

Chapter 1

Materials and Methods

1.1 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 [1] 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 [2] 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 Patient-Side Manipulators (PSMs) and of the Endoscope Camera Manipulator (ECM) are implemented: the High-Resolution Stereo Viewer (HRSV) shows the 3D virtual surgical scene, the Master Tool Manipulators (MTMs) correctly controls the virtual surgical tools, and the clutch foot-switch allows proper repositioning maneuvers.

1.2 The Surgical Simulator

The objective of investigating a high-specificity aspect regarding the impact of Virtual Fixtures (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 [3].

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 Operatory Room (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.

1.2.1 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. 1.1 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

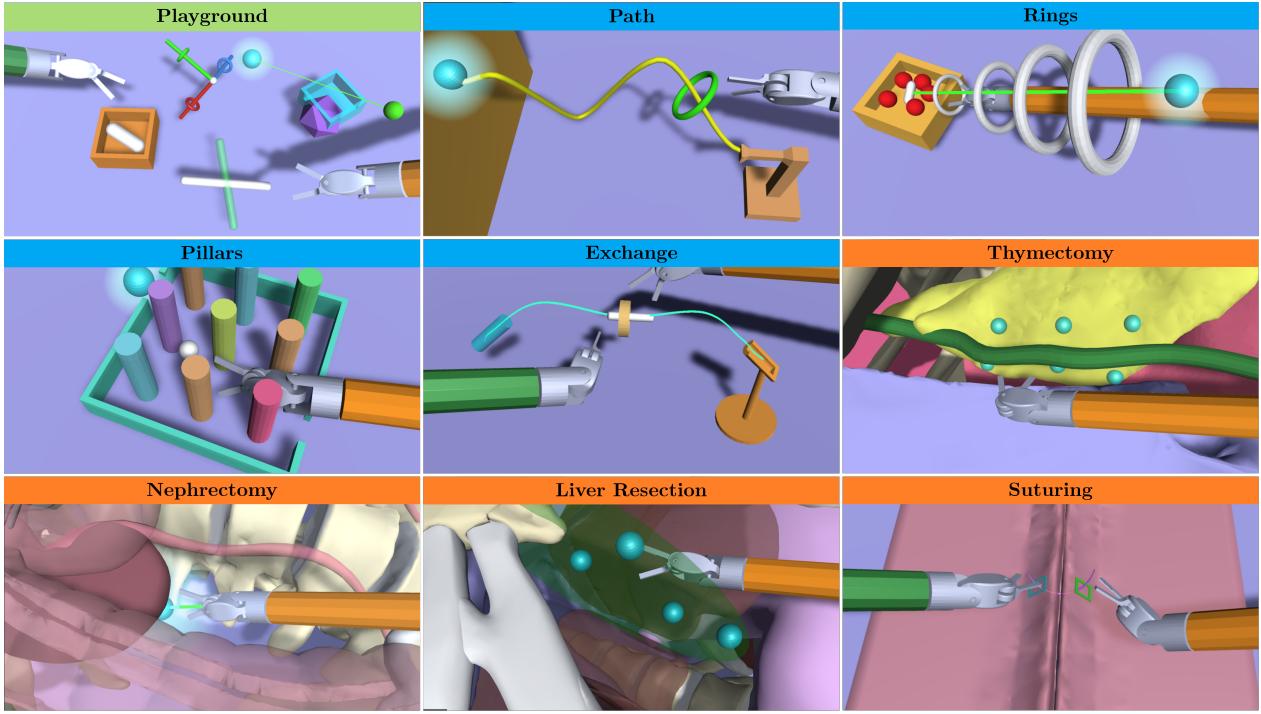


Figure 1.1: Snapshot of the simulated surgical tasks, with the respective denomination. Training tasks have blue headlines, while realistic evaluation tasks have orange headlines. *Playground* is a propaedeutic task and isn't featured in the experimental study

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 Virtual Reality (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.

1.3 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. 1.2.

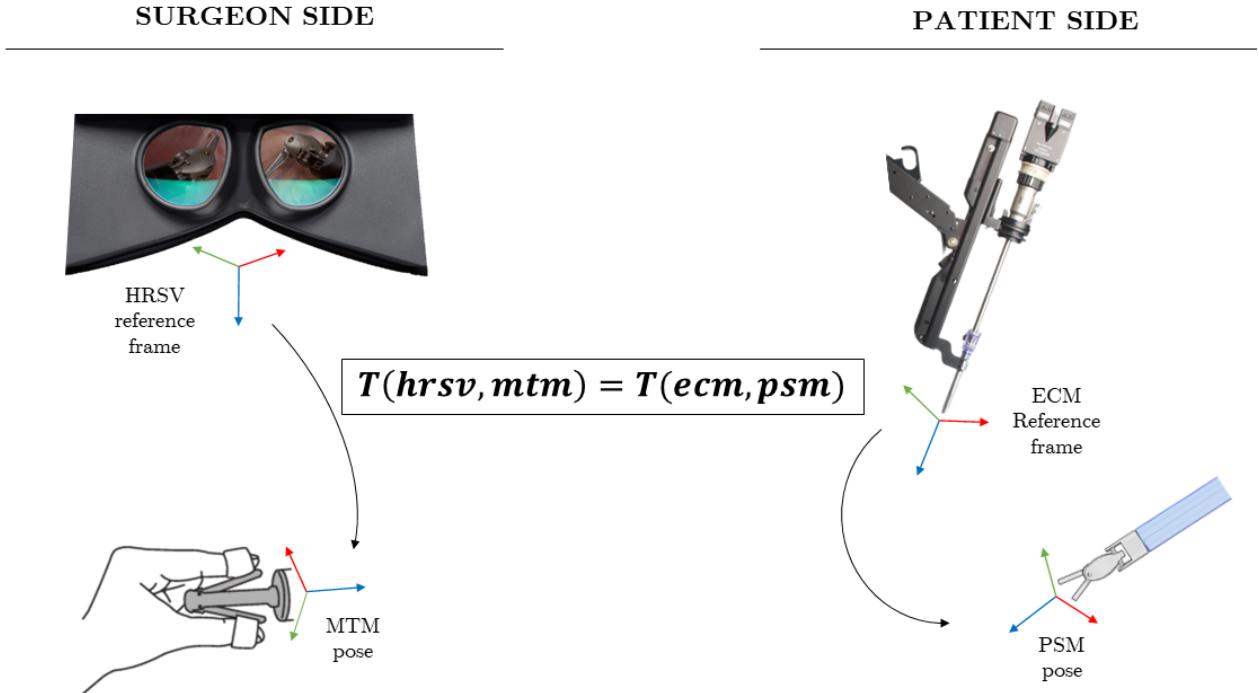


Figure 1.2: The equivalence of transformations between the console space and the surgical space. The transformation matrix that expresses the pose of the manipulator in the HRSV reference frame is used to compute the desired pose of the PSM in the RF of the endoscope camera.

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 [4] 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.

1.3.1 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:

$$f_{map}(x) = \frac{1}{1 + e^{5\delta w(x-t-h)}} \quad (1.1)$$

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.1.1.

1.3.2 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.

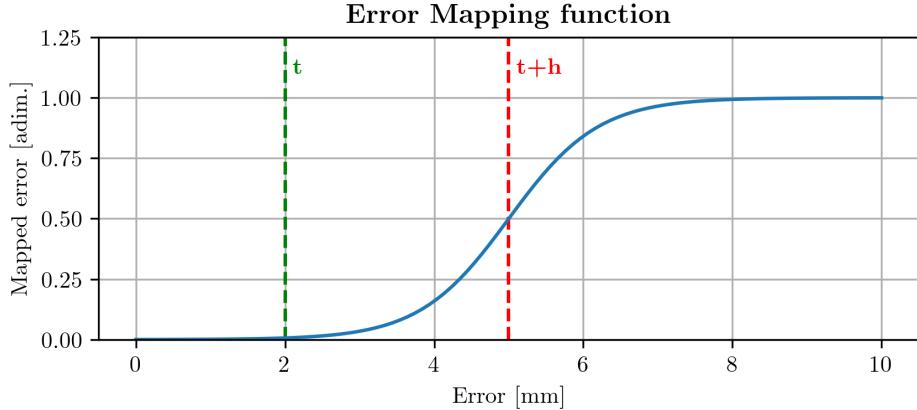


Figure 1.3: Plot of the Error Mapping function. The position of t and $t + h$ can be set manually to achieve a suitable behavior of the assistance strategy. The x-axis refers to a generic error shown in millimeters, but the same mapping function can be applied to an angular error.

Table 1.1: 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,

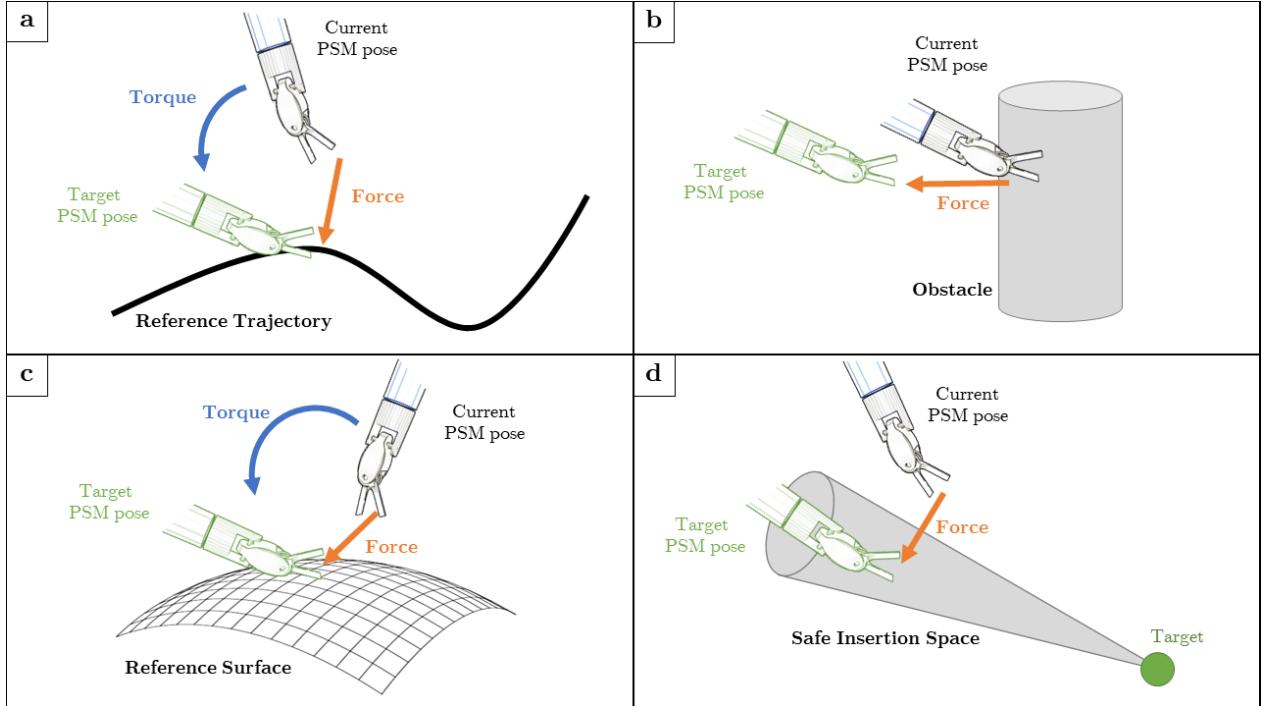


Figure 1.4: Graphics scheme of the 4 virtual fixtures featured as assistance strategies in the surgical simulator. **a.** Trajectory Guidance; **b.** Obstacle Avoidance; **c.** Surface Guidance; **d.** Insertion Guidance. A representative PSM's pose is shown, and a target pose is also depicted in a green hue, together with the force (orange arrows) and torque (blue arrows) that will guide the motion toward the target pose.

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.

Fig. 1.4 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

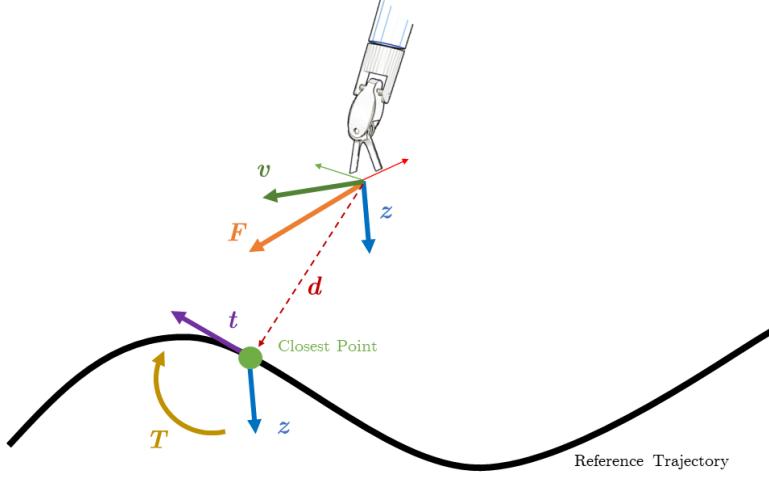


Figure 1.5: Vectors involved in the computation of the Trajectory Guidance VF. The PSM is shown in an example representative pose with respect to the reference trajectory.

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 calculated as

$$\mathbf{F} = k_F \cdot \mathbf{F}_{elastic} + \eta_F \cdot \mathbf{F}_{viscous} \quad (1.2)$$

$$\mathbf{T} = k_T \cdot \mathbf{T}_{elastic} + \eta_T \cdot \mathbf{T}_{viscous} \quad (1.3)$$

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. 1.5 illustrates the vectors involved in the computation of the virtual fixture; specifically, contributions in Eq. 1.2 expanded as:

$$\mathbf{F}_{elastic} = f_{map}(\|\mathbf{d}\|) \cdot \frac{\mathbf{d}}{\mathbf{d}} \quad (1.4)$$

$$\mathbf{F}_{viscous} = \begin{cases} \mathbf{d}, & \text{if } \mathbf{v} \cdot \mathbf{d} < 0 \\ \text{rotate}(\mathbf{v}, \theta, \mathbf{r}), & \text{otherwise} \end{cases} \quad (1.5)$$

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

$$\theta = (1 + \mathbf{v} \cdot \mathbf{d}) \cdot \frac{\pi}{2} \quad (1.6)$$

$$\mathbf{r} = \mathbf{v} \times \mathbf{d} \quad (1.7)$$

This implementation is adapted from [5]. Similarly, contributions to the torque (Eq. 1.3) are expanded as:

$$\mathbf{T}_{elastic} = \arccos(\mathbf{z} \cdot \mathbf{t}) \cdot \mathbf{z} \times \mathbf{t} \quad (1.8)$$

$$\mathbf{T}_{viscous} = \frac{d}{dt} [\arccos(\mathbf{z} \cdot \mathbf{t})] \cdot \mathbf{z} \times \mathbf{t} \quad (1.9)$$

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. 1.8 and Eq. 1.9, 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 is depicted in Fig. 1.6), the force is again computed as the sum of an elastic component, modulated by k , and a viscous one, modulated by η :

$$\mathbf{F} = \left[k \cdot f_{map}(\|\mathbf{d}\|) + \eta \cdot \frac{d}{dt} (f_{map}(\|\mathbf{d}\|)) \right] \cdot \frac{\mathbf{n}}{\|\mathbf{n}\|} \quad (1.10)$$

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. 1.7 (left), the closest point

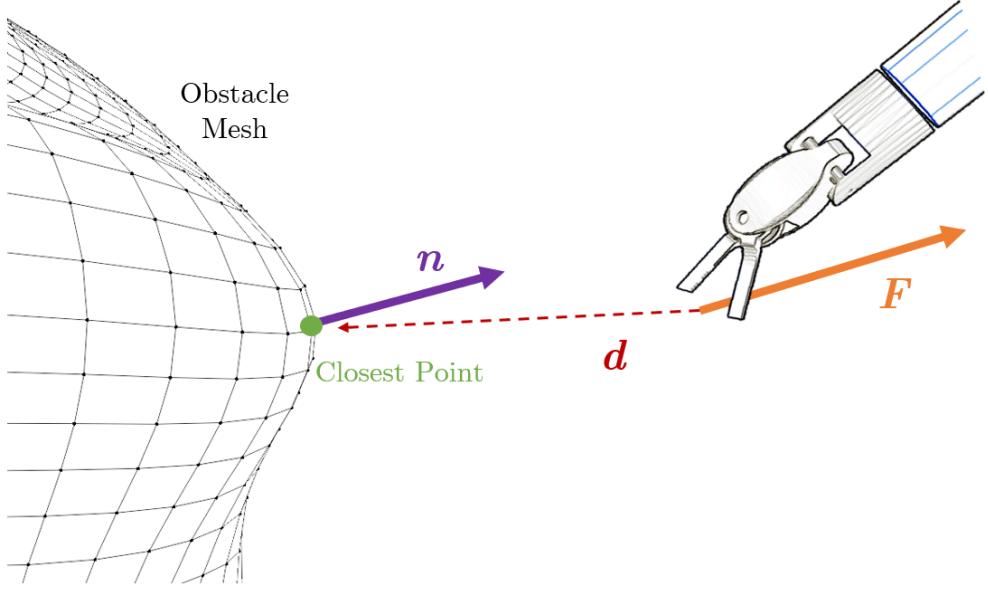


Figure 1.6: Vectors involved in the computation of the Obstacle Avoidance VF. The PSM is shown in an example representative pose with respect to the obstacle mesh.

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. 1.7 (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.

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

$$\hat{\mathbf{n}} = \frac{1}{S} \sum_{i=1}^S \mathbf{n}_i \quad (1.11)$$

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 align 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

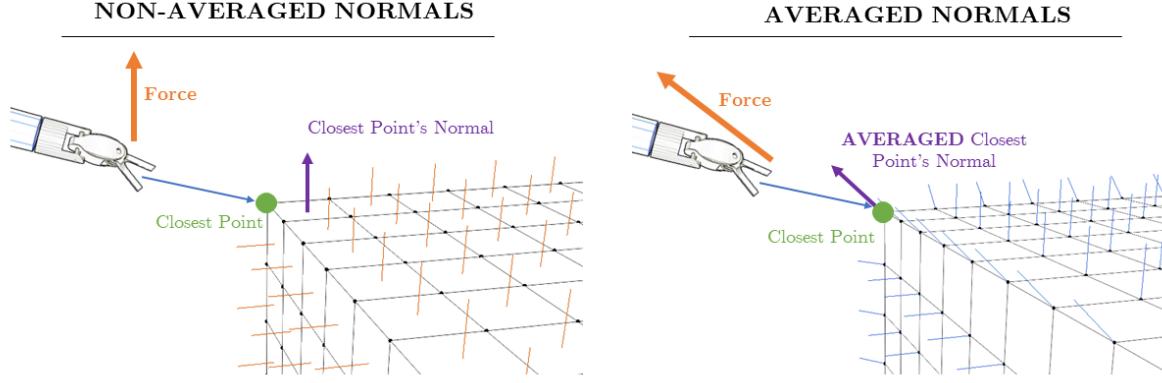


Figure 1.7: On the left, the assistance force is generated from the non-averaged normals, defined at the subsurface centerpoints. On the right, the assistance force is generated from the averaged normals, defined at the vertices. In this figure, the mesh is composed of 4-vertices “quads”, but the same concept holds to meshes with subsurface triangles.

attractive and is highest when the PSM is furthest from the surface mesh. The formula for the force is:

$$\mathbf{F} = \left[k_F \cdot f_{map}(\|\mathbf{d}\|) + \eta_F \cdot \frac{d}{dt} (f_{map}(\|\mathbf{d}\|)) \right] \cdot \frac{-\mathbf{n}}{\|\mathbf{n}\|} \quad (1.12)$$

Compared to Eq. 1.10, 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 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:

$$\mathbf{z}_{proj} = \mathbf{z} - (\mathbf{z} \cdot \mathbf{n})\mathbf{n} \quad (1.13)$$

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

$$\mathbf{T} = \left[k_T \cdot f_{map}(\theta) + \eta_T \cdot \frac{d}{dt} (f_{map}(\theta)) \right] \cdot \frac{\mathbf{z} \times \mathbf{z}_{proj}}{\|\mathbf{z} \times \mathbf{z}_{proj}\|} \quad (1.14)$$

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

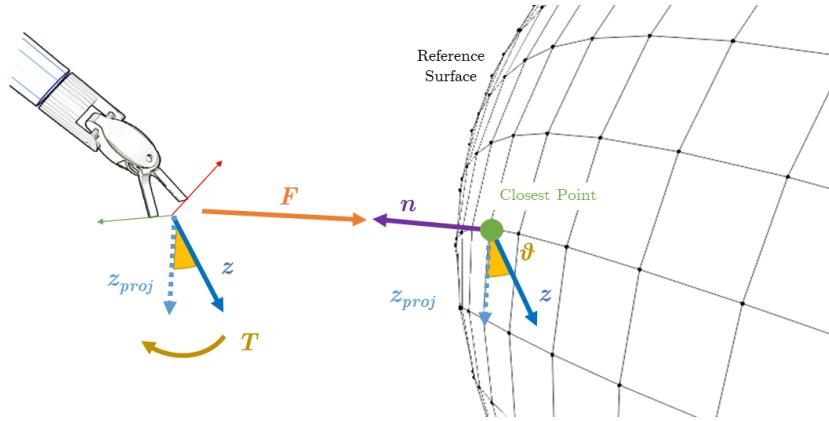


Figure 1.8: Vectors involved in the computation of the Surface Guidance VF. The PSM is shown in an example representative pose with respect to the surface mesh.

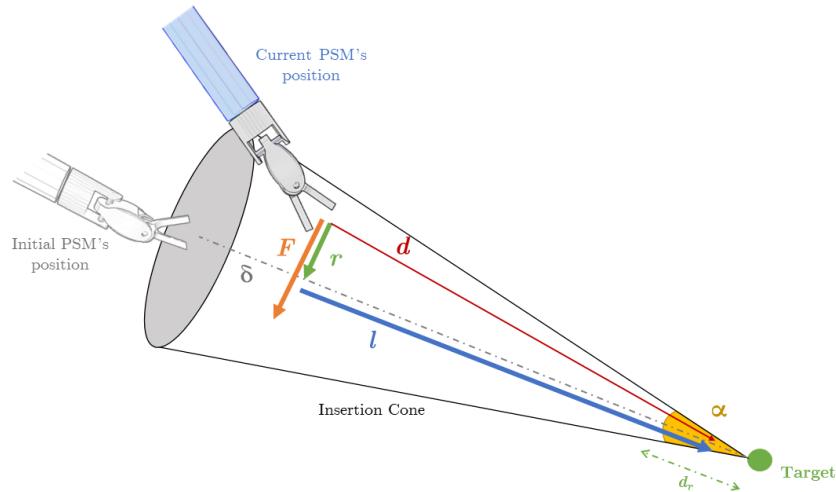


Figure 1.9: Vectors involved in the computation of the Insertion Guidance VF. The PSM is shown in a “current” example representative pose with respect to the insertion cone, which is defined from the initial PSM position and the target position.

The surface mesh is submitted to the same smoothing procedure as the obstacle mesh (Fig. 1.7, Eq. 1.11).

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 δ 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:

$$\mathbf{l} = \left(\mathbf{d} \cdot \frac{\boldsymbol{\delta}}{\|\boldsymbol{\delta}\|} \right) \frac{\boldsymbol{\delta}}{\|\boldsymbol{\delta}\|} \quad (1.15)$$

$$\mathbf{r} = \mathbf{d} - \mathbf{l} \quad (1.16)$$

Fig. 1.9 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:

$$a = \tan \left(\alpha \cdot \frac{\pi}{180} \right) \quad (1.17)$$

$$\|\mathbf{r}\| = \|\mathbf{l}\| \cdot a \quad (1.18)$$

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:

$$\mathbf{F} = r(\|\mathbf{l}\|) \left[k \cdot f_{map}(\|\mathbf{a}\|) + \eta \cdot \frac{d}{dt} (f_{map}(\|\mathbf{a}\|)) \right] \cdot \frac{\mathbf{a}}{\|\mathbf{a}\|} \quad (1.19)$$

with

$$r(\|\mathbf{l}\|) = \begin{cases} 1 & \text{if } \|\mathbf{l}\| > d_r \\ 0.5 & \text{if } \|\mathbf{l}\| \leq d_r \end{cases} \quad (1.20)$$

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.* [6].

1.4 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.

1.5 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.

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 1.10: 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 *daVinci*® system.

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. 1.1), 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 VF algorithms.

This training protocol is schematized in Fig. 1.10.

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 investigate 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.

1.6 Performance Metrics

Bibliography

- [1] 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.
- [2] 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.
- [3] 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.
- [4] Fontanelli G.A., Ficuciello F, Villali L., and Siciliano B. Modelling and identification of the da vinci research kit robotic arms. 2017.
- [5] 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.
- [6] 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.