence of singular configurations and imminent collisions. The shared-control algorithm autonomously controls 2 DoFs of the robotic manipulator, orienting the gripper toward the object to grasp. The other robotic arm (manipulator B) is equipped with a camera, and it moves autonomously to track a second object, placed near the one to grasp.

To demonstrate the feasibility and effectiveness of our methods, a human subject experiment in the simulated scenario was conducted. Fifteen subjects controlled manipulator A to grasp an object placed on a conveyor belt. Eight experimental conditions were tested, considering two levels of human involvement in the control (shared control vs. classic teleoperation, S vs. T), two feedback modalities (haptic feedback about imminent collisions and workspace/joints constraints vs. no haptic feedback, H vs. $\not$H), and two control policies for manipulator B (reactively moving away from the other manipulator vs. non-reactive/still, R vs. $\not$R). Results proved the effectiveness and viability of our haptic-enabled shared-control approaches. Using shared control (S) on manipulator A and the reactive mode (R) on manipulator B significantly improved the performance in most metrics (completion time, linear motion, perceived effectiveness). Conditions employing shared control were also the most preferred, confirming the all-round viability of such approach. Moreover, as expected, in conditions R, manipulator B was always able to prevent collisions with the other robot by moving away when the latter was approaching. Nonetheless, even in conditions $\not$R, repulsive haptic feedback (H) provided when the robots were too close showed good results (only two collisions happened in conditions TH, SH). This result is very promising, as haptic feedback acts only at the master side, leaving the action of manipulator B unaffected. In this respect, it is interesting to notice that, although the applied force $\boldsymbol{f}_m$ should go to infinite as the distance between the two robots goes to zero (see eq. (4.6)), we still experienced two collisions in conditions H. This is due to the limited actuation capabilities of our haptic interface, which is obviously not able to provide arbitrarily high forces. Finally, the experiment in the real scenario confirmed the results obtained with the simulated setup. Haptic guidance effectively steered the user toward the safe zones of the workspace, the reactive behavior enabled a safe interaction between manipulators, and the shared control made the task fast, easy, and intuitive to complete.

The proposed shared-control architecture is independent from the distribution of DoFs between the autonomous controller and the human operator. While having the autonomous controller keep the gripper oriented toward the object seemed a natural choice for this work, the underlying shared-control architecture is independent from the task, the primary task variables, and the level of human involvement. For the future, would be interesting to study how the number of DoFs controlled autonomously affects the task performance, *e.g.*, a system could use a highly-autonomous shared-control approach (*i.e.*, many DoFs managed by the autonomous controller) when it is operated

by novices, while it could implement a lowly-autonomous shared-control approach (*i.e.*, few DoFs managed by the autonomous controller) when it is operated by experts. This flexible approach could be also useful when teaching new operators, employing different levels of autonomy according to the operator's experience.

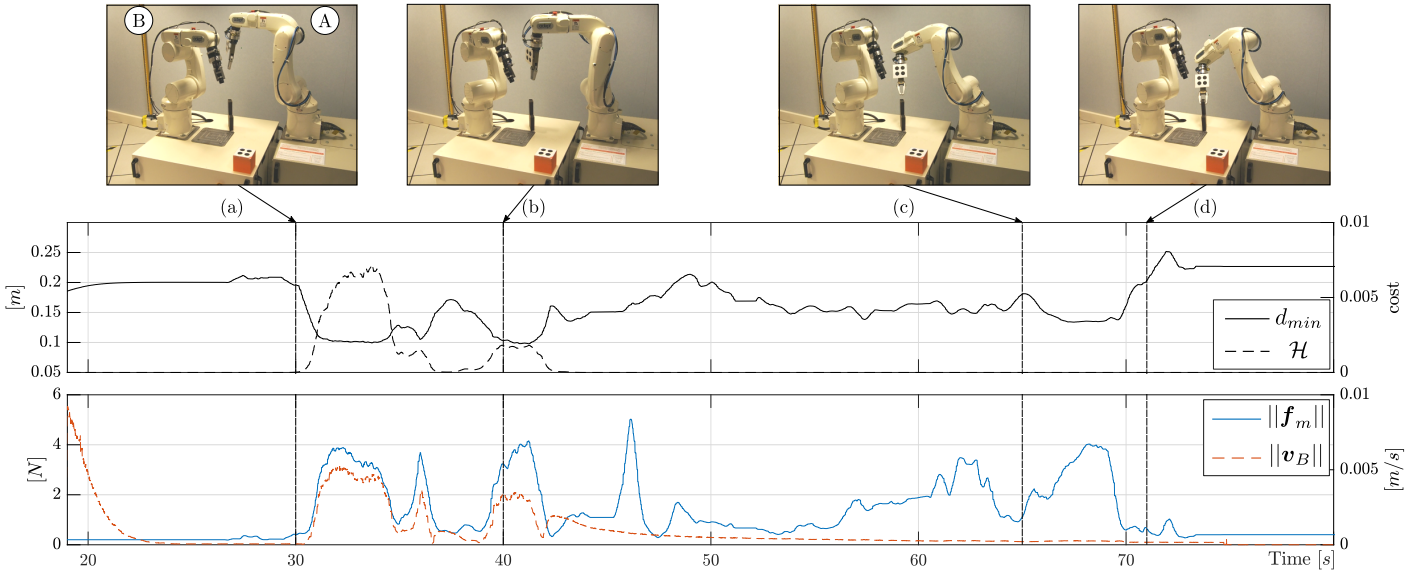


Figure 4.7: Data time history recorded during the real experiments. Upper graph: minimum distance d_{min} and collision cost value $\mathcal{H}$; bottom graph: haptic guidance $\boldsymbol{f}_m$ and escaping velocity $\boldsymbol{v}_B$ norms.

Chapter 5

A Task-Prioritized Shared Control Architecture

Robot teleoperation is widely used for several hazardous applications. To increase teleoperator capabilities shared-control methods can be employed. In this chapter, a passive task-prioritized shared-control method for remote telemanipulation of redundant robots is presented [184]. The proposed method fuses the task-prioritized control architecture with haptic guidance techniques to realize a shared-control framework for teleoperation systems. To preserve the semi-autonomous telerobotic system safety, passivity is analyzed and an energy-tanks passivity-based controller is developed. The proposed theoretical results are validated through experiments involving a real haptic device and a simulated slave robot.

5.1 Introduction

Autonomous robots are still far from being reliably and safely employed in hazardous and/or unstructured environments. Indeed, the current robots level of autonomy is still too low to have them effectively dealing with complex and non-structured situations. A possible solution to this problem is to rely on advanced teleoperation systems that implement *shared-control* techniques. Shared-control telerobotic systems are those in which a human operator shares the control of the slave robot with an autonomous controller to achieve a common goal. The aim of using an autonomous controller is to reduce the operator physical and mental workload when controlling a complex (e.g., highly redundant) robotic system [185]. In a larger sense, shared control methods try to

combine the intelligence of human operators and their sense of presence with safe and optimized task execution of autonomous control.

When highly redundant robots (*e.g.*, mobile manipulators, dual-arm systems, humanoids) are employed at the remote site, the slave robotic system can ensure the achievement of several additional tasks (possibly having different priorities) which may be autonomously executed. In this case, the most typical choice is to rely on a task-prioritized control architecture [186, 187, 18] and implement null-space projections [188] for the simultaneous execution of multiple hierarchical tasks (see Sect. 2.4). Additionally, it is fundamental for the user to receive an appropriate sensory feedback informing about the feasibility of her/his commands against the slave system constraints. To achieve this, a haptic guidance method, which informs the operator about constraints acting on the teleoperation system, needs to be designed.

For such a devised system, it is of paramount importance to guarantee a passive behaviour even during the autonomous regulation of different priority tasks. Enforcing passivity is equivalent to guarantee that the energy accumulated in the system is never larger than the power supplied through the interaction over time plus any initial stored energy (see Sect. 2.3). Classical passivity-based control approaches allow enforcing a passive behavior but they can result in an overly conservative control system design. To overcome this limitation, a less restrictive passivity-preserving control method is preferable.

In this chapter, a bilateral teleoperation architecture for a task-prioritized slave robot is presented and the possible loss of passivity that may arise when operating such a system is illustrated. A passivity-based controller built upon the energy-tanks approach [155] is developed and validated through experiments involving a real haptic device and a simulated slave robot.

5.2 Related Works

Related works involve two main fields.

Shared Control

The idea of shared control in telerobotics has appeared in various forms. For instance, it has been used to compensate for beating heart movements in robotic surgical interventions [189], to effectively avoid system constraints in dual-arm telerobotic systems [180, 190] or to combine decision making capabilities of a human operator with the high flexibility of a robot swarm [76].

More recently, the shared autonomy paradigm has started to be investigated [191, 192]: optimization techniques are used to infer user's goals, which are used, in turn, to provide assistance. The problem of semi-autonomous teleoperation of a redundant slave robot has been studied under communication delays in [193, 194]. The authors considered a number of kinematic constraints but only numerical simulations are shown. In this chapter, the aim is to devise a novel shared-control strategy that autonomously regulates a subset of the redundant slave robot DoFs while leaving the user control over the remaining ones. This is accomplished by using the task-prioritized control approach [183] in which the user input constitutes one of the tasks.

Haptic feedback and guidance have been shown to play a significant and promising role in shared-control applications [195, 88]. Haptic cues can be used to increase situation awareness [196] and/or to effectively steer the human operator towards the safe execution of some tasks. A possible choice to generate haptic guidance is to use virtual fixtures, *i.e.* software generated constraints that help the operator to follow a predefined path and/or avoid forbidden regions [197, 100, 101, 198]. In this chapter, haptic guidance is realized through the Gradient Projection Method [199] of opportunely defined scalar cost functions and is used to inform the user about constraints acting on the slave robotic system.

Passivity-based Control

Few authors have investigated stability issues caused by shared-control techniques in telerobotics. Passivity-based approaches consider the system energy flow to derive sufficient but sometimes unnecessary conditions for stability [158]. For this reason, enforcing passivity can result in overly conservative parameters design, leading to degraded performances of the system.

A state-dependent approach based on the concept of *energy tanks* can be exploited to overcome this limitation [154]. The energy-tanks passivity-based control method has been used to enforce passivity of variable impedance controllers [200, 201], to stably adapt to contacts in unknown environments [147] or to switch the system topology in robot swarms [77]. This approach is particularly suitable for teleoperation since it can activate its passifying action only when necessary, significantly improving the control performance when compared to fixed-structure passivity-enforcing controllers.

In the context of task-prioritized control, the projectors modify the energy flows in the system and destroy the precious and safety-critical property of passivity [202, 203]. Here this property is restored by developing an energy-

tanks-based method suitable for a task-prioritized shared-control teleoperation architecture with haptic guidance.

5.3 System Description

A telemanipulation system, which consists of a master haptic device and a slave robotic manipulator operating in an industrial scenario, is considered. The operator is asked to carry out some tasks by teleoperating the slave robot via the haptic device. A camera is placed at the slave side to provide visual feedback to the operator as well as to recover the object/gripper relative pose. A number of kinematic/workspace constraints acting on the slave side are considered (*e.g.*, joint limits, singularities, collisions) and the operator is provided with haptic guidance to effectively avoid them.

In the following, the master and slave robotic system models as well as their interconnection are described.

5.3.1 Master System

Let $\boldsymbol{q}_m \in \mathbb{R}^m$ be the master manipulator generalized coordinates vector and m denote its dimension. The master is a gravity pre-compensated robotic device with the following joint space dynamics

$$\boldsymbol{M}_m(\boldsymbol{q}_m)\ddot{\boldsymbol{q}}_m + \boldsymbol{C}_m(\boldsymbol{q}_m, \dot{\boldsymbol{q}}_m)\dot{\boldsymbol{q}}_m + \boldsymbol{B}_m\dot{\boldsymbol{q}}_m = \boldsymbol{\tau}_m + \boldsymbol{\tau}_h, \tag{5.1}$$

where $\boldsymbol{M}_m(\boldsymbol{q}_m) \in \mathbb{R}^{m\times m}$ is the positive-definite and symmetric joint space inertia matrix, $\boldsymbol{C}_m(\boldsymbol{q}_m, \dot{\boldsymbol{q}}_m) \in \mathbb{R}^{m\times m}$ consists of the Coriolis/centrifugal terms, $\boldsymbol{B}_m \in \mathbb{R}^{m\times m}$ accounts for the friction term, and $\boldsymbol{\tau}_m, \boldsymbol{\tau}_h \in \mathbb{R}^m$ are the input torques corresponding to master/slave coupling method (see Sect. 5.3.3) and human operator interaction forces, respectively. The control forces $\boldsymbol{\tau}_m \in \mathbb{R}^m$ are exploited to provide force cues meant to inform the user about the feasibility of her/his commands against slave system constraints.

5.3.2 Slave System

The slave robot is a shared-control manipulator with $\boldsymbol{q}_s \in \mathbb{R}^n$ being the n-dimensional vector of its generalized coordinates. The slave is modeled as an industrial manipulator with a joint velocity interface that allows to directly command $\dot{\boldsymbol{q}}_s \in \mathbb{R}^n$. For convenience, these velocity commands are partitioned as follows

$$\dot{\boldsymbol{q}}_s = \dot{\boldsymbol{q}}_{s,a} + \dot{\boldsymbol{q}}_{s,u} \tag{5.2}$$

where $\dot{\boldsymbol{q}}_{s,a} \in \mathbb{R}^n$ contains joint velocities corresponding to a certain number of autonomous operational space tasks, and $\dot{\boldsymbol{q}}_{s,u} \in \mathbb{R}^n$ represents joint velocities corresponding to operational space user inputs, respectively.

Let us consider the presence of an autonomous task-prioritized controller that accomplishes the primary task while trying to fulfill a number of secondary tasks. In this case, the autonomous control input can be specified as follows

$$\dot{\boldsymbol{q}}_{s,a} = \sum_{i=1}^{r} \boldsymbol{\Lambda}_i \boldsymbol{P}_{i-1} \boldsymbol{J}_{s,i}^{\dagger} \tilde{\boldsymbol{\sigma}}_i \tag{5.3}$$

where $\tilde{\boldsymbol{\sigma}}_i = \boldsymbol{\sigma}_i^* - \boldsymbol{\sigma}_i \in \mathbb{R}^{r_i}$ is the i-th task space error and r_i denotes its dimension, such that $\dot{\boldsymbol{\sigma}}_i = \boldsymbol{J}_{s,i}\dot{\boldsymbol{q}}_s$ where $\boldsymbol{J}_{s,i} \in \mathbb{R}^{r_i \times n}$ denotes the corresponding Jacobian matrix and $\boldsymbol{J}_{s,i}^{\dagger} \in \mathbb{R}^{n \times r_i}$ its Moore-Penrose pseudoinverse, *i.e.* $\boldsymbol{J}_{s,i}^{\dagger} = \boldsymbol{J}_{s,i}^{\mathrm{T}}(\boldsymbol{J}_{s,i}\boldsymbol{J}_{s,i}^{\mathrm{T}})^{-1}$, $\boldsymbol{\Lambda}_i \in \mathbb{R}^{n \times n}$ is the diagonal and positive-definite $i-$th task gain matrix, r is the number of tasks (autonomously executed) and $\boldsymbol{P}_i \in \mathbb{R}^{n \times n}$ the $i-$th null-space projector (with $\boldsymbol{P}_0 = \boldsymbol{I}_n$ being the $n \times n$ identity matrix). There exist several methods to compute $\boldsymbol{P}_i$: it can be obtained from the augmented Jacobian of all higher priority tasks, *i.e.*

$$\boldsymbol{J}_{s,1,\ldots,i} = \begin{bmatrix} \boldsymbol{J}_{s,1} \\ \vdots \\ \boldsymbol{J}_{s,i} \end{bmatrix} \qquad \boldsymbol{P}_i = \left(\boldsymbol{I}_n - \boldsymbol{J}_{s,1,\ldots,i}^{\dagger}\boldsymbol{J}_{s,1,\ldots,i}\right), \tag{5.4}$$

or by successive projections, *i.e,*

$$\boldsymbol{P}_i = \left(\boldsymbol{I}_n - \boldsymbol{J}_{s,i}^{\dagger}\boldsymbol{J}_{s,i}\right), \quad \boldsymbol{P}_i = \boldsymbol{P}_1 \ldots \boldsymbol{P}_{i-1}. \tag{5.5}$$

A more comprehensive overview of task-prioritized control methods, as well as their stability properties, is given in [18]. Additional details are given in Sect. 2.4.

On the other hand, the user-specified control velocity $\dot{\boldsymbol{q}}_{s,u}$ is given by the master/slave coupling method and is detailed in the following section.

5.3.3 Coupling Method

The proposed master/slave bilateral interconnection considers the human operator's input as an additional task for the slave manipulator while, at the same time, haptic guidance forces are fed back to avoid system constraints. Without loss of generality, in this work is chosen to set the operator 'task' as the one with lowest priority. Clearly, other design choices would be possible depending on the role the user input has to play in the overall task accomplishment.

The proposed coupling is realized by connecting the two systems with the following velocity-force interconnection

$$\dot{\boldsymbol{q}}_{s,u} = \boldsymbol{P}_r \boldsymbol{J}_{s,u}^{\dagger} \boldsymbol{Z}_c \boldsymbol{R}_c \boldsymbol{\Lambda}_m \boldsymbol{J}_m \dot{\boldsymbol{q}}_m, \tag{5.6}$$

$$\boldsymbol{\tau}_m = -\boldsymbol{J}_m^{\mathrm{T}} \boldsymbol{\Lambda}_s \boldsymbol{R}_c^{\mathrm{T}} \boldsymbol{Z}_c^{\mathrm{T}} \left(\boldsymbol{J}_{s,u}^{\dagger} \right)^{\mathrm{T}} \boldsymbol{P}_r^{\mathrm{T}} \nabla \mathcal{H} \left(\boldsymbol{q}_s \right), \tag{5.7}$$

where $\boldsymbol{\Lambda}_m \in \mathbb{R}^{6\times6}, \boldsymbol{\Lambda}_s \in \mathbb{R}^{6\times6}$ are diagonal and positive-definite interconnection gain matrices, $\boldsymbol{J}_m \in \mathbb{R}^{6\times m}$ is the master manipulator Jacobian matrix, $\boldsymbol{J}_{s,u} \in \mathbb{R}^{z\times n}$ is the Jacobian matrix associated with the teleoperation task, $\boldsymbol{Z}_c \in \mathbb{R}^{z\times6}$ is a selection matrix used to extract z components of the master device output velocity, $\boldsymbol{R}_c \in \mathbb{R}^{6\times6}$ is the block-diagonal coupling rotation matrix used to transform velocity commands from the master to the slave end-effector frames, and $\mathcal{H}\left(\boldsymbol{q}_s\right) : \mathbb{R}^n \to \mathbb{R} \geq 0$ is a cost function encoding the proximity to slave system constraints. Its gradient $\nabla\mathcal{H} \in \mathbb{R}^n$ is used to provide haptic cues (through (5.7)), meant to inform the user where to move in order to effectively avoid system constraints.

Note that, when adding (5.6) to (5.3), the operator's commands $\dot{\boldsymbol{q}}_m$ are correctly considered as the lowest priority task thanks to the projector $\boldsymbol{P}_r$ in (5.6). Furthermore, the coupling method given by (5.6) and (5.7) represents a power preserving interconnection if we design $\boldsymbol{\Lambda}_m = \boldsymbol{\Lambda}_s$, since it satisfies the relation $\dot{\boldsymbol{q}}_m^{\mathrm{T}} \boldsymbol{\tau}_m + \dot{\boldsymbol{q}}_{s,u}^{\mathrm{T}} \nabla\mathcal{H} = 0$. This property will be exploited in the passivity analysis given in the next section.

5.4 Passivity Analysis

In this section, the system passivity when subject to the proposed control inputs is analyzed. The standard definition of passivity given in Def. 1 is used. Considering the system described by (5.1), (5.3), (5.6) and (5.7) a possible choice for the storage function is the total energy of the system $\mathcal{V} : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R} \geq 0$, *i.e.*

$$\mathcal{V}\left(\boldsymbol{q}_m, \boldsymbol{q}_s\right) = \frac{1}{2} \dot{\boldsymbol{q}}_m^{\mathrm{T}} \boldsymbol{M}_m \dot{\boldsymbol{q}}_m + \mathcal{H}(\boldsymbol{q}_s) + \frac{1}{2} \sum_{i=1}^{r} \tilde{\boldsymbol{\sigma}}_i^{\mathrm{T}} \tilde{\boldsymbol{\sigma}}_i \tag{5.8}$$

which accounts for the master kinetic energy (first term) and for the constraint and task error 'energies' (second and third terms). The time derivative of $\mathcal{V}$ in (5.8) is given by (assuming $\boldsymbol{\sigma}_i^*$ constant)

$$\dot{\mathcal{V}} = \dot{\boldsymbol{q}}_m^{\mathrm{T}} \boldsymbol{M}_m \ddot{\boldsymbol{q}}_m + \frac{1}{2} \dot{\boldsymbol{q}}_m^{\mathrm{T}} \dot{\boldsymbol{M}}_m \dot{\boldsymbol{q}}_m + \dot{\boldsymbol{q}}_s^{\mathrm{T}} \nabla\mathcal{H} - \sum_{i=1}^{r} \tilde{\boldsymbol{\sigma}}_i^{\mathrm{T}} \dot{\boldsymbol{\sigma}}_i. \tag{5.9}$$

Substituting $\boldsymbol{M}_m\ddot{\boldsymbol{q}}_m$ from (7.19) and using the usual skew symmetry of the matrix $\left(\dot{\boldsymbol{M}}_m - 2\boldsymbol{C}_m\right)$, we further obtain

$$\dot{\mathcal{V}} = -\dot{\boldsymbol{q}}_m^{\mathrm{T}}\boldsymbol{B}_m\dot{\boldsymbol{q}}_m + \dot{\boldsymbol{q}}_m^{\mathrm{T}}\left(\boldsymbol{\tau}_m + \boldsymbol{\tau}_h\right) + \dot{\boldsymbol{q}}_s^{\mathrm{T}}\nabla\mathcal{H} - \sum_{i=1}^{r}\tilde{\boldsymbol{\sigma}}_i^{\mathrm{T}}\dot{\boldsymbol{\sigma}}_i. \tag{5.10}$$

Exploiting the operational-joint space velocity mapping $\dot{\boldsymbol{\sigma}}_i = \boldsymbol{J}_{s,i}\dot{\boldsymbol{q}}_s$, substituting $\dot{\boldsymbol{q}}_s = \dot{\boldsymbol{q}}_{s,a} + \dot{\boldsymbol{q}}_{s,u}$ from (5.3) and (5.6) we get

$$\begin{aligned}\dot{\mathcal{V}} = &- \dot{\boldsymbol{q}}_m^{\mathrm{T}}\boldsymbol{B}_m\dot{\boldsymbol{q}}_m + \dot{\boldsymbol{q}}_m^{\mathrm{T}}\boldsymbol{\tau}_m + \dot{\boldsymbol{q}}_m^{\mathrm{T}}\boldsymbol{\tau}_h + \dot{\boldsymbol{q}}_{s,a}^{\mathrm{T}}\nabla\mathcal{H} + \dot{\boldsymbol{q}}_{s,u}^{\mathrm{T}}\nabla\mathcal{H} + \\ &- \sum_{k=1}^{r}\tilde{\boldsymbol{\sigma}}_k\boldsymbol{J}_{s,k}\left(\dot{\boldsymbol{q}}_{s,a} + \dot{\boldsymbol{q}}_{s,u}\right).\end{aligned} \tag{5.11}$$

Using the power-preserving property of the interconnection (5.6)–(5.7) discussed before, and noting that $\sum_{k=1}^{r}\tilde{\boldsymbol{\sigma}}_k\boldsymbol{J}_{s,k}\dot{\boldsymbol{q}}_{s,u} = 0$ as $\dot{\boldsymbol{q}}_{s,u}$ contains the null-space of all higher priority tasks, we further obtain

$$\begin{aligned}\dot{\mathcal{V}} = &- \underbrace{\dot{\boldsymbol{q}}_m^{\mathrm{T}}\boldsymbol{B}_m\dot{\boldsymbol{q}}_m}_{\geq 0} + \underbrace{\dot{\boldsymbol{q}}_m^{\mathrm{T}}\boldsymbol{\tau}_h}_{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{u}} + \underbrace{\left(\sum_{i=1}^{r}\boldsymbol{\Lambda}_i\boldsymbol{P}_{i-1}\boldsymbol{J}_{s,i}^{\dagger}\tilde{\boldsymbol{\sigma}}_i\right)^{\mathrm{T}}}_{=w}\nabla\mathcal{H} + \\ &- \underbrace{\sum_{k=1}^{r}\tilde{\boldsymbol{\sigma}}_k\boldsymbol{J}_{s,k}\left(\sum_{i=1}^{r}\boldsymbol{\Lambda}_i\boldsymbol{P}_{i-1}\boldsymbol{J}_{s,i}^{\dagger}\tilde{\boldsymbol{\sigma}}_i\right)}_{\geq 0}.\end{aligned} \tag{5.12}$$

Referring to (5.12), the term $\dot{\boldsymbol{q}}_m^{\mathrm{T}}\boldsymbol{B}_m\dot{\boldsymbol{q}}_m$ is positive-definite and represents the power dissipated by the system; $\dot{\boldsymbol{q}}_m^{\mathrm{T}}\boldsymbol{\tau}_h$ is the input-output variables product and represents the power injected into or extracted from the system by the human operator interaction; the last term can be shown to be positive semi-definite (see [18]) by assuming that the tasks in (5.3) are *orthogonal*, *i.e.* $\boldsymbol{J}_{s,i}\boldsymbol{J}_{s,j}^{\dagger} = \boldsymbol{O}_{r_i\times r_i}\ \forall j > i \in \{1,\ldots,r\}$ (for the successive projection method given by (5.5)), or at least *linear independent*, *i.e.* not orthogonal but such that $\rho(\boldsymbol{J}_{s,i}^{\dagger}) + \rho(\boldsymbol{J}_{s,j}^{\dagger}) = \rho([\boldsymbol{J}_{s,i}^{\dagger}\ \boldsymbol{J}_{s,j}^{\dagger}])\ \forall i,j \in \{1,\ldots,r\}$ (for the augmented inverse-based Jacobian approach given by (5.4)) and gains matrices are positive definite.

The second term on the right-hand side of (5.12) can be compactly written as

$$w = \sum_{i=1}^{r} w_i, \quad w_i = \tilde{\boldsymbol{\sigma}}_i^{\mathrm{T}}(\boldsymbol{J}^{\dagger})_{s,i}^{\mathrm{T}}\boldsymbol{P}_{i-1}^{\mathrm{T}}\boldsymbol{\Lambda}_i^{\mathrm{T}}\nabla\mathcal{H}. \tag{5.13}$$

Each w_i represents the energetic coupling between the i-th autonomously regulated task and the constraints cost function $\mathcal{H}(\boldsymbol{q}_s)$. In general, the sign of each w_i cannot be determined a priori, since it depends on the effect that the regulation of the i-th task would make on $\mathcal{H}(\boldsymbol{q}_s)$ (increase or decrease). For instance, the regulation of a task could make the slave robot moving closer to a joint limit/singularity, thereby increasing $\mathcal{H}(\boldsymbol{q}_s)$ or viceversa. As a consequence, passivity of the system (that in our case would imply $\dot{\mathcal{V}} \leq \dot{\boldsymbol{q}}_m^{\mathrm{T}} \boldsymbol{\tau}_h$) cannot be easily guaranteed because of the arbitrary sign of the overall term $w = \sum w_i$, and closed-loop stability of the proposed teleoperation system could be threaten at runtime. The next section will then detail a suitable passivation strategy able to deal with this shortcoming.

5.5 Passivity Enforcing Via Energy Tanks

The energy tank machinery is employed to prevent a potential loss of passivity due to the w term in (5.12). More specifically, we add to the system dynamics an energy storing element whose aim is to recover the system dissipated energy and to use it to implement potentially 'active' actions (induced by the terms $w_i > 0$ in (5.12)) while preserving the system passivity. In more details, the tank element has the following storage function

$$\mathcal{T}(z) = \frac{1}{2} z^2, \tag{5.14}$$

with $z \in \mathbb{R}$ being the state of the tank whose dynamics is chosen as

$$\dot{z} = \frac{\varphi}{z} P_d - \frac{1}{z} \sum_{i=1}^{r} \gamma_i w_i \tag{5.15}$$

where $P_d = \dot{\boldsymbol{q}}_m^{\mathrm{T}} \boldsymbol{B}_m \dot{\boldsymbol{q}}_m \geq 0$ represents the power dissipated by the system, w_i can have any sign and represents the power extracted/injected by potential passivity violating actions, and φ and $\gamma_1, \ldots, \gamma_r \in [0, 1]$ are binary variables meant to guarantee a finite upper bound for the energy stored in the tank (see also [156]). With the choice in (5.15) the time derivative of (5.14) is given by

$$\dot{\mathcal{T}} = z\dot{z} = \varphi P_d - \sum_{i=1}^{r} \gamma_i w_i. \tag{5.16}$$

The tank action on the slave side dynamics is implemented through a set of variables $\alpha_1, \ldots, \alpha_r \in [0, 1]$ used to activate or deactivate tasks based on the

energy available in the tank. This mechanism is obtained by replacing (5.2–5.3) with

$$\dot{\boldsymbol{q}}_s = \dot{\boldsymbol{q}}^{\alpha}_{s,a} + \dot{\boldsymbol{q}}_{s,u}\dot{\boldsymbol{q}}^{\alpha}_{s,a} = \sum_{i=1}^{r} \alpha_i \boldsymbol{\Lambda}_i \boldsymbol{P}_{i-1} \boldsymbol{J}^{\dagger}_{s,i} \tilde{\boldsymbol{\sigma}}_i, \tag{5.17}$$

that is, by weighting each term in $\dot{\boldsymbol{q}}^{\alpha}_{s,a}$ by the activation parameter α_i. The associated power balance then becomes

$$\dot{\mathcal{V}} = -\dot{\boldsymbol{q}}_m^{\mathrm{T}} \boldsymbol{B}_m \dot{\boldsymbol{q}}_m + \dot{\boldsymbol{q}}_m^{\mathrm{T}} \boldsymbol{\tau}_h + \sum_{i=1}^{r} \alpha_i w_i - \sum_{k=1}^{r} \tilde{\boldsymbol{\sigma}}_k \boldsymbol{J}_{s,k} \dot{\boldsymbol{q}}^{\alpha}_{s,a}. \tag{5.18}$$

The extended system energy function is expressed by $\mathcal{G} = \mathcal{V} + \mathcal{T}$ and its time derivative takes the form (using (5.16-5.18))

$$\dot{\mathcal{G}} = \dot{\mathcal{V}} + \dot{\mathcal{T}} = -(1-\varphi)\dot{\boldsymbol{q}}_m^{\mathrm{T}} \boldsymbol{B}_m \dot{\boldsymbol{q}}_m + \dot{\boldsymbol{q}}_m^{\mathrm{T}} \boldsymbol{\tau}_h + \sum_{i=1}^{r} (\alpha_i - \gamma_i) w_i - \sum_{k=1}^{r} \tilde{\boldsymbol{\sigma}}_k \boldsymbol{J}_{s,k} \dot{\boldsymbol{q}}^{\alpha}_{s,a} \tag{5.19}$$

where $\dot{\boldsymbol{q}}^{\alpha}_{s,a}$ is given in (5.17). By denoting with $\bar{\mathcal{T}}$ and $\underline{\mathcal{T}}$ the tank energy upper and lower thresholds, respectively, we define $\forall i \in 1, \ldots, r$ the following switching policy for φ, γ_i

$$\varphi = \begin{cases} 1 & \text{if } \mathcal{T} < \bar{\mathcal{T}} \\ 0 & \text{otherwise} \end{cases} \qquad \gamma_i = \begin{cases} 0 & \text{if } \mathcal{T} \geq \bar{\mathcal{T}} \;\&\; w_i < 0 \\ \alpha_i & \text{otherwise} \end{cases} . \tag{5.20}$$

At this point, α_i can be chosen as any sufficiently smooth function[1] of the tank state $\mathcal{T}$ and of the input power w_i that satisfies $\alpha_i = 0$ if $\mathcal{T} <= \underline{\mathcal{T}}$ and $w_i > 0$, and $\alpha_i = 1$ when $\mathcal{T} = \bar{\mathcal{T}}$. In this work, this mechanism is implemented by choosing

$$\alpha_i = f\left(\mathcal{T}, \underline{\mathcal{T}}, \bar{\mathcal{T}}\right) g\left(w_i, \underline{w}_i\right) + \left(1 - g\left(w_i, \underline{w}_i\right)\right), \tag{5.21}$$

where

$$\begin{aligned} f\left(\mathcal{T}, \underline{\mathcal{T}}, \bar{\mathcal{T}}\right) &= \frac{1}{2}\left(1 - \cos\left(\pi \frac{\mathcal{T} - \underline{\mathcal{T}}}{\bar{\mathcal{T}} - \underline{\mathcal{T}}}\right)\right) \\ g\left(w_i, \underline{w}_i\right) &= \begin{cases} 0 & \text{if } w_i \leq \underline{w}_i \\ 1 & \text{if } w_i > 0 \\ f(w_i, \underline{w}_i, 0) & \text{otherwise} \end{cases} , \end{aligned} \tag{5.22}$$

and $\underline{w}_i < 0$ is a threshold on input power. With the choice in (5.20), (5.21) and (5.22), the extended system described by (5.1), (5.6), (5.7), (5.15) and (5.17) is

[1] Continuity of each α_i w.r.t. its arguments is, indeed, needed for preserving continuity of the joint velocity command $\dot{\boldsymbol{q}}^{\alpha}_{s,a}$ in (5.17).

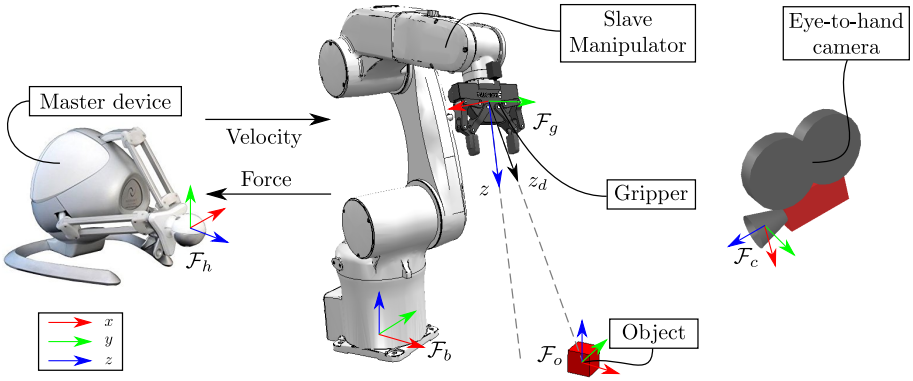


Figure 5.1: Experimental setup for task-prioritized shared-control teleoperation of a 6-DoF manipulator: z and z_d denote the current and desired end-effector frame vectors for the shared-control object grasping taken from [2].

passive w.r.t. the input-output pair $(\boldsymbol{\tau}_h, \dot{\boldsymbol{q}}_m)$ with storage function $\mathcal{G} = \mathcal{V} + \mathcal{T}$. It is possible to verify that (5.19) satisfies (2.3) in all conditions that can occur when implementing (5.20–5.21). Indeed, the term $(\alpha_i - \gamma_i)$ that multiplies w_i in (5.19) is always null for $w_i > 0$ regardless of the tank state. Furthermore, the only case in which $\alpha_i \neq 0$ and $\gamma_i = 0$ occurs when $w_i < 0$ and is used to guarantee the energy upper bound limitation. In all other cases, the choices in (5.20) allow implementing potentially non-passive actions using the energy stored in the tank while enforcing its upper $(\overline{\mathcal{T}})$ and lower $(\underline{\mathcal{T}})$ bounds.

5.6 Experiments

5.6.1 Experimental Setup

A telemanipulation system consisting of a haptic device and a 6-DoF serial manipulator in an industrial scenario endowed with a camera is considered (Fig. 5.1). An operator is asked to grasp an object by teleoperating the task-prioritized shared-control arm through the master system. The camera, placed at the slave side in a calibrated eye-to-hand configuration, provides visual feedback to the operator as well as the object pose in the robot base frame through standard vision-based techniques [28]. The considered tasks are:

- autonomous regulation of y coordinate of the robot end-effector frame $(\mathcal{F}_g)$ to y_d corresponding to the y object position $(\mathcal{F}_o)$ in the robot base frame $(\mathcal{F}_b)$ (1 DoF);

- autonomous orientation control around a sphere centered in the object (2 DoFs) plus alignment of the y axis between $\mathcal{F}_g$ and $\mathcal{F}_o$ (1 DoF). The approach is taken from [2].

The user input along the remaining directions (2 DoFs) constitutes the additional lowest priority task.

The experimental setup is composed of a simulated Adept Viper 850 (see Sect. 2.5 for its kinematic model) equipped with a Robotiq 2-finger gripper as end-effector. The master is a Force Dimension Omega 6 in which only 2 linear DoFs of the device are used to control the slave manipulator end-effector position (2 DoFs). In all the experiments, we use the successive projection method and numerically verify the successive tasks orthogonality.

Any constraint of the system can be modeled through the function $\mathcal{H}$. Without loss of generality, here joint limits and singularities at the slave side are considered and haptic guidance is provided to the operator to effectively avoid them. Constraints are encoded through the following functions (see [190, 176] for more details)

$$\mathcal{H}\left(\boldsymbol{q}_s\right) = \mathcal{H}_j\left(\boldsymbol{q}_s\right) + \mathcal{H}_s\left(\boldsymbol{q}_s\right) \tag{5.23}$$

$$\mathcal{H}_j\left(\boldsymbol{q}_s\right) = \sum_{i=1}^{n} \frac{1}{\lambda} \frac{\left(q_{s,i}^{+} - q_{s,i}^{-}\right)^2}{\left(q_{s,i}^{+} - q_{s,i}\right)\left(q_{s,i} - q_{s,i}^{-}\right)} \tag{5.24}$$

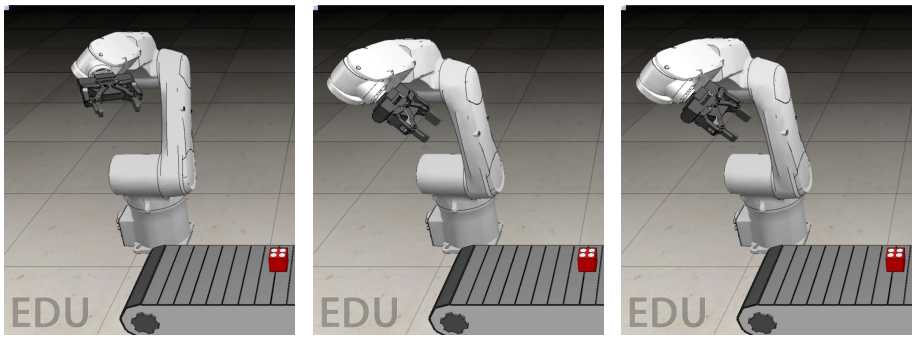
$$\mathcal{H}_s\left(\boldsymbol{q}_s\right) = \rho e^{-\varepsilon \det\left(\mathbb{J}\left(\boldsymbol{q}_s\right)\right)} \left(\det\left(\mathbb{J}_s\left(\boldsymbol{q}_s\right)\right)\right)^{-\beta}, \tag{5.25}$$

where λ, ρ, ε and β are scalar constants, $q_{s,i}$ is the $i-$th joint coordinate, $q_{s,i}^{+}$ and $q_{s,i}^{-}$ are the upper and the lower limits, respectively. The corresponding gradients have been calculated symbolically using the MATLAB Symbolic Toolbox. All the parameters used in the following experiments are listed in Tab. 7.1 while joint limits are given in Tab. 7.2.

The object position is retrieved through the blob tacker and planar homography functionalities provided by ViSP software library [28]. The control loop runs at 200 Hz.

Table 5.1: Parameters

Parameter	Value	Parameter	Value
$\boldsymbol{\Lambda}_1$	diag(0.5)	$\boldsymbol{\Lambda}_2$	diag(2.0)
$\boldsymbol{\Lambda}_m$	diag(0.7)	$\boldsymbol{\Lambda}_s$	diag(0.7)
$\underline{\mathcal{T}}$	0.2	$\bar{\mathcal{T}}$	0.02
λ	50	ρ	0.0005
ϵ	1.0	β	2.0
$\mathbb{M}_m$	diag(0.4171)	$\mathbb{B}_m$	diag(2.0)



(a) (b) (c)

Figure 5.2: Autonomous task regulation experiment. (a) Initial system configuration and (b) final robot configurations without and (c) with the energy-tanks passivity-based control.

5.6.2 Results

First the system evolution when the shared control is activated without any user input (*i.e.*, $\dot{\boldsymbol{q}}_m = \mathbf{0}$, $\boldsymbol{\tau}_m = \mathbf{0}$) is shown and the system responses with and without the passivity-based control action are compared. The slave robot autonomously executes the two above-mentioned tasks. Figure 5.2 contains snapshots of the slave robot initial and final configurations from the proposed experiment. The response of the system in terms of energies, task errors, control variables and forces that would have been fed back to the user, is shown in Fig. 5.3 with (right column) and without (left column) the energy-tanks passivity-based control.

As it can be noticed from the upper graph, when passivity control is not activated, the system undergoes an active time period (yellow shaded area) in which the total energy (given by (5.8)) increases. The task errors $\tilde{\boldsymbol{\sigma}}_i, i = 1, 2$ (on the second row) are brought to zero but the forces $\boldsymbol{f}_m = (\boldsymbol{J}_m^{\mathrm{T}})^{\dagger}\boldsymbol{\tau}_m$ that the user would have experienced if the bilateral teleoperation were activated (third row) reach a peak around $3s$ corresponding to a maximum value of 34.02 N along the y direction. Such an intensive force can cause abrupt user's move-

Table 5.2: Adept Viper Joint Limits $[deg]$

Joint #	1	2	3	4	5	6
q^+	170	120	120	170	120	170
q^-	-170	-120	-120	-170	-120	-170

ments that may guide the slave robot to encounter severe collisions or other dangerous configurations leading, eventually, to unstable behaviours.

On the other hand, the energy-tanks passivity-based control establishes a passive behavior of the system (total energy always decreasing in the upper graph), at the expense of larger errors in the tasks regulation (second row). Indeed, the tasks executions are stopped when the energy tank gets discharged according to the control laws in (5.20) and (5.21). The corresponding force intensities (third row) are overall bounded by lower and more acceptable values (in this case ± 5 N). The last row of Fig. 5.3 contains the plots of α_i and w_i recorded during the experiment.

Then, a bilateral teleoperation experimental test is performed using the presented haptic shared-control architecture. In this experiment the energy-tanks passivity-based control is always active. Figure 5.4 shows the most significant instants of the experiment while Figure 5.5 contains the time history of relevant quantities.

The experiment starts with the autonomous controller regulating the two higher priority tasks as in the previous case (Fig. 5.4a). Looking at the graphs in Fig. 5.5, autonomous actions increase the constraints cost function while the tank level drops down (analogously to the previous experiment). More specifically, this is caused by the second task driving the fifth slave robot joint towards its upper limit. The user experiences haptic guidance forces that try to keep the slave robot away from its constraints. The errors tend to zero during the subsequent time period.

The experiment continues with the user driving the slave robot toward the target object (Fig. 5.4b). The damping acting on the system makes the tank energy increase towards its upper bound. Around 20 s the slave robot approaches the object in a configuration close to the third joint lower limit. A similar situation happens around 32 s, when the operator moves around the object before grasping it. In these cases, the cost function increases but the first two tasks have reached steady-state, thus the system is passive with respect to the user actions. This leaves the tank energy level unchanged. Haptic guidance forces help the operator to correct the slave robot configuration around the grasping point.

To further demonstrate the capabilities of the proposed method, the target object is moved along the y direction by activating the conveyor belt around 38 s (Fig. 5.4c). Correspondingly, an increase in the second task error norm is registered (red shaded area in Fig. 5.5). The tank energy level drops down again and the second task execution is slowed down. The user starts again tele-

operating and, following the haptic guidance forces, drives the robot towards the new object position. The energy tank gets replenished by both the dissipated power and passive actions while the second task error approaches zero around 41 s. The grasping task is successfully accomplished around 63 s as shown in Fig. 5.4d.

Both experiments are also shown in the following video: https://youtu.be/abzHTFRjISQ.

5.7 Conclusions

In this chapter the concept of task-prioritized shared-control for remote teleoperation of redundant robots is introduced. The proposed method is endowed with haptic guidance used to steer the slave manipulator away from its kinematic constraints. A power-preserving coupling is designed, teleoperation system passivity analysis is provided and a passification strategy using energy-tanks passivity-based control is proposed. The passivity control action dynamically stops the tasks that drive the slave robot in dangerous workspace configurations, thus preventing the user from feeling large/abrupt haptic forces. The system is evaluated in remote manipulation experiments performed with a real haptic device and a simulated slave robot. Results show that the methodology effectively lets the operator accomplishing the task while ensuring safety of the telerobotic systems.

As future works the aim is to endow the method with task priority switching capabilities that can make the teleoperation system even more flexible and performing.

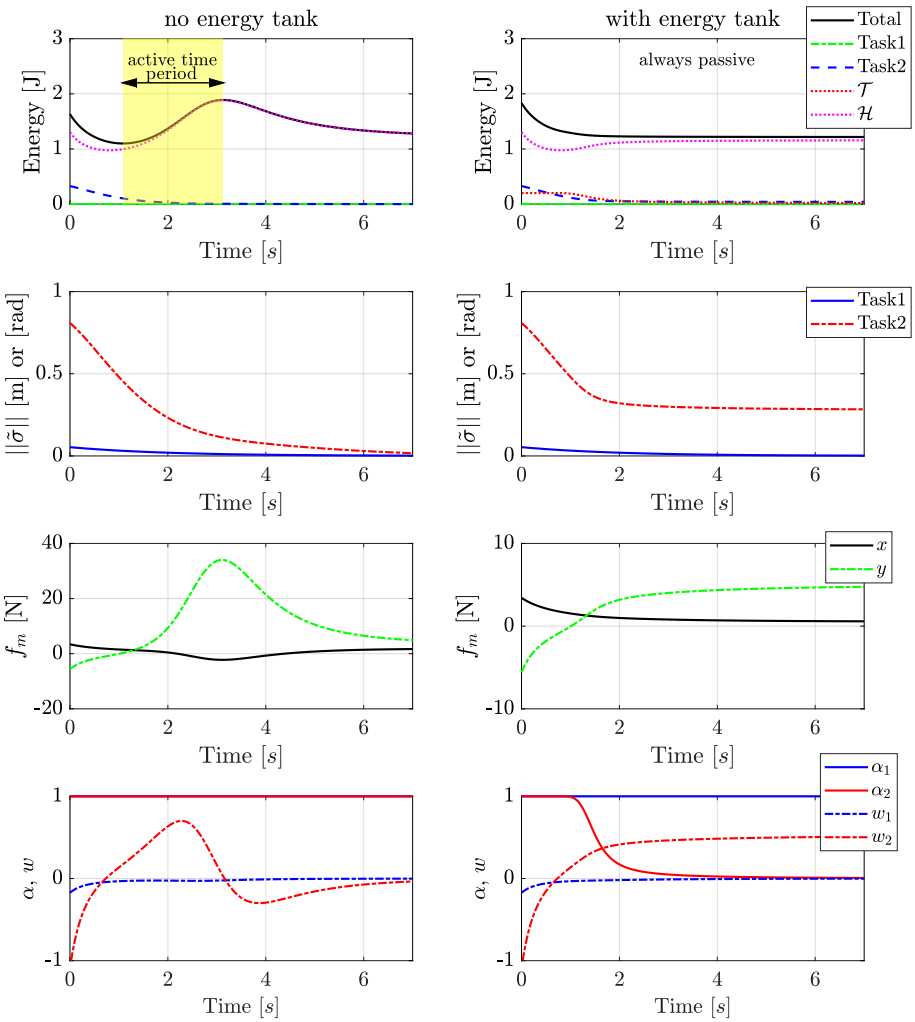
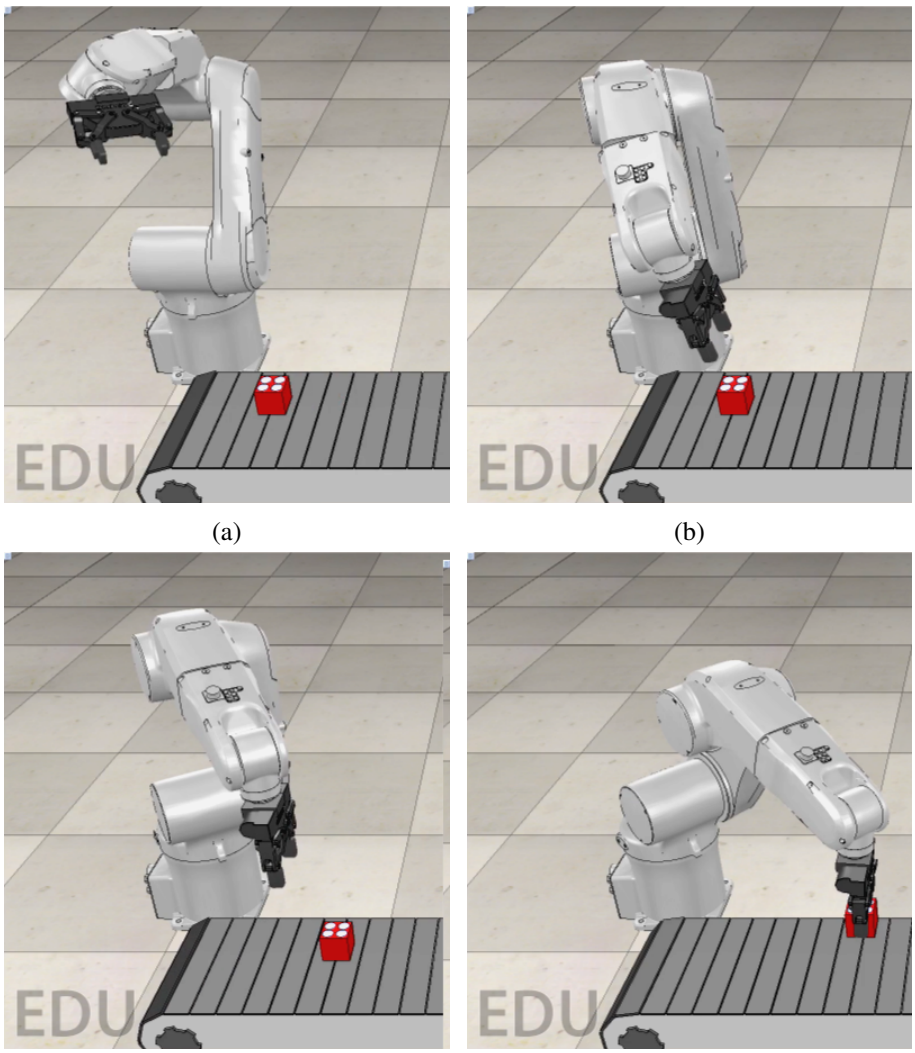


Figure 5.3: Comparison of energies, error norms, forces and control variables during the autonomous tasks regulation without (left) and with energy-tanks (right). The yellow shaded area indicates the active time period.



(a) (b)

(c) (d)

Figure 5.4: Shared-control teleoperation experiment. (a) Initial system configuration; (b) robot approaching the object; (c) object moving; (d) robot grasping the object.

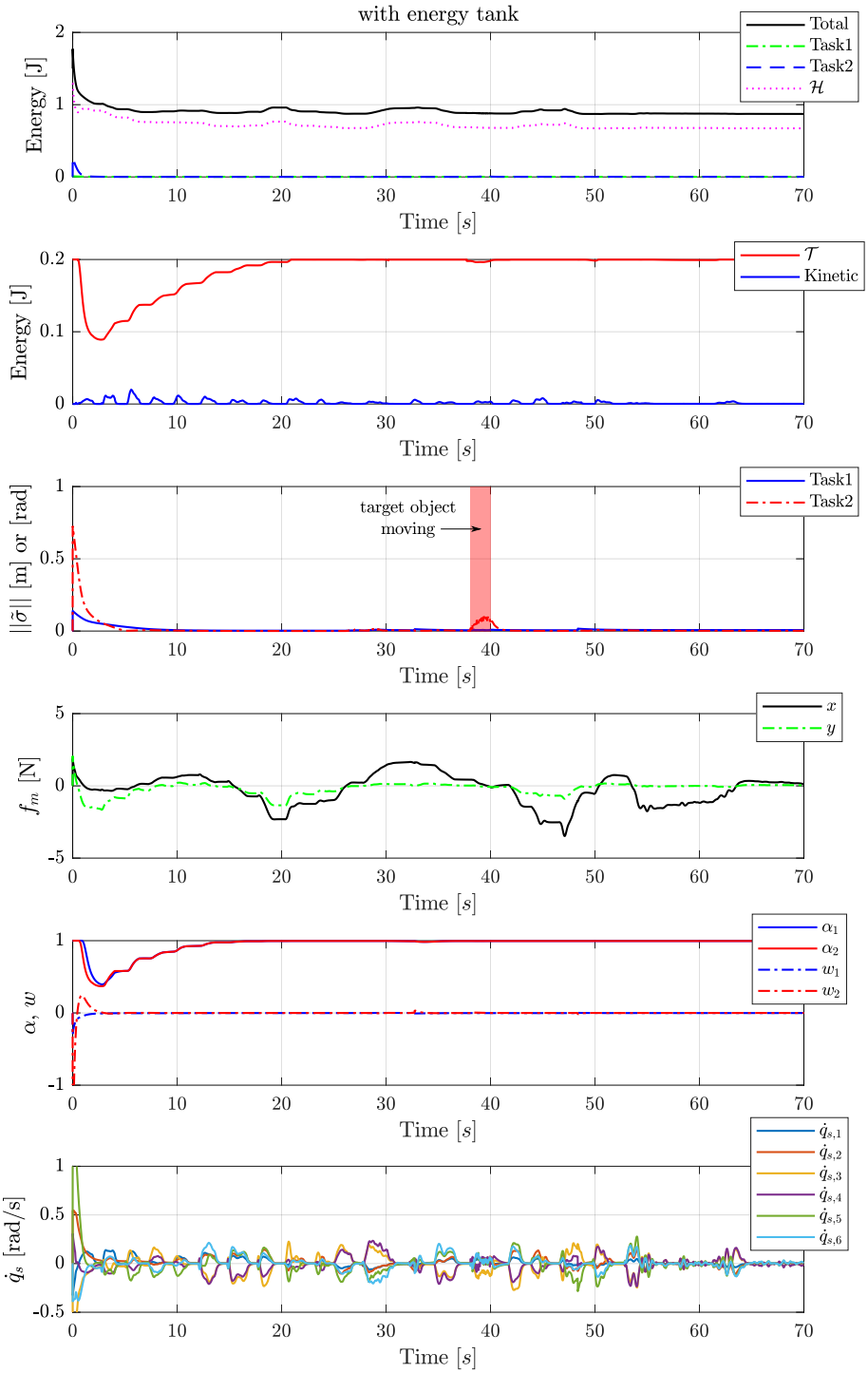


Figure 5.5: Time evolution of system energies, error norms, haptic guidance forces, relevant control quantities and joint velocities during the shared-control teleoperation experiment. The red shaded area indicates the time slot in which the object is moved.

Part II

Shared Control Methods for Surgical Scenarios

Chapter 6

Virtual Fixtures-based Shared Control Methods

During robot-aided surgical interventions, the surgeon can benefit from the application of Virtual Fixtures (VF). Though very effective, this technique is very often not practicable in unstructured surgical environments. In order to comply with the environmental deformation, both the VFs geometry and the constraint enforcement parameters need to be online defined/adapted. This paper proposes a strategy for an effective use of VF assistance in minimally invasive robotic surgical tasks. An online VF generation technique based on interaction force measurements is presented. Pose and geometry adaptations of the VF are considered. Passivity of the overall system is guaranteed by using energy tanks passivity-based control. The proposed method is validated through experiments on the da Vinci Research Kit [204].

6.1 Introduction

Robot-aided surgery permits filtering the surgeon's hand tremors and scaling down motions in order to make highly precise and dexterous movements inside the patient's body. This allows reducing trauma and postoperative pains which, in turn, results in significantly faster recovery time for patients.

In recent years, the application of shared-control techniques to robotic surgical interventions started to be investigated. Among these techniques, the use of virtual fixtures (VFs) has been identified as one of the most promising approaches [125]. VFs are software generated constraints which restrict the motion of a robotic manipulator into predefined regions or constrain it to move

along a preplanned path [14] (see Sect. 2.2). This allows improving the operator's performance by reducing the mental workload, increasing the precision and/or decreasing the task completion time.

A wide variety of surgical tasks can benefit from the use of VFs, such as percutaneous needle insertion [205], femur cutting for prosthetic implant [206], suturing [207, 208]. However, the main drawback of this methodology is the strong task dependency: switching among even structured tasks would require interruptions and reprogramming of the system. Moreover, unstructured environments pose additional challenges to the VFs application. For instance, in minimally invasive robotic surgery the environment is deformable due to the flexibility of the organs and this makes the precomputed VF techniques hardly practicable.

6.1.1 Problem statement

Automatic VF generation/adaptation can be performed by exploiting vision-based techniques [100, 101]. Vision systems allow tracking organs as they move or deform. However, this can be a demanding task, particularly in cases where surgical targets have few structural features for tracking. Alternatively, force controlled exploration of patient-specific anatomy can be a time consuming procedure and thus not practicable in some surgical interventions [177]. Whatever method is used to generate them, to be effective in the surgical scenario, VFs must be adapted online as the environment moves or deforms, *i.e.* they must be repositioned and/or opportunely refined in order to reflect the different environmental configuration. Very few papers make a significant consideration of *adaptive* VF, where the constraint geometry (semi-)autonomously moves as a result of environmental changes [209].

On the other hand, guaranteeing a stable behavior of the robot endowed with adaptive VF control is of a paramount importance in robotic surgery. Interactive VF application results in variable impedance controllers that could threaten the stability of the robotic system [200, 201]. The analysis of the system behavior is complicated by the interaction with the user which further contributes to the overall system impedance parameters variation. Thus, passivity-based control techniques need to be exploited in order to ensure the safe use of the system for every change in the impedance parameters.

6.1.2 Related Works

Related works in this field can be separated into two main topics: *1) adaptive virtual fixtures*, *2) passivity-based control*.

Adaptive virtual fixtures

Historically, robotic teleoperation systems have made use of VF as a perceptual overlay to enhance the human experience in performing remote manipulation tasks. Rosenberg pioneered VF in his work [13]. Since then, renewed interest has been shown in their use for robotic surgical procedures. In [210] the first attempt to transfer VF to the operating room is shown. The authors used VF generated from preoperative computed tomography images and constrained the motions of a surgical robot to a predefined path during the dissection of the internal mammary artery. Interactive generation of VF in surgical applications has been addressed in [117] where constrained optimization is used to enforce the VF constraint with objective function derived by user inputs.

More recently, it has been shown that VF can be opportunely generated using scans of the area of interest [97, 117]. A dynamic VF technique to enhance the surgical operation accuracy of admittance-type medical robots in deforming environments is presented in [110]: the target deformation is tracked actively and the proxy motion is constrained on a deforming sphere to simulate the beating heart surgery environment. One of the major obstacles in implementing vision generated VF in surgical applications is the organs displacement and deformation: whether the constraint geometry is defined preoperatively or intraoperatively, it must be mapped correctly to the organs as they move or deform.

Alternatively, VF may be opportunely adapted according to the currently being executed task or on the estimation of surgeon's skills. Learning from demonstration has been used for task dependent VFs application in [107], whereas adaptive VFs, based on the surgeon's level of expertise, are shown in [122]. An algorithm to select an appropriate admittance ratio based on the nature of the task was developed in [129]: automatic admittance ratio tuning is recommended for an efficient use of VFs. A method for online task tracking and on the use of adaptive VFs that can cope with their inherent inflexibility is presented in [108].

Passivity-based control

Few authors have addressed the stability issues caused by VF adaptation as this can threaten the system stability. In [211] the authors applied and experimentally validated a non-energy-storing class of dynamic guidance VFs that do not suffer from internal energy accumulation. Using a friction model to enforce constraints ensures that energy is never accumulated into the system [111]. However, frictional constraints suffer from problems related to forbidden regions replication. Redirection of VF forces in surgical robotics based on a passivity preserving condition has been presented in [212]. Stability of the closed-loop system can be investigated using the small-gain theorem: a sufficient condition, that guarantees stability in the presence of time-varying communication delay, is derived in [121]. Variable impedance/admittance control of robotic manipulators makes largely use of passivity-based control techniques to stabilize the system. An interesting approach relies on the concept of energy tanks and has been effectively used in [200, 201, 213]. The energy tanks approach is particularly suitable for the teleoperation, where large damping factors may degrade the transparency of the bilateral telerobotic system.

6.1.3 Contribution

This chapter presents a methodology for online interactive generation and passive adaptation of VF for robotic assisted surgical interventions. More specifically, this work proposes:

- a VF pose and geometry adaptation strategy that uses a human-in-the-loop VF generation technique;
- an energy tanks passivity-based control method that ensures a stable system behavior.

The developed methodologies are validated through experiments performed on the da Vinci Research Kit (dVRK) [214], in which surgical dissection tasks have been considered.

The remainder of the chapter is organized as follows: in Sect. 6.2 preliminary methods used for VFs implementation are presented; Section 6.3 contains VF generation and adaptation techniques; Section 6.4 shows the energy tanks passivity-based control; Section 6.5 contains the performed experiments and results alongside a discussion; Section 6.6 concludes the chapter.

6.2 Preliminaries

In this section, the methods used for the implementation of VFs to generic teleoperated robotic systems are illustrated.

6.2.1 Robot Impedance Control

A very common practice in robotic teleoperation is to have an admittance/impedance controlled master robot. Considering a n-DoF manipulator and defining a task space vector $\boldsymbol{x} \in \mathbb{R}^r$, with $r \leq n$, the following impedance dynamics can be achieved through control

$$\boldsymbol{M}\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{D}\dot{\tilde{\boldsymbol{x}}} = \boldsymbol{f}_{\text{h,A}} + \boldsymbol{f}_{\text{vf}}(\cdot), \tag{6.1}$$

where $\tilde{\boldsymbol{x}} = \boldsymbol{x}_{\text{d}} - \boldsymbol{x}$, with $\boldsymbol{x}_{\text{d}}$ being the desired value for the robot task space variable, $\boldsymbol{M} \in \mathbb{R}^{r\times r}$ and $\boldsymbol{D} \in \mathbb{R}^{r\times r}$ are inertia and damping matrices respectively, usually designed to be fixed, diagonal and positive definite, $\boldsymbol{f}_{\text{h}} \in \mathbb{R}^r$ is the vector of the external forces applied by an interacting user and $\boldsymbol{f}_{\text{vf}}(\cdot)$ is the additional force due to the possible presence of VFs. The above dynamics can be obtained by setting the torque control input $\boldsymbol{\tau} \in \mathbb{R}^n$ of the master robot as (see [11] for more details)

$$\boldsymbol{\tau} = \boldsymbol{B}(\boldsymbol{q})\,\boldsymbol{v} + \boldsymbol{N}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{J}^{\text{T}}(\boldsymbol{q})\,\boldsymbol{f}_{\text{h}} \tag{6.2}$$

$$\boldsymbol{v} = \boldsymbol{J}_{\text{A}}^{-1}(\boldsymbol{q})\,\boldsymbol{M}^{-1}(\boldsymbol{M}\ddot{\boldsymbol{x}}_d + \boldsymbol{D}\dot{\tilde{\boldsymbol{x}}} - \boldsymbol{M}\dot{\boldsymbol{J}}_{\text{A}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\,\dot{\boldsymbol{q}} - \boldsymbol{f}_{\text{h,A}} - \boldsymbol{f}_{\text{vf}}(\cdot)), \tag{6.3}$$

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

$$\boldsymbol{N}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{6.4}$$

accounts for Coriolis and centrifugal contributions $\big(\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}\big)$, gravity $\big(\boldsymbol{g}(\boldsymbol{q})\big)$, friction and other disturbance torques $\big(\boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}})\big)$. Notice that the term $\boldsymbol{f}_{\text{h,A}}$ differs from $\boldsymbol{f}_{\text{h}}$ by a mapping, depending on the orientation representation.

6.2.2 External Force Estimation

Equation (6.2) requires the measurement of the external forces $\boldsymbol{f}_{\text{h}}$. When they are not directly measurable, force estimation could be performed by resorting to a nonlinear dynamic observer [215, 216]. This method allows the estimation

of unknown external forces without the need of measuring the usually noisy acceleration signal.

In greater details, considering the measured torque vector $\boldsymbol{\tau}_r \in \mathbb{R}^n$, a residual vector can be defined as follows

$$\boldsymbol{r}(t) = \boldsymbol{K}_I \left(\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}} - \int_0^t \left(\boldsymbol{r}(\sigma) + \boldsymbol{\tau}_r + \tilde{\boldsymbol{N}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\, d\sigma \right) \right), \tag{6.5}$$

where $\boldsymbol{K}_I$ is a diagonal and positive definite gain matrix and

$$\tilde{\boldsymbol{N}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{C}^T(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}). \tag{6.6}$$

Hence, the estimated external forces $\hat{\boldsymbol{f}}_{\mathrm{h}}$ can computed using the following equality

$$\hat{\boldsymbol{f}}_{\mathrm{h}} = \boldsymbol{J}^{-T}(\boldsymbol{q})\, \boldsymbol{r}, \tag{6.7}$$

where $\boldsymbol{J}^{-T}(\boldsymbol{q})$ denotes the generalized inverse of the robot Jacobian transpose. Hereafter, we will consider $\boldsymbol{f}_{\mathrm{h}} = \hat{\boldsymbol{f}}_{\mathrm{h}}$.

6.2.3 Virtual Fixtures

VFs can be classified into two main classes: Forbidden-Region Virtual Fixtures (FRVF) and Guidance Virtual Fixtures (GVF) [14]. Generally speaking, FRVF are suitable for simulating barriers, constraining surfaces or delicate regions that the user should be forbidden to enter. In contrast, a GVF has an attractive behavior that pulls the robot end-effector towards a desired path (see Fig. 6.1).

In this work, the focus on the latter VF type (hereafter, we will refer to GVF simply as VF). Two quantities are essential to describe a VF: its *geometry* and the *constraint enforcement* method.

VF geometry model

A simple, yet general, way of geometrically formulating a smooth continuous VF is through parametric curves. Without loss of generality, we adopt cubic splines. In its 1-dimensional form, a cubic spline is defined by

$$\Gamma_i(s) = C_0 + C_1(s - x_i) + C_2(s - x_i)^2 + C_3(s - x_i)^3 \tag{6.8}$$

where $\Gamma_i(s)$ denotes the curve in its i-th interval $[x_i, x_{i+1}]$, $s \in [0, 1]$ is the curve parameter, C_0, C_1, C_2, C_3 are constants determined by imposing four conditions (usually being boundary constraints $\Gamma_i(0), \Gamma_i(1), \Gamma_i'(0), \Gamma_i'(1)$, where $\Gamma_i'(\cdot)$ denotes the curve derivative w.r.t. the parameter s). As explained

later in this work (Sect. 6.3), we build the spline geometry by fitting a set of recorded interaction points. For this purpose, we use parabolically terminated splines.

VF constraint enforcement

A GVF exhibits attractive behavior towards the desired path. The simplest constraint enforcement method consists in applying a spring-damper like force. In the linear case this can be defined as follows

$$\boldsymbol{f}_{\text{vf}}\left(\tilde{\boldsymbol{x}}, \dot{\tilde{\boldsymbol{x}}}\right) = -\boldsymbol{K}_{\text{vf}}\tilde{\boldsymbol{x}} - \boldsymbol{D}_{\text{vf}}\dot{\tilde{\boldsymbol{x}}}, \tag{6.9}$$

where $\boldsymbol{K}_{\text{vf}} \in \mathbb{R}^{r\times r}$ and $\boldsymbol{D}_{\text{vf}} \in \mathbb{R}^{r\times r}$ are properly designed diagonal and positive definite matrices and $\boldsymbol{x}_{\text{d}}$ is the set-point belonging to the constraint geometry having minimum distance from $\boldsymbol{x}$ (see Fig. 6.1). An impedance controlled manipulator (7.22), endowed with VF control forces defined in (6.9), exhibits a closed-loop behavior that can be described by

$$\boldsymbol{M}\ddot{\tilde{\boldsymbol{x}}} + \hat{\boldsymbol{D}}\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_{\text{vf}}\tilde{\boldsymbol{x}} = \boldsymbol{f}_{\text{h,A}}, \tag{6.10}$$

where $\hat{\boldsymbol{D}} = \boldsymbol{D} + \boldsymbol{D}_{\text{vf}}$ contains the damping assigned both by the impedance control and the VF constraint enforcement method. The desired dynamics (6.10) can be easily obtained by adding the elastic and damping contributions shown in the right-hand side of (6.9) to the control input defined in (6.3).

6.2.4 Minimun Distance

For cubic splines, described by (6.8), there does not exist an analytical solution to the problem of minimum Euclidean distance computation. However, this problem can be tackled by resorting to iterative methods, such as Newton-Raphson (NR). This represents a general method of finding the extrema (minima or maxima) of a given function in an iterative manner. Our goal is to find the spline parameter $\bar{s}$ corresponding to the minimum distance point $\boldsymbol{x}_{\text{d}} = \Gamma(\bar{s})$. For this purpose, starting from a generic initial condition $s_0 \in [0, 1]$, we use the customary NR update law

$$s_{k+1} = s_k + \frac{\delta(\boldsymbol{x}, s_k)}{\delta'(\boldsymbol{x}, s_k)}, \tag{6.11}$$

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

$$\delta(\boldsymbol{x}, s) = \sqrt{(\boldsymbol{x} - \Gamma(s))^{\mathrm{T}}(\boldsymbol{x} - \Gamma(s))} \tag{6.12}$$

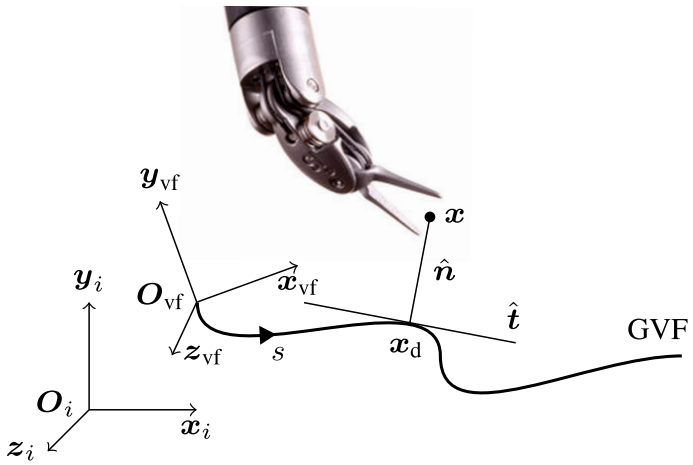


Figure 6.1: Example of Guidance Virtual Fixture (GVF) spline geometry and minimum distance from the robot tool central point $\boldsymbol{x}$. $\hat{\boldsymbol{t}}$ and $\hat{\boldsymbol{n}}$ denote the tangent and the normal directions, respectively, with origin in $\boldsymbol{x}_{\text{d}}$, *i.e.*, the $\boldsymbol{x}$ nearest point on the curve. $\mathcal{F}_i : \{\boldsymbol{O}_i; \boldsymbol{x}_{\mathbf{i}}, \boldsymbol{y}_{\mathbf{i}}, \boldsymbol{z}_{\mathbf{i}}\}$ = inertial frame, $\mathcal{F}_{\text{vf}} : \{\boldsymbol{O}_{\text{vf}}; \boldsymbol{x}_{\text{vf}}, \boldsymbol{y}_{\text{vf}}, \boldsymbol{z}_{\text{vf}}\}$ = virtual fixture frame.

and $\delta'(\boldsymbol{x}, s_k)$ denotes the derivative at s_k of (6.12) with respect to the curve parameter s.[1]

6.3 Virtual Fixture Generation and Adaptation

In this section, the VF *generation* technique and the *adaptation* strategies are described.

6.3.1 VF Generation

The aim is to let the user interactively program VFs for interaction tasks (*e.g.* surgical dissections). Among many other choices, the policy of recording a set of interaction points that are then used to build the VF geometry is adopted. In this way, the surgeon is given the ability to program the VF geometric path by simply interacting with the environment. Interaction detection is possible by

[1] Although computationally very efficient, NR method can converge to local maxima/minima. To tackle this problem, the previously determined $\bar{s}$ is used as initial guess for the next minimum distance query. This method was empirically found to be effective for the scope of this work.

measuring forces at the slave side. Recorded points are then fitted through a penalized regression spline fitting algorithm in which coefficients of (6.8) are obtained by minimizing the sum of least squares plus a penalty function which suppresses nonlinearity and controls the curve smoothing. Mathematically, the problem is described by

$$\Gamma(s) = \arg\min_{\Gamma(s)} \left(\sum_i (y_i - \Gamma(s_i))^2 + \lambda \int (\Gamma''(s))^2 ds \right) \tag{6.13}$$

where λ is the regularization parameter that penalizes nonlinearities in the path, y_i is the i-th recorded interaction point and $\Gamma''(s)$ is the curve second derivative of $\Gamma(s)$ with respect to its parameter s.

Alternatively, the VFs geometry can be created via specific control points obtained directly from images of the surgical scene and can be updated accordingly by the vision algorithm as demonstrated in [217].

6.3.2 VF Adaptation

To adapt a VF preserving the proposed human-in-the-loop approach, a non linear and time varying stiffness profile is adopted for $\boldsymbol{K}_{\text{vf}}$ in (6.10). This is used to both limit the spatial and the temporal influence of a VF. More specifically, we design each non-zero entry of the stiffness matrix to be

$$k_{\text{vf,ii}}(\tilde{x}, t) = \beta(\tilde{x}, t) K_{\max} \quad \forall\ i = 1, \ldots, r \tag{6.14}$$

where $k_{\text{vf,ii}}$ is the (i,i) entry of the $\boldsymbol{K}_{\text{vf}}$ matrix, $\beta(\tilde{x}, t)$ is an impedance shaping function, $\tilde{x}$ is the i-th component of the vector $\tilde{\boldsymbol{x}}$, t denotes time and $K_{\max}$ is the maximum stiffness value adopted. The definition of $\beta(\tilde{x}, t)$ allows realizing different adaptation strategies as detailed in the following two sections.

Pose Adaptation

The pose adaptation strategy consists in positioning a predefined VF geometry into a desired location. With reference to Fig. 6.1 the problem is to define a desired reference frame $\mathcal{F}_{\text{vf,d}} : \{\boldsymbol{O}_{\text{vf,d}}; \boldsymbol{x}_{\text{vf,d}}, \boldsymbol{y}_{\text{vf,d}}, \boldsymbol{z}_{\text{vf,d}}\}$ to which the current VF reference frame $\mathcal{F}_{\text{vf}} : \{\boldsymbol{O}_{\text{vf}}; \boldsymbol{x}_{\text{vf}}, \boldsymbol{y}_{\text{vf}}, \boldsymbol{z}_{\text{vf}}\}$ must converge to. As explained, to make this procedure interactive for the user, the adopted policy consists in recording a set of slave robot interaction points with the environment that are then used to fit the predefined geometry. To allow the user to freely record new interaction points, is convenient to limit the spatial influence of the current VF

adopting the $\beta(\tilde{x})$ function qualitatively depicted in Fig. 6.2: this function allows the operator to easily exit the VF constraining zone and freely record new interaction points as sought. For each task space variable $\beta(\tilde{x})$ is mathematically described by

$$\beta\left(\tilde{x}\right) = \begin{cases} 0 & \text{if } |\tilde{x}| \geq l \\ \frac{1}{2}\left(1 + \cos\left(\frac{\pi\left(|\tilde{x}| - d\right)}{l - d}\right)\right) & \text{otherwise} \\ 1 & \text{if } |\tilde{x}| \leq d \end{cases} \tag{6.15}$$

where l is the distance at which the VF attractive action vanishes completely and d is the threshold distance value inside which the stiffness perceived is $K_{\max}$. Once a set of recorded interaction points is available, the classical least-squares minimization method is used to fit the predefined geometry onto it. This gives the desired VF pose that minimizes the sum of squared residuals between the VF from the point set, *i.e.* $\mathcal{F}_{\text{vf,d}}$. This pose is then tracked online by suitably defining the pose error between $\mathcal{F}_{\text{vf,d}}$ and $\mathcal{F}_{\text{vf}}$ frames. Indicating with $\boldsymbol{R} = [\boldsymbol{n}, \boldsymbol{s}, \boldsymbol{a}]$ and $\boldsymbol{R}_d = [\boldsymbol{n}_d, \boldsymbol{s}_d, \boldsymbol{a}_d]$ the rotation matrices associated with $\mathcal{F}_{\text{vf}}$ and $\mathcal{F}_{\text{vf,d}}$, respectively, the error can be written as follows (see [11])

$$\boldsymbol{e} = \begin{bmatrix} \boldsymbol{e}_P \\ \boldsymbol{e}_O \end{bmatrix} = \begin{bmatrix} \boldsymbol{O}_{\text{vf,d}} - \boldsymbol{O}_{\text{vf}} \\ \frac{1}{2}(\boldsymbol{n} \times \boldsymbol{n}_d + \boldsymbol{s} \times \boldsymbol{s}_d + \boldsymbol{a} \times \boldsymbol{a}_d) \end{bmatrix}, \tag{6.16}$$

where $\boldsymbol{O}_{\text{vf,d}}$ and $\boldsymbol{O}_{\text{vf}} \in \mathbb{R}^3$ denote the desired and current frame origins, respectively. A simple proportional control law defined on the error in (6.16) allows it to asymptotically converge to zero guaranteeing a smooth regulation behavior. The resulting error dynamics can be written as follows

$$\dot{\boldsymbol{e}} + \boldsymbol{\Lambda}_p \boldsymbol{e} = 0 \qquad \boldsymbol{\Lambda}_p > 0, \tag{6.17}$$

where $\boldsymbol{\Lambda}_p$ is a positive definite diagonal matrix containing control gains that are numerically different to account the non-homogeneous dimensions of the error blocks.

Geometry Adaptation

The geometry adaptation strategy consists in transforming the current VF geometry into a desired one.

Suppose that the user is performing a task aided by the currently active VF and necessarily needs to deviate, *e.g.* to comply with the environment deformation. The same interaction points fitting strategy used for the VF generation

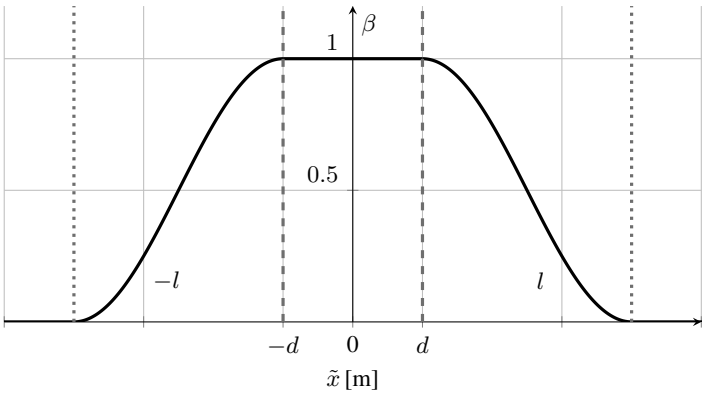


Figure 6.2: VF impedance shaping function $\beta(\tilde{x})$ used to limit the VF spatial influence. $\tilde{x}$ denotes the difference between the desired and current value or the master task space variable.

(see Sect. 6.3.1) can be used to define a new desired geometry. In this case, the user may want to completely deactivate the current VF to freely record new interaction points. To let the impedance parameters of the former VF to gradually vanish, we use the following temporal variation of the stiffness

$$\beta(t) = \frac{1}{2}\left(1 + \cos\left(\frac{\pi\,(t - t_s)}{t_i - t_s}\right)\right) \quad t_s < t < t_i \tag{6.18}$$

where t_s, t_i are the starting and final instant of the stiffness variation time interval. Once a new desired geometry has been defined, the problem is to redirect the robot tool central point $\boldsymbol{x}$ towards the new VF.

To achieve a smooth behavior of the system, the impedance parameters must gradually materialize. This is realized by the following temporal stiffness variation

$$\beta(t) = \frac{1}{2}\left(1 - \cos\left(\frac{\pi\,(t - t_i)}{t_f - t_i}\right)\right) \quad t_i < t < t_f \tag{6.19}$$

where t_i, t_f are initial and final instants of the positive impedance variation, respectively. To activate/deactivate the VF according to the stiffness variation laws (6.18) and (6.19), different strategies can be adopted. In this work, this functionality is associated to the pressing of a dVRK console foot pedal.

It is worth noting that the proposed impedance variation strategy can be also employed to smoothly apply a VF or switch between two of them: switching the attraction point $\boldsymbol{x}_d$ from one VF to another, when the stiffness reaches zero, guarantees a smooth transition of the system. From a passivity point of view this translates in not generating additional energy into the system.

6.4 Energy Tanks Passivity-Based Control

In this section, the system passivity when subject to the proposed impedance variations is analyzed (see Sect. 6.3.2). The definition of passivity given in Def. 1 is used.

It can be easily noted that the model in (6.10) is not guaranteed to be passive w.r.t. the input-output pair ($\boldsymbol{f}_{\text{h}}$,$\dot{\tilde{\boldsymbol{x}}}$) if we consider as storage function the system energy

$$H(\tilde{\boldsymbol{x}}, \dot{\tilde{\boldsymbol{x}}}) = \frac{1}{2}\dot{\tilde{\boldsymbol{x}}}^{\text{T}}\boldsymbol{M}\dot{\tilde{\boldsymbol{x}}} + \frac{1}{2}\tilde{\boldsymbol{x}}^{\text{T}}\boldsymbol{K}_{\text{vf}}\tilde{\boldsymbol{x}}, \tag{6.20}$$

where $\boldsymbol{M}$, and $\boldsymbol{K}_{\text{vf}}$ have been defined in Sect. 6.2. Indeed, the time derivative of (6.20) (assuming that $\boldsymbol{M}$ remains constant over time) can be written as follows (omitting arguments of $H(\tilde{\boldsymbol{x}}, \dot{\tilde{\boldsymbol{x}}})$)

$$\dot{H} = \dot{\tilde{\boldsymbol{x}}}^{\text{T}}\boldsymbol{M}\ddot{\tilde{\boldsymbol{x}}} + \tilde{\boldsymbol{x}}^{\text{T}}\boldsymbol{K}_{\text{vf}}\dot{\tilde{\boldsymbol{x}}} + \frac{1}{2}\tilde{\boldsymbol{x}}^{\text{T}}\dot{\boldsymbol{K}}_{\text{vf}}\tilde{\boldsymbol{x}}, \tag{6.21}$$

which, evaluated along the system trajectories, becomes

$$\dot{H} = \dot{\tilde{\boldsymbol{x}}}^{\text{T}}\boldsymbol{f}_{\text{h}} - \dot{\tilde{\boldsymbol{x}}}^{\text{T}}\hat{\boldsymbol{D}}\dot{\tilde{\boldsymbol{x}}} + \frac{1}{2}\tilde{\boldsymbol{x}}^{\text{T}}\dot{\boldsymbol{K}}_{\text{vf}}\tilde{\boldsymbol{x}}. \tag{6.22}$$

Since $\dot{\boldsymbol{K}}_{\text{vf}}$ can have both positive and negative eigenvalues, a sufficient, yet conservative, condition to satisfy (2.3) is to have a negative semidefinite $\dot{\boldsymbol{K}}_{\text{vf}}$ in (6.22).

A possible solution to this problem is to design a passivity preserving controller that tracks the desired stiffness profile while limiting its change when condition (2.3) is violated. To this end, the concept of energy tanks, introduced in [218], is exploited. It aims at recovering the dissipated energy of the system to implement a less conservative impedance variation without violating the overall passivity of the system. To this end, the master side manipulator is endowed with an energy storing element having the following storage function

$$T(z) = \frac{1}{2}z^2, \tag{6.23}$$

whose dynamics is described by the following equation

$$\dot{z} = \frac{\varphi}{z}\dot{\tilde{\boldsymbol{x}}}^{\text{T}}\hat{\boldsymbol{D}}\dot{\tilde{\boldsymbol{x}}} - \frac{\gamma}{z}\frac{1}{2}\tilde{\boldsymbol{x}}^{\text{T}}\dot{\boldsymbol{K}}_{\text{vf}}\tilde{\boldsymbol{x}}, \tag{6.24}$$

with $z \in \mathbb{R}$ being the state of the tank and $\varphi, \gamma \in \{0, 1\}$ are parameters used to enforce the upper bound limitation $\bar{T}$ for the energy stored in the tank [156].

Note that (6.24) is singular for $z = 0$, thus a lower threshold ε for the tank energy must also be set.

Thus, the extended dynamics can be rewritten as follows

$$\begin{cases} \boldsymbol{M}\ddot{\tilde{\boldsymbol{x}}} + \hat{\boldsymbol{D}}\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_{\text{vf}}\tilde{\boldsymbol{x}} = \boldsymbol{f}_{\text{h}} \\ \dot{\boldsymbol{K}}_{\text{vf}} = \alpha\left(\boldsymbol{\Lambda}_k\left(\boldsymbol{K}_{\text{vf,d}} - \boldsymbol{K}_{\text{vf}}\right) + \dot{\boldsymbol{K}}_{\text{vf,d}}\right) \\ \dot{z} = \dfrac{\varphi}{z}\dot{\tilde{\boldsymbol{x}}}^{\text{T}}\hat{\boldsymbol{D}}\dot{\tilde{\boldsymbol{x}}} - \dfrac{\gamma}{z}\dfrac{1}{2}\tilde{\boldsymbol{x}}^{\text{T}}\dot{\boldsymbol{K}}_{\text{vf}}\tilde{\boldsymbol{x}} \end{cases}, \tag{6.25}$$

where $\boldsymbol{\Lambda}_k \in \mathbb{R}^{r\times r}$ is a diagonal and positive definite matrix containing the stiffness tracking control parameter and $\alpha \in \{0, 1\}$ is a variable used to activate/deactivate the stiffness variation in case of passivity violation. The master system, endowed with the energy tank, has the following energy function

$$\mathcal{H} = H + T = \frac{1}{2}\dot{\tilde{\boldsymbol{x}}}^{\text{T}}\boldsymbol{M}\dot{\tilde{\boldsymbol{x}}} + \frac{1}{2}\tilde{\boldsymbol{x}}^{\text{T}}\boldsymbol{K}_{\text{vf}}\tilde{\boldsymbol{x}} + \frac{1}{2}z^2, \tag{6.26}$$

whose time derivative is given by

$$\dot{\mathcal{H}} = \dot{H} + \dot{T} = \dot{\tilde{\boldsymbol{x}}}^{\text{T}}\boldsymbol{M}\ddot{\tilde{\boldsymbol{x}}} + \tilde{\boldsymbol{x}}^{\text{T}}\boldsymbol{K}_{\text{vf}}\dot{\tilde{\boldsymbol{x}}} + \frac{1}{2}\tilde{\boldsymbol{x}}^{\text{T}}\dot{\boldsymbol{K}}_{\text{vf}}\tilde{\boldsymbol{x}} + z\dot{z} \tag{6.27}$$

which, evaluated along the system trajectories, becomes

$$\dot{\mathcal{H}} = \dot{\tilde{\boldsymbol{x}}}^{\text{T}}\boldsymbol{f}_{\text{h}} - (1-\varphi)\,\dot{\tilde{\boldsymbol{x}}}^{\text{T}}\hat{\boldsymbol{D}}\dot{\tilde{\boldsymbol{x}}} + (1-\gamma)\,\frac{1}{2}\tilde{\boldsymbol{x}}^{\text{T}}\dot{\boldsymbol{K}}_{\text{vf}}\tilde{\boldsymbol{x}}. \tag{6.28}$$

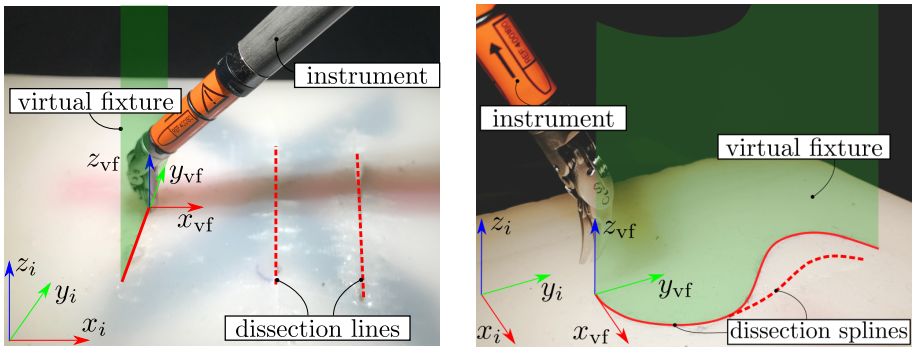
By defining the following control laws for α, φ and γ

$$\begin{aligned} \alpha &= \begin{cases} 0 & \text{if } T \le \varepsilon \;\&\; \dot{\boldsymbol{K}}_{\text{vf}} > 0 \\ 1 & \text{otherwise} \end{cases} \qquad \gamma = \begin{cases} \varphi & \text{if } \dot{\boldsymbol{K}}_{\text{vf}} < 0 \\ 1 & \text{otherwise} \end{cases} \\ \varphi &= \begin{cases} 1 & \text{if } T \le \bar{T} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{6.29}$$

the system in (6.25) is passive with respect to the input-output pair ($\boldsymbol{f}_{\text{h}}$,$\dot{\tilde{\boldsymbol{x}}}$) with storage function (6.26). Indeed, it can be easily verified that (6.28) always satisfies the condition (2.3).

6.5 Experiments and Results

In this section, the experimental setup is described and the VF pose and geometry adaptation experiments are presented.



(a) (b)

Figure 6.3: Virtual fixtures adaptation: experimental setup and tasks: (a) Linear dissection tasks with VF pose adaptation; (b) Curvilinear dissection task with VF geometry adaptation.

6.5.1 Experimental Setup

The experiments have been performed on the dVRK platform. The robot has been used in teleoperation mode, with one Patient Side Manipulator (PSM) commanded by one Master Tool Manipulator (MTM). The MTM has been controlled using impedance control inputs described in Sect. 6.2.1 with $m_{ii} = 1.5$, $d_{ii} = 0$ being the (i,i) entries of the matrices $\boldsymbol{M}$ and $\boldsymbol{D}$, respectively. The dVRK dynamic parameters used in (6.2) had been previously identified in [21]. The application of torque inputs has been possible thanks to the open-source hardware and software architecture developed by [214]. Given the discrete-time implementation and in order to have a critically damped system behavior, the $\boldsymbol{D}_{\text{vf}}$ has been adapted according to the stiffness variation such that $d_{\text{vf},ii} = 2\sqrt{m_{ii} k_{\text{vf},ii}}$ where $d_{\text{vf},ii}$ and $k_{\text{vf},ii}$ are the (i,i) entries of the matrices $\boldsymbol{D}_{\text{vf}}$ and $\boldsymbol{K}_{\text{vf}}$, respectively. Forces at the slave side have been measured making use of the dVRK trocar force sensor developed in [219]. The object used for experiments is a silicon rubber phantom commonly used by surgeons for training. It has been placed on a plastic 3D printed support. The fitting algorithm described by (6.13) has been implemented using the ALGLIB library [220]. The values of the β function limits in (6.15) have been fixed to $d = 0.005\,\text{m}$ and $l = 0.02\,\text{m}$, while $t_i = t_f = 2.5\,\text{s}$ in (6.18) and (6.19). The maximum stiffness has been fixed to $K_{\text{vf,max}} = 600\,\text{N/m}$.

The energy tank upper threshold has been chosen as $\bar{T} = 0.01$ J while the lower threshold, has been set to $\varepsilon = 0.002$ J. The gap between these two

quantities has been designed according to the maximum potential energy that is possible to store using the proposed $\beta(\tilde{x})$ function (see Fig. 6.2), *i.e.* $E_{\text{p,max}} = 1/2 K_{\text{vf,max}} d^2$. The VF control loop runs at 5 ms while the teleoperation loop at 1 ms.

6.5.2 Pose Adaptation Experiment

The pose adaptation strategy (introduced in Sect. 6.3.2) is evaluated in multiple dissection tasks executed in spatially separated regions. Without loss of generality, the geometry of the VF is fixed to be a spline representing a straight line in this experiment. This particular choice is made to present clearer and more intuitive results of the pose adaptation. In this case, the fitting strategy for the VF geometry reduces to a linear regression problem where the desired $\boldsymbol{y}_{\text{vf,d}}$ axis of the $\mathcal{F}_{\text{vf,d}}$ reference frame is fitted using the last $n_p = 50$ recorded interaction poses of the slave robot with the environment. In addition, the human operator is left free to move in the direction orthogonal to the phantom (see Fig. 6.3a). Every dissection task requires multiple interaction phases with the environment in which points are recorded. The VF pose is continuously updated by fitting these recorded points. With reference to Fig. 6.4 the experiment starts with the VF approximatively placed on the first dissection line. As soon as the robot starts interacting with the environment, the desired stiffness reaches the K_{max} value and the user is aided in accomplishing the first task by complying with the VF geometry. Around 20s the user switch to another task by exiting the current VF influence area ($|\tilde{x}| > l$). This can be seen in Fig. 6.4(e) in which we register a peak in the estimated force at the master side (≈ 5 N) when the stiffness starts to decrease. Notice that this exiting is made possible by the $\beta(\tilde{x})$ function chosen in (6.15). During the subsequent time period, the user moves in free motion, *i.e.* $f_h \approx 0$ as it can be seen in Fig. 6.4(e) (time interval [20,25]s). A new interaction phase takes place in a spatially separated region at 25s. The new desired VF pose is calculated and the VF is updated following the method described in Sect. 6.3.2.

Figure 6.4(d) shows the interaction force norm recorded at the slave side together with the chosen threshold $\bar{\delta} = 0.5$ N. This has been used to discriminate between interaction and free motion during the acquisition of points for the VF generation. Also note that, upon the starting of a new task, $x_{\text{vf,d}}$ changes significantly. However, the user does not experience guidance forces until the VF reaches the proximity of the master robot position. Indeed, only when $|\tilde{x}| < l$ the stiffness is increased. This effect is again produced by the chosen $\beta(\tilde{x})$ function (6.15). As mentioned in Sect. 6.4, the change in stiff-

ness threatens the system passivity. The energy tank passivity-based control ensures a passive behavior by implementing the change in stiffness only when sufficient tank energy is at disposal. This is evident looking at Figs. 6.4(b) and 6.4(c). In particular, it is possible to note that the stiffness is kept constant (*i.e.* $\dot{K}_{\text{vf}} = 0$) when the tank is at its lower threshold until it gets replenished, thus not introducing discontinuities. This is made possible through the use of control laws given in (6.25) and (6.29). Figure 6.4(b) contains a focus around $50\,\text{s}$ that emphasizes this behavior.

Finally, looking at the estimated forces in Fig. 6.4(e), it can be noticed that relatively high forces ($\approx 5\,\text{N}$) are only applied at the task switching.

6.5.3 Geometry Adaptation Experiment

The geometry adaptation experiment consists in the refinement of a VF geometry to comply with a possible environmental change. A simple yet effective method to perform the VF geometry adaptation respecting the human-in-the-loop paradigm is sought. This resulted in a procedure composed by the following steps: (i) the user can generate the desired VF geometry using the proposed interactive generation method presented in Sect. 6.3.1; (ii) the user can activate the VF by pressing a foot pedal and start performing the task; (iii) the user may want to deviate from the previously defined path to comply with the environment/plan change; thus she/he can deactivate the current VF and freely record new interaction points for generating a new VF path; (iv) the user can activate the latest generated VF by releasing the foot pedal and be aided during the task completion. An experiment involving the above defined steps has been performed. Referring to Fig. 6.5, the experiment starts with the first VF generation (time interval $[0, 13]\,\text{s}$) in which interaction points are recorded and the path is generated by the fitting algorithm[2]. The resulting VF geometry is shown in Fig. 6.5(a) as a solid red line, together with the corresponding interaction points (in pink). Subsequently, the VF is activated and the dissection task is started (time interval $[13, 32]\,\text{s}$). During the task the surgeon decides to change the previously defined VF geometry to comply with a possible plan/environmental change. First, the current VF is deactivated and a new VF generation phase is undertaken (time interval $[32, 51]\,\text{s}$). To deactivate the current VF the impedance is brought to zero following the variation law presented in (6.18) by pressing a foot pedal. Newly recorded points are fitted

[2]We note that the VF generation time is negligible with respect to recording phase, thus we can assume it is an instantaneous process.

as explained in Sect. 6.3.1 to generate a new VF. The resulting VF geometry is represented in Fig. 6.5(b) as a solid red line together with interaction points in gray. The VF is then activated to complete the task (time interval $[51, 70]$ s). Figure 6.5(c) contains the time history of the desired stiffness variation according to the laws (6.18) and (6.19). Even in this case, the passivity of the system is preserved by means of the energy tanks passivity-based control described in Sect. 6.4. Figure 6.5(d) contains the time evolution of the tank energy used to implement the stiffness variation. The effect of the passivity-based control action is evident in Fig. 6.5(c) (around time 51 s) where the tank is discharged and the stiffness is kept constant until it gets replenished. Finally, Figure 6.5(e) shows human operator's estimated forces on the master side during the task execution: it can be seen that when the VF is activated the user is aided by guidance forces in performing the task. A video showing both real and simulated experiments [5] is available at: `https://youtu.be/AnwzTuVuLAE`.

6.5.4 Discussion

When the impedance variation profile is known in advance, a state-independent stability constraint can be imposed [146]. However, this would imply the application of a fixed damping parameter on the system (opportunely defined on the basis of the maximum stiffness variation, *i.e.* worst case design) which degrades the system transparency when the bilateral teleoperation is enabled. The same applies to other passivity-based control techniques, such as the time domain passivity-based control approach [157]: here, the control action in the form of a dissipative element that absorbs the energy generated by the system causes, in general, the presence of higher damping on the system. These considerations motivated the use of the energy tanks passivity-based control approach.

Also, the proposed VF path generation, that uses interaction points recording, constitutes a simple yet effective method of programming a VF path in real-time. This can be used whenever the desired path is not definable pre-operatively. For instance, tumor resections are some of the most critical and precision demanding procedures that might benefit from the proposed approach. However, this can sometimes be time consuming and/or not very effective when large environmental deformations occur during the surgical procedure. In this case, vision systems might be adopted to increase the efficacy of the proposed method [100]. They can also be useful for the constraint enforcement definition: for instance, to estimate a complex surface normal direction to be left free. In any case, aside the methodology used for VF generation/update,

the developed passivity-based control technique can be equally applied.

6.6 Conclusions and Future Works

In this work, a methodology to generate and adapt VF pose and geometry in robot-aided minimally invasive surgical tasks has been introduced and experimentally validated. The methodology relies on a VF generation strategy in which recorded interaction points are used to create the constraint geometry. Pose and geometry adaptation strategies have been developed to allow flexible use of VF assistance. Energy tanks passivity-based control has been proposed to guarantee a stable behavior of the system. The devised strategies have been validated through experiments on the dVRK and have demonstrated stable and smooth behaviors.

The aim for the future is to combine the proposed adaptation strategies with task encoding [221, 222] and/or vision-based tracking methods for automatic VF switching/update. A human-subject evaluation involving novice and expert surgeons will be conducted to assess the effectiveness of our method. Ultimately, the generalization to other robotic surgical tasks will be considered, such as suturing [31]. It is believed that the proposed methodology may leverage the development of next generation assistive surgical systems that enable more precise and safer interventions.

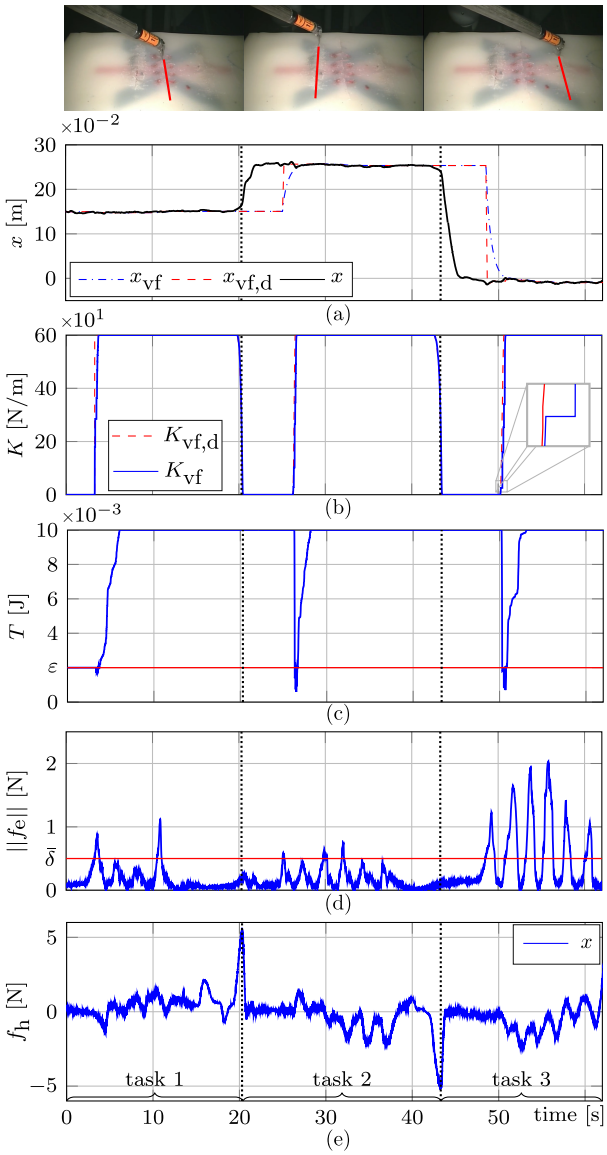


Figure 6.4: VF pose adaptation experiment. Time histories of: (a) VF and tool central point pose along the x direction; (b) stiffness; (c) energy tank level; (d) slave interaction force; (e) human operator force on the master side.

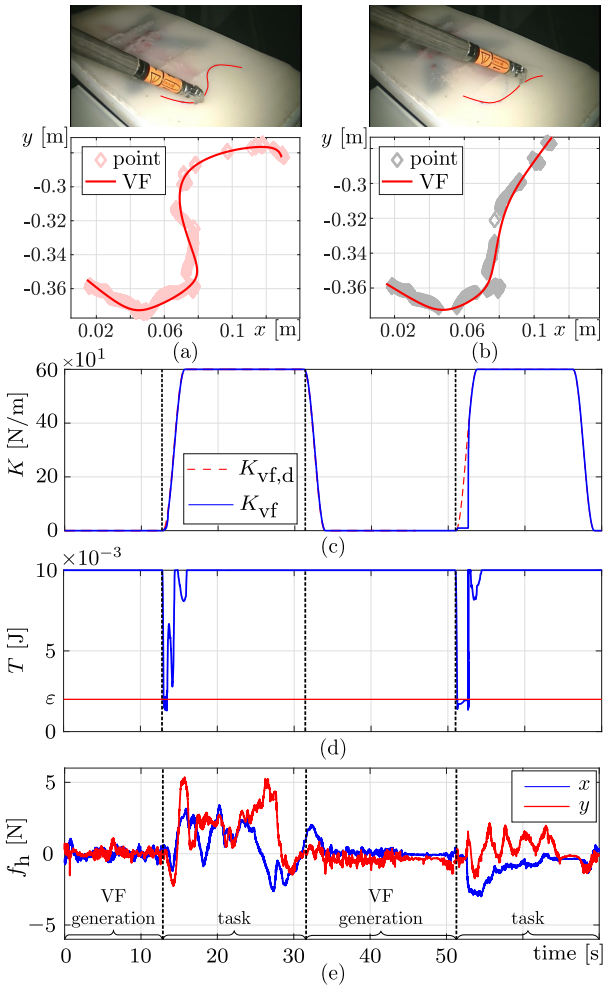


Figure 6.5: Geometry adapation experiment. (a) First VF geometry generated by the recorded interaction points. (b) Second VF geometry. (c-e) Time histories of: (c) stiffness; (d) tank energy level; (e) human operator estimated force on the master side.

Chapter 7

Shared Control for Needle Grasping Optimization

During suturing tasks performed with minimally invasive surgical robots, configuration singularities and joint limits often force surgeons to interrupt the task and re-grasp the needle using dual-arm movements. This yields an increased operator's cognitive load, time-to-completion and performance degradation. In this chapter, a haptic-guided shared control method for grasping the needle with the Patient Side Manipulator (PSM) of the da Vinci robot avoiding such issues is proposed. A cost function consisting of (i) the distance from robot joint limits and (ii) the task-oriented manipulability along the suturing trajectory is suggested. Evaluating the cost and its gradient on the needle grasping manifold allows obtaining the optimal grasping pose for joint-limit and singularity free robot movements during suturing. Force cues are computed and displayed through the Master Tool Manipulator (MTM) to guide the surgeon towards the optimal grasp. As such, the proposed system helps the operator to choose a grasping configuration that allows the robot to avoid joint limits and singularities during post-grasp suturing movements. The effectiveness of the proposed haptic-guided shared control method is shown during suturing using both simulated and real experiments. The results illustrate that the approach significantly improves the performance in terms of needle re-grasping [223].

7.1 Introduction

Autonomous control methods are not yet sufficiently trusted in robotic surgery because of safety-critical and high-consequence tasks to perform. On the other

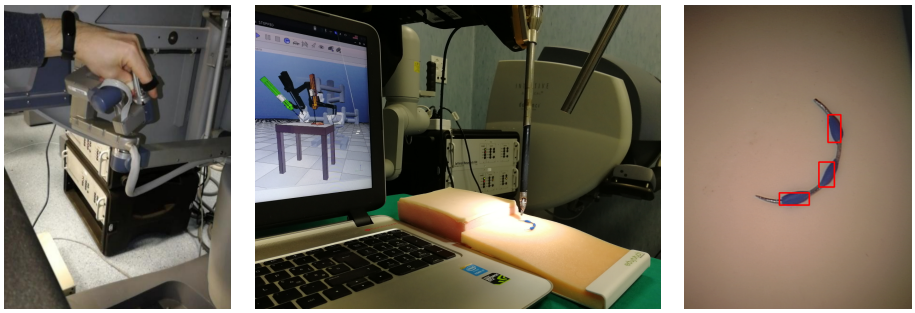
hand, remote teleoperation of surgical robotic systems imposes extreme cognitive loading to the human operator, causing severe fatigue and, consequently, a progressive degeneration in performance. However, patients are highly conservative and trust more a surgeon who remotely teleoperates the robotic system within the patient's body through a master console as in the da Vinci robot (see Fig. 7.1).

Haptic-guided shared control is a promising approach to reduce the human operator's cognitive load during teleoperation. This improves performance by providing haptic cues that guide the operator in carrying out safe and effective actions. For instance, Ghalamzan *et al.* in [163] showed that haptic cues, guiding the human operator based on a predictive cost of avoiding singularities during the teleoperation task, drastically improve the manipulation performance.

In da Vinci-like surgical systems, while tremor filtering and motion scaling enhance the movements precision, the robot mechanical structure also imposes some constraints. In particular, joint limits and singularities are common issues that, for example, force the surgeon to re-grasp the needle in the middle of the suturing task. As such, surgeons waste a considerable amount of time reconfiguring the robot during interventions despite the substantial training they undergo. Fontanelli *et al.* in [31] noticed that the suturing task, performed by a surgeon using a da Vinci robot, requires several hand-off movements using both arms to re-orient the needle before each stitch. This results in longer procedural time, increased fatigue, higher cognitive loading and degenerating performance.

This chapter proposes a haptic-guided shared control system that assists the surgeon in avoiding such problems during suturing tasks performed with a da Vinci robot. The proposed approach *'guides'* the operator during the reach-to-grasp phase to choose a needle grasping configuration that results in neither joint limits nor singularities in the course of suturing. The chosen grasping pose allows starting the suturing procedure immediately with no need for reconfigurations during post-grasp manipulative movements, *i.e.* over the course of the suturing trajectory. While the haptic-guided shared control system informs the user about the optimal grasping configuration, it allows the operator to ultimately choose the grasp making a trade-off between the computed cost and other non-modeled objectives.

The rest of this chapter is structured as follows. Section 7.2 describes the related work in robotic grasping and manipulation, needle grasping and haptic-guided shared control fields. Section 7.3 formulates the problem. Needle grasping optimization and haptic guidance are described in Section 7.4



(a) Master system (b) Slave system (c) Needle tracking

Figure 7.1: Experimental setup comprising of master (a) and slave (b) systems of da Vinci Research Kit. An operator commands the slave tool through the master device to grasp a needle. Figure (c) shows the tracking system to retrieve the needle pose.

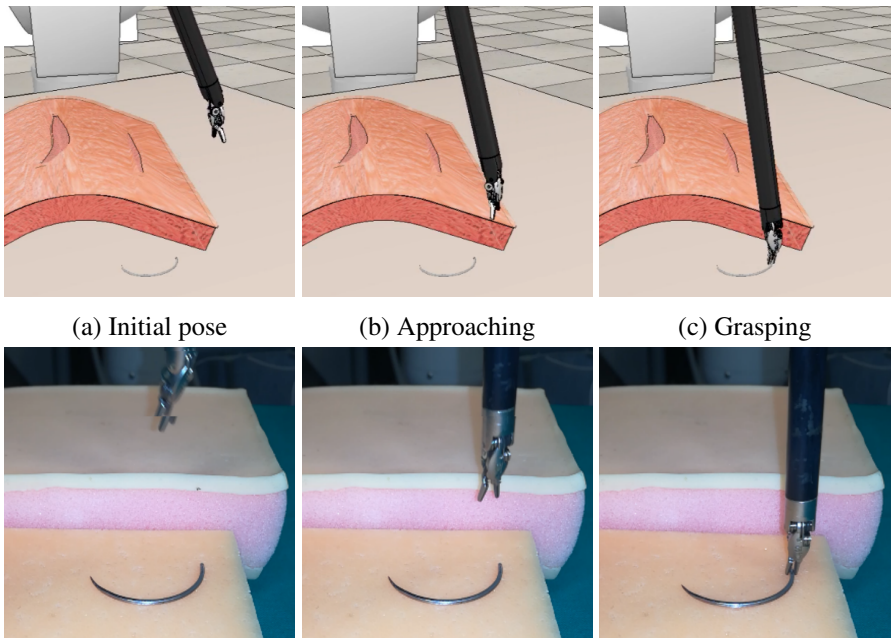
and 7.5, respectively. Section 7.6 presents simulations and experiments performed to evaluate the developed system. Section 7.7 concludes the paper.

7.2 Related Work

Performing a successful robotic manipulation task involves: robot approaching an object (g_0: reach-to-grasp); making stable contacts on the object surface (g_1: grasping); and moving the object to perform the task (g_2: post-grasp). Figures 7.2a-7.2b-7.2c and 7.2d-7.2e-7.2f, illustrate g_0 and g_1 phases for grasping a needle in simulated and real experiments, respectively. State-of-the-art approaches are able to efficiently computing stable grasp configurations (g_1) from a point cloud of a scene with high success rate (*e.g.* using probabilistic learning algorithms [224] or artificial neural networks [225, 226]).

In the robotic surgery domain, D'Ettorre *et al.* proposed in [227] a vision-guided method for autonomous needle grasping which allows the surgeon to immediately start suturing. Nevertheless, the authors only focused on planning the success of g_0 and g_1. For manipulation tasks, planning g_1 and g_2 cannot be performed in isolation as a chosen grasping configuration may lead to encounter system constraints during g_2: although the obtained needle grasping yields stable contacts, it may be impossible for the robot arm to deliver the entire intended suturing movement.

Autonomously performing $g_0 - g_1 - g_2$ is not sufficiently trusted by conservative industries in safety-critical tasks, such as robotic surgery. However,



(a) Initial pose (b) Approaching (c) Grasping

(d) Initial pose (e) Approaching (f) Grasping

Figure 7.2: Grasping a needle using the PSM in a simulated ((a), (b) and (c)) and a real ((d), (e) and (f)) environment: our proposed haptic-guided shared control system generates force cues guiding the operator to choose a grasping configuration yielding neither joint limits nor singularities during suturing movements.

there exist efficient approaches to autonomously perform g_2. For instance, motion planning algorithms, such as learning from demonstration [228], generalize demonstrated motions to new knot tying examples using a da Vinci robot [229]. The generalized motions are known before $g_0 - g_1$, thus allowing the computation of limitations the robot will face during g_2 given a certain grasp. Some g_2 objectives relevant to $g_0 - g_1$ may include the object affordance [230], mass distribution [231] and collisions with the environment [232]. In contrast, a human operator is not able to foresee these issues during the future movements resulting in several cycles of (i) planning, (ii) executing part of the planned motions, (iii) facing a limitation and (iv) re-planning.

By inspecting the JIGSAW dataset [233], Fontanelli *et al.* noted that the occurrence of hand-off movements during surgical sub-tasks is significant and yields an increased procedural time [31]. They proposed to use an additional DoF in the surgical robotic tool allowing in-hand needle re-orientation. Sen *et al.* proposed in [234] a mechanical device designed to align and hold the needle in a known orientation. The authors' aim was that of improving needle pose tracking capabilities of a computer vision software for automating multi-throw surgical suturing. However, these approaches require substantial modification of the robot mechanical structure that are not always practically feasible.

On the other hand, an operator can be successfully guided towards an optimal grasping configuration. To this end, haptic guidance was used to avoid constraints during g_0 and g_2 in [190, 235, 163]. These works reported significant performance improvement compared to direct teleoperation. However, they were only developed to grasp and manipulate simple objects whereas they did not account for the particular object shape. In this case, force cues, which continuously attract the operator's towards optimal grasping poses, may have non-intuitive components pushing away from the object shape.

Liu *et al.* presented in [236, 4] an offline optimization-based solution to needle grasping and robotic instruments entry ports selection. An exhaustive search method was used to find the optimal needle grasping and entry ports poses. However, this method cannot be utilized for real time implementation of haptic-guided shared-control systems due to the high computation time.

In this chapter, a haptic-guided shared control method for suturing tasks with a da Vinci robot to assist the operator during g_0 is proposed. A cost function which is comprised of two terms is used: (i) joint limits and (ii) task-oriented velocity (TOV) manipulability[1]. In contrast to previous works [163,

[1]In contrast, [4] used the classical manipulability measure. The superiority of TOV over classical manipulability in manipulation tasks is discussed in [163].

190], the shape of the object to be grasped, *i.e.* the needle, is known. To eliminate the force cues non-intuitive components, the cost is computed and its gradient is projected onto the space of feasible grasping pose (*grasping manifold*). Hence, the non-intuitive components are eliminated and force cues always attract the operator's hand towards the needle shape.

7.3 Problem Formulation

Let us consider the Patient Side Manipulator (PSM) of a da Vinci surgical robot performing a suturing task (Fig. 7.1b). The PSM is a 6-DoF manipulator with $\boldsymbol{q}_s \in \mathbb{R}^6$ being the set of its generalized coordinates (see [5] for a detailed PSM kinematic description). An operator commands the PSM through the da Vinci Master Tool Manipulator (MTM)(Fig. 7.1a) to grasp a needle (Fig. 7.1c) and performing a suturing task. Stitching trajectories can be computed based on the desired needle insertion and extraction points and include approaching the insertion point on a tissue, piercing, reaching the desired extraction point, piercing and pushing the needle out of the tissue [3]. We assume that the needle trajectory necessary to perform the stitching is given (Fig. 7.3) by either a planning algorithm [237] or learning from demonstration [238].

We denote by $\mathcal{F}_n : \{O_n; x_n, y_n, z_n\}$ a local frame attached to the needle and with $\mathcal{F}_r : \{O_r; x_r, y_r, z_r\}$ an inertial reference frame. $\mathcal{F}_n$ can be expressed in $\mathcal{F}_r$ through the transformation matrix[2] ${}^r\boldsymbol{T}_n \in \mathrm{SE}(3)$

$$ {}^r\boldsymbol{T}_n(t) = \left[\begin{array}{c|c} \boldsymbol{R}_{3\times3}(t) & \boldsymbol{p}_{3\times1}(t) \\ \hline \boldsymbol{O}_{1\times3} & 1 \end{array}\right], \tag{7.1} $$

where, as usual, $\boldsymbol{p} \in \mathbb{R}^3$ and $\boldsymbol{R} \in \mathrm{SO}(3)$ are the position and orientation of any point in Cartesian space. We denote by $\mathcal{F}_t : \{O_t; x_t, y_t, z_t\}$ a reference frame attached to the needle tip. The trajectory to be followed by the needle implies that $\mathcal{F}_t$ matches a sequence of planned stitching poses (see Fig. 7.3)

$$ \mathcal{F}_p(t) = \zeta(t) \qquad t \in [0, T], \tag{7.2} $$

where ζ is the stitching trajectory, t denotes the time and T is the total time to complete the stitching. In other words, $\mathcal{F}_p(t)$ determines a complete object pose (position and orientation) at every time t along the trajectory $\boldsymbol{\zeta}$.

Let us denote a local frame attached to the PSM end-effector by $\mathcal{F}_e : \{O_e; x_e, y_e, z_e\}$. Let also $\mathcal{F}_g : \{O_g; x_g, y_g, z_g\}$ be a local reference frame

[2] In general, ${}^b\boldsymbol{T}_a \in \mathrm{SE}(3)$ denotes a homogeneous transformation matrix representing the pose of local frame a into local frame b.

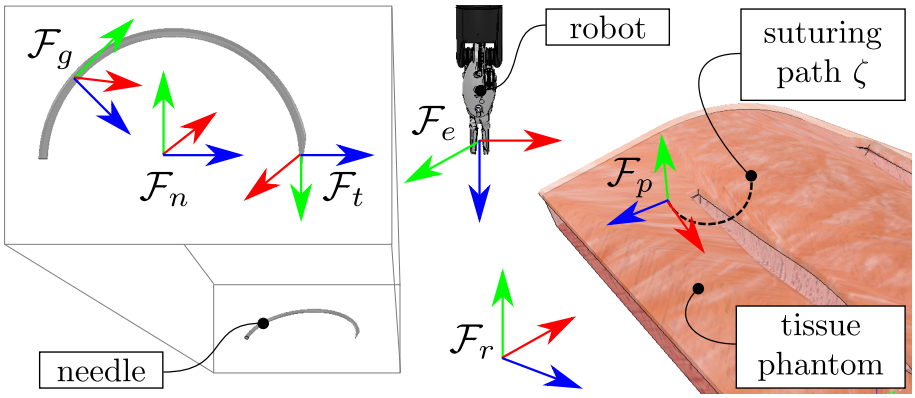


Figure 7.3: A schematic surgical setting during the suturing task: $\mathcal{F}_r$, $\mathcal{F}_e$, $\mathcal{F}_n$ and $\mathcal{F}_t$ denote the inertial, the end-effector, the needle center and the needle tip reference frames, respectively. Furthermore, $\mathcal{F}_g$ represents a frame attached to the needle corresponding to a generic desired grasping pose for the robot end-effector. The needle tip trajectory (dashed black curve) from the insertion point (rightmost) to the exit point (leftmost) is assumed to be given by motion planning or learning by demonstration techniques. $\mathcal{F}_p$ denotes a generic needle-tip trajectory pose, *i.e.* position and orientation.

attached to the needle shape which serves as a grasping pose candidate for the robot end-effector. Since the needle is not deformable and the PSM end-effector forms stable contacts on the needle surface, the PSM end-effector pose $\mathcal{F}_g$ becomes equal to $\mathcal{F}_e$ once contacts are made. As such, $\mathcal{F}_g$ can be fully expressed at all time by a fixed transformation matrix, namely ${}^{n}\boldsymbol{T}_g$, w.r.t. needle local frame $\mathcal{F}_n$. The PSM end-effector trajectory during suturing can be derived given the planned needle trajectory $\boldsymbol{\zeta}$ in (7.2) as follows

$$ {}^{r}\boldsymbol{T}_g(t) = \{ {}^{r}\boldsymbol{T}_n(t) {}^{n}\boldsymbol{T}_g : 0 \leq t \leq T \}. \tag{7.3} $$

Finally, the post-grasp joint configuration trajectory corresponding to a given grasping ${}^{r}\boldsymbol{T}_e\,(t) = {}^{r}\boldsymbol{T}_g\,(t)$, can be computed as follows

$$ \hat{\boldsymbol{q}}_g\,(t) = \mathrm{IK}\,({}^{r}\boldsymbol{T}_g\,(t)) \tag{7.4} $$

where $\mathrm{IK}\,(\cdot)$ is the PSM inverse kinematics function which computes the joint space trajectory $\hat{\boldsymbol{q}}_g(t)$ corresponding to the grasping frame trajectory $\mathcal{F}_g(t)$. The problem is to find ${}^{n}\boldsymbol{T}_g$ in (7.3) and thus $\hat{\boldsymbol{q}}_g\,(t)$ in (7.4) which yields no singularities or joint limits along $\boldsymbol{\zeta}$ in (7.2).

7.4 Needle Grasping

To successfully perform a suturing task without hands-off movements, we guide the human operator towards a grasping pose which yields no singularities or joint limits during the post-grasping suturing trajectory. In contrast to previous studies [163, 190], here the geometrical shape of the object to be grasped is known. To obtain consistent results, one should account for the needle geometry in the problem formulation and optimize in the space of feasible grasping poses. In Sect. 7.4.1, parameterization of needle grasping poses is discussed while in Sect. 7.4.2 the joint limits and singularities cost function and the grasping manifold optimization method are proposed.

7.4.1 Grasp Manifold Parametrization

To grasp a curved needle with the PSM gripper, it is necessary to align the y_e-axis of $\mathcal{F}_e$ (green axis in Fig. 7.3)[3] with the needle tangent (this results in a 2-DoF rotational constraint). In addition, grasping requires the gripper position to belong to the needle shape (additional 2-DoF linear constraint) [4]. The remaining subspace of possible grasping configurations (2-DoF) can be conveniently parametrized using the needle curvilinear abscissa $n \in [0, n^\star]$ and the angle around the needle tangent $\alpha \in [\alpha^-, \alpha^+]$ (see Fig. 7.4). Let us denote by $\boldsymbol{z} = [n, \alpha]^T$ the vector identifying any point in the considered subspace $\mathcal{Z}$ denoting the needle grasping manifold (locally $\mathcal{Z} \subseteq \mathbb{R}^2$). Hence, $\mathcal{F}_e$ can be explicitly expressed as function of $\boldsymbol{z}$ at grasping through (7.3) with ${}^n\boldsymbol{T}_g$ being

$$
\begin{aligned}
{}^n\boldsymbol{T}_g(\boldsymbol{z}) &= \boldsymbol{T}_p(n)\boldsymbol{T}_r(\alpha) = \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & s(n\boldsymbol{\pi}) & c(n\boldsymbol{\pi}) & -rc(n\boldsymbol{\pi}) \\ 0 & -c(n\boldsymbol{\pi}) & s(n\boldsymbol{\pi}) & -rs(n\boldsymbol{\pi}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c(\alpha) & 0 & s(\alpha) & 0 \\ 0 & 1 & 0 & 0 \\ -s(\alpha) & 0 & c(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{7.5}
$$

where $c(\cdot)$ and $s(\cdot)$ are shorthand notations for the $\cos(\cdot)$ and $\sin(\cdot)$ functions, r is the needle radius and n, α have been introduced above. Denoting with $\dot{\boldsymbol{x}}_e = [\dot{\boldsymbol{p}}^T, \boldsymbol{\omega}^T]^T$ the stacked vector of the PSM linear ($\dot{\boldsymbol{p}} \in \mathbb{R}^3$) and angular ($\boldsymbol{\omega} \in \mathbb{R}^3$) velocities at grasping it yields

$$
{}^r\dot{\boldsymbol{x}}_e = {}^r\bar{\boldsymbol{R}}_n {}^n\dot{\boldsymbol{x}}_g = {}^r\bar{\boldsymbol{R}}_n \boldsymbol{J}_g(\boldsymbol{z})\,\dot{\boldsymbol{z}}, \quad {}^r\bar{\boldsymbol{R}}_n = \begin{bmatrix} {}^r\boldsymbol{R}_n & \boldsymbol{O} \\ \boldsymbol{O} & {}^r\boldsymbol{R}_n \end{bmatrix} \tag{7.6}
$$

[3] Subscript $*_e$ refers to the end-effector during reach-to-grasp whereas $*_g$ refers to the end-effector after making stable contacts between the end-effector and the needle.

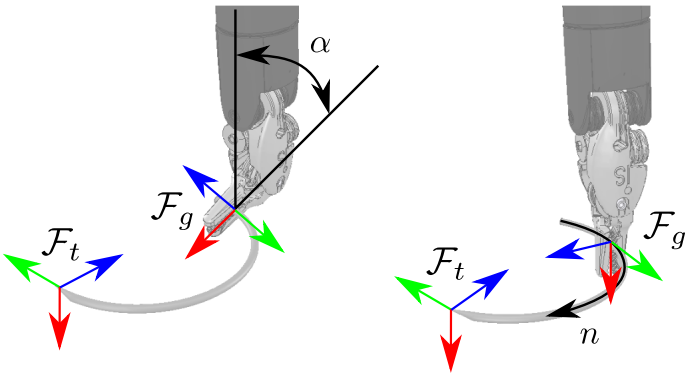


Figure 7.4: Grasp parameterization: grasping configurations can be parametrized by α, representing the angle around the needle tangent, and n, representing the needle curvilinear abscissa (see also [3, 4]).

where ${}^{r}\bar{\boldsymbol{R}}_{n} \in \mathbb{R}^{6\times6}$ transforms the twist $\dot{\boldsymbol{x}}_{g}$ from $\mathcal{F}_{n}$ to $\mathcal{F}_{r}$, $\boldsymbol{J}_{g}(\boldsymbol{z}) \in \mathbb{R}^{6\times2}$ is the *grasp Jacobian* specific to the object shape and the choice of grasping parameters, and mapping the grasping manifold velocities $\dot{\boldsymbol{z}}$ into ${}^{n}\dot{\boldsymbol{x}}_{g}$. Combining the PSM differential forward kinematics, *i.e.* ${}^{r}\dot{\boldsymbol{x}}_{e} = \boldsymbol{J}_{s}(\boldsymbol{q}_{s})\dot{\boldsymbol{q}}_{s}$ (where $\boldsymbol{J}_{s}$ is the conventional PSM geometric Jacobian), and (7.5)-(7.6) yields

$$\dot{\boldsymbol{q}}_{s} = \boldsymbol{J}_{s}^{\dagger}(\boldsymbol{q}_{s})^{r}\bar{\boldsymbol{R}}_{n}\boldsymbol{J}_{g}(\boldsymbol{z})\,\dot{\boldsymbol{z}}, \tag{7.7}$$

where $\boldsymbol{J}_{s}^{\dagger}$ denotes the usual $\boldsymbol{J}_{s}$ Moore-Penrose pseudoinverse. Equation (7.7) will be exploited in the following Section to optimize the needle grasping pose.

7.4.2 Optimal Grasping Pose Selection

As previously explained, the interest is to avoid joint limits and singularities during post-grasp movements, *i.e.* along the suturing trajectory. As such, let us define a cost function accounting for joint limits, *i.e.*

$$h_{j}\left(\hat{\boldsymbol{q}}_{g}(t)\right) = \sum_{i=1}^{n} \frac{1}{\lambda} \frac{\left(\hat{\boldsymbol{q}}_{g,i}^{+} - \hat{\boldsymbol{q}}_{g,i}^{-}\right)^{2}}{\left(\hat{\boldsymbol{q}}_{g,i}^{+} - \hat{\boldsymbol{q}}_{g,i}(t)\right)\left(\hat{\boldsymbol{q}}_{g,i}(t) - \hat{\boldsymbol{q}}_{g,i}^{-}\right)}, \tag{7.8}$$

and another accounting for task-oriented manipulability, *i.e.*

$$h_{s}\left(\hat{\boldsymbol{q}}_{g}(t)\right) = {}^{r}\dot{\hat{\boldsymbol{x}}}_{e}^{\mathrm{T}}(\boldsymbol{J}_{s}(\hat{\boldsymbol{q}}_{g}(t))\boldsymbol{J}_{s}(\hat{\boldsymbol{q}}_{g}(t))^{T})^{-1}{}^{r}\dot{\hat{\boldsymbol{x}}}_{e}, \tag{7.9}$$

where $\lambda \in \mathbb{R}^{+}$ is a constant scalar, $\hat{\boldsymbol{q}}_{g,i}(t)$ is the i^{th} joint coordinate at time t, $\hat{\boldsymbol{q}}_{g,i}^{+}$ and $\hat{\boldsymbol{q}}_{g,i}^{-}$ are its corresponding upper and lower limits, respectively, and

${}^r\hat{\dot{\boldsymbol{x}}}_e = {}^r\dot{\boldsymbol{x}}_e/||{}^r\dot{\boldsymbol{x}}_e||$ denotes the velocity direction along which the manipulability is measured. Thus, the total cost is (omitting t dependence)

$$h\left(\hat{\boldsymbol{q}}_g\right) = h_j\left(\hat{\boldsymbol{q}}_g\right) + h_s\left(\hat{\boldsymbol{q}}_g\right). \tag{7.10}$$

We parametrize the single stitch suturing trajectory (given in (7.2)) with $s \in [0, s^\star]$ obtaining ${}^r\boldsymbol{T}_n(s)$. Substituting this in (7.3) and using (7.5) yields ${}^r\boldsymbol{T}_g(s, \boldsymbol{z}) = {}^r\boldsymbol{T}_n(s)^n\boldsymbol{T}_g(\boldsymbol{z})$. Thus, given ${}^r\boldsymbol{T}_g(s, \boldsymbol{z})$ we can compute $\hat{\boldsymbol{q}}_g\left(s, \boldsymbol{z}\right)$ through (7.4). Hence, the cost function evaluated along the trajectory can be expressed as a function of s and $\boldsymbol{z}$ as follows

$$\mathcal{H}\left(\boldsymbol{z}\right) = \int_0^{s^\star} h\left(\hat{\boldsymbol{q}}_g\left(s, \boldsymbol{z}\right)\right) \mathrm{d}s. \tag{7.11}$$

The aim is to find the vector $\boldsymbol{z}$ that minimizes the cost function in (7.11). Mathematically, the problem writes as follows

$$\begin{aligned} \underset{\boldsymbol{z}}{\text{minimize}} \quad & \mathcal{H}\left(\boldsymbol{z}\right) \\ \text{subject to} \quad & \alpha^- \leq \alpha \leq \alpha^+. \\ & 0 \leq n \leq n^\star \end{aligned} \tag{7.12}$$

The problem in (7.12) is clearly non-convex and may have local minima that can be reached through iterative methods. For the purpose, the gradient descent is used, *i.e.* a first-order iterative optimization algorithm, to find a local minimum of the cost function in (7.11). At each optimization step $\boldsymbol{z}$ is updated as in the following succession

$$\boldsymbol{z}_{n+1} = \boldsymbol{z}_n - \gamma \nabla_{\boldsymbol{z}} \mathcal{H}, \tag{7.13}$$

where $\gamma \in \mathbb{R}^+$ represents the step increment and $\nabla_{\boldsymbol{z}}\mathcal{H}$ the cost function gradient with respect to $\boldsymbol{z}$. Convergence to a local minimum can be guaranteed and is not problematic for the scope since only a locally optimized solution is sought. Equation (7.13) requires the computation of $\nabla_{\boldsymbol{z}}\mathcal{H}$. Exploiting Leibniz's formulas, this can be written as

$$\nabla_{\boldsymbol{z}}\mathcal{H} = \frac{\partial \mathcal{H}}{\partial \boldsymbol{z}} = \int_0^{s^\star} \frac{\partial h}{\partial \boldsymbol{z}} \mathrm{d}s, \tag{7.14}$$

and using the chain rule, one can additionally write

$$\frac{\partial h}{\partial \boldsymbol{z}} = \frac{\partial h}{\partial \boldsymbol{q}_s} \frac{\partial \boldsymbol{q}_s}{\partial \boldsymbol{z}}. \tag{7.15}$$

where the term $\partial h/\partial \boldsymbol{q}_s$ is the derivative of the cost function with respect to the generalized coordinates vector of PSM and tells in which joint space direction the cost function increases most. The term $\partial \boldsymbol{q}_s/\partial \boldsymbol{z}$ can be computed from (7.7) as follows

$$\frac{\partial \boldsymbol{q}_s}{\partial \boldsymbol{z}} = \boldsymbol{J}_s^{\dagger}(\boldsymbol{q}_s)^r \bar{\boldsymbol{R}}_n \boldsymbol{J}_g(\boldsymbol{z}), \tag{7.16}$$

and depends only upon the robot kinematics and the chosen grasp parametrization. Substituting in (7.15) yields

$$\frac{\partial h}{\partial \boldsymbol{z}} = \frac{\partial h}{\partial \boldsymbol{q}_s} \boldsymbol{J}_s^{\dagger}(\boldsymbol{q}_s)^r \bar{\boldsymbol{R}}_n \boldsymbol{J}_g(\boldsymbol{z}). \tag{7.17}$$

The partial derivatives of (7.8) and (7.9) w.r.t. $\boldsymbol{q}_s$ can be easily computed analytically (see [163, 190]) and then plugged in (7.17) to evaluate (7.14), thus finding the optimal grasping parameter vector $\boldsymbol{z}^*$ according to (7.13). The optimal Cartesian pose for the PSM $\boldsymbol{x}_{g,d}$ can be easily calculated from the optimal grasping parameter vector $\boldsymbol{z}^*$ given the needle kinematics (7.5) and its global pose (7.1).

7.5 Haptic Guidance

As discussed above, the purpose of the haptic guidance is to guide the user toward the optimal Cartesian space grasping pose. Let $\boldsymbol{x}_{g,d} = [\boldsymbol{p}_{g,d}^{\mathrm{T}}, \boldsymbol{\phi}_{g,d}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^6$ be the optimal desired pose for the PSM end-effector frame $\mathcal{F}_e$ with $\boldsymbol{p}_{g,d} \in \mathbb{R}^3$ denoting the position and $\boldsymbol{\phi}_{g,d} \in \mathbb{R}^3$ any parametrization of the orientation, *e.g.* Euler angles. $\boldsymbol{x}_{g,d}$ can be calculated as discussed in Sect. 7.4.2. The corresponding desired MTM pose $\boldsymbol{x}_{m,d} = [\boldsymbol{p}_{m,d}^{\mathrm{T}}, \boldsymbol{\phi}_{m,d}^{\mathrm{T}}]^{\mathrm{T}}$ can be calculated from $\boldsymbol{x}_{g,d}$ through the following master-slave transformation

$$\boldsymbol{x}_{m,d} = \bar{\boldsymbol{R}}_c \boldsymbol{x}_{g,d} + \boldsymbol{x}_c, \qquad \bar{\boldsymbol{R}}_c = \begin{bmatrix} \boldsymbol{R}_c & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{R}_c \end{bmatrix} \tag{7.18}$$

where $\bar{\boldsymbol{R}}_c \in \mathbb{R}^{6\times 6}$ is the master-slave coupling rotation matrix and $\boldsymbol{x}_c = [\boldsymbol{p}_c^{\mathrm{T}}, \boldsymbol{\phi}_c^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^6$ its offset. Given $\boldsymbol{x}_{m,d}$, haptic cues can be displayed on the MTM using impedance control. The MTM is a 7-DoF serial robot arm with $\boldsymbol{q}_m \in \mathbb{R}^7$ denoting the vector of its generalized coordinates. The MTM exhibits the following joint space dynamics

$$\boldsymbol{M}_m(\boldsymbol{q}_m)\ddot{\boldsymbol{q}}_m + \boldsymbol{C}_m(\boldsymbol{q}_m, \dot{\boldsymbol{q}}_m)\dot{\boldsymbol{q}}_m + \boldsymbol{B}_m\dot{\boldsymbol{q}}_m + \boldsymbol{g}_m(\boldsymbol{q}_m) = \boldsymbol{u}_m - \boldsymbol{J}_m^{\mathrm{T}}(\boldsymbol{q}_m)\,\boldsymbol{w}_{\mathrm{h}}, \tag{7.19}$$

where $\boldsymbol{M}_m(\boldsymbol{q}_m) \in \mathbb{R}^{7\times7}$ is the symmetric and positive-definite joint space inertia matrix, $\boldsymbol{C}_m(\boldsymbol{q}_m,\, \dot{\boldsymbol{q}}_m) \in \mathbb{R}^{7\times7}$ consists of the Coriolis/centrifugal terms, $\boldsymbol{B}_m \in \mathbb{R}^{7\times7}$ accounts for the friction term, $\boldsymbol{g}_m \in \mathbb{R}^7$ is the gravity term and $\boldsymbol{u}_m \in \mathbb{R}^7$ is a vector of control torques used to display haptic guidance forces to the operator. To realize a linear and decoupled Cartesian space impedance the control, inputs are designed as follows

$$\boldsymbol{u}_m = \boldsymbol{M}_m\left(\boldsymbol{q}_m\right)\boldsymbol{v} + \boldsymbol{C}_m(\boldsymbol{q}_m, \dot{\boldsymbol{q}}_m)\dot{\boldsymbol{q}}_m + \boldsymbol{g}_m\left(\boldsymbol{q}_m\right) + \boldsymbol{J}_m^{\mathrm{T}}\left(\boldsymbol{q}_m\right)\boldsymbol{w}_{\mathrm{h}} \tag{7.20}$$

$$\begin{aligned}\boldsymbol{v} = \boldsymbol{J}_{m,\mathrm{A}}^{\dagger}\left(\boldsymbol{q}_m\right)\boldsymbol{M}_m^{-1}(\boldsymbol{M}_m\ddot{\boldsymbol{x}}_d + \boldsymbol{K}_p\tilde{\boldsymbol{x}}_m + \boldsymbol{K}_d\dot{\tilde{\boldsymbol{x}}}_m + \\ -\boldsymbol{M}_m\dot{\boldsymbol{J}}_{m,\mathrm{A}}\left(\boldsymbol{q}_m, \dot{\boldsymbol{q}}_m\right)\dot{\boldsymbol{q}}_m - \boldsymbol{w}_{\mathrm{h,A}}),\end{aligned} \tag{7.21}$$

where $\tilde{\boldsymbol{x}}_m = \boldsymbol{x}_{m,d} - \boldsymbol{x}_m \in \mathbb{R}^6$ with $\boldsymbol{x}_{m,d}$ given by (7.18), $\boldsymbol{J}_m\left(\boldsymbol{q}\right)$, $\boldsymbol{J}_{m,\mathrm{A}}\left(\boldsymbol{q}\right) \in \mathbb{R}^{6\times7}$ are MTM geometric and the analytical Jacobians, respectively, $\boldsymbol{K}_p, \boldsymbol{K}_d \in \mathbb{R}^{6\times6}$ are diagonal and positive-definite gain matrices to be opportunely designed and $\boldsymbol{w}_{\mathrm{h}} \in \mathbb{R}^6$ is the vector of the external forces applied by an interacting user. Notice that the term $\boldsymbol{w}_{\mathrm{h,A}}$ differs from $\boldsymbol{w}_{\mathrm{h}}$ by a mapping depending on the adopted orientation representation. The choice in (7.20) and (7.21) allows realizing the following linear and decoupled Cartesian space impedance

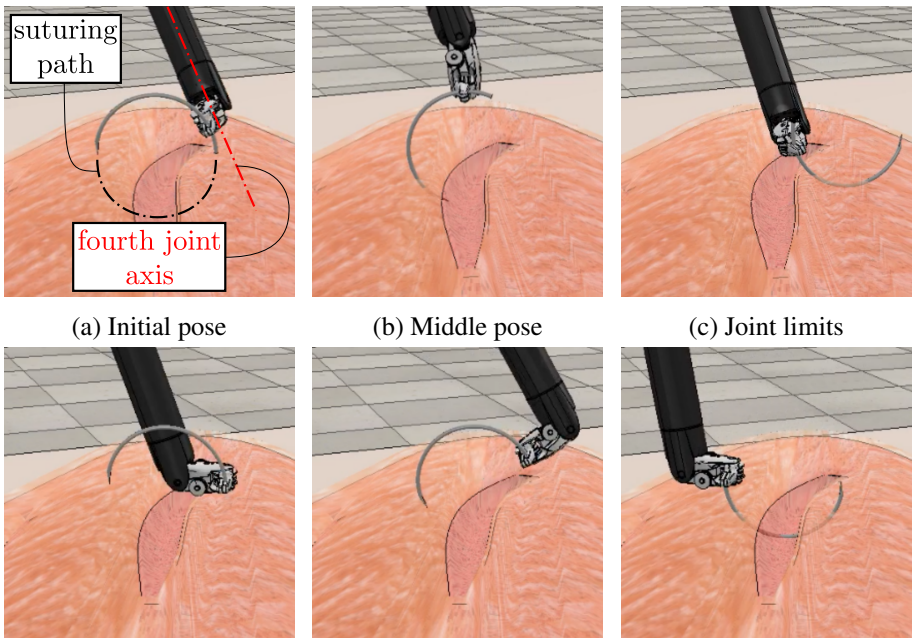
$$\boldsymbol{M}_m\ddot{\tilde{\boldsymbol{x}}}_m + \boldsymbol{K}_d\dot{\tilde{\boldsymbol{x}}}_m + \boldsymbol{K}_p\tilde{\boldsymbol{x}}_m = \boldsymbol{w}_{\mathrm{h,A}}, \tag{7.22}$$

which represents the sought haptic-guided shared control system. More details on the use of impedance control for haptic rendering with the dVRK robot are given in [198].

7.6 Experiments and Results

7.6.1 Experimental Setup

In this section, experimental results using the dVRK platform (see Fig. 7.1a and 7.1b) are shown. One PSM is commanded by one MTM through the standard dVRK teleoperation architecture. The MTM impedance control was previously developed using the robot dynamic parameters identified in [21]. External force measurement is required by the impedance control laws (7.20) and (7.21). Since forces applied to the MTM are not directly measurable, estimation is performed through the nonlinear dynamic observer developed in [216]. The application of torque inputs is possible thanks to the open-source hardware and software architecture developed in [214]. The complete list of parameters used in this work are shown in Tab. 7.1, while PSM joint limits are given in Tab. 7.2. The needle pose estimation follows the work in [227]



(a) Initial pose (b) Middle pose (c) Joint limits

(d) Initial pose (e) Near-singularity pose (f) Final pose

Figure 7.5: The first row ((a)-(b)-(c)) shows the effect of the joint limits occurrence during the suturing trajectory execution: the needle deviates from the desired trajectory when the fourth joint (see [5] for more information on PSM kinematics) reaches its upper limit (c). The second row ((d)-(e)-(f)) shows the singularity occurrence: the needle follows the path accurately but the manipulator undergoes large joint velocities in (e). Snapshot (e) corresponds to $t = 0.3$ s in Fig. 7.6c.

.

(see Fig. 7.1c). The PSM inverse kinematics is implemented using the inverse Jacobian Closed Loop Inverse Kinematics Algorithm (CLIK) [11]. The overall pipeline is first tested in simulation using the recently developed dVRK V-REP simulator [5]. The control loop rate is fixed at 5 ms while the lower level teleoperation loop runs at 1 ms.

7.6.2 Results

Three experiments are designed to show that joint limits and singularities create serious issues. The first two experiments demonstrate the occurrence of these constraints during an autonomous suturing trajectory execution. The last experiment shows the effectiveness of our proposed haptic-guided shared control in choosing an optimal needle grasping pose (that yields neither joint limits nor singularities during the suturing task) and in guiding the operator towards it. The experiments are also shown in the accompanying video.

In Experiment 1, a single stitch semi-circular trajectory (see Fig. 7.9) is defined between $\boldsymbol{p}_i = [-0.08, -0.08, -0.106]$ and $\boldsymbol{p}_e = [0.14, -0.08, -0.106]$ with center at $\boldsymbol{c} = [0.03, -0.08, -0.105]$ where $\boldsymbol{p}_i$, $\boldsymbol{p}_e$ and $\boldsymbol{c}$ are insertion point, exit point and the needle center position during stitching, respectively. All the coordinates are expressed in the world reference frame $\mathcal{F}_r$ shown in Fig. 7.3. The needle is grasped at $\boldsymbol{z} = [0.15, 0.0]$, see Fig. 7.4. This grasping conimguration causes the robot to encounter joint limits during the suturing trajectory as shown in Figs. 7.5a-7.5b-7.5c. We apply a saturation policy in the joint values to simulate encountering joint limits. Figures 7.5a-7.5b-7.5c show that the needle tip diverges from the desired suturing trajectory due to the joint limit occurrence. This may be experienced by the surgeon quite often as it is not trivial for her/him to predict the joint limits and avoid them during the reach-to-grasp phase. Fig. 7.9 shows the desired (solid blue line) and the actual (red dashed line) trajectories. Furthermore, Fig. 7.6a shows the PSM joint values during the experiment. The joint limits occurrence time slot is highlighted by the yellow shaded area. The joint exceeding its limit (in particular

Table 7.1: Parameters

Parameter	Value	Parameter	Value
$\boldsymbol{M}_m$	$\mathrm{diag}(1.5)$		
$\boldsymbol{K}_p$	$\mathrm{diag}(100)$	$\boldsymbol{K}_d$	$\mathrm{diag}(20)$
$\dot{\tilde{\boldsymbol{x}}}_d$	$\boldsymbol{O}_{6\times1}$	$\ddot{\tilde{\boldsymbol{x}}}_d$	$\boldsymbol{O}_{6\times1}$
α^+	$\boldsymbol{\pi}/2$	α^-	$-\boldsymbol{\pi}/2$
$n^\star$	1.0	$s^\star$	1.0

the fourth joint) is saturated in this region. Thus, the PSM cannot follow the desired trajectory yielding the Cartesian space error shown in Fig. 7.6b. In real procedures, this translates into (i) interrupting the task execution, (ii) releasing the needle and (iii) re-grasping it from a different pose. In Experiment 2, we have chosen $\boldsymbol{p}_i = [-0.08, -0.01, -0.106]$ and $\boldsymbol{p}_e = [0.14, -0.01, -0.106]$ with center at $\boldsymbol{c} = [0.03, -0.01, -0.105]$ to stress the issue caused by singularity during the suturing. Snapshots shown in Figs. 7.5d-7.5e-7.5f illustrates a sequence of PSM pose during this second trajectory. Although the trajectory is very similar to the one in Experiment 1, the fast change of joint configuration (shown in Figs. 7.5d-7.5e-7.5f) occurs due to the proximity to singularity. Singularities are a common concern in manipulation that causes dangerous situations and must be avoided. Singularity occurrence translates into high joint velocities generated for small commanded Cartesian space displacements. Fig. 7.6c shows that joint velocities of the PSM reach a large value during the second trajectory execution.

In Experiment 3, we consider again the suturing trajectory of experiment 1. The optimal grasping pose $\boldsymbol{x}_{s,d}(\boldsymbol{z}^* = [0, -0.138])$ is obtained through the above-explained optimization method (see Sect. 7.4.2) using as initial condition $\boldsymbol{z}_0 = [0, 0]$. This choice is supported by real suturing procedures observations: surgeons often grasp the needle towards its tail (to favor needle insertion), normal to the needle sagittal plane (to minimize collisions between the tool and patient's anatomy). Fig. 7.7 shows the evolution of the cost function $\mathcal{H}$ (given by (7.11)) and $\boldsymbol{z}$ during the optimization routine. The optimization problem can be solved relatively fast with the proposed gradient descent method. A non-optimized version of the code, over 1000 queries, takes on average $\mu_t = 1.0395$ s for each optimization step, with a variance of $\sigma_t^2 = 2.6046 \times 10^{-4}$ s.

The obtained grasping pose is used to generate force cues and inform a human operator during the reach-to-grasp phase. The operator feels the haptic cues, shown in Fig. 7.7c, during the real experiment shown in Fig. 7.2. The force cues intensity decreases by the closeness to the optimal grasping pose. Finally, Figs. 7.8a, 7.8b and 7.8c show the corresponding post-grasp joint positions, Cartesian space errors, and joint velocities that are obtained

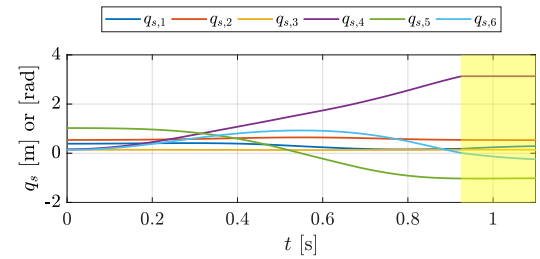
Table 7.2: PSM joint limits ([deg] or [m])

Joint #	1	2	3	4	5	6
q_s^+	60	90	0.25	180	90	90
q_s^-	-60	-90	0.05	-180	-90	-90

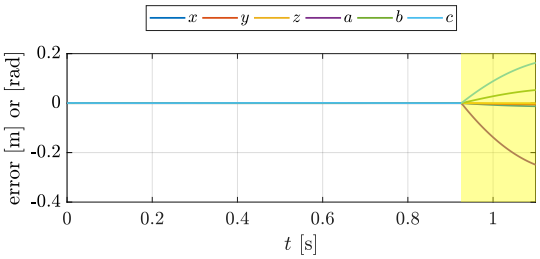
using the optimal grasping conimguration during the suturing tasks execution. These figures demonstrate that the obtained grasping conimguration allows simultaneously avoiding issues encountered in Experiment 1 and 2. Hence, the proposed haptic-guided shared control method helps the operator to avoid joint limits and singularities that translate in significant Cartesian space errors and high joint velocities respectively during the suturing task execution.

7.7 Conclusions

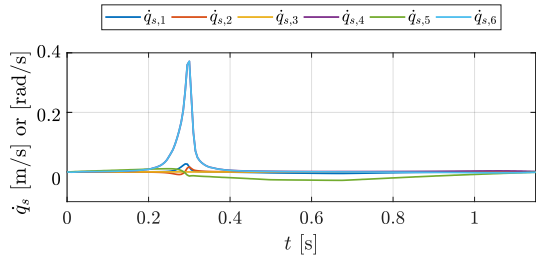
In this work, a novel haptic-guided shared control method for the da Vinci robot which assists the surgeon in grasping the needle in an optimal configuration is devised. The approach helps the surgeon to grasp the needle such that the robot does not face joint limits and singularities during the post-grasping suturing movements. To achieve this, the joint limits and singularities cost is optimized by evaluating its gradient on the needle grasping manifold. This allows computing the optimal grasping pose and, sequentially, force cues (exerted via the MTM) attracting the operator's hand towards it. Although haptic guidance forces inform the surgeon about the optimal needle grasping, the operator ultimately controls the system and decides which grasping pose to choose, thus taking into account other non-modelled objectives. The effectiveness of the proposed approach for optimal needle grasping is illustrated using a real MTM and both simulated and real PSMs. The results show that the proposed haptic-guided shared control system significantly improves the performance during suturing in terms of distance from joint limits and singularities.



(a)

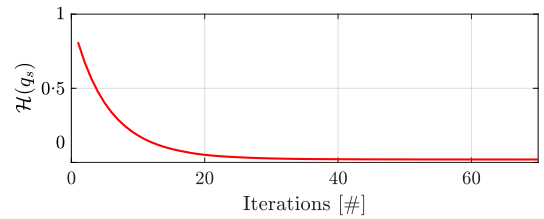


(b)

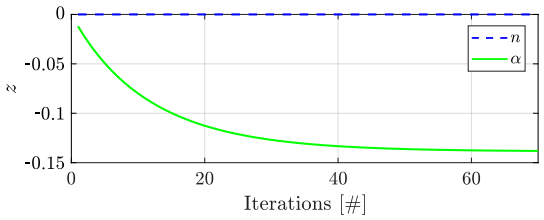


(c)

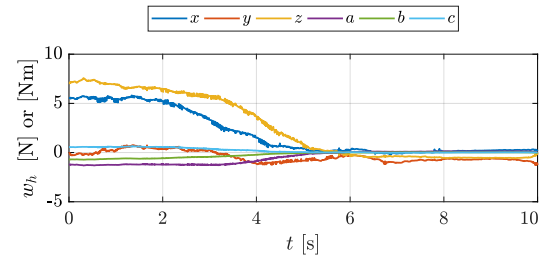
Figure 7.6: *Experiment 1*: (a) shows joint values reaching joint limits and saturated within the yellow shaded area; (b) shows the task space tracking error along the suturing path, *i.e.* $p_d - p$ in Fig. 7.9. *Experiment 2*: (c) shows the manipulator undergoing high joint velocities value during the suturing path. This corresponds to small TOV manipulability at that part of the path.



(a)



(b)



(c)

Figure 7.7: *Experiment 3*: (a) and (b) show the evolution of the cost and parameters values during the execution of the optimization routine; (c) shows haptic guidance force felt by the operator during the haptic-guided shared control grasping experiment. Higher forces are felt in the initial part of the experiment when the operator is far from the optimal grasping pose.

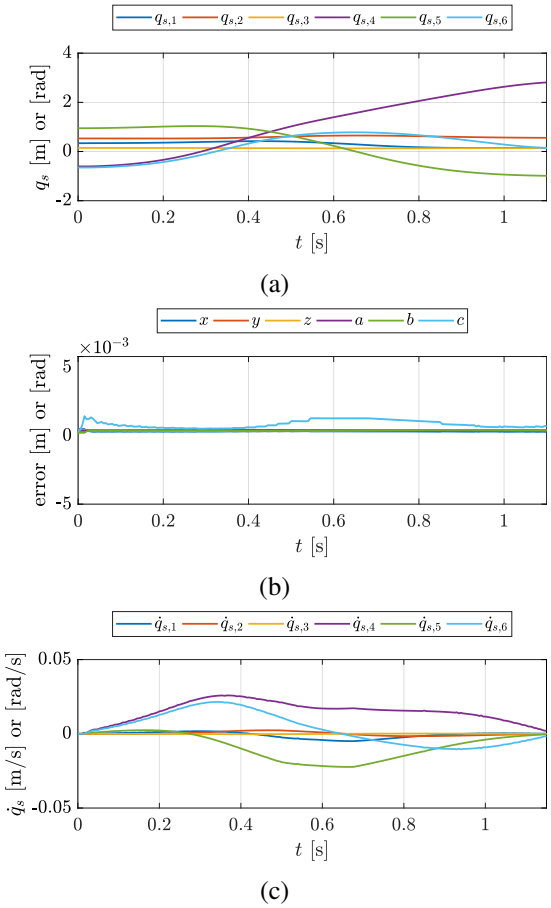


Figure 7.8: *Experiment 3*: (a) shows optimal joint values, far from reaching joint limits; (b) shows the Cartesian space tracking error along the suturing path, *i.e.* $p_d - p$; (c) shows that minimizing the cost of TOV manipulability results in reduced joint velocities (one order of magnitude smaller than those in experiment 2, shown in Fig. 7.6c).

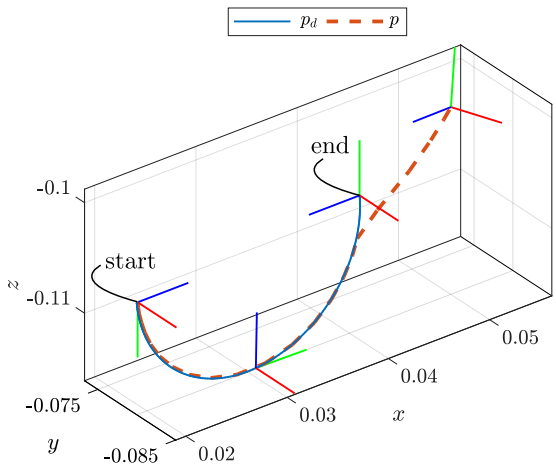


Figure 7.9: Desired path p_d (defined between $p_i = [-0.08, -0.08, -0.106]$ and $p_e = [0.14, -0.08, -0.106]$ with center at $c = [0.03, -0.08, -0.105]$ where p_i, p_e and c are insertion and exit points and the position of the semi-circular path center, respectively) for the needle tip during suturing shown with blue solid line. When the needle can not follow the desired path due to joint limits, this results in actual path p shown in red dashed line.

Chapter 8

Shared Control of a new Surgical Tool for Needle Reorientation

8.1 Introduction

Some of the most critical and delicate tasks in laparoscopic Minimally Invasive Robotic Surgery (MIRS) are reconstructive procedures because of their time demand, the high dexterity required, the risks of causing damage to organs and/or tissues and the risks related to poorly executed sutures. These procedures are very difficult and stressful even for a skilled surgeon, mostly because of the reduced workspace, the high precision required, the lack of haptic perception and the complexity induced by artificial vision feedback.

Common robotic surgical instruments used in MIRS are designed to extend the surgeon dexterity in such delicate procedures. For instance, the da Vinci Surgical System is ideally suited for this scope, thanks to the EndoWrist technology from Intuitive Surgical, that allows the articulated motion of the surgical tool and tremor filtering[1]. The EndoWrist system mimics the human wrist capabilities adding two extra DoFs to the standard laparoscopic instruments, which are mostly rigid and straight tools ending with a gripper. In this way, the surgeon dexterity is comparable to that achieved in open surgery using a standard needle-holder forceps.

[1] http://www.davincisurgery.com/da-vinci-gynecology/\da-vinci-surgery/da-vinci-surgical-system/system-sa-\fety.php

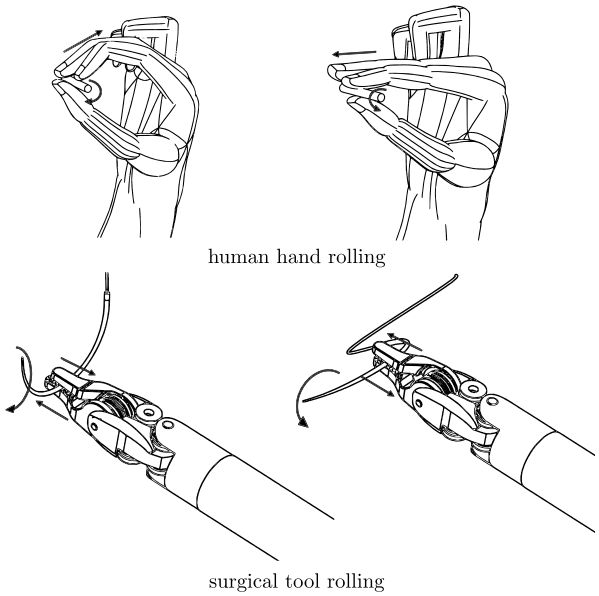


Figure 8.1: Object rolling capability of the novel suturing tool inspired by that of the human hand.

More advanced suturing tools have been developed for laparoscopic interventions to allow automatic suturing executions; this is the case of the Endo360[2] and the PROXISURE[3]. These smart instruments are not yet available for the da Vinci robot and do not give the surgeons the total control of the needle motion neither allow selecting the best needle type and pose for each surgical procedure. Despite the development of enhanced laparoscopic tools [239], surgeons' manipulation capabilities are still far from those of the human hand.

During suturing, the surgeon needs to constantly change the orientation of the needle in order to find the appropriate pose. To this end, the reorientation phase is conducted through successive grasp and release operations, according to a *grasping - release - positioning - re-grasping* sequence, using both arms of the robot. This sequence of operations is shown in the flow chart on the left of Fig. 8.2.

[2] http://www.endoevolution.com/endo360

[3] http://www.ethicon.com/healthcare-professionals/\products/advanced-suturing-system/proxisure-suturing-\devices

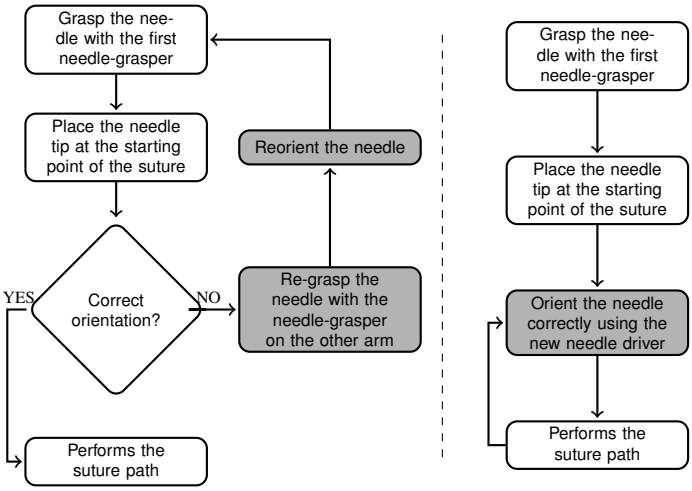


Figure 8.2: Suturing sequence flow charts. Left: standard tool (ST); right: modified tool (MT). The steps involving needle reorientation are highlighted in gray.

This problem may be overcome by modifying the needle driver with a slot specifically designed to keep the needle perpendicular to the gripper [240, 234]. However, even in this case, the needle position and orientation inside the gripper would not be under the surgeon's control. In some cases, also the reaching of the instrument's joint limits might require releasing and re-grasping the needle in a different configuration with the second arm intervention. Haptic cues have been employed to inform the operator about joint limits [176, 100] but dual-arm reorientation cannot be avoided.

Starting from these premises, an additional Degree of Freedom (DoF) can be extremely useful to manipulate rigid objects with a circular cross-section, such as needles for suturing. In this work, a new needle driver tool that allows a more natural and intuitive manipulation of the surgical needle is presented. Inspired by the human hand motion (see Fig. 8.1), the new design enables the possibility to reorient the suturing needle without interruptions, controlling the additional DoF of the tool in telemanipulation or in autonomous mode. Hence, some surgical tasks, such as suturing, might be executed using only one arm. To the best of the authors' knowledge, there are no surgical tools specifically designed for in-hand manipulation.

The rest of this chapter is organized as follows: Section 8.2 presents the

motivation of the work and clinical needs. in Sect. 8.3 the novel tool design solution is detailed. Section 8.4 presents a scale prototype of the tool used to test the mechanical design. Section 8.5 discusses the results obtained in two case studies. Section 8.6 presents a dexterity analysis along with a task-prioritized shared-control method for the proposed tool while Section 8.7 concludes the chapter.

8.2 Motivation

This work takes inspiration from the study presented in [241] in which the most common surgeons' movements performed during open surgery are evaluated. Among these movements, two are of interest for the purpose of the presented work: *"rolling between fingers"* and *"rolling for reorientation"*. These two movements are common both during tissue and needle manipulation and are not performable using the currently adopted robotic surgical instruments. On the other hand, several works can be found in the recent robotic literature focusing about the development of advanced grippers for in-hand manipulation. For instance, in [242, 243, 244] three different design concepts are presented. The proposed approach aims at transferring the most recent results in robotic manipulation to the robotic surgical scenario.

In this work, the focus is in particular on the suturing procedure, defined as a row of stitches holding together the edges of a wound or surgical incision. Suturing is one of the most challenging tasks in minimally invasive surgery and microsurgery [245, 246]; an error in suturing can produce significant tissue damage and is more likely to happen when the needle orientation is not completely under the surgeons' control [234], especially in the absence of force feedback information [247, 219, 248]. Due to the structure of the standard needle drivers, the orientation of the needle during the suturing procedure is not completely controllable and multiple pairs of hand-off movements are required, before the execution of each stitch [234].

An evaluation of the occurrence of this behavior can be found by inspecting replicated suturing procedures. To this end, the suturing video data in the JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS) [233], captured using the dVRK [20] are considered. The database comprises 39 suturing tasks each composed of four stitches on a bench-top model, performed by eight surgeons with different levels of skills. In addition to this, a number of videos acquired during in vivo surgical procedures performed by expert surgeons are also inspected. In the considered videos, the number of stitches that require

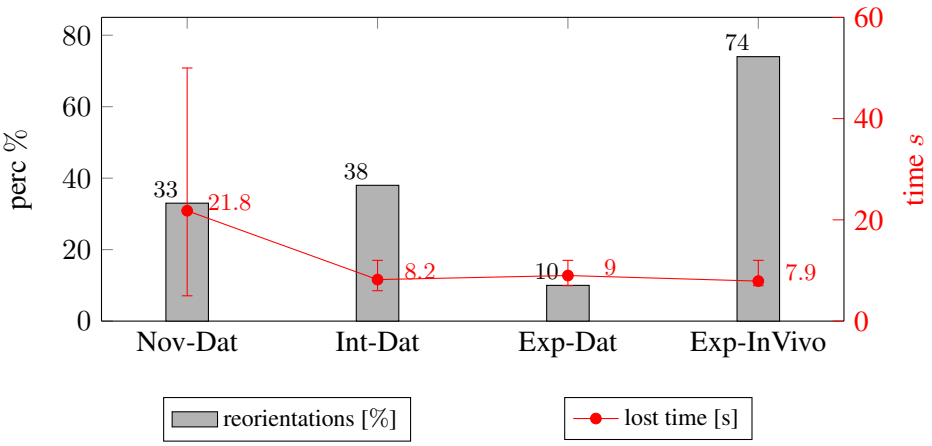


Figure 8.3: Percentage of stitches requiring needle reorientation and average/variance of the time lost for reorienting the needle. Data provided by the JIGSAWS dataset (-Dat) and real procedures (-InVivo) performed by novice (Nov-), intermediate (Int-) and expert (Exp-) surgeons.

needle reorientation are counted and the average and the variance of time lost in this operation are measured, for the three levels of surgical skills. The results in Fig. 8.3 show that the percentage of stitches requiring needle reorientation is rather high, although it decreases when the surgeons' skills increase. The same trend can be observed for the average time lost. It is worth noting that in the real scenarios, in which the needle needs to be dropped and re-grasped to make knots or move organs, the number of stitches requiring reorientation is considerably higher than for the sutures performed on the bench-top model.

8.3 Working Principle

In this section, the working principle of the new suturing tool is described.

8.3.1 Mechanics

In this work, a gripping mechanism capable of impressing tangential motions to a circular cross-section object has been designed and developed. The tendon driven actuation mechanism of a standard da Vinci laparoscopic tool has been modified by adding an additional pulley used to actuate the extra DoF responsible for the rolling motion. This solution is fully compatible with the

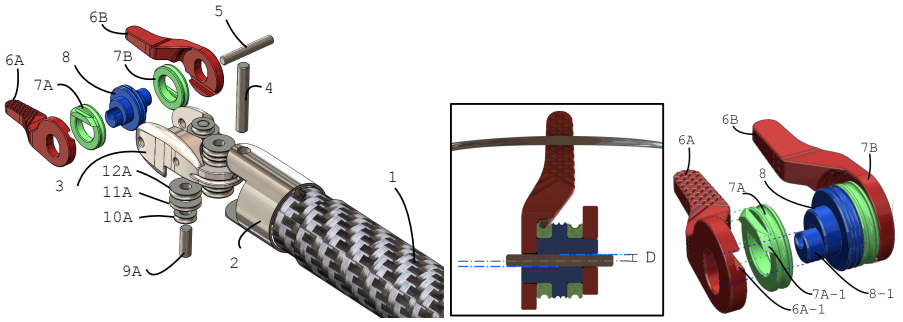


Figure 8.4: The new suturing tool: exploded view and cross section (in the frame).

instruments of the ultimate da Vinci robots, such as the da Vinci Xi, which is equipped with an extra actuated DoF that can be used for advanced tools.

The design of the tool has been carried out considering the following constraints:

- the external radius of the tool must be smaller than the internal radius of the trocar (8.5 mm) used by the da Vinci robotic system;
- the dimensions of the two fingers must be equal to those of the fingers of the da Vinci standard needle driver tool whose efficiency has been largely demonstrated in their long time of use.

Moreover, the maximization of the rolling motion that can be impressed to the most used needles in laparoscopic procedures [249] has been taken in to account.

Figure 8.4 shows an exploded view of the conceived tool. The fingers 6A and 6B are actuated indirectly by the pulleys 7A and 7B to allow the fingers rotation. The pulley 8 creates the desired linear displacement along the fingers major direction. To this end, the pulley 8 has been designed with two eccentric cylinders disposed specularly with a phase displacement of 180° (8-1 in Fig. 8.4). These cylinders rotate within two slots made on the fingers to produce a linear motion. Two linear guides, represented by the parts 7A-1 and 6A-1 in Fig. 8.4, are used to constrain the linear motion between the fingers and the pulleys 7A and 7B. Therefore, a rotation of the eccentric cam causes a shift of the finger with respect to its rest position. Conversely, leaving the cam in the rest position (see Fig. 8.5 on the center) the fingers remain aligned. The

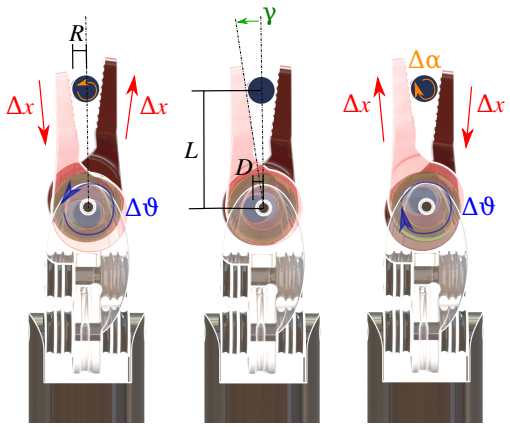


Figure 8.5: New suturing tool working principle. A rotation of the internal pulley of an angle $\Delta\vartheta$ causes the opposite translation of the two fingers (Δx) and hence the rotation of the grasped object of an angle $\Delta\alpha$.

actuation of the pulleys 7A, 7B and 8 is obtained by means of three pairs of tendons (direct and antagonist) fixed to the respective pulleys.

8.3.2 Mathematical Model

The displacement Δx of the two fingers is mechanically related to the central pulley rotation angle $\Delta\vartheta$ (see Fig. 8.5). This displacement causes a certain rotation $\Delta\alpha$ of the cylindrical object depending on the object radius and on its position between the fingers. Assuming no slippage between the object and the fingers, the following equations hold

$$\begin{cases} \Delta x = D\,\sin(\Delta\vartheta) \\ \Delta x = R\,\Delta\alpha \end{cases} \quad \Longrightarrow \quad \Delta\alpha = \frac{D\,\sin(\Delta\vartheta)}{R} \tag{8.1}$$

where R is the object radius and D is the misalignment between the center of the central pulley and the center of the eccentric (refer to Fig. 8.4).

To maximize the rolling capability of the tool, the needle needs to be grasped with its tangent parallel to the tool joint axis (the axis of the pulley 8 in Fig. 8.4). If there is a misalignment the velocity impressed by the instrument is not converted only into a rolling velocity but another component is determined.

In Fig. 8.6 this behavior is shown more in details: consider two planes that translate in two opposite directions with velocities V_T and V_B and a cylinder

in contact with the two planes oriented with an angle β with respect to the direction of motion. The velocity V_B can be decomposed into two components V_B^t, V_B^n, where

$$V_B^t = V_B \cos(\beta) \qquad V_B^n = V_B \sin(\beta) \tag{8.2}$$

and the same equations can be written for the V_T velocity. Then, (8.1) must be rewritten taking into account that the rolling motion when $\beta \neq 0$ is reduced by $\cos(\beta)$

$$\Delta\alpha = \frac{D \sin(\Delta\vartheta)}{R} \cos(\beta). \tag{8.3}$$

Therefore, if the angle β is not equal to zero, both the normal and tangential components of the velocities V_B and V_T are different from zero. Notice that the normal velocities V_B^n and V_T^n are realizable only considering slippage in the direction of the object axis. Hence, if $\beta \neq 0$, during the rolling motion a higher velocity is required to rotate the needle by the same quantity, resulting in a more expensive operation. This means that, if the needle is not gripped with its tangent orthogonal to the direction of motion of the two fingers, the needle can be still rotated but a lower angular displacement can be accomplished.

To comply with specifications of Sect. 8.3.1, a constraint on the misalignment D has been imposed. In particular, we choose $D = 0.5\,\text{mm}$ while $\Delta\vartheta$ is in the range $[-\pi/2,\ \pi/2]$. The rolling angle ranges for four classes of needle, most used in laparoscopic surgery, with three different β values can be computed using (8.3) are given in Tab. 8.1.

Finally, the aperture angle of the gripper γ is related to the object radius and to the distance of the object from the gripper center of rotation (R and L

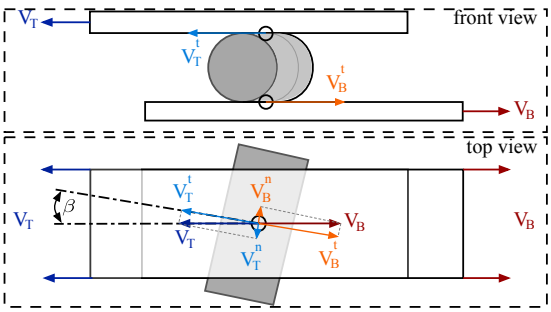


Figure 8.6: Rolling model representation.

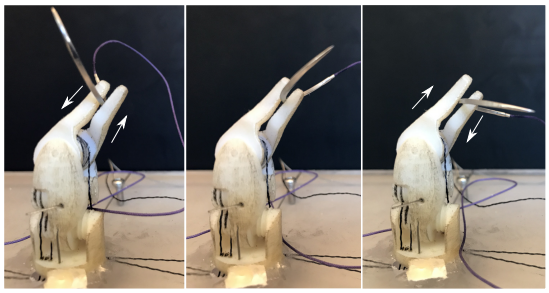


Figure 8.7: 3D printed prototype of the new tool (scale 2:1): evaluation of the working principle with a circular shape needle.

in Fig. 8.5) by the equation

$$R = L\,\sin(\gamma) \quad \Longrightarrow \quad \gamma = \arcsin\left(\frac{R}{L}\right). \tag{8.4}$$

8.4 Scale Prototype

Due to the difficulty of having a complete and fully functional metal tool attached to a surgical robot, the evaluation of the mechanism has been carried out through a plastic 2:1 scale prototype. The prototype was realized using a 3D printed technology based on the polyjet process[4]. This technology allows achieving sufficiently high precision and accuracy.

Figure 8.7 shows a sequence of the motion. The mechanism was manually actuated by tendons and the executed rotation angle was measured using a protractor. The experiments were performed on a needle with a circular cross-section with diameter doubled with respect to the SH-Plus needle. When the

[4] http://www.stratasys.com/3d-printers/design-\series/objet24

Table 8.1: Maximum rolling angles for different needles

	R [mm]	$\beta = 0$	$\beta = \pi/12$	$\beta = \pi/4$
RB-1	0.25	$\pm$ 114°	$\pm$ 110.1°	$\pm$ 80.3°
SH-Plus	0.352	$\pm$ 88°	$\pm$ 85°	$\pm$ 62°
GL-222	0.38	$\pm$ 81°	$\pm$ 78.2°	$\pm$ 57°
UR-6	0.5	$\pm$ 57.5°	$\pm$ 55.3°	$\pm$ 40.5°

tendons were actuated to move the mechanism in the entire range of motion, we measured an angle of about $80°$. This value is close to the predicted value of $88°$ for a SH-plus needle actuated using a non-scaled tool. Hence, despite the high friction of the employed material and the errors introduced by a non perfect needle alignment, the experiments showed a working behaviour of the prototype close to the expected one.

8.5 Case Studies

To measure the performance of the new needle driver a set of real suturing procedures executed on the dVRK with a standard tool (ST) is considered and replicated in a simulated environment with our modified tool (MT). Two case studies have been considered: the first is focusing on a *single stitch* trajectory execution; in the second, *complete sutures* procedures are considered, all with 5 stitches performed on different phantoms.

The goal of the analysis is to prove that the MT can both improve the surgeon precision and reduce the execution time by allowing in-hand needle reorientation during real suturing procedures. Moreover, it is shown that tool is able to overcome the problem of reaching joint limits, within the range of movement.

Given these good results obtained on the scale prototype, in the simulation it is assumed that the MT is able to rotate the needle as expected, without slipping.

8.5.1 Simulation Environment

The simulation environment is composed by a simulated PSM of the dVRK that can be equipped with ST and MT. The simulator has been developed in V-REP and interfaced with the MTM of the dVRK. Simulated trajectories have been planned with MATLAB.

PSM Arm

The PSM arm is a 7-DoF actuated arm, which moves the attached instrument around a Remote Center of Motion (RCM), *i.e.* a mechanically-fixed point that is invariant with respect to the configuration of the PSM joints. In detail, the first 3 DoFs correspond to Revolute (R) and Prismatic (P) joints, in an RRP sequence and allows the rotation and translation of the surgical tools around

the RCM. Moreover, the last 3 DoFs, in an RRR sequence, constitute the instrument wrist. Finally, the last DoF (seventh) allows opening/closing of the grippers jaw. For more details on the robot kinematics the reader can refer to [21].

Tools

Two different tools moved by the first 3 PSM DoFs are used in the case studies: the first is the ST whose kinematic model is that described in [21]; the second is the MT whose kinematic is described by the DH parameters in Tab. 8.2. In detail, the MT is a 4-DoF kinematic chain in which the first 3 DoFs are the three joints of the ST wrist. The fourth DoF corresponds to a joint placed in the center of the circular section of the needle at the grasping point and is used to model the rotation induced by our mechanism. Notice that, in the experiments, is assumed that the object is already grasped and then the opening/closing DoF is not explicitly taken into account.

8.5.2 First Case Study: Single Stitch

Due to its kinematic structure, the PSM arm of the DVRK can easily reach configurations that are near to joint limits. When this happens, the real trajectory of the robot PSM may deviate from the trajectory commanded by the surgeon through the master robot, because of the occurrence of joint saturations. The goal of this case study is that of showing how our instrument could help to overcome this problem. To this end, an experiment was carried out using the ST mounted on the PSM arm: a needle trajectory along a circular path was commanded through the master robot, starting from a configuration close to a joint limit, so that joint saturations occur. Then, the desired circular trajectory was extrapolated using the part of the real trajectory not influenced by joint saturations. This desired trajectory was used in simulation as reference for the model of the PSM arm with the MT.

Table 8.2: DH parameters of the modified tool (MT)

link	joint	a_i [m]	α_i [rad]	d_i [m]	θ_i [rad]
1	R	0	$\pi/2$	–	θ_4
2	R	0.009	$-\pi/2$	–	$\theta_5 + \pi/2$
3	R	L	0	–	θ_6
4	R	0	$-\pi/2$	–	$\theta_7 - \pi/2$

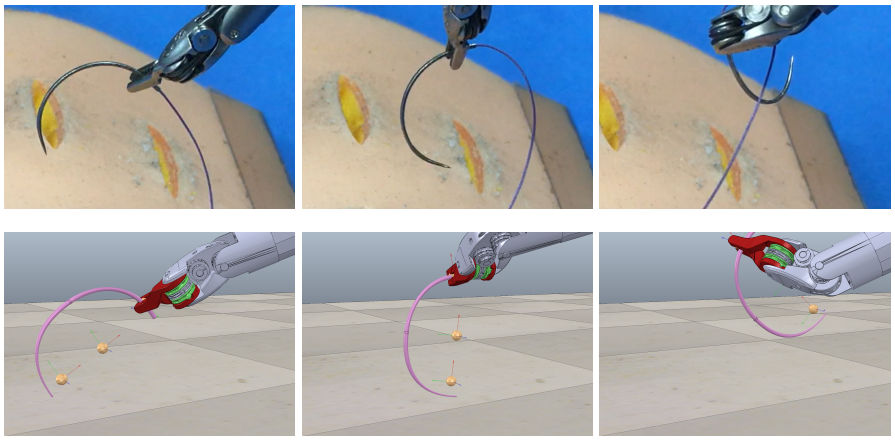


Figure 8.8: Snapshots sequence of a single stitch trajectory. Top: standard tool (ST); Bottom: modified tool (MT).

The joint limits considered in the experiments are given in Tab. 8.3. A standard inverse kinematics algorithm has been implemented to solve for the da Vinci PSM joint values given the desired position and orientation of the needle frame [11]. Figure 8.8 contains some snapshots from the real performed trajectory and the V-Rep simulated environment.

The different behaviour of the two instruments can be better understood in Fig. 8.9 showing the needle tip path executed using the ST (pe_s) and our MT (pe_m), respectively. When the ST is used, due to the presence of the joint limits, the desired path cannot be successfully executed, as shown in Fig. 8.9 (dot dashed blue line). It can be observed that only in the first part of the commanded trajectory, before the joint limits occurrence, the path of the needle's tip has a circular shape; when the joint limits occur, the corresponding joint variables saturate and the actual path significantly deviates from the desired circular path. The desired circular trajectory, extrapolated using the part of the real trajectory not influenced by joint saturations, is depicted using dotted black line. This trajectory was commanded in simulation to the PSM arm with

Table 8.3: PSM joint limits ([rad] or [m])

	q_1 [rad]	q_2 [rad]	q_3 [m]	q_4 [rad]	q_5 [rad]	q_6 [rad]	q_7 [rad]
Std.	$\pm\pi/2$	$\pm\pi/3$	$[0, 1]$	$\pm 3/2\pi$	± 1.39	± 1.39	–
Mod.	$\pm\pi/2$	$\pm\pi/3$	$[0, 1]$	$\pm 3/2\pi$	± 1.39	± 1.39	± 1.4

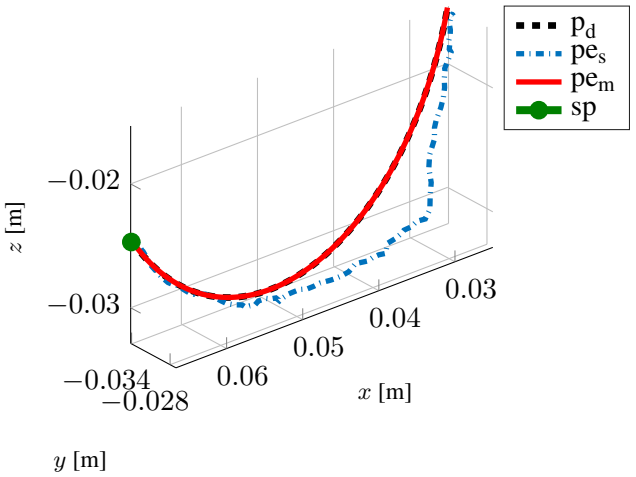


Figure 8.9: Needle tip paths. pe_s is the path performed using the ST, p_d is the desired circular path and pe_m is the path simulated using the MT. The point sp is the trajectory starting point.

our MT. Since now the system is redundant, infinite solutions exist for the inverse kinematics problem and the redundancy could be used to efficiently avoid joint limits as shown later. Here, redundancy is not explicitly exploited; instead, the goal is just to show that a feasible solution could be easily found by a surgeon teleoperating the instrument. Therefore a simple solution corresponding to the minimum norm velocity is computed by using the Jacobian pseudo-inverse; despite this, the joint limits are not violated and the desired path for the needle is respected as shown in Fig. 8.9 (red continuous line).

The quantitative results of the experiment are shown in Fig. 8.10. In particular, the figure shows the time histories of the joint variables $\boldsymbol{q}$ using both the ST and the MT where the dashed lines represent the time history of the seventh joint, active only in the MT, and the dot-and-dash lines represent the time history of the sixth joint, which reaches its upper limit in the ST. It is clear that, with our MT, the redundancy allows avoiding joint limits. This is quantitatively explained by considering the weighted dexterity index, better described in Appendix, which allows evaluating the manipulator's overall ability to move, by tacking into account the distance from both joint limits and singularities. This index has been chosen because it allows comparing the dexterity of manipulators with the same task space dimension independently from the joint space dimension, thus it constitutes a suitable measure of the introduced

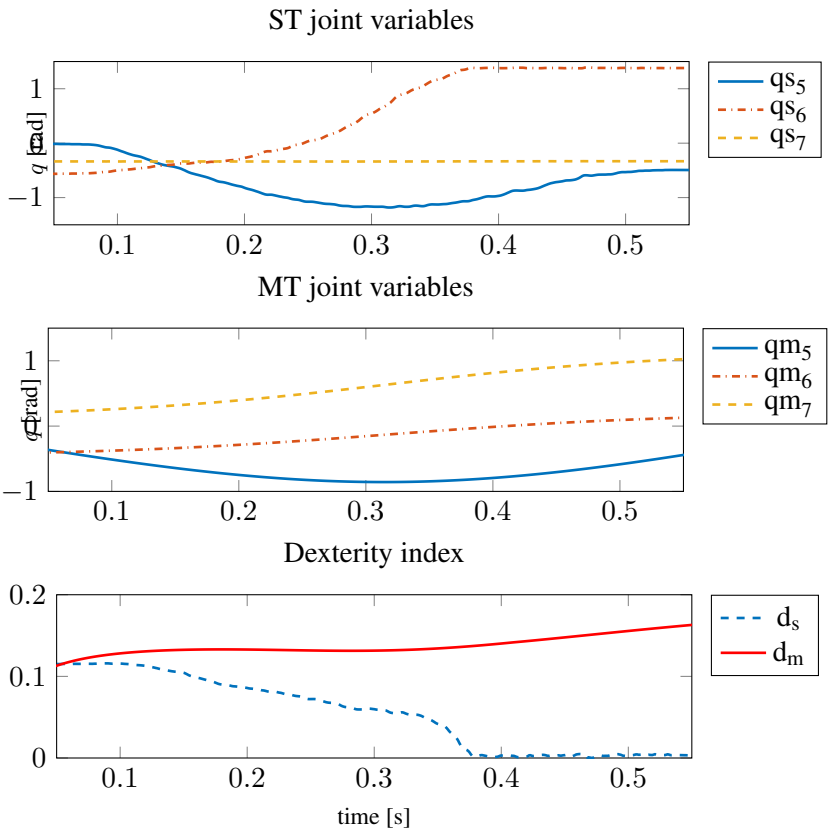


Figure 8.10: Numerical results for the single stitch trajectory. Top: time history of the joint variables for ST; Middle: time history of the joint variables for MT. Bottom: time history of the dexterity index with ST (d_s) and MT (d_m).

enhancements. As it can be seen from Fig. 8.10, this measure remains greater than zero for the whole trajectory execution only when the MT is used.

8.5.3 Second Case Study: Complete Suturing Procedures

In the second case study, the set of suturing procedures reported in Tab. 8.4 was recorded. The sutures have been executed by novice (N), intermediate (I) and expert (E) surgeons, using three types of needles. The procedures consist in 10 planar sutures with different wound angles, executed on two different types of phantoms, and 3 circular anastomosis on a vessel phantom. Each suture

consists of 5 stitches. In the table, for each suture, it is also indicated if a joint limit was reached at least once.

The 13 suturing procedures have been monitored and, for each sequence of 5 stitches, the following data have been evaluated and reported in Fig. 8.11 (Top):

- the number of stitches that required needle reorientation (gray bars);
- the number of reorientations performed along the needle tangent, for which the MT would have been helpful (red bars);
- the number of reorientations with rotation angles that were lower than the maximum rolling angles for the used needle (considering the values reported in Tab. 8.1), for which the MT would have been successful (blue bars).

The overall performance is summarized in Ta. 8.5. It can be seen that 69% of the stitches required reorientation and that the MT would be helpful in the 66% of the situations, allowing to complete the reorientation in the 55% of the cases.

To get a better understanding of the problems encountered during the suture performance we asked the surgeons to explain the reasons of each reorientation. In the 37.1% of the cases we found that the needle was dropped to move organs or to make knots and wasn't in the optimal orientation; in the 48.6% of the cases the needle was gripped in a bad orientation; finally, in the 14.3% of the cases, the needle lost the correct orientation during the stitch execution and needed to be reoriented.

Furthermore, for each suturing procedure, the total time lost in reorienting the needle was recorded. The results are reported in Fig. 8.11 (bottom, red line). This time can be compared to the predicted reorientation time using the MT, reported in the same plot (green line). This latter has been computed as follows:

- for the reorientations that can be performed with the MT (i.e., those classified as successful and represented by the blue bars in Fig. 8.11), the execution time have been estimated considering a velocity of about 45deg/sec for the rolling degree of freedom of the tool;
- for all the other reorientations the same time measured during he execution of the real stitch with a standard tool has been considered.

The results show that the MT allows a significant reduction of the time spent for needle reorientation.

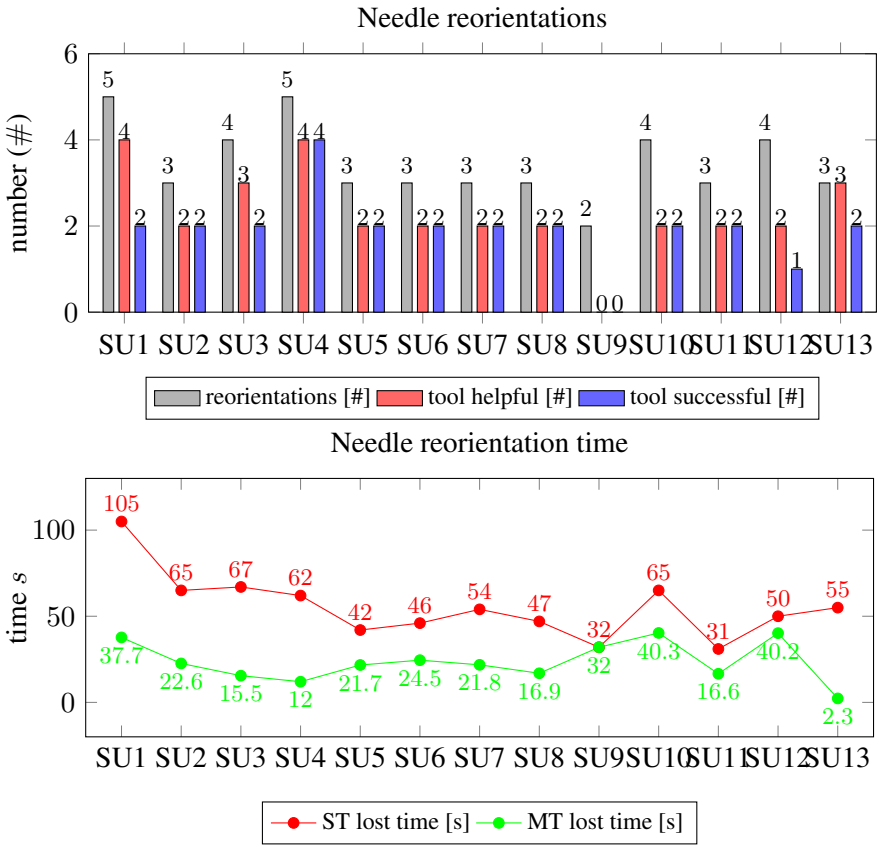


Figure 8.11: Collected data in 13 sutures (SU1, . . . , SU13) of 5 stitches. Top: Number of stitches requiring needle reorientation for each suture (gray bar), number of reorientations for which the MT would have been helpful (red bar), number of reorientations for which the MT would have been successful (blue bar). Bottom: Total time lost for reorienting the needle with the ST (red line) and with the MT (green line).

8.6 Dexterity-based Shared Control Strategy

In order to quantitatively evaluate the benefits introduced by the proposed tool, a dexterity analysis has been performed along suturing trajectories [250]. A suitable dexterity measure has been used to quantify the overall motion capability of the robot with the new tool in comparison with the same robot using a standard da Vinci tool. This measure can be computed directly starting from the robot Jacobian matrix.

The influence of the joint limits on the robot's dexterity can be taken into account by weighting the entries of the Jacobian matrix according to a joint limits performance criterion [251, 252]. More in detail, a penalization matrix $L(\boldsymbol{q}) \in \mathbb{R}^{r\times n}$ is introduced, where r and n are the task and the joint space dimensions, respectively. This matrix is used to compute the elements of the augmented Jacobian $\tilde{J}(\boldsymbol{q}) \in \mathbb{R}^{r\times n}$ as

$$\tilde{J}_{i,j}(\boldsymbol{q}) = L_{i,j}(\boldsymbol{q})J_{i,j}(\boldsymbol{q}), \quad i = 1,\dots r, \quad i = 1,\dots n, \tag{8.5}$$

where $J_{i,j}(\boldsymbol{q})$ is the (i,j) element of the robot Jacobian and $L_{i,j}(\boldsymbol{q})$ is defined as

$$L_{i,j}(\boldsymbol{q}) = \frac{1}{\sqrt{1+|\nabla h_j(\boldsymbol{q})|}}. \tag{8.6}$$

The scalar function $h\left(\boldsymbol{q}\right) : \mathbb{R}^n \rightarrow \mathbb{R}$ in (8.6) is a differentiable function of the joint vector $\boldsymbol{q}$ which tends to infinity as the joint variables approach the corresponding joint limits. In this work, taking inspiration from [175], we use the following function

$$h\left(\boldsymbol{q}\right) = \sum_{i=1}^{n} \frac{1}{4} \frac{\left(q_{i,max} - q_{i,min}\right)^2}{\left(q_{i,max} - q_i\right)\left(q_i - q_{i,min}\right)}. \tag{8.7}$$

The gradient $\nabla h\left(\boldsymbol{q}\right)$ represents the direction of fastest increase of $h\left(\boldsymbol{q}\right)$ and is useful to build the corresponding penalization index (8.6). The $i-$th component of the gradient $\nabla h_i(\boldsymbol{q}) = \partial h\left(\boldsymbol{q}\right)/\partial q_i$ can be computed as

$$\nabla h_i(\boldsymbol{q}) = \frac{1}{4} \frac{\left(q_{i,max} - q_{i,min}\right)^2 \left(2q_i - q_{i,max} - q_{i,min}\right)}{\left(q_{i,max} - q_i\right)^2 \left(q_i - q_{i,min}\right)^2}. \tag{8.8}$$

At this point, according to [253], a weighted dexterity measure d can be computed from the augmented Jacobian as

$$d = \frac{\sqrt{rn}}{\sqrt{\mathrm{tr}\left[\left(\tilde{J}\tilde{J}^{\mathrm{T}}\right)^{-1}\right]}}, \tag{8.9}$$

where tr$(\cdot)$ denote the trace operator. This index provides similar information of the standard manipulability index, but allows comparing manipulators with the same task space dimension independently from the joint space dimension. Moreover, it takes into account the distance from both joint limits and singularities.

The modified laparocopic instruments can be modelled as a 7-DoF manipulator whose end-effector is the needle tip Σ_t (see Fig. 8.12). Assuming that the manipulator is velocity controlled, its motion can be described by the equation $\dot{\boldsymbol{q}}_s = \boldsymbol{v}$, where $\boldsymbol{q}_s \in \mathbb{R}^7$ denote the vector of the manipulator generalized coordinates and $\boldsymbol{v}$ is the control input. Employing the standard inverse Jacobian map the joint control input $\boldsymbol{v} \in \mathbb{R}^7$ can be written as [183]

$$\boldsymbol{v} = \underbrace{\boldsymbol{J}_s^{\dagger}\boldsymbol{u}_s}_{\text{primary task}} + \underbrace{(\boldsymbol{I} - \boldsymbol{J}_s^{\dagger}\boldsymbol{J}_s)\dot{\boldsymbol{q}}_{s,0}}_{\text{secondary task}}, \tag{8.10}$$

where $\boldsymbol{J}_s \in \mathbb{R}^{6\times 7}$ is the manipulator task Jacobian, $\boldsymbol{J}_s^{\dagger}$ denotes its Moore-Penrose pseudoinverse. The control input includes a Cartesian space velocity vector $\boldsymbol{u}_s \in \mathbb{R}^6$ representing the primary task, and an additional velocity $\dot{\boldsymbol{q}}_{s,0} \in \mathbb{R}^7$, projected into the null space of the primary task.

A shared control strategy is proposed which consists in a human operator action that steers the manipulator along its 6-D Cartesian space and an autonomous regulation of the manipulator redundancy for joint limits avoidance.

This is realized by setting $\boldsymbol{u}_s = \boldsymbol{u}_m \in \mathbb{R}^6$, where $\boldsymbol{u}_m$ is the Cartesian velocity of the master input device, while $\dot{\boldsymbol{q}}_{s,0}$ is the autonomous control action that maximizes the distance from joint limits.

To compute $\dot{\boldsymbol{q}}_{s,0}$ the scalar cost function $h\left(\boldsymbol{q}_s\right)$ in (8.7) is considered. The gradient $\nabla h\left(\boldsymbol{q}_s\right)$ in (8.8) is used to compute the autonomous control contribution for joint limits avoidance according to

$$\dot{\boldsymbol{q}}_{s,0} = -\nabla h\left(\boldsymbol{q}_s\right). \tag{8.11}$$

This term is used to implement the secondary task in (8.10). It is worth to note that the case $\dot{\boldsymbol{q}}_{s,0} = \boldsymbol{0}$ corresponds to the least-norm joint velocity solution.

Simulated experiment are performed using the V-REP simulator. The simulated setup recreates the real dVRK slave side. A pre-recorded single stitch trajectory is used and $\boldsymbol{u}_m(t)$ is computed from this. The task is executed using both the 6-DoF and 7-DoF tools. In the latter case both the least-norm (i.e., $\dot{\boldsymbol{q}}_{s,0} = 0$) and the null-space solutions are considered. Results of the proposed experiments are given in Figs. 8.13, 8.14 and 8.15. As it can be noticed in the 6-DoF case the trajectory is not successfully completed due to the

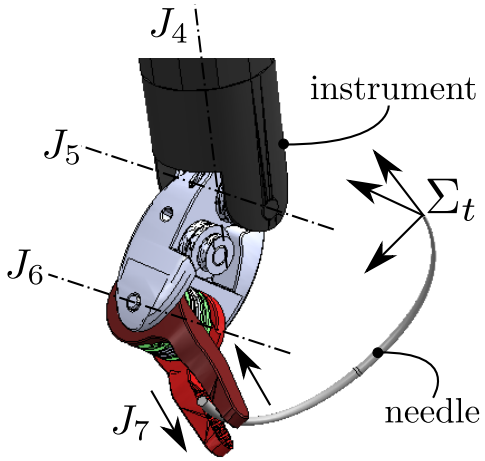


Figure 8.12: Representation of the modified dVRK instrument. $J_4, \ldots, J_7$ represent the joints 4 to 7 axes of the instrument.

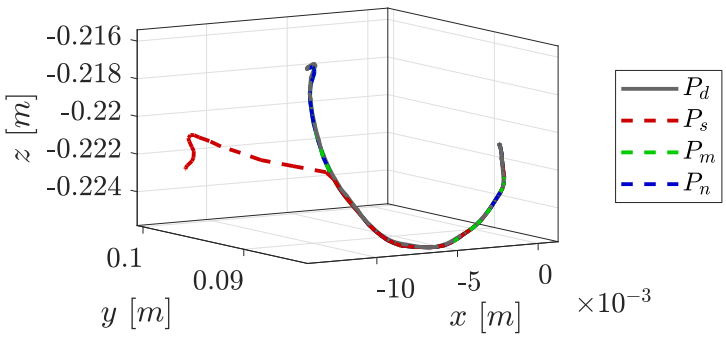


Figure 8.13: Desired needle tip path (P_d) versus standard tool (P_s), modified tool least-norm (P_m) and null-space (P_n) paths.

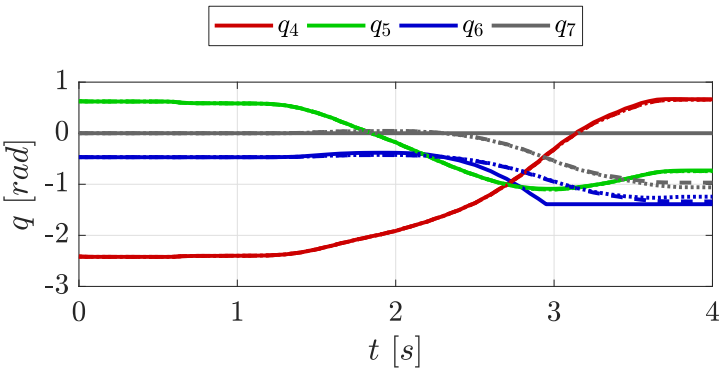


Figure 8.14: Time history of joint angles during the proposed experiment. Solid lines represent the 6-DoF case, dashed and dotted lines represent the 7-DoF case least-norm and null-space solutions, respectively.

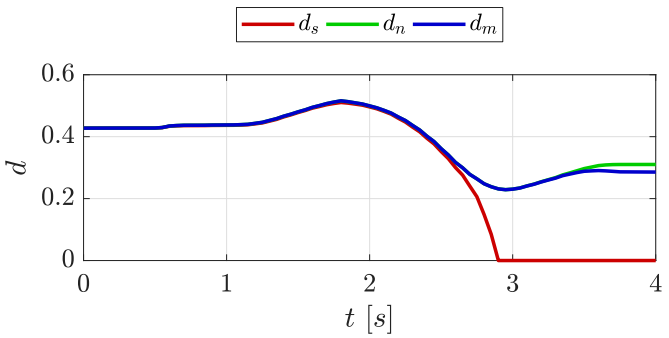


Figure 8.15: Dexterity measure along the executed trajectory. d_s represents 6-DoF case, d_m and d_n the 7-DoF least-norm and null-space solutions, respectively.

occurrence of joint limit on the sixth joint (Figs. 8.13 and 8.14). This is quantitatively explained by the dexterity measure which approaches zero accordingly (Fig. 8.15).

In the 7-DoF case both the least-norm and the null-space solutions are efficient in terms of joint limits avoidance (Figs. 8.13 and 8.14). However, the null space solution demonstrates slightly improved dexterity towards the end of the experiment (Fig. 8.15) .

It is worth noting that, although simulated results are encouraging, the 7-DoF case requires accurate and robust sensing technologies for needle-grasper relative pose estimation in the real scenario.

8.7 Conclusions and Future Works

In this work, a new concept of robotic surgical tool designed for in-hand manipulation of a suturing needle is shown. The mechanical design has been validated through a 2:1 scale prototype. Moreover, a comparison between real suturing procedures, using a standard tool, and simulated procedures, using the proposed modified tool, has been presented. The results show that the robot's dexterity improves when the modified tool is used. Moreover, the in-hand needle reorientation capability of the modified tool allows a significant reduction of the execution time of complete suturing procedures. A shared control architecture has been proposed to exploit the redundancy introduced by the novel tool in order to cope with the joints limits and to maximize the dexterity. In future works, a real prototype of the proposed tool with standard dimensions will be realized using materials suitable for sterilization and for real surgical interventions. Moreover, the tool functionalities will be evaluated in both ex vivo and in vivo procedures performed by a statistically significant population of both novice and expert surgeons. For this purpose, a device integrated in the dVRK robot master device will be designed to allow precise and real-time control of the new advanced tool. Finally, a study will be done to evaluate the best needle surface and material to maximize gripping and rolling capabilities.

Table 8.4: Suture procedures

	expertise	type	needle	joint limit
SU1	novice	planar vertical	GL-222	NO
SU2	novice	planar 15°	GL-222	YES
SU3	novice	planar 20°	RB-1	YES
SU4	novice	planar 110°	RB-1	NO
SU5	novice	planar 20°	RB-1	YES
SU6	intermediate	planar vertical	UR-6	NO
SU7	intermediate	planar vertical	UR-6	NO
SU8	intermediate	planar vertical	UR-6	NO
SU9	intermediate	planar vertical	UR-6	NO
SU10	intermediate	planar vertical	UR-6	NO
SU11	expert	circular vessel	UR-6	NO
SU12	expert	circular vessel	UR-6	NO
SU13	expert	circular vessel	UR-6	YES

Table 8.5: Overall performance of the MT

	stitches	reorient	help	success
#	65	45	30	25
		reorient/stitches	help/reorient	success/reorient
%		69%	66%	55%

Chapter 9

Conclusion and Discussion

This thesis addressed the problem of designing effective and safe telerobotic systems employing the haptic-based shared control paradigm and passivity-based control. As contemporary master-slave robotic interfaces function as a sophisticated, yet passive, extension of the user's hands, sensor-based control methods and algorithms demonstrated the potential to incorporate further intelligence and shared knowledge in the telerobotic system. After a review of the state-of-the-art methods given in Chapter 3, with a particular focus on telerobotics, haptics, and passivity-based control and related applications, contributions of the thesis were presented.

In Part I, two sensor-based haptic shared-control methods for safe telemanipulation in industrial scenarios were proposed.

A shared-control architecture comprising several features for a dual-arm robotic system was described in Chapter 4, while a task-prioritized shared-control architecture for redundant robots was proposed in Chapter 5. These two telerobotic architectures consider the overall system partially controlled by a human operator through a haptic device, while some degrees of freedom are autonomously regulated using vision-based and collision avoidance control. Task priority is used to implement the division of roles. While remotely controlling the system, the user receives opportunely designed haptic cues that helps in maintaining her/him motions compatible with the system constraints (collisions, joint limits and singularities). The control architecture design was explained in details, passivity was analyzed, modeling of the system, experiments, and human subject study were reported to show the performance improvements during remote grasping tasks.

However, several limiting aspects still exist for the proposed shared con-

trol architectures. One of the most prominent aspects is the design of the cost functions and the tuning of their parameters, which are still chosen heuristically. Since haptic cues are computed from the gradient of the devised cost, its design is supposed to play a relevant role in the overall user's experience with the telerobotic system. Moreover, the presented shared-control approaches are local, in the sense that the operator can only provide current velocity commands, and receive instantaneous force cues. The ability to 'steer' a whole future trajectory in the task space, and to receive predictive force feedback correspondingly, could significantly enhance the operator's performance, especially when dealing with complex tasks. In addition, the fixed behavior of the autonomous control action, opportunely designed to help the operator in the specific task at hand, may limit the flexibility of the system in performing different tasks. This is a crucial aspect to address since, in this case, switching among different shared control strategies may generate counter-intuitive and unsafe situations for the operator. Future directions may account for these aspects, possibly also addressing the problem of suitably balancing the amount of operator/autonomy control level.

In Part II of the thesis, three other contributions were presented.

An interactive virtual fixtures generation and adaptation method was proposed in Chapter 6 along with an energy-tanks passivity-based control method to keep the system safe, while a shared-control approach suitable for assisted needle grasping was proposed in Chapter 7. Chapter 8 proposed a novel laparoscopic tool that allows in-hand needle re-orientation along with a suitable shared-control strategy. Besides the application domain, these methods share the underlying idea of providing to the surgeon haptic cues that help her/him in effectively carry out the considered task. In this case, haptic guidance informs the user while leaving her/him the ultimate control of the system to deal with unforeseen situations thus keeping the system safe.

However, even though all the proposed methods showed promising results, yet not sufficient proof of acceptance is demonstrated in the surgical robotics domain. As future work, addressing the problem of human factors in real surgical environments is paramount to assess the applicability and the robustness of the proposed methods. Moreover, customized/tailored levels of assistance would probably be a desirable system feature. A system could, for example, exhibit stiffer guidance (*i.e.*, less freedom for the operator) when it is operated by novices, while it could implement softer guidance (*i.e.*, more freedom for the operator) when it is operated by experts.

All the presented methods enable humans to effectively and safely realize

complex tasks by physically or remotely interacting with semi-autonomous robotic systems capable of providing assistance. In the near future, the availability of datasets and the advancement of machine learning techniques will enable enhanced flexibility of shared control systems that will be capable of providing contextual or personalized assistance and seamless adaption of the autonomy level. However, this desirable trend will raise new challenges for safety and stability certification of shared control robotic systems, thus requiring new advanced control methods to implement the continuously evolving division of roles.

Bibliography

[1] M. Sagardia, K. Hertkorn, T. Hulin, S. Schtzle, R. Wolff, J. Hummel, J. Dodiya, and A. Gerndt, "Vr-oos: The dlr's virtual reality simulator for telerobotic on-orbit servicing with haptic feedback," in *2015 IEEE Aerospace Conference*, pp. 1–17, March 2015.

[2] F. Abi-Farraj, N. Pedemonte, and P. R. Giordano, "A visual-based shared control architecture for remote telemanipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4266–4273, 2016.

[3] F. Nageotte, P. Zanne, C. Doignon, and M. de Mathelin, "Stitching planning in laparoscopic surgery: Towards robot-assisted suturing," *The International Journal of Robotics Research*, vol. 28, no. 10, pp. 1303–1321, 2009.

[4] 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.

[5] G. A. Fontanelli, M. Selvaggio, M. Ferro, F. Ficuciello, M. Venditelli, and B. Siciliano, "A V-REP simulator for the da vinci research kit robotic platform," in *7th IEEE International Conference on Biomedical Robotics and Biomechatronics*, pp. 1056–1061, Aug 2018.

[6] R. C. Goertz, "Mechanical master-slave manipulator," 1954.

[7] R. C. Goertz and F. Bevilacqua, "A force-reflecting positional servomechanism," 1952.

[8] T. B. Sheridan and W. R. Ferrell, "Remote manipulative control with transmission delay," *IEEE Transactions on Human Factors in Electronics*, vol. HFE-4, pp. 25–29, Sep. 1963.

[9] T. B. Sheridan, "Space teleoperation through time delay: review and prognosis," *IEEE Transactions on Robotics and Automation*, vol. 9, pp. 592–606, Oct 1993.

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

[11] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. 2010.

[12] B. Hannaford and A. M. Okamura, *Haptics*, pp. 1063–1084. Cham: Springer International Publishing, 2016.

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

[14] S. A. Bowyer, B. L. Davies, and F. Rodriguez y Baena, "Active constraints/virtual fixtures: A survey," *IEEE Transactions on Robotics*, vol. 30, pp. 138–157, Feb 2014.

[15] J. Abbott, P. Marayong, and A. Okamura, "Haptic virtual fixtures for robot-assisted manipulation," *Robotics Research*, pp. 49–64, 2007.

[16] A. van der Schaft, *L2-Gain and Passivity Techniques in Nonlinear Control*. Springer Publishing Company, Incorporated, 3rd ed., 2016.

[17] V. Duindam and S. Stramigioli, "Port-based asymptotic curve tracking for mechanical systems," *European journal of control*, vol. 10, pp. 411–420, 1 2004.

[18] G. Antonelli, "Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems," *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 985–994, 2009.

[19] W. S. Newman and Y. Zhang, "Stable interaction control and coulomb friction compensation using natural admittance control," *Journal of Robotic Systems*, vol. 11, no. 1, pp. 3–11, 1994.

[20] 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," in *IEEE International Conference on Robotics and Automation*, pp. 6434–6439, May 2014.

[21] G. A. Fontanelli, F. Ficuciello, L. Villani, and B. Siciliano, "Modelling and identification of the da vinci research kit robotic arms," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1464–1469, 2017.

[22] "da Vinci research kit (DVRK) wiki." http://research.intusurg.com/dvrkwiki. [Online; accessed 7-January-2020].

[23] G. Guthart and J. Salisbury, "The intuitive^{TM} telesurgery system: overview and application," in *IEEE International Conference on Robotics and Automation*, pp. 618–621, 2000.

[24] E. Rohmer, S. P. N. Singh, and M. Freese, "V-REP: A versatile and scalable robot simulation framework," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1321–1326, Nov 2013.

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

[26] G. A. Fontanelli, M. Selvaggio, M. Ferro, F. Ficuciello, M. Vendittelli, and B. Siciliano, "Portable dvrk: an augmented V-REP simulator of the da vinci research kit," *Acta Polytechnica Hungarica*, vol. 16, no. 8, pp. 79–98, 2019.

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

[28] E. Marchand, F. Spindler, and F. Chaumette, "Visp for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robotics and Automation Magazine*, vol. 12, no. 4, pp. 40–52, 2005.

[29] 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*, vol. 15, no. 3, p. e1981, 2019. e1981 rcs.1981.

[30] 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 (Submitted)*, 2020.

[31] 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.

[32] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: A survey," *Foundations and Trends in Human-Computer Interaction*, vol. 1, pp. 203–275, Jan. 2007.

[33] A. D. Santis, B. Siciliano, A. D. Luca, and A. Bicchi, "An atlas of physical humanrobot interaction," *Mechanism and Machine Theory*, vol. 43, no. 3, pp. 253 – 270, 2008.

[34] W. R. Ferrell and T. B. Sheridan, "Supervisory control of remote manipulation," *IEEE Spectrum*, vol. 4, pp. 81–88, Oct 1967.

[35] S. Thrun, "Toward robotic cars," *Communication of the ACM*, vol. 53, pp. 99–106, Apr. 2010.

[36] "Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems (sae international, 2016)." http://standards.sae.org/j3016_201609/, 2016. [Online; accessed 8-December-2019].

[37] G.-Z. Yang, J. Cambias, K. Cleary, E. Daimler, J. Drake, P. E. Dupont, N. Hata, P. Kazanzides, S. Martel, R. V. Patel, V. J. Santos, and R. H. Taylor, "Medical robotics—regulatory, ethical, and legal considerations for increasing levels of autonomy," *Science Robotics*, vol. 2, no. 4, 2017.

[38] G. A. Fontanelli, G. Yang, and B. Siciliano, "A comparison of assistive methods for suturing in mirs," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4389–4395, Oct 2018.

[39] Y. Li and S. S. Ge, "Humanrobot collaboration based on motion intention estimation," *IEEE/ASME Transactions on Mechatronics*, vol. 19, pp. 1007–1014, June 2014.

[40] D. P. Losey, C. G. McDonald, E. Battaglia, and M. K. O'Malley, "A Review of Intent Detection, Arbitration, and Communication Aspects of Shared Control for Physical HumanRobot Interaction," *Applied Mechanics Reviews*, vol. 70, 02 2018. 010804.

[41] S. Nikolaidis, Y. X. Zhu, D. Hsu, and S. Srinivasa, "Human-robot mutual adaptation in shared autonomy," in *ACM/IEEE International Conference on Human-Robot Interaction*, pp. 294–302, 2017.

[42] S. Nikolaidis, S. Nath, A. D. Procaccia, and S. Srinivasa, "Game-theoretic modeling of human adaptation in human-robot collaboration," in *ACM/IEEE International Conference on Human-Robot Interaction*, pp. 323–331, 2017.

[43] S. Nikolaidis, D. Hsu, and S. Srinivasa, "Human-robot mutual adaptation in collaborative tasks: Models and experiments," *The The International Journal of Robotics Research*, vol. 36, no. 5-7, pp. 618–634, 2017.

[44] Y. Li, K. P. Tee, W. L. Chan, R. Yan, Y. Chua, and D. K. Limbu, "Role adaptation of human and robot in collaborative tasks," in *IEEE International Conference on Robotics and Automation*, pp. 5602–5607, May 2015.

[45] Y. Li, K. P. Tee, R. Yan, W. L. Chan, and Y. Wu, "A framework of humanrobot coordination based on game theory and policy iteration," *IEEE Transactions on Robotics*, vol. 32, pp. 1408–1418, Dec 2016.

[46] Y. Li, K. P. Tee, W. L. Chan, R. Yan, Y. Chua, and D. K. Limbu, "Continuous role adaptation for humanrobot shared control," *IEEE Transactions on Robotics*, vol. 31, pp. 672–681, June 2015.

[47] V. Gabler, T. Stahl, G. Huber, O. Oguz, and D. Wollherr, "A game-theoretic approach for adaptive action selection in close proximity human-robot-collaboration," in *IEEE International Conference on Robotics and Automation*, pp. 2897–2903, May 2017.

[48] J. Vertut and P. Coiffet, *Teleoperations and Robotics: Evolution and Development*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1986.

[49] A. K. Bejczy, *Toward Advanced Teleoperation In Space*, pp. 107–138. 1994.

[50] P. S. Green, J. W. Hill, J. F. Jensen, and A. Shah, "Telepresence surgery," *IEEE Engineering in Medicine and Biology Magazine*, vol. 14, pp. 324–329, May 1995.

[51] R. H. 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, May 1995.

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

[53] A. J. Madhani, G. Niemeyer, and J. K. Salisbury, "The black falcon: a teleoperated surgical instrument for minimally invasive surgery," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 936–944 vol.2, Oct 1998.

[54] G. S. Guthart and J. K. Salisbury, "The intuitive/sup tm/ telesurgery system: overview and application," in *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 618–621 vol.1, April 2000.

[55] O. Khatib, X. Yeh, G. Brantner, B. Soe, B. Kim, S. Ganguly, H. Stuart, S. Wang, M. Cutkosky, A. Edsinger, P. Mullins, M. Barham, C. R. Voolstra, K. N. Salama, M. L'Hour, and V. Creuze, "Ocean one: A robotic avatar for oceanic discovery," *IEEE Robotics Automation Magazine*, vol. 23, pp. 20–29, Dec 2016.

[56] G. Brantner and O. Khatib, "Controlling ocean one," in *Field and Service Robotics* (M. Hutter and R. Siegwart, eds.), (Cham), pp. 3–17, Springer International Publishing, 2018.

[57] A. Birk, T. Doernbach, C. Mueller, T. uczynski, A. Gomez Chavez, D. Koehntopp, A. Kupcsik, S. Calinon, A. K. Tanwani, G. Antonelli, P. Di Lillo, E. Simetti, G. Casalino, G. Indiveri, L. Ostuni, A. Turetta, A. Caffaz, P. Weiss, T. Gobert, B. Chemisky, J. Gancet, T. Siedel,

S. Govindaraj, X. Martinez, and P. Letier, "Dexterous underwater manipulation from onshore locations: Streamlining efficiencies for remotely operated underwater vehicles," *IEEE Robotics Automation Magazine*, vol. 25, pp. 24–33, Dec 2018.

[58] D. Di Vito, P. Di Lillo, F. Arrichiello, and G. Antonelli, "Assistive control framework for remotely operated vehicles," in *2018 26th Mediterranean Conference on Control and Automation (MED)*, pp. 364–369, June 2018.

[59] P. Di Lillo, D. Di Vito, E. Simetti, G. Casalino, and G. Antonelli, "Satellite-based tele-operation of an underwater vehicle-manipulator system. preliminary experimental results," in *IEEE International Conference on Robotics and Automation*, pp. 7504–7509, May 2018.

[60] W. R. Ferrell, "Remote manipulation with transmission delay," *IEEE Transactions on Human Factors in Electronics*, vol. HFE-6, pp. 24–32, Sep. 1965.

[61] G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl, "Sensor-based space robotics-rotex and its telerobotic features," *IEEE Transactions on Robotics and Automation*, vol. 9, pp. 649–663, Oct 1993.

[62] J. H. Park and T. B. Sheridan, "Supervisory teleoperation control using computer graphics," in *Proceedings 1991 IEEE International Conference on Robotics and Automation*, pp. 493–498 vol.1, April 1991.

[63] T. Blackmon and L. W. Stark, "Model-based supervisory control in telerobotics," *Presence: Teleoperators and Virtual Environments*, vol. 5, pp. 205–223, 1996.

[64] G. Hirzinger, J. Heindl, K. Landzettel, and B. Brunner, "Multisensory shared autonomy - a key issue in the space robot technology experiment rotex," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 221–230, July 1992.

[65] B. Brunner, K. Arbter, and G. Hirzinger, "Task directed programming of sensor based robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 1080–1087, Sep. 1994.

[66] R. L. Williams, F. W. Harrison, and D. L. Soloway, "Shared control of multiple-manipulator, sensor-based telerobotic systems," in *Proceedings of International Conference on Robotics and Automation*, vol. 2, pp. 962–967 vol.2, April 1997.

[67] L. Conway, R. Volz, and M. Walker, "Tele-autonomous systems: Methods and architectures for intermingling autonomous and telerobotic technology," in *IEEE International Conference on Robotics and Automation*, vol. 4, pp. 1121–1130, March 1987.

[68] S. Hayati and S. T. Venkataraman, "Design and implementation of a robot control system with traded and shared control capability," in *IEEE International Conference on Robotics and Automation*, pp. 1310–1315 vol.3, May 1989.

[69] W. B. Griffin, W. R. Provancher, and M. R. Cutkosky, "Feedback strategies for telemanipulation with shared control of object handling forces," *Presence*, vol. 14, pp. 720–731, Dec 2005.

[70] 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, Oct 2005.

[71] A. Mrtl, M. Lawitzky, A. Kucukyilmaz, M. Sezgin, C. Basdogan, and S. Hirche, "The role of roles: Physical cooperation between humans and robots," *The The International Journal of Robotics Research*, vol. 31, no. 13, pp. 1656–1674, 2012.

[72] S. Musić and S. Hirche, "Control sharing in human-robot team interaction," *Annual Reviews in Control*, vol. 44, pp. 342 – 354, 2017.

[73] S. Musić, G. Salvietti, P. B. g. Dohmann, F. Chinello, D. Prattichizzo, and S. Hirche, "Humanrobot team interaction through wearable haptics for cooperative manipulation," *IEEE Transactions on Haptics*, vol. 12, pp. 350–362, July 2019.

[74] D. Sieber and S. Hirche, "Human-guided multirobot cooperative manipulation," *IEEE Transactions on Control Systems Technology*, vol. 27, pp. 1492–1509, July 2019.

[75] D. Sieber, S. Musić, and S. Hirche, "Multi-robot manipulation controlled by a human with haptic feedback," in *IEEE/RSJ International*

Conference on Intelligent Robots and Systems, pp. 2440–2446, Sep. 2015.

[76] A. Franchi, C. Secchi, M. Ryll, H. H. Bulthoff, and P. R. Giordano, "Shared control: Balancing autonomy and human assistance with a group of quadrotor uavs," *IEEE Robotics Automation Magazine*, vol. 19, pp. 57–68, Sep. 2012.

[77] C. Secchi, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano, "Bilateral teleoperation of a group of uavs with communication delays and switching topology," in *IEEE International Conference on Robotics and Automation*, pp. 4307–4314, 2012.

[78] A. Franchi, C. Secchi, H. I. Son, H. H. Bulthoff, and P. R. Giordano, "Bilateral teleoperation of groups of mobile robots with time-varying topology," *IEEE Transactions on Robotics*, vol. 28, pp. 1019–1033, Oct 2012.

[79] A. Franchi, P. Robuffo Giordano, C. Secchi, H. I. Son, and H. H. Blthoff, "A passivity-based decentralized approach for the bilateral teleoperation of a group of uavs with switching topology," in *IEEE International Conference on Robotics and Automation*, pp. 898–905, May 2011.

[80] F. Abi-Farraj, T. Osa, N. P. J. Peters, G. Neumann, and P. R. Giordano, "A learning-based shared control architecture for interactive task execution," in *IEEE International Conference on Robotics and Automation*, pp. 329–335, May 2017.

[81] D. Rakita, B. Mutlu, M. Gleicher, and L. M. Hiatt, "Shared control–based bimanual robot manipulation," *Science Robotics*, vol. 4, no. 30, 2019.

[82] I. Havoutis and S. Calinon, "Learning from demonstration for semi-autonomous teleoperation," *Autonomous Robots*, vol. 43, no. 3, pp. 713–726, 2019.

[83] I. Havoutis and S. Calinon, "Supervisory teleoperation with online learning and optimal control," in *IEEE International Conference on Robotics and Automation*, pp. 1534–1540, May 2017.

[84] P. Berthet-Rayne, M. Power, H. King, and G. Z. Yang, "Hubot: A three state human-robot collaborative framework for bimanual surgical

tasks based on learned models," in *IEEE International Conference on Robotics and Automation*, pp. 715–722, May 2016.

[85] M. J. A. Zeestraten, I. Havoutis, and S. Calinon, "Programming by demonstration for shared control with an application in teleoperation," *IEEE Robotics and Automation Letters*, vol. 3, pp. 1848–1855, July 2018.

[86] I. Havoutis and S. Calinon, "Learning assistive teleoperation behaviors from demonstration," in *IEEE International Symposium on Safety, Security, and Rescue Robotics*, pp. 258–263, Oct 2016.

[87] M. Shahbazi, S. F. Atashzar, C. Ward, H. A. Talebi, and R. V. Patel, "Multimodal sensorimotor integration for expert-in-the-loop telerobotic surgical training," *IEEE Transactions on Robotics*, vol. 34, pp. 1549–1564, Dec 2018.

[88] M. K. OMalley, A. Gupta, M. Gen, and Y. Li, "Shared Control in Haptic Systems for Performance Enhancement and Training," *Journal of Dynamic Systems, Measurement, and Control*, vol. 128, pp. 75–85, 11 2005.

[89] S. Vozar, Z. Chen, P. Kazanzides, and L. L. Whitcomb, "Preliminary study of virtual nonholonomic constraints for time-delayed teleoperation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4244–4250, 2015.

[90] J. Smisek, M. van Paassen, and A. Schiele, "Haptic guidance in bilateral teleoperation: Effects of guidance inaccuracy," in *IEEE World Haptics Conference*, pp. 500–505, 2015.

[91] J. v. Oosterhout, J. G. W. Wildenbeest, H. Boessenkool, C. J. M. Heemskerk, M. R. d. Baar, F. C. T. v. d. Helm, and D. A. Abbink, "Haptic shared control in tele-manipulation: Effects of inaccuracies in guidance on task execution," *IEEE Transactions on Haptics*, vol. 8, pp. 164–175, April 2015.

[92] D. Prattichizzo, C. Pacchierotti, and G. Rosati, "Cutaneous force feedback as a sensory subtraction technique in haptics," *IEEE Transactions on Haptics*, vol. 5, pp. 289–300, Fourth 2012.

[93] C. Pacchierotti, D. Prattichizzo, and K. J. Kuchenbecker, "Cutaneous feedback of fingertip deformation and vibration for palpation in robotic surgery," *IEEE Transactions on Biomedical Engineering*, vol. 63, pp. 278–287, Feb 2016.

[94] C. Pacchierotti, L. Meli, F. Chinello, M. Malvezzi, and D. Prattichizzo, "Cutaneous haptic feedback to ensure the stability of robotic teleoperation systems," *The The International Journal of Robotics Research*, vol. 34, no. 14, pp. 1773–1787, 2015.

[95] S. Follmer, D. Leithinger, A. Olwal, A. Hogge, and H. Ishii, "inform: Dynamic physical affordances and constraints through shape and object actuation," in *26th Annual ACM Symposium on User Interface Software and Technology*, pp. 417–426, 2013.

[96] B. Long, S. A. Seah, T. Carter, and S. Subramanian, "Rendering volumetric haptic shapes in mid-air using ultrasound," *ACM Transaction on Graphics*, vol. 33, pp. 181:1–181:10, Nov. 2014.

[97] 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, pp. 953–966, Dec 2004.

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

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

[100] M. Selvaggio, G. Notomista, F. Chen, B. Gao, F. Trapani, and D. Caldwell, "Enhancing bilateral teleoperation using camera-based online virtual fixtures generation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1483–1488, 2016.

[101] M. Selvaggio, F. Chen, B. Gao, G. Notomista, F. Trapani, and D. Caldwell, "Vision based virtual fixture generation for teleoperated robotic manipulation," in *IEEE International Conference on Advanced Robotics and Mechatronics*, pp. 190–195, 2016.

[102] C. P. Quintero, M. Dehghan, O. Ramirez, M. H. Ang, and M. Jagersand, "Flexible virtual fixture interface for path specification in telemanipulation," in *IEEE International Conference on Robotics and Automation*, pp. 5363–5368, May 2017.

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

[104] J. G. Petersen and F. R. Baena, "A dynamic active constraints approach for hands-on robotic surgery," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1966–1971, Nov 2013.

[105] M. Li, M. Ishii, and R. H. Taylor, "Spatial motion constraints using virtual fixtures generated by anatomy," *IEEE Transactions on Robotics*, vol. 23, pp. 4–19, Feb 2007.

[106] F. Chen, B. Gao, M. Selvaggio, Z. Li, D. Caldwell, K. Kershaw, A. Masi, M. Di Castro, and R. Losito, "A framework of teleoperated and stereo vision guided mobile manipulation for industrial automation," in *IEEE International Conference on Mechatronics and Automation*, pp. 1641–1648, Aug 2016.

[107] D. Aarno, S. Ekvall, and D. Kragic, "Adaptive virtual fixtures for machine-assisted teleoperation tasks," in *IEEE International Conference on Robotics and Automation*, pp. 1139–1144, April 2005.

[108] S. Ekvall, D. Aarno, and D. Kragic, "Online task recognition and real-time adaptive assistance for computer-aided machine control," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 1029–1033, 2006.

[109] R. Kikuuwe, N. Takesue, and H. Fujimoto, "A control framework to generate nonenergy-storing virtual fixtures: Use of simulated plasticity," *IEEE Transactions on Robotics*, 2008.

[110] D. Zhang, Q. Zhu, J. Xiong, and L. Wang, "Dynamic virtual fixture on the euclidean group for admittance-type manipulator in deforming environments," *BioMedical Engineering OnLine*, vol. 13, no. 1, p. 51, 2014.

[111] S. A. Bowyer and F. Rodriguez y Baena, "Dynamic frictional constraints for robot assisted surgery," in *World Haptics Conference*, pp. 319–324, April 2013.

[112] S. A. Bowyer and F. R. y. Baena, "Dynamic frictional constraints in translation and rotation," in *IEEE International Conference on Robotics and Automation*, pp. 2685–2692, May 2014.

[113] N. Enayati, E. C. A. Costa, G. Ferrigno, and E. De Momi, "A dynamic non-energy-storing guidance constraint with motion redirection for robot-assisted surgery," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4311–4316, Oct 2016.

[114] D. Nicolis, A. M. Zanchettin, and P. Rocco, "A hierarchical optimization approach to robot teleoperation and virtual fixtures rendering," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5672 – 5679, 2017.

[115] B. L. Davies, S. J. Harris, W. J. Lin, R. D. Hibberd, R. Middleton, and J. C. Cobb, "Active compliance in robotic surgerythe use of force control as a dynamic constraint," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 211, no. 4, pp. 285–292, 1997. PMID: 9330539.

[116] A. Munawar and G. Fischer, "A surgical robot teleoperation framework for providing haptic feedback incorporating virtual environment-based guidance," *Frontiers in Robotics and AI*, vol. 3, p. 47, 2016.

[117] M. Li, A. Kapoor, and R. H. Taylor, *Telerobotic Control by Virtual Fixtures for Surgical Applications*, pp. 381–401. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[118] K. Leibrandt, H. J. Marcus, K. W. Kwok, and G. Z. Yang, "Implicit active constraints for a compliant surgical manipulator," in *IEEE International Conference on Robotics and Automation*, pp. 276–283, May 2014.

[119] J. Melo, E. Snchez, and I. Daz, "Adaptive admittance control to generate real-time assistive fixtures for a cobot in transpedicular fixation surgery," in *4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics*, pp. 1170–1175, 2012.

[120] Z. Chen, A. Malpani, P. Chalasani, A. Deguet, S. S. V. Vedula, R. H. Taylor, and P. Kazanzides, "Virtual Fixture Assistance for Needle Passing and Knot Tying," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2343–2350, 2016.

[121] M. Shahbazi, S. F. Atashzar, and R. V. Patel, "A dual-user teleoperated system with virtual fixtures for robotic surgical training," in *2013 IEEE International Conference on Robotics and Automation*, pp. 3639–3644, 2013.

[122] M. Shahbazi, S. F. Atashzar, H. A. Talebi, and R. V. Patel, "An expertise-oriented training framework for robotics-assisted surgery," in *IEEE International Conference on Robotics and Automation*, pp. 5902–5907, 2014.

[123] G. P. Moustris, A. I. Mantelos, and C. S. Tzafestas, "Enhancing surgical accuracy using virtual fixtures and motion compensation in robotic beating heart surgery," in *21st Mediterranean Conference on Control and Automation*, pp. 1254–1260, 2013.

[124] G. P. Moustris, A. I. Mantelos, and C. S. Tzafestas, "Shared control for motion compensation in robotic beating heart surgery," in *IEEE International Conference on Robotics and Automation*, pp. 5819–5824, 2013.

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

[126] A. Saracino, A. Deguet, F. Staderini, M. N. Boushaki, F. Cianchi, A. Menciassi, and E. Sinibaldi, "Haptic feedback in the da vinci research kit (dvrk): A user study based on grasping, palpation, and incision tasks," *Int J Med Robotics Comput Assist Surg*, vol. 15, no. 4, p. e1999, 2019.

[127] P. Schleer, S. Drobinsky, and K. Radermacher, "Evaluation of different modes of haptic guidance for robotic surgery," *IFAC-PapersOnLine*, vol. 51, no. 34, pp. 97 – 103, 2019. 2nd IFAC Conference on Cyber-Physical and Human Systems CPHS 2018.

[128] M. M. Coad, A. M. Okamura, S. Wren, Y. Mintz, T. S. Lendvay, A. M. Jarc, and I. Nisky, "Training in divergent and convergent force fields

during 6-dof teleoperation with a robot-assisted surgical system," in *IEEE World Haptics Conference*, pp. 195–200, June 2017.

[129] P. Marayong and A. M. Okamura, "Speed-accuracy characteristics of human-machine cooperative manipulation using virtual fixtures with variable admittance," *Human Factors*, vol. 46, no. 3, pp. 518–532, 2004. PMID: 15573549.

[130] M. Power, H. Rafii-Tari, C. Bergeles, V. Vitiello, and G.-Z. Yang, "A cooperative control framework for haptic guidance of bimanual surgical tasks based on learning from demonstration," in *IEEE International Conference on Robotics and Automation*, pp. 5330–5337, May 2015.

[131] C. J. P. del Pulgar, J. Smisek, V. F. Muoz, and A. Schiele, "Using learning from demonstration to generate real-time guidance for haptic shared control," in *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 003205–003210, Oct 2016.

[132] A. Chowriappa, R. Wirz, A. R. Ashammagari, and Y. W. Seo, "Prediction from expert demonstrations for safe tele-surgery," *International Journal of Automation and Computing*, vol. 10, no. 6, pp. 487–497, 2013.

[133] Y. Kassahun, B. Yu, A. T. Tibebu, D. Stoyanov, S. Giannarou, J. H. Metzen, and E. Vander Poorten, "Surgical robotics beyond enhanced dexterity instrumentation: a survey of machine learning techniques and their role in intelligent and autonomous surgical actions," *Int J Comput Assist Radiol Surg*, vol. 11, p. 847, May 2016.

[134] A. Murali, S. Sen, B. Kehoe, A. Garg, S. McFarland, S. Patil, W. D. Boyd, S. Lim, P. Abbeel, and K. Goldberg, "Learning by observation for surgical subtasks: Multilateral cutting of 3d viscoelastic and 2d orthotropic tissue phantoms," in *IEEE International Conference on Robotics and Automation*, pp. 1202–1209, May 2015.

[135] A. Safavi and M. H. Zadeh, "Model-Based Haptic Guidance in Surgical Skill Improvement," in *Proceedings - 2015 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015*, pp. 1104–1109, 2016.

[136] K. Leibrandt, C. Bergeles, and G. Z. Yang, "On-line collision-free inverse kinematics with frictional active constraints for effective control

of unstable concentric tube robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3797–3804, Sept 2015.

[137] K. Leibrandt, C. Bergeles, and G. Z. Yang, "Implicit active constraints for safe and effective guidance of unstable concentric tube robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1157–1163, Oct 2016.

[138] K. Leibrandt and G. Z. Yang, "Efficient proximity queries for continuum robots on parallel computing hardware," *IEEE Robotics and Automation Letters*, vol. 2, pp. 1548–1555, July 2017.

[139] K. W. Kwok, K. H. Tsoi, V. Vitiello, J. Clark, G. C. T. Chow, W. Luk, and G. Z. Yang, "Dimensionality reduction in controlling articulated snake robot for endoscopy under dynamic active constraints," *IEEE Transactions on Robotics*, vol. 29, pp. 15–31, Feb 2013.

[140] D. A. Lawrence, "Stability and transparency in bilateral teleoperation," in *31st IEEE Conference on Decision and Control*, pp. 2649–2655 vol.3, Dec 1992.

[141] D. A. Lawrence, "Stability and transparency in bilateral teleoperation," *IEEE Transactions on Robotics and Automation*, vol. 9, pp. 624–637, Oct 1993.

[142] R. J. Anderson and M. W. Spong, "Asymptotic stability for force reflecting teleoperators with time delays," in *IEEE International Conference on Robotics and Automation*, pp. 1618–1625 vol.3, May 1989.

[143] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035–2057, 2006.

[144] E. Nuo, L. Basaez, and R. Ortega, "Passivity-based control for bilateral teleoperation: A tutorial," *Automatica*, vol. 47, no. 3, pp. 485 – 495, 2011.

[145] S. Musić and S. Hirche, "Passive noninteracting control for human-robot team interaction," in *IEEE Conference on Decision and Control*, pp. 421–427, Dec 2018.

[146] K. Kronander and A. Billard, "Stability considerations for variable impedance control," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1298–1305, 2016.

[147] E. Shahriari, A. Kramberger, A. Gams, A. Ude, and S. Haddadin, "Adapting to contacts: Energy tanks and task energy for passivity-based dynamic movement primitives," in *IEEE-RAS 17th International Conference on Humanoid Robotics*, pp. 136–142, 2017.

[148] G. Niemeyer and J. . E. Slotine, "Stable adaptive teleoperation," *IEEE Journal of Oceanic Engineering*, vol. 16, pp. 152–162, Jan 1991.

[149] G. Niemeyer, "Using wave variables in time delayed force reflecting teleoperation," 1996.

[150] Y. Yokokohji, T. Imaida, and T. Yoshikawa, "Bilateral control with energy balance monitoring under time-varying communication delay," in *IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2684–2689 vol.3, April 2000.

[151] D. Lee and M. W. Spong, "Passive bilateral teleoperation with constant time delay," *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 269–281, 2006.

[152] Dongjun Lee and P. Y. Li, "Passive coordination control of nonlinear bilateral teleoperated manipulators," in *Proceedings 2002 IEEE International Conference on Robotics and Automation*, vol. 3, pp. 3278–3283 vol.3, May 2002.

[153] C. Schindlbeck and S. Haddadin, "Unified passivity-based cartesian force/impedance control for rigid and flexible joint robots via task-energy tanks," in *IEEE International Conference on Robotics and Automation*, pp. 440–447, May 2015.

[154] F. Ferraguti, N. Preda, A. Manurung, M. Bonfè, O. Lambercy, R. Gassert, R. Muradore, P. Fiorini, and C. Secchi, "An energy tank-based interactive control architecture for autonomous and teleoperated robotic surgery," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1073–1088, 2015.

[155] M. Franken, S. Stramigioli, S. Misra, C. Secchi, and A. Macchelli, "Bilateral telemanipulation with time delays: a two-layer approach combining passivity and transparency," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 741–756, 2011.

[156] D. Lee and K. Huang, "Passive-set-position-modulation framework for interactive robotic systems," *IEEE Transactions on Robotics*, vol. 26, pp. 354–369, April 2010.

[157] B. Hannaford and Jee-Hwan Ryu, "Time-domain passivity control of haptic interfaces," *IEEE Transactions on Robotics and Automation*, vol. 18, pp. 1–10, Feb 2002.

[158] Jee-Hwan Ryu, Dong-Soo Kwon, and B. Hannaford, "Stable teleoperation with time-domain passivity control," *IEEE Transactions on Robotics and Automation*, vol. 20, pp. 365–373, April 2004.

[159] Jee-Hwan Ryu, Yoon Sang Kim, and B. Hannaford, "Sampled- and continuous-time passivity and stability of virtual environments," *IEEE Transactions on Robotics*, vol. 20, pp. 772–776, Aug 2004.

[160] Jee-Hwan Ryu, C. Preusche, B. Hannaford, and G. Hirzinger, "Time domain passivity control with reference energy following," *IEEE Transactions on Control Systems Technology*, vol. 13, pp. 737–742, Sep. 2005.

[161] S. A. Bowyer and F. Rodriguez y Baena, "Dissipative control for physical humanrobot interaction," *IEEE Transactions on Robotics*, vol. 31, pp. 1281–1293, Dec 2015.

[162] C. Freschi, V. Ferrari, F. Melfi, M. Ferrari, F. Mosca, and A. Cuschieri, "Technical review of the da vinci surgical telemanipulator," *Int J Med Robot*, vol. 9, no. 4, pp. 396–406, 2013.

[163] A. Ghalamzan E., F. Abi-Farraj, P. R. Giordano, and R. Stolkin, "Human-in-the-loop optimisation: Mixed initiative grasping for optimally facilitating post-grasp manipulative actions," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3386–3393, 2017.

[164] D. Aschenbrenner, M. Fritscher, F. Sittner, M. Krauß, and K. Schilling, *Teleoperation of an Industrial Robot in an Active Production Line*. 2015.

[165] C. Pacchierotti, F. Ongaro, F. van den Brink, C. Yoon, D. Prattichizzo, D. H. Gracias, and S. Misra, "Steering and control of miniaturized untethered soft magnetic grippers with haptic assistance," *IEEE Transaction Automation Science and Engineering*, vol. 15, no. 1, pp. 290–306, 2018.

[166] C. Pacchierotti, S. Scheggi, D. Prattichizzo, and S. Misra, "Haptic feedback for microrobotics applications: a review," *Frontiers in Robotics and AI*, vol. 3, no. 53, 2016.

[167] P. LeBel, C. Gosselin, and A. Campeau-Lecours, "An anticipative kinematic limitation avoidance algorithm for collaborative robots: Three-dimensional case," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3075–3080, 2017.

[168] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 500–505, 1985.

[169] A. D. Santis, A. Albu-Schaffer, C. Ott, B. Siciliano, and G. Hirzinger, "The skeleton algorithm for self-collision avoidance of a humanoid manipulator," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1–6, 2007.

[170] A. Dietrich, T. Wimbock, A. Albu-Schaffer, and G. Hirzinger, "Integration of reactive, torque-based self-collision avoidance into a task hierarchy," *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1278–1293, 2012.

[171] C. Dube, M. Tsoeu, and J. Tapson, "A model of the humanoid body for self collision detection based on elliptical capsules," in *IEEE International Conference on Robotics and Biomimetics*, pp. 2397–2402, 2011.

[172] K. Steinbach, J. Kuffner, T. Asfour, and R. Dillmann, "Efficient collision and self-collision detection for humanoids based on sphere trees hierarchies," in *IEEE-RAS International Conference on Humanoid Robots*, pp. 560–566, 2006.

[173] F. Seto, K. Kosuge, and Y. Hirata, "Self-collision avoidance motion control for human robot cooperation system using robe," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3143–3148, 2005.

[174] B. Dariush, G. B. Hammam, and D. Orin, "Constrained resolved acceleration control for humanoids," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 710–717, 2010.

[175] T. F. Chan and R. V. Dubey, "A weighted least-norm solution based scheme for avoiding joint limits for redundant manipulators," in *IEEE International Conference on Robotics and Automation*, pp. 395–402 vol.3, 1993.

[176] M. Selvaggio, S. Grazioso, G. Notomista, and F. Chen, "Towards a self-collision aware teleoperation framework for compound robots," in *IEEE World Haptics Conference*, pp. 460–465, 2017.

[177] H. Saeidi, J. R. Wagner, and Y. Wang, "A mixed-initiative haptic teleoperation strategy for mobile robotic systems based on bidirectional computational trust analysis," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1500–1507, 2017.

[178] H. Boessenkool, D. A. Abbink, C. J. M. Heemskerk, F. C. T. van der Helm, and J. G. W. Wildenbeest, "A task-specific analysis of the benefit of haptic shared control during telemanipulation," *IEEE Transactions on Haptics*, vol. 6, no. 1, pp. 2–12, 2013.

[179] B. Khademian and K. Hashtrudi-Zaad, "Shared control architectures for haptic training: Performance and coupled stability analysis," *Int J Rob Res*, vol. 30, no. 13, pp. 1627–1642, 2011.

[180] D. Nicolis, M. Palumbo, A. M. Zanchettin, and P. Rocco, "Occlusion-free visual servoing for the shared autonomy teleoperation of dual-arm robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 796–803, 2018.

[181] J. Cascio, M. Karpenko, Q. Gong, P. Sekhavat, and I. M. Ross, "Smooth proximity computation for collision-free optimal control of multiple robotic manipulators," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2452–2457, 2009.

[182] F. Chaumette and S. Hutchinson, "Visual servo control, part i: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[183] B. Siciliano and J. J. E. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *International Conference on Advanced Robotics*, pp. 1211–1216 vol.2, 1991.

[184] M. Selvaggio, P. Robuffo Giordano, F. Ficuciello, and B. Siciliano, "Passive task-prioritized shared-control teleoperation with haptic guidance," in *2019 International Conference on Robotics and Automation*, pp. 430–436, May 2019.

[185] A. D. Dragan and S. S. Srinivasa, "A policy-blending formalism for shared control," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, 2013.

[186] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *The International Journal of Robotics Research*, vol. 6, no. 2, pp. 3–15, 1987.

[187] P. Baerlocher and R. Boulic, "Task-priority formulations for the kinematic control of highly redundant articulated structures," in *IEEE/RSJ International Conference Intell. Rob. Sys.*, vol. 1, pp. 323–329, 1998.

[188] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. Robot. Autom.*, vol. 3, no. 1, pp. 43–53, 1987.

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

[190] M. Selvaggio, F. Abi-Farraj, C. Pacchierotti, P. R. Giordano, and B. Siciliano, "Haptic-based shared-control methods for a dual-arm system," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4249–4256, 2018.

[191] S. Javdani, H. Admoni, S. Pellegrinelli, S. S. Srinivasa, and J. A. Bagnell, "Shared autonomy via hindsight optimization for teleoperation and teaming," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 717–742, 2018.

[192] M. Laghi, M. Maimeri, M. Marchand, C. Leparoux, M. Catalano, A. Ajoudani, and A. Bicchi, "Shared-autonomy control for intuitive bimanual tele-manipulation," in *IEEE-RAS 18th International Conference on Humananoid Robots*, pp. 1–9, 2018.

[193] Y. Liu and N. Chopra, "Semi-autonomous teleoperation in task space with redundant slave robot under communication delays," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 679–684, 2011.

[194] Y.-C. Liu and N. Chopra, "Control of semi-autonomous teleoperation system with time delays," *Automatica*, vol. 49, no. 6, pp. 1553 – 1565, 2013.

[195] F. Abi-Farraj, C. Pacchierotti, and P. R. Giordano, "User evaluation of a haptic-enabled shared-control approach for robotic telemanipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1–9, 2018.

[196] J. Bimbo, C. Pacchierotti, M. Aggravi, N. Tsagarakis, and D. Prattichizzo, "Teleoperation in cluttered environments using wearable haptic feedback," in *IEEE/RSJ International Conference Intelligent Robots and Systems*, pp. 3401–3408, 2017.

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

[198] 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, pp. 3129–3136, Oct 2018.

[199] B. Siciliano, "Kinematic control of redundant robot manipulators: A tutorial," *Journal of Intelligent and Robotic Systems*, vol. 3, no. 3, pp. 201–212, 1990.

[200] F. Ferraguti, C. Secchi, and C. Fantuzzi, "A tank-based approach to impedance control with variable stiffness," in *IEEE International Conference on Robotics and Automation*, pp. 4948–4953, May 2013.

[201] C. T. Landi, F. Ferraguti, L. Sabattini, C. Secchi, and C. Fantuzzi, "Admittance control parameter adaptation for physical human-robot interaction," in *IEEE International Conference on Robotics and Automation*, pp. 2911–2916, 2017.

[202] A. Dietrich, C. Ott, and S. Stramigioli, "Passivation of projection-based null space compliance control via energy tanks," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 184–191, 2016.

[203] A. Dietrich, X. Wu, K. Bussmann, C. Ott, A. Albu-Schffer, and S. Stramigioli, "Passive hierarchical impedance control via energy tanks," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 522–529, 2017.

[204] R. Moccia, M. Selvaggio, B. Siciliano, A. Arezzo, and F. Ficuciello, "A virtual fixture adaptation strategy for MIRS dissection tasks," *8th Joint Workshop on New Technologies for Computer/Robot Assisted Surgery*, 2018.

[205] C. Simone and A. M. Okamura, "Modeling of needle insertion forces for robot-assisted percutaneous therapy," in *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 2085–2091 vol.2, May 2002.

[206] T. C. Clark and F. H. Schmidt, "Robot-assisted navigation versus computer-assisted navigation in primary total knee arthroplasty: Efficiency and accuracy," *ISRN orthopedics*, vol. 2013, pp. 794827–794827, Jun 2013. 24967115[pmid].

[207] A. Kapoor, M. Li, and R. H. Taylor, "Spatial motion constraints for robot assisted suturing using virtual fixtures," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005* (J. S. Duncan and G. Gerig, eds.), (Berlin, Heidelberg), pp. 89–96, Springer Berlin Heidelberg, 2005.

[208] M. M. Marinho, H. Ishida, K. Harada, D. Kyoichi, and M. Mitsuishi, "Virtual fixture assistance for suturing in robot-aided pediatric endoscopic surgery," *IEEE Robotics and Automation Letters (RA-L)*, vol. 5, pp. 524–531, Apr. 2020.

[209] 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, pp. 1166–1185, Oct. 2019.

[210] S. Park, R. D. Howe, and D. F. Torchiana, "Virtual fixtures for robotic cardiac surgery," in *Medical Image Computing and Computer-Assisted Intervention*, pp. 1419–1420, Springer Berlin Heidelberg, 2001.

[211] N. Enayati, E. C. A. Costa, G. Ferrigno, and E. D. Momi, "A dynamic non-energy-storing guidance constraint with motion redirection for robot-assisted surgery," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4311–4316, 2016.

[212] F. Ferraguti, N. Preda, M. Bonf, and C. Secchi, "Bilateral teleoperation of a dual arms surgical robot with passive virtual fixtures generation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4223–4228, Sep. 2015.

[213] A. Toedtheide, E. Shahriari, and S. Haddadin, "Tank based unified torque/impedance control for a pneumatically actuated antagonistic robot joint," in *IEEE International Conference on Robotics and Automation*, pp. 1255–1262, 2017.

[214] 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," in *IEEE International Conference on Robotics and Automation*, pp. 6434–6439, 2014.

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

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

[217] R. Moccia, M. Selvaggio, L. Villani, B. Siciliano, and F. Ficuciello, "Vision-based virtual fixtures generation for robotic-assisted polyp dissection procedures," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 7928–7933, 2019.

[218] C. Secchi, S. Stramigioli, and C. Fantuzzi, "Position drift compensation in port-hamiltonian based telemanipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4211–4216, 2006.

[219] 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," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 131–136, 2017.

[220] S. Bochkanov, "Alglib."

[221] M. Selvaggio, G. A. Fontanelli, F. Ficuciello, L. Villani, and B. Siciliano, "Task classification of robotic surgical reconstructive procedures using force measurements," in *2017 7th Joint Workshop on New Technologies for Computer/Robot Assisted Surgery*, 2017.

[222] M. Selvaggio, L. Villani, B. Siciliano, and F. Ficuciello, "Physics-based task classification of da vinci robot surgical procedures," in *2018 6th National Congress of Bioengineering*, 2018.

[223] 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," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3617–3623, 2019.

[224] M. Kopicki, R. Detry, M. Adjigble, R. Stolkin, A. Leonardis, and J. L. Wyatt, "One-shot learning and generation of dexterous grasps for novel objects," *The The International Journal of Robotics Research*, vol. 35, no. 8, p. 959976, 2015.

[225] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[226] M. Gualtieri, A. ten Pas, K. Saenko, and R. Platt, "High precision grasp pose detection in dense clutter," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 598–605, 2016.

[227] C. D'Ettorre, G. Dwyer, X. Du, F. Chadebecq, F. Vasconcelos, E. D. Momi, and D. Stoyanov, "Automated pick-up of suturing needles for robotic surgical assistance," in *IEEE International Conference on Robotics and Automation*, pp. 1370–1377, 2018.

[228] M. Ragaglia, "Robot learning from demonstrations: Emulation learning in environments with moving obstacles," *Robotics and Autonomous Systems*, vol. 101, pp. 45–56, 2018.

[229] T. Osa, N. Sugita, and M. Mitsuishi, "Online trajectory planning in dynamic environments for surgical task automation.," in *Robotics: Science and Systems*, pp. 1–9, 2014.

[230] R. Detry, J. Papon, and L. Matthies, "Task-oriented grasping with semantic and geometric scene understanding," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3266–3273, 2017.

[231] N. Mavrakis, R. Stolkin, L. Baronti, M. Kopicki, and M. Castellani, "Analysis of the inertia and dynamics of grasped objects, for choosing optimal grasps to enable torque-efficient post-grasp manipulations," in *IEEE-RAS International Conference Humanoid Robots*, pp. 171–178, 2016.

[232] T. Pardi, R. Stolkin, and A. Ghalamzan E, "Choosing grasps to enable collision-free post-grasp manipulations," in *IEEE-RAS 18th International Conference Humanoid Robots*, pp. 299–305, 2018.

[233] N. Ahmidi, L. Tao, S. Sefati, Y. Gao, C. Lea, B. Bejar, L. Zappella, S. Khudanpur, R. Vidal, and G. D. Hager, "A dataset and benchmarks for segmentation and recognition of gestures in robotic surgery," *IEEE Transactions Biomedical Engineering*, pp. 2025–2041, 2017.

[234] 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," *IEEE International Conference on Robotics and Automation*, pp. 4178–4185, 2016.

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

[236] T. Liu and M. C. Cavusoglu, "Optimal needle grasp selection for automatic execution of suturing tasks in robotic minimally invasive surgery," in *IEEE International Conference on Robotics and Automation*, pp. 2894–2900, 2015.

[237] R. C. Jackson and M. C. Cavusoglu, "Needle path planning for autonomous robotic surgical suturing," in *IEEE International Conference on Robotics and Automation*, pp. 1669–1675, 2013.

[238] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Learning movement primitives," in *International Symposium on Robotics Research*, pp. 561–572, Springer, 2005.

[239] P. Francis, K. W. Eastwood, V. Bodani, K. Price, K. Upadhyaya, D. Podolsky, H. Azimian, T. Looi, and J. Drake, "Miniaturized instruments for the da Vinci research kit: Design and implementation of custom continuum tools," *IEEE Robotics Automation Magazine*, vol. 24, pp. 24–33, June 2017.

[240] D. Martin, J. Woodard, C. Shurtleff, and A. Yoo, "Articulating needle driver," Nov. 15 2012. US Patent App. 13/466,188.

[241] A. Spiers, S. Baillie, T. Pipe, and R. Persad, "Experimentally driven design of a palpating gripper with minimally invasive surgery considerations," *Haptics Symposium 2012*, pp. 261–266, 2012.

[242] N. Rahman, L. Carbonari, M. D'Imperio, C. Canali, D. G. Caldwell, and F. Cannella, "A dexterous gripper for in-hand manipulation," in *IEEE International Conference on Advanced Intelligent Mechatronics*, pp. 377–382, July 2016.

[243] N. Rahman, M. D'Imperio, L. Carbonari, F. Chen, C. Canali, D. G. Caldwell, and F. Cannella, "A novel bio-inspired modular gripper for in-hand manipulation," in *IEEE International Conference on Robotics and Biomimetics*, pp. 7–12, Dec 2015.

[244] N. Rojas, R. R. Ma, and A. M. Dollar, "The GR2 gripper: An underactuated hand for open-loop in-hand planar manipulation," *IEEE Transactions on Robotics*, vol. 32, pp. 763–770, June 2016.

[245] R. M. Greenhalgh, "Techniques of anastomosis," in *Vascular Surgical Techniques*, pp. 5 – 13, 1984.

[246] F. Fazioli, F. Ficuciello, G. A. Fontanelli, B. Siciliano, and L. Villani, "Implementation of a soft-rigid collision detection algorithm in an open-source engine for surgical realistic simulation," in *IEEE International Conference on Robotics and Biomimetics*, pp. 2204–2208, 2016.

[247] A. Talasaz, A. L. Trejos, and R. V. Patel, "The role of direct and visual force feedback in suturing using a 7-DOF dual-arm teleoperated system," *IEEE Transactions on Haptics*, vol. 10, no. 2, pp. 276–287, 2017.

[248] A. Petit, F. Ficuciello, G. A. Fontanelli, L. Villani, and B. Siciliano, "Using physical modeling and RGB-D registration for contact force sensing

on deformable objects," in *IEEE International Conference on Informatics in Control, Automation and Robotics*, pp. 24–33, 2017.

[249] T. Ahlering, *da Vinci prostatectomy 3-arm and 4-arm approach*. Intuitive Surgical, Inc., 2013.

[250] M. Selvaggio, G. A. Fontanelli, F. Ficuciello, L. Villani, and B. Siciliano, "Enhancing dexterity with a 7-dof laparoscopic suturing tool," in *2018 Hamlyn Symposium on Medical Robotics*, 2018.

[251] N. Vahrenkamp, T. Asfour, G. Metta, G. Sandini, and R. Dillmann, "Manipulability analysis," in *IEEE-RAS International Conference Humanoid Robots*, pp. 568–573, 2012.

[252] F. Chen, M. Selvaggio, and D. G. Caldwell, "Dexterous grasping by manipulability selection for mobile manipulator with visual guidance," *IEEE Transactions on Industrial Informatics*, vol. 15, pp. 1202–1210, Feb 2019.

[253] S. Tadokoro, I. Kimura, and T. Takamori, "A dexterity measure for trajectory planning and kinematic design of redundant manipulators," in *Annual Conference of IEEE Industrial Electronics Society*, pp. 415–420, 1989.