The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Nonlinear Control and Geometric Constraint Enforcement for Teleoperated Task Execution

Adolfo Rodríguez, Emmanuel Nuño, Leopold Palomo, Luis Basañez

Abstract— **This work presents a multimodal teleoperation framework that makes use of novel tools and techniques, such as: nonlinear *teleoperators control*, for ensuring position tracking in the presence of variable time-delays; *relational positioning*, for increasing operator performance on precise movement execution by visually and haptically displaying geometric constraints; and *augmented reality*, for visually combining real and virtual information in a compelling way. Experimental evidence is presented that validates the aptitude of the different components of the proposed framework.**

I. Introduction

Teleoperated robotic systems are characterized by a robot that executes the movements/actions commanded by an operator. The teleoperated execution of a task is justified because it is often not practical either to perform the task with an autonomous robot, or to perform it with a human operator for reasons as diverse as hazardous environments, physical separation of the execution site, or precision and scale issues [1]. The main objective of such systems is to reproduce—and if possible, enhance—in a remote environment, the sensing and actuation capabilities of an operator [2], so that the mental and physical effort required to accomplish a given task is not negatively affected by its remote execution. However, until recently, performing complex tasks with classical teleoperated systems demanded very skillful operators.

This work presents a multimodal teleoperation framework that takes on these issues by incorporating recent advances in the fields of nonlinear control for variable time-delay systems, relational positioning, and augmented reality. These components can increase the reliability, security, and task performance of a bilateral robotic teleoperation system.

Concerning teleoperators control, one of the most important drawbacks of many existing schemes is that they cannot ensure position tracking between the local and remote manipulators [3], [4]. Moreover, schemes that provide position tracking often rely on strong assumptions such as linear time invariant system models or the absence of time-delays [5], [6]. The controllers presented in this work *do* provide position and velocity tracking, and can handle the nonlinear dynamics of common robot manipulators, as well as variable time-delays in the communication channel.

When an operator performs a remote task, he/she is subject to less sensory throughput, which can lead to reduced performance on tasks requiring precise movements. Relational positioning provides a means of stating these movements in terms of geometric constraints (*e.g.,* distances and angles between points, curves, and surfaces), and combines geometric constraint solving techniques with visual/haptic sensory cues to guide and restrict operator movements to regions of the workspace that are meaningful to the task at hand. Furthermore, since most geometric constraints relevant to teleoperated tasks are defined in the position domain, relational positioning requires teleoperator control schemes that provide position tracking in order to enforce constraint satisfaction on the remote site.

A. System components

A scheme of the teleoperation framework is depicted in Fig. 1. The main physical components are a haptic device on the local site, and a robot manipulator on the remote site. Both are interconnected by a communication channel, which in this work is the Internet. The software and control components are: *teleoperators control*, that ensures asymptotic stabilization of the haptic device and the remote robot despite time-delays; *relational positioning*, that locally generates operator-defined virtual constraints and displays them both visually and haptically; *augmented reality*, that combines real and virtual visual information in a single display; *haptic rendering* that transmits forces/torques to the operator; and *video stream* that sends live video streams from the remote to the local site.

The haptic and visual information that is displayed to the operator corresponds both to data that is fed back from the remote site as well as locally generated aids that act like a feed forward component, as detailed in Fig. 1.

In what follows, Sections II–IV detail the main components of the framework, and Section V presents different validating experiments. Prior art references relevant to each framework component are cited in their respective sections.

II. Teleoperator Control

The haptic device and the remote manipulator are coupled to one another by means of a control algorithm that sends/receives information through the communication channel. Such channel imposes limited data transfer and, depending on its nature, time-delays that can be constant or variable. These delays affect the overall stability of the

The authors are with the Institute of Industrial and Control Engineering (IOC), Technical University of Catalonia (UPC) 08028 Barcelona, Spain. adolfo.rodriguez@ieee.org; {emmanuel.nuno; leopold.palomo; luis.basanez}@upc.edu.

Adolfo Rodríguez is also with PAL Robotics S.L. Barcelona, Spain.

Emmanuel Nuño is also with the Department of Computer Science, University of Guadalajara. Guadalajara 44430, Mexico.

This work has been partially supported by the Spanish CICYT projects DPI2008-02448 and DPI2007-63665, FPI Ref. BES-2006-13393. The second author acknowledges the support of the CONACyT Mexico under the postdoctoral grant 121978.

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

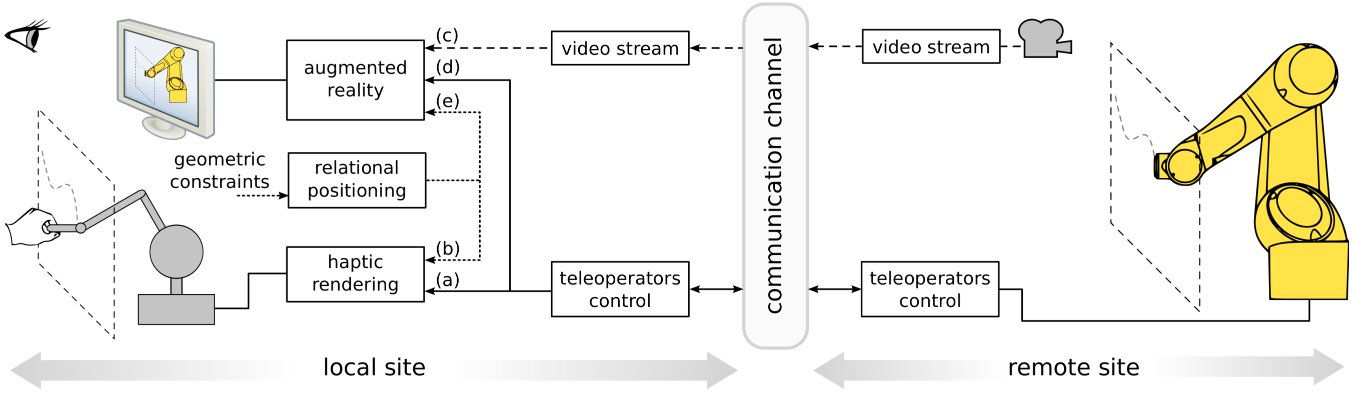


Fig. 1. Scheme of the teleoperation framework. The operator interacts with the system by means of haptic and visual displays, and provides geometric constraints as input to the *relational positioning* component. In what concerns the haptic display, the operator simultaneously feels the forces due to the coupling between the haptic device and the remote robot, *feedback* (Fig. 1a), and the forces that enforce virtual constraints, *feed forward* (Fig. 1b). Additionally, the visual display shows a live video stream of the remote site, *feedback* (Fig. 1c) overlaid with a (optional) virtual reconstruction of the remote robot from teleoperators control data, *feedback* (Fig. 1d), and a visual representation of the virtual constraints, *feed forward* (Fig. 1e).

teleoperation system. Controlling these systems has become a highly active research field (for guides to teleoperators control, the reader may refer to [5], [6]).

The controllers that have been developed in the present framework are: a proportional plus damping (P+d), a proportional-derivative plus damping (PD+d), a scattering-based, and an adaptive controller. Table I depicts their respective mathematical expressions. Detailed descriptions of these controllers, along with proofs for the statements in this paper can be found in [7], [8], [9], [10]. Each controller exhibits different capabilities and, in general, all of them can handle time-delays and can provide asymptotic stability. The main differences between them are stated as follows. The P+d and PD+d can provide stiffer force reflections of the remote environment, however, an increase in time-delays represents an increase in damping, thus overdamped behaviors can happen. For small time-delays the P+d provides better transparency than all the others. The scattering-based controller is more robust to changes in time-delays, but injects more damping than the P+d and the PD+d, and is potentially subject to wave reflections. These three controllers are able to handle variable time-delays, but on the downside, they need to compensate the gravity forces, requiring some previous knowledge of the teleoperators nonlinear model. Finally, the adaptive controller estimates the physical parameters of the manipulators and the rate convergence of errors is faster than with the other schemes, but it cannot handle variable time-delays. An statistical performance comparison, between these schemes, is currently underway. For a similar analysis see [11]. Based on the aforementioned characteristics, the system can employ a different control scheme depending on the physical test-bed in which it is implemented. For example, if delays are small (up to 300 ms) the P+d or PD+d are the best choice, while for larger delays the scattering-based controller is preferred, for constant time-delays and no prior knowledge of the manipulator physical parameters, the adaptive controller can be more suitable.

Let us now review the basics of the control laws. The haptic device and the remote manipulator are modeled as a pair of n-Degrees Of Freedom (DOF) serial chains with revolute joints. These dynamics together with the human and environment interactions are given by

$$\begin{aligned}\mathbf{M}_l(\mathbf{q}_l)\ddot{\mathbf{q}}_l + \mathbf{C}_l(\mathbf{q}_l,\dot{\mathbf{q}}_l)\dot{\mathbf{q}}_l + \mathbf{g}_l(\mathbf{q}_l) &= \boldsymbol{\tau}_h - \boldsymbol{\tau}_l \qquad (1)\\ \mathbf{M}_r(\mathbf{q}_r)\ddot{\mathbf{q}}_r + \mathbf{C}_r(\mathbf{q}_r,\dot{\mathbf{q}}_r)\dot{\mathbf{q}}_r + \mathbf{g}_r(\mathbf{q}_r) &= \boldsymbol{\tau}_r - \boldsymbol{\tau}_e,\end{aligned}$$

where: $\mathbf{q}_i, \dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_i \in \mathbb{R}^n$ are the joint position, velocity and acceleration; $\mathbf{M}_i(\mathbf{q}_i) \in \mathbb{R}^{n\times n}$, the inertia matrices; $\mathbf{C}_i(\mathbf{q}_i,\dot{\mathbf{q}}_i) \in \mathbb{R}^{n\times n}$, the Coriolis and centrifugal effects; $\mathbf{g}_i(\mathbf{q}_i) \in \mathbb{R}^n$, the gravitational forces; $\boldsymbol{\tau}_i \in \mathbb{R}^n$, the controllers; and $\boldsymbol{\tau}_h \in \mathbb{R}^n$, $\boldsymbol{\tau}_e \in \mathbb{R}^n$, the torques at the joints due to the forces exerted by the human and the environment. The subscript i stands for both l and r, local[1] and remote manipulators, respectively.

Let $T_i(t)$ represent the time-delays, and $\mathbf{e}_i \in \mathbb{R}^n$, the position error, defined as

$$\mathbf{e}_l = \mathbf{q}_l - \mathbf{q}_r(t - T_r(t)); \quad \mathbf{e}_r = \mathbf{q}_r - \mathbf{q}_l(t - T_l(t)).$$

Using standard Lyapunov analysis together with Barbălat's Lemma it can be proved that using the P+d or the PD+d controllers, the velocities and the position error are bounded, provided that the control gains, $K_i, B_i > 0$, satisfy

$$4B_lB_r > ({}^*T_l^2 + {}^*T_r^2)K_lK_r, \qquad (2)$$

under the assumptions that the human operator and the environment are passive. If time-delays are variable then, they must have known upper bounds ${}^*T_i$. *i.e.,* $T_i(t) \le {}^*T_i < \infty$; and their time derivatives must not grow or decrease faster than time itself, thus, $|\dot{T}_i(t)| < 1$.

Moreover, if the operator does not inject forces on the haptic device and the remote manipulator does not come in contact with the environment (*i.e.,* $\boldsymbol{\tau}_h = \boldsymbol{\tau}_e = \mathbf{0}$), then velocities and position errors asymptotically converge to zero, *i.e.,*, $|\dot{\mathbf{q}}_i| \to 0$, $|\mathbf{e}_i| \to 0$ as $t \to \infty$. Note that the key feature for the stability of the teleoperator with

[1] In the rest of the section, local manipulator stands for haptic device.

TABLE I

CONTROL LAWS FOR THE LOCAL AND REMOTE MANIPULATORS. $K_i, B_i, K_d, K_{di}, K \in \mathbb{R}^+$ ARE THE CONTROL GAINS AND $(\hat{\cdot})$ MEANS ESTIMATION.

Scheme	Control laws
P+d	$\boldsymbol{\tau}_l = K_l \mathbf{e}_l + B_l \dot{\mathbf{q}}_l - \mathbf{g}_l(\mathbf{q}_l)$ $\boldsymbol{\tau}_r = -K_r \mathbf{e}_r - B_r \dot{\mathbf{q}}_r + \mathbf{g}_r(\mathbf{q}_r)$
PD+d	$\boldsymbol{\tau}_l = K_d \dot{\mathbf{e}}_l + K_l \mathbf{e}_l + B_l \dot{\mathbf{q}}_l - \mathbf{g}_l(\mathbf{q}_l)$ $\boldsymbol{\tau}_r = -K_d \dot{\mathbf{e}}_r - K_r \mathbf{e}_r - B_r \dot{\mathbf{q}}_r + \mathbf{g}_r(\mathbf{q}_r)$
Scatt.-based	$\boldsymbol{\tau}_l = \boldsymbol{\tau}_{ld} + K \mathbf{e}_l + B_l \dot{\mathbf{q}}_l - \mathbf{g}_l(\mathbf{q}_l); \qquad \boldsymbol{\tau}_{ld} = K_{dl}[\dot{\mathbf{q}}_l - \dot{\mathbf{q}}_{ld}]$ $\boldsymbol{\tau}_r = \boldsymbol{\tau}_{rd} - K \mathbf{e}_r - B_r \dot{\mathbf{q}}_r + \mathbf{g}_r(\mathbf{q}_r); \qquad \boldsymbol{\tau}_{rd} = -K_{dr}[\dot{\mathbf{q}}_r - \dot{\mathbf{q}}_{rd}]$
Adaptive	$\boldsymbol{\tau}_l = \hat{\mathbf{M}}_l(\mathbf{q}_l)\boldsymbol{\lambda}\dot{\mathbf{e}}_l + \hat{\mathbf{C}}_l(\mathbf{q}_l, \dot{\mathbf{q}}_l)\boldsymbol{\lambda}\mathbf{e}_l - \hat{\mathbf{g}}_l(\mathbf{q}_l) + K_l(\dot{\mathbf{q}}_l + \boldsymbol{\lambda}\mathbf{e}_l) + B\dot{\mathbf{e}}_l$ $\boldsymbol{\tau}_r = -\hat{\mathbf{M}}_r(\mathbf{q}_r)\boldsymbol{\lambda}\dot{\mathbf{e}}_r - \hat{\mathbf{C}}_r(\mathbf{q}_r, \dot{\mathbf{q}}_r)\boldsymbol{\lambda}\mathbf{e}_r + \hat{\mathbf{g}}_r(\mathbf{q}_r) - K_r(\dot{\mathbf{q}}_r + \boldsymbol{\lambda}\mathbf{e}_r) - B\dot{\mathbf{e}}_r$

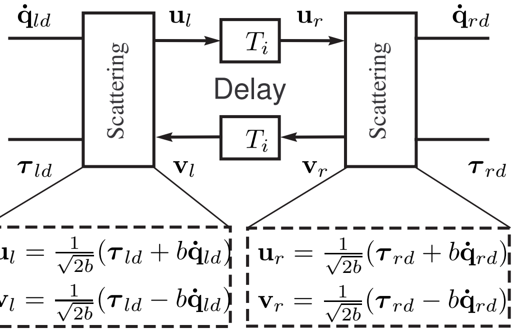

$\dot{\mathbf{q}}_{ld}$
$\mathbf{u}_l$
$\mathbf{u}_r$
$\dot{\mathbf{q}}_{rd}$
Scattering
T_i
Delay
T_i
Scattering
$\boldsymbol{\tau}_{ld}$
$\mathbf{v}_l$
$\mathbf{v}_r$
$\boldsymbol{\tau}_{rd}$
$\mathbf{u}_l = \frac{1}{\sqrt{2b}}(\boldsymbol{\tau}_{ld} + b\dot{\mathbf{q}}_{ld})$
$\mathbf{v}_l = \frac{1}{\sqrt{2b}}(\boldsymbol{\tau}_{ld} - b\dot{\mathbf{q}}_{ld})$
$\mathbf{u}_r = \frac{1}{\sqrt{2b}}(\boldsymbol{\tau}_{rd} + b\dot{\mathbf{q}}_{rd})$
$\mathbf{v}_r = \frac{1}{\sqrt{2b}}(\boldsymbol{\tau}_{rd} - b\dot{\mathbf{q}}_{rd})$

Fig. 2. Scattering transformation. b is the impedance of the virtual transmission line.

these controllers, is condition (2), that clearly states that larger time-delays require injecting more damping in order to maintain stability and position tracking.

Using the scattering-based controller with gains satisfying (2), position tracking can be also established for variable time-delays. In this case, the desired velocities are encoded using the classic scattering transformation proposed in [3], [12] and shown in Fig. 2. For variable time-delays $T_i(t)$, the local and the remote manipulators are interconnected as $\mathbf{u}_r = \gamma_l \mathbf{u}_l(t - T_l(t))$ and $\mathbf{v}_l = \gamma_r \mathbf{v}_r(t - T_r(t))$ where $\gamma_i^2 \leq 1 - \dot{T}_i(t)$.

The adaptive controllers in Table I can be also written as

$$\begin{aligned} \boldsymbol{\tau}_l &= -\mathbf{Y}_l(\mathbf{q}_l, \dot{\mathbf{q}}_l, \mathbf{e}_l, \dot{\mathbf{e}}_l)\hat{\boldsymbol{\theta}}_l + K_l \boldsymbol{\epsilon}_l + B\dot{\mathbf{e}}_l \\ \boldsymbol{\tau}_r &= \mathbf{Y}_r(\mathbf{q}_r, \dot{\mathbf{q}}_r, \mathbf{e}_r, \dot{\mathbf{e}}_r)\hat{\boldsymbol{\theta}}_r - K_r \boldsymbol{\epsilon}_r - B\dot{\mathbf{e}}_r. \end{aligned} \tag{3}$$

Defining a synchronizing signal $\boldsymbol{\epsilon}_i$, as

$$\boldsymbol{\epsilon}_i = \dot{\mathbf{q}}_i + \boldsymbol{\Lambda}\mathbf{e}_i, \tag{4}$$

where $\boldsymbol{\Lambda} > 0$ is diagonal, then with (3), (4) and $\boldsymbol{\tau}_h = \boldsymbol{\tau}_e = \mathbf{0}$, we can write (1) as

$$\mathbf{M}_i(\mathbf{q}_i)\dot{\boldsymbol{\epsilon}}_i + \mathbf{C}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)\boldsymbol{\epsilon}_i + \mathbf{K}_i \boldsymbol{\epsilon}_i + \mathbf{B}\dot{\mathbf{e}}_i = \mathbf{Y}_i \tilde{\boldsymbol{\theta}}_i$$

where $\tilde{\boldsymbol{\theta}}_i = \hat{\boldsymbol{\theta}}_i - \boldsymbol{\theta}_i$ is the error between the estimated and the real parameters. Now, using the estimation law $\dot{\hat{\boldsymbol{\theta}}}_i = -\boldsymbol{\Gamma}_i \mathbf{Y}_i^\top \boldsymbol{\epsilon}_i$, with $\boldsymbol{\Gamma}_i = \boldsymbol{\Gamma}_i^\top > 0$, the adaptive controllers (3) yield $|\boldsymbol{\epsilon}_i| \to 0$ as $t \to \infty$ and $|\dot{\mathbf{q}}_i| \to |\mathbf{e}_i| \to 0$.

III. RELATIONAL POSITIONING

Tasks where an object has to be positioned with respect to its surroundings are ubiquitous in robotics, and oftentimes can be decomposed into a series of constrained movements which do not require using the six degrees of freedom a rigid body has in free space. Although operator skills are needed for the successful execution of a teleoperated task, maintaining the tool or the manipulated object inside a specific region of space can be both challenging and tiring. Such regions can be described in terms of geometric constraints, that when satisfied define a submanifold of SE(3) (the group of rigid motions of $\mathbb{R}^3$) of allowed movements. Haptic feedback can be used to assist the operator by restricting their movements to a submanifold of interest, lowering the mental burden needed to execute the task.

Existing approaches aim at keeping the tool from entering forbidden regions [13], [2], as would be the case of a virtual wall, or guiding it along desired paths [14], [15], [16], [17]. A common disadvantage of most of these approaches is the lack of an intuitive constraint creation interface. The creation of virtual constraints often requires knowledge of the underlying mathematical and software models, hence an experienced user. Furthermore, changing a constraint scenario (*i.e.,* by adding, removing, or modifying constraints) is usually accomplished by manual reprogramming, and thus cannot be done online and interactively.

A. Geometric constraint solving

The relational positioning component explicitly addresses the above issues. Its core consists of a geometric constraint solver called PMF—Positioning Mobile with respect to Fixed—that finds the map between constraint sets and parametrized solution submanifolds [18]. As input, the solver accepts distance and angle relations between points, lines, and planes, and exploits the fact that, for these geometric constraints and their combinations, the rotational component can often be separated from the translational one and solved independently. By means of logic reasoning and constraint rewriting, the solver is able to map a broad family of input constraints to a few rotational and translational scenarios with known closed-form solution. The solver can handle under-, well-, and overconstrained (redundant or incompati-

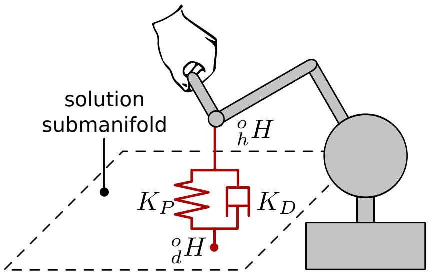


Fig. 3. Haptic display of geometric constraints used in ***Combination 1***.

ble) problems with multiple solutions, and is computationally very efficient, so it can be included in high-frequency loops that require response times within the millisecond order of magnitude. To achieve high computational efficiency, a compromise has been made on the completeness/generality of the solver (while still remaining competent in its application domain).

B. Haptic display of geometric constraints

Haptic feedback quality largely depends on the characteristics of the haptic device and on its control algorithm. Two different and complementary combinations of haptic device and haptic control algorithm have been explored.

Combination 1 consists of an impedance haptic device (a PHANToM Premium 6DOF), which has low inertia/friction and is highly backdrivable, and a control algorithm that leaves the dynamics of the unconstrained directions unchanged and generate forces in the constrained directions. These forces are based on the difference ${}^{h}_{d}H$ between the actual and desired configurations of the end-effector, and are implemented by means of a virtual spring-damper system in task space coordinates (Fig. 3). The desired configuration ${}^{o}_{d}H$ is computed by the PMF solver as the constraint-satisfying configuration that is closest to the current haptic configuration ${}^{o}_{h}H$, and their difference is computed as ${}^{h}_{d}H = {}^{o}_{h}H^{-1}\ {}^{o}_{d}H$. Let $\mathbf{e}_c$ be a $\mathbb{R}^6$ representation of the constraint satisfaction error ${}^{h}_{d}H$, then the wrench $\mathbf{f}_c$ that is applied to the haptic end-effector is $\mathbf{f}_c = K_P \mathbf{e}_c + K_D \dot{\mathbf{e}}_c$.

Notice that $\mathbf{f}_c$ is computed using only local site data, hence the choice of K_P and K_D is not affected by communication channel time-delays, but rather limited by the characteristics of the haptic device and its control loop.

Combination 2 consists of an admittance haptic device (the Cobotic Hand Controller [19]), which has relatively high intertia/friction, but its backdrivability can be continuously varied. In this case, an algorithm based on the simulation of constrained dynamic systems is used. Consider the task space Euler-Lagrange equations of the *simulated* mechanical system

$$\mathbf{M}_s(\mathbf{x})\ddot{\mathbf{x}} + \mathbf{C}_s(\mathbf{x},\dot{\mathbf{x}})\dot{\mathbf{x}} = \boldsymbol{\tau} + A(\mathbf{x})^T\boldsymbol{\psi} \qquad (5)$$

$$A(\mathbf{x})\dot{\mathbf{x}} = \mathbf{0}, \qquad (6)$$

where $\mathbf{x}$ is a $\mathbb{R}^6$ representation of the configuration of the haptic end-effector, $\mathbf{M}_s(\mathbf{x})$ and $\mathbf{C}_s(\mathbf{x},\dot{\mathbf{x}})$ are the dynamics to be simulated, $A(\mathbf{x})^T\boldsymbol{\psi}$ are the forces owing to the imposed geometric constraints, and $\boldsymbol{\tau}$ represents operator forces and virtual springs/dampers (if present). The rows of $A(\mathbf{x})$—also called the Pfaffian constraint matrix—point in the constrained directions. The algorithm solves (5) and (6) for accelerations and integrates numerically twice to obtain the new state of the end-effector. To guarantee constraint satisfaction, the integration is not done in task coordinates, but in a parametrization of the constraint-satisfying submanifold and is then mapped back to task coordinates [17].

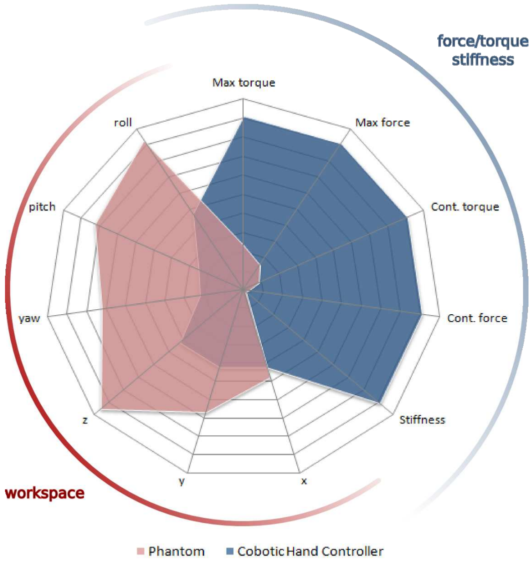


Fig. 4. Qualitative comparison between the PHANToM and Cobotic Hand Controller haptic devices. While the former excels in workspace size, the latter excels in displayable force/torque/stiffness. Source: [19], [20].

In contrast with the previous approach, actuator forces are applied along the unconstrained directions to simulate custom dynamics (*e.g.,* those of the remote robot), while constraint forces are generated by the mechanical structure of the device rather than by its actuators, so high constraint forces can be sustained for long periods of time with little or no power consumption.

The choice between the two approaches depends on application requirements. Haptic device capabilities are complementary with respect to workspace size and displayable force/torque/stiffness, as shown in Fig. 4. Similarly, control algorithms are complementary with respect to implementation complexity, computational cost, and haptic feedback quality. ***Combination 1*** is better suited for applications that can compromise the display of rigid constraints in favor of an ample workspace and a fast and simple haptic display algorithm. Conversely, ***Combination 2*** should be favored in applications that require the display of rigid constraints and/or the simulation of custom dynamics, and can compromise device workspace and computational complexity/cost. The experiments of Section V-A show this complementarity.

IV. Augmented Reality

Augmented reality refers to a real-world environment representation—video streams, in this case—that is enhanced

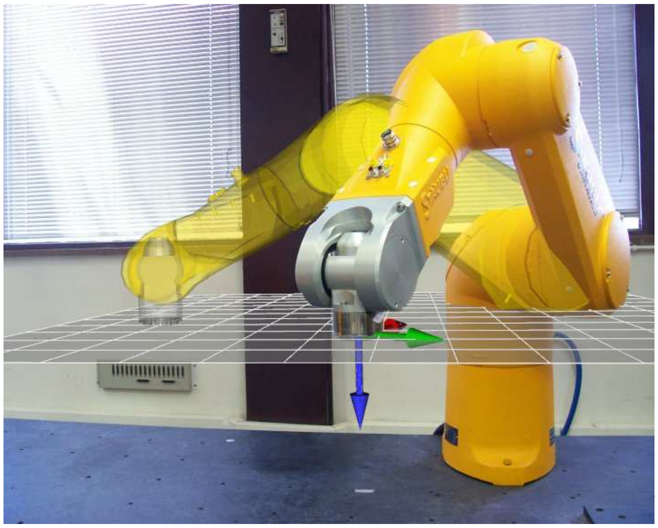


Fig. 5. View of the remote site with augmented reality overlays.

with the addition of computer-generated data. The operator skillfulness can be enhanced at the local site by overlaying a co-located virtual scene on the video stream, akin to predictive/preview displays [21], [22]. This virtual scene contains visual cues and annotations that are not present in the real world, but can improve usability and performance, such as the magnitude and direction of interaction forces, robot workspace boundaries, and regions of allowed (or forbidden) movements for a given task. Moreover, an augmented reality user interface for creating constraint scenarios at interactive rates has been implemented (*c.f.,* accompanying video).

Furthermore, the transmission of live video streams across limited-bandwidth communication channels may yield frame rates below acceptable levels. In such cases, a virtual rendering of the remote manipulator can be displayed and refreshed at a higher frequency while using very little bandwidth, since it only requires updating the current joint positions.

The appearance of an augmented environment with both real and virtual objects must be visually compelling [23] and, for this, must obey overlay and occlusion visibility rules. That is, parts of a virtual object that are in the foreground are rendered and block the real objects that lie behind (if the virtual object is semitransparent, a blending effect occurs), and parts of a virtual object that are behind a real one are not rendered. Occlusions are achieved by rendering transparent models of the real objects in the virtual scene. Fig. 5 depicts a view of the remote site augmented with the currently supported virtual entities: the region of allowed movements associated to the geometric constraints acting on the manipulator (a plane in this case), the end-effector coordinate frame, and a semitransparent rendering of the manipulator reconstructed from remote site data.

V. Experiments and validation

Two different sets of experiments have been carried out. The first set consists of a locally executed (no time-delays) trajectory-following task that highlights the benefits of assisting operator movements with visually and haptically displayed geometric constraints. In the second set, the operator performs teleoperated tasks in which variable time-delays appear, and shows how the selected controller maintains stability and position tracking in different scenarios.

A. Assisted constrained movements

An experiment has been carried out to measure the performance of an operator in a trajectory-following task. The constraint scenario restricts the position of the haptic end-effector to a circular trajectory with $35mm$ radius contained in a horizontal plane. Orientations are left unconstrained. An operator has been instructed to follow the circular constraint as best as possible, and to trace complete loops around the circle.

The two combinations of haptic device and haptic control algorithm introduced in Section III-A have been tested. For ***Combination 1***, information has been displayed to the operator under three sensory modes: *visual only* by displaying a virtual rendering of the circle, *haptic only* by generating forces that attract the haptic end-effector to a constraint satisfying configuration, and *visual+haptic* by simultaneously combining the two previous display modes. For ***Combination 2***, only the *visual+haptic* mode is reported, because performance for the *haptic only* mode is similar.

Operator performance has been measured in terms of the average time required to complete a loop around the circle t_{loop}, the magnitude of the constraint following error $|\mathbf{e}_c|$, and the integral of the absolute error over a single loop $\int_0^{t_{loop}} |\mathbf{e}_c| \, dt$.

Haptic device performance has been measured in terms of the magnitudes of the applied force $|\mathbf{f}_c|$ and displayed stiffness k_P normal to the constraint. Table II summarizes the results for a subject with previous experience using haptic devices, and Fig. 6 shows sample trajectories described by the haptic end-effector for all the tested scenarios.

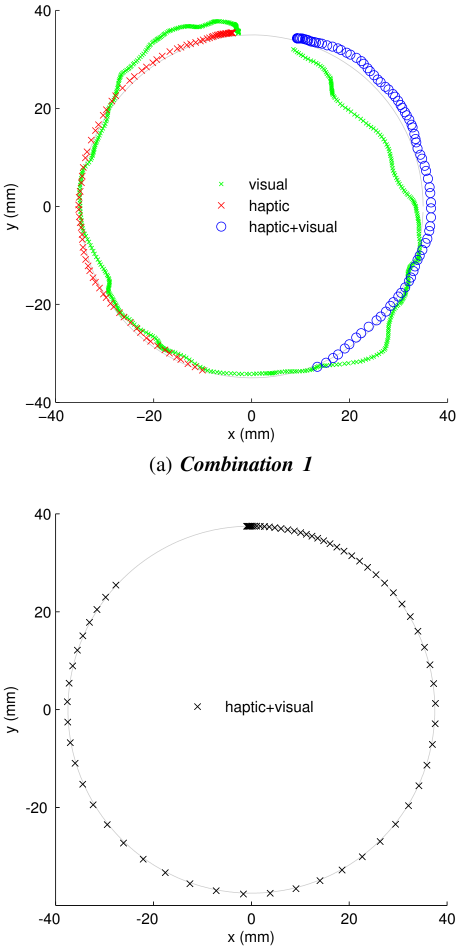
For ***Combination 1***, operator performance for the *visual only* case is the poorest of all three. Furthermore, it has been verified that the position error is highly dependent on the viewpoint of the virtual environment. The *haptic only* case shows a substantial improvement over the *visual only* case, demonstrating that locally generated haptic guidance forces can greatly increase the performance of tasks featuring precise movements. Enabling both *haptic+visual* display of geometric constraints does not affect operator performance significantly with respect to the *haptic only* case. Moreover, Table II reveals that although execution times are slightly lower, there is also a slight increase in the position error. It is conjectured that the real-time visualization of the circle-following, and more importantly, of the path that lies ahead, encourages the operator to move the haptic end-effector at a higher speed, and thus less accurately.

Combination 2 (*haptic+visual*) outperforms ***Combination 1*** in all reported performance parameters. Although both combinations feature sub-millimeter constraint following errors, the fact that the Cobotic Hand Controller is able to exert greater forces and display very high stiffness

TABLE II

PERFORMANCE METRICS FOR THE CIRCLE-FOLLOWING EXPERIMENT

Haptic device +	Constraint	t_{loop} (s)	$\|\mathbf{e}_c\|$ (mm)			$\int \|\mathbf{e}_c\|\, dt$ (mm s)	$\|\mathbf{f}_c\|$ (N)			k_P (kN/m)
haptic control algorithm	display mode	mean	mean	stdev	max	mean	mean	stdev	max	mean
	Visual only	24.0	1.48	1.57	6.20	35.51	-	-	-	-
Combination 1	Haptic only	5.5	0.42	0.36	1.54	2.31	0.17	0.14	0.62	0.4
	Haptic+visual	4.4	0.49	0.45	2.13	2.13	0.20	0.18	0.85	0.4
Combination 2	Haptic+visual	1.6	0.18	0.06	0.34	0.33	8.92	5.45	16.70	65.5



(a) ***Combination 1***

(b) ***Combination 2***

Fig. 6. Sample translational trajectories described by the haptic end-effector for the circle-following task. The two combinations of haptic device and haptic control algorithm introduced in Section III-A are considered.

permits the rejection of greater perturbation forces without compromising accuracy. Consequently, circle tracing times are much lower for this case.

B. *Teleoperated experiments*

The physical components of the framework, in which the teleoperated task has been performed, consist of a TX-90 Stäubli robot as remote manipulator, and a PHANToM 1.5$^{\text{TM}}$ 6DOF haptic device as local manipulator. Additionally, two CANON VC–C50 video cameras that provide a 20 fps video stream have been used.

It should be underscored that the teleoperation framework has been tested from different locations, such as the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, USA; and the Rovira and Virgili University, Tarragona, Spain (*c.f.*, accompanying video). In all cases, the remote site has been located at the IOC Robotics Laboratory of the Technical University of Catalonia, Barcelona, Spain (see [8] for details on the hardware setup). Time delays have been in the order of 0.25 ± 0.05s.

This paper presents two teleoperated experiments with the following common characteristics:

a) The P+d controller has been used in both the haptic device and the robot manipulator.
b) The position commands correspond to the desired position of the haptic end-effector.
c) The communication channel is implemented using UDP/IPv4 sockets in a classical client-server application.
d) Although the 6 DOF of the robot manipulator have been controlled, only data concerning the translational DOFs is presented.

The first experiment deals with the tracking capabilities of one of the framework controllers, when interacting with stiff environments and subject to variable time-delays. The P+d controller has been chosen because the robot physical parameters are known and $^{*}T_i = 0.3$s is small. $K_l = 20$kN/m, $B_l = 5$kN s/m, $K_r = 750$kN/m and $B_r = 200$kN s/m have been used to satisfy (2). The results, that have been originally reported in [8], are presented in Fig. 7 in cartesian space. In this experiment a steel wall has been located in the remote site parallel to the ground at $z = -100$mm, with z pointing upwards. The human operator guides the remote manipulator to touch the wall in two occasions, from $\sim$ 10s to $\sim$ 16s, and from $\sim$ 27s to $\sim$ 31s. During this interaction, large tracking errors occur: along the z-axis, due to the non-penetration constraint imposed by the wall; and along the xy plane (more visible in the x-axis plot) as a consequence of static friction. From the plots in Fig. 7 can be observed that position error asymptotically converges to zero despite stiff contacts.

In the second experimental test, the robot end-effector is constrained to move along a line parallel to the (horizontal) x-axis. Fig. 8 plots the time evolution of positions along the x-, y-, and z-axes. Two regions can be identified in the plots. The first one, from 0s to $\sim$ 10s and from $\sim$ 31s to the end, depicts an unrestricted motion in free space, and the second one, from $\sim$ 10s to $\sim$ 31s shows the positions resulting from a line restriction activated by the operator. It can be seen that

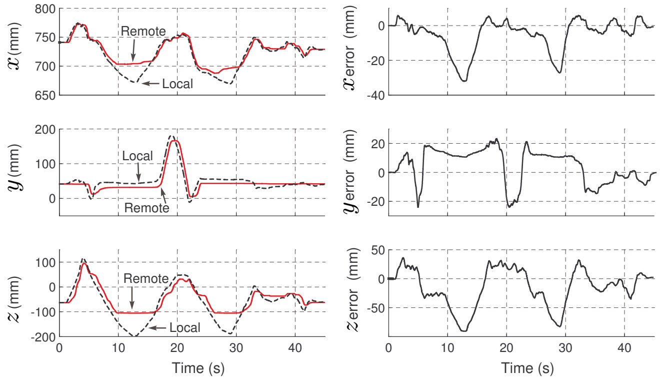


Fig. 7. Cartesian coordinates of the teleoperator interacting with a steel wall located at -100mm in the (vertical) z-axis.

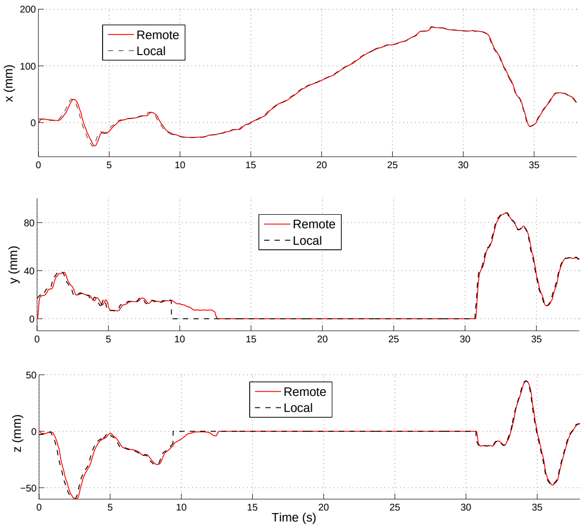


Fig. 8. Cartesian coordinates of the teleoperator while being constrained to a line parallel to the (horizontal) x-axis in the 10s–31s time interval.

position values in the y- and z-axes drop to zero, and motion only takes place along the unconstrained x-axis.

VI. Conclusions

The teleoperation framework presented in this paper summarizes different contributions that improve the overall operator skillfulness while performing a teleoperated robot task. Different nonlinear *teleoperators control* algorithms have been presented that ensure position tracking despite variable time-delays, and criteria for choosing between the different alternatives has been provided. *Relational positioning* has been shown to increase operator performance on tasks requiring precise movements by visually and haptically displaying operator-defined geometric constraints. Two different but complementary haptic display setups have been presented and experimentally compared, showing that for trajectory-following tasks, higher display stiffness values are preferred, and that the haptic feedback plays a stronger role than visual feedback in increasing performance. Finally, *augmented reality* techniques have been used to complement video streams with virtual information such as acting geometric constraints, or a remote robot reconstructed from control data in low-bandwidth scenarios. The aptitude of the different components has been experimentally validated.

References

[1] L. Basañez and R. Suárez, *Teleoperation*. Handbook of Automation. Springer Verlag, 2009, ch. 27, pp. 449–468.

[2] L. Rosenberg, “Virtual fixtures: Perceptual tools for telerobotic manipulation,” in *IEEE Annual Int. Symp. Virtual Reality*, 1993, pp. 76–82.

[3] R. Anderson and M. Spong, “Bilateral control of teleoperators with time delay,” *IEEE Trans. Autom. Control*, vol. 34, no. 5, pp. 494–501, May 1989.

[4] N. Chopra and M. Spong, *Advances in Robot Control From Everyday Physics to Human-Like Movements*. Springer, 2007, ch. Passivity-Based Control of Multi-Agent Systems, pp. 107–134.

[5] P. Hokayem and M. Spong, “Bilateral teleoperation: An historical survey,” *Automatica*, vol. 42, no. 12, pp. 2035–2057, 2006.

[6] E. Nuño and L. Basañez, “Nonlinear bilateral teleoperation: Stability analysis,” in *IEEE Int. Conf. Robot. Automat.*, Kobe, Japan, May 2009, pp. 3718–3723.

[7] E. Nuño, R. Ortega, N. Barabanov, and L. Basañez, “A globally stable PD controller for bilateral teleoperators,” *IEEE Trans. Rob.*, vol. 24, no. 3, pp. 753–758, June 2008.

[8] E. Nuño, L. Basañez, R. Ortega, and M. Spong, “Position tracking for nonlinear teleoperators with variable time-delay,” *Int. Jour. Robotics Research*, vol. 28, no. 7, pp. 895–910, July 2009.

[9] E. Nuño, R. Ortega, and L. Basañez, “An adaptive controller for nonlinear bilateral teleoperators,” *Automatica*, vol. 46, no. 1, pp. 155–159, January 2010.

[10] E. Nuño, L. Basañez, and R. Ortega, “Passivity-based control for bilateral teleoperation: A tutorial,” *Automatica (Provisionally Accepted)*, 2010.

[11] E. Rodriguez-Seda, D. Lee, and M. Spong, “Experimental comparison study of control architectures for bilateral teleoperators,” *IEEE Trans. Rob.*, vol. 25, no. 6, pp. 1304–1318, December 2009.

[12] G. Niemeyer and J. Slotine, “Stable adaptive teleoperation,” *IEEE Jour. Oceanic Eng.*, vol. 16, no. 1, pp. 152–162, Jan. 1991.

[13] J. J. Abbott and A. M. Okamura, “Stable forbidden-region virtual fixtures for bilateral telemanipulation,” *ASME Jour. Dynamic Systems, Measurement, and Control*, vol. 128, no. 1, pp. 53–64, 2006.

[14] ——, “Pseudo-admittance bilateral telemanipulation with guidance virtual fixtures,” *Int. Jour. Robotics Research*, vol. 26, no. 8, pp. 865–884, 2007.

[15] B. P. DeJong, E. L. Faulring, J. E. Colgate, and M. A. Peshkin, “Lessons learned from a novel teleoperation testbed,” *Industrial Robot: An International Journal*, vol. 33, no. 3, pp. 187–193, 2006.

[16] L. Joly and C. Andriot, “Imposing motion constraints to a force reflecting telerobot through real-time simulation of a virtual mechanism,” in *IEEE Int. Conf. Robot. Automat.*, 1995, pp. 357–362.

[17] 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. Jour. Robotics Research. In press*, 2009, doi.10.1177/0278364909104841, http://ijr.sagepub.com/cgi/content/abstract/0278364909104841v1.

[18] A. Rodríguez, L. Basañez, and E. Celaya, “A relational positioning methodology for robot task specification and execution,” *IEEE Trans. Robot.*, vol. 24, no. 3, pp. 600–611, 2008.

[19] E. L. Faulring, J. Colgate, and M. A. Peshkin, “A high performance 6-dof haptic cobot,” in *IEEE Int. Conf. Robot. Automat.*, New Orleans, USA, 2004, pp. 1980–1985.

[20] Sensable Technologies, “Specifications comparison for the PHANTOM premium 1.5 6DOF & 1.5 high force 6DOF haptic devices,” http://sensable.com/documents/documents/Premium_1.5_6DOF.pdf.

[21] A. Bejczy, P. Fiorini, K. Soo, and P. Schenker, “Toward integrated operator interface for advanced teleoperation under time-delay,” in *IEEE Int. Conf. Int. Robot. Sys.*, 1994, pp. 563–570.

[22] N. Chong, T. Kotoku, K. Ohba, H. Sasaki, K. Komoriya, and K. Tanie, “Audio-visual guided predictive simulator in multi-telerobot coordination,” in *IEEE IECON*, 2000, pp. 614–619.

[23] C. Duchesne and J. Herve, “A regularized technique for real-time interposition of 3D objects in augmented reality,” in *IEEE Int. Conf. on Systems, Man and Cybernetics*, 2005, pp. 1402–1408.