UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
INGENIERÍA ELÉCTRICA – CONTROL

HAPTIC RENDERING FROM HOLONOMIC AND NONHOLONOMIC CONSTRAINTS FOR INTERACTION WITH VIRTUAL SURFACES USED IN MEDICAL SIMULATION

TESIS
QUE PARA OPTAR POR EL GRADO DE:
DOCTOR EN INGENIERÍA

PRESENTA:
M. I. JOSÉ DANIEL CASTRO DÍAZ

TUTOR PRINCIPAL
DR. MARCO ANTONIO ARTEAGA PÉREZ, FACULTAD DE INGENIERÍA,UNAM
COMITÉ TUTOR
DR. FERNANDO ARÁMBULA COSÍO, IIMAS, UNAM
DR. YU TANG XU, FACULTAD DE INGENIERÍA, UNAM

CIUDAD DE MÉXICO, 2021

JURADO ASIGNADO:

Presidente:	DR. LUIS AGUSTÍN ÁLVAREZ ICAZA LONGORIA
Secretario:	DR. YU TANG XU
1 er. Vocal:	DR. MARCO ANTONIO ARTEAGA PÉREZ
2 do. Vocal:	DR. FERNANDO ARÁMBULA COSÍO
3 er. Vocal:	DR. ALEJANDRO RODRÍGUEZ ÁNGELES

Lugar donde se realizó la tesis: FACULTAD DE INGENIERÍA, UNAM.

TUTOR DE TESIS:

DR. MARCO ANTONIO ARTEAGA PÉREZ

FIRMA

To my students:

Iván Torres
Andrés de la Torre
Pedro Peña
Rafael Huerta
Axel Torres
Enrique Martínez
Tonatiuh Vilchis
Brayan Alonso.

For becoming the reason and inspiration to obtain my Ph.D.

Acknowledgments

First, I would like to thank my advisor and mentor Dr. Marco Arteaga for supporting me so steadfastly throughout my whole academic career and for mark a before and an after in my professional life.

I must thank to Dr. Alejandro Gutiérrez-Giles, for his countless contributions to this work: the mathematical formality given to the developed theory, his invaluable help with the experimental development, and, above all, his counseling on completing my Ph.D.

I must also thank Dr. Javier Pliego for contributing with his ideas to the solution of the problem presented in this work and for always encouraging me to never give up.

I want also like to thank my thesis committee, Dr. Fernando Arámbula, Dr. Yu Tang, Dr. Alejandro Rodríguez and Dr. Luis Álvarez-Icaza, whose comments helped me to considerably improve the present document.

A special thanks to José Luis Sánchez-Carmona for helping me with the design and implementation of the virtual environment presented in this dissertation.

I would like to thank my dear friend Mauro López for becoming the perfect college and Ph.D. companion.

A special thanks to Dr. Luis Álvarez-Icaza and to his daughter Dr. Dení Álvarez-Icaza for always giving me their advice and support during difficult times.

My eternal love and gratitude to my mother, my father and my brother for be there always I needed, supporting and encouraging me.

Last but not least, my sincere gratitude to my *alma mater*, the National Autonomous University of México (UNAM) for giving me the opportunity and privilege to spread knowledge and provide integral education to new generations of Mexicans engineers.

This work was supported by CONACyT under scholarship CVU No. **424680** and DGAPA-UNAM under Grant **IN114617** and **IN117820**.

Abstract

Virtual reality systems for medical training were inspired by teleoperation systems. In both cases, robotic arms play a major role because they provide a force response to be transmitted to the human operator. Such force produces a tactile sensation that allow to feel some properties of either the remote or virtual environment. However, in the last two decades, the research efforts in the area have been focused on visually simulating, as realistic as possible, the virtual environments present in surgical training. This entails the force response to be generated by methods that cannot reproduce some characteristics of the virtual surfaces, as in the case of penetrable objects. To counteract such problem, in this work a virtual reality system with haptic feedback is studied using a teleoperation approach. By defining the interface manipulated by the operator as the master robot and the virtual environment as the slave robot, the force response is obtained by approaching the virtual environment as a problem of a robot in constrained motion.

The main objective of this work is to reproduce the tactile properties of both a penetrable or non penetrable virtual surface by using virtual constraints. Additionally, a control algorithm based on a teleoperation system is implemented to feedback the corresponding force to the operator. To achieve this objective, it was necessary to design a virtual environment consisting in a robot dynamic model in contact with either holonomic and nonholonomic constraints. Furthermore, according with the functioning of a medical training simulator, before the contact, there is always an stage of free motion. For this reason and specially in the case of nonholonomic constraints, a collision detection algorithm was programmed using the implicit equation of a sphere. This allows to have a tactile perception before and after contact to consequently make appropriate comparisons of the operation of the control algorithm, either in free or constrained motion.

A set of experiments were carried out by considering the force feedback from a sensor at the master side and reproducing computationally the virtual force by means of holonomic and nonholonomic constraints. Furthermore, a virtual environment was built in order to al-

low the operator to have visual feedback with the aim of heightening the realism of the application. Such environment needs forcefully a good performance of the control scheme since it is necessary an adequate tracking, both in force and position, to reproduce the virtual Cartesian space and the sensation of being in contact with a surface.

The experimental results shown some interesting differences between using holonomic and nonholonomic constraints. The principal of them is the modification of the control scheme structure with respect to the force feedback. Furthermore, a good performance of such scheme was obtained for both position and force. This means that the human operator receives a visual image according to his/her movements and also a force response that reproduces the tactile properties of the virtual surface. Since this haptic sensation is entirely subjective, a discussion was carried out with special emphasis on the validity of the proposed approach and how it can be used in the virtual simulation of a medical procedure.

Resumen

Los sistemas de realidad virtual para entrenamiento médico fueron inspirados por los sistemas de teleoperación. En ambos casos, los brazos robóticos juegan un papel importante dado que proporcionan una respuesta de fuerza que es transmitida al operador humano. Dicha fuerza produce una sensación táctil que permite sentir algunas propiedades del entorno remoto o virtual. Sin embargo, en las últimas dos décadas, los esfuerzos de investigación en el área se han centrado en simular de manera visual, lo más realistamente posible, los entornos virtuales presentes en el entrenamiento quirúrgico. Esto implica que la respuesta de fuerza sea generada por métodos que no pueden reproducir algunas características de las superficies virtuales, como en el caso de los objetos penetrables. Para contrarrestar este problema, en este trabajo se estudia un sistema de realidad virtual con retroalimentación háptica utilizando un enfoque de teleoperación. Al definir la interfaz manipulada por el operador como el robot maestro y el entorno virtual como el robot esclavo, la respuesta de fuerza es obtenida abordando al entorno virtual como un problema de un robot en movimiento restringido.

El objetivo principal de este trabajo es reproducir las propiedades táctiles de una superficie virtual, ya sea penetrable o no, mediante el uso de restricciones virtuales. Además, se implementa un algoritmo de control basado en un sistema de teleoperación que retroalimenta la fuerza correspondiente al operador. Para lograr este objetivo fue necesario diseñar un entorno virtual que consiste en el modelo dinámico de un robot en contacto con restricciones holonómicas y no holonómicas. Además, de acuerdo con el funcionamiento de un simulador de entrenamiento médico, antes del contacto, siempre hay una etapa de movimiento libre. Por esta razón y especialmente en el caso de restricciones no holonómicas, se programó un algoritmo de detección de colisiones utilizando la ecuación implícita de una esfera. Esto permite tener una percepción táctil antes y después del contacto para consiguientemente hacer las comparaciones apropiadas del funcionamiento del algoritmo de control, ya sea en movimiento libre o movimiento restringido.

Se llevó a cabo un conjunto de experimentos considerando retroalimentación de un sensor de fuerza en el lado maestro y reproduciendo computacionalmente la fuerza virtual por medio de restricciones holonómicas y no holonómicas en el lado esclavo. Además, se creó un entorno virtual que permite al operador contar con retroalimentación visual, aumentando así el realismo de la aplicación. Dicho entorno necesita forzosamente un buen rendimiento del esquema de control, tanto en fuerza como en posición, ya que es necesario un adecuado seguimiento en ambas variables para reproducir tanto el espacio cartesiano virtual como la sensación de estar en contacto con una superficie.

Los resultados experimentales muestran algunas diferencias interesantes entre el uso de restricciones holonómicas y no holonómicas. La principal implicó modificar la estructura del esquema de control con respecto a la retroalimentación de fuerza. Además, se obtuvo un buen rendimiento de dicho esquema tanto para posición como para fuerza. Esto significa que el operador humano recibe una imagen visual de acuerdo con sus movimientos y también una respuesta de fuerza que reproduce las propiedades táctiles de la superficie virtual. Dado que esta sensación háptica es completamente subjetiva, se llevó a cabo una discusión haciendo especial énfasis en la validez del enfoque propuesto y cómo puede ser utilizado en la simulación virtual de un procedimiento médico.

Contents

Chapter 1

Introduction

Teleoperation and virtual reality systems are intrinsically related since they make a human operator to interact with environments without being in physical contact with them. In the former, such environments are real while in the latter they are generated in a computer simulation. Nevertheless, both types of systems have to make the operator to perceive, as realistically as possible, the characteristics of either the remote or virtual environment. Some variables used to reproduce such characteristics are position and force, which provide visual and haptic feedback respectively. The main difference is how the information received by the operator is generated. In the teleoperation case, both signals physically exist and they are transmitted by means of a control algorithm that receives them from sensors present in the remote environment. On the other hand, in virtual reality systems such signals do not exist and they must be produced computationally.

Stimulate the senses of sight and touch, as precise as possible, is essential during the interaction process since they are the principal channels with which the operator perceives the world around. For teleoperation, in the visual case, the communication comes directly from the environment or, if the operator is in a remote place, using a camera and a computer screen. On the other hand, the virtual reality system generates the environment through a digital simulation, and the operator receives the visual information through a screen. The case of touch is more complex since additional devices capable to transmit the forces generated in the environments are needed. Such process implies including haptic robots into the systems due to their capability to generate forces and torques that the human operator can perceive in a tactile way. For teleoperation, two physical robots are needed while in virtual reality systems, only one is used, considering the virtual environment as the other.

The medical area has actively seized on both teleoperation and virtual reality systems. In the first case, a specialist can perform surgery procedures over long distances, eliminating the need of physical presence of either the physicians or patients in the same location [Avgousti et al., 2016]. Practically, he/she can examine or operate on the patient at a different geographic location without having to travel. In Figure 1.1 the emblematic da Vinci system is shown, operating on the basis of a master-slave control concept. The system provides the medical expert with a realistic operating environment that includes a high-quality stereo visualization and a human-machine interface that directly transfers the doctor's hand gestures to the instrument tip movement inside the patient [Ballantyne and Moll, 2003].

Figure 1.1: The da Vinci surgical system®.

In the second case, virtual reality systems have been widely used for medical training simulation. With the development of computer graphics, nowadays practically any surgical procedure can be visually simulated. Nevertheless, the most benefited area has been the minimally invasive surgery, where virtual environments for laparoscopy, neurosurgery and urology, among many others have been implemented [Basdogan et al., 2004]. As an example, in Figure 1.2 the simulator of Transurethral Resection of the Prostate TURP Mentor is shown. By using this kind of systems, the practice *in vivo* with humans can be avoided and therefore the practitioners can heighten their skills without putting in danger their patients. An important factor for the improvement of such skills is the inclusion of tactile feedback, which must be synchronized with both the virtual reality simulation and the movements of the operator.

Figure 1.2: The TURP Mentor simulator®.

The challenge for the researchers in graphic computing and control systems is to design mathematical tools that fit with the physical characteristics of the objects to be simulated within the virtual environment. Regarding to the visual feedback, the position where such objects are located is essential for the operator to perceive his/her movements in the Cartesian virtual space. With respect to the force, reproducing rigidity and softness takes special interest when the virtual environment includes penetrable and non penetrable objects. In this case, the complexity of the mathematical tools tends to increase because their physical laws are not always easy to simulate by a computer. For this reason, a tradeoff between visual and haptic realism must be established due to the finite capacity of digital processing.

1.1 Sate of the art

A wide range of teleoperation and virtual reality robotic applications has been developed in areas as different as automotive and video gaming. However, a particular important application is for medical surgery [Hannaford et al., 2013], where the operator at the master's side needs to be confident about the force he/she is feeling. Such a force, generated in opposition to his/her movements, must be ideally the same to that the robot applies over the patient at the slave side. On the other hand, virtual reality systems have been widely used in minimally invasive surgical simulation, where the operator should feel the same forces that he/she would feel in a real procedure [Dy et al., 2015].

The first master-slave teleoperation system was presented by Goertz [Goertz and Thompson, 1952]. It was used to manipulate toxic waste through two coupled manipulators and, a rudimentary force rendering device was included later by using electric signals. Since then, this kind of systems have been used in areas as micro and nanomanipulation [Hollis et al., 1990] underwater exploration [Khatib et al., 2016], and telesurgery [Kim et al., 2013]. Haptic feedback takes special relevance in this last area since it is crucial for the surgeon to receive an accurate force response. The most effective way to do so is to have force sensors at both the master and slave sides, and a control algorithm that guarantees an accurate tracking between the contact forces present in the remote environment and those send to the operator [Hansen et al., 2012].

On the other hand, the idea of virtual reality systems with force feedback started with the seminal work of Sutherland [Sutherland, 1965]. He established that the interaction between the human operator and the virtual environment should not only be visual but also tactile. It was not until the 1990's that the Goertz device was adapted to provide force feedback during virtual molecular dock [Brooks et al., 1990]. Thereafter, the use of manipulators in virtual reality applications has been extended to CAD/CAM assembly [Chu et al., 1997], aerospace maintenance [Angus and Stone, 1995] and above all medical training simulation [Hamza-Lup et al., 2011] where, in contrast with master-slave teleoperation systems, neither the environments nor the contact forces exist. Nonetheless, real forces have to be accurately rendered to the operator and whose quality depends on the characteristics of the haptic interface and the corresponding force control algorithm [Basdogan et al., 2004].

Articulated robots play a major role in medical training simulation systems with force feedback since this kind of electromechanical devices are capable of measuring spatial position and generating torques. Consequently, there has been a large effort to design robot haptic interfaces such as the widely used Phantom serial robot [Massie and Salisbury, 1994], the Delta Haptic parallel robot [Grange et al., 2001] and the combination of passive elements as brakes and springs with motors [Conti and Khatib, 2009]. Such robots are examples of impedance types devices, *i.e.* they read position and control force in response. On the other hand, there is another kind of robots that read forces and control motion, called admittance type devices. The difference between using one or another type relies on the characteristics of the virtual environment (*e.g.*, stiffness, inertia, damping, friction).

Along with haptic interfaces, there has been an intense development of graphical simulation tools capable of reproducing a wide range of virtual environments. The main objective is

for the operator to perceive, as realistic as possible, objects with a high quality of detail. The applications developed include microscopic exploration, [Finch et al., 1995], aviation [Bliss et al., 1997] and clinical neuropsychology [Rizzo et al., 2000], among many others. In the case of medical training simulation, a correct sinergy between visual and tactile feedback is essential to heighten the skills of medicine students. However, to increase the degree of immersion and consequently the realism of the virtual reality displays, it is necessary to model environments that combine haptic and graphics at the same degree of complexity [Ruspini et al., 1997]. This is not always possible since the computational processing is limited and the applications cannot be executed in real time.

Salisbury et al. [2004] presented the basic architecture for a virtual reality application with visual and haptic feedback. They established that the force rendering algorithms must be geometry dependent. This is a disadvantage in medical training simulation since the virtual objects to be reproduce include bones and/or organs with irregularities or indentations. Moreover, there are cases where the interaction does not occur only on the object's surfaces but forces due to penetration must be computed, as in surgery simulators. The alternative is to design algorithms based on physical laws that involve the dynamic and movement of the objects when the operator interacts with them [Escobar-Castillejos et al., 2016]. The perfect scenario would be to render forces by combining physical approaches with the most sophisticated haptic interfaces. However, as mentioned before, do that is computationally more expensive, not to mention the high costs it would entail.

The most realistic the force transmitted to the operator, the greater the method's quality. Nevertheless, the factors mentioned previously make necessary a series of tradeoffs between haptic and visual realism, execution in real time and costs of the system. Such tradeoffs led to establish two principal methods to render forces from virtual environments [Duriez et al., 2006]. The first one is the penalty method, which has been widely used due to its simplicity. It requires a penetration measurement starting at the contact point with the virtual object. The second approach is the imposed motion method, where the contact is considered as a bilateral constraint and the force response of the contact is calculated using Lagrange multipliers. Both methods have been used in computer graphics and haptic applications being Ruspini et al. [1997] who established the differences between use one or the other.

The penalty method assumes that the virtual objects and shapes are composed by geometrically or algebraically defined primitives [Terzopoulus et al., 1987] such as lines, planes, spheres and cylinders. Therefore, the force rendering depends on an implicit equation and a

contact point given by a collisions detection algorithm. Sclaroff and Pentland [1991] propose a method where implicit functions representation is generalized to allow collision detection for common shapes in 3D, replacing the polygon and spline representation used before. By using this technique, it is possible to use local gradients in the normal direction of the virtual surface [Minsky et al., 1990]. In this sense, the concept of *impedance* takes special relevance since the force rendering problem is addressed as an energy exchange phenomena, allowing to study the stability of the haptic system [Adams and Hannaford, 1999].

On the other hand, using the imposed motion method implies to consider the contacts with the virtual object as bilateral constraints [Duriez et al., 2006]. Therefore, the haptic system is adressed as a master-slave scheme where the stiffer object, *i.e.* the virtual representation of the haptic interface, is the master. This entails to employ Lagrange multipliers to compute the magnitude of the contact force. Bayo and Avello [1994] designed an algorithm where the dynamics of a multibody system is considered as a constraint. The advantage of this approach is that the force response can be described in terms of the dynamic characteristic of the surfaces and not only in terms of a single contact point [Zilles and Salisbury, 1995]. However, the hardware requirement for haptic rendering in 3D is for the haptic interface to have at least 3 degrees of freedom.

The methods mentioned above have been the cornerstone for virtual forces generation both in graphic computing and haptic systems. One of the principal requirements in such areas is that the systems be capable of reproducing the forces that would be present during the contact with rigid and soft objects. This is especially important in the case of medical training simulation where the tactile sensations caused by the contact between a virtual tool and bones or organs must be reproduced [Mavhash and Hayward, 2004]. Nonetheless, depending on the objectives of the system's designer, a tradeoff between haptic and visual realism must be established.

1.2 Motivation and Problem Statement

While visual interaction features unidirectional information, haptic interaction exchanges information in two directions, from and towards the operator, *i.e.* it can be considered as a bilateral system. The process depends on a series of sequential stages, being collision-detection and force reflection the most important [Salisbury et al., 2004]. The force response algorithms compute the interaction forces when a collision arises between the visual representation of

the operator movements, or avatar, and the virtual objects. Subsequently, they return values of force and torque that are transmitted to the haptic interface. Ideally, this force response has to be identical to the one arising during the contact between the real tool and the objects to simulate, but since computational models do not represent exactly such objects, the task is challenging.

When rigid objects are defined by implicit equations, the force is generated by means of geometric methods. Arimoto et al. [1993] used the orthogonalization principle to decompose the force and position task into two orthogonal subspaces for a robot manipulator in contact with geometric constraints. If the virtual objects are considered as constraints, this principle can be extended to virtual surfaces since its mathematical description allow to obtain orthogonal forces in a single point. The difference is that in the first case, the constraint exists physically and in the second case they do not, nor does the contact force. This coincides with the work done in the graphic computing area where complex surfaces are build by means of planes defined by implicit equations and joined by vertices [Zilles and Salisbury, 1995].

The real challenge is when the object to be simulated is soft. For start, it is well known that deformable objects have infinite degrees of freedom and their behavior equations are impossible to solve analytically. Initially, the researchers incorporated physically-based dynamics of flexible materials into purely geometric models [Terzopoulus et al., 1987]. Later, with the aim of heighten visual realism, they used visco-elastic deformations that usually included both dynamics and geometric representation of soft surfaces [Saupin et al., 2008]. This implies the incorporation of continuous models as B-Splines, finite differences and finite elements [Montagnat et al., 2001]. As might be expected, the computational processing increased significantly, restricting the possibility to include an efficient algorithm to render forces in a realistic manner from deformable surfaces. Consequently, if the application search for both highly realistic visual and haptic rendering, the only way to do it is working offline.

Accordingly, the main objective of this dissertation is to reproduce the tactile properties of either a penetrable or non penetrable virtual surface by using virtual constraints. The penetrable surface is assumed to belong to a rigid object and the non penetrable one to a soft object. In the first case, the implicit equation of the surface is known and therefore the force can be rendered by defining a single point on the equation and obtaining the orthogonal direction of such force. However, in the second case, due to the complexity of the equations of the deformable surfaces, the force response cannot be rendered as easily as before. This is due to the fact that there is not only a normal vector at a single contact point over the surface. On

the contrary, as the virtual tool moves forward, a set of contact points arises around it as well as the corresponding lateral forces. In a practical sense, the force rendering produced in this way must make feel the operator that he/she cannot move laterally once inside the surface.

The solution proposed in this dissertation implies to generate a force from a penetrable object immersed in a virtual environment without considering the deformable model of a soft surface. Furthermore, an implicit equation of such surface would not be needed either if the method designed is purely kinematic. It is convenient that the adopted approach be far from the common practice of associating the virtual avatar position and orientation of that of the haptic interface. This is due to the fact that, by considering a set of independent coordinates, the virtual constraints can be defined in terms of the virtual environment's dynamic model, similar to what happen in a teleoperation system. If so, a position-force control algorithm can be designed to obtain both haptic and visual feedback.

1.3 Methodology

The problem of force rendering from soft surfaces is an active research area. However, even when some authors suggest that the virtual reality systems with haptic feedback can be addressed as bilateral systems [Zilles and Salisbury, 1995; Salisbury et al., 2004; Duriez et al., 2006], there is no work in the literature that model the virtual environments using an independent set of coordinates as in teleoperation. The exception is presented by Faurling et al. [2005] and Faurling et al. [2007] whose ideas are the basis for the work presented in this dissertation. They stated that, from a haptic rendering point of view, using an independent set of coordinates for the virtual environment would be the most effective way to take into consideration the physical laws involved in the contact between the virtual tool and the surfaces. In fact, those laws have been extensively used in computer graphics in order to heighten the visual feedback realism. Nevertheless, as mentioned before, it entails resorting to finite element methods and aggregating the visco-elastic properties of the simulated tissues.

Before introducing the solution to haptic rendering from penetrable surfaces, it is convenient to mention the solution found in the literature for non penetrable surfaces. In this sense, holonomic constraints have been widely used to render forces from virtual environments since they can represent rigid objects from an implicit equation. For example, Faurling et al. [2007] simulate a virtual planar manipulator constrained to a circle in the x–y plane. In this case, an operator try to keep a constant velocity tangential to the constraint and an

admittance robot exerts the virtual normal force produced during the contact. In addition, Rodríguez et al. [2010] implement a *geometric constraint solver* to find all the possible configurations of a mobile object satisfying a set of holonomic constraints. Since complex virtual environments can be composed by simple geometric surfaces as planes, spheres, cylinders etc., the force can be rendered from a single point on them or from a small vicinity close to such point by using the penalty method.

In the case of non penetrable surfaces, the common strategy is to establish a collision detection algorithm considering only a single contact point. This solution is easy but not realistic since it ignores lateral and axial forces that are present during a classic deformation phenomenon as needle puncture. In this case, the research efforts are focused principally on reproducing, with high quality of details, the deformable surface in a visual manner. As an alternative, nonholonomic constraints are presented in this work to simulate the force response during the contact with a penetrable object. The works in the literature dealing with this type of constraints usually study the control of wheeled cars but in [Webster III et al., 2006], the first nonholonomic model to represent the deformation of a needle immersed in soft tissue is presented. Interestingly, in the cited work, the system is modeled to control the insertion and the shaft rotation speed in 2D. Rucker et al. [2013] use the Webster's model to formulate a sliding mode control law that causes the needle to reach a desired target in 3D within an error expressed as a function of the control input speeds. However, in spite of their modeling of the nonholonomic constraints in a relatively simple way, they do not deal neither with haptic nor visual feedback. Faurling et al. [2005] point out that in order to model a surgeon's scalpel both holonomic and nonholonomic constraints could be employed by limiting the depth of its incision and the direction of its motion respectively.

To fulfill the main objective of this dissertation, the work presented by Faurling et al. [2007] is extended, where two set of independent Cartesian coordinates are defined. Moreover, a new approach that address the virtual reality application as a teleoperation control scheme is developed. This last feature has been little exploited as an effort to use the benefits of incorporate some concepts of virtual environments to teleoperation. As an example, Rodríguez-Angeles et al. [2015] use a virtual planar surface in order to reproduce the force exerted by a slave robot in contact with a real plane at the remote environment. However, such plane is defined as an holonomic constraint with an implicit equation associated, and the virtual force is produced using the recursive algorithm presented by Bayo and Avello [1994]. The result was that, in an indirect way, the operator interacts with the plane at the local environment and

the forces exerted by the master robot track the signals produced in the local environment and obtained by a force sensor.

With respect to the simulation of soft surfaces, the developments are mainly focused on finite element methods for soft tissues simulations. For example, DiMaio and Salcudean [2003] present a methodology developed to determine needle forces during deformable body puncture. It is based on obtaining both axial and lateral forces from a finite set of mesh nodes but it is developed in 2D only. On the other hand, Nordberg and Servin [2018] propose a particle-based method for modeling and simulation of an elastoplastic solid as a multibody system. Even when its model heightens the visual realism by adding the dynamic model of the surface, it is computationally expensive and non-suitable for real-time force rendering.

The use of a bilateral teleoperation approach implies adding the dynamic model of the virtual environment in the digital simulation. Intuitively, this would be the most effective way to rendering forces from non penetrable surfaces since by simulating its dynamics, a force as close to reality as possible would be obtained. However, since the objective of this work is to propose an alternative method using nonholonomic constraints, the dynamic model of a 5 degrees of freedom constrained robot is used as an illustrative example. This allows to take into account the masking phenomenon, where the force is not sent to the operator in direct reflection, but inertial properties that do not belong neither to the virtual objects nor to the haptic interface are transmitted [Gallace and Spence, 2014]. For example, if the dynamics of the virtual tool is considered, it could be possible to reproduce its inertial properties by modifying the dynamic model of the simulation.

This feature can be illustrated as follows: in the case of medical training simulations, a scalpel has very low inertia and no centripetal forces or viscous friction are acting on it. That changes for a forceps, a retractor or an endoscope whose dynamic properties are more complex and it is desirable that they be transmitted to the operator. In some cases, the masking effect is not desired like in human-interface applications as touching screen electrovibration [Vardar et al., 2018] or in augmented reality vibrotactile applications [Asano et al., 2013]. But in others, such as in 3-D models security watermarking [Kim et al., 2010] is actively seized on. The relevance of this approach for the work presented in this dissertation is that, apart from holding a stylus attached to the master robot end-effector with the form of such medical tool, the operator could feel its dynamic characteristics [Papadopoulos et al., 2002].

When taking into account the masking effect, the election of an appropriate haptic device is crucial. As mentioned before, that depends on the properties of the virtual environment.

For very stiff objects, admittance devices are the ideal choice, while for soft objects impedance devices are chosen. The advantage of the former is that they include force sensors that allow admittance control schemes to be implemented and therefore, being the most complete option since force control is performed in a direct manner. In the case of the work presented in this dissertation, an impedance device is used due to both rigid and soft objects are simulated. Nonetheless, a 6 DOF force sensor is mounted on the haptic device in order to develop the teleoperation control scheme. In this way, the advantage of having force feedback is exploited and the masking effect can be correctly addressed by avoiding the high inertia that the admittance devices have.

Finally, by addressing the virtual reality application as a bilateral teleoperation system, it is possible to propose control schemes and carry out analysis techniques that are not usually used in computer graphics. This entails the advantage of improving application's performance of some important aspects like delays produced in the communication channels between system's haptic and graphic blocks. In this work, a comparison when using the teleoperation controller with holonomic and nonholonomic constraints is presented. This is important since there is some particularities on the implementation that must be taken into account, both in software as in hardware. The most important are the changes to the control scheme when using force feedback for nonholonomic constraints.

1.4 Contributions

The contributions of this work are summarized as follows:

- A teleoperation control approach for the study of virtual reality systems with haptic feedback was addressed. Such approach resulted effective when considering the virtual environment as the dynamic model of a virtual robot in constrained motion.
- The position-force control scheme was successfully implemented to give the operator visual and haptic feedback both in free and constrained motion.
- Taking as basis the seminal ideas presented by Webster III et al. [2006] and Faurling et al. [2007], the use of holonomic and nonholonomic constraints was verified to render forces from non-penetrable and penetrable surfaces respectively.

- To the best of the author's knowledge, this is the first time that nonholonomic constraints have been used to give the operator the tactil sensation of being in contact with a non-penetrable virtual object.
- The suitability of the approach was experimentally tested successfully. The differences between using holonomic and nonholonomic constraints were identified both for the control scheme as for changes in the hardware.

1.5 Dissertation Structure

The present document is organized as follows. In Chapter 2, a mathematical model for the virtual teleoperation system is presented as well as some of its properties. A general overview of holonomic and nonholonomic constraints representation is also introduced. Chapter 3 presents the position–force controller design and its integration with the graphic application. In Chapter 4, the experimental setup and results are shown to illustrate the effectiveness of the proposed approach. Finally, some conclusions and directions for future work are given in Chapter 5.

Chapter 2

Preliminaries

To implement a virtual reality application it is important to combine and to match both visual and haptic feedback in real-time. In this chapter, the fundamentals of virtual surfaces representation are introduced with the aim of showing how the method to render forces proposed in this thesis can be effectively applied in a practical way. Additionally, the mathematical nature of holonomic and nonholonomic constraints is presented by introducing some basic concepts of differential topology as well as the mathematical model of the haptic system. To do so, a teleoperation scheme is adapted by considering the slave robot as virtual, *i.e.*, its dynamic model is simulated within the virtual environment. Such robot is in contact with either the virtual holonomic or nonholonomic constraints, whose mathematical representation is also presented. Thereafter, the process to render the forces to be transmitted to the operator is detailed as well as some remarks about the validity of the approach, especially regarding the differences when using both types of constraints.

2.1 Virtual surfaces representation

It is important to differentiate between virtual surfaces representation both in graphic computing and haptic systems. Because the physics of light (in the case of visual representation) differs from the physics of mechanical interactions, it is important to take into account that although graphic and haptic simulation can share the encoding of certain properties, such as shape, they must differ in other aspect, such as models, mathematical techniques and implementation [Mavhash and Hayward, 2004]. It is important to highlight that the central objective of this thesis requires a focus on the haptic rendering, which leads to avoid in the practice,

the numerous and complex representations developed by the graphic computing community through the years. However, in this section the basic aspects of such representations are given with the aim of introducing the fundamentals of the haptic rendering method proposed.

The central idea of the research carried out arose from relating the holonomic and non-holonomic constraints, that mathematically have a kinematic basis, with the rigid and soft tissues present in medicine. Nevertheless, the complexity of the surfaces dynamics is avoided and instead classifying them as *non-penetrable* and *penetrable* from a purely haptic approach. Moreover, by using the approach of a manipulator in constrained motion, the dynamic of the virtual robot is included with the aim of exemplifying that it is possible to model the virtual tool, in a more complex way than that of a single point probe used commonly in computer graphics.

2.1.1 Non-penetrable virtual surfaces

To render forces from rigid surfaces it is sufficient to algebraically define an implicit equation in task-space coordinates or at least, two of its geometric characteristics (normal and tangential vectors or distance and angle relations between points, lines and planes) [Rodríguez et al., 2008]. This simplifies both graphic and haptic implementation since the surface is defined as the zero set of a function f valued in $\mathbb{R}$ as $\boldsymbol{S}_f = \{\boldsymbol{x} \in \mathbb{R}^3 | f(\boldsymbol{x}) = 0\}$ [Montagnat et al., 2001]. Based on this approach, the rigid virtual environments are usually composed of many lines, points and principally zero-width polygons assembled by vertices [Zilles and Salisbury, 1995] . For example, in Figure 2.1 a human skull is modeled using a mesh of triangles with the graphic engine NVIDIA Physx®.

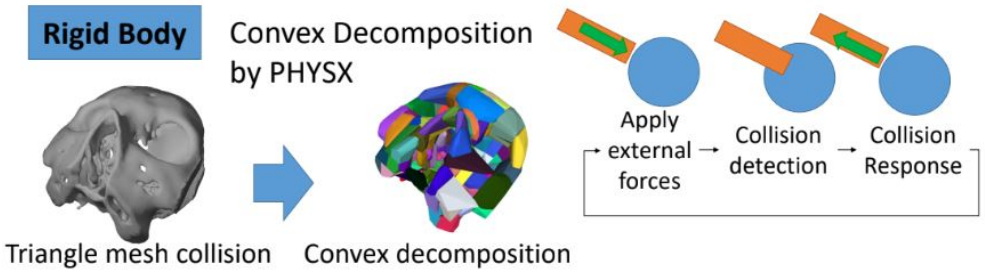

Figure 2.1: Human skull with Physx [Heredia et al., 2018].

In this case, a geometry class is used from the common base class, PxGeometry. Each geometry class defines a volume or surface with a fixed position and orientation. There are

many geometry types that can be implemented, as simple as spheres, capsules, boxes and planes and others more complex as convex meshes, triangle meshes and height fields [Heredia et al., 2018]. Besides, the methods to built a complex object as a human skull as well as its properties includes triangle mesh collision and convex decomposition. Nevertheless, the haptic interaction for virtual objects made up by such geometries and methods is obtained from one normal vector over the geometry types as can be appreciated in the right side of Figure 2.1, where the haptic interaction with a sphere occurs by means of a single-point collision detection algorithm.

2.1.2 Penetrable virtual surfaces

The case of deformable surfaces is more complex since, from a biomedical approach, a physically realistic simulation must take into account all the nonlinearities of a material deformation (*e.g.* stress, strain, elasticity and viscoelasticity). One strategy is to combine a finite element discretization of the geometry together with a finite difference discretization of time and an updated Lagrangian iterative scheme [Maurel et al., 1998]. Another very used representations of deformable surfaces in computer graphics are the particle-based models. Particles are described by their location, speed, acceleration, mass and any other parameter needed for a given application and they evolve according to Newtonian mechanical laws [Montagnat et al., 2001]. For example, in Figure 2.2 a deformable tissue is modeled using the graphic engine NVIDIA Flex® where, from a polygon-based mesh, a particle system is obtained through a Delaunay triangulation.

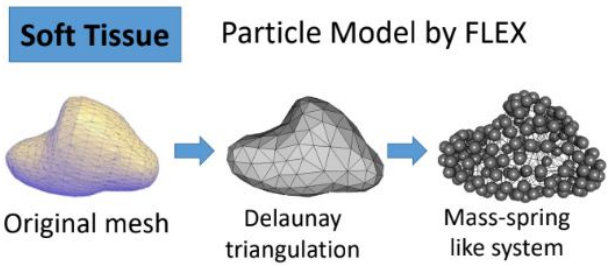

Figure 2.2: Soft tissue with Flex [Heredia et al., 2018].

Using particle-based models allows to visually reproduce more complex processes as cutting and indentation, very commons in medical training simulation. However, no matter how complicated the underlying model, the force response due to deformation is a function of

deflection only[1], even if a large deformation occurs [Mavhash and Hayward, 2004]. This characteristic allows to study the contact from a kinematic perspective, that is, the manner the tool is moved once inside the soft object.

More sophisticated graphic tools as SOFA (Simulation Open Framework Architecture) address the objects description typically by using three models: an internal model with independent degrees of freedom, the mass and the constitutive laws, a collision model with contact geometry, and a visual model with detailed geometry and rendering parameters [Faure et al., 2012]. During run-time, the models are synchronized using a generic mechanism called *mapping* to propagate forces and displacements and to enforce consistency in between, typically the internal model, acting as the master, imposes its displacements to slaves, typically the visual model and the collision model.

Let $\boldsymbol{f}$ be the function used to map the positions $\boldsymbol{x}_\mathrm{m}$ of a master model to the position $\boldsymbol{x}_\mathrm{s}$ of a slave model

$$\boldsymbol{x}_\mathrm{s} = \boldsymbol{f}(\boldsymbol{x}_\mathrm{m}). \tag{2.1}$$

The velocities are mapped in a similar way as

$$\dot{\boldsymbol{x}}_\mathrm{s} = \boldsymbol{J}(\boldsymbol{x}_\mathrm{m})\dot{\boldsymbol{x}}_\mathrm{m}, \tag{2.2}$$

where the Jacobian matrix $\boldsymbol{J}(\boldsymbol{x}_\mathrm{m}) = \frac{\partial \boldsymbol{x}_\mathrm{s}}{\partial \boldsymbol{x}_\mathrm{m}}$ encodes the linear relation between the master and slave velocities. In linear mappings, operators $\boldsymbol{f}$ and $\boldsymbol{J}(\boldsymbol{x}_\mathrm{m})$ are the same, otherwise $\boldsymbol{f}$ is nonlinear with respect to $\boldsymbol{x}_\mathrm{m}$ and it can not be written as a matrix.

Given forces $\boldsymbol{\lambda}_\mathrm{s}$ applied to a slave model, the mapping computes the equivalent forces $\boldsymbol{\lambda}_\mathrm{m}$ applied to its master. Since equivalent forces must have the same energy [Faure et al., 2012], the following relation holds

$$\dot{\boldsymbol{x}}_\mathrm{m}^\mathrm{T}\boldsymbol{\lambda}_\mathrm{m} = \dot{\boldsymbol{x}}_\mathrm{s}^\mathrm{T}\boldsymbol{\lambda}_\mathrm{s}. \tag{2.3}$$

The kinematic relation (2.2) allows to rewrite the equation (2.3) as

$$\dot{\boldsymbol{x}}_\mathrm{m}^\mathrm{T}\boldsymbol{\lambda}_\mathrm{m} = \boldsymbol{J}^\mathrm{T}(\boldsymbol{x}_\mathrm{m})\dot{\boldsymbol{x}}_\mathrm{m}^\mathrm{T}\boldsymbol{\lambda}_\mathrm{s}. \tag{2.4}$$

Since equation (2.4) holds for all possible velocities $\dot{\boldsymbol{x}}_\mathrm{m}$, the principle of virtual work allows us to simplify it to obtain

$$\boldsymbol{\lambda}_\mathrm{m} = \boldsymbol{J}^\mathrm{T}(\boldsymbol{x}_\mathrm{m})\boldsymbol{\lambda}_\mathrm{s}. \tag{2.5}$$

[1] It is defined as the displacement of the initial point of contact between an instrument and a deformable body

The kinematic mappings (2.1), (2.2) and (2.5) allows to compute displacements and to apply forces. They are also used to connect generalized coordinates such as joint angles to task-space geometries.

2.2 Geometry of a constrained submanifold

From a robot control approach, a tool in contact with an object can be considered as a robot in constrained motion. The constraints of this system are well defined if they are associated with physically realizable forces. This occurs, for example, in the case of an industrial robot in contact with a real surface like a car bonnet in a painting or welding task. But in the case of virtual environments, where surfaces do not exist, there are no physical constraint forces associated to them. Thus, the constraints are not well defined and they are called *virtual constraints* [Selig, 1996]. In the context of this thesis, the non-penetrable and penetrable virtual surfaces are mathematically addressed as *virtual constraints*. In this sense, it is important to introduce the geometric properties of such constraints in order to define them as either holonomic or nonholonomic.

Let $\mathcal{Q}$ be the n–dimensional smooth manifold configuration space of an unconstrained manipulator and $\boldsymbol{q} \in \mathbb{R}^n$ its local generalized coordinates. The tangent space to $\mathcal{Q}$ at $\boldsymbol{q}$, denoted $\mathcal{T}_{\boldsymbol{q}}\mathcal{Q}$ consists of all generalized velocity vectors $\dot{\boldsymbol{q}} \in \mathbb{R}^n$ of the system.

Definition 2.1. A *geometric constraint* on $\mathcal{Q}$ is a relation of the form

$$\boldsymbol{h}_i(\boldsymbol{q}) = \boldsymbol{0}, \qquad i = 1, \ldots, k < n, \tag{2.6}$$

where $\boldsymbol{h}_i : \mathcal{Q} \mapsto \mathbb{R}$ limits the admissible motions of the system to a $(n-k)$-dimensional smooth submanifold of $\mathcal{Q}$. △

Those constraints involving not only the generalized coordinates but also their first derivatives in the form

$$\boldsymbol{a}_i(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{0}, \qquad i = 1, \ldots, k < n, \tag{2.7}$$

with $\boldsymbol{a}_i(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathcal{T}_{\boldsymbol{q}}\mathcal{Q}$, are referred to as *kinematic constraints*. They limit the admissible motions of the manipulator to a $(n-k)$-dimensional smooth submanifold of $\mathcal{Q}$ by restricting the set of generalized velocities that can be attained at a given configuration.

Definition 2.2. A *Pfaffian constraint* on $\mathcal{Q}$ is a set of k kinematic constraints linear in velocity of the form

$$\boldsymbol{a}_i^{\mathrm{T}}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{0}, \qquad i = 1,\ldots,k < n, \tag{2.8}$$

where $\boldsymbol{a}_i : \mathcal{Q} \mapsto \mathbb{R}^n$ are assumed to be smooth and linearly independent. △

A kinematic constraint can be integrable, that is, there are k real valued functions $\boldsymbol{h}_i(\boldsymbol{q})$ such that

$$\frac{\partial \boldsymbol{h}_i(\boldsymbol{q})}{\partial \boldsymbol{q}} = \boldsymbol{a}_i^{\mathrm{T}}(\boldsymbol{q}), \qquad i = 1,\ldots,k < n. \tag{2.9}$$

In this case, the kinematic constraints are, in fact, geometric constraints. The set of Pfaffian constraints $\boldsymbol{a}_i(\boldsymbol{q})$ is called holonomic if it is integrable, that is, the system has a geometric limitation. For example, consider a set of holonomic constraints that can be characterized by

$$\boldsymbol{\varphi}_i(\boldsymbol{q}) = \boldsymbol{0}, \qquad i = 1,\ldots,k < n. \tag{2.10}$$

By taking into equation (2.9), the following is obtained

$$\frac{\partial \boldsymbol{\varphi}_i(\boldsymbol{q})}{\partial \boldsymbol{q}} = \boldsymbol{J}_{\varphi}^{\mathrm{T}}(\boldsymbol{q}) \tag{2.11}$$

where $\boldsymbol{J}_{\varphi}(\boldsymbol{q}) \in \mathbb{R}^{k\times n}$ is the Jacobian of the holonomic constraint. Therefore, holonomic constraints are characterized by equivalent equations in terms of position variables, *i.e.* they can be integrated to position equations if initially described by velocity equations [Xiaoping and Sarkar, 1998].

Property 2.1. Given n generalized coordinates $\boldsymbol{q}$ in a submanifold $\mathcal{Q}$ and k holonomic constraints, the tangent space to $\mathcal{Q}$ at a given configuration can be described by properly defining $(n-k)$ new generalized coordinates of the restricted submanifold that characterize the actual degrees of freedom of the system [Murray et al., 1994]. △

The set of Pfaffian constraints $\boldsymbol{a}_i(\boldsymbol{q})$ is called nonholonomic if it is non integrable, that is, the system has a kinematic limitation. Assuming again that the vectors $\boldsymbol{a}_i : \mathcal{Q} \mapsto \mathbb{R}^n$ are smooth and linearly independent, the nonholonomic constraints can be expressed as

$$\boldsymbol{A}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{0} \tag{2.12}$$

where $\boldsymbol{A}(\boldsymbol{q}) \in \mathbb{R}^{k\times n}$ is the Pfaffian matrix of nonholonomic constraints and which image space

produces forces to ensure that the system does not move in those directions. The presence of these constraints limits the system mobility in a completely different way if compared to holonomic ones: even if its generalized velocities at each point are constrained to an $(n-k)$-dimensional submanifold space, it is still possible to reach any configuration in $\mathcal{Q}$.

Property 2.2. Given n generalized coordinates $\boldsymbol{q}$ in a submanifold $\mathcal{Q}$ and k nonholonomic constraints, the tangent space to $\mathcal{Q}$ at a given configuration has $(n-k)$ degrees of freedom but the number of generalized coordinates cannot be reduced [Luca and Oriolo, 1995]. △

Remark 2.1. It is assumed that nonholonomic constraints are given by velocity–level equation (2.12) and holonomic constraints are described by position–level equation (2.10). In practical problems, both types of constraints may be described as velocity-level equations. △

2.2.1 Integrability of the constraints

A *vector field* $\boldsymbol{g} : \mathbb{R}^n \mapsto \mathcal{T}_{\boldsymbol{q}}\mathbb{R}^n$ is a smooth mapping assigning to each point $\boldsymbol{q} \in \mathbb{R}^n$ a tangent vector $\boldsymbol{g}(\boldsymbol{q}) \in \mathcal{T}_{\boldsymbol{q}}\mathbb{R}^n$. In local coordinates, $\boldsymbol{q}$ can be represented as a column vector whose elements depend on $\boldsymbol{q}$ as

$$\boldsymbol{g}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{g}_1(\boldsymbol{q}) \\ \vdots \\ \boldsymbol{g}_n(\boldsymbol{q}) \end{bmatrix}, \tag{2.13}$$

where $\boldsymbol{g}$ is smooth if each $\boldsymbol{g}_i(\boldsymbol{q})$ is smooth.

Given $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$, the *Lie bracket* of this vectors fields is defined as

$$[\boldsymbol{g}_1, \boldsymbol{g}_2] = \frac{\partial \boldsymbol{g}_2}{\partial \boldsymbol{q}} \boldsymbol{g}_1 - \frac{\partial \boldsymbol{g}_1}{\partial \boldsymbol{q}} \boldsymbol{g}_2, \tag{2.14}$$

where $[\boldsymbol{g}_1, \boldsymbol{g}_2]$ is a new vector field.

A *distribution* assigns a subspace of the tangent space to each point in $\mathbb{R}^n$ in a smooth way. A special case is a distribution defined by a set of smooth vector fields, $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_m$. In this case a distribution can be defined as

$$\boldsymbol{\Delta} = \operatorname{span}\{\boldsymbol{g_1}, \ldots, \boldsymbol{g}_m\}, \tag{2.15}$$

where the span over the set of smooth real-valued function on $\mathbb{R}^n$ is taken. Evaluated at any

point $\boldsymbol{q} \in \mathbb{R}^n$, the distribution defines a linear subspace of the tangent space

$$\boldsymbol{\Delta}_{\boldsymbol{q}} = \text{span}\{\boldsymbol{g_1}(\boldsymbol{q}), \ldots, \boldsymbol{g}_m(\boldsymbol{q})\} \subset \mathcal{T}_{\boldsymbol{q}}\mathbb{R}^n. \tag{2.16}$$

A distribution is *involutive* if it is closed under the Lie bracket, *i.e.*,

$$[\boldsymbol{g}_i, \boldsymbol{g}_j] \in \boldsymbol{\Delta}, \quad \forall \boldsymbol{g}_i, \boldsymbol{g}_j \in \boldsymbol{\Delta}. \tag{2.17}$$

Moreover, a distribution $\boldsymbol{\Delta}$ of dimension k is said to be *integrable* if for every point $\boldsymbol{q} \in \mathbb{R}^n$, there exist a set of smooth functions $\boldsymbol{h}_i : \mathbb{R}^n \mapsto \mathbb{R}$ for $i = 1, \ldots, n-k$ such that the row vectors $\frac{\partial \boldsymbol{h}_i}{\partial \boldsymbol{q}}$ are linearly independent at $\boldsymbol{q}$ and for every $\boldsymbol{g} \in \boldsymbol{\Delta}$

$$\frac{\partial \boldsymbol{h}_i}{\partial \boldsymbol{q}} \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0}, \qquad i = 1, \ldots n-k. \tag{2.18}$$

The hypersurfaces defined by the level sets $\{\boldsymbol{h}_1(\boldsymbol{q}) = c_1, \ldots, \boldsymbol{h}_{n-k}(\boldsymbol{q}) = c_{n-k}\}$ are called *integral manifolds* for the distribution $\boldsymbol{\Delta}$. Equation (2.18) indicates that $\boldsymbol{\Delta}$ coincides with the tangent space to its integral manifold at $\boldsymbol{q}$.

Integral manifolds are related to involutive distributions by the following so-called Frobenious theorem [Murray et al., 1994].

Theorem 2.1. *A distribution is integrable if and only if it is involutive.*

This theorem gives a necessary and sufficient condition for the complete integrability of a distribution. Thus, if $\boldsymbol{\Delta}$ is a k-dimensional involutive distribution, then locally there exist $n-k$ functions $\boldsymbol{h}_i : \mathbb{R}^n \mapsto \mathbb{R}$ such that integral manifolds of $\boldsymbol{\Delta}$ are given by the level surfaces $\boldsymbol{h} = (\boldsymbol{h}_1, \ldots, \boldsymbol{h}_{n-k})$.

The result of the mentioned above gives conditions for the integrability of a set of kinematic constraints in the following proposition [Luca and Oriolo, 1995].

Proposition 2.1. *The set of k Pfaffian constraints (2.8) is holonomic if and only if its distribution $\boldsymbol{\Delta}$ is involutive.*

Then, it is possible to establish when a Pfaffian constraints is nonholonomic by verifying if its distribution is non involutive.

2.3 Haptic system overview

A common practice of the graphics computing community has been to directly associate the position and orientation of virtual tools with that of the haptic interface. This assumption is based on the fact that some real tools have negligible dynamic, as the case of an scalpel in medical simulation. Conversely, from a teleoperation approach, it is assumed that even the simplest tool has some dynamic properties to be considered in the virtual environment. In this section a description of this proposal is presented, both mathematically and intuitively.

To describe the operation of the haptic system, two independent sets of task space coordinates are considerated as shown in Figure 2.3. The operator manipulates the haptic interface, *i.e.*, the master robot in the real environment and whose Cartesian coordinates are denoted as $\boldsymbol{x}_{\mathrm{m}} \in \mathrm{SE}(3)$, where $\boldsymbol{p}_{\mathrm{m}} \in \mathbb{R}^3$ is the end-effector position and $\boldsymbol{R}_{\mathrm{m}} \in \mathrm{SO}(3)$ its orientation. On the other hand, the virtual tool must respond to the movements of such interface in the virtual environment with Cartesian coordinates $\boldsymbol{x}_{\mathrm{v}} \in \mathrm{SE}(3)$, where $\boldsymbol{p}_{\mathrm{v}} \in \mathbb{R}^3$ is the virtual tool position and $\boldsymbol{R}_{\mathrm{v}} \in \mathrm{SO}(3)$ its orientation. In a teleoperation context, the position of the master robot

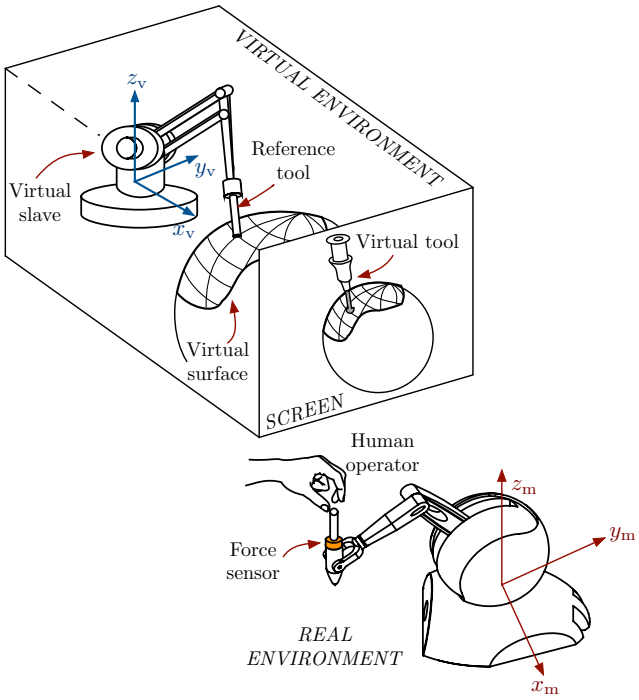

Figure 2.3: Haptic system

acts as a reference for the virtual tool and it is projected visually on the screen by means of

the virtual avatar of the system. The operator moves freely the virtual tool until the collision detection algorithm indicates that a contact with the virtual surface is taking place. At that moment, the master robot exerts a force which is measured by a force sensor that serves as a reference for the virtual robot and must be applied on such surface. By closing the feedback loop, the control algorithm produces a tactile sensation on the operator. Ideally, both visual and haptic feedback must coincide, allowing the operator to have a visual reference of the virtual tool and the feeling of the dynamic changes of its contact with the virtual surface.

In a similar approach, Faurling *et al.* [Faurling et al., 2007] make evident that the virtual environment can be represented by a set of generalized coordinates $\boldsymbol{q}_{\mathrm{v}} \in \mathbb{R}^3$, which are related to the task–space coordinates of the master robot by a nonlinear kinematic equation

$$\boldsymbol{x}_{\mathrm{m}} = \boldsymbol{f}(\boldsymbol{q}_{\mathrm{v}}), \tag{2.19}$$

which is a mapping between the real and virtual environment, similar to that of equation (2.1) but where the former acts as master model.

The set of coordinates $\boldsymbol{q}_{\mathrm{v}}$ allows the dynamic model of the virtual tool to be described in terms of Euler-Lagrange equations of motion. Moreover, a set of holonomic or nonholonomic constraints that represent the virtual surface can be embedded into the kinematic mapping (2.19), relating independent master robot task-space coordinates and dependent virtual robot task-space coordinates. The virtual tool moves according to the physic simulation propagated in the virtual environment coordinates and always satisfies such constraints.

2.3.1 Dynamic model and properties

Consider a real master (m) and a virtual slave (v) robot system composed by two manipulators each of them with n degrees of freedom but not necessarily with the same kinematic configuration. Each robot spans a task space of dimension k and depending on the master/virtual devices they might be scaled to fulfill the intended virtual application. The master dynamics is given by

$$\boldsymbol{H}_{\mathrm{m}}(\boldsymbol{q}_{\mathrm{m}})\ddot{\boldsymbol{q}}_{\mathrm{m}} + \boldsymbol{C}_{\mathrm{m}}(\boldsymbol{q}_{\mathrm{m}}, \dot{\boldsymbol{q}}_{\mathrm{m}})\dot{\boldsymbol{q}}_{\mathrm{m}} + \boldsymbol{D}_{\mathrm{m}}\dot{\boldsymbol{q}}_{\mathrm{m}} + \boldsymbol{g}_{\mathrm{m}}(\boldsymbol{q}_{\mathrm{m}}) = \boldsymbol{\tau}_{\mathrm{m}} - \boldsymbol{\tau}_{\mathrm{h}} \tag{2.20}$$

while the virtual slave dynamics is modeled by

$$\boldsymbol{H}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}})\ddot{\boldsymbol{q}}_{\mathrm{v}} + \boldsymbol{C}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}, \dot{\boldsymbol{q}}_{\mathrm{v}})\dot{\boldsymbol{q}}_{\mathrm{v}} + \boldsymbol{D}_{\mathrm{v}}\dot{\boldsymbol{q}}_{\mathrm{v}} + \boldsymbol{g}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}) = \boldsymbol{\tau}_{\mathrm{v}} + \boldsymbol{\tau}_{\mathrm{s}}, \tag{2.21}$$

where the subscripts m and v denote the real master and the virtual slave manipulators respectively. For $i = \mathrm{m,v}$, $\boldsymbol{q}_i \in \mathbb{R}^n$ is the vector of generalized coordinates, $\boldsymbol{H}_i(\boldsymbol{q}_i) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $\boldsymbol{C}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i)\dot{\boldsymbol{q}}_i \in \mathbb{R}^n$ is the vector of Coriolis and centripetal forces, $\boldsymbol{D}_i \in \mathbb{R}^{n\times n}$ is a diagonal matrix of viscous friction coefficients, $\boldsymbol{g}_i(\boldsymbol{q}_i) \in \mathbb{R}^n$ is the vector of gravitational torques, $\boldsymbol{\tau}_i \in \mathbb{R}^n$ is the vector of generalized inputs, $\boldsymbol{\tau}_\mathrm{h} \in \mathbb{R}^n$ is the real torque applied by the human operator on the master side and $\boldsymbol{\tau}_\mathrm{s} \in \mathbb{R}^n$ is the virtual torque generated due to the contact with the virtual constraint.

Property 2.3. With a proper definition of the robot parameters, it is possible to express the dynamics of the robot as

$$\boldsymbol{H}_i(\boldsymbol{q}_i)\ddot{\boldsymbol{q}}_i + \boldsymbol{C}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i)\dot{\boldsymbol{q}}_i + \boldsymbol{D}_i\dot{\boldsymbol{q}}_i + \boldsymbol{g}_i(\boldsymbol{q})_i = \boldsymbol{Y}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i, \ddot{\boldsymbol{q}}_i)\boldsymbol{\theta}_i \tag{2.22}$$

where $\boldsymbol{Y}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i, \ddot{\boldsymbol{q}}_i) \in \mathbb{R}^{n_i\times l}$ is the regressor and $\boldsymbol{\theta}_i \in \mathbb{R}^l$ is a constant vector of parameters. △

Assumption 2.1. *The master and the virtual slave robots share the same geometric structure but they do not necessarily have the same dynamic model parameters, i.e. none of the different model matrices and vectors in (2.20) and (2.21) need to be equal.* △

External torques are acting in both robots, either the real torque $\boldsymbol{\tau}_\mathrm{h}$ applied by the human on the master side or the virtual torque $\boldsymbol{\tau}_\mathrm{s}$ generated due to the contact between the virtual robot and the virtual surface. The torque applied by the human operator can be defined as

$$\boldsymbol{\tau}_\mathrm{h} = \boldsymbol{J}_\mathrm{m}^\mathrm{T}(\boldsymbol{q}_\mathrm{m})\boldsymbol{F}_\mathrm{h}, \tag{2.23}$$

where $\boldsymbol{F}_\mathrm{h} \in \mathbb{R}^3$ is the force applied by the operator in task-space coordinates and $\boldsymbol{J}_\mathrm{m} \in \mathbb{R}^{3\times n}$ is the geometric Jacobian of the master manipulator. In the same way, the torque applied on the virtual surface can be expressed as

$$\boldsymbol{\tau}_\mathrm{s} = \boldsymbol{J}_\mathrm{v}^\mathrm{T}(\boldsymbol{q}_\mathrm{v})\boldsymbol{F}_\mathrm{s}, \tag{2.24}$$

where $\boldsymbol{F}_\mathrm{s} \in \mathbb{R}^3$ is the force applied on such surface in task-space coordinates.

2.3.2 Virtual holonomic constraints

In the case of holonomic constraints it is assumed that, in virtual task space coordinates, the virtual robot is subject to k virtual holonomic constraints characterized by

$$\boldsymbol{\varphi}_{\mathrm{v}}(\boldsymbol{x}_{\mathrm{v}}) = \mathbf{0}, \tag{2.25}$$

where a suitable normalization is done for the gradient of this constraint, $\boldsymbol{J}_{\varphi \mathrm{xv}}(\boldsymbol{x}_{\mathrm{v}}) = \nabla \boldsymbol{\varphi}_{\mathrm{v}}(\boldsymbol{x}_{\mathrm{v}}) \in \mathbb{R}^{k \times n}$, to be unitary.

The representation of constraint (2.25) in generalized virtual coordinates is

$$\boldsymbol{\varphi}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}) = \mathbf{0}, \tag{2.26}$$

where $\boldsymbol{q}_{\mathrm{v}} \in \mathbb{R}^n$ is the vector of the virtual robot end-effector joint coordinates. The gradient of the constraint (2.26) is $\boldsymbol{J}_{\varphi \mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}) = \nabla \boldsymbol{\varphi}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}) \in \mathbb{R}^{k \times n}$. These two gradients are related by

$$\boldsymbol{J}_{\varphi \mathrm{xv}}(\boldsymbol{q}_{\mathrm{v}}) = \boldsymbol{J}_{\varphi \mathrm{xv}}(\boldsymbol{x}_{\mathrm{v}}) \boldsymbol{J}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}), \tag{2.27}$$

where $\boldsymbol{J}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}) \in \mathbb{R}^{n \times n}$ is the geometric Jacobian of the virtual manipulator. Hence, the torque due to the contact with the virtual surface in (2.21) can be defined as

$$\boldsymbol{\tau}_{\mathrm{s}} = \boldsymbol{J}_{\varphi \mathrm{v}}^{\mathrm{T}}(\boldsymbol{q}_{\mathrm{v}}) \boldsymbol{\lambda}_{\mathrm{v}}, \tag{2.28}$$

where $\boldsymbol{\lambda}_{\mathrm{v}} \in \mathbb{R}^k$ is a vector of Lagrange multipliers that represents the virtual force applied over the surface. Then, it is possible to rewrite the whole equation (2.21) as

$$\boldsymbol{H}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}) \ddot{\boldsymbol{q}}_{\mathrm{v}} + \boldsymbol{C}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}, \dot{\boldsymbol{q}}_{\mathrm{v}}) \dot{\boldsymbol{q}}_{\mathrm{v}} + \boldsymbol{D}_{\mathrm{v}} \dot{\boldsymbol{q}}_{\mathrm{v}} + \boldsymbol{g}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}) = \boldsymbol{\tau}_{\mathrm{v}} + \boldsymbol{J}_{\varphi \mathrm{v}}^{\mathrm{T}}(\boldsymbol{q}_{\mathrm{v}}) \boldsymbol{\lambda}_{\mathrm{v}}.$$

According to Property 2.1 The virtual holonomic constraints (2.26) reduces the number of degrees of freedom of the virtual robot and the dimension of its configuration space to an $(n-k)$-dimensional submanifold [Luca and Oriolo, 1995].

2.3.3 Virtual nonholonomic constraints

In the case of nonholonomic constraints, it is well-known that they cannot be expressed as a function of only the generalized coordinates as in (2.25) or (2.26). Instead, they are commonly

expressed as Pfaffian constraints. In the present case, these kind of constraints are written more intuitively in terms of the virtual end-effector velocities $\boldsymbol{v}_\mathrm{v} = \begin{bmatrix} \dot{\boldsymbol{p}}_\mathrm{v} & \boldsymbol{\omega}_\mathrm{v} \end{bmatrix}^\mathrm{T}$ as

$$\boldsymbol{A}_\mathrm{v}(\boldsymbol{x}_\mathrm{v})\,\boldsymbol{v}_\mathrm{v} = \boldsymbol{0}\,, \tag{2.29}$$

where $\dot{\boldsymbol{p}}_\mathrm{v}, \boldsymbol{\omega}_\mathrm{v} \in \mathbb{R}^3$ are the linear and angular velocities of the virtual end-effector and $\boldsymbol{A}_\mathrm{v}(\boldsymbol{x}_\mathrm{v}) \in \mathbb{R}^{k\times n}$ is the corresponding Pfaffian constraint matrix. If the dynamic equations are defined in the virtual joint-space coordinates $\boldsymbol{q}_\mathrm{v}$, these constraints are projected via Faurling et al. [2007]

$$\boldsymbol{A}_\mathrm{v}(\boldsymbol{q}_\mathrm{v}) = \boldsymbol{A}_\mathrm{v}(\boldsymbol{x}_\mathrm{v})\,\boldsymbol{J}_\mathrm{v}(\boldsymbol{q}_\mathrm{v})\,. \tag{2.30}$$

Assuming that the virtual robot is subject to k velocity-level equations of nonholonomic constraints characterized by

$$\boldsymbol{A}_\mathrm{v}(\boldsymbol{q}_\mathrm{v})\,\dot{\boldsymbol{q}}_\mathrm{v} = \boldsymbol{0}\,, \tag{2.31}$$

the torque due to the contact with the virtual environment in (2.21) can be expressed as

$$\boldsymbol{\tau}_\mathrm{s} = \boldsymbol{A}_\mathrm{v}^\mathrm{T}(\boldsymbol{q}_\mathrm{v})\,\boldsymbol{\lambda}_\mathrm{v}, \tag{2.32}$$

where $\boldsymbol{\lambda}_\mathrm{v} \in \mathbb{R}^k$ is the vector of Lagrange multipliers which determines the magnitude of the constraint forces over the virtual surface. Then, it is possible to rewrite equation (2.21) as

$$\boldsymbol{H}_\mathrm{v}(\boldsymbol{q}_\mathrm{v})\,\ddot{\boldsymbol{q}}_\mathrm{v} + \boldsymbol{C}_\mathrm{v}(\boldsymbol{q}_\mathrm{v}, \dot{\boldsymbol{q}}_\mathrm{v})\,\dot{\boldsymbol{q}}_\mathrm{v} + \boldsymbol{D}_\mathrm{v}\dot{\boldsymbol{q}}_\mathrm{v} + \boldsymbol{g}_\mathrm{v}(\boldsymbol{q}_\mathrm{v}) = \boldsymbol{\tau}_\mathrm{v} + \boldsymbol{A}_\mathrm{v}^\mathrm{T}(\boldsymbol{q}_\mathrm{v})\,\boldsymbol{\lambda}_\mathrm{v}.$$

The nonholonomic constraints reduce the number of available degrees of freedom of the virtual robot to an $(n-k)$-dimensional submanifold, but they do not reduce the dimension of its configuration space [Luca and Oriolo, 1995].

Chapter 3

Implementation of the system

In this chapter the theoretical and practical aspects of implementing a virtual reality system that validates the virtual constraints approach proposed in this dissertation is presented. The principal aspect concerned is the design of a controller capable to perform an accurate haptic feedback that make to feel the operator to be in contact with either a penetrable or non-penetrable virtual surface. Additionally, the methodology to visually reproduce the virtual tool in contact with the virtual objects is presented. It is important to highlight that such methodology avoids the complexity of the virtual environments currently implemented in medical training simulators. However, the basic aspects addressed are sufficient to show that the virtual constraints approach can be used in a practical way and eventually be adapted to sophisticated graphic computing tools.

3.1 Design of the virtual environment

As mentioned in Chapter 2, the important aspect to obtain a realistic haptic feedback from a surface embedded into a virtual environment consists on defining its geometry. In Figure 3.1 an idealized representation of a virtual point probe in contact with either a non-penetrable or penetrable virtual surface is shown. In the first case, it is assumed that the contact arises in a single point over the surface from where the virtual probe cannot move forward, *i.e.*, its velocity is equal to zero. Therefore, a normal force vector, which magnitude increases depending on the force applied by the operator, avoids motion. In a robot control context, if it is assumed that the probe is attached to a robot manipulator's end-effector, according with Property 2.1,

the number of degrees of freedom of the system is reduced once in contact with the surface. In an intuitive way, that means that the virtual probe cannot move forward from where the contact arises, which can be any point on the surface. Actually, if the virtual object is built up by using a polygonal method, there will be a set of surfaces $(\boldsymbol{\varphi}_0, \boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_n)$ joined by vertices as shown in Figure 3.1. The best way to find the place of the contact point (which belongs to a set of points defining each surface) is by establishing an implicit equation $\boldsymbol{\varphi}_{\mathrm{v}}(\boldsymbol{x}_{\mathrm{v}})$ containing such point. Therefore, the set of points defining the virtual surface can be expressed as $\boldsymbol{\varphi}_{\mathrm{v}}(\boldsymbol{x}_{\mathrm{v}}) = \mathbf{0}$, which coincides with the holonomic constraint of equation (2.25) in Cartesian coordinates or equation (2.26) in generalized coordinates. From those expressions, a collision detection algorithm can be established by defining the following conditions:

- If $\boldsymbol{\varphi}_{\mathrm{v}}(\boldsymbol{x}_{\mathrm{v}}) > \mathbf{0}$, the virtual probe is in free motion *i.e.*, it is not in contact with the virtual surface.
- If $\boldsymbol{\varphi}_{\mathrm{v}}(\boldsymbol{x}_{\mathrm{v}}) = \mathbf{0}$ the virtual probe is in contact with the virtual surface and it stay over the surface only, staying in constrained motion.
- If $\boldsymbol{\varphi}_{\mathrm{v}}(\boldsymbol{x}_{\mathrm{v}}) < \mathbf{0}$ the virtual probe is in constrained motion but the constraint is violated *i.e.*, it is inside of the virtual surface.

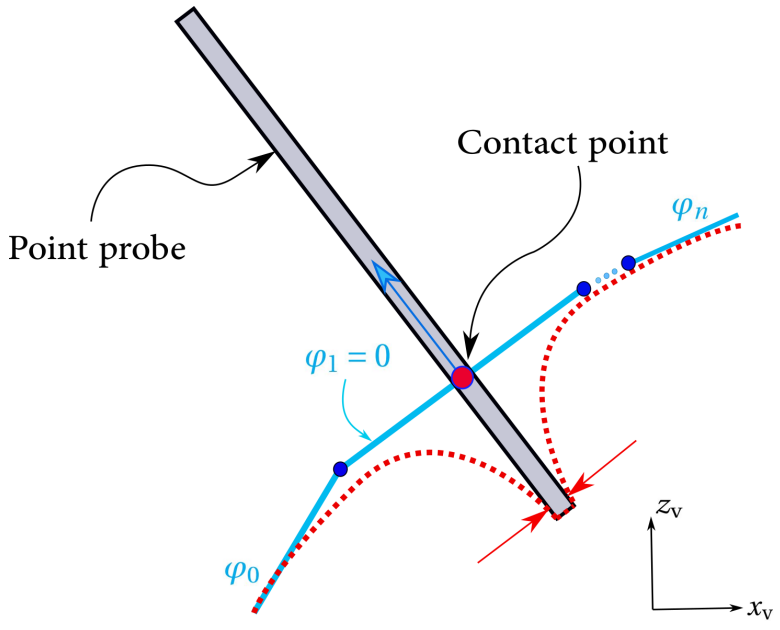

Figure 3.1: Penetrable (···) and non-penetrable (—) surfaces.

For the virtual reality system it is important to remember that the vector $\boldsymbol{x}_v$ represents the virtual robot's end-effector position. In this sense, the virtual probe would share such position. By extending the approach proposed, the virtual robot acts in fact as the virtual tool and its position is projected to the operator by means of the avatar of the system.

In the second case, for a non-penetrable constraint, even when de contact starts in a single point, the properties of the surface allows the virtual probe to stay in motion as can be seen in Figure 3.1. Nevertheless, in an intuitive way, the process is more complex since in a certain moment the virtual probe must stop. In a medical context, that means the deformable tissue have a limited resistance that depends on its elastic properties. Since the objective is to obtain a reaction force that depends on the motion of the virtual probe once inside of the deformable object, it is necessary to properly describe such motion. Unlike the non-deformable surfaces, the force does not arise normal to the surface in a single point, but lateral forces occur when the operator tries to move the virtual probe in such directions. Ideally, this would be true for any method to represent soft tissues including finite element meshes and particle-based models.

In Figure 3.2 the motion of a virtual tool inside a virtual object is shown. For an easy visualization, the motion is shown in 2D but during the simulation it must be reproduced in 3D with the aim of increase the realism of the application by improving the operator's dexterity. The contact begins in stage A where the virtual tool penetrates the object by following a straight trajectory, represented by a blue arrow, to reach the position in the stage B. Moreover, it can follow other trajectories, represented by dashed red lines, to reach the position of stage D or C. However, due to the surrounding "tissue", the tool cannot be moved laterally since reaction forces, represented by red arrows, avoid it all along the trajectories. In contrast to what happens in the holonomic case, the virtual tool is allowed to stay in motion, *i.e.*, its velocity is different from zero until the operator stops voluntarily. The process described above is similar to the motion of a wheeled car in 2D, which is perfectly described by nonholonomic constraints. In fact, if a third dimension is added, Property 2.2 holds for all time, and the virtual tool can reach any point of the virtual object.

The trajectories of the virtual tool showed in Figure 3.2 are common in non-invasive surgical procedures. For example, in the simulator of Transurethral Resection of the Prostate of Figure 1.2, the medical trainee performs a straight trajectory that, in the first place, simulates the insertion of the resectoscope into the patient's penis. Once situated inside the virtual prostate, he/she needs to move the resectoscope in order to eliminate, by means of an in-

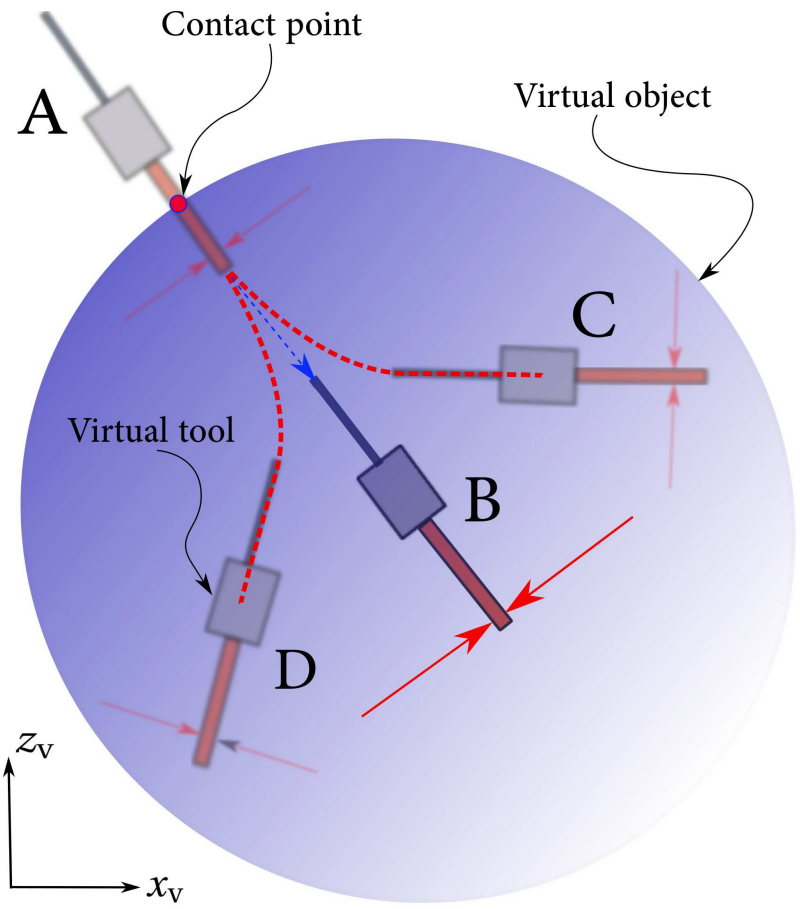

Figure 3.2: Motion in 2D of the virtual tool inside a virtual object.

candescent resection loop, the benign tissue that obstruct the flux of the urine to the urethra. Such motions follow a path similar to that represented with red dashed lines in Figure 3.2 and using a pivoting where the tool changes its direction.

The contribution of the approach proposed in this dissertation is that, in contrast with the common methods of single-point haptic feedback, forces avoiding lateral motion of the virtual tool are produced. However, the principal disadvantage is that the elastic properties of the surroundings cannot be considered. As a consequence, the force reaction limiting the operator motion, depending on such properties, is not reproduced and he/she can move the virtual tool indistinctly inside the virtual object, which does not occur in real life. For example, in human organs the elastic properties and parameters as the Young modulus or the Poisson ratio establish motion limits for the tool that, when they are exceeded, the tissue results damaged.

3.1.1 Virtual constraints modeling

In Figure 3.3 a scheme in 2D of the contact between a virtual tool and a virtual surface is shown to illustrate the use of the model given by equation (2.21). The tool is attached to the virtual robot's last DOF, acting as its end-effector. It is important to note that the manipulator dynamic model is used with the aim of reproducing a classic bilateral teleoperation system and assuming that, since it is simulated digitally, it can be exchanged by a simpler or more complex model, including those of medical instruments as forceps endoscopes, gripers and retractors. This assumption leads to the proposition that, if the model of a surgical tool is used during the simulation, the realism of the contact with the surface would increase.

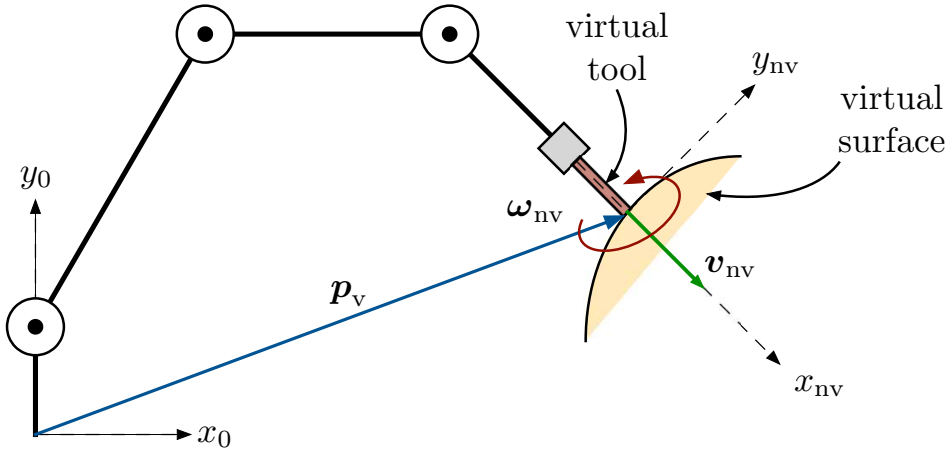

Figure 3.3: Virtual robot in interaction with a penetrable surface

The principal difference between defining a holonomic and a nonholonomic constraint is the need of an expression for $\boldsymbol{\varphi}_v(\boldsymbol{x}_v)$. Based on Section 2.1.1, from an implicit representation approach, rigid virtual objects are built from 3D basic geometric primitives as cones, pyramids, planes, cubes and spheres [Ruspini et al., 1997]. Ultimately, the base of a highly complex virtual environment composed by rigid objects is a set of basic geometric shapes that can be represented by means of mathematical expressions. Therefore, it is sufficient to define a zero set of functions as in (2.25) that individually are expressed as $\varphi_v(\boldsymbol{x}_v)$ and a collision detection algorithm based on the inequalities established above.

On the other hand, for nonholonomic constraints and considering again the virtual robot of the Figure 3.3, let ${}^0\boldsymbol{p}_v \in \mathbb{R}^3$ be the Cartesian position of the virtual robot end-effector and ${}^0\boldsymbol{R}_v \in SO(3)$ a rotation matrix which describes its orientation. Dividing this rotation matrix

into three column vectors as

$$ {}^0\boldsymbol{R}_{\mathrm{v}} = \begin{bmatrix} {}^0\boldsymbol{x}_{n\mathrm{v}} & {}^0\boldsymbol{y}_{n\mathrm{v}} & {}^0\boldsymbol{z}_{n\mathrm{v}} \end{bmatrix}, \tag{3.1} $$

for which each column represent a vector of the end-effector coordinate frame, described in the base frame. This allows to define Pfaffian constraints like (2.29) in an intuitive form, *i.e.*,

$$ \boldsymbol{A}_{\mathrm{v}}({}^0\boldsymbol{x}_{n\mathrm{v}}, {}^0\boldsymbol{y}_{n\mathrm{v}}, {}^0\boldsymbol{z}_{n\mathrm{v}})\,\boldsymbol{v}_{\mathrm{v}} = \boldsymbol{0}. \tag{3.2} $$

It is claimed that a set of nonholonomic constraints can be defined if the manipulator degrees of freedom is greater than those necessary to control the end-efector position, *i.e.*, $n > 2$ for planar robots and $n > 3$ for robots in a three dimensional workspace.

The end-effector velocities of the virtual robot can be described by

$$ \boldsymbol{v}_{\mathrm{v}} = \begin{bmatrix} {}^0\dot{\boldsymbol{p}}_{\mathrm{v}} & {}^0\omega_n \end{bmatrix}^{\mathrm{T}}, \tag{3.3} $$

where ${}^0\dot{\boldsymbol{p}}_{\mathrm{v}} = \begin{bmatrix} {}^0\boldsymbol{p}_{\mathrm{v}x} & {}^0\boldsymbol{p}_{\mathrm{v}y} \end{bmatrix}^{\mathrm{T}}$ is the linear velocity and ${}^0\omega_n$ is the angular velocity over an axis normal to the robot plane. If the robot is not allowed to move in the ${}^0\boldsymbol{y}_{n\mathrm{v}}$ direction, the corresponding Pfaffian constraint is given by

$$ \begin{bmatrix} {}^0\boldsymbol{y}_{n\mathrm{v}}^{\mathrm{T}} & 0 \end{bmatrix} \boldsymbol{v}_{\mathrm{v}} = \begin{bmatrix} -s_{123} & c_{123} & 0 \end{bmatrix} \begin{bmatrix} {}^0\dot{p}_{\mathrm{v}x} \\ {}^0\dot{p}_{\mathrm{v}y} \\ {}^0\omega_n \end{bmatrix}, \tag{3.4} $$

where $s_{123} = \sin(q_{\mathrm{v}1} + q_{\mathrm{v}2} + q_{\mathrm{v}3})$ and $c_{123} = \cos(q_{\mathrm{v}1} + q_{\mathrm{v}2} + q_{\mathrm{v}3})$. By choosing the distribution $\boldsymbol{\Delta} = [\boldsymbol{g}_1, \boldsymbol{g}_2]$, where $\boldsymbol{g}_1 = \begin{bmatrix} c_{123} & s_{123} & 0 \end{bmatrix}^{\mathrm{T}}$ and $\boldsymbol{g}_2 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^{\mathrm{T}}$, as a basis for the null-space of the Pfaffian matrix, the equivalent control system $\dot{\boldsymbol{q}}_{\mathrm{v}} = \boldsymbol{g}_1 u_1 + \boldsymbol{g}_2 u_2$ can be constructed, representing the directions of allowed motion [Murray et al., 1994, p. 320]. It is easy to verify that the Lie bracket is

$$ [\boldsymbol{g}_1, \boldsymbol{g}_2] = \begin{bmatrix} -s_{123} & c_{123} & 0 \end{bmatrix}^{\mathrm{T}}, \tag{3.5} $$

which shows the non-involutivity of the distribution and thus, establishes the nonholonomic nature of the constraints according to Proposition 2.1. Notice also that if the degrees of freedom were 2, the null-space would be of dimension 1, which is necessarily involutive, and the constraints would be holonomic.

3.2 Position–force controllers design

A correct haptic rendering largely depends on the force control algorithm. In classic haptic systems the common solution is to define indirect impedance or compliance control schemes. Conversely, in this chapter two hybrid-control algorithms for haptic interaction with virtual constrained systems are presented. As mentioned in Chapter 2, the usual practice is to directly associate the position of the haptic interface to that of the virtual avatar. Therefore, it is not necessary any position control scheme since the operator's movements are reflected in an exact way on the graphic application. However, in the proposed approach, the task space coordinates of the virtual environment depend on the correct tracking between the position of the haptic robot and that of the virtual one, *i.e.*, the control algorithm generates the virtual environment itself. This is due to the inclusion of the virtual robot dynamics and the fact that the operator should feel the virtual tool as a consequence of the masking effect. To address that, a control scheme used in teleoperation to achieve both position and force tracking is explored. A block diagram of such scheme is shown in Figure 3.4.

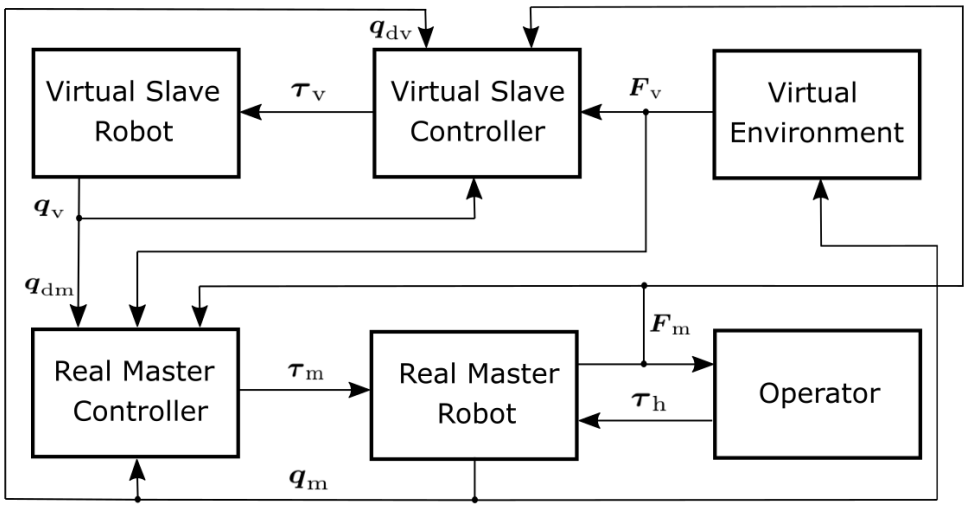

Figure 3.4: Block diagram of the proposed scheme.

Considering once again $i, j = \mathrm{m}, \mathrm{v}$, where $i \neq j$. Define

$$\boldsymbol{q}_{\mathrm{d}i}(t) \triangleq \boldsymbol{q}_j(t) \tag{3.6}$$

as the desired position trajectories, and

$$\dot{\boldsymbol{q}}_{\mathrm{d}i}(t) \triangleq \dot{\boldsymbol{q}}_j(t) \tag{3.7}$$

as the desired velocity trajectories, i.e., if $i = \mathrm{m}$, then $j = \mathrm{v}$ and vice versa. The corresponding tracking error is defined as

$$\Delta \boldsymbol{q}_i = \boldsymbol{q}_i - \boldsymbol{q}_{\mathrm{d}i}. \tag{3.8}$$

Based on Arteaga et al. [2006], it is proposed

$$\boldsymbol{s}_i = \dot{\boldsymbol{q}}_i - \dot{\boldsymbol{q}}_{\mathrm{d}i} + \boldsymbol{\Lambda}_{\mathrm{x}i} \Delta \boldsymbol{q}_i \tag{3.9}$$

and

$$\dot{\boldsymbol{\sigma}}_i = \boldsymbol{K}_{\beta i} \boldsymbol{s}_i + \mathrm{sign}(\boldsymbol{s}_i), \tag{3.10}$$

where $\boldsymbol{K}_{\beta i} \in \mathbb{R}^{n \times n}$ is a positive definite diagonal matrix and $\mathrm{sign}(\boldsymbol{s}_i) = [\ \mathrm{sign}(\boldsymbol{s}_{i1}), \ldots, \mathrm{sign}(\boldsymbol{s}_{in})]^{\mathrm{T}}$ with s_{ij} element of $\boldsymbol{s}_i$ for $j = 1, \ldots, n$. Consider now the velocity reference as

$$\dot{\boldsymbol{q}}_{\mathrm{r}i} = \dot{\boldsymbol{q}}_{\mathrm{r}i} + \boldsymbol{\Lambda}_{\mathrm{x}i} \Delta \boldsymbol{q}_i - \boldsymbol{K}_{\gamma i} \boldsymbol{\sigma}_i, \tag{3.11}$$

where $\boldsymbol{K}_{\gamma i} \in \mathbb{R}^{n \times n}$ is a positive definite diagonal matrix. Define also the auxiliary variable

$$\boldsymbol{s}_{\mathrm{a}i} = \dot{\boldsymbol{q}}_i - \dot{\boldsymbol{q}}_{\mathrm{r}i}. \tag{3.12}$$

Supposing that both robots are in free movement, for that case the control laws for the master and the virtual robots are proposed as

$$\boldsymbol{\tau}_{\mathrm{m}} = -\boldsymbol{K}_{\mathrm{am}} \dot{\boldsymbol{q}}_{\mathrm{m}} - \boldsymbol{K}_{\mathrm{pm}} \boldsymbol{s}_{\mathrm{am}} \tag{3.13}$$

$$\boldsymbol{\tau}_{\mathrm{v}} = \quad \boldsymbol{K}_{\mathrm{av}} \dot{\boldsymbol{q}}_{\mathrm{v}} + \boldsymbol{K}_{\mathrm{pv}} \boldsymbol{s}_{\mathrm{av}}, \tag{3.14}$$

respectively, where $\boldsymbol{K}_{\mathrm{am}}$, $\boldsymbol{K}_{\mathrm{av}}$, $\boldsymbol{K}_{\mathrm{pm}}$, $\boldsymbol{K}_{\mathrm{pv}} \in \mathbb{R}^{n \times n}$ are positive definite diagonal matrices.

3.2.1 Virtual holonomic constraints

Making an approximation to what happens during the tactile interaction of a point probe with a rigid surface, the one-dimensional case ($\boldsymbol{\varphi}_{\mathrm{v}} : \mathbb{R}^n \to \mathbb{R}$) is considered. As mentioned in

Section 3.1, that is ideally the normal force generated at a single point of contact where other reactions, as friction or tangential forces, can be omitted [Mavhash and Hayward, 2004]. To reproduce this effect, the implicit surface method is used, so that $\boldsymbol{\lambda}_\mathrm{v} = \lambda_\mathrm{v} \in \mathbb{R}$ represents the normal force of the virtual manipulator over the virtual surface. To reflect such contact force, a Lagrange multiplier is computed by the *Generic Penalty Method* as used by Gutiérrez-Giles and Arteaga-Pérez [2017], *i.e.*,

$$\lambda_\mathrm{v} = \alpha_\mathrm{v}\left(\ddot{\boldsymbol{\varphi}}_\mathrm{v}(\boldsymbol{q}_\mathrm{v}) + 2\xi\omega_\mathrm{n}\dot{\boldsymbol{\varphi}}_\mathrm{v}(\boldsymbol{q}_\mathrm{v}) + \omega_\mathrm{n}^2\boldsymbol{\varphi}_\mathrm{v}(\boldsymbol{q}_\mathrm{v})\right), \tag{3.15}$$

where $\xi, \omega_\mathrm{n} > 0$. Considering that force measurements are available at the master side in Cartesian coordinates and mapping the virtual force to this space as

$$\boldsymbol{F}_\mathrm{v} = \boldsymbol{J}_{\varphi\mathrm{xv}}^\mathrm{T}(\boldsymbol{x}_\mathrm{v})\lambda_\mathrm{v}, \tag{3.16}$$

a PID-like controller can be used for the virtual reality system. Consider that $\boldsymbol{F}_\mathrm{h} \in \mathbb{R}^3$ is the normal force component measured with a force sensor mounted at the master robot end-effector. After (3.15) and (2.23)–(2.24), a PI controller can be used for the virtual reality system. Define

$$\boldsymbol{F}_{\mathrm{d}i}(t) \triangleq \boldsymbol{F}_j(t), \tag{3.17}$$

as the desired force trajectory where if $i = \mathrm{h}$ then $j = \mathrm{v}$ and vice versa as stated before. The force tracking errors are

$$\Delta\boldsymbol{F}_i = \boldsymbol{F}_i - \boldsymbol{F}_{\mathrm{d}i} \tag{3.18}$$

and the corresponding integral, the momenta tracking error is

$$\Delta\boldsymbol{p}_i = \int_0^t \Delta\boldsymbol{F}_i \, \mathrm{d}t. \tag{3.19}$$

Note that the standard notation for momenta $\boldsymbol{p}$ is used although also the same notation is for position. It is claimed that there is no confusion because it always appears $\Delta\boldsymbol{p}$ for that case. Instead of (3.13)–(3.14), for the master and the virtual robot the corresponding control laws are respectively given by

$$\boldsymbol{\tau}_\mathrm{m} = \boldsymbol{Y}_\mathrm{m}(\boldsymbol{q}_\mathrm{m}, \dot{\boldsymbol{q}}_\mathrm{m}, \ddot{\boldsymbol{q}}_\mathrm{m})\boldsymbol{\theta}_\mathrm{m} - \boldsymbol{K}_\mathrm{am}\dot{\boldsymbol{q}}_\mathrm{m} - \boldsymbol{K}_\mathrm{pm}\boldsymbol{s}_\mathrm{am} + \boldsymbol{J}_\mathrm{m}^\mathrm{T}(\boldsymbol{q}_\mathrm{m})(\boldsymbol{F}_\mathrm{v} - \boldsymbol{K}_\mathrm{fm}\Delta\boldsymbol{p}_\mathrm{h}) \tag{3.20}$$

$$\boldsymbol{\tau}_\mathrm{v} = \boldsymbol{K}_\mathrm{av}\dot{\boldsymbol{q}}_\mathrm{v} + \boldsymbol{K}_\mathrm{pv}\boldsymbol{s}_\mathrm{av} - \boldsymbol{J}_\mathrm{v}^\mathrm{T}(\boldsymbol{q}_\mathrm{v})(\boldsymbol{F}_\mathrm{h} - \boldsymbol{K}_\mathrm{fv}\Delta\boldsymbol{p}_\mathrm{v}), \tag{3.21}$$

where $\boldsymbol{K}_{fi} \in \mathbb{R}^{n\times n}$ are diagonal matrices. In order for the operator to feel the virtual tool in contact with the virtual environment, a dynamic cancellation of the dynamics of the master manipulator can be carried out, as shown in equation (3.20).

3.2.2 Virtual nonholonomic constraints

In contrast with the holonomic case, when the constraints are nonholonomic, they cannot be defined as function of a set of generalized coordinates, as stated by the Frobenius theorem. As a consequence, the Lagrange multipliers cannot be computed as in (3.15). In turn, these constraints are defined in the form (2.29) or equivalently (2.31). One problem arising with this kind of constraints is how to compute the Lagrangian multipliers to satisfy (2.33). Moreover, these multipliers represent the forces required to maintain such constraints. Unfortunately, most methods to compute these Lagrangian multipliers are designed for systems under holonomic constraints [Bayo and Avello, 1994; Gudiño Lau and Arteaga, 2005; Gutiérrez-Giles and Arteaga-Pérez, 2017] and thus require a position-level definition of the Pfaffian constraints like those in (2.25) or (2.26). As stated in Faurling et al. [2007], the computation presented in Murray et al. [1994] can be employed for this case. However, it is well-known that this solution is unstable, since its underlying mechanism is a second order integrator with zero input. In this work, a modification of the approach employed by Gudiño Lau and Arteaga [2005] is proposed as follows. For simplicity's sake, let define $\boldsymbol{H}_\mathrm{v} = \boldsymbol{H}_\mathrm{v}(\boldsymbol{q}_\mathrm{v})$, $\boldsymbol{C}_\mathrm{v} = \boldsymbol{C}_\mathrm{v}(\boldsymbol{q}_\mathrm{v}, \dot{\boldsymbol{q}}_\mathrm{v})$, $\boldsymbol{g}_\mathrm{v} = \boldsymbol{g}_\mathrm{v}(\boldsymbol{q}_\mathrm{v})$, $\boldsymbol{A}_\mathrm{v} = \boldsymbol{A}_\mathrm{v}(\boldsymbol{q}_\mathrm{v})$, and $\boldsymbol{\psi} = \boldsymbol{\psi}(\boldsymbol{q}_\mathrm{v}, \dot{\boldsymbol{q}}_\mathrm{v}) = \boldsymbol{A}_\mathrm{v}(\boldsymbol{q}_\mathrm{v})\dot{\boldsymbol{q}}_\mathrm{v}$. Then, the Lagrange multipliers can be computed as

$$\boldsymbol{\lambda}_\mathrm{v} = (\boldsymbol{A}_\mathrm{v}\boldsymbol{H}_\mathrm{v}^{-1}\boldsymbol{A}_\mathrm{v}^\mathrm{T})^{-1}\left[\dot{\boldsymbol{\psi}} - \dot{\boldsymbol{A}}_\mathrm{v}\dot{\boldsymbol{q}}_\mathrm{v} - \boldsymbol{A}_\mathrm{v}\boldsymbol{H}_\mathrm{v}^{-1}(\boldsymbol{\tau}_\mathrm{v} - \boldsymbol{C}_\mathrm{v}\dot{\boldsymbol{q}}_\mathrm{v} - \boldsymbol{D}_\mathrm{v}\dot{\boldsymbol{q}}_\mathrm{v} - \boldsymbol{g}_\mathrm{v})\right], \tag{3.22}$$

where the constraints are forced to satisfy

$$\dot{\boldsymbol{\psi}} + 2\alpha_\mathrm{v}\boldsymbol{\psi} + \beta_\mathrm{v}\int_{t_0}^{t}\boldsymbol{\psi}\,\mathrm{d}\vartheta = \boldsymbol{0}, \tag{3.23}$$

with $\alpha_\mathrm{v}, \beta_\mathrm{v} > 0$ chosen to ensure fast convergence to the origin. Notice that the constraint function $\boldsymbol{\psi}$ can be defined in terms of the end-effector velocities, *i.e.*, $\boldsymbol{\psi} = \boldsymbol{\psi}(\boldsymbol{x}_\mathrm{v}, \boldsymbol{v}_\mathrm{v}) = \boldsymbol{A}_\mathrm{v}(\boldsymbol{x}_\mathrm{v})\boldsymbol{v}_\mathrm{v}$. Therefore, the initial condition of the integral term on the left-hand side of (3.23) can be set to zero. In general, each element of $\boldsymbol{\lambda}_\mathrm{v}$ is a function of $\boldsymbol{q}_\mathrm{v}$, $\dot{\boldsymbol{q}}_\mathrm{v}$, and $\boldsymbol{\tau}_\mathrm{v}$ since the constraints change with the configuration, velocity and virtual applied force. By substituting (3.22) in the motion equation (2.33), a complete description of the dynamics of the system is gotten. With

respect to force, sensor measurements $\boldsymbol{F}_{\mathrm{h}}$ on the master side can be used to calculate the real Lagrange multiplier as

$$\boldsymbol{\lambda}_{\mathrm{m}} = \{\boldsymbol{A}_{\mathrm{v}}\boldsymbol{A}_{\mathrm{v}}^{\mathrm{T}}\}^{-1}\boldsymbol{A}_{\mathrm{v}}\boldsymbol{F}_{\mathrm{h}}. \tag{3.24}$$

Define

$$\boldsymbol{\lambda}_{\mathrm{d}i}(t) \triangleq \boldsymbol{\lambda}_j(t) \tag{3.25}$$

as the desired force trajectory in joint space. The corresponding integral is

$$\Delta\boldsymbol{\lambda}_i = \int_0^t (\boldsymbol{\lambda}_i - \boldsymbol{\lambda}_{\mathrm{d}j})\mathrm{d}t. \tag{3.26}$$

Finally, instead of (3.13)–(3.14), for the master and virtual robot the proposed position–force control for virtual dynamic system subject to nonholonomic constraints is

$$\boldsymbol{\tau}_{\mathrm{m}} = \boldsymbol{Y}_{\mathrm{m}}(\boldsymbol{q}_{\mathrm{m}}, \dot{\boldsymbol{q}}_{\mathrm{m}}, \ddot{\boldsymbol{q}}_{\mathrm{m}})\boldsymbol{\theta}_{\mathrm{m}} - \boldsymbol{K}_{\mathrm{am}}\dot{\boldsymbol{q}}_{\mathrm{m}} - \boldsymbol{K}_{\mathrm{pm}}\boldsymbol{s}_{\mathrm{am}} + \boldsymbol{A}_{\mathrm{v}}^{\mathrm{T}}(\boldsymbol{q}_{\mathrm{v}})(\boldsymbol{\lambda}_{\mathrm{v}} - \boldsymbol{K}_{\mathrm{fm}}\Delta\boldsymbol{\lambda}_{\mathrm{m}}) \tag{3.27}$$

$$\boldsymbol{\tau}_{\mathrm{v}} = \boldsymbol{K}_{\mathrm{av}}\dot{\boldsymbol{q}}_{\mathrm{v}} + \boldsymbol{K}_{\mathrm{pv}}\boldsymbol{s}_{\mathrm{av}} - \boldsymbol{A}_{\mathrm{v}}^{\mathrm{T}}(\boldsymbol{q}_{\mathrm{v}})(\boldsymbol{\lambda}_{\mathrm{m}} - \boldsymbol{K}_{\mathrm{fv}}\Delta\boldsymbol{\lambda}_{\mathrm{v}}). \tag{3.28}$$

Note that the novelty of the approach is not the control scheme because very well-known techniques are employed, but instead the novelty lies in the effective use of nonholonomic constraints to describe penetrable virtual surfaces. Therefore, a technical stability proof is not provided but a set of reliable experiments are shown in the next section with the aim of validate the proposed approach.

3.3 Visual components of the virtual environment

A fundamental part of the developed virtual reality system is the visual feedback. Generally, in dynamic systems and control research, there is not interest at including such element but in real-world applications, as surgery simulators, it is essential. Nowadays, in those developments the virtual environments are composed by merging several numeric techniques that, combined with the fast velocity of today processors, give the virtual objects and surfaces a realism that before would seem impossible to reach. However, since one of the goals of this dissertation is to demonstrate how a teleoperation control scheme can be used in a virtual reality system, the environment was designed by using the basic fundamentals of graphic computing.

The tool utilized to design the virtual environment was the graphic standard *OpenGL 2.0* that is an API which is a software library for accessing features in graphics hardware. It contains different commands that are used to specify objects, images and operations needed to produce interactive three-dimensional graphic applications [Shreiner et al., 2013]. Among those operations, the possibility to give texture[1] and lighting to the virtual objects is possible, besides of proportioning position and orientation changes to the scene's camera, *i.e.*, the way the operator sees the images on the computer screen with respect to height, deep, viewing angles as pitch, roll and yaw, etc.

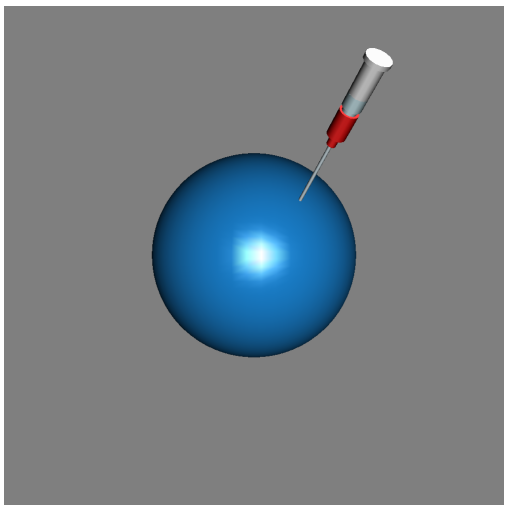

Figure 3.5: Virtual environment developed in *OpenGL 2.0*

As can be seen in Figure 3.5, the environment of the developed application consists of a motionless floating sphere and the virtual avatar of the system. For simplicity sake, there is no changes in the camera's position and orientation but lighting and texture were given to the scene. A notably difference can be appreciated in Figure 3.6, where the avatar have no lighting, the quality of its texture is less and the background color changes. In this case, a set of aligned cylinders become the avatar which position and orientation are directly related to those of the end-effector of the haptic interface. Since *OpenGL* has the necessary instructions to create elements from primitives, the generation of such cylinders and the floating sphere was straightforward. Nevertheless, the aim of showing how the virtual environment was built is to proportionate an initial framework of how the haptic approach proposed can be used.

[1] In the context of graphic computing, texture is referred to the feature of give color or combinations of colors to the objects.

Figure 3.6: Virtual avatar of the system

3.3.1 Rigid sphere

It is important to remember that, from a haptic rendering approach, the classification of non-penetrable and penetrable object was established for rigid and deformable objects respectively. In the case of the rigid object, the *Open GL* instruction `glutSolidSphere()` build automatically a solid sphere with a specific radius by defining the number of subdivisions around (sectors) and along (stacks) the z axis, as can be seen in Figure 3.7[2].

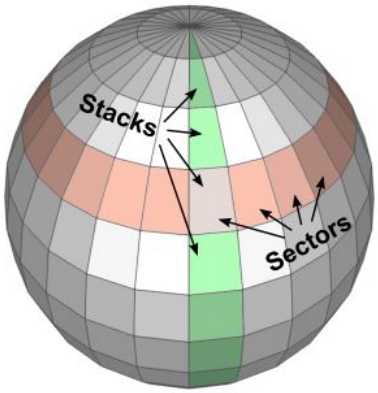

Figure 3.7: Sectors and stacks of a solid sphere

The effect of rigidity is given because that vertices position is not modified when contact with the virtual avatar occurs. However, giving the haptic effect of highly rigid objects to the operator was difficult since an impedance device was used. For such reason, it is important to establish the control scheme (3.20)-(3.21) that compensates, as possible, the limitations of hardware and making feel that, effectively, a contact with a rigid object is produced.

[2]This image and a complete description of the process to build rigid virtual spheres as well as its mathematical basis is given at http://www.songho.ca/opengl/gl_sphere.html.

3.3.2 Deformable sphere

The real challenge is when the virtual object is deformable. Say that an object is *deformable* has a lot of implications, principally related to the continuum mechanics. Therefore, the visual effect of deformation is more complex than the produced by rigidity, because a real deformable object has an infinite number of degrees of freedom. For this reason, virtual objects need a high resolution, which gives a better rendering quality to both visual and haptic feedback [Zafer and Yilmaz, 2016] and, consequently, more realism to te application. However, this is always limited to the available computational resources. In the case of the virtual environment designed in this dissertation, both the graphics and the control algorithm run in a single program, with a sample time of 2 [ms]. If the graphic part takes up more processing resources, such sample time will increase, and non desired effects will appear (*i.e.* delays), and eventually a crash down of the application.

The sphere was drawn by defining four vertices a, b, c and d, which form a plane that is replicated iteratively according a number of parallels p and meridians m defined by the operator. The value of the iteration is intrinsically linked to the resolution of the sphere. In Figure 3.8 the deformable sphere with different resolution is shown. In the sphere on the left, the values used were $p = m = 50$ while in the central sphere were $p = m = 100$. On the other hand, for the sphere on the right the values were $p = m = 150$.

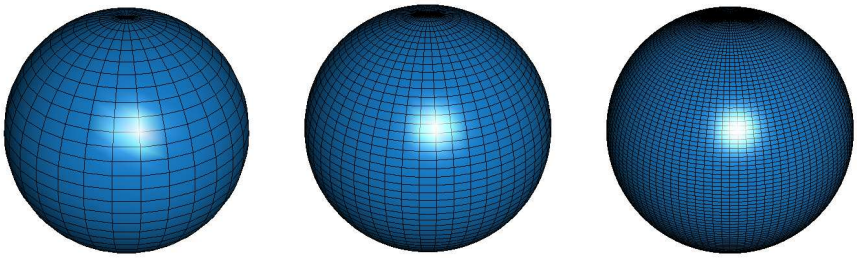

Figure 3.8: Surface mesh generated using different values for p and m

As mentioned before, the more the resolution of the sphere, the more the realism of the application. However, the computational processing when using the resolution of the last case, did not allow a correct performance of the graphic part nor of the haptic. For this reason, a resolution of $p = m = 100$ was chosen.

For simplicity's sake, the sphere was built by placing two hemispheres, one above other, with respect to a common axis. The contact with the virtual avatar will arise in a single point

(x_v, y_v, z_v) computed parametrically as

$$x_v = r\cos(\alpha)\cos(\beta) \tag{3.29}$$

$$y_v = r\cos(\alpha)\sin(\beta) \tag{3.30}$$

$$z_v = r\sin(\alpha), \tag{3.31}$$

where r is the radius of the sphere, and α and β are the angles from whose ranges parallels and meridians are drawn. For code optimization meridians have the range of $[0,360][^\circ]$ and parallels of $[0,180][^\circ]$. Such ranges correspond to the upper hemisphere while the lower is drawn by considering the z_v axis negative part.

Every vertex a, b, c and d must take the value of equations (3.29)-(3.31) in order to visualize their initial position in the virtual environment. To improve the interaction with the virtual avatar, an offset r_{off} is included in the equations that define the vertices as

$$v_{nx} = (r - r_{off})\cos(\alpha)\cos(\beta) \tag{3.32}$$

$$v_{ny} = (r - r_{off})\cos(\alpha)\sin(\beta) \tag{3.33}$$

$$v_{nz} = (r - r_{off})\sin(\alpha), \tag{3.34}$$

where $n = a, b, c, d$ and r_{off} takes an arbitrary value defined experimentally. The contact with the virtual avatar occurs in some plane defined iteratively by (3.32)-(3.34) using a collision-detection algorithm consisting on validating the value of each vertex of every plane and comparing the value of each component (x_v, y_v, z_v) with the the position of the master robot $\boldsymbol{p}_m$. If such values belong to the range of the plane, then the avatar is in contact with the sphere. The next step is to produce the effect of motion of the contact plane. Algorithm 1 shows the pseudocode of such process, which uses an auxiliary normal force in which direction the plane will move. This force does not belong to that calculated using the nonholonomic constraint but the gradient of equation (2.27) and it is used only for visual effect purposes.

The process described above makes only one plane to move and, if that happens, the deformation effect is not realistic. For this reason, moving the adjacent planes is necessary. The more planes are moved, the more realistic the effect of the object deformation. However, in the application developed only the position of the surrounding planes to the contact plane is modified since the method is purely geometric and not based on continuum mechanics. Using such approach implies a lot of considerations that are beyond of the scope of this re-

search.

Algorithm 1 shows the part of the pseudocode where the modification of the adjacent planes takes place. This does not occur at the same time than the modification of the contact plane since these have not yet been created by the code. Figure 3.9 shows the visual effect that the motion, both of the contact plane and adjacent planes produces. It is important to note that the code implemented needs to be optimized and above all, to be adapted to the force rendering algorithm through nonholonomic constraints.

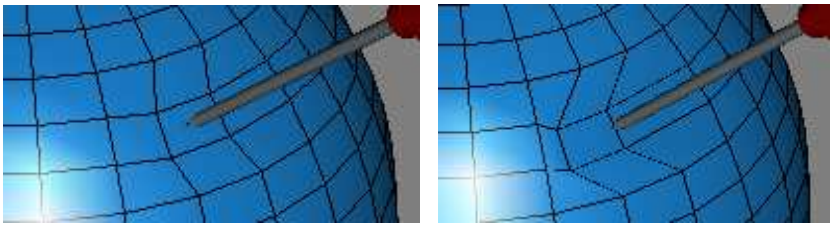

Figure 3.9: Deformation effect of the sphere

Algorithm 1 Deformable sphere

Require: Number of Parallels p
Require: Number of Meridians m
Require: Radius r
Require: Offset r_{off}
Define: Vertex v_{ax}, v_{ay}, v_{az}
Define: Vertex v_{bx}, v_{by}, v_{bz}
Define: Vertex v_{cx}, v_{cy}, v_{cz}
Evaluate: $\Delta_1 = 180° / p$ and $\Delta_2 = 360° / m$

```
for all i = 0 to p/2 do
    for all j = 0 to m do
        α=i * Δ1;
        β=j * Δ2;
        Compute vertices of the upper hemisphere using (3.32)-(3.34)
        if avatar touches a superior plane then
            Move the contact plane through the auxiliary normal force
            Store data of the contact plane (based in p and m)
        end if
        Compute vertices of the lower hemisphere
        if avatar touches an inferior plane then
            Move the contact plane using (3.32)-(3.34) and the negative numbers of z_v axis
            Store data of the contact plane (based in p and m)
        end if
    end for
end for
if avatar touches a plane then
    Reordering adjacent planes
end if
for all i = 0 to p/2 do
    for all j = 0 to m do
        Draw plane
    end for
end for
```

Chapter 4

Experimental results

In this chapter, the validation of the proposed virtual constraints approach presented in Chapter 3 is carried out. Since such validation includes the subjective variable of what the operator feels, it is convenient to give the experiments a practical interpretation. To do so, a complete description of the task performed by the operator is given, taking into account the similarity that the application has with a medical simulator with respect to the produced tactile perception. The experiments were carried out separately for holonomic and nonholonomic constraints in order to make an adequate comparison between them. Moreover, the performance of the control scheme is studied as well as the suitability of integrating the teleoperation approach into a virtual reality system with haptic feedback.

In the context of this dissertation, the most important goal is to verify if there is a difference between the tactile sensation produced by contact with a penetrable virtual constraint or that produced by a non-penetrable one, as well as validate the suitability of the nonholonomic constraints for the latter.

4.1 Experimental platform

The experimental platform consists of a Geomagic Touch haptic robot with six revolute joints, having only the first three of them actuated. An *ATI Nano-17* six–axis force sensor is adapted at the last link, as shown in Figure 4.1. A PC executes the control loop with a sample time of $T = 2$ [ms]. As mentioned in Chapter 3, the virtual environment consists of a sphere developed using the graphic standard *OpenGL 2.0*. It should be noted that both the control algorithm and the graphic simulation run in the same application developed in *Visual Studio*/*C++*.

Figure 4.1: Experimental platform

One practical limitation of the Geomagic Touch robot is that only the first three joints are actuated. Therefore, a projection of both the force reflection and the controller torques is necessary, *i.e.*, the contribution of the last two joint torques is neglected. On the other hand, the virtual robot does not have this limitation, and therefore is considered to be fully-actuated. The master robot limitation is not so restrictive, since the virtual environment considers only force but not end-effector torque feedback, avoiding the problem of sensor/actuators asymmetry in haptic interfaces [Barbagli and Salisbury, 2003]. Furthermore, the contribution of the last two joints to the force reflection is much less in magnitude when compared with the contribution of the first three joints.

4.2 Description of the task

A detailed description of the interaction process between the virtual tool and the virtual environment is presented, simulating separately a rigid and a penetrable sphere. Since the goal of this research was to extend the use of the control scheme to medical training applications, the shape of the avatar as a needle was adopted, as can be seen in Figure 4.2. In medicine, the procedures using this tool are very common, being the needle insertion the most studied and simulated Yang et al. [2018]. In such procedure, the operator takes a sterile needle and slowly approaches it to the patient. Once in contact, he/she has to be very careful and, through

a tactile feeling, know if either soft (muscle, organ) or rigid (bone) tissue has been affected. In both cases, the contact surface produces a reaction force in opposition to the operator's movements. While in a rigid surface the force does not let the needle to penetrate the tissue, in a penetrable surface this is possible. Furthermore, the force behaviors are different as in the first case there is a major contribution in the normal direction, which would allow the operator to move the needle laterally over the surface. In the second case, the normal force contribution is smaller and the surrounding tissue would not allow moving the needle in the lateral directions. In the approach presented, it is assumed that the virtual tool is attached to the end-effector of a five degrees of freedom manipulator, which is not visible in the graphic simulation. It may seem counterintuitive because, evidently in real life, a needle does not have such dynamics. Nevertheless, the robot model is used as a demonstrative example since other medical tools such as an endoscope, resectoscope, forceps, etc. attached to a teleoperated surgical robot arm, have such complex dynamics that must be modeled. The graphic simulations in those cases include pulling, gripping, clamping and cutting and therefore it is convenient to have a complete description of both the kinematics and the dynamics of the tool–tissue interaction Basdogan et al. [2004].

The task starts with the Geomagic Touch robot in its home position. The operator grasps the master robot stylus using the force sensor adapter tip to later gently bring it closer to the virtual surface. The desired trajectory in free motion is imposed in this way. At the same time, the virtual robot moves following such trajectory in the virtual environment, without any scaling between the virtual and the real workspaces. Both the avatar movement and the virtual surface are perceived visually through a computer screen. When the collision-detection algorithm detects contact with the surface, the force-response algorithm generates a virtual force trajectory by computing the corresponding Lagrangian multipliers, either by employing (3.15) or (3.22). The operator perceives an interaction force exerted by the master robot and registered by the Nano–17 force sensor until the contact is over. Finally, the operator returns the sensor adapter to its initial position, thus completing the task.

4.3 Holonomic constraint experiment

For simplicity's sake, the surface used to test the validity of the proposed approach is a sphere described by

$$\varphi_{\mathrm{v}}(\boldsymbol{x}_{\mathrm{v}}) = (x_{\mathrm{v}} - h)^2 + (y_{\mathrm{v}} - k)^2 + (z_{\mathrm{v}} - l)^2 - r^2 = 0, \tag{4.1}$$

where $\boldsymbol{x}_\mathrm{v} = \begin{bmatrix} x_\mathrm{v} & y_\mathrm{v} & z_\mathrm{v} \end{bmatrix}^\mathrm{T}$ is the vector that stands for the virtual environment task–space coordinates, $r = 0.1[\mathrm{m}]$ is the radius, and $(h,k,l) = (0.4,0,0)[\mathrm{m}]$ are the sphere center coordinates. It is important to note that, in contrast with other works, a third dimension $\boldsymbol{z}_\mathrm{v}$ was added in order to heighten realism of the virtual reality application. For example, Faurling et al. [2007] and Gutiérrez-Giles and Arteaga-Pérez [2017] consider only two dimensions to test different control schemes for a haptic and a teleoperation system respectively.

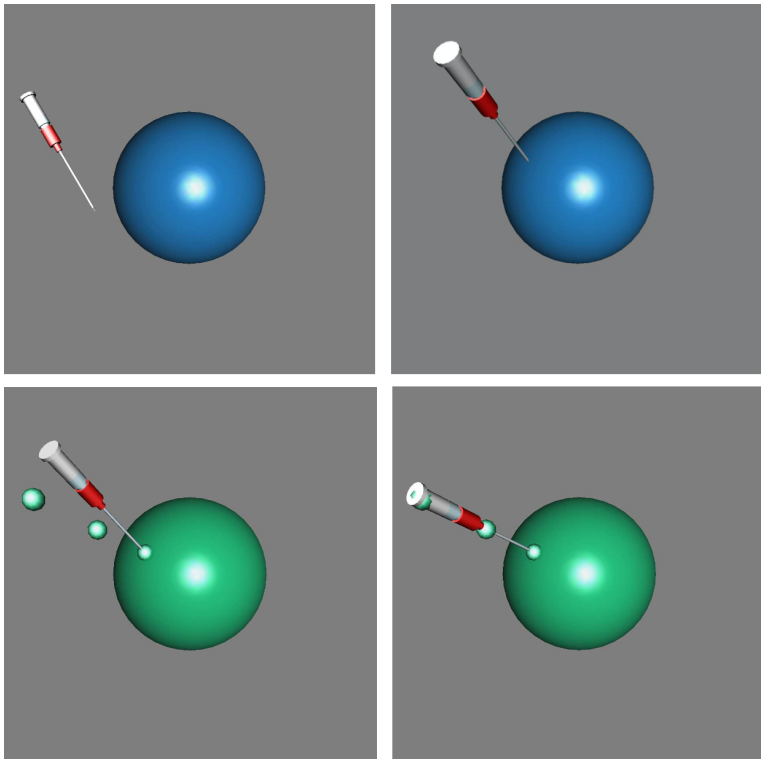

Figure 4.2: Interaction sequence between the avatar and the rigid virtual surface

The gains for the control law (3.20) are $\boldsymbol{K}_\mathrm{am} = 0.055\boldsymbol{I}$, $\boldsymbol{K}_\mathrm{pm} = 0.0055\boldsymbol{I}$, $\boldsymbol{K}_\mathrm{fm} = 10.05\boldsymbol{I}$, $\boldsymbol{\Lambda}_\mathrm{xm} = 0.25\boldsymbol{I}$, $\boldsymbol{K}_{\beta\mathrm{m}} = 0.01\boldsymbol{I}$, and $\boldsymbol{K}_{\gamma\mathrm{m}} = 0.015\boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix of appropiate dimensions. For the virtual manipulator $\boldsymbol{K}_\mathrm{av} = 0.2\boldsymbol{I}$, $\boldsymbol{K}_\mathrm{pv} = \mathrm{diag}(0.2,0.2,0.2,0.1,0.1)$, $\boldsymbol{K}_\mathrm{fv} = 0.2\boldsymbol{I}$, $\boldsymbol{\Lambda}_\mathrm{xv} = 20\boldsymbol{I}$, $\boldsymbol{K}_{\beta\mathrm{v}} = \boldsymbol{I}$, and $\boldsymbol{K}_{\gamma\mathrm{v}} = 0.2\boldsymbol{I}$ were chosen. Finally, by using the Generic Penalty Method, the surface parameters are $\alpha_\mathrm{v} = 0.02$, $\xi = 100$ and $\omega_\mathrm{n} = 200$.

An interaction sequence is presented in Figure 4.2. In the first frame (top left), the virtual avatar starts from its initial position and moves freely in the virtual environment. This movement responds to the operator thanks to the position controller part of (3.21). In the second

frame (top right) the avatar slowly approaches the virtual sphere but has no contact yet. In the third box (bottom left), the avatar begins the contact with the virtual surface, changing its color to green to show contact. At the same time, three spheres that indicate the normal vector to the surface at the contact point appear in the simulation. These spheres are used as markers and help the operator to align the direction of the sensor to obtain a correct measurement of the force. Then, the user tries to orient his/her wrist to align the sensor with these markers, as seen in the last frame of the sequence (bottom right). It is important to note that even when force measurements are not precise during the lapse of time between contact and the alignment with the markers, the force controller must be capable of correctly reproducing the virtual force generated at the virtual environment and consequently, of allowing the operator to feel the sphere all along the experiment.

The position tracking in Cartesian coordinates is shown in Figure 4.3. The operator's hand trajectory can be appreciated approximately along the first 8 seconds of the experiment. From this time to 40 seconds, the virtual robot gets in contact with the virtual surface using the equation of the sphere as collision algorithm. Eventually, the contact is over and the virtual robot stays in free movement until the experiment finishes.

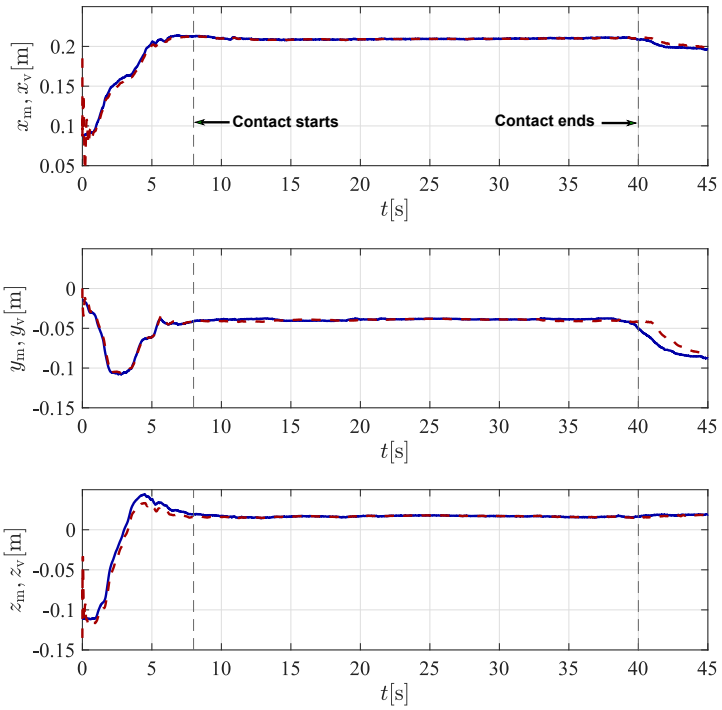

Figure 4.3: Position tracking in Cartesian coordinates: master (—), virtual (– –).

In Figure 4.4, the corresponding position tracking error is presented. Note that it is larger in free motion than in contact with the virtual surface. The reason is that the movement is slower in constrained motion, which depends entirely on the velocity applied by the operator.

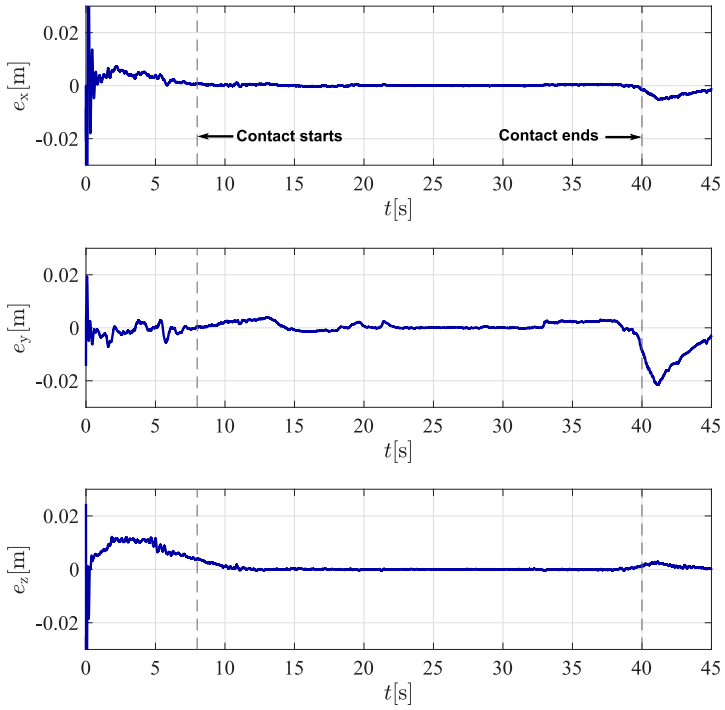

Figure 4.4: Position tracking error.

Force tracking is presented in Figure 4.5, where the virtual force $\boldsymbol{F}_{\mathrm{v}}$ is zero until the contact begins. On the other hand, the master force $\boldsymbol{F}_{\mathrm{h}}$ presents values different from zero because the operator is holding the corresponding end-effector during the whole experiment.

At the end of the contact, oscillations take place as a characteristic of the Generic Penalty Method. Nevertheless, those oscillations are vaguely perceived by the operator and force tracking is preserved. In Figure 4.6, the corresponding force tracking error is presented.

4.4 Nonholonomic constraint experiment

As mentioned before, a deformable surface cannot be expressed implicitly, even when the operator perceives it as a sphere both visually and haptically. In contrast, a discrete repre-

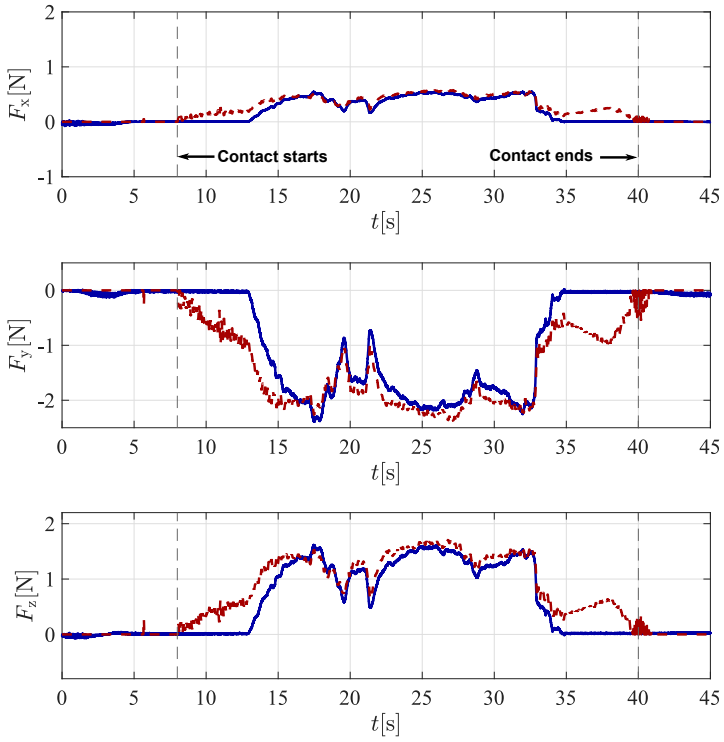

Figure 4.5: Force tracking in Cartesian coordinates: master (—), virtual (– –).

sentation similar to that presented by Zafer and Yilmaz [2016], where the surface is assumed to be made up of a high number of neighboring planes defined by shared nodes, is used. In this dissertation, a technique that consists in iteratively choosing a small vicinity of planes where the contact will occur, depending on the position of the virtual tool, is proposed. Afterwards, the Lagrange multipliers of equation (3.22) are associated to a pair of planes using the impulse-based technique for multirigid-body simulations [Constantinescu et al., 2005]. The micro–collisions with this technique occurs only in the chosen vicinity of the sphere and the impulses preventing body interpenetration are replaced by Lagrange multipliers. In the case of the collision detection algorithm, the convex polyhedra decomposition [Kim et al., 2002] is replaced by the implicit surface representation, using the equation (4.1). This was done for ease and to reduce the computational cost of the application, otherwise, the control algorithm sample time would increase.

Considering the case where the needle is inside the sphere, but it is not allowed to move laterally. However, it is allowed to pivoting to change orientation. This kind of scenario is adequately described by employing nonholonomic constraints. As mentioned in Chapter 1, non-

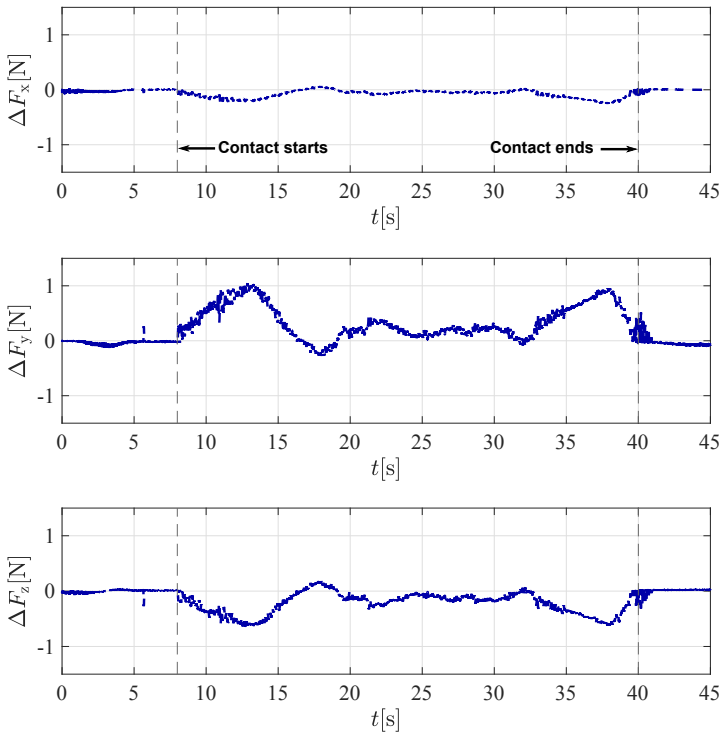

Figure 4.6: Force tracking error.

holonomic constraints have been little exploited to represent the interaction with penetrable surfaces. For example, in Faurling et al. [2005] it is claimed that to model a surgeon scalpel both holonomic and nonholonomic constraints could be employed by limiting the depth of its incision and the direction of its motion respectively. However, there is not any analysis or modeling of this process in such work. The most representative proposal is the derivation of the nonholonomic generalized unicycle model presented by Webster III et al. [2006], where a coordinate-free representation is used to model the insertion of a flexible needle into soft tissue. A similar approach is employed by using the homogeneous matrix representation, but taking into consideration both the kinematic and the dynamic model of the virtual robot and the fact that nonholonomic constraints are more intuitively obtained if they are defined in task space coordinates. Moreover, the computation of the Lagrangian multipliers for nonholonomic constraints, which is proposed in (3.23), is an important improvement with respect to the cited works.

The experiment consisted on a five degree of freedom virtual manipulator interacting with a deformable sphere. Once in contact, the end-effector is not allowed to move laterally, *i.e.*,

along the ${}^0\boldsymbol{y}_{5v}$ and ${}^0\boldsymbol{z}_{5v}$ axes, after a conventional Denavit-Hartenberg allocation, but it is allowed to move along the ${}^0\boldsymbol{x}_{5v}$ axis, *i.e.*, along the pointing direction of the end-effector. Moreover, the end-effector is permitted to rotate (pivoting) to change direction (as a three dimensional version of the nonholonomic unicycle) and the Pfaffian matrix is computed as

$$\boldsymbol{A}_v(\boldsymbol{x}_v)\,\boldsymbol{v}_v = \begin{bmatrix} {}^0\boldsymbol{y}_{5v} & \boldsymbol{0}_{1\times 3} \\ {}^0\boldsymbol{z}_{5v} & \boldsymbol{0}_{1\times 3} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{p}}_v \\ \boldsymbol{\omega}_v \end{bmatrix} = \boldsymbol{0}. \tag{4.2}$$

The experiment has four steps, as shown in Figure 4.7 and the video available at https://youtu.be/z1f058uPZfI. First, the virtual robot is in free motion and only the teleoperation part of the controller (3.28) is active. In the second part, the needle is inserted into the sphere. Next, in the third part, the needle is rotated approximately 45 degrees without changing its position. Finally, the needle is inserted deeper into the sphere with the new orientation.

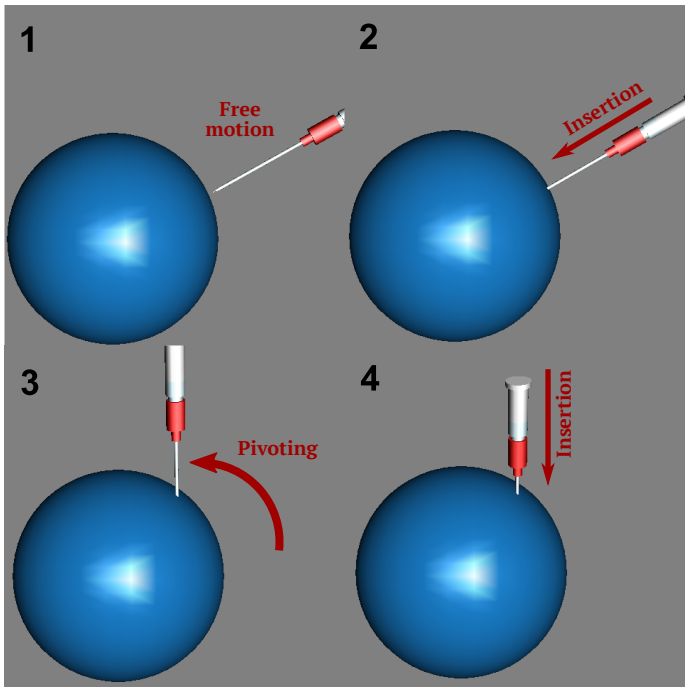

Figure 4.7: Interaction sequence between the avatar and the nonholonomic virtual surface

The human operator force in the needle lateral directions is difficult to measure directly with the force sensor. Instead, the projection of such forces into the master manipulator joint torque is exploited, *i.e.*, $\boldsymbol{\lambda}_m$ from (2.20), (3.24), and (2.23) are computed. The gains for the master robot control law (3.20) are $\boldsymbol{K}_{am} = 0.055\boldsymbol{I}$, $\boldsymbol{K}_{pm} = 0.055\boldsymbol{I}$, $\boldsymbol{K}_{fm} = \text{diag} = 0.01\boldsymbol{I}$, $\boldsymbol{\Lambda}_{xm} =$

$0.25\boldsymbol{I}$, $\boldsymbol{K}_{\beta\mathrm{m}} = 0.01\boldsymbol{I}$, and $\boldsymbol{K}_{\gamma\mathrm{m}} = 0.015\boldsymbol{I}$, whereas for the virtual manipulator $\boldsymbol{K}_{\mathrm{av}} = 0.2\boldsymbol{I}$, $\boldsymbol{K}_{\mathrm{pv}} = \mathrm{diag}(0.2, 0.2, 0.2, 0.1, 0.1)$, $\boldsymbol{K}_{\mathrm{fv}} = 2\boldsymbol{I}$, $\boldsymbol{\Lambda}_{\mathrm{xv}} = 20\boldsymbol{I}$, $\boldsymbol{K}_{\beta\mathrm{v}} = \boldsymbol{I}$, and $\boldsymbol{K}_{\gamma\mathrm{v}} = 0.2\boldsymbol{I}$ were chosen.

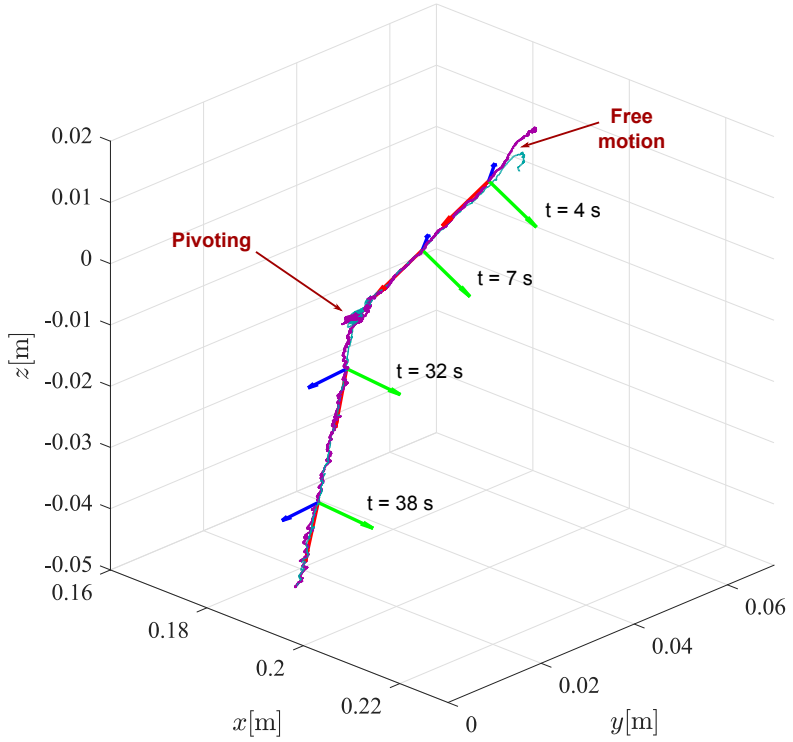

Figure 4.8: Nonholonomic experiment: Cartesian trajectory of the master (—) and virtual (- - -) manipulators, along with the virtual end-effector coordinate frame for different instants of time (red, green and blue arrows represent ${}^0\boldsymbol{x}_{5\mathrm{v}}$, ${}^0\boldsymbol{y}_{5\mathrm{v}}$ and ${}^0\boldsymbol{z}_{5\mathrm{v}}$ axes, respectively).

Figure 4.8 presents a 3D plot of the virtual end-effector position and orientation, showing all four stages of the experiment. Before $t = 4$[s] the manipulator is in free motion, driven only by the master manipulator by means of the teleoperation scheme. Approximately at $t = 4$[s] the virtual robot enters in contact with the sphere. From this time to approximately $t = 10$[s], the end-effector, *i.e.*, the needle, is forced to follow a straight trajectory driven by the nonholonomic constraint forces. Between approximately $t = 10$[s] and $t = 30$[s] the operator changes the needle orientation to a vertical position (pivoting). After this time, the operator drives the needle in a straight line with the new orientation, until the experiment ends. In the same figure, the end-effector coordinate frame axes are shown for some time instants to show that the needle trajectory is always along the $\boldsymbol{x}_{5\mathrm{v}}$ axis (red arrow), which is very difficult to satisfy without the aid of the nonholonomic constraints. The corresponding time-evolution

of the Cartesian position coordinates is displayed in Figure 4.9, whereas the tracking error in these coordinates is shown in Figure 4.10. The virtual constraint forces, represented by the Lagrange multipliers $\boldsymbol{\lambda}_{\mathrm{v}}$ and the forces felt by the human operator, represented by $\boldsymbol{\lambda}_{\mathrm{m}}$ are shown in Figure 4.11 and the corresponding errors are shown in Figure 4.12.

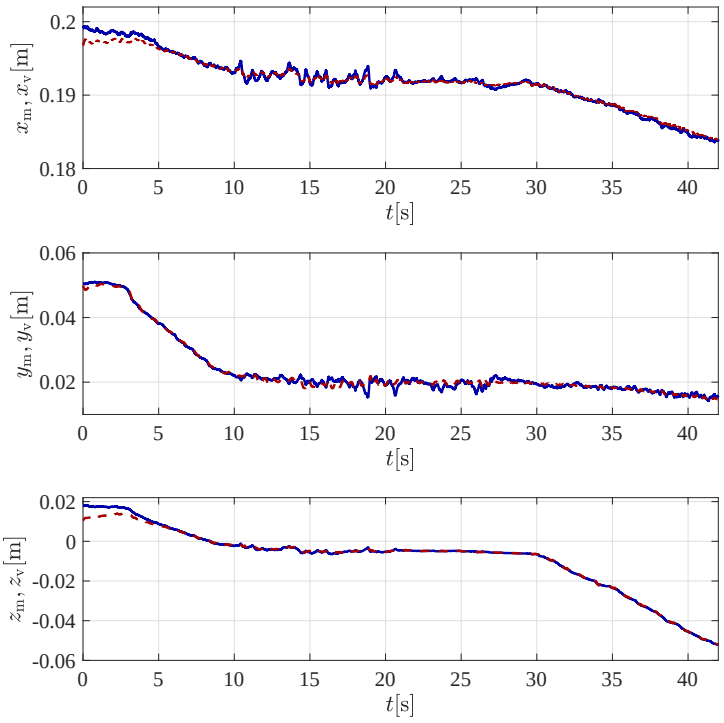

Figure 4.9: Nonholonomic experiment: Cartesian position of the master (—) and virtual (- - -) manipulators.

4.5 Discussion

Some important aspects during the development of the experiments were identified, especially regarding the force rendering. In the first place, it is convenient to remark the differences between using either holonomic and nonholonomic constraints with respect to the definition of the virtual surface. While in the former it is always necessary an implicit equation like (4.1) to generate the reaction force, in the latter only the angle measurements of the manipulator joints are required. Equation (3.15) needs an expression for $\boldsymbol{\varphi}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}})$ to evaluate λ_{v}. But, since in graphic computing it is common practice to work in Cartesian coordinates Ruspini and

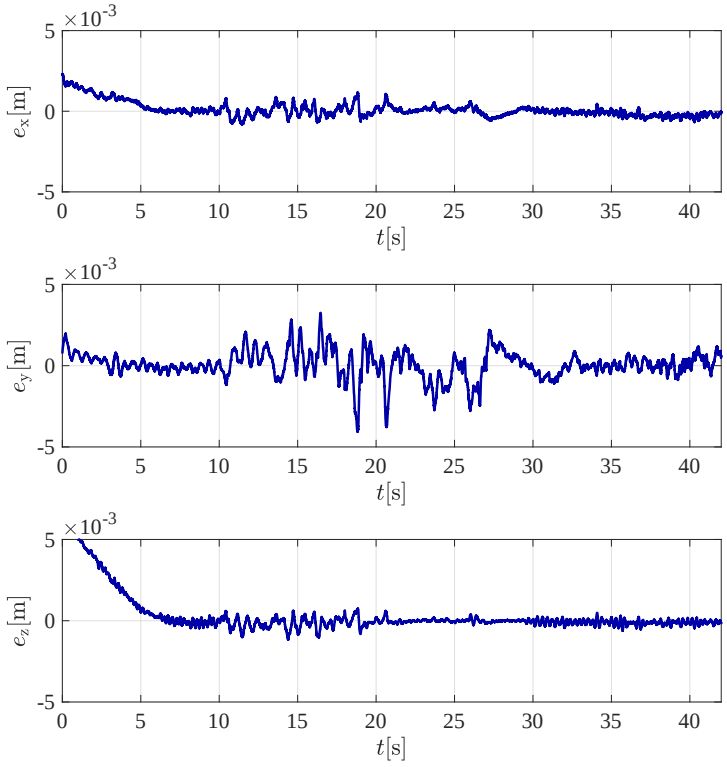

Figure 4.10: Nonholonomic experiment: Cartesian position error.

Khatib [2001], it is necessary to use the gradient $\boldsymbol{J}_{\varphi \mathrm{xv}}(\boldsymbol{q}_\mathrm{v})$ in (3.16). On the other hand, to generate the virtual forces $\boldsymbol{\lambda}_\mathrm{v}$ of equation (3.22), either the virtual robot dynamic model and the matrix $\boldsymbol{A}_\mathrm{v}(\boldsymbol{q}_\mathrm{v})$ are needed.

If, as in the case of the present work, the nonholonomic constraints are defined in task-space coordinates, it is necessary to employ the Jacobian $\boldsymbol{J}_\mathrm{v}(\boldsymbol{q}_\mathrm{v})$ as in (2.30) to project such constraints in joint-space coordinates. Nevertheless, an implicit equation of the virtual constraint is no required, which is relevant in medical simulation of soft penetrable tissues. In such case, no implicit equation can be defined and finite element methods must be used. Tasks as penetration, cutting and indentation becomes computationally expensive and a trade-off between force and visual rendering must be established [Faure et al., 2012]. The common choice for the computing community is to sacrifice force realism for better visualization using position measurements and spring-damper equations.

A second aspect to highlight is the realism of the application. Since a medical training simulation system must be capable of provide the human operator with a heighten dexterity,

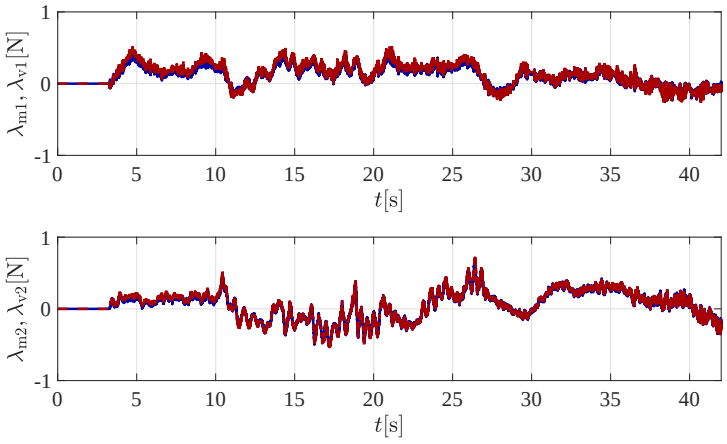

Figure 4.11: Nonholonomic experiment: Cartesian forces at the end -effector of the master (—) and virtual (- - -) manipulators.

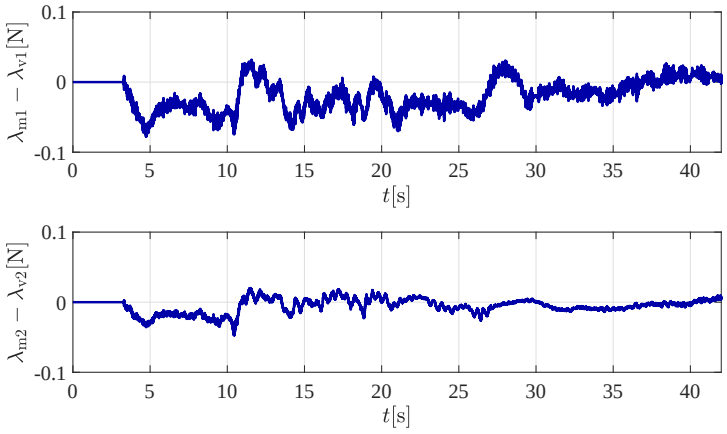

Figure 4.12: Nonholonomic experiment: Cartesian force errors.

we use 5 DOF of the master robot. This allows to visualize the position of the virtual tool and 2 DOF of its orientation. However, only forces but no torques are rendered because of only the first 3 DOF are actuated. In the case of holonomic constraints, it is evident that, considering a single point contact over the virtual sphere, the operator must feel the force that prevents him/her to move the virtual tool in the normal direction at such point (and where ideally the velocity is zero), as can be appreciated in red in Figure 4.5. In fact, it is necessary to have the three green markers in the visual rendering with the aim that the operator can orient the robot's end-effector and generate a suitable force measurement to be used in the control algorithm.

For nonholonomic constraints, however, the objective is not to reproduce a normal force but lateral forces such as would be present in a venipuncture procedure. In that scenario, the operator is able to perform trajectories like the one observed in Figure 4.8. He/she is capable of changing the direction of the virtual tool by making the value of $\boldsymbol{\omega}_{\mathrm{v}}$ different from zero, which is compatible with the Pfaffian constraint matrix $\boldsymbol{A}_{\mathrm{v}}(\boldsymbol{x}_{\mathrm{v}})$. Furthermore, obtaining the forces by means of (3.22) would allow to include dynamic characteristics that not necessarily belong to the virtual robot but to the virtual environment. Such characteristics could include parameters as Poisson ratio, the Young's modulus or the Lamé coefficients [Nordberg and Servin, 2018], which are generally used for graphic applications where the principal objective is to give a better description of deformable surfaces.

Finally, the effectiveness of the control algorithm is studied. Because its nature, it is not possible to make a controller performance comparison between holonomic and nonholonomic constraints, above all for force. In a certain way, this was expected since, for holonomic constraints the force sensor is used to capture the behavior of $\boldsymbol{F}_{\mathrm{h}}$. This could be done because of ideally, the contact force is measured at a single point of the rigid surface. Conversely, for the nonholonomic constraints there is not a single point where the sensor can obtain a reliable force measurement. Since the operator can move the master robot end-effector forward (the velocity is different from zero) but no laterally, any force obtained for a sensor mounted on it during the process would affect the performance of the control algorithm and consequently the operator's tactile perception.

In the case of holonomic constraints, for position tracking, the difference between free and constrained motion are shown in Figures 4.3 and 4.4 respectively. In the first 8 seconds of the experiment the operator moves freely the master robot's end-effector. It is observed a greater position tracking error in this time lapse than in constrained motion. Since the virtual environment coordinates are dependent of the task-space coordinates, a poor performance of the controller would provoke an incorrect visualization of the virtual avatar. As mentioned before, a common practice is to directly associate the position of the latter to that of the robot's end-effector. However making this wouldn't allow to embed the virtual constraint equation into the kinematic mapping between spaces.

For nonholonomic constraints the contact starts at 4 seconds and, as before, there is a bigger position tracking error in free motion. At this point, it is convenient to highlight the role that plays the human operator's behavior for both type of constraints. Such behaviour has a dynamic represented by $\boldsymbol{\tau}_{\mathrm{h}}$ whose parameters and model are difficult to compute [Mendez-

Iglesias et al., 2005] and depends on the position and velocity imposed by the operator. Depending on his/her ability and skills, the master robot's end-effector will be manipulated with higher or lower velocity. This means that, for different operators, different will be the controller performance. For this dissertation, the same operator developed the sets of experiments, showing an acceptable performance in free and constrained motion. However, because of the opposition forces exerted by the master robot, the operator movements are constrained, which means that the velocity is reduced during the contact and therefore the controller's performance is higher.

With respect to the force tracking, for holonomic constraints, an acceptable behavior is observed in Figures 4.5 and 4.6, which in fact makes possible the sensation of touching a sphere without penetrating it. As mentioned before, due to their nature, holonomic constraints allow the use of a force sensor. This is an advantage since that information from the master side allows to register the force interaction between the operator and the virtual constraint. This enhances more the realism of the application by taking a measurement of the operator's force behavior and including it into the virtual environment. On the other hand, which is evident is that to take such measurements implies for the control algorithm must counteract the human dynamics both in free and constrained motion (for example when operator aligns the end-effector with the virtual markers). This causes an error whose maximum value reaches up to 1[N] as can be seen in Figure 4.6.

The case of nonholonomic constraints is different. Since it is not possible to use the sensor, the force feedback comes directly from the robot's torques by using the equations (2.20), (3.24), and (2.23). This does not allow for measurements of the force interaction between the human operator and the virtual environment and, in consequence, have limited information sharing between the real and virtual environments. Nevertheless, as can be seen in Figures 4.11 and 4.12, the performance of the control algorithm is quite good and apparently better than in the holonomic case if an analytical comparison could be done. The relevant aspect in our approach is to feel the reaction forces present while executing the trajectory of Figure 4.8, avoiding the lateral motion of the virtual tool. Such forces are represented in task-space coordinates by the vector $\boldsymbol{\lambda}_{\mathrm{m}}$ and their direction is not normal as in the case of holonomic constraints but lateral, similar to what happen with a needle inserted into a soft tissue.

Chapter 5

Conclusions

In this dissertation, a proposal on haptic interaction with holonomic and nonholonomic virtual constraints was presented. Since a wide research on haptic interaction with rigid surfaces has been presented in the literature, the principal goal was reproduce the forces generated by the interaction with soft surfaces from a force feedback approach. Throughout the document, the necessary theory to establish an optimal relationship between the visual and haptic interaction for the virtual reality system developed was introduced. The key lies in adapting the mathematical properties of the holonomic and nonholonomic constraints to the kinematics of a tool in contact with a non-penetrable and penetrable virtual surface respectively. However, it is important to remark that this adaptation was done to reach only haptic feedback purposes and considering basic contact properties of the rigid and soft tissues simulated.

Adapting a teleoperation control scheme to a virtual reality system was the strategy to follow since it allowed to embed a robot's dynamic model into the virtual environment. By doing this, the teleoperated slave system was addressed as a problem of a virtual robot in constrained motion and whose contact force is given by either holonomic or nonholonomic constraints. We studied the differences between one or another representation, both mathematically and intuitively, and the particularities of each one. Among them there is the fact that they could be employed to render forces using the Generic Penalty Method or the Pfaffian constraints matrix respectively.

The interest was centered on reproducing the contact with a soft surface by employing nonholonomic constraints. The principal use detected is that such method can be employed to render similar forces to those arising from contact between a tool and a penetrable surface. To the best of the author's knowledge this is the first time that nonholonomic constraints are

used to reproduce tactile sensations on the operator with a practical meaning. This approach can be eventually used in virtual reality medical simulators, and the fundamentals to do that were presented throughout this document. However, adapting the developed method to complex virtual environments, such as those found in the medical field, requires more research both in control and computer graphics. In the first case, adapt more accurately the teleoperation controller presented is necessary and make it fit with the current methods of graphic computing. On the other hand, find more optimal ways to modeling force using nonholonomic constraints is essential to heighten the realism of the applications.

With respect to graphic computing, it is necessary to design numerical methods that adapt more efficiently the control algorithms designed and that are capable of run with continuum mechanics models. Naturally, this will increase the computational process and further analysis will be needed to establish the appropriate trade-offs between real time processing and control performance, without sacrificing the realism of the applications. All of this requires a wide range of knowledge that not necessarily belongs to control nor graphic computing. However, this thesis is a first effort to start linking this two areas, which connection could lead to a very interesting combination of interdisciplinary work in the future.

5.1 Future directions

As pointed out above, the work presented in this dissertation belongs to a combination of two different areas of knowledge. Therefore, multiple options to continue the research are available. Some of them will focus on:

- The study of the stability of the system.
- The combination of holonomic and nonholonomic constraints in a same virtual reality application using more sophisticated visual models.
- The development of collision-detection algorithms for nonholonomic constraints.
- Adaptation of the nonholonomic constraints to finite element methods for soft virtual surfaces generation and perception.
- The use of different dynamic models for the virtual tool and its correspondence with real surgical tools.

- The study of delays on the virtual teleoperation system due to the processing of complex visual models and its effects on the controller performance.

The long term objective of all this activities will be provide a theoretical and practical basis for the development of full medical simulators that could be used by surgery trainees, in order to improve their skills and avoid risks for the patients during real surgeries.

5.2 Journal publication

During the author's doctorate period, the following research paper was written. It contains the results presented in this dissertation.

Castro-Díaz, J. D., Sánchez-Sánchez, P., Gutierrez-Giles, A., Arteaga-Pérez, M. A. and Pliego-Jiménez, J. (2020). Experimental Results for Haptic Interaction With Holonomic and Non-holonomic Constraints. *IEEE Acces*, 8:120959-120973. doi:10.1109/ACCESS.2020.3006715.

Bibliography

Adams, R. J. and Hannaford, B. (1999). Stable haptic interaction with virtual environments. *IEEE Transactions on Robotics and Automation*, 15(3):465–474.

Angus, J. and Stone, R. J. (1995). Virtual maintenance. In *Aerospace*, pages 17–21.

Arimoto, S., Liu, Y., and Naniwa, T. (1993). Model-based adaptive hybrid control for geometrically constrained robots. pages 618-623. Proccedings of the 1993 IEEE International Conference on Robotics and Automation.

Arteaga, M. A., Castillo-Sánchez, A., and Parra-Vega, V. (2006). Cartesian control of robots without dynamic model and observer design. *Automatica*, 42(3):473 – 480.

Asano, S., Okamoto, S., and Yamada, Y. (2013). Toward augmented reality of textures: Vibrotactile high-frequency stimuli mask texture perception to be rougher or smoother? In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 510–515.

Avgousti, S., Christofouru, E. G., Panaydes, A. S., Voskarides, S., Novales, C., Nouaille, L., Pattichis, C. S., and Vieyres, P. (2016). Medical telerobotic systems: current status and future trends. *BioMed Eng OnLine*, 15(96).

Ballantyne, G. H. and Moll, F. (2003). The da vinci telerobotic surgical system: the virtual operative field and telepresence surgery. *Surg Clin North Am.*, 6(83).

Barbagli, F. and Salisbury, K. (2003). The effect of sensor/actuator asymmetries in haptic interfaces. In *11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2003. HAPTICS 2003. Proceedings.*, pages 140–147.

Basdogan, C., De, S., Kim, J., Manivannan Muniyandi, Kim, H., and Srinivasan, M. A. (2004). Haptics in minimally invasive surgical simulation and training. *IEEE Computer Graphics and Applications*, 24(2):56–64.

Bayo, E. and Avello, A. (1994). Singularity-free augmented lagrangian algorithms for constrained multibody dynamics. *Nonlinear Dynamics*, 5(2):209–231.

Bliss, J. P., Tidwell, P. D., and Guest, M. A. (1997). The effectiveness of virtual reality for administering spatial navigation training to firefighters. *Presence: Teleoperators and Virtual Environments*, 6(1):73–86.

Brooks, F. P., Ouh-Young, M., Batter, J. J., and Jerome Kilpatrick, P. (1990). Project gropehaptic displays for scientific visualization. *SIGGRAPH Comput. Graph.*, 24(4):177–185.

Chu, C. . P., Dani, T. H., and Gadh, R. (1997). Multimodal interface for a virtual reality based computer aided design system. In *Proceedings of International Conference on Robotics and Automation*, volume 2, pages 1329–1334 vol.2.

Constantinescu, D., Salcudean, S. E., and Croft, E. A. (2005). Haptic rendering of rigid contacts using impulsive and penalty forces. *IEEE Transactions on Robotics*, 21(3):309–323.

Conti, F. and Khatib, O. (2009). A new actuation approach for haptic interface design. *The International Journal of Robotics Research*, 28(6):834–848.

DiMaio, S. P. and Salcudean, S. E. (2003). Needle insertion modeling and simulation. *IEEE Transactions on Robotics and Automation*, 19(5):864–875.

Duriez, C., Dubois, F., Kheddar, A., and Andriot, C. (2006). Realistic haptic rendering of interacting deformable objects in virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 12(1).

Dy, M.-C., Tagawa, K., Hiromi, T. T., and Masaru, K. (2015). Collision detection and modeling of rigid and deformable objects in laparoscopic simulator. In *SPIE Medical Imaging*, volume 9415.

Escobar-Castillejos, D., Noguez, J., Neri, L., Magana, A., and Benes, B. (2016). A review of simulators with haptic devices for medical training. *Journal of Medical Systems*, 104(40):177–185.

Faure, F., Duriez, C., Delignette, H., Allard, J., and Gilles, B. (2012). *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, volume 11, chapter SOFA: A Multi-Model Framework for Interactive Physical Simulation, pages 283–321. Springer.

Faurling, E. L., Lynch, K. M., Colgate, J. E., and Peshkin, M. A. (2005). Haptic Interaction With Constrained Dynamic Systems. In *IEEE International Conference on Robotics and Automation*. Barcelona, Spain.

Faurling, E. L., Lynch, K. M., Colgate, J. E., and Peshkin, M. A. (2007). Haptic Display of Constrained Dynamic Systems via Admittance Displays. *IEEE Transactions on Robotics*, 23:101–111.

Finch, M., Chi, V. L., Taylor, R. M., Falvo, M., Washburn, S., and Superfine, R. (1995). Surface modification tools in a virtual environment interface to a scanning probe microscope. In *Proceedings of the 1995 Symposium on Interactive 3D Graphics*, I3D '95, page 13–ff., New York, NY, USA. Association for Computing Machinery.

Gallace, A. and Spence, C. (2014). *In touch with the future: The sense of touch from cognitive neuroscience to virtual reality*. Oxford University Press, New York, USA.

Goertz, R. C. and Thompson, W. M. (1952). Electronically controlled manipulator. *Nucleonics (US) Ceased publication*, 12.

Grange, S., Conti, F., Helmer, P., Rouiller, P., and Baur, C. (2001). Overview of the Delta Haptic Device. In *Eurohaptics '01*, volume 1. Birminghan, England.

Gudiño Lau, J. and Arteaga, M. A. (2005). Dynamic model and simulation of cooperative robots: A case study. *Robotica*, 23:615–624.

Gutiérrez-Giles, A. and Arteaga-Pérez, M. A. (2017). Transparent bilateral teleoperation interacting with unknown remote surfaces with a force/velocity oberver design. *International Journal of Control*, pages 1–18.

Hamza-Lup, F. G., Bogdan, C. M., Popovici, D. M., and Costea, O. D. (2011). A survey of visuo-haptic simulation in surgical training. In *Proceedings 2011 The Third International Conference on Mobile, Hybrid, and On-line Learning*, pages 57–62.

Hannaford, B., Rosen, J., Friedman, D. W., King, H., Roan, P., Cheng, L., Glozman, D., Ma, J., Kosari, S. N., and White, L. (2013). Raven-ii: An open platform for surgical robotics research. *IEEE Transactions on Biomedical Engineering*, 60(4):954–959.

Hansen, T., Henningsen, C. T., Nielsen, J. J. M., Pedersen, R., Schwensen, J., Sivabalan, S., Larsen, J. A., and Leth, J. (2012). Implementing force-feedback in a telesurgery environment, using parameter estimation. In *2012 IEEE International Conference on Control Applications*, pages 859–864.

Heredia, S. A., Harada, K., Padilla- Castaneda, M., Marques-Marinho, M., Márquez-Flores, J. A., and Mitsuishi, M. (2018). Virtual reality simulation of robotic transsphenoidal brain tumor resection: Evaluating dynamic motion scaling in a master-slave system. *The international Journal of Medical Robotics and Computer Assisted Surgery*, 15.

Hollis, R. L., Salcudean, S., and Abraham, D. W. (1990). Toward a tele-nanorobotic manipulation system with atomic scale force feedback and motion resolution. In *IEEE Proceedings on Micro Electro Mechanical Systems, An Investigation of Micro Structures, Sensors, Actuators, Machines and Robots.*, pages 115–119.

Khatib, O., Yeh, X., Brantner, G., Soe, B., Kim, B., Ganguly, S., Stuart, H., Wang, S., Cutkosky, M., Edsinger, A., Mullins, P., Barham, M., Voolstra, C. R., Salama, K. N., L'Hour, M., and Creuze, V. (2016). Ocean one: A robotic avatar for oceanic discovery. *IEEE Robotics Automation Magazine*, 23(4):20–29.

Kim, K., Barni, M., and Tan, H. Z. (2010). Roughness-adaptive 3-d watermarking based on masking effect of surface roughness. *IEEE Transactions on Information Forensics and Security*, 5(4):721–733.

Kim, K., Song, H., Suh, J., and Lee, J. (2013). A novel surgical manipulator with workspace-conversion ability for telesurgery. *IEEE/ASME Transactions on Mechatronics*, 18(1):200–211.

Kim, Y. J., Otaduy, M. A., Lin, M. C., and Manocha, D. (2002). Six-degree-of-freedom haptic display using localized contact computations. In *Proceedings 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. HAPTICS 2002*, pages 209–216.

Luca, A. D. and Oriolo, G. (1995). *Kynematics and Dynamics of Multi-Body Systems*, volume 360, chapter Modeling and Control of Nonholonomic Mechanical Systems. Springer.

Massie, T. and Salisbury, J. K. (1994). The PHANTOM Haptic Interface: A Device for Probing Virtual Objects. In *ASME Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, volume 1, pages 295–301. Chicago, IL.

Maurel, W., Wu, Y., Magnenat, N., and Thalman, D. (1998). *Biomedichal Models for Soft Tissue Simulation*. Springer, Heidelberg, Germany.

Mavhash, M. and Hayward, V. (2004). High-Fidelity Haptic Synthesis of Contact with Deformable Bodies. *IEEE Computer Graphics and Applications*, 24(2):48–55.

Mendez-Iglesias, J. A., Parra-Vega, V., and Ruíz-Sánchez, F. (2005). Identification of the human behaviour in virtual environments tasks as a non-linear control block. In *Proceedings of the 16th IFAC World Congress*, volume 38, pages 42–47.

Minsky, M., Ming, O., Steele, O., Brooks, F. P., and Behensky, M. (1990). Feeling and seeing: Issues in force display. *SIGGRAPH Comput. Graph.*, 24(2):235–241.

Montagnat, J., Delignette, H., and Ayache, N. (2001). A review of deformable surfaces: topology, geometry and deformation. *Image and Vision Computing*, 19:1023–1040.

Murray, R. M., Li, Z., and Sastry, S. S. (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Boca Raton, Florida, USA.

Nordberg, J. and Servin, M. (2018). Particle-based solid for nonsmooth multidomain dynamics. *Computational Particle Mechanics*, 5(2):125–139.

Papadopoulos, E., Vlachos, K., and Mitropoulos, D. (2002). Design of a 5-DOF haptic simulator for urological operations. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, volume 2, pages 2079–2084 vol.2.

Rizzo, A., Buckwalter, J. G., van der Zaag, C., Neumann, U., Thiebaux, M., Chua, C., van Rooyen, A., Humphrey, L., and Larson, P. (2000). Virtual environment applications in clinical neuropsychology. In *Proceedings IEEE Virtual Reality 2000 (Cat. No.00CB37048)*, pages 63–70.

Rodríguez, A., Basañez, L., Colgate, J. E., and Faulring, E. L. (2008). Haptic display of dynamic systems subject to holonomic constraints. In *IEEE International Conference on Intelligent Robots and Systems*. Nice, France.

Rodríguez, A., Basañez, L., Colgate, J. E., and Faulring, E. L. (2010). A Framework for the Simulation and Haptic Display of Dynamic Systems Subject to Holonomic Constraints. *The International Journal of Robotics Research*, 29(4):336–352.

Rodríguez-Angeles, A., Arteaga-Pérez, M. A., Portillo-Vélez, R. d. J., and Cruz-Villar, C. A. (2015). Transparent bilateral master–slave control based on virtual surfaces: Stability analysis and experimental results. *International Journal of Robotics and Automation*, 30(2):128–139.

Rucker, D. C., Das, J., Gilbert, H. B., Swaney, P. J., Miga, M. I., Sarkar, N., and Webster, R. J. (2013). Sliding mode control of steerable needles. *IEEE Transactions on Robotics*, 29(5):1289–1299.

Ruspini, D. and Khatib, O. (2001). Haptic Display for Human Interaction with Virtual Dynamic Environments. *Journal of Robotic Systems*, 18(12):769–783.

Ruspini, D. C., Kolarov, K., and Khatib, O. (1997). The Haptic display of complex graphical environments. In *Computer graphics Proceedings Anual Conference*, pages 140–147.

Salisbury, K., Conti, F., and Barbagli, F. (2004). Haptic Rendering: Introductory Concepts. *IEEE Computer Graphics and Applications*, 14:24–32.

Saupin, G., Duriez, C., and Cotin, S. (2008). *Biomedical Simulation. ISBMS 2008. Lecture Notes in Computer Science*, volume 5104, chapter Contact Model for Haptic Medical Simulations. Springer.

Sclaroff, S. and Pentland, A. (1991). Generalized implicit functions for computer graphics. *SIGGRAPH Comput. Graph.*, 25(4):247–250.

Selig, J. M. (1996). *Geometric Fundamentals of Robotics*. Springer Science and Business, New York, USA.

Shreiner, D., Sellers, G., Kessenich, J., and Licea-Kane, B. (2013). *OpenGL Programming Guide Eight Edition*. Addison-Weasley.

Sutherland, I. E. (1965). The ultimate display. In *Proceedings of the IFIP Congress*, volume 2, pages 506–509.

Terzopoulus, D., Platt, J., Barr, A., and Fleischer, K. (1987). Elastically Deformable Models. *Computer Graphics*, 21(4):205–214.

Vardar, Y., Güçlü, B., and Basdogan, C. (2018). Tactile Masking byElectrovibration. *IEEE Transactions on Haptics*, 11(4):623–635.

Webster III, R. J., Seob, J., Cowan, N. J., Chirikjian, G. S., and Okamura, A. M. (2006). Nonholonomic Modeling of Needle Steering. *The International Journal of Robotic Research*, pages 509–525.

Xiaoping, Y. and Sarkar, N. (1998). Unified Formulation of Robotics Systems with Holonomic and Nonholonomic Constraints. *IEEE Transactions on Robotics and Automation*, 14:640–650. S 1042-296X(98)06074-1.

Yang, C., Xie, Y., Liu, S., and Sun, D. (2018). Force Modeling, Identification, and Feedback Control of Robot-Assisted Needle Insertion: A Survey of the Literature. *Sensors*, 18(2).

Zafer, N. and Yilmaz, S. (2016). Nonlinear viscoelastic contact and deformation of freeform virtual surfaces. *Advanced Robotics*, 30(4):246–257.

Zilles, C. B. and Salisbury, J. K. (1995). A constraint-based god-object method for haptic display. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 3, pages 146–151 vol.3.