



POLITECNICO
MILANO 1863

**DEPARTMENT OF ELECTRONICS, INFORMATION AND
BIOENGINEERING**

MASTER OF SCIENCE IN BIOMEDICAL ENGINEERING

**Implementation and Assessment of Augmented
Surgical Training with a *da Vinci* robot: an
experimental study**

Author

Alberto Rota

Supervisor

Prof. Elena De Momi

Co-supervisor

Ke Fan

Contents

Abstract	5
Sommario	6
1 Introduction	9
1.1 Context	11
1.2 Motivation	12
2 State of the Art	13
2.1 The <i>daVinci</i> ® Surgical System	13
2.1.1 The Surgeon Console	14
2.1.2 The Patient Cart	15
2.1.3 The Vision Cart	17
2.2 VR Surgical Simulators	17
2.3 Virtual Fixtures and Augmented Training	19
3 Materials and Methods	21
3.1 Equipment	21
3.2 The Surgical Simulator	21
3.2.1 The Surgical Tasks	22
3.3 Virtual Fixtures	24
3.3.1 Error Mapping	26
3.3.2 Virtual Fixtures Algorithms	26
3.4 Clinical Validation	35
3.5 Experimental Protocol	35
3.6 Performance Metrics	37
4 Results	41

5 Discussion	45
6 Conclusion	47
List of Figures	49
List of Tables	51
References	53

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Sommario

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Chapter 1

Introduction

Since its advent in the middle of the 1980s, robotic surgery has revolutionized the healthcare industry introducing safer and more efficient solutions to the challenges of surgical practices. By embracing robotic devices in a collaborative effort, surgeons and medical practitioners have been empowered with tools engineered to enhance their skills and optimize patient outcomes. From dental to orthopedic to neurosurgery, almost every area of the medical panorama has now been reached by robotics, with regulatory organizations approving novel procedures with increasing frequency.

Formally, *robot-assisted surgeries* are operations where the medical team works with a robotic device that actively interacts with the patient's body, manipulating tissues with precision instruments that are mounted on the machine itself. However, there is a major distinction between two methodologies of deploying robotic assistance [1]:

- **Teleoperated systems** involve a user who directly and continuously interacts with the robot, controlling its motion mainly by hand-held manipulators which replicate the surgeon's movement to the robotic actuation
- **Image-guided systems** rely on the acquisition of medical imaging data (in the form of CT scans, MRIs, *etc.*), later used for planning the motion of the robotic tools, without a direct and real-time interface with the surgeon

Mixed approaches are also possible, where teleoperation is accompanied by fully automated sub-tasks or where image guidance is updated in real-time by information gathered from sensors. In all cases, controlling the position, orientation and motion of the robotic system is, however, a responsibility of the surgeon himself, that takes advantage of the high precision potentiality of the mechanical system for achieving a less invasive, less error-prone and safer

procedure. These systems are not meant to replace the physician, but rather to augment its capabilities [2]. Such a synergistic perspective highlights the role of the Human-Robot Interaction (HRI) paradigm, which involves all aspects of understanding, designing, and evaluating robotic systems for use by or with humans [3]: in the context of teleoperated surgical robots, HRI includes all the hardware and software features that enhance the surgeon’s experience, improve his performance, grant a higher level of safety for the patient and achieve better surgical outcomes.

Modern surgical robotic systems like the *daVinci*® (Intuitive Surgical, Inc.) are extremely complex devices and, as such, demand long and extensive training programs to medical facilities and surgeons who want to operate them. In the past, surgical training was conducted on semi-realistic plastic phantoms, animals or cadavers, which other than not being a reusable resource in a lot of cases came out to be non-cost-effective solutions. More modern approaches consist, instead, of virtual environments where a simulated surgical scenario is re-constructed with a discrete level of realism, and where physics engines emulate the interaction between the virtual objects. A Virtual Reality (VR) environment has multiple advantages compared to the dated approaches above: infinite customizability and repeatability, non-destructiveness, easy setup, reduced costs, accurate progress monitoring, *etc.* Virtual environments, also, allow an easier developing and deploying of assistive algorithms that, by running “in parallel” to the surgeon’s teleoperation, contribute to the HRI paradigm in terms of performance and safety.

Surgical assistance has become an impactful element in the most recent surgical robotic solutions on the market, for the most part concerning visual cues super-imposed to the camera feed. For example, such visual cues may consist of the detection and localization, on the screen, of delicate surgical structures that should remain untouched by the instrumentation. Deep Learning and other modern AI-based computer vision techniques are the most useful in this context, and a few commercialized surgical robots already employ such assistive strategies.

Most of the surgical robotics solutions on the market consist of a teleoperation console that interfaces with the practitioner and, separate from it, the surgical robot itself, which mimics the movements of the surgeon in real-time. This setup allows for higher motion accuracy, tremor filtration and magnified viewing of the surgical area; nonetheless tactile forces, friction and texture perception are excluded from the so crucial visuo-haptic feedback loop that would guide the surgeon in a standard “non-robotic” procedure. This work, specifically, investigates the role of haptic assistance in the context of HRI, and analyses the role of mechanical forces when employed as a guidance medium.



Figure 1.1: A surgical team in the operating room using a *daVinci*® robot. Photo courtesy of *Intuitive Surgical Inc.*

1.1 Context

Surgical robotics training is a crucial aspect of the medical education process: aspiring robotic surgeons should develop a highly-specific skillset that includes cognitive, motor, and perceptual abilities that are not typical of any other medical field. Usually, medical facilities collaborate with the company commercializing the robot with the aim of defining a correct, extensive and effective training program that will yield skilled surgeons who will make the most out of the robot utilization in the Operatory Room (OR). If in the past these training programs were targeted only to experienced surgeons with established surgical skillsets, more recently they have been extended to medical students and residents, who conduct this phase in parallel to the standard medical education program: this attests how the surgical robotics field is becoming more and more mainstream, and how the medical community is embracing this new technology.

Assistance strategies are usually featured in the training programs, and they are meant to guide the trainee toward the optimal execution of a surgical task or to help him overcome a

specific difficulty. This role is also covered by an expert supervisor, who is usually present alongside the trainee and provides suggestions and corrections regarding the execution of the procedure. With simulated environments and VR, a supervision of this kind can be implemented in the software and delivered autonomously, for example showing the trainee the correct way to perform a specific maneuver or highlighting the presence of an error.

Haptic assistance however is not yet a common feature in surgical robotics training programs. In a few cases, this is due to the lack of a haptic interface in the training setup (the manipulators may not be equipped with motors able to generate a force), but in most cases it is due to the lack of a clear understanding of the role of haptic feedback in the training process. Purpose of this work is, therefore, to investigate if the introduction of a haptic interface in the training setup can be beneficial for the trainee and which aspects of the learning process can be improved by such a feature.

1.2 Motivation

A single surgical robot on the market, the *Senhance*®(Asensus Surgical Inc.), is equipped with a force-torque sensor exploited for providing haptic feedback to the surgeon at the console: the sensor measures the forces and torques received by the end-effector when it comes in contact with the tissues and organs in the operatory space, which are then re-created in real-time by actuators integrated to the manipulators at the teleopereation station. As beneficial and effective as it is, it is not an assistance strategy since it cannot be used to re-direct the surgeon's movement towards targets or away from obstacles.

Moreover, none of the surgical simulators on the market include haptics as a tool for error correction and enhanced learning. A few research projects have developed highly specific applications where mechanical feedback was applied for training purposes, but none of them evaluates haptic assistance in the context of a generic surgical training program. The lack of a clear understanding of this concept is what motivates this project toward the implementation and assessment of a haptic interface in a surgical training setup where a VR simulator is built *ad-hoc* and from scratch.

Chapter 2

State of the Art

2.1 The *da Vinci*® Surgical System

Since the FDA approval received by Intuitive Surgical Inc. (Sunnyvale, CA, USA) in 2000, the surgical robotics market has been dominated by the *da Vinci*® surgical system and its multiple evolutions (*da Vinci X*, *da Vinci Xi*, *da Vinci Si* and most recently the *da Vinci SP*). In 2021 there were more than 6,500 *da Vinci* surgical systems installed in 67 countries, and more than 55000 surgeons worldwide have trained on the use of *da Vinci* systems [4]. The numerous advantages in safety, non-invasiveness, precision and dexterity were the key factors that made this technology so disruptive, as well as the role that this surgical robot was designed to undertake in the OR: it is, as a matter of fact, a tool for the surgeon to exploit to enhance his performance, and not a replacement for the surgeon himself.

Three distinct components make up a *da Vinci* surgical system:

- The **Surgeon Console** accommodates the practitioner and acts as the primary interface between him and the robot. Seated at the console the surgeon grips the Master Tool Manipulators (MTMs) to control the motion of the robot arms, sees in 3D the surgical scene by looking into the High-Resolution Stereo Viewer (HRSV) and presses foot-switches to change specific settings (Fig. 2.1 left and Fig. 2.2).
- The **Patient Cart** is positioned next to the patient, and it mounts the Patient-Side Manipulators (PSMs) and the Endoscope Camera Manipulator (ECM). The PSMs are the robot arms that are used to manipulate the surgical instruments, while the ECM is used to control the stereo endoscope camera (Fig. 2.1 right and Fig. 2.4)
- The **Vision Cart** handles the communication between all the hardware components,

and it is responsible for the image processing and the motion planning of the robot arms (Fig. 2.1 center).



Figure 2.1: The macro constituents of a *daVinci*® surgical system in the operatory room: the Surgeon Console (left), the Vision Cart (middle) and the Patient Cart (right).

2.1.1 The Surgeon Console

The console is the main input-output interface between the surgeon and the robotic arms. The surgeon, in fact, holds the grippers of the MTMs (one for the right hand and one for the left hand) and moves his hands, wrists and fingers. Each MTM is a 8 degrees-of-freedom serial kinematic chain: by knowing the coordinate of each rotational joint in the chain, the position and orientation of the gripper in the cart reference frame is known through forward kinematics. A detailed kinematics analysis of the *daVinci*® was conducted in [5], and the Denavit-Hartenberg (DH) parameter for FK and IK are determined aswell.

The pose of the MTMs grippers with respect to the console reference frame will then be used to compute the desired position of the end-effector of the PSMs, and inherently the joint coordinates through an inverse kinematics algorithm.

Seated at the console, the practitioner feels immersed into the surgical scene thanks to the 3D viewing capability of the HRSV. This pair of oculars is directly connected to the pair of cameras situated on the ECM: the disparity between the images displayed in the left and right lenses gives the surgeon the perception of depth, ultimately rendering the surgical scene three-dimensional.

Finally, the operator has also available a set of foot switches that may be pressed without

removing the hands from the manipulators. The switches clutch the system (for repositioning purposes), change the control from the PSMs to the ECM (to achieve a different viewing point of the surgical scene) and energize the mono-polar or bi-polar electrosurgical instruments whether they are mounted.

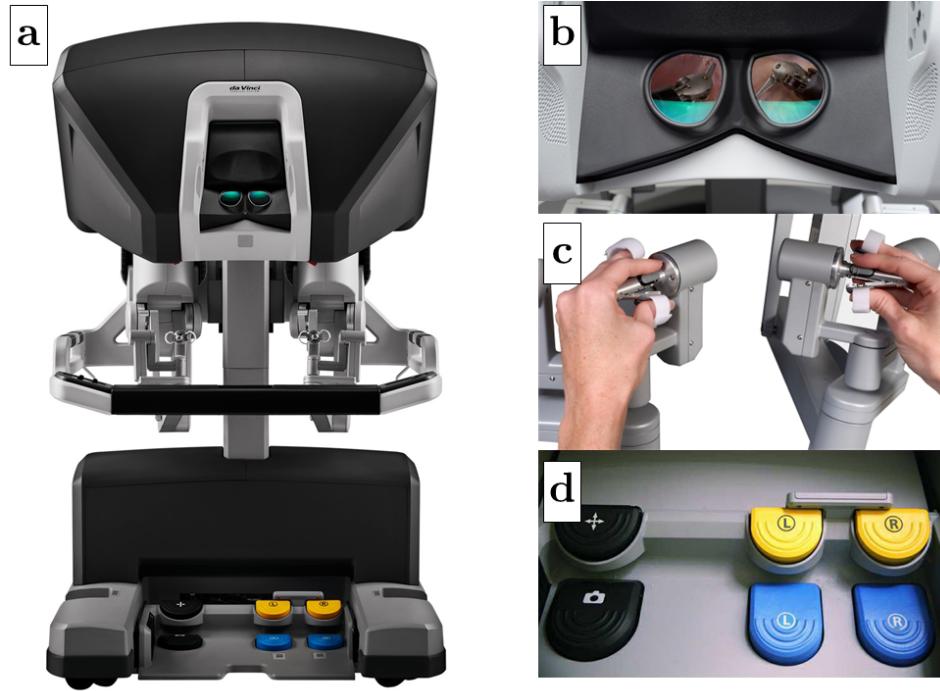


Figure 2.2: a. The *daVinci*® surgeon console; b. The HRSV oculars; c. The MTMs; d. The foot switches

2.1.2 The Patient Cart

A robot-assisted surgery with the *daVinci*® is set up by positioning the patient cart in proximity of the surgical table: the relative position of the two depends on the type of operation to be executed. The cart is the frame of allocation for the PSMs and the ECM: the ECM is the kinematic structure that controls the pose of the stereo endoscope camera, while the PSMs are the robot arms that are used to position and rotate the surgical instruments. A standard *daVinci*® robot mounts four PSMs, controllable by two surgeons independently seated at two different consoles; however, a single surgeon can teleoperate a two-PSMs-robot by himself as it's the case for this study. The position of both PSMs and the ECM relative to the frame of the patient cart can be further adjusted to optimize their pose with respect to the surgical site and the type of operation to be performed: a set of 6 Set-Up Joints (SUJs) can be used to position in space the base of each of the PSMs and ECM. The SUJ kinematic chain consists of 1 prismatic joint and 5 revolute joints moved manually by the surgeon or the OR nurse during the pre-operative phase.

Patient-Side Manipulators Each PSM is a 7-DOF actuated arm (5 revolute joints, 1 prismatic joint, and the gripper angle of opening), which moves a surgical instrument about a Remote Center of Motion (RCM), *i.e.* a fixed fulcrum point that is invariant to the configuration of the PSM joints and the position of which is dependent only on the base link reference frame of the arm. When the surgeon or the OR nurse manually positions the SUJs, ultimately he is positioning the RCM in space: because the RCM is a fixed fulcrum point, its position will not change while the PSM is moving. This feature is what makes the *daVinci®* a truly minimally invasive technology, since if the RCM is positioned exactly at the surgical incision on the patient’s body, the surgeon has a wide range of motion that will still not require a larger opening on the patient’s skin.



Figure 2.3: A visual representation of the concept of Remote Center of Motion (RCM). Independently by the configuration of the PSM, the RCM is always located at the same point in space.

Intuitive Surgical Inc. makes available a variety of surgical instruments that can be mounted on the PSMs and that cover a vast range of use cases, from tissue manipulation to suturing to cauterization. An attachment mechanism unique for all instruments allows one to quickly swap them even during the procedure.

Each of the available instruments designed by Intuitive Surgical exploits the *EndoWrist®* technology: a set of minuscule pulleys and cable allows the instruments’ tooltips to achieve greater dexterity and range-of-motion when compared to the human wrist. This factor highly contributes to the minimally invasive nature of this surgical robot, as it permits achieving articulate and complex orientations that would require a larger incision if performed with a conventional approach.

Endoscope Camera Manipulator One of the robotic arms mounted on the patient cart holds the stereo endoscope camera and determines its pose in space. Naturally, the position and orientation of the endoscope camera determine how the surgical scene is framed and, ultimately, what the surgeon sees in the HRSV. Contrary to a standard laparoscopic approach

where the endoscope placement is a responsibility of the assistant surgeon, with a *daVinci*[®] robot it's the operating surgeon himself that controls the position of the camera. As a matter of fact, when the operator at the console desires to change the viewing point of the surgical scene, he switches the control from the PSMs to the ECM by pressing one of the foot pedals. While that pedal is held down, the control signal from the MTMs is sent to the ECM instead of the PSMs, which moves accordingly.

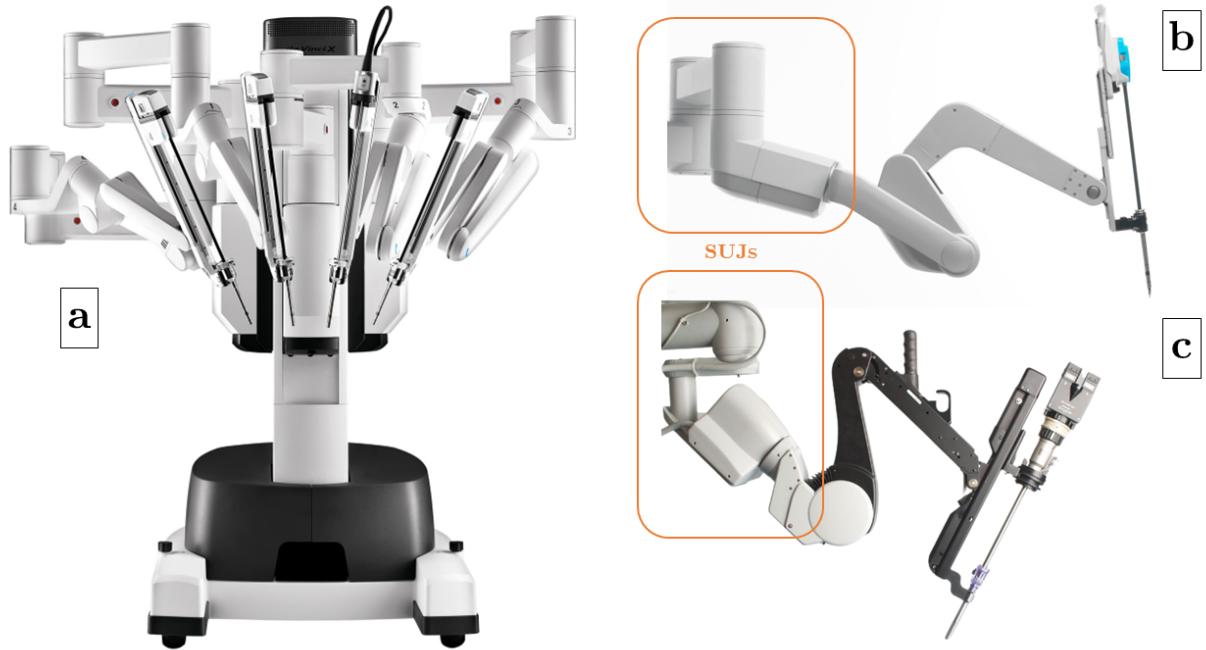


Figure 2.4: a. The patient cart of the *daVinci*[®] robot; b. A PSM; c. The ECM; The SUJs of the PSM and ECM are in the orange boxes

2.1.3 The Vision Cart

The computational, control and communication hardware is hosted on the vision cart, which acts as the main interface between the components. It is also responsible for computing FK and IK and for powering the motors, sensors and endoscope camera.

2.2 VR Surgical Simulators

Aghazadeh *et al.* [6] demonstrated a statistical correlation between robotic performance in a simulated environment and robotic clinical performance, effectively showcasing the potential of VR simulators for surgical training and their comparability to the standard dry-lab phantom training. Most of the surgical robotics companies on the market have developed

and marketed their own general-purpose VR simulator, and a lot of research projects have been conducted to develop simulators for specific robotic platforms or for testing specialized applications.

Intuitive Surgical commercializes the *daVinci SimNow* virtual surgical simulator (previously *dVSS*, *daVinci Skill Simulator*) as a skill-building framework for learning how to safely and successfully operate its surgical robot. It offers a realistic simulated environment comprising 47 skill exercises and 33 virtual surgical procedures, with performance tracking and procedure-specific feedback.

Other relevant simulators on the market [7] [8] are the *Mimic dV-Trainer* (dV-Trainer; Mimic Technologies, Inc, Seattle, WA, USA), the *Robotic Surgical Simulator* (RoSS; Simulated Surgical Systems, Buffalo, NY, USA), the *Sim-Surgery Educational Platform* (SEP, Sim-Surgery, Norway), and the *PromIS* hybrid simulator (*Canadian Aviation Electronics Healthcare, Canada*).

In terms of augmenting the surgical training, visual assistance algorithms are present in most of the listed simulators, but haptic guidance is missing in all cases.

An anticipation of the role of virtual reality environments in surgery was published in the literature as early as 2001 [9], while the very first research work on the topic developing and validating a VR simulator interfaceable with the *daVinci®* appeared in 2008 [10]. From there, several studies and advancements have been published in literature [11]: initially, most of the work was focused on validating and comparing the simulators on the market, assessing their role in the acquisition of surgical skills in general [12] or for specific surgical fields [13]. Later, the focus shifted towards the development of new simulators and the improvement of existing ones, with the goal of making them more realistic, effective and affordable [14].

In 2014 Smith *et al.* established and summarized, upon a 14-society consensus, a set of basic robotic surgery skills that every training curriculum should include [15]. The authors also proposed a set of guidelines for the development of a robotic surgical training curriculum, which is still valid today. This skillset is featuring pre, intra and post operatory skills: VR simulators only cover intraoperative skills, but do so in quite a comprehensive and adaptive way. Surgical simulators can be, indeed, built and developed to train the surgeon in intensively improving a single one of such skills, or to train him in a more general way, covering all of them. Most of the VR simulators developed in literature belong to the former category, and so do the assistive algorithms that they feature.

2.3 Virtual Fixtures and Augmented Training

The very first example of a Virtual Fixture (VF) algorithm was implemented on a PUMA 560 robot and consisted of a real-time collision avoidance algorithm [16]. Since then, the concept of VF has been extended to a wide range of robotic applications, including surgical robotics.

The first definition of Virtual Fixtures was proposed by Louis B. Rosenberg, who proposed the definition of Virtual Fixtures as “abstract sensory information overlaid on top of reflected sensory feedback from a remote environment” [17]. More recently, Bowyer *et al.* [18] provided an extensive review that included most of the existing VF algorithms and proposed a taxonomy of the different approaches. Neither [17] nor [18], however, analyzed the context of surgical robotics or the role of VF in the training of surgeons, but rather circumscribed the topic in the broader field of *telemanipulation*.

One of the earliest research work investigating the role of VFs for surgical training application concluded that the introduction of haptic guidance into a training protocol increases the reusability of paths generated with manual control [19]. Similar positive results were obtained in the assessment of the benefits of a Guidance *Trusted-Region* VF in laparoscopic training [20], which was confirmed to improve the training process both in a real and in a simulated context. An Obstacle Avoidance (or Forbidden Region) Active Constraint was similarly shown to improve the training process in a simulated context [21].

Chapter 3

Materials and Methods

3.1 Equipment

This research has been conducted on a first-generation *daVinci*® surgical system decommissioned in 2016 and equipped with the dVRK (*daVinci*® Research Kit) framework. The dVRK [22] is an open-source mechatronics system, consisting of electronics, firmware, and software, that is being used to control research systems based on the first-generation *daVinci*® systems. Based on a ROS [23] framework, the dVRK implements high-level accessibility to the sensors, actuators and control algorithms of the *daVinci*® robot, making it more easily interfaceable with advanced strategies and algorithms developed in the most diverse software environments.

The simulator developed for this project renders necessary only the surgeon console, as the ROS messages are sent solely to the virtual surgical scene and not to the physical robot. However, all features of the PSMs and of the ECM are implemented: the HRSV shows the 3D virtual surgical scene, the MTMs correctly controls the virtual surgical tools, and the clutch foot-switch allows proper repositioning maneuvers.

3.2 The Surgical Simulator

The objective of investigating a high-specificity aspect regarding the impact of VFs introduced the necessity of developing an *ad-hoc* surgical simulator with specialized surgical tasks and training exercises: this would allow quantifying surgical performance and monitor training over the key surgical skills as indicated by Smith *et al.* in [15].

The simulator is based on *Unity*, a cross-platform game engine that allows the development

of 3D applications and games. The Unity engine is a powerful tool for the development of virtual environments, as it allows the creation of complex 3D scenes with a high level of realism, and the implementation of complex interactions between the virtual objects and the user. Specifically, the simulator implements gravity, object collisions and manipulation. A 3D model of the *daVinci®* patient cart is present in the simulator and responds in real-time to the ROS messages received from the console, therefore the virtual PSMs replicate the motion of the real ones.

Two virtual cameras are positioned in the Unity scene on the tip of the endoscope mounted on the ECM: the horizontal distance between the two virtual cameras (5.3mm) matches the one of the real endoscope, as does the Field-of-View (80deg²). The feeds of the virtual cameras, rendering the 3D scene in real-time, are sent separately to the two oculars of the HRSV. The slightly different images from the left and right eye yield the sensation of depth perception and allow the user to perceive the virtual scene in three dimensions, as it happens when teleoperating with the real robot.

The training surgeon interacts with the console in the same exact way as he would when teleoperating in the OR: he views the surgical scene in the oculars, the virtual instruments respond in real-time to the movements of the manipulators and with the same kinematics, and the 3D objects behave in the same way as they would if they were real thanks to the simulated physics computed by the Unity Engine.

3.2.1 The Surgical Tasks

The simulator comprises eight surgical tasks, four of which (*Path*, *Rings*, *Pillars* and *Exchange*) are simplistic training tasks built with objects of simple geometry, while the remaining four (*Liver Resection*, *Nephrectomy*, *Thymectomy* and *Suturing*) emulate *in-vivo* surgical procedures and are therefore more realistic. Fig. 3.1 collects snapshots of the tasks. All of these are constructed and set up in order to be as challenging as possible in relation to a specific surgical skill. Specifically:

- *Path* and *Liver Resection* require articulate wrist motion and stability
- *Rings* and *Nephrectomy* survey the depth perception skills
- *Pillars* and *Thymectomy* are hand-eye coordination tasks
- *Exchange* and *Suturing*, both bi-manual tasks, challenge the capabilities in terms of instrument exchange

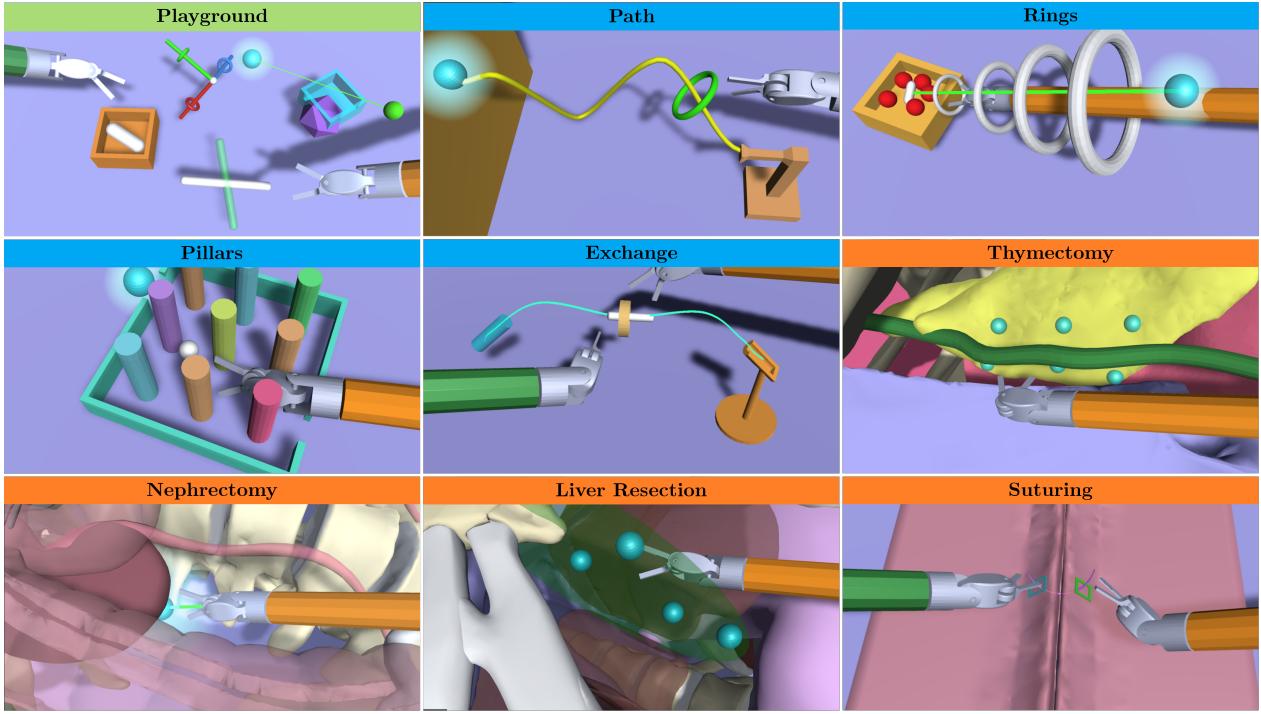


Figure 3.1: Snapshot of the simulated surgical tasks, with the respective denomination. Training tasks have blue headlines, while realistic evaluation tasks have orange headlines. *Playground* is a propaedeutic task and isn't featured in the experimental study

Path Objective of this task is to grab a torus-like object and carry it along a reference trajectory. To achieve this, discrete wrist articulation and motion smoothness are required. The reference trajectory is set up in order to require a wrist rotation of at least 90deg around 2 perpendicular axes, the one pointing from right to left and the one pointing forward with respect to the camera view.

Rings In this task, the operator is required to precisely insert the instrument inside a narrow constrained space defined by a set of increasingly smaller rings, grab a target object and carry it out of the constrained space. The task tests the depth perception skills of the operator, as the rings are positioned at an angle with respect to the camera view.

Pillars This task test the trainee's steady-hand skills, as well as hand-eye coordination. While carrying an object through a set of obstacles that most times obscure the free path, the operator is also required to extrapolate the path that he must take to reach the intended target.

Exchange To test the bi-manual coordination and the ability to exchange an object from one hand to another, this task demands the operator to carry a cylinder-shaped object from

a starting position to a target location, exchanging it from the right to the left hand at the mid-point of the path.

Thymectomy In a surgical thymectomy, a lobe of the thymus gland is cut and removed. The vicinity of the so delicate phrenic nerve requires the utmost care and precision and, in most cases, a minimally-invasive approach is, indeed, preferred. In the emulated version of this task, the surgeon shall pinch a set of 6 targets located on the surface of the virtual thymus while staying as far as possible from the phrenic nerve.

Nephrectomy A careful insertion of the surgical instrument inside the abdominal cavity is required to perform a nephrectomy. The insertion is often performed at a steep angle with respect to the camera view, and the surgeon must be able to perceive the depth of the target object. This virtual nephrectomy re-creates both of these aspects.

Liver Resection The surgeon must perform a series of cuts along the surface between two lobes of the liver to perform a successful liver resection. Naturally, to minimize tissue damage, the instrument's tooltip shall remain as close as possible to the surface between the two lobes to be separated.

Suturing Suturing with a recurve circular needle is a challenging task requiring the medical doctor to approach the tissue at an optimal angle and to properly exchange the needle from one hand to the other.

Playground This propaedeutic task allows the trainee to familiarize himself with the simulator and the virtual surgical tools. A few simple objects are scattered Win the scene and the trainee is free to interact with them. The scope of this task is to better interface a novice user with teleoperation and the VR environment, understanding how its movements are mapped to the motion of the instruments, how to perform a pinching action and what intensity of force and torque to expect when VFs are applied.

3.3 Virtual Fixtures

The tasks as described in the previous paragraph are equipped, in the virtual environment, with high-level assistance strategies that act applying mechanical forces to the MTMs, with the aim of re-directing the motion of the surgeon's arm towards intended targets or away from obstacles. The MTMs are, indeed, 8-DOFs robotic actuated arms (the last degree of

freedom is not actuated and controls the opening and closing of the gripper with a magnetic Hall sensor): each of the 7 rotational joints is equipped with both an encoder for sensing purposes and with a motor for actuation purposes. The encoders determine the joint angles that will ultimately be used for estimating the pose of the gripper in the console's space, through FK; it's this pose, transformed with respect to the HRSV Reference Frame (RF), that the PSMs are actuated to recreate, with respect to the endoscope RF, as in Fig. 3.2.

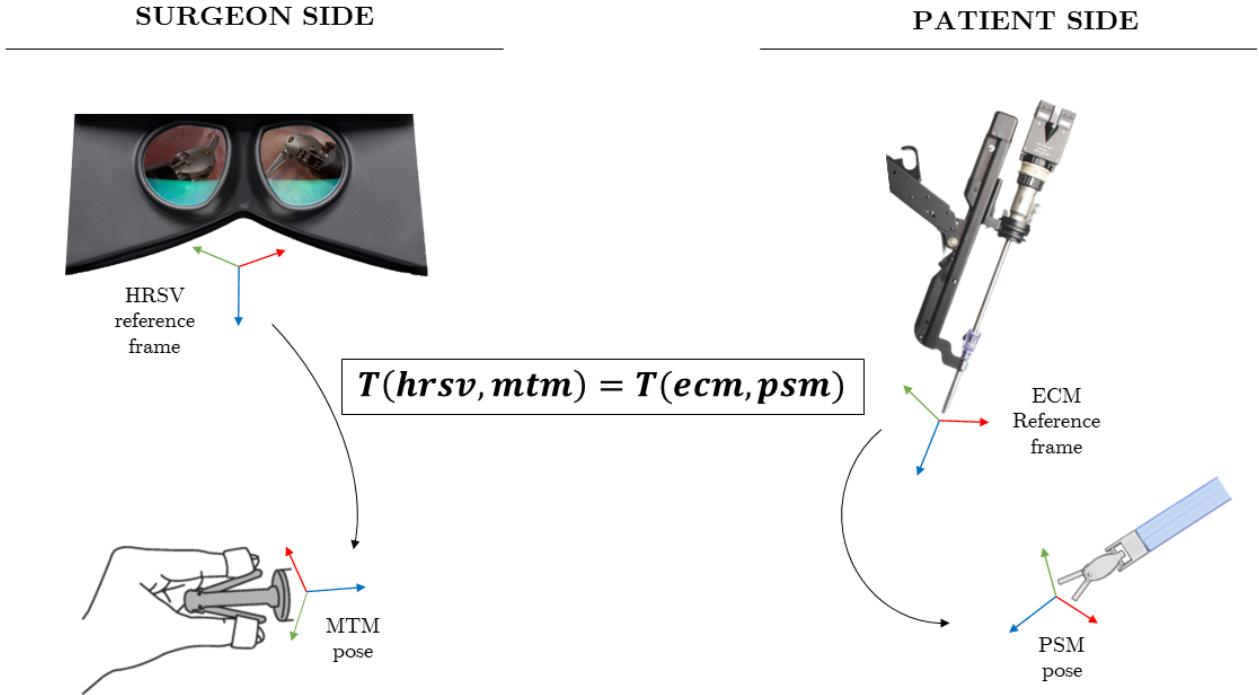


Figure 3.2: The equivalence of transformations between the console space and the surgical space. The transformation matrix that expresses the pose of the manipulator in the HRSV reference frame is used to compute the desired pose of the PSM in the RF of the endoscope camera.

In the real surgical context, the motors are used to automatically position the manipulators at the start of the procedure, for gravity compensation and for exploiting the redundancy of the kinematic chain to optimally position the links of the MTM in order to avoid collisions with the surgeon's wrist. However, the same dynamic model used for these purposes can be exploited to apply assistive mechanical forces to the manipulators, the direction and magnitude of which are determined from the pose of the PSMs's tooltip in the surgical RF. A force to be applied to the manipulator tooltip is converted to the set of torques to be applied at each joint by the respective actuators. The parameters of this inverse dynamics model [5] for the generic *i-th* link are:

- The mass m_i of the link
- The three components of the first moment \mathbf{m}_i of the link

- The six independent elements of the inertia tensor \mathbf{I}_i of the link
- The static and viscous coefficients of the link, $F_{s,i}$ and $F_{v,i}$ respectively

The VF force computed in the virtual surgical space - in the simulator - is therefore converted, in real-time, to a set of 7 torques to be communicated as a ROS message to the MTMs. The same inverse dynamics model converts a torque to be applied to the end effector into a set of 7 torques to be applied to the joint motors.

3.3.1 Error Mapping

Most of the assistance strategies implemented here will use the distance from the PSM to the target or obstacle as the primary metric for determining the intensity of the feedback force or torque. However, different surgical tasks and situations require a level of control over how the distance is taken into account. For this reason a sigmoidal mapping function is employed for the normalization of the linear or angular error into a suitable interval. Specifically, such mapping is formulated as:

$$f_{map}(x) = \frac{1}{1 + e^{5\delta w(x-t-h)}} \quad (3.1)$$

with $\delta = +1$ for guidance VFs and $\delta = -1$ for avoidance VFs (δ determines if the function increases or decreases monotonically). Here:

- t is the fixture *threshold*, hence the value at which the sigmoid starts to significantly increase from zero
- h is the distance from the threshold at which *half* of the maximum force is provided
- w controls the *width* of the linear region, hence the steepness of the curve

For example, if $t = 2\text{mm}$ and $h = 3\text{mm}$ the surgeon will start to feel a force for errors higher than 2mm, and at 5mm he will experience half of the maximum force that can be delivered.

t , h and w are set manually in order to achieve an optimal "feeling" of the VF. Their values, different from task to task, are reported in Tab.3.1.

3.3.2 Virtual Fixtures Algorithms

As stated previously the force feedback delivered at the level of MTM is computed, in terms of magnitude and direction, from the surgical space and therefore based on the relative position and orientation of the PSM's tooltip with respect to the objects in the virtual surgical scene.

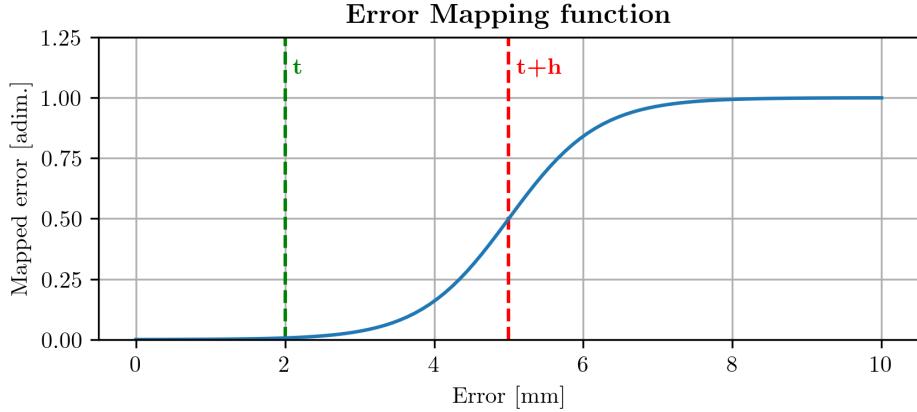


Figure 3.3: Plot of the Error Mapping function. The position of t and $t + h$ can be set manually to achieve a suitable behavior of the assistance strategy. The x-axis refers to a generic error shown in millimeters, but the same mapping function can be applied to an angular error.

Table 3.1: Error mapping parameters for the different tasks. *Exchange*, *Liver Resection* and *Suturing* map both the distance error and the angular error, so the values of both mapping functions are reported.

Task	t	h	w
Path	2 mm	2 mm	500
Rings	2 mm	2 mm	500
Pillars	0.5 mm	1 mm	1000
Exchange (distance)	3 mm	2 mm	500
Exchange (angular)	5°	5°	2
Thymectomy	0.5 mm	1 mm	1000
Nephrectomy	2 mm	2 mm	500
Liver Resection (distance)	2 mm	5 mm	200
Liver Resection (angular)	5°	15°	1
Suturing (distance)	1 mm	3 mm	300
Suturing (angular)	5°	5°	2

The simulator features 4 assistance algorithms implementing 4 different declinations of haptic assistance. Different algorithms are deployed inside specific surgical tasks based on the task morphology and goals:

- **Trajectory Guidance:** This algorithm outputs a force vector that assists the surgeon in following a pre-defined 3D trajectory. A visco-elastic force pulls the End-Effect (EE) tooltip towards the closest point of the trajectory, and a visco-elastic torque aligns the tooltip's orientation with the trajectory tangent vector, computed at the closest point.
- **Obstacle Avoidance:** An Obstacle Avoidance algorithm prevents the practitioner from colliding with the virtual objects in the scene, which may represent, for example,

Table 3.2: The surgical tasks featured in the simulator and the virtual fixtures algorithms used for each of them. Although the modular implementation of the simulator allows to include multiple VF algorithms in a single task (multiple-steps executions will be included in the future), for this study only one VF algorithm is used per task.

Path	<i>Trajectory Guidance</i>
Rings	<i>Insertion Guidance</i>
Pillars	<i>Obstacle Avoidance</i>
Exchange	<i>Trajectory Guidance</i>
Thymectomy	<i>Obstacle Avoidance</i>
Nephrectomy	<i>Insertion Guidance</i>
Liver Resection	<i>Surface Guidance</i>
Suturing	<i>Trajectory Guidance</i>

delicate anatomical structures that must not be touched during surgery. A visco-elastic force is, therefore, applied to the EE tooltip in order to push it away from the closest point of the obstacle.

- **Insertion Guidance:** Some surgical tasks require the insertion of the instrument's tooltip into a narrow space. This algorithm, from an initial position of the PSM and a target position, aids the surgeon in approaching the target on an optimal insertion path, without deviating from it.
- **Surface Guidance:** After defining, in the pre-operative stage, an ideal surface of operation (the surface can be of any morphology and should be defined as a mesh of points), a Surface Guidance algorithm generates forces and torques that keep the surgical tool on such surface and with an orientation tangent to it.

Tab. 3.2 highlights which of these VF algorithms are used in each of the task featured in the simulator. Fig. 3.4 graphically illustrates the VF algorithms as described above, but a detailed analytical formulation of each algorithm follows in the next sections.

Trajectory Guidance Considering a generic three-dimensional reference trajectory planned in the pre-operative stage, the feedback forces will be calculated according to the relative position and orientation of the trajectory itself and the tooltip reference frame. Both the forces and torques are computed in real-time as the sum of an elastic and a viscous contribution: while the elastic component accounts for the positional or angular error to the closest point of the reference trajectory - and the tangent computed in such point, accordingly - the viscous components are proportional to the temporal rate of change of the errors themselves.

A viscoelastic model allows to achieve a guidance and error-compensating assistance that is less prone to overshooting behaviors and to oscillations. Moreover, with the force and torque

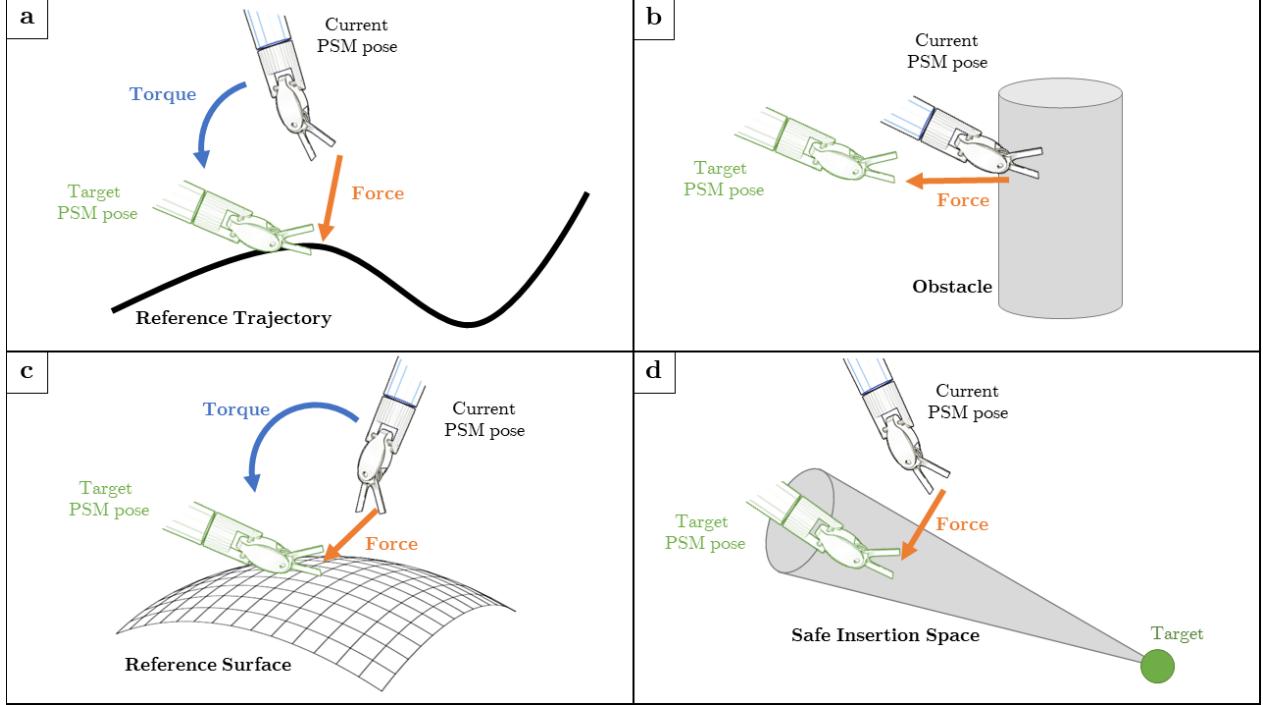


Figure 3.4: Graphics scheme of the 4 virtual fixtures featured as assistance strategies in the surgical simulator. **a.** Trajectory Guidance; **b.** Obstacle Avoidance; **c.** Surface Guidance; **d.** Insertion Guidance. A representative PSM's pose is shown, and a target pose is also depicted in a green hue, together with the force (orange arrows) and torque (blue arrows) that will guide the motion toward the target pose.

calculated as

$$\mathbf{F} = k_F \cdot \mathbf{F}_{elastic} + \eta_F \cdot \mathbf{F}_{viscous} \quad (3.2)$$

$$\mathbf{T} = k_T \cdot \mathbf{T}_{elastic} + \eta_T \cdot \mathbf{T}_{viscous} \quad (3.3)$$

one tunes the elastic gains k_F and k_T and the viscous damping coefficients η_F and η_T in order to achieve a comfortable balance between the components and a stable behavior of the feedback force, which may vary from operator to operator, as well as from task to task.

Fig. 3.5 illustrates the vectors involved in the computation of the virtual fixture; specifically, contributions in Eq. 3.2 expanded as:

$$\mathbf{F}_{elastic} = f_{map}(\|\mathbf{d}\|) \cdot \frac{\mathbf{d}}{\|\mathbf{d}\|} \quad (3.4)$$

$$\mathbf{F}_{viscous} = \begin{cases} \mathbf{d}, & \text{if } \mathbf{v} \cdot \mathbf{d} < 0 \\ \text{rotate}(\mathbf{v}, \theta, \mathbf{r}), & \text{otherwise} \end{cases} \quad (3.5)$$

Here, \mathbf{d} is the distance vector going from the surgical instrument to the closest point in the trajectory, \mathbf{v} is the velocity of the surgical instrument, while θ and \mathbf{r} are the angle and axis

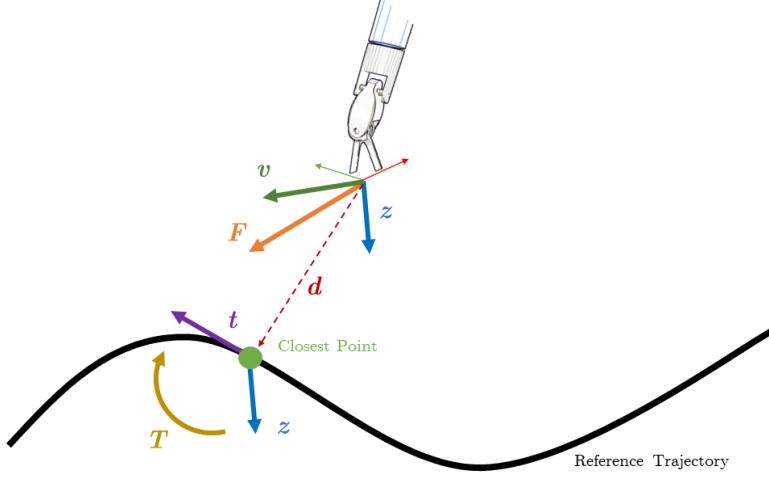


Figure 3.5: Vectors involved in the computation of the Trajectory Guidance VF. The PSM is shown in an example representative pose with respect to the reference trajectory.

of rotation which will align the velocity vector \mathbf{v} with \mathbf{d} , respectively:

$$\theta = (1 + \mathbf{v} \cdot \mathbf{d}) \cdot \frac{\pi}{2} \quad (3.6)$$

$$\mathbf{r} = \mathbf{v} \times \mathbf{d} \quad (3.7)$$

This implementation is adapted from [24]. Similarly, contributions to the torque (Eq. 3.3) are expanded as:

$$\mathbf{T}_{elastic} = \arccos(\mathbf{z} \cdot \mathbf{t}) \cdot \mathbf{z} \times \mathbf{t} \quad (3.8)$$

$$\mathbf{T}_{viscous} = \frac{d}{dt} [\arccos(\mathbf{z} \cdot \mathbf{t})] \cdot \mathbf{z} \times \mathbf{t} \quad (3.9)$$

The role of the torque is to align the z-axis of the surgical tool’s reference frame with the tangent of the trajectory \mathbf{t} at its closest point. In Eq. 3.8 and Eq. 3.9, the angle and axis of rotation which will achieve this alignment are $\arccos(\mathbf{z} \cdot \mathbf{t})$ and $\mathbf{z} \times \mathbf{t}$, respectively.

Obstacle Avoidance The format for representing 3D objects in the simulator developed for this work is **stl** (Standard Triangle Language): in this format, a solid is defined as a list of 3D spatial coordinates of vertex points and a list of edges connecting them in groups of 3. Under a different light, an **stl** solid is a group of triangles, or “subsurfaces”, sharing sides: the **stl** format also embeds a list of unique vector normals, one for each triangle, useful for easily accessing the orientation of each. The closest vertex point to the PSM’s tooltip is identified and, with \mathbf{d} as the normalized distance vector going from such point to the tooltip and \mathbf{n} as the normal vector of the object mesh at that point (a graphic representation

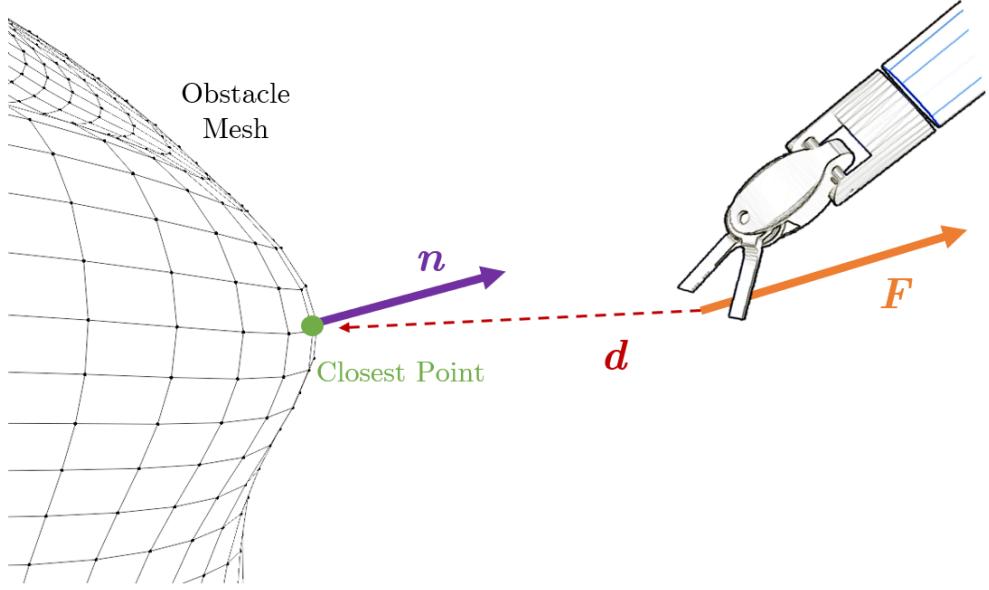


Figure 3.6: Vectors involved in the computation of the Obstacle Avoidance VF. The PSM is shown in an example representative pose with respect to the obstacle mesh.

is depicted in Fig. 3.6), the force is again computed as the sum of an elastic component, modulated by k , and a viscous one, modulated by η :

$$\mathbf{F} = \left[k \cdot f_{map}(\|\mathbf{d}\|) + \eta \cdot \frac{d}{dt} (f_{map}(\|\mathbf{d}\|)) \right] \cdot \frac{\mathbf{n}}{\|\mathbf{n}\|} \quad (3.10)$$

k and η can be adjusted while performing the task and the visco-elastic balance will change accordingly, in real-time. Here, the addition of viscous component was suggested by clinical experts, as the elastic-only model was felt “jelly-like” and, in a few cases, oscillating. $f_{map}(\cdot)$ is the error mapping function with $\delta = -1$. With the addition of the viscous component, the same clinicians confirmed a more appropriate behavior of the assistance force.

It is to be highlighted that, for meshes with particularly sharp edges, the normal vector \mathbf{n} at the closest point may not be optimally defined: in Fig. 3.7 (left), the closest point normals points upwards, which is not necessarily the motion direction that maximizes the distance with the PSM’s tooltip. To solve this, instead of defining the normal vectors at the centerpoint of each subsurface, the normals taken into consideration are defined at the vertex points and are computed as the average of the normals of the subsurfaces that share the vertex point. This is illustrated in Fig. 3.7 (right), where it’s also evident how the direction of the force generated in this case actually maximizes the distance with the EE. With this adjustment, the spatial distribution of normal vectors is smoothed and so is the force field generated by the obstacle avoidance module.

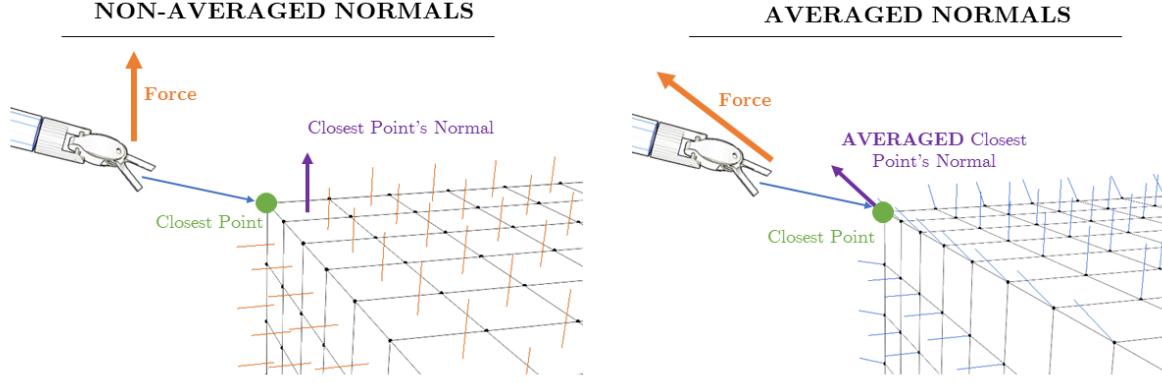


Figure 3.7: On the left, the assistance force is generated from the non-averaged normals, defined at the subsurface centerpoints. On the right, the assistance force is generated from the averaged normals, defined at the vertices. In this figure, the mesh is composed of 4-vertices “quads”, but the same concept holds to meshes with subsurface triangles.

Numerically, if a vertex is shared among S subsurfaces, the averaged normal vector $\hat{\mathbf{n}}$ at that vertex is computed as:

$$\hat{\mathbf{n}} = \frac{1}{S} \sum_{i=1}^S \mathbf{n}_i \quad (3.11)$$

Surface Guidance Similarly to the Obstacle Avoidance algorithm, an **stl** surface may be used in some context as a reference for guidance: this implementation indeed computes a force and a torque that attracts and aligns the surgical EE towards the surface itself.

The force is computed with the same logic as in the Obstacle Avoidance algorithm, with the distance vector \mathbf{d} from the PSM tooltip to the closest point of the surface mesh, and \mathbf{n} the normal vector of the surface at that point. However, in the error mapping function $f_{map}(\cdot)$ the value of δ is set to $+1$ and the curve increases monotonically, so that the force is always attractive and is highest when the PSM is furthest from the surface mesh. The formula for the force is:

$$\mathbf{F} = \left[k_F \cdot f_{map}(\|\mathbf{d}\|) + \eta_F \cdot \frac{d}{dt} (f_{map}(\|\mathbf{d}\|)) \right] \cdot \frac{-\mathbf{n}}{\|\mathbf{n}\|} \quad (3.12)$$

Compared to Eq. 3.10, the sign of the normal vector \mathbf{n} is inverted.

To compute the torque, one considers the relative orientation of the \mathbf{z} -axis of the PSM’s tooltip reference frame and the normal vector \mathbf{n} at the closest point on the surface. In particular, the torque generated on the manipulator will aim at aligning the \mathbf{z} -axis with its projection on the tangent plane of the surface, defined by the normal vector \mathbf{n} at the closest

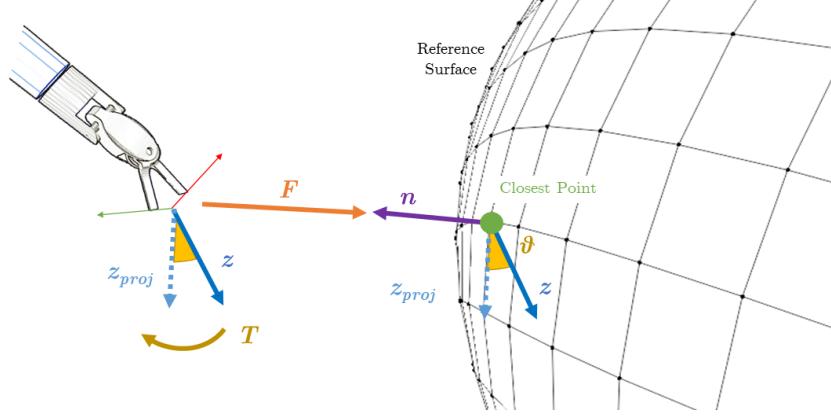


Figure 3.8: Vectors involved in the computation of the Surface Guidance VF. The PSM is shown in an example representative pose with respect to the surface mesh.

point. The vector projection on the tangent plane is the difference between the vector itself and its projection on the normal vector \mathbf{n} , obtained from the dot product operator:

$$\mathbf{z}_{proj} = \mathbf{z} - (\mathbf{z} \cdot \mathbf{n})\mathbf{n} \quad (3.13)$$

From this, the torque is again the sum of an elastic component dependent on the angle θ between \mathbf{z} and \mathbf{z}_{proj} (specifically $\theta = \arccos(\mathbf{z} \cdot \mathbf{z}_{proj})$), and a viscous one proportional to the angle's derivative in time. The formula for the torque is

$$\mathbf{T} = \left[k_T \cdot f_{map}(\theta) + \eta_T \cdot \frac{d}{dt}(f_{map}(\theta)) \right] \cdot \frac{\mathbf{z} \times \mathbf{z}_{proj}}{\|\mathbf{z} \times \mathbf{z}_{proj}\|} \quad (3.14)$$

where $\mathbf{z} \times \mathbf{z}_{proj}$ is the axis of rotation.

The surface mesh is submitted to the same smoothing procedure as the obstacle mesh (Fig. 3.7, Eq. 3.11).

Insertion Guidance From a starting pose and a target object it's possible to define an *insertion cone* that defines a safe space for the EE to move while approaching the target object. An insertion angle α determines the conical aperture and, therefore, how narrow the safe space is as the tooltip gets closer to the target. The vector $\boldsymbol{\delta}$ links the initial PSM position to the target.

The position of the EE in its longitudinal and radial components with respect to the reference insertion cone, \mathbf{l} and \mathbf{r} respectively, as follows:

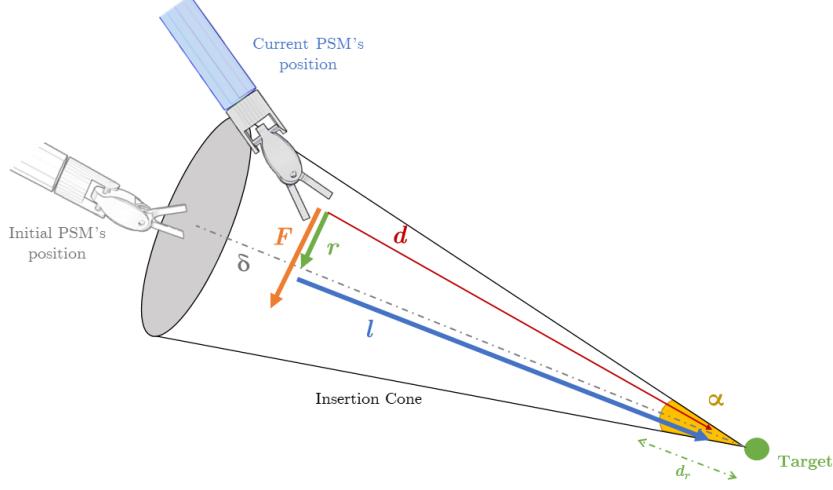


Figure 3.9: Vectors involved in the computation of the Insertion Guidance VF. The PSM is shown in a “current” example representative pose with respect to the insertion cone, which is defined from the initial PSM position and the target position.

$$\mathbf{l} = \left(\mathbf{d} \cdot \frac{\boldsymbol{\delta}}{\|\boldsymbol{\delta}\|} \right) \frac{\boldsymbol{\delta}}{\|\boldsymbol{\delta}\|} \quad (3.15)$$

$$\mathbf{r} = \mathbf{d} - \mathbf{l} \quad (3.16)$$

Fig. 3.9 shows a representative example for the calculations of the Insertion Guidance VF force, and the vectors involved.

For the insertion cone, fixed in space, the aperture a relates the longitudinal and radial components through the angle α (expressed in degrees) as follows:

$$a = \tan \left(\alpha \cdot \frac{\pi}{180} \right) \quad (3.17)$$

$$\|\mathbf{r}\| = \|\mathbf{l}\| \cdot a \quad (3.18)$$

The computed feedback force aims at maintaining the EE position inside the safe insertion cone, therefore the threshold in the error mapping function $f_{map}(\cdot)$ is set to $t = \|\mathbf{r}\|$, the value of which varies over time and depends on the longitudinal coordinate $\|\mathbf{l}\|$. When the tooltip of the surgical instrument is very close to the target, the threshold becomes very small and the force gets very close to the maximum value very rapidly, which causes instabilities. To solve this issue, a “relax distance” d_r is introduced: when the EE is close enough to the target, the force is relaxed and scaled down to half of its value to interfere less with the surgeon’s motion so close to the target. A clinical expert suggested that d_r should be 20%

of the height of the cone. Considering this adjustment, the force feedback for the Insertion Guidance assistance strategy is:

$$\mathbf{F} = r(\|l\|) \left[k \cdot f_{map}(\|\mathbf{a}\|) + \eta \cdot \frac{d}{dt} (f_{map}(\|\mathbf{a}\|)) \right] \cdot \frac{\mathbf{a}}{\|\mathbf{a}\|} \quad (3.19)$$

with

$$r(\|l\|) = \begin{cases} 1 & \text{if } \|l\| > d_r \\ 0.5 & \text{if } \|l\| \leq d_r \end{cases} \quad (3.20)$$

scaling the force down to its relaxed value depending on the insertion depth. As per all the other assistance algorithms, k and η control the PD gains of the force feedback and are set manually by the practitioner, who is also able to adjust them in real-time.

A similar implementation of this VF, from which this algorithm is inspired, was proposed by Bettini *et al.* [25].

3.4 Clinical Validation

Two resident surgeons from the *Istituto Europeo di Oncologia* (Milano, MI, Italy), both regularly performing Robot-Assisted Minimally Invasive Surgery (RAMIS) procedures with the *daVinci®* robot, kindly dedicated their time in testing the surgical simulator in all its aspects, from the motion truthfulness to the complexity of the wrist articulation to the invasiveness and visco-elastic balance of the virtual fixtures. Their opinion and expertise were precious and insightful tools that guided the development towards a clinically validated robotic surgical simulator. Moreover, the most expert resident surgeon allowed to have his performance recorded when practicing with the simulator, which will be considered “peak performance” in the experimental analysis.

3.5 Experimental Protocol

In order to assess the effectiveness and benefits of the developed assistance strategies and the simulator they have been implemented in, an experimental study was conducted on a group of novice users interfacing with the simulated surgical environment.

A group of 8 volunteers was asked to execute the surgical tasks implemented in the simulator

over multiple repetitions: some of the subjects were assisted with the VF algorithms described in the previous paragraphs, and others performed the tasks without any haptic assistance. Considering that the scope is to evaluate the role of haptic assistance in the learning process, the program was carried out over the course of a week in order to appreciate a possible variation in performance, which does not usually occur in the context of learning surgical skills.

The 8 volunteers participating in the study were divided into 2 groups:

- An **Assisted Group** (Group A) of 4 subjects, who received haptic assistance while performing the tasks.
- A **Control Group** (Group C) of 4 subjects, who were not assisted by VFs while teleoperating.

Subjects were 25% females and 75% males, between 23 and 27 years of age, all right-handed and either had never teleoperated a surgical robot or did it less than 5 times. Assignment to the control or assisted group was random.

The subjects underwent a week-long training phase:

- On *Day 1*, they were given a concise explanation about the *daVinci®*surgical system, the simulator, haptics, and the training protocol. Later, each subject experienced 5 minutes in a *Playground* environment (Fig. 3.1), purposely designed to understand the elementary mechanics of teleoperation, clutching and object grasping. No performance metric was recorded at this time
- From *Day 1* to *Day 4*, after familiarizing with in the *Playground*, they were asked to execute the four training tasks of the simulator (*Path*, *Rings*, *Pillars* and *Exchange*), each one for a total of 3 repetitions. Tasks appeared sequentially in a random order, contributing to un-bias the results. Subjects in group A were assisted by the VFs while training, while subjects in group C received no mechanical assistance force. At the beginning of each daily session, subjects could interact with the *Playground* environment for 1 minute at maximum.
- On *Day 5* and *Day 6* subjects were given a break period and no training was performed.
- On *Day 7* they were asked to execute the four surgical evaluation tasks of the simulator (*Thymectomy*, *Nephrectomy*, *Liver Resection* and *Suturing*), each one for a total of 3 repetitions. Again, tasks appeared sequentially in a random order, contributing to unbiasing the results. On this day, neither group A nor group C was assisted by the

EXPERIMENTAL TRAINING PROTOCOL

= With assistance

= Without assistance

= Playground

	Day 1	Day 2 to Day 4	Day 5 and 6	Day 7
Control Group	Playground & Training	Training Tasks	Break	Evaluation Tasks
Assisted Group	Playground & Training	Training Tasks	Break	Evaluation Tasks

Figure 3.10: Schematic of the training protocol applied to the subject undergoing the experimental phase. On the days highlighted in orange, the VF assistance was provided to the group indicated; on the days highlighted in blue, the subjects were not assisted by VFs. In green, the *Playground* environment is a propedeutic task shown on the first day only, to familiarize the subjects with the simulator and the *da Vinci*® system.

VF algorithms.

This training protocol is schematized in Fig. 3.10.

The layout of the training phase was studied and developed with a focus on examining how an enhanced training phase influences performance in a real surgical scenario. This aspect is known as *Skill Transfer* and it is crucial in the development of a surgical simulator which, no matter how realistic and true-to-life it is, will never be able to replicate the complexity of a real surgical environment. To compensate for this, the 4 evaluation tasks executed on the last day were never shown to the subjects during the first 6 days, thus allowing to study if and to what extent the skills learned with the training tasks were transferred to the evaluation tasks. Naturally, to ensure a correct comparison of skill transfer, no assistance was provided to the subjects in the assisted group while performing the evaluation tasks on the last day.

The 2-days break was, instead, introduced to indagate the *Skill Retention* that may be introduced by augmented training: a positive skill retention occurs if the performance drop caused by a break period is reduced compared to normal. In this case, it is of interest to understand if the haptic assistance provided during the training phase causes a decrease in the performance drop across the break period.

3.6 Performance Metrics

The simulator keeps track in real time of the pose of all the objects in the scene, as well as other relevant parameters that can be employed to quantify performance. In order to properly conduct a comparative study between the two groups, the execution of the subjects involved

in the training phase was considered in relation to the execution of the expert clinicians, as in Section 3.4. Their expertise, the number of robotic surgeries in their curricula and their daily usage of a *daVinci*® robot made it possible to consider their execution of the training and evaluation surgical tasks as “peak performance”. Therefore, the quality of the execution of a subject is always considered in relation to the execution of expert clinicians.

A group of *metrics* is recorded at each frame of the simulation and then logged automatically in a .csv file. The simulator begins and ends the metrics acquisition automatically, detecting if the user has begun interacting with the objects and tools as well as if he has reached the target. This level of automatization guarantees that the metrics are logged in the same way for all subjects, and prevents the introduction of human errors when deciding when to start and terminate the logging. The metrics are the following:

- D : Distance error, from the EE to the closest point of the target (trajectory, surface or object)
- A : Angular error, from the EE to tangent vector in the closest point of the reference trajectory or to the tangent plane in the closest point of a surface
- F : Feedback force, computed from the VF. The force is computed also for the subjects in the non-assisted control group, but it's not delivered during the operation
- T : Feedback torque, computed from the VF. The torque is computed also for the subjects in the non-assisted control group, but it's not delivered during the operation
- M : Missed exchanges. When, in the bi-manual tasks, an object is dropped when passing it from the left PSM to the right, or vice versa
- C : Clutch time, the fraction of the total task time during which the clutch pedal was pressed for repositioning

Such parameters are recorded and logged at every frame, and they are averaged at the time of task completion in order to obtain a single mean value that refers to the execution as a whole. Thus, a set of 6 averaged metrics is available for each task execution.

As stated previously, in order to sensibly account a metric as a performance indicator for a comparative study, it must be normalized in relation to the execution of the expert clinicians. This is done by considering a *relative metric* \hat{X} as follows:

$$\hat{X} = \frac{X_{subject}}{X_{expert}} \quad (3.21)$$

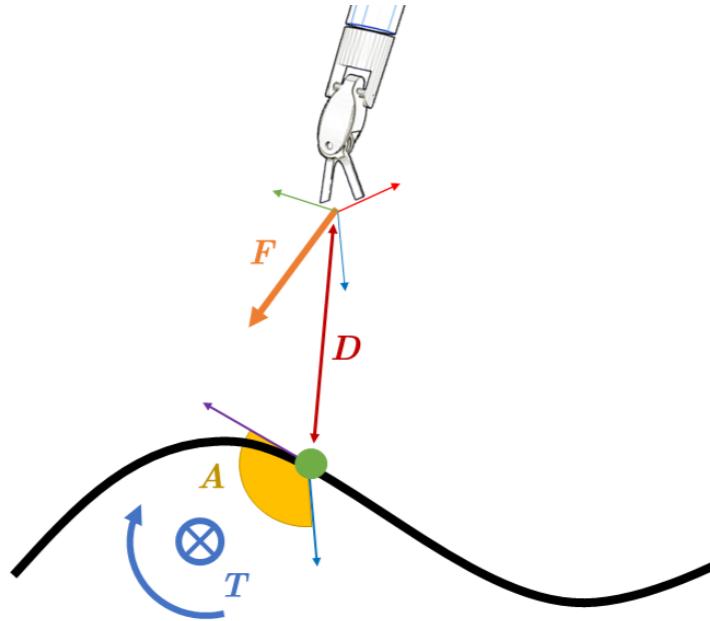


Figure 3.11: Graphical representation of some of the performance metrics recorded during task execution.

Table 3.3: Weights assigned to the metrics in the computation of the performance score. The weight of a generic relative metric \hat{X} is denoted as $w_{\hat{X}}$.

Task	$w_{\hat{D}}$	$w_{\hat{A}}$	$w_{\hat{F}}$	$w_{\hat{T}}$	$w_{\hat{M}}$	$w_{\hat{C}}$
<i>Path</i>	3	2	3	1	0	1
<i>Rings</i>	5	0	4	0	0	1
<i>Pillars</i>	5	0	4	0	0	1
<i>Exchange</i>	2	2	2	1	2	1
<i>Thymectomy</i>	5	0	4	0	0	1
<i>Nephrectomy</i>	5	0	4	0	0	1
<i>Liver Resection</i>	3	2	3	1	0	1
<i>Suturing</i>	2	3	1	2	1	1

The numerical value of performance is then established as a weighted average of the relative metrics. Depending on the surgical skills that each task requires, the weights are determined to reflect a resulting performance that properly accounts for the metrics themselves. The values of the weights in each task are reported in Tab. 3.3

If one defines the set of normalized metrics as $M = \{w_{\hat{D}}, w_{\hat{A}}, w_{\hat{F}}, w_{\hat{T}}, w_{\hat{M}}, w_{\hat{C}}\}$, the numerical formulation for the performance score P is:

$$P = \frac{1}{10} \sum_{m \in M} w_m \cdot \hat{m} \quad (3.22)$$

The value of P gets closer to 1 if the performance of the subject is similar to the performance of the expert clinicians: in a few cases, some subjects recorded metrics that, when averaged, were better than the ones obtained from the execution of the surgeons. For this reason, in a few cases the value of P slightly exceeds 1.

During the experimental phase of this study, all the averaged metrics, the performance scores and the relative task, repetition and subjects were progressively logged and stored in a database: associating the performance to the subject and repetition allows to analyze its variations and trends.

Chapter 4

Results

Fig.4.1 shows the performance trends for the four training tasks (*Path*, *Rings*, *Pillars* and *Exchange*), where the performance at each repetition is the average among the subjects in the assisted or control group. Apart from the *Pillars* task, the trends are increasing both for the assisted group and the unassisted group. Most significantly, the performance in the assisted group is consistently higher than the one in the control group in all training tasks. The performance on the four validation tasks (*Thymectomy*, *Nephrectomy*, *Liver Resection* and *Suturing*) recorded on the last day of the experimental phase is shown in Fig.4.2 with boxplots. For each task, the graph reports the distribution of performances collected from the 4 subjects executing 3 repetitions. Crucially, neither the subjects in the assisted group nor the ones in the control group were guided with VFs on these tasks, nevertheless the performance recorded from the assisted subjects is distributed on higher values for all the tasks.

Quantitative results are reported in the tables below: for each task, the mean, standard deviation and median values of performance are compared between the assisted and the control group. Given the data scarcity and their non-Gaussian distribution, the most meaningful conclusions will be drawn from the median values. Apart from *Nephrectomy* showing a slightly reduced median performance on assisted subject (accompanied, however, by a much larger standard deviation, as per Fig.4.2), all other tasks present an increase in both the mean and median performance, as high as +21.54%.

The standard deviation in the performance isn't consistent when comparing the assisted and control group.

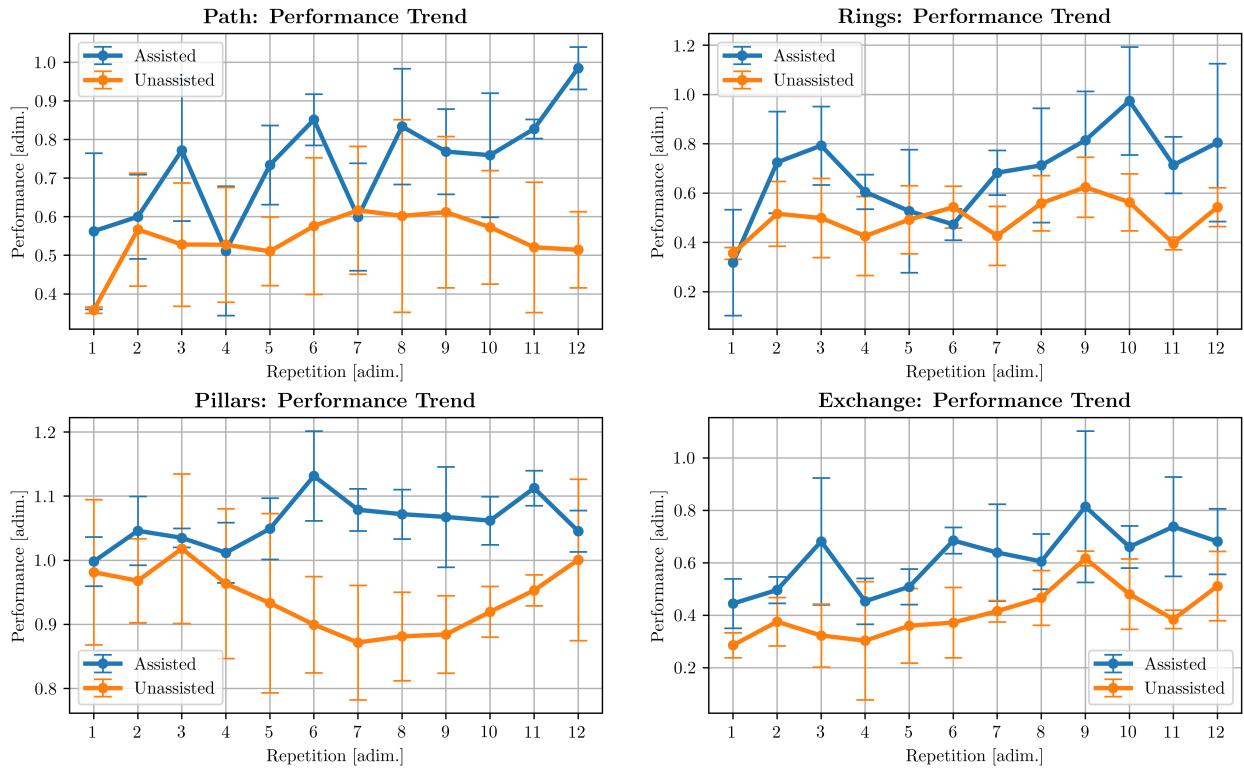


Figure 4.1: Performance Trend for the four Training Tasks. The value at each repetition is averaged across the subjects belonging to the same group. Error bars show the standard variation at each repetition

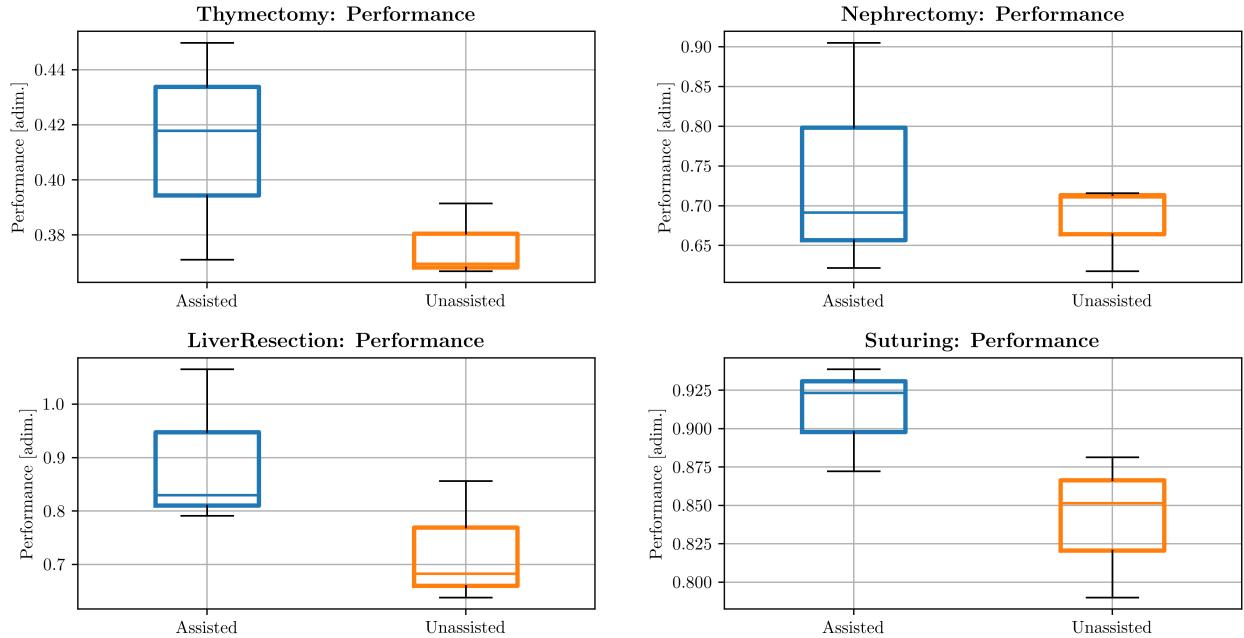


Figure 4.2: Boxplots of the average performance of assisted subjects (blue) and unassisted subjects (orange) on the four validation tasks

Thymectomy			
	Mean	STD	Median
CONTROL	0.38	0.01	0.37
ASSISTED	0.41	0.04	0.42
% VAR	9.83%	193.94%	13.07%

Nephrectomy			
	Mean	STD	Median
CONTROL	0.68	0.06	0.71
ASSISTED	0.74	0.15	0.69
% VAR	8.50%	167.46%	-2.71%

Liver Resection			
	Mean	STD	Median
CONTROL	0.73	0.12	0.68
ASSISTED	0.90	0.15	0.83
% VAR	23.42%	28.85%	21.54%

Suturing			
	Mean	STD	Median
CONTROL	0.84	0.05	0.85
ASSISTED	0.91	0.03	0.92
% VAR	8.38%	-25.39%	8.44%

Chapter 5

Discussion

Chapter 6

Conclusion

List of Figures

1.1	A surgical team in the operating room using a <i>daVinci</i> ® robot. Photo courtesy of <i>Intuitive Surgical Inc.</i>	11
2.1	The macro constituents of a <i>daVinci</i> ® surgical system in the operatory room: the Surgeon Console (left), the Vision Cart (middle) and the Patient Cart (right).	14
2.2	a. The <i>daVinci</i> ® surgeon console; b. The HRSV oculars; c. The MTMs; d. The foot switches	15
2.3	A visual representation of the concept of Remote Center of Motion (RCM). Independently by the configuartion of the PSM, the RCM is always located at the same point in space.	16
2.4	a. The patient cart of the <i>daVinci</i> ® robot; b. A PSM; c. The ECM; The SUJs of the PSM and ECM are in the orange boxes	17
3.1	Snapshot of the simulated surgical tasks, with the respective denomination. Training tasks have blue headlines, while realistic evaluation tasks have orange headlines. <i>Playground</i> is a propaedeutic task and isn't featured in the experimental study	23
3.2	The equivalence of transformations between the console space and the surgical space. The transformation matrix that expresses the pose of the manipulator in the HRSV reference frame is used to compute the desired pose of the PSM in the RF of the endoscope camera.	25
3.3	Plot of the Error Mapping function. The position oft and $t + h$ can be set manually to achieve a suitable behavior of the assistance strategy. The x-axis refers to a generic error shown in millimeters, but the same mapping function can be applied to an angular error.	27

3.4	Graphics scheme of the 4 virtual fixtures featured as assistance strategies in the surgical simulator. a. Trajectory Guidance; b. Obstacle Avoidance; c. Surface Guidance; d. Insertion Guidance. A representative PSM’s pose is shown, and a target pose is also depicted in a green hue, together with the force (orange arrows) and torque (blue arrows) that will guide the motion toward the target pose.	29
3.5	Vectors involved in the computation of the Trajectory Guidance VF. The PSM is shown in an example representative pose with respect to the reference trajectory.	30
3.6	Vectors involved in the computation of the Obstacle Avoidance VF. The PSM is shown in an example representative pose with respect to the obstacle mesh.	31
3.7	On the left, the assistance force is generated from the non-averaged normals, defined at the subsurface centerpoints. On the right, the assistance force is generated from the averaged normals, defined at the vertices. In this figure, the mesh is composed of 4-vertices “quads”, but the same concept holds to meshes with subsurface triangles.	32
3.8	Vectors involved in the computation of the Surface Guidance VF. The PSM is shown in an example representative pose with respect to the surface mesh.	33
3.9	Vectors involved in the computation of the Insertion Guidance VF. The PSM is shown in a “current” example representative pose with respect to the insertion cone, which is defined from the initial PSM position and the target position.	34
3.10	Schematic of the training protocol applied to the subject undergoing the experimental phase. On the days highlighted in orange, the VF assistance was provided to the group indicated; on the days highlighted in blue, the subjects were not assisted by VFs. In green, the <i>Playground</i> environment is a pre-operative task shown on the first day only, to familiarize the subjects with the simulator and the <i>daVinci®</i> system.	37
3.11	Graphical representation of some of the performance metrics recorded during task execution.	39
4.1	Performance Trend for the four Training Tasks. The value at each repetition is averaged across the subjects belonging to the same group. Error bars show the standard variation at each repetition	42
4.2	Boxplots of the average performance of assisted subjects (blue) and unassisted subjects (orange) on the four validation tasks	42

List of Tables

3.1	Error mapping parameters for the different tasks. <i>Exchange</i> , <i>Liver Resection</i> and <i>Suturing</i> map both the distance error and the angular error, so the values of both mapping functions are reported.	27
3.2	The surgical tasks featured in the simulator and the virtual fixtures algorithms used for each of them. Although the modular implementation of the simulator allows to include multiple VF algorithms in a single task (multiple-steps executions will be included in the future), for this study only one VF algorithm is used per task.	28
3.3	Weights assigned to the metrics in the computation of the performance score. The weight of a generic relative metric \hat{X} is denoted as $w_{\hat{X}}$	39

References

- [1] Mathias Hoeckelmann, Imre J. Rudas, Paolo Fiorini, Frank Kirchner, and Tamas Haidegger. Current capabilities and development potential in surgical robotics. *International Journal of Advanced Robotic Systems*, 12, 5 2015.
- [2] Kevin Cleary and Charles Nguyen. State of the art in surgical robotics: Clinical applications and technology challenges. *Computer Aided Surgery*, 6:312–328, 1 2001.
- [3] Michael A. Goodrich and Alan C. Schultz. Human-robot interaction: A survey, 2007.
- [4] Intuitive Surgical Inc. Intuitive reaches 10 million procedures performed using da vinci surgical systems, 2021.
- [5] Fontanelli G.A., Ficuciello F, Villali L., and Siciliano B. Modelling and identification of the da vinci research kit robotic arms. 2017.
- [6] Monty A. Aghazadeh, Miguel A. Mercado, Michael M. Pan, Brian J. Miles, and Alvin C. Goh. Performance of robotic simulated skills tasks is positively associated with clinical robotic surgical performance. *BJU International*, 118:475–481, 9 2016.
- [7] Justin D. Bric, Derek C. Lumbard, Matthew J. Frelich, and Jon C. Gould. Current state of virtual reality simulation in robotic surgery training: a review, 6 2016.
- [8] Anup Kumar, Roger Smith, and Vipul R. Patel. Current status of robotic simulators in acquisition of robotic surgical skills, 3 2015.
- [9] Rory McCloy and Robert Stone. Virtual reality in surgery. *Science, medicine, and the future*, 2001.
- [10] Thomas S Lendvay, Pasquale Casale, Robert Sweet, and Craig Peters. Initial validation of a virtual-reality robotic simulator. *Journal of Robotic Surgery*, 2(3):145–149, 2008.

- [11] Andrea Moglia, Vincenzo Ferrari, Luca Morelli, Mauro Ferrari, Franco Mosca, and Alfred Cuschieri. A systematic review of virtual reality simulators for robot-assisted surgery, 6 2016.
- [12] Michelle A Lerner, Mikias Ayalew, William J Peine, and Chandru P Sundaram. Does training on a virtual reality robotic simulator improve performance on the da vinci® surgical system? *Journal of Endourology*, 24(3):467–472, 2010.
- [13] Ruslan Korets, Adam C Mues, Joseph A Graversen, Mantu Gupta, Mitchell C Benson, Kimberly L Cooper, Jaime Landman, and Ketan K Badani. Validating the use of the mimic dv-trainer for robotic surgery skill acquisition among urology residents. *Urology*, 78(6):1326–1330, 2011.
- [14] Sem F Hardon, Anton Kooijmans, Roel Horeman, Maarten van der Elst, Alexander LA Bloemendaal, and Tim Horeman. Validation of the portable virtual reality training system for robotic surgery (polars): a randomized controlled trial. *Surgical endoscopy*, 36(7):5282–5292, 2022.
- [15] Roger Smith, Vipul Patel, and Richard Satava. Fundamentals of robotic surgery: A course of basic robotic surgery skills based upon a 14-society consensus template of outcomes measures and curriculum development. *International Journal of Medical Robotics and Computer Assisted Surgery*, 10:379–384, 9 2014.
- [16] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots, 1986.
- [17] Louis B Rosenberg. Virtual fixtures: Perceptual tools for telerobotic manipulation. In *Proceedings of IEEE virtual reality annual international symposium*, pages 76–82. Ieee, 1993.
- [18] Stuart A. Bowyer, Brian L. Davies, and Ferdinando Rodriguez Y Baena. Active constraints/virtual fixtures: A survey. *IEEE Transactions on Robotics*, 30:138–157, 2014.
- [19] Shahram Payandeh and Zoran Stanisic. On application of virtual fixtures as an aid for telemanipulation and training. pages 18–23. Institute of Electrical and Electronics Engineers Inc., 2002.
- [20] A Iranfar, M Motaharifar, and HD Taghirad. A dual-user teleoperated surgery training scheme based on virtual fixture. In *2018 6th RSI international conference on robotics and mechatronics (IcRoM)*, pages 422–427. IEEE, 2018.

- [21] Minsik Hong and Jerzy W Rozenblit. A haptic guidance system for computer-assisted surgical training using virtual fixtures. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 002230–002235. IEEE, 2016.
- [22] Peter Kazanzidesf, Zihan Chen, Anton Deguet, Gregory S. Fischer, Russell H. Taylor, and Simon P. Dimaio. An open-source research kit for the da vinci® surgical system. pages 6434–6439. Institute of Electrical and Electronics Engineers Inc., 9 2014.
- [23] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.
- [24] Nima Enayati, Eva C.Alves Costa, Giancarlo Ferrigno, and Elena De Momi. A dynamic non-energy-storing guidance constraint with motion redirection for robot-assisted surgery. volume 2016-November, pages 4311–4316. Institute of Electrical and Electronics Engineers Inc., 11 2016.
- [25] Alessandro Bettini, Panadda Marayong, Samuel Lang, Allison M. Okamura, and Gregory D. Hager. Vision-assisted control for manipulation using virtual fixtures. *IEEE Transactions on Robotics*, 20:953–966, 12 2004.