IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 16, 2020, accepted June 27, 2020, date of publication July 2, 2020, date of current version July 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3006715

Experimental Results for Haptic Interaction With Virtual Holonomic and Nonholonomic Constraints

JOSÉ DANIEL CASTRO–DÍAZ[1], PABLO SÁNCHEZ–SÁNCHEZ[2], ALEJANDRO GUTIÉRREZ–GILES[3], MARCO A. ARTEAGA–PÉREZ[1], AND JAVIER PLIEGO–JIMÉNEZ[4]

[1]Electrical Engineering Division, Control and Robotics Department, School of Engineering, UNAM, Mexico City 04510, Mexico
[2]Robotics and Control Department, School of Electronic Sciences, BUAP, Puebla 72000, Mexico
[3]Mechatronics Section, Electrical Engineering Department, CINVESTAV–IPN, Mexico City 07300, Mexico
[4]Applied Physics Division, Electronics and Telecommunications Department, CISESE–CONACYT, Ensenada 22860, Mexico

Corresponding author: Pablo Sánchez–Sánchez (pablo.sanchez@correo.buap.mx)

This work was supported in part by the DGAPA–UNAM under Grant IN117820, and in part by the Grant PRODEP ID–90934.

ABSTRACT Force–feedback rendering of virtual tools in contact with virtual surfaces is an open research problem. A common strategy is to define an integrable implicit equation that represents the surface and to use the penalty method to haptically render the corresponding force. Nevertheless, some surfaces cannot be described this way, e. g., soft tissues, thus making real–time force rendering inaccurate. In this work, we propose a bilateral teleoperation scheme where the interaction between a virtual tool and a surface is represented by a virtual slave robot subject to holonomic and nonholonomic constraints. This approach allows to define a position–force controller that, on the one hand, guarantees position tracking between the virtual tool and the robot end–effector by projecting the operator's movements on a computer screen via the simulated robotic dynamic model, while on the other hand the felt reaction force is rendered by considering the forces generated by the master robot as references for the control system. To the best of the authors' knowledge this is the first time that nonholonomic constraints have been used to reproduce haptically deformable surfaces in a virtual reality system. Since such tactile sensations are subjective, a set of experiments are carried out to validate the proposed approach by comparing the force responses when using holonomic and nonholonomic constraints. Special emphasis is made on describing how both representations can be used to reproduce some tactile phenomena presented in medical training simulators.

INDEX TERMS Haptics, virtual reality system, bilateral teleoperation, holonomic constraint, nonholonomic constraint, position–force control.

I. INTRODUCTION

Teleoperation and virtual reality systems are two different approaches developed to allow human operators to interact with environments without being in actual physical contact with them. In the former, a slave robot performs a task on a remote environment commanded by a master manipulator that provides kinesthetic feedback to the human operator. In the latter, the remote environment is replaced by a computer simulation in order for the operator to perceive the effect caused by his/her movements in a virtual environment. Both approaches have been used extensively, but virtual reality systems have gained a growing interest in the scientific community.

The associate editor coordinating the review of this manuscript and approving it for publication was Zheng Chen.

A. BACKGROUND

An example of robotic systems interacting with virtual environments are medical training simulators, where it is essential to describe as accurately as possible such virtual environments both in a visual and in a tactile way. Recently, the research efforts are principally aimed at transmitting the force applied on the real environment to the operator. The accuracy depends on a number of factors that together heighten the realism of virtual reality applications. One of them is the choice between an impedance or an admittance device depending on the characteristics of the virtual environment (e.g., stiffness, inertia, damping, friction) and the type of

contact with the virtual tool. Another factor is the presence of masking phenomena [1], [2], where the force is not sent to the operator in direct reflection, but instead inertial properties are transmitted which do not belong neither to the environment nor to the haptic device [3]. However, a crucial factor to take into account is the mathematical description of the surfaces and its viability to render forces in a realistic manner.

B. PROBLEM FORMULATION

In this work we deal with the enhancement of realism for robotic systems interacting with virtual rigid or deformable environments. Realism is essential in simulations for medical applications since for example medical trainees need to have a high accurate force response to enhance their tactile perception of living tissues. Achieving this goal can be done by considering a simulation training device as a bilateral teleoperation system as suggested in [4] and [5]. This entails the design of a control law which considers the dynamic models of the master robot and the virtual environment to achieve position–force tracking and, in consequence, to take advantage of the masking effect. With this in mind, the dynamic model of the tool held by the operator can be taken to be the dynamic model of the virtual medical tool in contact with the virtual surface [6]. The relevance of this approach is that, apart from holding a stylus attached to the master robot end-effector with the form of such a medical tool (e.g., scalpel, forceps, retractors, endoscope), the operator can feel the dynamic characteristics of the contact [7]. Additionally, to be useful in building real-world applications, a rendering system using this approach must be capable of displaying the models found in common graphic applications [8]. Finally, although time varying delays have been an area of active research for many years [9]–[11], it is worthy pointing out that the system under study does not contain any time delays because both the control law for the physical manipulator as well as the virtual slave (tool) are programmed in the same computer.

C. LITERATURE REVIEW

Haptic rendering should reproduce the location and the geometry of the analytical representation of a desired virtual surface, and if necessary also its texture and internal structure, as well as friction effects [12]. With respect to the basic characteristics of location and geometry, the case of robot manipulators subject to holonomic constraints has been widely used in virtual reality systems. For example, in [13] a virtual planar manipulator is simulated with two rotational joints constrained to a circle in the plane. In this case, an operator attempts to keep a constant velocity tangential to the constraint and an admittance–type robot exerts the virtual normal force produced during the contact. In addition, in [14] a *geometric constraint solver* is implemented to find all the possible configurations of a mobile object satisfying a set of holonomic constraints. On the other hand, in a bilateral teleoperation context, a virtual planar surface is used in [15] to reproduce the force exerted by a slave robot in contact with a real plane at the remote environment. As a result, the operator perceives such plane at the local environment through the forces produced by the master robot.

The common factor in the aforementioned examples is that the equation of the virtual surface is algebraically defined in task–space coordinates or, at least, two of its geometric characteristics (e.g., normal and tangential vectors or distance and angle relationships between points, lines and planes) are known [16]. This simplifies both graphic and haptic implementation due to the fact that the surface is defined implicitly [17], but it limits force rendering to a very small neighborhood restricted to a single point on the surface by using the penalty method. For that reason some key features that a virtual reality system should have in order to increase realism cannot be achieved. For example, lateral and axial forces present in classic deformation phenomena, such as needle puncture or needle steering, cannot be rendered properly when using the penalty method. In an effort to overcome this drawback, in [18] a methodology is developed to determine needle forces during deformable body puncture based on obtaining axial and lateral forces from a finite set of mesh nodes for 2D surfaces. On the other hand, in [19] a particle–based method for the modeling and simulation of elastoplastic solids as multibody systems is proposed. Even when this method heightens the realism by adding the dynamic model of the surface, it is computationally expensive and non suitable to be used on real–time applications. Therefore, it is necessary to make a trade–off between the complexity of the surface models and force rendering realism.

A very little exploited tool to solve the problems mentioned above is the nonholonomic representation of virtual surfaces. The works in the literature dealing with this kind of constraints are for the most part focused on the control of wheeled cars [20]–[23]. In a different application, Webster *et al.* [24] uses a nonholonomic model to represent the deformation of a needle steering immersed in soft tissue. Interestingly, the system is modeled to control the insertion and the shaft rotation speed in 2D. Rucker *et al.* [25] uses 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 with visual feedback. Faurling *et al.* [26] outline 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. Nevertheless, no further information or mathematical modeling about such phenomenon is presented.

Another important problem when dealing with nonholonomic constraints is that of numerical solvers for Lagrange multipliers representing the necessary forces to satisfy the constraints. In this aspect, there are few results in the literature. A basic method presented in [27] consists in inverting the dynamics of the Lagrange–d'Alembert equations to solve for the multipliers. Such a solution is not stable as mentioned

in [28], where a stable computation of the Lagrange multipliers is proposed for holonomic constraints. As to solving nonholonomic constraints, in [29] a solution is introduced for port Hamiltonian systems based on the theory of feedback integrators.

D. CONTRIBUTION

In this paper, we propose a teleoperation based approach to render forces haptically from 3D virtual rigid or deformable environments. By using a spherical surface as a representative example, experiments are carried out to show how a human operator is capable of feeling it with an enhanced dexterity sensation while getting at the time some visual feedback. To achieve this goal, it is considered that the real and virtual environments have independent sets of coordinates and that the virtual tool has its own dynamic characteristics, represented by a robotic manipulator model. A needle–shaped avatar is attached to the end–effector of the virtual robot to allow the operator to get visual information of his/her movements. Regarding the haptic interaction, a position–force control scheme is employed that has been used previously in robot teleoperation systems to perform force rendering for actual physical slaves. Furthermore, in this work both holonomic and nonholonomic constraints are used to model the virtual surface so that the operator can feel not only contact forces but also the penetration of the object should it be deformable. The main two contributions of the paper are

1) To the best of the authors' knowledge, this is the first time that nonholonomic constraints are used to feel the penetration of a deformable object in haptic systems.
2) A novel computation approach for Lagrange multipliers of this kind of constraints is proposed.

E. ORGANIZATION OF THE MANUSCRIPT

The rest of the paper is organized as follows. In Section II, 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 representations is also introduced. Section III presents the position–force control scheme and its integration with the graphic application. In Section IV, the experimental setup and the corresponding outcomes are shown to illustrate the effectiveness of the proposed approach. Finally, some concluding remarks are given in Section V.

II. SYSTEM OVERVIEW

To describe the operation of the haptic system we consider two independent sets of task space coordinates as shown in Figure 1. The operator manipulates the haptic interface, i.e., the master robot in the real environment with Cartesian coordinates $\boldsymbol{x}_{\mathrm{m}} \in SE(3)$, where $\boldsymbol{p}_{\mathrm{m}} \in \mathbb{R}^3$ is the end–effector position and $\boldsymbol{R}_{\mathrm{m}} \in SO(3)$ represents its orientation. On the other hand, the virtual tool must respond to the movements of such interface in the virtual environment with Cartesian

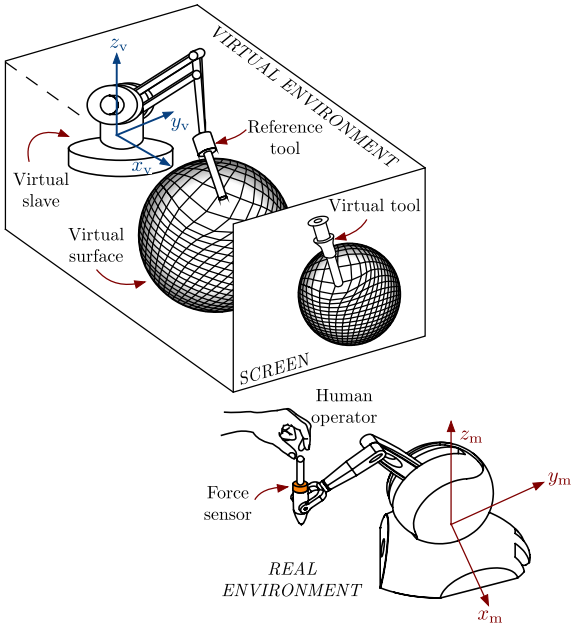

FIGURE 1. Haptic system.

coordinates $\boldsymbol{x}_{\mathrm{v}} \in SE(3)$, where $\boldsymbol{p}_{\mathrm{v}} \in \mathbb{R}^3$ is the virtual tool position and $\boldsymbol{R}_{\mathrm{v}} \in SO(3)$ represents its orientation.

In a teleoperation context, both position and orientation of the master robot are taken as reference values for the virtual tool and they are projected visually on the screen through the avatar of the system. The operator moves freely the virtual tool until a contact with the virtual surface takes place. From that moment on, the master robot exerts a force that is used as well as a reference for the virtual manipulator and must be applied on the virtual surface. By closing the feedback loop, the control algorithm produces a tactile sensation on the operator. Ideally, both visual and haptic feedback must match, thus allowing to have a visual reference of the virtual tool combined with the feeling of possible dynamic changes of the contact with the virtual sphere. In a similar approach, Faurling *et al.* [13] make evident that the virtual environment can be represented by a set of generalized coordinates $\boldsymbol{q}_{\mathrm{v}} \in \mathbb{R}^n$, which are related to the task–space coordinates of the virtual robot by the nonlinear kinematic equation

$$\boldsymbol{x}_{\mathrm{v}} = \boldsymbol{f}(\boldsymbol{q}_{\mathrm{v}}). \tag{1}$$

The set of coordinates $\boldsymbol{q}_{\mathrm{v}}$ allows the dynamic model of the virtual tool to be fully described in terms of the Euler–Lagrange equations of motion. Furthermore, we can embed into the kinematic map (1) a set of holonomic or nonholonomic constraints that represent the virtual surface, relating the master robot task–space coordinates with the virtual robot task–space coordinates. The virtual tool behaves according to the equations of motion simulated in the computer and it always satisfies the programmed constraints.

A. DYNAMIC MODEL AND PROPERTIES

Consider a real master (m) and a virtual slave (v) robot system made up of two manipulators both of them with n degrees of freedom. 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

$$\boldsymbol{H}_{\mathrm{m}}(\boldsymbol{q}_{\mathrm{m}})\ddot{\boldsymbol{q}}_{\mathrm{m}} + \boldsymbol{C}_{\mathrm{m}}(\boldsymbol{q}_{\mathrm{m}}, \dot{\boldsymbol{q}}_{\mathrm{m}})\dot{\boldsymbol{q}}_{\mathrm{m}} + \boldsymbol{D}_{\mathrm{m}}\dot{\boldsymbol{q}}_{\mathrm{m}} + \boldsymbol{g}_{\mathrm{m}}(\boldsymbol{q}_{\mathrm{m}}) = \boldsymbol{\tau}_{\mathrm{m}} - \boldsymbol{\tau}_{\mathrm{h}}\,, \quad (2)$$

and the virtual slave dynamics is modeled by

$$\boldsymbol{H}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}})\ddot{\boldsymbol{q}}_{\mathrm{v}} + \boldsymbol{C}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}, \dot{\boldsymbol{q}}_{\mathrm{v}})\dot{\boldsymbol{q}}_{\mathrm{v}} + \boldsymbol{D}_{\mathrm{v}}\dot{\boldsymbol{q}}_{\mathrm{v}} + \boldsymbol{g}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}) = \boldsymbol{\tau}_{\mathrm{v}} + \boldsymbol{\tau}_{\mathrm{e}}, \quad (3)$$

where $\boldsymbol{q}_{\mathrm{m,v}} \in \mathbb{R}^n$ are the vectors of generalized coordinates, $\boldsymbol{H}_{\mathrm{m}}(\boldsymbol{q}_{\mathrm{m}}), \boldsymbol{H}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}) \in \mathbb{R}^{n\times n}$ are the positive definite inertia matrices, $\boldsymbol{C}_{\mathrm{m}}(\boldsymbol{q}_{\mathrm{m}}, \dot{\boldsymbol{q}}_{\mathrm{m}})\dot{\boldsymbol{q}}_{\mathrm{m}}, \boldsymbol{C}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}, \dot{\boldsymbol{q}}_{\mathrm{v}})\dot{\boldsymbol{q}}_{\mathrm{v}} \in \mathbb{R}^n$ are the vectors of Coriolis and centripetal forces, $\boldsymbol{D}_{\mathrm{m}}, \boldsymbol{D}_{\mathrm{v}} \in \mathbb{R}^{n\times n}$ are diagonal matrices of viscous friction coefficients, $\boldsymbol{g}_{\mathrm{m}}(\boldsymbol{q}_{\mathrm{m}}), \boldsymbol{g}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}) \in \mathbb{R}^n$ are the vectors of gravitational torques, and $\boldsymbol{\tau}_{\mathrm{m}}, \boldsymbol{\tau}_{\mathrm{v}} \in \mathbb{R}^n$ are the vectors of generalized inputs,[1] $\boldsymbol{\tau}_{\mathrm{h}} \in \mathbb{R}^n$ is the torque applied by the human operator on the master side and $\boldsymbol{\tau}_{\mathrm{e}} \in \mathbb{R}^n$ is the virtual torque generated due to the contact with the virtual environment. Note that in general some disturbances and complex friction terms could be included in the dynamic models (2)–(3). However, the goal of this work is not to test the robustness of the control law, but to get the maximal realism in a haptic system where virtual deformable surfaces are described by nonholomic constraints. For that reason, it is unnecessary to include perturbations or friction terms on the virtual side because they exist only if you want to program them, and we do not want to. On the other hand, the actual physical manipulator can undergo some external perturbations of course, but the robot used in the experiments is a highly reliable haptic device practically free of them and of nonlinear friction effects as well. Therefore, neither external perturbations nor complex friction terms are considered in (2)–(3).

Property 1: With a proper definition of the robot parameters, it is [30, pp. 263]

$$\boldsymbol{H}_i(\boldsymbol{q}_i)\ddot{\boldsymbol{q}}_i + \boldsymbol{C}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i)\dot{\boldsymbol{q}}_i + \boldsymbol{D}_i\dot{\boldsymbol{q}}_i + \boldsymbol{g}_i(\boldsymbol{q}_i) = \boldsymbol{Y}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i, \ddot{\boldsymbol{q}}_i)\boldsymbol{\theta}_i\,, \quad (4)$$

where $\boldsymbol{Y}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i, \ddot{\boldsymbol{q}}_i) \in \mathbb{R}^{n\times l_i}$ is the regressor and $\boldsymbol{\theta}_i \in \mathbb{R}^{l_i}$ is a constant vector of parameters, for $i = \mathrm{m}, \mathrm{v}$ and l_i the number of parameters of each robot model. △

The interested reader can find an example about how to choose a vector of parameters in [31].

Assumption 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) and (3) need to be equal. △

[1]Note that the virtual robot, denoted with the subscript v, takes the place of the slave robot in a master–slave teleoperation scheme.

The torque applied by the human operator can be characterized as

$$\boldsymbol{\tau}_{\mathrm{h}} = \boldsymbol{J}_{\mathrm{m}}^{\mathrm{T}}(\boldsymbol{q}_{\mathrm{m}})\boldsymbol{F}_{\mathrm{h}}, \quad (5)$$

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

$$\boldsymbol{\tau}_{\mathrm{e}} = \boldsymbol{J}_{\mathrm{v}}^{\mathrm{T}}(\boldsymbol{q}_{\mathrm{v}})\boldsymbol{F}_{\mathrm{v}}, \quad (6)$$

where $\boldsymbol{F}_{\mathrm{v}} \in \mathbb{R}^3$ is the force applied in virtual task–space coordinates and $\boldsymbol{J}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}) \in \mathbb{R}^{3\times n}$ is the geometric Jacobian of the virtual manipulator. Note that since our test bed has only three degrees of freedom actuated, we do not consider that the operator can apply a torque on the master end–effector and that the virtual force has to comply with this as well.

B. VIRTUAL HOLONOMIC CONSTRAINTS

It is said that a constraint is well defined if it is associated with physically realizable constraint forces. That is the case for example of an industrial robot in contact with a car bonnet in a painting or welding task. But for virtual environments there do not exist constraint forces associated to any contact with a surface unless they are properly programmed. Thus, the constraint is not well defined and it is called *virtual constraint* [32]. For instance, one can assume that the virtual robot is subject to k holonomic constraints given by

$$\boldsymbol{\varphi}_{\mathrm{v}}(\boldsymbol{x}_{\mathrm{v}}) = \boldsymbol{0}, \quad (7)$$

where a suitable normalization is done for its gradient, $\boldsymbol{J}_{\varphi\mathrm{xv}}(\boldsymbol{x}_{\mathrm{v}}) = \nabla\boldsymbol{\varphi}_{\mathrm{v}}(\boldsymbol{x}_{\mathrm{v}}) \in \mathbb{R}^{k\times n}$, to have rows with unitary norms. The representation of the same constraint (7) but in virtual coordinates is

$$\boldsymbol{\varphi}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}) = \boldsymbol{0}. \quad (8)$$

The gradient of constraint (8) 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

$$\boldsymbol{J}_{\varphi\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}) = \boldsymbol{J}_{\varphi\mathrm{xv}}(\boldsymbol{x}_{\mathrm{v}})\boldsymbol{J}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}). \quad (9)$$

Hence, the torque due to the contact with the virtual surface in (3) can be defined as

$$\boldsymbol{\tau}_{\mathrm{e}} = \boldsymbol{J}_{\varphi\mathrm{v}}^{\mathrm{T}}(\boldsymbol{q}_{\mathrm{v}})\boldsymbol{\lambda}_{\mathrm{v}}, \quad (10)$$

where $\boldsymbol{\lambda}_{\mathrm{v}} \in \mathbb{R}^k$ is a vector of Lagrange multipliers that represents the virtual forces applied on the surface. This allows to rewrite (3) as

$$\boldsymbol{H}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}})\ddot{\boldsymbol{q}}_{\mathrm{v}} + \boldsymbol{C}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}, \dot{\boldsymbol{q}}_{\mathrm{v}})\dot{\boldsymbol{q}}_{\mathrm{v}} + \boldsymbol{D}_{\mathrm{v}}\dot{\boldsymbol{q}}_{\mathrm{v}} + \boldsymbol{g}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}) = \boldsymbol{\tau}_{\mathrm{v}} + \boldsymbol{J}_{\varphi\mathrm{v}}^{\mathrm{T}}(\boldsymbol{q}_{\mathrm{v}})\boldsymbol{\lambda}_{\mathrm{v}}. \quad (11)$$

Constraint (8) reduces the number of degrees of freedom of the virtual robot and the dimension of its configuration space to a $(n-k)$-dimensional submanifold [33].

C. VIRTUAL NONHOLONOMIC CONSTRAINTS

Nonholonomic constraints cannot be expressed as a function of only the generalized coordinates as in (7) or (8). Instead, they are commonly expressed by means of the so–called *Pfaffian* constraints. In this work, 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

$$\boldsymbol{A}_{\mathrm{v}}(\boldsymbol{x}_{\mathrm{v}})\boldsymbol{v}_{\mathrm{v}} = \boldsymbol{0}\,, \tag{12}$$

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 [13]

$$\boldsymbol{A}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}) = \boldsymbol{A}_{\mathrm{v}}(\boldsymbol{x}_{\mathrm{v}})\boldsymbol{J}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}})\,. \tag{13}$$

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

$$\boldsymbol{A}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}})\dot{\boldsymbol{q}}_{\mathrm{v}} = \boldsymbol{0}\,, \tag{14}$$

the torque due to the contact with the virtual environment in (3) can be expressed as

$$\boldsymbol{\tau}_{\mathrm{e}} = \boldsymbol{A}_{\mathrm{v}}^{\mathrm{T}}(\boldsymbol{q}_{\mathrm{v}})\boldsymbol{\lambda}_{\mathbf{v}}, \tag{15}$$

where $\boldsymbol{\lambda}_{\mathbf{v}} \in \mathbb{R}^k$ is the vector of Lagrange multipliers which determines the magnitude of the constraint forces on the virtual surface. Then, it is possible to rewrite equation (3) as

$$\begin{aligned}\boldsymbol{H}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}})\ddot{\boldsymbol{q}}_{\mathrm{v}} + \boldsymbol{C}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}, \dot{\boldsymbol{q}}_{\mathrm{v}})\dot{\boldsymbol{q}}_{\mathrm{v}} + \boldsymbol{D}_{\mathrm{v}}\dot{\boldsymbol{q}}_{\mathrm{v}} + \boldsymbol{g}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}) \\ = \boldsymbol{\tau}_{\mathrm{v}} + \boldsymbol{A}_{\mathrm{v}}^{\mathrm{T}}(\boldsymbol{q}_{\mathrm{v}})\boldsymbol{\lambda}_{\mathrm{v}}.\end{aligned} \tag{16}$$

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 [33].

III. CONTROL SCHEME

When working with virtual environments, a common practice consists in associating the position of the haptic interface directly to that of the avatar. Therefore, it is not necessary any position control since the operator's movements are reflected in an exact way on the graphic application. On the contrary to this approach, we use the position of the haptic master robot as desired trajectory for the virtual robot to track. Although this procedure appears to be unnecessarily more complex, it allows to assign a *desired* dynamic behavior to the virtual environment, which in turn allows the operator to feel the virtual tool as a consequence of the masking effect by the implementation of a control law designed for this goal. In this section we propose the use of two hybrid control algorithms for haptic interaction with constrained virtual mechanical systems based on a scheme employed in robotic teleoperation to achieve both position and force tracking. A block diagram is shown in Figure 2.

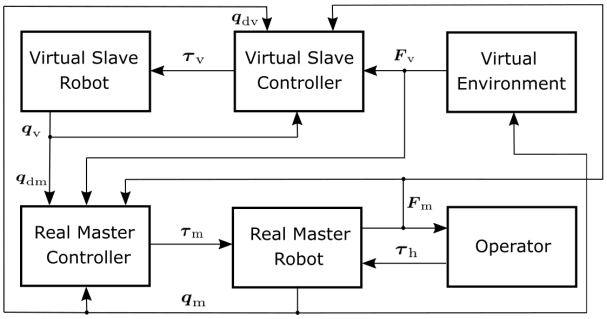

FIGURE 2. Block diagram of the proposed scheme.

A. POSITION CONTROL

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

$$\boldsymbol{q}_{\mathrm{d}i}(t) \triangleq \boldsymbol{q}_j(t) \tag{17}$$

as desired position trajectories, and

$$\dot{\boldsymbol{q}}_{\mathrm{d}i}(t) \triangleq \dot{\boldsymbol{q}}_j(t) \tag{18}$$

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

$$\Delta\boldsymbol{q}_i = \boldsymbol{q}_i - \boldsymbol{q}_{\mathrm{d}i}. \tag{19}$$

Based on [34], we propose

$$\boldsymbol{s}_i = \dot{\boldsymbol{q}}_i - \dot{\boldsymbol{q}}_{\mathrm{d}i} + \boldsymbol{\Lambda}_{\mathrm{x}i}\Delta\boldsymbol{q}_i \tag{20}$$

and

$$\dot{\boldsymbol{\sigma}}_i = \boldsymbol{K}_{\beta i}\boldsymbol{s}_i + \mathrm{sign}(\boldsymbol{s}_i), \tag{21}$$

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

$$\mathrm{sign}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}. \tag{22}$$

Consider now the velocity reference as

$$\dot{\boldsymbol{q}}_{\mathrm{r}i} = \dot{\boldsymbol{q}}_{\mathrm{r}i} + \boldsymbol{\Lambda}_{\mathrm{x}i}\Delta\boldsymbol{q}_i - \boldsymbol{K}_{\gamma i}\boldsymbol{\sigma}_i, \tag{23}$$

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

$$\boldsymbol{s}_{\mathrm{a}i} = \dot{\boldsymbol{q}}_i - \dot{\boldsymbol{q}}_{\mathrm{r}i}. \tag{24}$$

Suppose that both robots are in free motion; for that case the control laws for the master and the virtual manipulators are proposed as[2]

$$\boldsymbol{\tau}_{\mathrm{m}} = -\boldsymbol{K}_{\mathrm{am}}\dot{\boldsymbol{q}}_{\mathrm{m}} - \boldsymbol{K}_{\mathrm{pm}}\boldsymbol{s}_{\mathrm{am}} \tag{25}$$

$$\boldsymbol{\tau}_{\mathrm{v}} = \boldsymbol{K}_{\mathrm{av}}\dot{\boldsymbol{q}}_{\mathrm{v}} + \boldsymbol{K}_{\mathrm{pv}}\boldsymbol{s}_{\mathrm{av}}, \tag{26}$$

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.

[2]Note that the discontinuity due to the sign function in (21) disappears in the control input because it is the integral $\boldsymbol{\sigma}_i$ what is used in (23), thus avoiding undesirable *chattering* effects.

B. POSITION–FORCE CONTROL FOR VIRTUAL HOLONOMIC CONSTRAINTS

As explained before, we consider that a sphere can be used as a representative example among many possible kind of objects with different dimensions and geometries because it is rounded but infinitesimally it can be reconstructed by tangent planes on each point of its surface. Since the movement is constrained only in one direction, it is also only necessary to take into account the one–dimensional case for the constraint ($\varphi_{\mathrm{v}} : \mathbb{R}^n \rightarrow \mathbb{R}$). That means, however, that the force generated is normal at a single point of the contact surface and other possible effects, as friction or tangential forces, must be omitted [12]. To reproduce this effect, we use first the implicit surface method, so that $\boldsymbol{\lambda}_{\mathrm{v}} = \lambda_{\mathrm{v}} \in \mathbb{R}$ represents the normal force of the virtual manipulator on the sphere. To reflect contact forces, a Lagrange multiplier is computed by means of the *Generic Penalty Method* used in [35], i.e.,

$$\lambda_{\mathrm{v}} = \alpha_{\mathrm{v}} \left(\ddot{\varphi}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}) + 2\xi\omega_{\mathrm{n}}\dot{\varphi}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}) + \omega_{\mathrm{n}}^2\varphi_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}) \right), \tag{27}$$

where $\xi, \omega_{\mathrm{n}} > 0$. Note that the corresponding virtual force can be obtained through the relationship

$$\boldsymbol{F}_{\mathrm{v}} = \boldsymbol{J}_{\varphi \mathrm{xv}}^{\mathrm{T}}(\boldsymbol{x}_{\mathrm{v}})\lambda_{\mathrm{v}}. \tag{28}$$

On the other hand, the force exerted by the human operator, $\boldsymbol{F}_{\mathrm{h}}$, can be measured with a sensor mounted at the master manipulator end–effector. By taking into account (27)–(28) together with (5)–(6), a PI force controller can be used for the virtual reality system. Define

$$\boldsymbol{F}_{\mathrm{d}i}(t) \triangleq \boldsymbol{F}_j(t), \tag{29}$$

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

$$\Delta\boldsymbol{F}_i = \boldsymbol{F}_i - \boldsymbol{F}_{\mathrm{d}i} \tag{30}$$

and the corresponding integral, the momenta tracking error is

$$\Delta\boldsymbol{p}_i = \int_0^t \Delta\boldsymbol{F}_i \, \mathrm{d}t. \tag{31}$$

Note that we use the standard notation for momenta $\boldsymbol{p}_i$ although the same notation is used for position. We claim there is no confusion because it always appears $\Delta\boldsymbol{p}_i$ for that case. Instead of (25)–(26), for the master and the virtual robot the corresponding control laws are respectively given by

$$\begin{aligned} \boldsymbol{\tau}_{\mathrm{m}} &= \boldsymbol{Y}_{\mathrm{m}}(\boldsymbol{q}_{\mathrm{m}}, \dot{\boldsymbol{q}}_{\mathrm{m}}, \ddot{\boldsymbol{q}}_{\mathrm{m}})\boldsymbol{\theta}_{\mathrm{m}} - \boldsymbol{K}_{\mathrm{am}}\dot{\boldsymbol{q}}_{\mathrm{m}} - \boldsymbol{K}_{\mathrm{pm}}\boldsymbol{s}_{\mathrm{am}} \\ &\quad + \boldsymbol{J}_{\mathrm{m}}^{\mathrm{T}}(\boldsymbol{q}_{\mathrm{m}})(\boldsymbol{F}_{\mathrm{v}} - \boldsymbol{K}_{\mathrm{fm}}\Delta\boldsymbol{p}_{\mathrm{h}}) \qquad (32) \\ \boldsymbol{\tau}_{\mathrm{v}} &= \boldsymbol{K}_{\mathrm{av}}\dot{\boldsymbol{q}}_{\mathrm{v}} + \boldsymbol{K}_{\mathrm{pv}}\boldsymbol{s}_{\mathrm{av}} - \boldsymbol{J}_{\mathrm{v}}^{\mathrm{T}}(\boldsymbol{q}_{\mathrm{v}})(\boldsymbol{F}_{\mathrm{h}} - \boldsymbol{K}_{\mathrm{fv}}\Delta\boldsymbol{p}_{\mathrm{v}}), \qquad (33) \end{aligned}$$

where $\boldsymbol{K}_{fi} \in \mathbb{R}^{n\times n}$ are diagonal matrices. In the master control law (32) the term $\boldsymbol{Y}_{\mathrm{m}}(\boldsymbol{q}_{\mathrm{m}}, \dot{\boldsymbol{q}}_{\mathrm{m}}, \ddot{\boldsymbol{q}}_{\mathrm{m}})\boldsymbol{\theta}_{\mathrm{m}}$ is employed to enhance the transparency of the system by canceling online the master manipulator dynamics. Therefore, the identification of the model parameters $\boldsymbol{\theta}_{\mathrm{m}}$ must be carried out offline, e.g., by employing metholodogies such as those described in [36]. Although one can expect that there will be an error in the nominal value of $\boldsymbol{\theta}_{\mathrm{m}}$ used in the implementation, this drawback is compensated by the fact that once in contact with the virtual surface the operator will necessarily move the master end–effector slowly which implies that $\dot{\boldsymbol{q}}_{\mathrm{m}}$ and $\ddot{\boldsymbol{q}}_{\mathrm{m}}$ are very small and therefore the elements of $\boldsymbol{\theta}_{\mathrm{m}}$ associated with the most complex nonlinear effects in the robot dynamics (the inertia matrix, the Coriolis and centripetal forces and friction) become negligible. For that reason, as shown by the experimental results in the next section, tracking errors are smaller in constrained motion than in free motion, which is in our advantage.

C. POSITION–FORCE CONTROL FOR VIRTUAL NONHOLONOMIC CONSTRAINTS

When constraints are nonholonomic, they cannot be defined as a function of a set of generalized coordinates and they have to be defined in the Pfaffian form (12) or equivalently (14). As a consequence, the Lagrange multipliers cannot be computed by using (27), so that a problem to be solved is how to get them in order to satisfy (16). Unfortunately, most computation methods for Lagrange multipliers are thought for systems with holonomic constraints [28], [35], [37] and thus require a position–level definition like those in (7) or (8). A solution is given in [27], but it turns out to be unstable because its underlying mechanism is a second order integrator with zero input. In this work, we propose a modification of the approach employed in [28] for holomic constraints. Let us define, for simplicity's sake, $\boldsymbol{H}_{\mathrm{v}} = \boldsymbol{H}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}), \boldsymbol{C}_{\mathrm{v}} = \boldsymbol{C}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}, \dot{\boldsymbol{q}}_{\mathrm{v}})$, $\boldsymbol{g}_{\mathrm{v}} = \boldsymbol{g}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}), \boldsymbol{A}_{\mathrm{v}} = \boldsymbol{A}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}})$, and $\boldsymbol{\psi} = \boldsymbol{\psi}(\boldsymbol{q}_{\mathrm{v}}, \dot{\boldsymbol{q}}_{\mathrm{v}}) = \boldsymbol{A}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}})\dot{\boldsymbol{q}}_{\mathrm{v}}$. Note that actually $\boldsymbol{\psi}$ represents the nonholonomic constraint given in (14), whose derivative is given by

$$\dot{\boldsymbol{\psi}} = \boldsymbol{A}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}})\ddot{\boldsymbol{q}}_{\mathrm{v}} + \dot{\boldsymbol{A}}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}})\dot{\boldsymbol{q}}_{\mathrm{v}} = 0. \tag{34}$$

By recalling that $\boldsymbol{H}_{\mathrm{v}}$ is a positive definite matrix whose inverse always exists, $\ddot{\boldsymbol{q}}_{\mathrm{v}}$ can be gotten from (16) to get the Lagrange multipliers by substituting in (34) as

$$\begin{aligned} \boldsymbol{\lambda}_{\mathrm{v}} = (\boldsymbol{A}_{\mathrm{v}}\boldsymbol{H}_{\mathrm{v}}^{-1}\boldsymbol{A}_{\mathrm{v}}^{\mathrm{T}})^{-1} \big[\dot{\boldsymbol{\psi}} - \dot{\boldsymbol{A}}_{\mathrm{v}}\dot{\boldsymbol{q}}_{\mathrm{v}} \\ - \boldsymbol{A}_{\mathrm{v}}\boldsymbol{H}_{\mathrm{v}}^{-1}(\boldsymbol{\tau}_{\mathrm{v}} - \boldsymbol{C}_{\mathrm{v}}\dot{\boldsymbol{q}}_{\mathrm{v}} - \boldsymbol{D}_{\mathrm{v}}\dot{\boldsymbol{q}}_{\mathrm{v}} - \boldsymbol{g}_{\mathrm{v}}) \big]. \end{aligned} \tag{35}$$

Note that it is very tempting to substitute $\dot{\boldsymbol{\psi}} \equiv \boldsymbol{0}$, but the fact is that the constraint $\boldsymbol{\psi} = \boldsymbol{0}$ does not hold on its own and it has to be *enforced* when computing and simulating the virtual robot dynamics. In principle of course, one would have to calculate $\dot{\boldsymbol{\psi}}$ from (34) and then compute its integral to get $\boldsymbol{\psi} = \boldsymbol{0}$. However, an integrator is intrinsically unstable, which means that any small numerical drift will cause eventually to obtain a value for $\boldsymbol{\psi}$ far away from zero. Pretty much in the same fashion as done in [28], we propose instead the following solution to describe the derivative of the nonholonomic constraint in substitution of (34)

$$\dot{\boldsymbol{\psi}} = -2\alpha_{\mathrm{v}}\boldsymbol{\psi} - \beta_{\mathrm{v}} \int_{t_0}^{t} \boldsymbol{\psi} \, d\vartheta, \tag{36}$$

with $\alpha_{\mathrm{v}}, \beta_{\mathrm{v}} > 0$ chosen to ensure fast convergence to the origin while the initial condition of the integral is set to zero.

This equation is therefore completely equivalent to $\dot{\boldsymbol{\psi}} = \boldsymbol{\psi} = 0$ but stable. Since $\dot{\boldsymbol{\psi}}$ in (36) is computed without needing the knowledge of $\ddot{\boldsymbol{q}}_{\mathrm{v}}$, it can be substituted in (35) to get $\boldsymbol{\lambda}_{\mathrm{v}}$ without causing a computational loop. Also note that on the contrary to [28] where it is the second derivative of the constraint what is involved in computing the Lagrange multipliers, here it is the first derivative of $\boldsymbol{\psi}$ which is involved, so that one could use $\dot{\boldsymbol{\psi}} = -a\boldsymbol{\psi}$ with $a > 0$ as well. Experimental results show that it is better to employ the relationship given in (36). To the best of the author's knowledge this is the first time that this relationship is used to compute the Lagrange multipliers for nonholonomic constraints. Now, by substituting (35) and (36) in the motion equation (16), we get a complete description of the dynamics of a virtual system satisfying constraint (14).

Note that the virtual surface has to be reproduced for the human operator through a force controller. The first step for this goal is to take into consideration that the virtual force $\boldsymbol{F}_{\mathrm{v}}$ is related to $\boldsymbol{\lambda}_{\mathrm{v}}$ by the relationship

$$\boldsymbol{\lambda}_{\mathrm{v}} = \{\boldsymbol{A}_{\mathrm{v}}\boldsymbol{A}_{\mathrm{v}}^{\mathrm{T}}\}^{-1}\boldsymbol{A}_{\mathrm{v}}\boldsymbol{J}_{\mathrm{v}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{v}}, \tag{37}$$

which is obtained by equating (6) with (15). Then, desired values are defined as

$$\boldsymbol{\lambda}_{\mathrm{d}i}(t) \triangleq \boldsymbol{\lambda}_{j}(t), \tag{38}$$

where as usual if $i = \mathrm{v}$ then $j = \mathrm{m}$ and vice versa. However, there does not exist an actual physical sphere on the master side, so that one cannot get a "real" Langrange multiplier $\boldsymbol{\lambda}_{\mathrm{m}}$ to be employed as desired value; but since we want to reconstruct the virtual surface on the master side we propose to use

$$\boldsymbol{\lambda}_{\mathrm{m}} = \{\boldsymbol{A}_{\mathrm{v}}\boldsymbol{A}_{\mathrm{v}}^{\mathrm{T}}\}^{-1}\boldsymbol{A}_{\mathrm{v}}\boldsymbol{J}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{h}}, \tag{39}$$

by mimicking (37) and where $\boldsymbol{F}_{\mathrm{h}}$ will be gotten from the force sensor once the surface reconstruction begins.

Finally, instead of (25)–(26), for the master and the virtual robot the proposed position–force control for the virtual dynamic system subject to nonholonomic constraints is

$$\begin{aligned}\boldsymbol{\tau}_{\mathrm{m}} &= \boldsymbol{Y}_{\mathrm{m}}(\boldsymbol{q}_{\mathrm{m}}, \dot{\boldsymbol{q}}_{\mathrm{m}}, \ddot{\boldsymbol{q}}_{\mathrm{m}})\boldsymbol{\theta}_{\mathrm{m}} - \boldsymbol{K}_{\mathrm{am}}\dot{\boldsymbol{q}}_{\mathrm{m}} - \boldsymbol{K}_{\mathrm{pm}}\boldsymbol{s}_{\mathrm{am}} \\ &\quad + \boldsymbol{A}_{\mathrm{v}}^{\mathrm{T}}(\boldsymbol{q}_{\mathrm{v}})(\boldsymbol{\lambda}_{\mathrm{v}} - \boldsymbol{K}_{\mathrm{fm}}\Delta\boldsymbol{\lambda}_{\mathrm{m}}) \qquad (40)\\ \boldsymbol{\tau}_{\mathrm{v}} &= \boldsymbol{K}_{\mathrm{av}}\dot{\boldsymbol{q}}_{\mathrm{v}} + \boldsymbol{K}_{\mathrm{pv}}\boldsymbol{s}_{\mathrm{av}} - \boldsymbol{A}_{\mathrm{v}}^{\mathrm{T}}(\boldsymbol{q}_{\mathrm{v}})(\boldsymbol{\lambda}_{\mathrm{m}} - \boldsymbol{K}_{\mathrm{fv}}\Delta\boldsymbol{\lambda}_{\mathrm{v}}). \qquad (41)\end{aligned}$$

Note that in a natural way a force error will be created when closing the control loop, so that we introduce an integral term of that error in the form

$$\Delta\boldsymbol{\lambda}_{i} = \int_{0}^{t} (\boldsymbol{\lambda}_{i} - \boldsymbol{\lambda}_{\mathrm{d}j})\,\mathrm{d}t. \tag{42}$$

It is important to stress once again that the novelty of our approach is *not* the control design because we employ very well-known techniques, but instead the novelty lies in the effective use of nonholonomic constraints to describe deformable virtual surfaces. Therefore, we do not provide a technical stability proof but a set of reliable experiments, which are shown in the next section.

IV. EXPERIMENTAL RESULTS

This section presents the experimental setup and the results of the proposed approach as well as the performance of the controllers for both holonomic and nonholonomic virtual constraints. The experimental platform consists of a Geomagic Touch haptic robot with six revolute joints. An *ATI Nano-17* six–axis force sensor is adapted at the last link, as shown in Figure 3. A PC executes the control loop with a sample time of $T = 2$ [ms] (the fastest that we could get). 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 3. Experimental platform.

One practical limitation of the Geomagic Touch robot is that only the first three joints are actuated. Still, the operator is able to move the master end–effector with any orientation. Therefore, we have made a trade–off and decided to use only five of the six joints since it is unnecessary for the operator to rotate the last one. Furthermore, a projection of both the force reflection and the controller torques is necessary, but we neglect the contribution of the last two joints. This consideration is not restrictive because our virtual environment considers force but not torque feedback, avoiding the problem of sensor/actuators asymmetry in haptic interfaces [38]. Furthermore, the contribution of the last two joints to the force reflection is much smaller in magnitude when compared with the contribution of the first three joints. On the other hand, the virtual robot does not have this limitation, and therefore it is considered to be fully actuated.

A. TASK DESCRIPTION

We present a detailed description of the interaction process between the virtual tool and the virtual environment, both for a rigid and a deformable sphere. Since as future research we would like to extend these basic simulations to the more complex case of medical training applications, we adopt the avatar shape of a needle, as can be seen in Figure 4. In medicine, the procedures using this tool are very common, being the needle insertion the most studied and simulated [39], for which the operator takes a sterile

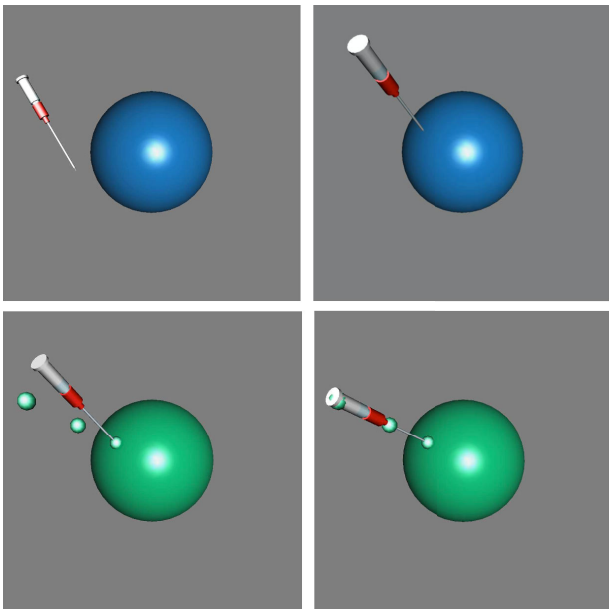

FIGURE 4. Holonomic constraint: Interaction sequence between the avatar and the rigid virtual surface.

needle and slowly approaches it to the patient. Once in contact, he/she needs to know, through a tactile feeling, whether 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 deformable one this is possible. Furthermore, the force behaviors are different since in the first case there is a major contribution in the normal direction, which would allow the operator to move the needle laterally on the surface. However, in the second case the normal force contribution is smaller and the surrounding tissue would not allow moving the needle along the lateral directions. We look to carry out an experiment on the virtual sphere reproducing these behaviors. Therefore, in our approach we assume that the virtual tool is attached at the end–effector of a five degrees of freedom manipulator (not visible in the graphic simulation). This may seem to be unrealistic because evidently a needle does not have such a dynamics. Nevertheless, we use the robot model as a representative example of other medical tools like endoscopes, resectoscopes, forceps, etc., attached to a teleoperated surgical robot arm. For that case one would actually have such a 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 [40].

The task starts with the Geomagic Touch robot in its home position. The operator takes the master robot stylus adapted at the tip to 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 follows it 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 (27) for the rigid sphere or (35) for the deformable one. The operator perceives an interaction force exerted by the master robot and registered by the Nano–17 force sensor until the contact is over.

B. RIGID SPHERE: HOLONOMIC CONSTRAINT

Rigid virtual objects are built from 3D basic geometric primitives as cones, pyramids, planes, cubes or spheres for an implicit representation approach [41]. In turn, complex environments are built from these objects. Ultimately, the basis of a highly complex virtual environment is a set of basic geometric shapes that we can represent with mathematical equations. In this context, the surface used to test the validity of the control algorithm proposed in Section III is a sphere described by

$$\varphi_{\mathrm{v}}(\boldsymbol{x}_{\mathrm{v}}) = (x_{\mathrm{v}} - h)^2 + (y_{\mathrm{v}} - k)^2 + (z_{\mathrm{v}} - l)^2 - r^2 = 0, \quad (43)$$

where $(x_{\mathrm{v}}, y_{\mathrm{v}}, z_{\mathrm{v}})$ stands for the virtual environment task–space coordinates, i.e. $\boldsymbol{x}_{\mathrm{v}} = \begin{bmatrix} x_{\mathrm{v}} & y_{\mathrm{v}} & z_{\mathrm{v}} \end{bmatrix}^{\mathrm{T}}$, $r = 0.1$[m] is the sphere radius, and $(h, k, l) = (0.4, 0, 0)$[m] are the sphere center coordinates. It is important to note that, in contrast with other works, a third dimension coordinate z_{v} was added in order to improve realism of the virtual reality application. For example, [13] and [35] consider only two dimensions to test different control schemes for a haptic and a teleoperation system, respectively. The interaction process described in Section IV-A is summarized in **Algorithm 1** for holonomic constraints.

The gains for the control law (32) 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 $\omega_{\mathrm{n}} = 200$.

The proposed interaction sequence is presented in Figure 4. 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 (33). 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. At the same time, three circles showing the normal direction to the surface at the contact point appear in the simulation to help the operator align the direction of the sensor and thus to obtain a more reliable force measurement. Therefore, the user should try to orient his/her wrist according with these markers, as seen in the last frame of the sequence (bottom right). It is important to note that even when force measurements

Algorithm 1 Virtual Environment Simulation for Holonomic Constraints

Require: Sample Time T
Require: Forward kinematics $\boldsymbol{x}_{\mathrm{v}} = f(\boldsymbol{q}_{\mathrm{v}})$
Require: Vector of parameters $\boldsymbol{\theta}_{\mathrm{m}}$
Require: Holonomic constraint description $\boldsymbol{\varphi}_{\mathrm{v}}(\boldsymbol{x}_{\mathrm{v}})$
Require: Parameters α_{v}, ξ and ω_{n} in (27)
Require: Gain matrices in (20)-(23), (32) and (33)
Initialize $\boldsymbol{q}_{\mathrm{v}} = \boldsymbol{q}_{\mathrm{v0}}$
repeat
  Measure $\boldsymbol{q}_{\mathrm{m}}$ and $\boldsymbol{F}_{\mathrm{m}}$
  Derivate $\boldsymbol{q}_{\mathrm{m}}$ and $\dot{\boldsymbol{q}}_{\mathrm{m}}$
  Evaluate $\boldsymbol{H}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}})$, $\boldsymbol{C}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}}, \dot{\boldsymbol{q}}_{\mathrm{v}})$ and $\boldsymbol{g}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}})$
  Evaluate $\ddot{\boldsymbol{q}}_{\mathrm{v}}$ of (11)
  Integrate $\ddot{\boldsymbol{q}}_{\mathrm{v}}$ and $\dot{\boldsymbol{q}}_{\mathrm{v}}$
  Evaluate $\boldsymbol{Y}_{\mathrm{m}}(\boldsymbol{q}_{\mathrm{m}}, \dot{\boldsymbol{q}}_{\mathrm{m}}, \ddot{\boldsymbol{q}}_{\mathrm{m}})\boldsymbol{\theta}_{\mathrm{m}}$
  Evaluate $\boldsymbol{J}_{\mathrm{m}}(\boldsymbol{q}_{\mathrm{m}})$ and $\boldsymbol{J}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}})$
  Evaluate $\boldsymbol{J}_{\varphi \mathrm{xv}}(\boldsymbol{x}_{\mathrm{v}})$
  Evaluate λ_{v} of (27)
  Evaluate $\boldsymbol{\tau}_{\mathrm{e}} = \boldsymbol{J}_{\varphi \mathrm{xv}}(\boldsymbol{x}_{\mathrm{v}})\boldsymbol{J}_{\mathrm{v}}(\boldsymbol{q}_{\mathrm{v}})\lambda_{\mathrm{v}}$
  Define $\boldsymbol{q}_{\mathrm{d}i}(t)$ and $\dot{\boldsymbol{q}}_{\mathrm{d}i}(t)$, as in (17)–(18)
  Evaluate $\Delta \boldsymbol{q}_i$ as in (19).
  Evaluate $\boldsymbol{s}_i$ as in (20)
  Evaluate $\dot{\boldsymbol{\sigma}}_i$ as in (21)
  Evaluate $\dot{\boldsymbol{q}}_{\mathrm{r}i}$ as in (23)
  Evaluate $\boldsymbol{s}_{\mathrm{a}i}$ as in (24)
  Evaluate $\boldsymbol{F}_{\mathrm{v}} = \boldsymbol{J}^{\mathrm{T}}_{\varphi \mathrm{xv}}(\boldsymbol{x}_{\mathrm{v}})\lambda_{\mathrm{v}}$
  Define $\boldsymbol{F}_{\mathrm{d}i}(t) \triangleq \boldsymbol{F}_j(t)$ as in (29)
  Evaluate $\Delta \boldsymbol{F}_i$ in (30)
  Evaluate $\Delta \boldsymbol{p}_i$ in (31)
  Evaluate $\boldsymbol{\tau}_{\mathrm{v}}$ of (33)
  Evaluate $\boldsymbol{\tau}_{\mathrm{m}}$ of (32)
  Output motion commands to display
until haptic simulation ends

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 5. The operator moves freely the master end–effector until $t = 8$[s]. From this time on until $t = 40$[s], the virtual robot is in contact with the virtual surface, so that also the master motion is constrained and from $t > 40$[s] both manipulators are in free movement again until the experiment finishes.

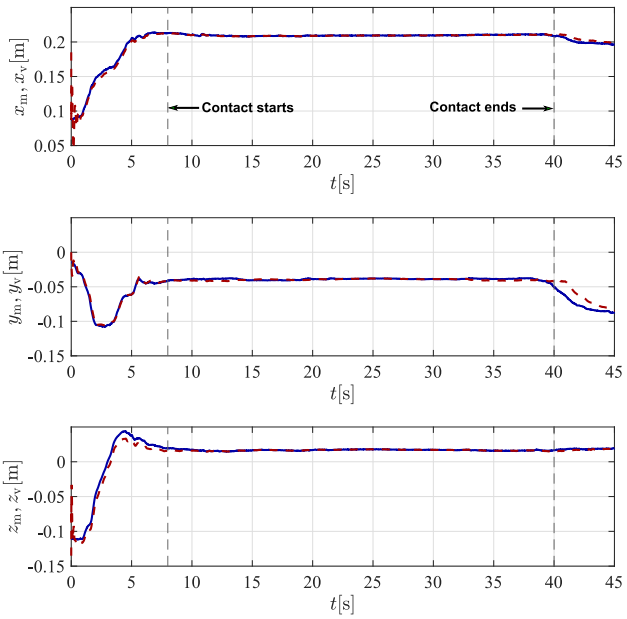

FIGURE 5. Holonomic constraint: Position tracking in Cartesian coordinates: master (–), virtual (– –).

In Figure 6, 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.

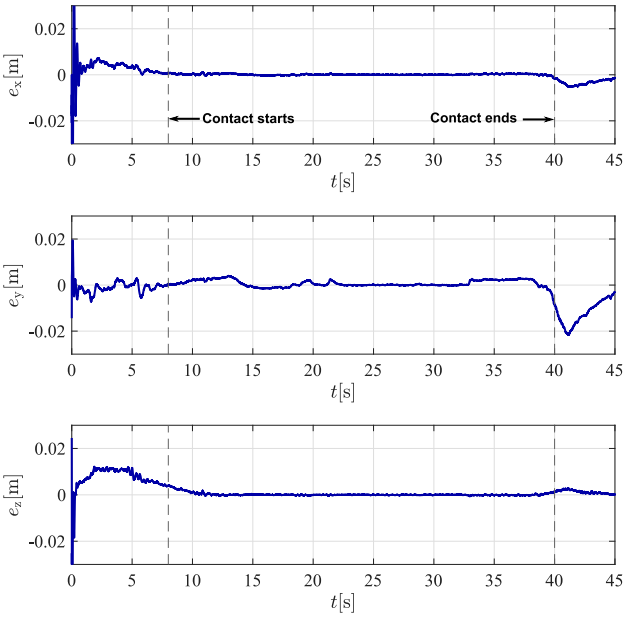

FIGURE 6. Holonomic constraint: Position tracking error.

Force tracking is presented in Figure 7, where the virtual force $\boldsymbol{F}_{\mathrm{v}}$ is zero till the contact begins. On the other hand, the master force $\boldsymbol{F}_{\mathrm{h}}$ is not exactly zero because the operator is holding the corresponding end–effector the whole experiment. At the end of the contact, oscillations take place as a characteristic of the Generic Penalty Method. Nevertheless,

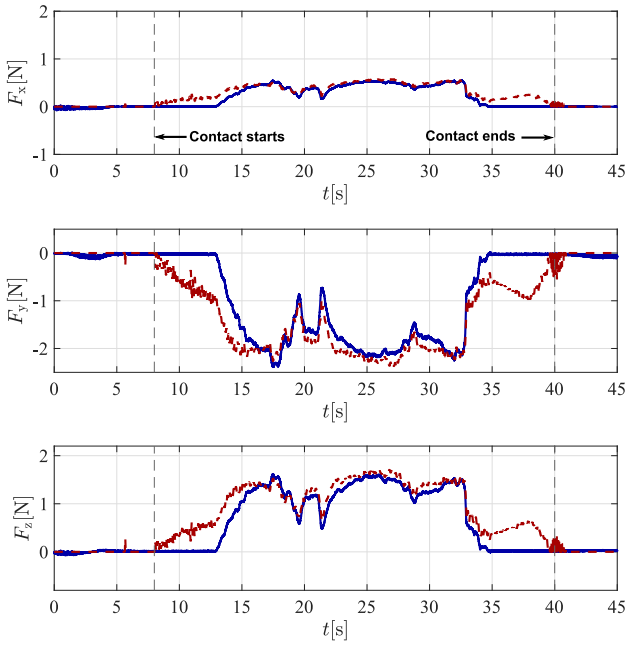

FIGURE 7. Holonomic constraint: Force tracking in Cartesian coordinates: master (–), virtual (– –).

those oscillations are vaguely perceived by the operator and force tracking is preserved. In Figure 8, the corresponding force tracking error is presented.

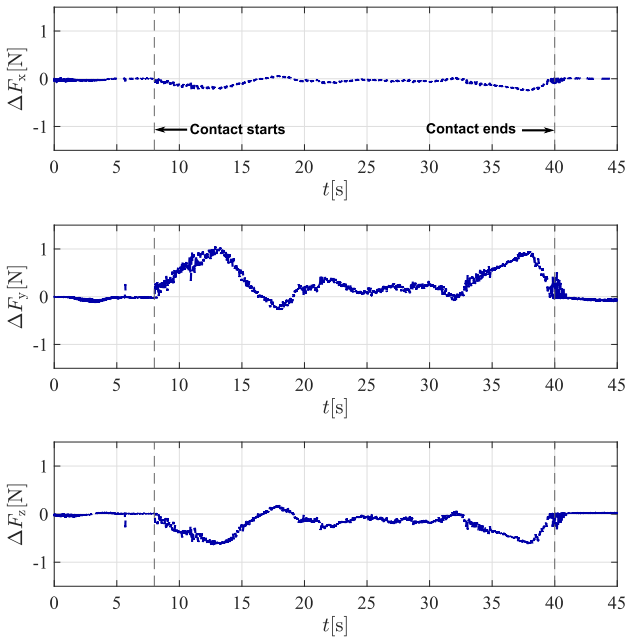

FIGURE 8. Holonomic constraint: Force tracking error.

C. DEFORMABLE SPHERE: NONHOLONOMIC CONSTRAINT

Let us assume that we want the needle to penetrate the sphere, so that it is no longer able to move laterally once inside, just like in a venipuncture scenario. However, we also want the needle to pivot to change its orientation. We claim that this is the kind of scenario that can adequately be described by employing nonholonomic constraints. To the best of the authors' knowledge, nonholonomic constraints have been little exploited to represent the interaction with deformable surfaces. For example, in [26] 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 no analysis or modeling of this process. The most representative proposal is the derivation of the nonholonomic generalized unicycle model presented in [24], where a coordinate-free representation is used to model the insertion of a flexible needle into soft tissue. We use a similar approach by employing 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 we proposed in (36), is an important improvement with respect to previous works.

Let ${}^0\boldsymbol{p}_{\mathrm{v}} \in \mathbb{R}^3$ be the Cartesian position of the virtual robot end–effector and ${}^0\boldsymbol{R}_{\mathrm{v}} \in \mathrm{SO}(3)$ a rotation matrix which describes its orientation. Let us divide this rotation matrix into three column vectors as

$$ {}^0\boldsymbol{R}_{\mathrm{v}} = \left[{}^0\boldsymbol{x}_{nv} \; {}^0\boldsymbol{y}_{nv} \; {}^0\boldsymbol{z}_{nv} \right], \tag{44} $$

for which each column represents the direction of the three axes of the end–effector coordinate frame, but described in the base frame. This allows to define Pfaffian constraints like (12) in an intuitive form, i.e.,

$$ \boldsymbol{A}_{\mathrm{v}}({}^0\boldsymbol{x}_{nv}, {}^0\boldsymbol{y}_{nv}, {}^0\boldsymbol{z}_{nv})\boldsymbol{v}_{\mathrm{v}} = \boldsymbol{0}\,. \tag{45} $$

We claim that one can define a set of nonholonomic constraints if the manipulator degrees of freedom is greater than those necessary to control the end–effector position, i.e., $n > 2$ for planar robots and $n > 3$ for robots in a three dimensional workspace.

For example, consider a virtual planar robot with a tool attached at its end–effector as shown in Figure 9. In this case, the end–effector velocities can be described by $\boldsymbol{v}_{\mathrm{v}} = \left[{}^0\dot{\boldsymbol{p}}_{\mathrm{v}} \; {}^0\omega_n \right]^{\mathrm{T}}$, where ${}^0\dot{\boldsymbol{p}}_{\mathrm{v}} = \left[{}^0\dot{p}_{\mathrm{v}x} \; {}^0\dot{p}_{\mathrm{v}y} \right]^{\mathrm{T}}$ is the linear velocity and ${}^0\omega_n$ is the angular velocity around an axis normal to the robot plane. If the robot is not allowed to move in the ${}^0\boldsymbol{y}_{nv}$ direction, the corresponding Pfaffian constraint is given by

$$ \left[{}^0\boldsymbol{y}_{nv}^{\mathrm{T}} \; 0 \right] \boldsymbol{v}_{\mathrm{v}} = \left[-s_{123} \; c_{123} \; 0 \right] \begin{bmatrix} {}^0\dot{p}_{\mathrm{v}x} \\ {}^0\dot{p}_{\mathrm{v}y} \\ {}^0\omega_n \end{bmatrix} = 0, \tag{46} $$

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 $\boldsymbol{e}_1 = \left[c_{123} \; s_{123} \; 0 \right]^{\mathrm{T}}$ and $\boldsymbol{e}_2 = \left[0 \; 0 \; 1 \right]^{\mathrm{T}}$ as a basis for the null-space of the Pfaffian matrix, one can construct the equivalent control system $\dot{\boldsymbol{q}}_{\mathrm{v}} = \boldsymbol{e}_1 \, u_1 + \boldsymbol{e}_2 \, u_2$, which represents the directions of allowed motion

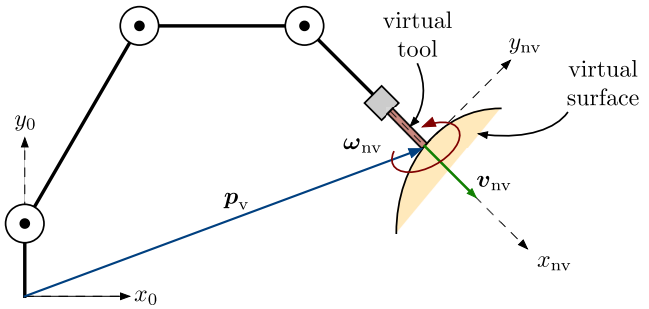

FIGURE 9. Nonholonomic constraint: Virtual robot in interaction with a penetrable surface.

[27, p. 320]. It is easy to verify that the Lie bracket is $[\boldsymbol{e}_1, \boldsymbol{e}_2] = \begin{bmatrix} -s_{123} & c_{123} & 0 \end{bmatrix}^{\rm T}$, which shows the non-involutivity of the distribution and thus establishes the nonholonomic nature of the constraints. 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.

Based on the same logic, our experiment consists on a five degrees of freedom virtual manipulator interacting with the same virtual sphere as for holonomic constraints. Once in contact with it, the end–effector is not allowed to move laterally, in our case along the ${}^0\boldsymbol{y}_{5\rm v}$ and ${}^0\boldsymbol{z}_{5\rm v}$ axes, after a conventional Denavit–Hartenberg allocation, but it is allowed to move along the ${}^0\boldsymbol{x}_{5\rm v}$ axis, i.e., along the pointing direction of the end–effector. Moreover, the end–effector is allowed to rotate to change direction, as a three dimensional version of the nonholonomic unicycle.

The experiment has four steps, as shown in Figure 10. First, the virtual robot is in free motion and only the teleoperation part of control (41) 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. A video of this experiment and the one presented in Section IV-B is available at https://www.youtube.com/watch?v=e3PRBz4QOgQ&feature=youtu.be.

The process described in Section IV-A is summarized in **Algorithm 2**, with the Pfaffian constraints given by

$$\boldsymbol{A}_{\rm v}(\boldsymbol{x}_{\rm v})\boldsymbol{\nu}_{\rm v} = \begin{bmatrix} {}^0\boldsymbol{y}_{5\rm v} & \boldsymbol{0}_{1\times 3} \\ {}^0\boldsymbol{z}_{5\rm v} & \boldsymbol{0}_{1\times 3} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{p}}_{\rm v} \\ \boldsymbol{\omega}_{\rm v} \end{bmatrix} = \boldsymbol{0}. \tag{47}$$

Note that on the contrary to the rigid sphere where the dimension of the Langrange multipliers vector is one, for the deformable sphere it is two. The reason is that in the first case the one single multiplier represents the magnitude of the force perpendicular to the surface, while the two multipliers of the nonholonomic constraint represent the magnitudes of the forces that prevent lateral movements to take place once the needle has penetrated the deformable body.

The human operator forces in lateral directions is difficult to measure directly with the force sensor. Instead, we exploit

Algorithm 2 Virtual Environment Simulation Method for Nonholonomic Constraints

Require: Sample Time T
Require: Forward kinematics $\boldsymbol{x}_{\rm v} = f(\boldsymbol{q}_{\rm m})$
Require: Vector of parameters $\boldsymbol{\theta}_{\rm m}$
Require: Nonholonomic constraint description $\boldsymbol{A}_{\rm v}(\boldsymbol{x}_{\rm v})$
Require: Gain matrices of (20)-(23), (40) and (41)
Initialize $\boldsymbol{q}_{\rm v} = \boldsymbol{q}_{\rm v0}$
repeat
  Measure $\boldsymbol{q}_{\rm m}$ and $\boldsymbol{F}_{\rm h}$
  Derivate $\boldsymbol{q}_{\rm m}$ and $\dot{\boldsymbol{q}}_{\rm m}$
  Evaluate $\boldsymbol{H}_{\rm v}(\boldsymbol{q}_{\rm v})$, $\boldsymbol{C}_{\rm v}(\boldsymbol{q}_{\rm v}, \dot{\boldsymbol{q}}_{\rm v})$ and $\boldsymbol{g}_{\rm v}(\boldsymbol{q}_{\rm v})$
  Evaluate $\ddot{\boldsymbol{q}}_{\rm v}$ of (11)
  Integrate $\ddot{\boldsymbol{q}}_{\rm v}$ and $\dot{\boldsymbol{q}}_{\rm v}$
  Evaluate $\boldsymbol{Y}_{\rm m}(\boldsymbol{q}_{\rm m}, \dot{\boldsymbol{q}}_{\rm m}, \ddot{\boldsymbol{q}}_{\rm m})\boldsymbol{\theta}_{\rm m}$
  Evaluate $\boldsymbol{J}_{\rm m}(\boldsymbol{q}_{\rm m})$ and $\boldsymbol{J}_{\rm v}(\boldsymbol{q}_{\rm v})$
  Evaluate $\boldsymbol{J}_{\varphi \rm xv}(\boldsymbol{x}_{\rm v})$
  Evaluate $\boldsymbol{\lambda}_{\rm v}$ as in (35)
  Evaluate $\boldsymbol{\tau}_{\rm e} = \boldsymbol{A}_{\rm v}^{\rm T}(\boldsymbol{q}_{\rm v})\boldsymbol{\lambda}_{\rm v}$
  Define $\boldsymbol{q}_{{\rm d}i}(t)$ and $\dot{\boldsymbol{q}}_{{\rm d}i}(t)$ as in (17)-(18)
  Evaluate $\Delta\boldsymbol{q}_i$ as in (19)
  Evaluate $\boldsymbol{s}_i$ as in (20)
  Evaluate $\dot{\boldsymbol{\sigma}}_i$ as in (21)
  Evaluate $\dot{\boldsymbol{q}}_{{\rm r}i}$ as in (23)
  Evaluate $\boldsymbol{s}_{ai}$ as in (24)
  Evaluate $\boldsymbol{\lambda}_{\rm m} = \{\boldsymbol{A}_{\rm v}\boldsymbol{A}_{\rm v}^{\rm T}\}^{-1}\boldsymbol{A}_{\rm v}\boldsymbol{J}_{\rm m}^{\rm T}\boldsymbol{F}_{\rm h}$
  Define $\boldsymbol{\lambda}_{{\rm d}i}(t) \triangleq \boldsymbol{\lambda}_j(t)$ as in (38)
  Evaluate $\Delta\boldsymbol{\lambda}_i$ as in (42)
  Evaluate $\boldsymbol{\tau}_{\rm v}$ of (41)
  Evaluate $\boldsymbol{\tau}_{\rm m}$ of (40)
  Output motion commands to display
until haptic simulation ends

the projection of such forces into the master manipulator joint torque, i.e., we compute $\boldsymbol{\lambda}_{\rm m}$ from (39). The gains for the master robot control law (32) are $\boldsymbol{K}_{\rm am} = 0.055\boldsymbol{I}$, $\boldsymbol{K}_{\rm pm} = 0.055\boldsymbol{I}$, $\boldsymbol{K}_{\rm fm} = {\rm diag} = 0.01\boldsymbol{I}$, $\boldsymbol{\Lambda}_{\rm xm} = 0.25\boldsymbol{I}$, $\boldsymbol{K}_{\beta\rm m} = 0.01\boldsymbol{I}$, and $\boldsymbol{K}_{\gamma\rm m} = 0.015\boldsymbol{I}$, whereas for the virtual manipulator $\boldsymbol{K}_{\rm av} = 0.2\boldsymbol{I}$, $\boldsymbol{K}_{\rm pv} = {\rm diag}(0.2, 0.2, 0.2, 0.1, 0.1)$, $\boldsymbol{K}_{\rm fv} = 2\boldsymbol{I}$, $\boldsymbol{\Lambda}_{\rm xv} = 20\boldsymbol{I}$, $\boldsymbol{K}_{\beta\rm v} = \boldsymbol{I}$, and $\boldsymbol{K}_{\gamma\rm v} = 0.2\boldsymbol{I}$ were chosen.

Figure 11 presents a 3D plot of the virtual end–effector position and orientation, showing all four stages of the experiment. Before $t = 4$[s] the virtual manipulator is in free motion, driven only by the master robot by means of the teleoperation scheme. Approximately at $t = 4$[s] the virtual robot gets in contact with the sphere. From this time on until approximately $t = 10$[s], the end–effector, i.e., the

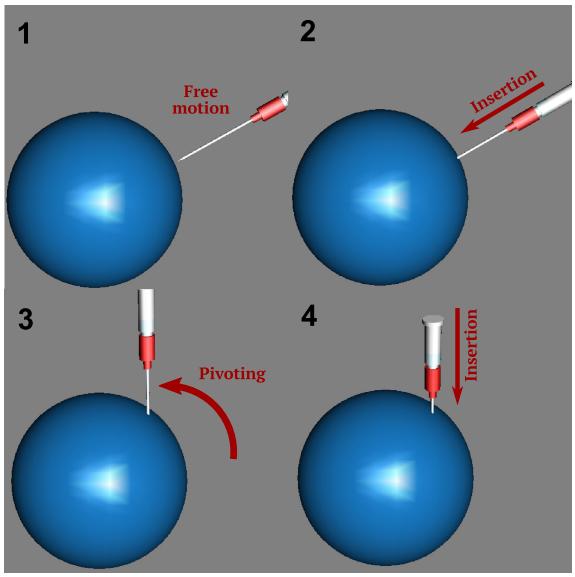

FIGURE 10. Nonholonomic constraint: Interaction sequence between the avatar and the nonholonomic virtual surface.

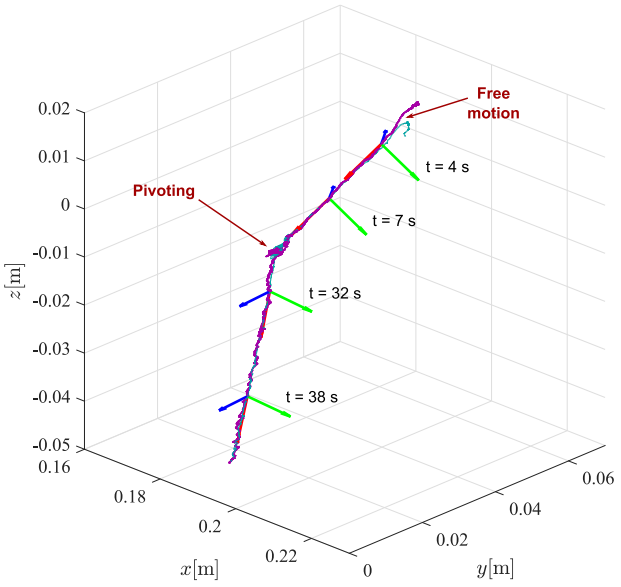

FIGURE 11. Nonholonomic constraint: Cartesian trajectory of the master (—) and virtual (- - -) manipulators, along with the virtual end–effector coordinate frame for different instants of time (red, green and blue arrows represent ${}^0x_{5v}$, ${}^0y_{5v}$ and ${}^0z_{5v}$ axes, respectively).

needle, is *enforced* to follow a straight trajectory driven by the constraint forces. Between approximately $t = 10$[s] and $t = 30$[s] the user changes the needle orientation to a vertical position (pivoting). Then he 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 depicted for some time instants to show that the needle trajectory is always along the x_{5v} axis (red arrow), which is very difficult to satisfy without the aid of nonholonomic constraints. The corresponding time–evolution of the Cartesian position coordinates is displayed in Figure 12,

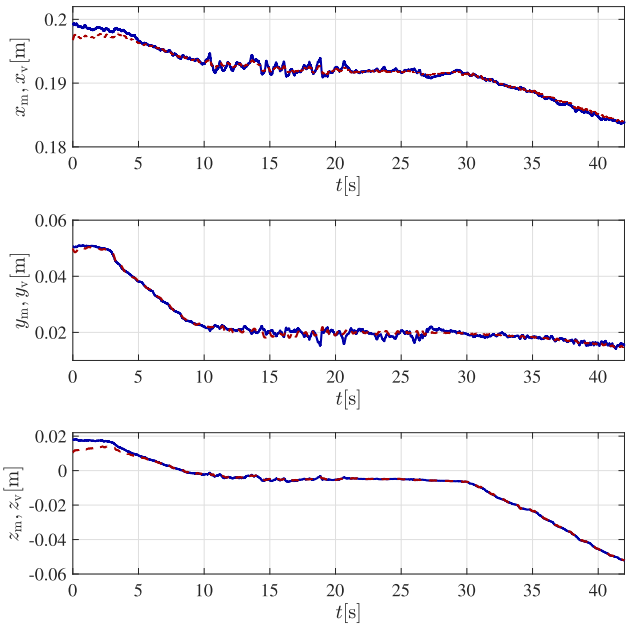

FIGURE 12. Nonholonomic constraint: Cartesian position of the master (—) and the virtual (- - -) manipulators.

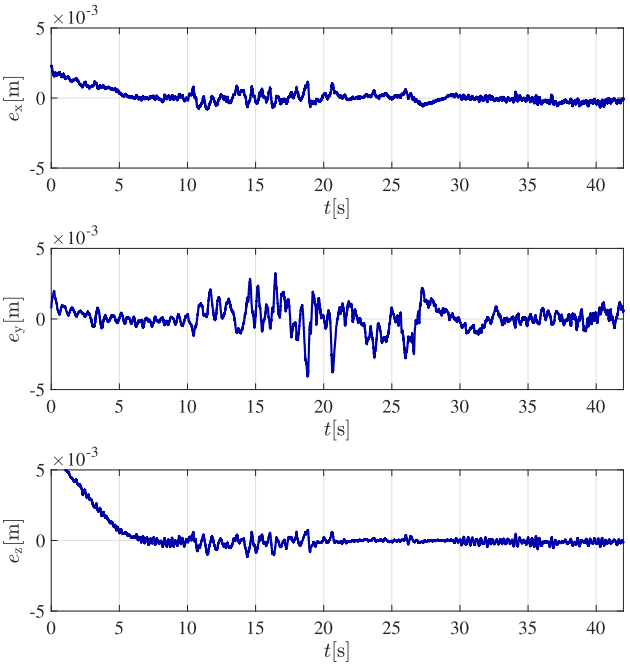

FIGURE 13. Nonholonomic constraint: Cartesian position error.

whereas the tracking error in these coordinates is shown in Figure 13. The virtual constraint forces, represented by the Lagrange multipliers $\boldsymbol{\lambda}_v$ and the forces felt by the human operator, represented by $\boldsymbol{\lambda}_m$ are shown in Figure 14 and the corresponding errors in Figure 15.

D. DISCUSSION

It is convenient to stress once again the differences between using either holonomic or nonholonomic constraints with

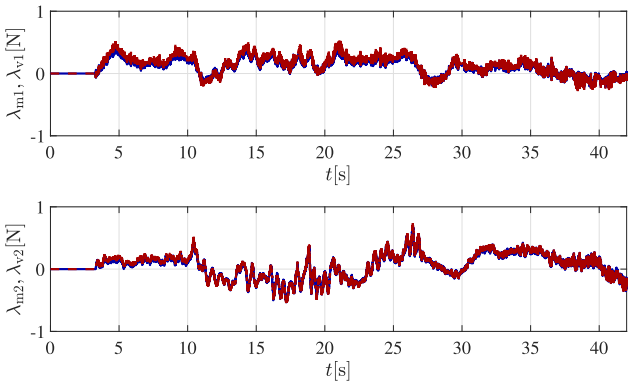

FIGURE 14. Nonholonomic constraint: Cartesian forces at the end–effector of the master (—) and the virtual (- - -) manipulators.

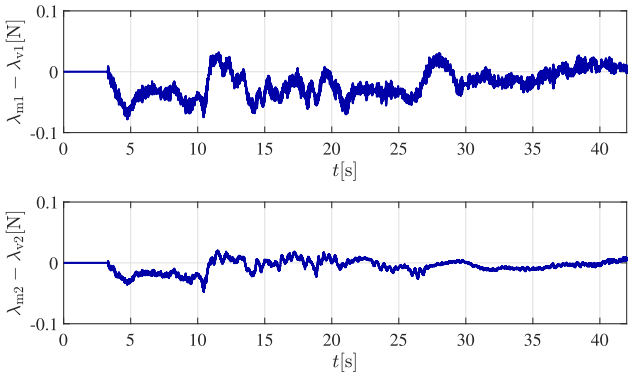

FIGURE 15. Nonholonomic constraint: Cartesian force errors.

respect to the definition of the virtual surface. In the former it is always necessary an implicit expression like (43) to generate the reaction force. Therefore, equation (27) requires $\varphi_v(\boldsymbol{q}_v)$ to evaluate λ_v. In addition, since in graphic computing it is usual to work in Cartesian coordinates [8], it is mandatory to employ the gradient $\boldsymbol{J}_{\varphi xv}(\boldsymbol{q}_v)$ in (28). On the other hand, for nonholonomic constraints both the virtual robot dynamic model and the matrix $\boldsymbol{A}_v(\boldsymbol{q}_v)$ are needed to generate the virtual forces $\boldsymbol{\lambda}_v$ of equation (35). But if the constraint is defined in task–space coordinates, just like we do, it is also necessary to employ the Jacobian $\boldsymbol{J}_v(\boldsymbol{q}_v)$ as in (13) to project them in joint–space coordinates. Nevertheless, an implicit equation of the virtual constraint is not required. This happens to be very relevant for haptic rendering of tasks involving penetration, cutting or indentation for example, because for those cases no implicit equation can be defined and finite element methods are usually employed, which is computationally expensive, so that a trade–off between force and visual rendering must be established [42]. A common choice is to give up force realism for better visualization by using position measurements and spring–damper equations. This problem is solved with our proposal.

A second aspect to highlight is the realism of the application for force rendering, although no torques can be rendered because only the first 3 joints of the master robot are actuated. In the case of holonomic constraints it is evident that by considering a single point of contact the operator must feel the force that prevents him/her to move the virtual tool in the normal direction at that position, where the velocity is zero, as can be appreciated in red in Figure 7. In fact, it is necessary to have the three green markers in the visual rendering in order for the operator to be able to orient the robot end–effector and generate a suitable force measurement to be used in the control algorithm. On the contrary, for nonholonomic constraints the goal is not to reproduce the normal force but the lateral forces that would be present in the penetration of a deformable body, just as for a venipuncture procedure for example. In that scenario, the operator is able to perform trajectories like the one observed in Figure 11. He/she is capable of changing the direction of the virtual tool by making the value of $\boldsymbol{\omega}_v$ different from zero, which is compatible with the Pfaffian constraint matrix $\boldsymbol{A}_v(\boldsymbol{x}_v)$. It is important to note that in a much more complex scenario, obtaining the forces by means of (35) has the advantage of allowing the inclusion of 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 [19], which are generally used for graphic applications where the principal objective is to give a better description of deformable surfaces.

Finally, the performance of the control algorithms is studied. It is not possible to make a direct comparison between holonomic and nonholonomic constraints because the main goals are different in each case, specially for force rendering. For the former the contact force is measured at a single point of the rigid surface, so that we observe a good behavior in Figures 7 and 8, which means that the operator is correctly feeling the sensation of touching a rigid sphere without penetrating it. However, for nonholonomic constraints there is not a single point where the sensor can obtain a reliable measurement because the operator can move the master robot end–effector forward (being the velocity different from zero) but not laterally. A direct consequence of this is that no reliable force measurements are available and its direct use must be given up. Instead, force feedback must come indirectly through the manipulator joint torques by using equations (2), (5), and (39). This avoids to capture the behavior of the force interaction between the human operator and the virtual environment and, in consequence, one has less quality information exchange between the real and the virtual environments. Nevertheless, as can be seen in Figures 14 and 15, the performance of the control algorithm is quite good. The relevant aspect in our approach is to make the operator feel the reaction forces present when executing the trajectory of Figure 11, avoiding the lateral motion of the virtual tool. Such forces are represented in task–space coordinates by the vector $\boldsymbol{\lambda}_m$ and their direction is not normal as in the case of holonomic constraints but lateral, similar to what happens with a needle inserted in soft tissue. On the other hand, for position tracking we show that for both kinds of constraints the corresponding

errors are larger for free than for constrained motion. The reason is that the latter implies a slower motion which makes the controller task much easier because the higher nonlinear effects of robot manipulators come both from Coriolis and centrifugal forces, as well as from the inertia matrix and these terms are practically neglected for low velocities. Note that this is a huge advantage since the free motion case is of little relevance in our study while a larger position error in constrained motion would cause an incorrect visualization of the virtual avatar. Certainly, it also plays an important roll the human operator's skillfulness, because depending on his/her abilities and skills, the master robot end–effector will be manipulated with higher or lower velocities, meaning that for different operators the controllers' performances are expected to be different as well.

V. CONCLUSIONS AND FUTURE WORK

In this work, a proposal on haptic interaction with holonomic and nonholonomic virtual constraints is presented. This is carried out by adapting a teleoperation control scheme to a virtual reality system, allowing to embed a robot dynamic model into a virtual environment. By doing so we address the teleoperated slave system as a problem of a virtual robot in constrained motion described by either holonomic or nonholonomic constraints. We studied the differences between the use of one or the other representation both mathematically and intuitively, and the particularities of each case, e. g., they can be employed to render forces by using the Generic Penalty Method or the Pfaffian constraints matrix, respectively.

Since holonomic constraints for representing rigid surfaces have been widely studied, we centered our interest in nonholonomic constraints and the differences that they have with the former. The principal advantage is that they can be employed to render forces that could potentially approximate those present during manipulations inside deformable surfaces. To the best of the authors' knowledge this is the first time that this kind of constraints are used to reproduce tactile sensations on the operator with a practical significance. Although only a sphere is employed, we showed that the approach can be used in more practical applications, as for instance virtual reality in medical simulators. A complete discussion is presented at the end of the experimental section.

Future research will focus on the combination of holonomic and nonholonomic constraints in a similar application using more sophisticated virtual reality models as in a real surgical simulator. This entails a number of challenges as the development of collision–detection algorithms for nonholonomic constraints or its adaptation to finite element methods for virtual surfaces generation. Our approach should allow to combine such methods with virtual dynamic models and control algorithms to let the operator perceive a large variety of tactile sensations, which is needed to develop haptic interaction in practical real–world applications.

REFERENCES

[1] A. Gallace and C. Spence, *In Touch with the Future: The Sense of Touch from Cognitive Neuroscience to Virtual Reality*. New York, NY, USA: Oxford Univ. Press, 2014.
[2] R. W. Cholewiak, "The perception of tactile distance: Influences of body site, space, and time," *Perception*, vol. 28, no. 7, pp. 851–875, Jul. 1999.
[3] E. Papadopoulos, K. Vlachos, and D. Mitropoulos, "Design of a 5-dof haptic simulator for urological operations," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, May 2002, pp. 2079–2084.
[4] K. Salisbury, F. Conti, and F. Barbagli, "Survey–Haptic rendering: Introductory concepts," *IEEE Comput. Graph. Appl.*, vol. 24, no. 2, pp. 24–32, Mar. 2004.
[5] C. Duriez, F. Dubois, A. Kheddar, and C. Andriot, "Realistic haptic rendering of interacting deformable objects in virtual environments," *IEEE Trans. Vis. Comput. Graphics*, vol. 12, no. 1, pp. 36–47, Jan. 2006.
[6] S. Misra, K. T. Ramesh, and A. M. Okamura, "Modeling of tool-tissue interactions for computer-based surgical simulation: A literature review," *Presence, Teleoperators Virtual Environ.*, vol. 17, no. 5, pp. 463–491, Oct. 2008.
[7] J. Ruurda, T. van Vroonhoven, and I. Broeders, "Robot-assisted surgical systems: A new era in laparoscopic surgery," *Ann. Roy. College Surgeons*, vol. 84, no. 4, pp. 223–226, Jul. 2002.
[8] D. Ruspini and O. Khatib, "Haptic display for human interaction with virtual dynamic environments," *J. Robotic Syst.*, vol. 18, no. 12, pp. 769–783, 2001.
[9] L. G. Garcia-Valdovinos, V. Parra-Vega, and M. A. Arteaga, "Higher-order sliding mode impedance bilateral teleoperation with robust state estimation under constant unknown time delay," in *Proc. IEEE ASME Int. Conf. Adv. Intell. Mechatronics*, Monterey, CA, USA, Jul. 2005, pp. 1293–1298.
[10] Z. Chen, F. Huang, W. Sun, J. Gu, and B. Yao, "RBF-neural-network-based adaptive robust control for nonlinear bilateral teleoperation manipulators with uncertainty and time delay," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 2, pp. 906–918, Apr. 2020.
[11] M. A. Arteaga-Pérez, M. López, E. Nuño, and Ó. Hernández-Ortiz, "On the delayed kinematic correspondence with variable time delays for the control of the bilateral teleoperation of robots," *Int. J. Control*, 2019, doi: 10.1080/00207179.2019.1707287.
[12] M. Mahvash and V. Hayward, "High-fidelity haptic synthesis of contact with deformable bodies," *IEEE Comput. Graph. Appl.*, vol. 24, no. 2, pp. 48–55, Mar. 2004.
[13] E. L. Faulring, K. M. Lynch, J. E. Colgate, and M. A. Peshkin, "Haptic display of constrained dynamic systems via admittance displays," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 101–111, Feb. 2007.
[14] A. Rodríguez, L. Basañez, J. E. Colgate, and E. L. Faulring, "A framework for the simulation and haptic display of dynamic systems subject to holonomic constraints," *Int. J. Robot. Res.*, vol. 29, no. 4, pp. 336–352, Apr. 2010.
[15] A. Rodriguez-Angeles, M. A. Arteaga-Pérez, R. de J. Portillo-Vélez, and C. A. Cruz-Villar, "Transparent bilateral master-slave control based on virtual surfaces: Stability analysis and experimental results," *Int. J. Robot. Autom.*, vol. 30, no. 2, pp. 128–139, 2015.
[16] A. Rodriguez, L. Basaez, J. E. Colgate, and E. L. Faulring, "Haptic display of dynamic systems subject to holonomic constraints," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nice, France, Sep. 2008, pp. 3002–3007.
[17] J. Montagnat, H. Delingette, and N. Ayache, "A review of deformable surfaces: Topology, geometry and deformation," *Image Vis. Comput.*, vol. 19, no. 14, pp. 1023–1040, Dec. 2001.
[18] S. P. DiMaio and S. E. Salcudean, "Needle insertion modeling and simulation," *IEEE Trans. Robot. Autom.*, vol. 19, no. 5, pp. 864–875, Oct. 2003.
[19] J. Nordberg and M. Servin, "Particle-based solid for nonsmooth multidomain dynamics," *Comput. Part. Mech.*, vol. 5, no. 2, pp. 125–139, Apr. 2018.
[20] W. Sun, S. Tang, H. Gao, and J. Zhao, "Two time-scale tracking control of nonholonomic wheeled mobile robots," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 6, pp. 2059–2069, Nov. 2016.
[21] X. Zhang, Y. Fang, B. Li, and J. Wang, "Visual servoing of nonholonomic mobile robots with uncalibrated Camera-to-Robot parameters," *IEEE Trans. Ind. Electron.*, vol. 64, no. 1, pp. 390–400, Jan. 2017.
[22] J. Alonso-Mora, P. Beardsley, and R. Siegwart, "Cooperative collision avoidance for nonholonomic robots," *IEEE Trans. Robot.*, vol. 34, no. 2, pp. 404–420, Apr. 2018.
[23] C. M. Pappalardo and D. Guida, "On the dynamics and control of underactuated nonholonomic mechanical systems and applications to mobile robots," *Arch. Appl. Mech.*, vol. 89, no. 4, pp. 669–698, Apr. 2019.

[24] R. J. Webster, J. S. Kim, N. J. Cowan, G. S. Chirikjian, and A. M. Okamura, "Nonholonomic modeling of needle steering," *Int. J. Robot. Res.*, vol. 25, nos. 5–6, pp. 509–525, May 2006.
[25] D. C. Rucker, J. Das, H. B. Gilbert, P. J. Swaney, M. I. Miga, N. Sarkar, and R. J. Webster, "Sliding mode control of steerable needles," *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1289–1299, Oct. 2013.
[26] E. L. Faulring, K. M. Lynch, J. E. Colgate, and M. A. Peshkin, "Haptic interaction with constrained dynamic systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, Apr. 2005, pp. 2458–2464.
[27] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL, USA: CRC Press, 1994.
[28] J. Gudiño-Lau and M. A. Arteaga, "Dynamic model and simulation of cooperative robots: A case study," *Robotica*, vol. 23, no. 5, pp. 615–624, Sep. 2005.
[29] D. E. Chang and M. Perlmutter, "Feedback integrators for nonholonomic mechanical systems," *J. Nonlinear Sci.*, vol. 29, no. 3, pp. 1165–1204, Jun. 2019.
[30] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Cham, Switzerland: Springer, 2010.
[31] M. A. Arteaga and R. Kelly, "Robot control without velocity measurements: New theory and experimental results," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 297–308, Apr. 2004.
[32] J. M. L. Selig, *Geometric Fundamentals of Robotics*. New York, NY, USA: Springer, 1996.
[33] A. De Luca and G. Oriolo, "Modelling and control of nonholonomic mechanical systems," in *Kinematics and Dynamics of Multi-Body Systems*, vol. 360. Cham, Switzerland: Springer, 1995.
[34] M. A. Arteaga, A. Castillo-Sánchez, and V. Parra-Vega, "Cartesian control of robots without dynamic model and observer design," *Automatica*, vol. 42, no. 3, pp. 473–480, Mar. 2006.
[35] A. Gutiérrez-Giles and M. A. Arteaga-Pérez, "Transparent bilateral teleoperation interacting with unknown remote surfaces with a force/velocity observer design," *Int. J. Control*, vol. 92, no. 4, pp. 840–857, 2017.
[36] J. Swevers, W. Verdonck, and J. D. Schutter, "Dynamic model identification for industrial robots," *IEEE Control Syst.*, vol. 27, no. 5, pp. 58–71, Oct. 2007.
[37] E. Bayo and A. Avello, "Singularity-free augmented lagrangian algorithms for constrained multibody dynamics," *Nonlinear Dyn.*, vol. 5, no. 2, pp. 209–231, 1994.
[38] F. Barbagli and K. Salisbury, "The effect of sensor/actuator asymmetries in haptic interfaces," in *Proc. 11th Symp. Haptic Interfaces Virtual Environ. Teleoperator Syst. (HAPTICS)*, Mar. 2003, pp. 140–147.
[39] C. Yang, Y. Xie, S. Liu, and D. Sun, "Force modeling, identification, and feedback control of robot-assisted needle insertion: A survey of the literature," *Sensors*, vol. 18, no. 2, p. 561, Feb. 2018.
[40] C. Basdogan, S. De, J. Kim, M. Muniyandi, H. Kim, and M. A. Srinivasan, "Haptics in minimally invasive surgical simulation and training," *IEEE Comput. Graph. Appl.*, vol. 24, no. 2, pp. 56–64, Mar. 2004.
[41] D. C. Ruspini, K. Kolarov, and O. Khatib, "The haptic display of complex graphical environments," in *Proc. 24th Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*, Aug. 1997, pp. 140–147.
[42] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, and S. Cotin, "SOFA: A multi-model framework for interactive physical simulation," in *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, vol. 11. Cham, Switzerland: Springer, 2012, pp. 283–321.

JOSÉ DANIEL CASTRO–DÍAZ received the B.Eng. degree in electrical and electronic engineering and the M.Eng. degree in electrical engineering from the National Autonomous University of Mexico (UNAM), in 2011 and 2014, respectively, where he is currently pursuing the Ph.D. degree in electrical engineering. From 2012 to 2017, he worked as an Assistant at the Laboratory for Robotics, Department of Control and Robotics, School of Engineering, UNAM, and from 2015 to 2019, he was a Professor of practical subjects in control and automation. His research interests include biomedical systems, virtual reality simulation, and Research and Development processes.

PABLO SÁNCHEZ–SÁNCHEZ received the B.Eng. degree in electronic engineering from the Technological Institute of Puebla (ITP), in 1999, the M.Sc.Eng. degree in electronics with specialization in automation and control from the Autonomous University of Puebla (BUAP), in 2005, and the Ph.D. degree in electric engineering from the National Autonomous University of Mexico, in 2020. Since 2006, he has been a Full Research Professor with the School of Electronic Sciences, BUAP. His research interests include robotics, control, electronics, programming, and computer aided design.

ALEJANDRO GUTIÉRREZ–GILES received the Ph.D. degree in electrical engineering from the National Autonomous University of Mexico (UNAM), in 2016. He was a Visiting Researcher with the Prisma Lab, University of Naples Federico II, Naples, Italy, from 2016 to 2018. Since August 2018, he has been a Visiting Researcher with the Center of Research and Advanced Studies, National Polytechnic Institute (Cinvestav-IPN), Mexico City, Mexico. He is currently a part of the National Researchers System of Mexico (SNI). His research interests include robotics, nonlinear control, observers, and electromechanical systems.

MARCO A. ARTEAGA–PÉREZ was born in Huejutla de Reyes, Hidalgo, Mexico, in September 1967. He received the B.S. degree in computer engineering and the M.S. degree in electrical engineering from the National Autonomous University of Mexico (UNAM), in 1991 and 1993, respectively, and the Dr.-Ing. degree from Gerhard-Mercator University, Duisburg, Germany, in 1998. Since 1998, he has been a Full Professor with the Department of Electrical Engineering, School of Engineering, National University of Mexico. His main research interest includes robotics and control.

JAVIER PLIEGO–JIMÉNEZ received the Ph.D. degree in electrical engineering with specialization in control from the National Autonomous University of Mexico (UNAM), in 2017. From 2013 to 2017, he was a Professor with the Department of Control and Robotics, School of Engineering, UNAM. He is currently a part of the Applied Engineering Division, Scientific Research and Higher Education Center (CISESE), Department of Electronics and Telecommunications, Ensenada, Mexico.