A Framework for Teleoperators Control

Emmanuel Nuño, Adolfo Rodríguez, Leopold Palomo, and Luis Basañez

Institute of Industrial and Control Engineering (IOC)
Technical University of Catalonia (UPC)
Barcelona, Spain.
{emmanuel.nuno;adolfo.rodriguez;leopold.palomo;luis.basanez}@upc.edu

(Paper received on February 29, 2008, accepted on April 15, 2008)

Abstract. This work presents a framework for teleoperators control which is composed of a local and a remote station connected through a communication channel. This framework makes use of novel tools and techniques, such as robust controllers and geometric guidance. The controllers render asymptotic zero-convergence of velocities and position error despite variable time-delays. Experimental evidence is presented to validate this framework.

1 Introduction

A teleoperator is commonly referred as the interconnection of five elements: a *human operator* that exerts force on a *local manipulator* connected through a *communication channel* to a *remote manipulator* that interacts with an *environment*. The application of such a system spans multiple fields, the most illustrative being space, underwater, medicine, and, in general, tasks with hazardous environments.

One of the main control objectives in such system is force reflection, which is a means to provide the operator with a sense of immersion on the remote station. However, this objective is compromised by instabilities caused by time-delays. In 1989, Anderson and Spong [1] presented the basis of delay independent teleoperation control. Their approach was to render the communications passive by using scattering theory and the analogy of a lossless transmission line. They showed that the scattering transformation ensures passivity of the communications regardless any constant time-delay. Following the previous approach, Niemeyer and Slotine [2] introduced wave variables (scattering transformation), and proved that by matching the impedance of the local and remote manipulator controllers with the impedance of the *virtual* transmission line, reflections are avoided. Since then, the scattering transformation has dominated the field of teleoperation control.

On the other hand, an ever-growing number of devices connected to the Internet are now accessible to a multitude of users. Being an ubiquitous communication means, the Internet enables users to reach and command any device connected to the network. However, Internet imposes variable time-delays, and the aforementioned schemes cannot guarantee a stable behavior under such conditions. Several works have tackled the issue of providing position tracking under variable time-delays. Chopra et al. [3], with an adaptation of Lozano et al. [4], have proposed a position-drift free scheme in which the remote manipulator has a control term that depends on the position error between the local and remote manipulators. Hirche and Buss [5] extended these results analyzing the effects of packet loss in packet switched communication networks, like the Internet. Munir and Book [6] used a Kalman filter and a position-drift compensator to provide position tracking, and Nuño et. al. [7, 8] have proved that simple PD controllers can also achieve position tracking despite variable time-delays, without the use of the scattering variables. The reader is invited to see [9] and [10] for two interesting survey articles focused on control of teleoperators.

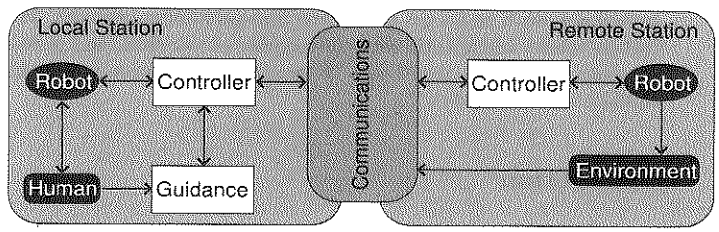

Fig. 1. Framework's overall structure

This work presents a framework for teleoperators control composed of three parts: 1) local station; 2) communications, and; 3) remote station. Fig. 1 shows the overall structure of this framework and, Fig. 2 for its main physical components. The framework copes with variable time-delays, position-drift, operator uncertainty and safety at the remote station. Moreover, it gathers two important tools: geometric guidance and the possibility to use Internet2 and Quality of Service (QoS). The main contribution of this work is the framework as a whole, incorporating various advanced tools, some of them developed by the authors, for rendering bilateral robotic teleoperation secure and reliable, like robust controllers and geometric guidance. Experimental evidence supports the advantages of the presented framework.

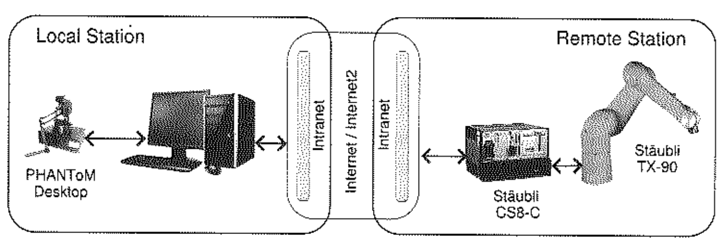

Fig. 2. Physical components of the system

2 Local Station

The human operator is the main component of the local station, it interacts with the local robot manipulator and the Guidance module (Fig. 1). The human selects which controller will be used, sets the geometric restriction –if any– that will be haptically displayed, exerts forces on the local manipulator, and 'feels' what the remote manipulator is touching. The user is also provided with a 3D video stream form the remote station. However, this aspect is not covered in the present article. The following Sections explain how the interactions between the physical system and the software take place.

2.1 Physical System

As can be seen in Fig. 2, the physical elements in the local station are a PC and a robot manipulator, which in this case is a haptic device. Also, communications require an IPv4/IPv6 switch (*cf.* Section 5.2). The haptic device is a PHANToM 1.5™ from Sensable Technologies, connected through a parallel port to the PC. The device provides position and velocity sensing on 6 Degrees of Freedom (DOF), and most importantly, 6 DOF force reflection. The human interacts directly with this device.

2.2 Software Structure

All software is written in C++. The interaction with the haptic device is done using the educational version of Sensable's OpenHaptics™ library. The Graphic User Interface (GUI) has been developed using Trolltech's Qt 4. The left side of Fig. 3 shows the GUI, in which the human operator can: choose different control schemes (*cf.* Section 4.2); enable or disable different geometric restrictions (*cf.* Section 5.1); and modify the controller's gains. The software structure is composed by seven POSIX threads that run in parallel with equal priority, and whose interaction is regulated by semaphores and mutex locks.

The *Guidance* thread is responsible for applying geometric restrictions to the workspace of the haptic device. It does so by generating forces locally on the device that restrict operator movements to a submanifold of free space that is meaningful for the task at hand. The *Controller* thread contains the implementation of different control laws (Table 1), and the operator can select one of them through the GUI. The *Haptic Rendering* thread maps the incoming forces and torques from the *Guidance* and *Controller* threads into the haptic device frame.

The *Haptic ServoLoop* thread directly interacts with the PHANToM device at a frequency of 1 kHz. The *Guidance* and *Controller* threads receive position and velocity signals from the *Haptic ServoLoop* and send force and torque signals to the *Haptic Rendering* thread. While the *Server* thread receives all incoming data from the remote station, the *Client* thread returns position and velocity signals. Both threads use UDP sockets with either IPv4 or IPv6. The right side of Fig. 3 depicts the threads and the flow of information between them.

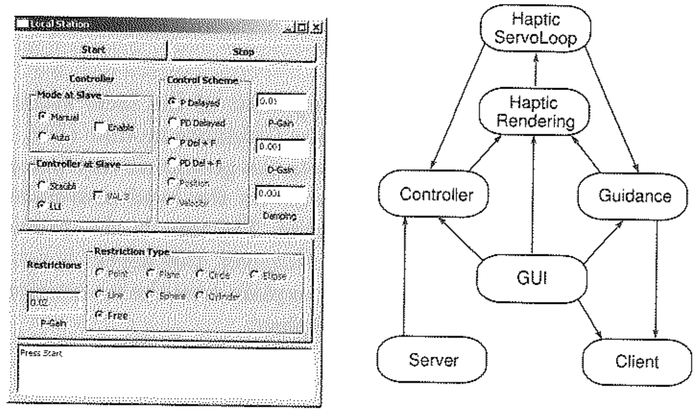

Fig. 3. Graphic User Interface (GUI) and program threads at local station

3 Remote Station

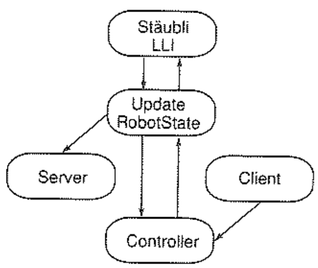

Fig. 4. Program threads on the remote station

The physical system on the remote station consists of a TX-90™ Stäubli robot and a CS8-C™ Stäubli controller, four cameras with actuated pan, tilt and zoom, and a high speed switch connected to Internet and Internet2. Fig. 2 shows these physical components. The software program runs under Wind River's VxWorks™, a Real-Time Operating System (RTOS). The programs are compiled with the cross platform Tornado™ environment and are written in C++. Fig. 4 depicts the threads running on the RTOS.

The *Stäubli LLI* is a thread that runs the Low Level Interface library provided by Stäubli, and runs at 250 Hz. In each interruption it updates all state variables, that is, position, velocity and motor torques. The *Update RobotState* thread contains a zero order hold for the state variables, and all the information needed for the *Controller*, *Client* and *Server* threads. The *Client* and *Server* threads are UDP sockets. Semaphores and Mutex locks are used to synchronize the program threads. As in the local station, the *Controller* thread can use different control laws that provide the torques τ_r to be applied to the joints of the remote manipulator (Table 1). The clocks of both sites are synchronized each time the program starts using a Network Time Protocol (NTP) Server.

The LLI library provided by Stäubli allows direct access to motor torques on the remote manipulator, but unfortunately it is a black–box, thus limiting manipulation capabilities on the *Controller* thread. Moreover, VxWorks™ , although suitable for real-time control, makes difficult to implement new software. One approach to overcome these difficulties consists on wrapping the LLI and the VxWorks^{TM} on a higher-level layer and connect it to an external PC with an open operating system through a deterministic dedicated communication channel. Thus, *Controller, Client, Server* and other possible threads could be implemented with less difficulties. This approach can open new horizons for the presented framework, ranging from cooperative control to multi-arm teleoperation.

4 Controlling the Teleoperator

The statements about the control schemes in this Section are presented without proof. The interested reader can find them in [7, 8].

In the following, $\mathbb{R}$ stands for the real number set, $\mathbb{R}^+$ for the positive real number set, $\mathbb{R}_0^+$ for the set containing $\mathbb{R}^+$ and zero, $|\cdot|$ for the Euclidean norm and $\|\cdot\|_\infty$ for the $\mathcal{L}_\infty$ norm.

Table 1. Control Laws for the local and remote robot manipulators. Where $\{K_i, B_i, K_d, K_{di}, K\} \in \mathbb{R}^+$ are the control gains

Type	Control Laws
P	$\tau_l = K_l[\mathbf{q}_r(t - T_r(t)) - \mathbf{q}_l] - B_l\dot{\mathbf{q}}_l$ $\tau_r = K_r[\mathbf{q}_r - \mathbf{q}_l(t - T_l(t))] + B_r\dot{\mathbf{q}}_r$
PD	$\tau_l = K_d[\gamma_r\dot{\mathbf{q}}_r(t - T_r(t)) - \dot{\mathbf{q}}_l] + K_l[\mathbf{q}_r(t - T_r(t)) - \mathbf{q}_l] - B_l\dot{\mathbf{q}}_l$ $\tau_r = K_d[\dot{\mathbf{q}}_r - \gamma_l\dot{\mathbf{q}}_l(t - T_l(t))] + K_r[\mathbf{q}_r - \mathbf{q}_l(t - T_l(t))] + B_r\dot{\mathbf{q}}_r$
Scatt.-based	$\tau_l = \tau_{ld} + K[\mathbf{q}_r(t - T_r(t)) - \mathbf{q}_l] - B_l\dot{\mathbf{q}}_l; \quad \tau_{ld} = -K_{dl}[\dot{\mathbf{q}}_l - \dot{\mathbf{q}}_{ld}]$ $\tau_r = \tau_{rd} + K[\mathbf{q}_r - \mathbf{q}_l(t - T_l(t))] + B_r\dot{\mathbf{q}}_r; \quad \tau_{rd} = K_{dr}[\dot{\mathbf{q}}_r - \dot{\mathbf{q}}_{rd}]$

4.1 Mathematical Model

The local and remote manipulators together with the human and environment interactions, that conform the teleoperator, are modeled as a pair of n-DOF serial links with *revolute* joints. Their corresponding nonlinear dynamics are

$$\begin{aligned} \mathbf{M}_l(\mathbf{q}_l)\ddot{\mathbf{q}}_l + \mathbf{C}_l(\mathbf{q}_l, \dot{\mathbf{q}}_l)\dot{\mathbf{q}}_l + \mathbf{g}_l(\mathbf{q}_l) &= \tau_l^* - \tau_h \\ \mathbf{M}_r(\mathbf{q}_r)\ddot{\mathbf{q}}_r + \mathbf{C}_r(\mathbf{q}_r, \dot{\mathbf{q}}_r)\dot{\mathbf{q}}_r + \mathbf{g}_r(\mathbf{q}_r) &= \tau_e - \tau_r^*, \end{aligned} \tag{1}$$

where: $\ddot{\mathbf{q}}_i, \dot{\mathbf{q}}_i, \mathbf{q}_i \in \mathbb{R}^n$ are the joint acceleration, velocity and position; $\mathbf{M}_i(\mathbf{q}_i) \in \mathbb{R}^{n\times n}$ 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 forces exerted by the human and the environment. The subscript i stands for both l and r, local and remote manipulators, respectively.

4.2 Control Schemes

There are three control schemes that can be used in this framework: P, PD and Scattering–based, which are summarized in Table 1. These schemes have been developed in the context of this framework.

Using standard Lyapunov arguments it can be shown that using P or PD controllers with the teleoperator dynamics (1), the closed-loop positions and velocities are bounded. *i.e.*, $\{\dot{\mathbf{q}}_i, \mathbf{q}_l - \mathbf{q}_r\} \in \mathcal{L}_\infty$, if the gains are set according to

$$4B_lB_r > [{}^*T_l^2 + {}^*T_r^2]K_lK_r, \tag{2}$$

under the assumptions that

a) the human operator and the environment define passive maps. *i.e.*, $\exists\, \kappa_i \in \mathbb{R}_0^+$ s.t. $\forall t \geq 0$,
$$\int_0^t \dot{\mathbf{q}}_l^\top \boldsymbol{\tau}_h d\sigma \geq -\kappa_l, \qquad -\int_0^t \dot{\mathbf{q}}_r^\top \boldsymbol{\tau}_e d\sigma \geq -\kappa_r; \tag{3}$$
b) the gravitational forces are pre-compensated by the controllers $\boldsymbol{\tau}_i^*$. That is $\boldsymbol{\tau}_l^* = \boldsymbol{\tau}_l + \mathbf{g}_l(\mathbf{q}_l)$ and $\boldsymbol{\tau}_r^* = \boldsymbol{\tau}_r - \mathbf{g}_r(\mathbf{q}_r)$;
c) the variable time-delay has a known upper bound ${}^*T_i$. *i.e.* $T_i(t) \leq {}^*T_i < \infty$, and its time derivative does not grow or decrease faster than time itself, thus, $|\dot{T}_i(t)| \leq 1$.

Moreover, if the human does not move the local manipulator and the remote manipulator does not touch anything (*i.e.* $\boldsymbol{\tau}_h = \boldsymbol{\tau}_e = \mathbf{0}$), then the teleoperator's velocities asymptotically converge to zero and position tracking is achieved:

$$|\dot{\mathbf{q}}_i| \to 0 \qquad |\mathbf{q}_l - \mathbf{q}_r(t - T_r(t))| \to 0 \qquad t \to \infty.$$

If the P or PD controller is replaced by the scattering–based one, with gains satisfying (2), then boundedness of position error and velocities together with position tracking can be also established for variable time-delays. For this case the desired velocities are encoded using the classic scattering transformation given by

$$\begin{array}{ll} \mathbf{u}_l = \frac{1}{\sqrt{2b}}[\boldsymbol{\tau}_{ld} - b\dot{\mathbf{q}}_{ld}] & \mathbf{u}_r = \frac{1}{\sqrt{2b}}[\boldsymbol{\tau}_{rd} - b\dot{\mathbf{q}}_{rd}] \\ \mathbf{v}_l = \frac{1}{\sqrt{2b}}[\boldsymbol{\tau}_{ld} + b\dot{\mathbf{q}}_{ld}] & \mathbf{v}_r = \frac{1}{\sqrt{2b}}[\boldsymbol{\tau}_{rd} + b\dot{\mathbf{q}}_{rd}] \end{array} \tag{4}$$

where b is the *virtual* impedance of the communications, $\gamma_i^2 = 1 - \dot{T}_i(t)$ and $\mathbf{u}_r = \gamma_l \mathbf{u}_l(t - T_l(t))$, $\mathbf{v}_l = \gamma_r \mathbf{v}_r(t - T_r(t))$.

Note that the key feature for the stability of the teleoperator is condition (2), it clearly states that if time-delay increases then damping injection has to be increased in order to maintain stability and position tracking. This condition may seem somewhat restrictive, however, on-earth time-delays are in the order of magnitude of hundreds of milliseconds, thus over-damped behaviors can be avoided.

5 Additional Features

Besides the use of robust controllers, the framework enhances its capabilities with the employment of geometric guidance and the QoS of Internet2.

5.1 Geometric Guidance

Teleoperated tasks can often be decomposed into a sequence of simple movements that do not require using the six DOF an object has in free space. For example, the insertion of a cylindrical peg in a hole only requires two DOFs, translations and rotations around the hole's axis, provided that the axis of the two objects are aligned. Haptic feedback can be used to assist the operator by restricting his/her movements to a region of interest, lowering the mental burden needed to execute the task [11, 12]. Geometric restrictions can be explicitly created by means of the user interface shown in Fig. 3, or with the aid of a geometric constraint solver like PMF [13, 14], in order to define a submanifold of allowed movements.

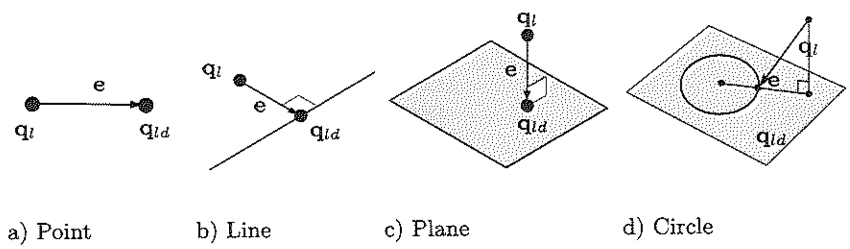

Fig. 5. Geometric restrictions

Guidance forces are generated in the constrained directions based on the difference ($\mathbf{e}$) between the actual ($\mathbf{q}_l$) and desired ($\mathbf{q}_{ld}$) positions of the end-effector, where $\mathbf{q}_{ld}$ is computed as the projection of $\mathbf{q}_l$ on the restriction submanifold. Currently, the position of the haptic end-effector can be translationally restricted to points, lines, planes, spheres, cylinders, and ellipses. Fig. 5 shows four examples of geometric restrictions.

Depending on the task, two different reference motion commands for the remote manipulator can be used. One of these is given by $\mathbf{q}_{ld}$ and then the remote manipulator movements are strictly along the restriction submanifold. The other option is to use $\mathbf{q}_l$ as commands, in this case, the remote manipulator reproduces the operator's movements, which are locally constrained by the haptic device to the restriction submanifold. Both operation modes are possible and switching between them can be done online.

5.2 Internet and Internet2 Communications

The UDP sockets on the local and remote stations are implemented using either the version 4 or 6 of the Internet Protocol. The 'classic' Internet runs over IPv4, while the Internet2 uses both IPv4 and IPv6. Amongst the several differences that exist between the two, the most relevant from a teleoperation point of view are:

- IPv4 has 2^{32} assignable addresses while IPv6 has 2^{128} [15, 16].
- IPv6 incorporates the Quality of Service (QoS) paradigm, in which packets can be sent using different priorities. Its predecessor, IPv4 did not have it, and in its place other protocols, namely RSVP, had to be used [17].

Internet is far from being a deterministic communication channel. However, by using the QoS paradigm, priorities can be imposed on control packages, thus providing more reliable communications by lowering the probability of instability caused by highly time–varying communications.

6 Experimental Validation

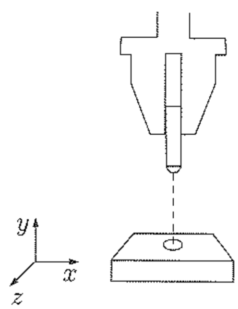

Fig. 6. Peg-in-hole scheme

In order to provide experimental evidence of the presented framework, several teleoperated tasks have been designed. One of these is a peg-in-hole task that has been remotely performed using the proposed architecture. The peg-in-hole insertion has the following characteristics:

- Motion of the peg is restricted to the hole's centerline, providing a natural guide towards the task goal.
- The input to the remote manipulator controller are positions of the human operator, namely $\mathbf{q}_l$.
- Packets have been transmitted using UDP/IPv6 sockets.

For the experiments, the P controller was used with control gains set such that (2) holds. The gains are: for the local manipulator, $K_l = 20$ and $B_l = 5$, and for the remote manipulator $K_r = 750$ and $B_r = 200$. The experiments have been performed using only 3 DOF of each manipulator. Fig. 7 plots the time evolution of positions along the x, y, and z directions of the remote manipulator end-effector. The force plots show the components of the two forces acting on the haptic device: the restriction force $\mathbf{f}_r$, and the force provided by the controller $\mathbf{f}_l$. The insertion direction is along the y axis. From 0s to 1.5s no restriction has been set and the manipulator moves freely in space, thus $\mathbf{f}_r = \mathbf{0}$. Once the restriction is set, at 1.5s, the corresponding restriction forces appear on the x and z directions. No restriction force is exerted in the y axis since it corresponds to the unconstrained direction. The position plot in the y direction shows the approach and insertion processes, that take place until the 9s time mark, and from that moment on, the peg is removed from the hole.

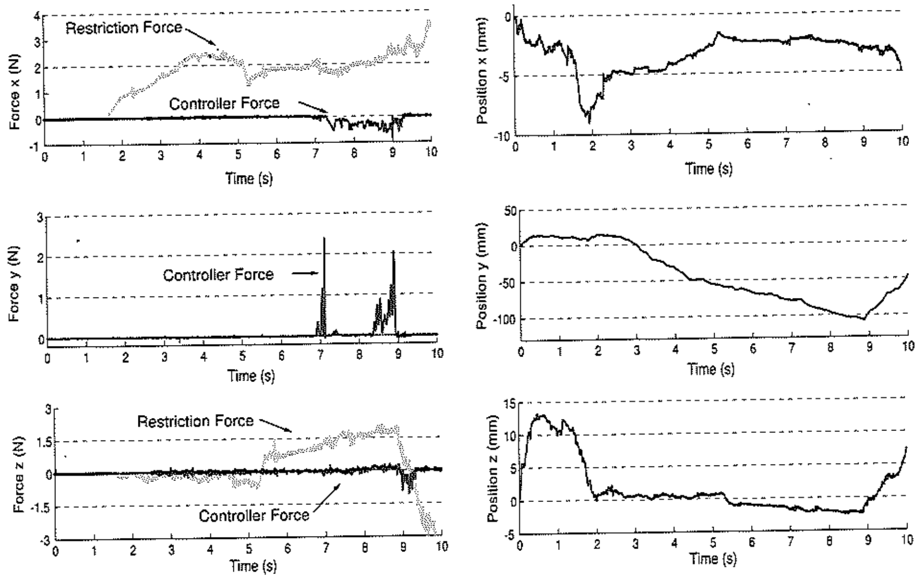

Fig. 7. Force and position in the x, y, z directions

7 Conclusions

This work has presented a framework for teleoperators control. The human operator can decide which controller to use in the local and remote manipulators, he can easily set and remove geometric motion restrictions such as points, lines, planes, spheres, cylinders, and ellipses. The framework makes use of robust controllers that provide position tracking despite variable time-delays. Thus, they allow to use the Internet as communication channel. The framework also gives the option to use the Internet2, with the IPv6 protocol, hence providing QoS over network traffic. Real experiments have demonstrated the effectiveness of the proposed framework. Future research include the study of an active human interaction.

Acknowledgement. The authors would like to thank the members of the robotics division of the IOC for their valuable help in this work. This work has been partially supported by the Spanish CICYT projects: DPI2005-00112 and DPI2007-63665, the FPI program with reference BES-2006-13393, and also by the Mexican CONACyT grant-169003.

References

1. Anderson, R.J., Spong, M.W.: Bilateral control of teleoperators with time delay. IEEE Transactions on Automatic Control, 34(5):494–501, May (1989)

2. Niemeyer, G., Slotine, J.J. E.: Stable adaptive teleoperation. IEEE Journal of Oceanic Engineering, 16(1):152–162, Jan. (1991)
3. Chopra, N., Spong, M.W., Hirche, S., Buss, M.: Bilateral teleoperation over the internet: the time varying delay problem. In: Proc. American Control Conference, 1(4):155 – 160, June (2003)
4. Lozano, R., Chopra, N., Spong, M.W.: Passivation of force reflecting bilateral teleoperators with time varying delay. In: Proc. Mechatronics, June (2002)
5. Hirche, S., Buss, M.: Packet loss effects in passive telepresence systems. In: Proc. IEEE Conference on Decision and Control, pp.4010–4015, Dec. (2004)
6. Munir, S., Book, W.J.: Control techniques and programming issues for time delayed internet based teleoperation. Journal of Dynamic Systems, Measurement and Control, 125(2):205–214, June (2004)
7. Nuño, E., Ortega, R., Barabanov, N., Basañez, L.: A globally stable PD controller for bilateral teleoperators. IEEE Transactions on Robotics (in Press), Feb. (2008)
8. Nuño, E., Basañez, L., Ortega, R., Spong, M.W.: On position tracking for nonlinear teleoperators with variable time-delay. Int. Journal of Robotics Research (Submitted), Nov. (2007)
9. Arcara, P., Melchiorri, C.: Control schemes for teleoperation with time delay: A comparative study. Robotics and Autonomous Systems, 38(1):49–64, Jan. (2002)
10. Hokayem, P.F., Spong, M.W.: Bilateral teleoperation: An historical survey. Automatica, 42:2035–2057 (2006)
11. Nuño, E., Basañez, L., Ortega, R.: Passive bilateral teleoperation framework for assisted robotic tasks. In: Proc. IEEE Conference on Robotics and Automation, pp.1645–1650, April (2007)
12. Nuño, E., Rodríguez, A., Basañez L.: Force Reflecting Teleoperation via IPv6 Protocol with Geometric Constraints Haptic Guidance. In: Advances in Telerobotics, Chapter 26. STAR series. pp.445-458. Springer-Verlag. (2007)
13. Rodríguez, A., Basañez, L., Celaya, E.: Robot Task Specification and Execution Through Relational Positioning. IFAC Workshop on Intelligent Manufacturing Systems, May 23-25, (2007)
14. Rodríguez, A., Basañez, L., Celaya, E.: A Relational Positioning Methodology for Robot Task Specification and Execution. IEEE Transactions on Robototics (Conditionally accepted), Feb. (2008)
15. Waddington, D.G., Chang, F.: Realizing the Transition to IPv6. IEEE Communications Magazine, 40(6):138–147, June (2002)
16. Geer, D.: In brief: IPv6 and distributed applications. IEEE Distributed Systems Online, 6(12):1–4 Dec. (2005)
17. Lee, D.C., Lough, D.L., Midkiff, S.F., Davis, N.J., Benchoff, P.E.: The next generation of the Internet: aspects of the IPv6. IEEE Network, 12(1):28–33, Feb. (1998)