Robotica (2000) volume 18, pp. 677–686. Printed in the United Kingdom © 2000 Cambridge University Press

Hand controller for bilateral teleoperation of robots

José F. Postigo, Vicente A. Mut, Ricardo O. Carelli, Luis A. Baigorria and Benjamín R. Kuchen

Instituto de Automática – Facultad de Ingeniería, Universidad Nacional de San Juan, Av. San Martín 1109 (oeste), 5400 San Juan (Argentina)
e-mail: jpostigo@inaut.unsj.edu.ar

(Received in Final Form: April 15, 2000)

SUMMARY
Teleoperation, one of the oldest areas of robotics, has experienced considerable growth in the last two decades. Main causes for this trend are the need for increased safety levels for human operators and lower production costs. In this work, a three d.o.f. local manipulator (two d.o.f. for force and one d.o.f. for torque) is developed. This hand controller, intended for robot or mobile teleoperation systems, has force reflection in two axes and torque reflection in the third axis. using a robotic hand developed at INAUT as a remote device, laboratory experiments on each axis (one at a time) have shown good results. An impedance controller at the remote system allows one to carry out interactive tasks with the environment such as polishing, insertion and grinding, where it is necessary to control and accommodate the interaction forces and torques in order to avoid hazards for both the manipulated objects and the remote robot.

KEYWORDS: Robotics; Robot teleoperation; Hand-controller; Man-machine interface; Force and position control.

1. INTRODUCTION

The teloperation of robotic manipulators has experienced a considerable growth in the last two decades. Among all robotic devices, manually teleoperated manipulators have a special place.[1] Main reasons can be found in the need for more safety for human operators and lower overall costs of industrial automatic processes. Other tasks, in which the principles of robotic teleoperation can be applied, may vary widely: Repair and maintenance of nuclear reactors, equipment maintenance of earthbase-commanded spaceships, defusing of explosives, fire extinction, mining, teleoperated repair of high and mid-voltage overhead lines, applications in agriculture, telesurgery in medicine, and other applications.[2]

Typical teleoperation systems consist of two robotic manipulators (one at the local station, the other at the remote site), a communication channel, a remote robot interaction environment, and a human operator.[3] Both manipulators are partially controlled by the human operator and by their corresponding local control algorithms, in a shared control structure.[4–6] Usually, master devices for teleoperation systems are developed and built at university research laboratories in order to obtain mechanical characteristics, manoeuvrability and performance tailored to the needs of this laboratory equipment. This is the main reason for the development of a hand controller for bilateral teleoperation systems presented in this work.

When manipulating objects, the human operator uses mainly two of his/her senses: visual and tactile perception of interaction forces.[7,8] One of the main objectives of telemanipulation or remote operation is the total transparency of the interface being used; that is, the actuators of the robotic manipulators execute their commands, the sensors backfeed the measured signals and the operator *feels* like as if he/she were really operating on the device at the remote site.[9,10]

The design and development of a 3 dof local manual controller (hand-controller), two of force and one of torque (with reflection of force in two of their axes and of torque in the third one), and its application in a robot teleoperation control system (or its possible use to teleoperate mobile robots) is presented in this paper. The interface between the local and the remote stations is a Pentium IBM-fully compatible computer. This PC is in charge of the communications (a full-duplex link), and of supplying information to the human operator, about the events at both ends of the teleoperation system. The developed system allows to modify various robot controller parameters, such as the gains of control loops, the transmission speed between local and remote stations, etc. Communication is via a full duplex link, managed by a PC-presiding program in $C++$ language.[11]

The present work is organized as follows: Section 2 describes the constructive features of the developed hand controller. Section 3 covers the proposed control structure for the robotic teleoperation system. In Section 4, the experiments carried out with the hand controller are shown and discussed. Finally, Section 5 summarizes the conclusions and future work.

2. HAND-CONTROLLER FOR ROBOTIC TELEOPERATION

The mechanical structure of the 3 dof hand-controller for bilateral teleoperation of mobile robots and robotic manipulators will be described next.

2.1 *Mechanical features*

The structure of the hand-controller consists of a steel frame to which the mechanical components are fixed. A hole in the

https://doi.org/10.1017/S0263574700002782 Published online by Cambridge University Press

frame houses an steel axis, the load cell prolongation which causes the motions of the end effector. This steel axis joins mechanically the load cell to the actuators (DC motors). At the other end of the load cell, there is a hilt which acts on the hand-controller. A small screw links the steel axis to a steel piece connected to the DC motor shaft which, in turn, acts on the torsion axis (z axis in space).

The mechanical union for motion on the x and y axes is through a pair of crescent-shaped, steel guides with a pair of holes at their ends for connecting them to motor axes. Being of different size for each axis (x and y), these guides are placed one below another concentrically, though separated at an angle of 90°. Each DC motor has a high-ratio reduction gearbox (314:1 for the x- and y-axis motors and 2500:1 for the z-axis motor). Optical encoders placed on supports of the frame sense the hand controller's position. These supports are placed in such a way to have the axes of the motor and the optical encoder in parallel. Transmission between these axes is through a jagged belt-and-pulley set.

Similarly to the x and y axes of above, the motor mounted on the z axis is mechanically coupled through a gearbox reducer to the steel axis. The frame has supports to mount the joint-limiting switches that will operate in case the motor comes close to an extreme position. This will serve as a protection for the motor and to limit the motion on this z axis.

According to its mechanical components, this device allows motions on the three axes (x, y and z), thus leading to a 3 dof hand-controller. Figure 1 shows the mechanical components of the hand controller.

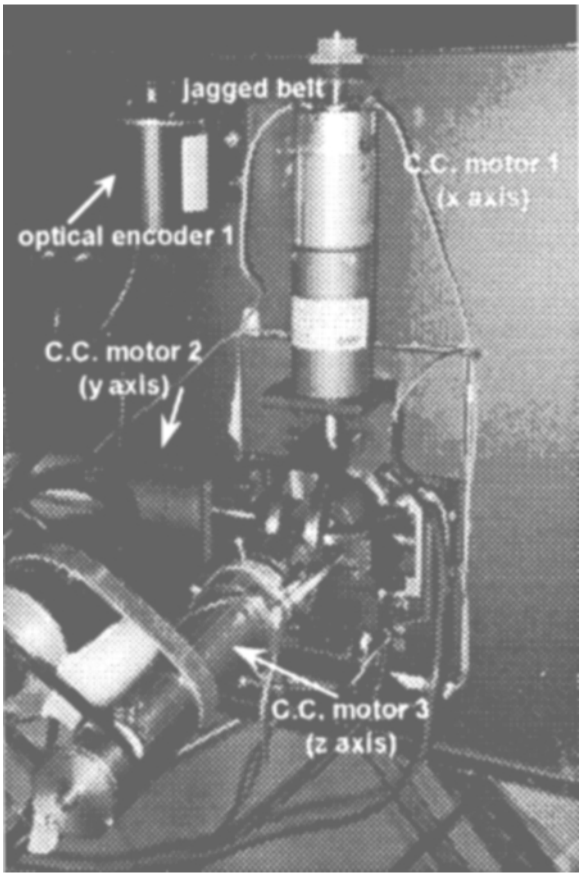


Fig. 1. Mechanical structure of the hand controller.

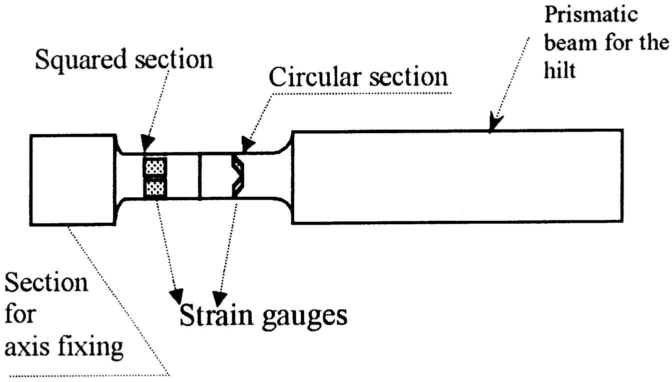


Fig. 2. Hand controller load cell.

2.2 *End effector*

The load cell of the previously described hand-controlled is of cantilever beam type. The beam's dimensions and geometry are shown in Figure 2.

In Figure 2, three main areas can be distinguished:

- The area at left: it is used to fix the load cell to the axis that drives the hand controller's motions.
- The central area: in this zone, the cross-section is smaller in order to concentrate and increase the surface strains of the beam when applying a force or a torque. This section also has two subsections: a square-area, where the strain gauges (for sensing deformations on the x and y axes) are glued to; and a cylindrical portion for measuring the torsion or torque applied to the load cell (in this portion of the load cell, the strain gauges are placed at a 45° angle from the main axis). It can be demonstrated that this 45° angle for the strain gauges is adequate to measure torsion in circular sections. Thus, with this geometrical arrangement, the load cell allows to sense forces in two axes and torque in the remaining one.
- The area at right: this is a longer area, where the hilt is located on. The hilt delimits the grasping area of the end-effector in a way that the human operator may manipulate the hand controller.

In order to process the signals sensed, the strain gauges are connected as a Wheatstone bridge configuration. The bridge, with the strain gauges placed in opposite faces of the square section, allows for a higher sensibility of sensed deformations along the x and y axes. For the circular section, strain gauges especially designed for this type of applications have been used. These commercial strain gauges, constituting a half Wheatstone bridge, are placed at 45° angle with the beam's axis, but with a 90° difference in orientation between both bands of the half bridge.

2.3 *Configurations of the hand controller*

Several connection configurations for the developed hand controller may be devised:

- **Hand controller – Simulated environment:** the hand controller may be connected to a PC containing a program that simulates a virtual environment. This outline (Figure 3) is limited in that the program should administer the communication with the hand controller,

https://doi.org/10.1017/S0263574700002782 Published online by Cambridge University Press

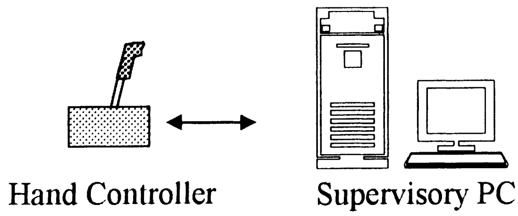


Fig. 3. Hand controller and simulated environment.

respecting the protocol that was defined for the developed applications.

- **Hand controller-PC – PC-remote device:** In this mode, the hand-controller is connected to the PC performing as a supervisor, from which the information goes to another remote PC that drives another device, possibly a robot manipulator or a mobile robot. This outline (Figure 4) does not have any limitation for the local system since the *C++* programs for the PC may be adapted to this supervisory function. It should be foreseen that the PC that manages the remote device, will perform the communication protocol (to be described in 2.4).
- **Hand Controller – PC – Remote System:** In this case, the hand controller is connected to a single PC which is also connected to the remote system. The PC administers the communication between the hand controller and the remote robot, performing as well as an user interface (Figure 5). This scheme has been used for development and experimentation with the hand controller. It should be noticed that the remote system uses the communication protocol that the hand controller uses as well.
- **Hand Controller – Remote Manipulator:** Here, the hand controller is connected directly to the remote manipulator without intervention by the supervisory PC (Figure 6). In this case, the information cannot be shown, something that is possible when having a supervisory PC (as in previous schemes). The remote manipulator will manage the communication to and from the hand controller, according its defined protocol.

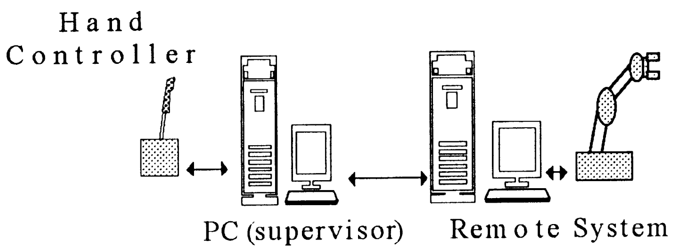


Fig. 4. Configuration: Hand Controller-PC-PC-remote system.

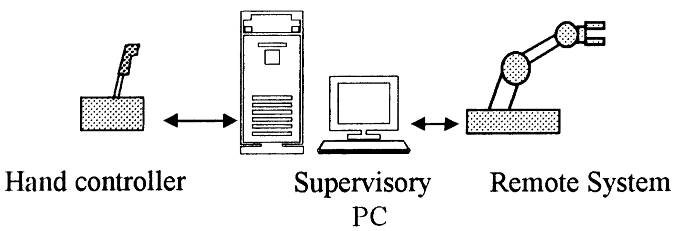


Fig. 5. Hand Controller-PC-Remote Robot configuration.

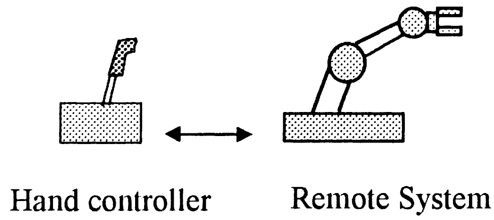


Fig. 6. Hand Controller - Remote System configuration.

2.4 Operation of the Hand controller

The general control structure proposed for the local manipulator's control (hand-controller), which meets the initial requirements for versatility, user-friendly environment and electronics optimisation, is depicted in Figure 7.

From Figure 7, the following functional blocks may be highlighted:

(i) *Supervisory PC:* It carries out the graphic interfacing with the user and manages the communication between local manipulator (hand-controller) and remote manipulator or device.

(ii) *Communication system:* It conveys the information digital signals at the transmission speed as specified by the user. This system uses the RS-232 C protocol to communicate with the PC.

(iii) *Control unit (CU):* This unit conforms the main element for operating the hand controller, and it is a part of the 80C196KC microcontroller. The control unit main task consists of executing the algorithms in Assembler language. These algorithms conform the position and force control structure of the local hand controller. The CU generates the electric PWM signals, the gyrating sense -clockwise or counter-clockwise-, being the control actions that command the actuator through the amplifier block, and other signals. In this block, control algorithms for each one of the variables to be controlled have been programmed. It operates with feedback signals from different blocks of the measurement electronic circuits, and signals from the supervisory PC as well.

(iv) *Amplifier-Actuator:* This block carries out the signal adaptation of the control actions generated in the control block, to apply them properly to the motor-actuator block. This system generates the signal to drive the motors of the actuators, based on the information stemming from the control block.

(v) *Protection system:* Its function consists of protecting the motors of the three axes of the hand controller during its operation, not allow them to exceed the nominal current of the motors.

(vi) *Motor-Actuator:* The driving D.C. motor supports the torques and the opposition forces that are applied on the hand controller's end effector and is in charge of transmitting the motions to the mechanical system.

(vii) *Mechanical system:* It is constituted by the structure of motion transmission to the end effector and by the iron supporting shell.

(viii) *Force measurements:* This block is constituted by the devises in charge of obtaining the signals coming from the Wheatstone bridges.

https://doi.org/10.1017/S0263574700002782 Published online by Cambridge University Press

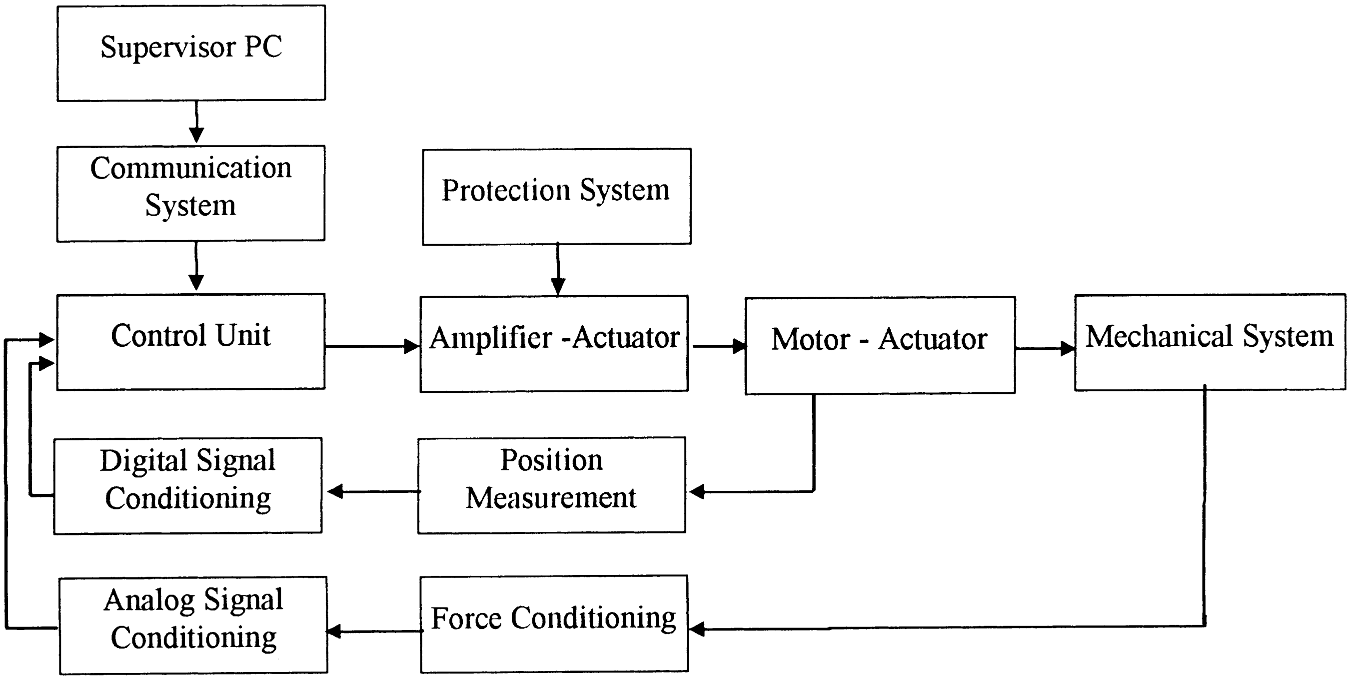


Fig. 7. General operation structure for the hand controller.

(ix) *Analog signal conditioning:* This block is the responsible of giving the appropriate electric format to the measured force signals.
(x) *Position measurement:* This device executes the position measurement of each one of the three axes.
(xi) *Digital signal conditioning:* After obtaining the signals of the position measurement block, they are processed in the block of digital signal conditioning, giving them an appropriate format for their interpretation in the control unit. this block generates signals of gyrating sense and frequency for each motor.

The linking of all these functional groups, allows one to carry out the hand-controller's control, fulfilling the wanted outlined requirements of functionality.

3. CONTROL STRUCTURE

As previously mentioned, the supervisor PC is in charge of defining in what mode the hand controller will operate. Initially, the system begins in a position control mode until interconnecting both systems (remote system and local system through the communication channel). Once the position references, sensed and backfed by the remote system, are given, the local system executes them to position the hand controller. At the same time, the supervisor PC verifies that the positions of both systems be closely enough. When the local and the remote robot's positions are close to each other, the supervisor PC enables the human operator to commute the hand controller to a force control mode. This will remain in this mode until the human operator instructs the supervisor PC to commute the hand controller again to position control.

The control algorithms that are used to generate the control actions in each control loop are the conventional PI type. Since the hand controller has three degrees of freedom, the three axes should be controlled in an independent form by means of individual control loops. This way, the system has three control loops: one for the y axis, another for the x axis and one for the z axis, respectively. The control algorithms were developed in the *assembler* language provided for the 80C196KC microcontroller (later, these algorithms are implanted in it.)

3.1 *Control algorithms*

The control algorithms for the robotic teleoperation hand controller were developed in *assembler* language in order to manage the 80C196KC microcontroller.[12] A *C++* language algorithm has been developed to be an interface (from the server PC) between the user and the rest of the robotic teleoperation system.[13] In the server PC one may visualise the state of the variables that are being controlled, as position and forces of the remote manipulator. This program also allows one to modify the gain constants of the control loops, as well as the data transmission speed in a bi-directional mode between the local and the remote stations. These developed algorithms are described in the following.

3.2 *Installation in assembler*

The developed program consists basically of a main program and several subroutines that carry out specific tasks, which are mentioned in the following sections.

3.2.1 The main program. The function of processing the algorithms of the three independent control loops implanted in each axis of the hand-controller is carried out by the main program. In this program, the position update of the three D.C. motors that drive the hand controller, is executed. The commutation of position control to force control, or vice-versa, can be done with this program. Therefore, the

https://doi.org/10.1017/S0263574700002782 Published online by Cambridge University Press

operation sequence of each implanted algorithm, is controlled.

3.2.2 Control loop communication method. Since the main algorithm should process the three control loops implanted in the teleoperation system, it is necessary to control the evolution of the program, in such way that these loops are executed in a sequential and periodic way, independently of the hand controller's control mode (position control or force control). To carry out this function, the program is provided with a variable denominated "axis" that acts as an indicator to know which it is the axis that should work next (according to hexadecimal values, previously established). When the program finishes processing a control loop corresponding to any axis, and before entering to the following control loop, the program verifies the "*axis*" value, jumping to the control loop subroutine that this variable indicates. Once it has entered the loop and it has been processed, before leaving the loop, it puts in the variable "axis" the value that corresponds to the following axis of the sequence. In this way, the program process each control algorithm in a sequential and periodic mode.

3.2.3 Method to commute between position control and force control. The method that is used to commute from position control to force control (and vice versa) in the hand controller, is based on the use of a variable denominated "*mode*", that indicates, according to a predetermined hexadecimal value, to the main program if it should process an algorithm of position control or an algorithm of force control, for the axis in question. Initially, the program begins controlling position, because the hand controller should be positioned until the communication with the remote system has been established. The commutation from one control mode to the other one will depend on the supervisor PC. The commutation signal is sent through a chain of data containing the order of commuting, the operation mode. This chain is inserted with the normal information transmission chains, so that its effect is immediate. This operation is made by the program running in the supervisor PC, in the moment that it enabled the human operator to commute the operation mode.

The manual controller also has a protection against communication unexpected interruptions. This protection is activated if the communication is interrupted by any reason, between the local station and the remote station. In that case, the hand controller will be permanently controlling position according to the last position reference that it received as reference position. In this way the human operator makes strong efforts sustaining the hand-controller till the situation is retrieved.

3.3 C++ installation

The communications handling between the local station and the remote system of the robotic bilateral teleoperation system is done through a computer. This PC is responsible for the control of the communications, the error handling, the supervision of the change of variables and also some other tasks. The program was developed in *C++* language due to the power of this language to generate functions such as the computer interruption handling, processing of mathematical equations, readiness to low level programming and the possibility of developing information visualisation functions in a simple way.

The program running in the computer uses the serial port 1 (COM1) of the computer to communicate with the hand controller, because the microprocessor 80C196KC communicates using a series protocol that has been adapted to the RS-232C protocol. We also selected the COM2 port to link the local station with the remote system. This should be changed for some other type of adaptor to carry out the communication task (such as a net communication board, the parallel port of the computer, etc.).

The program developed in *C++* uses the interruptions of reception of the series parallel PC adapter to administer the whole communication scheme of the robotic teleoperation system. The program periodically executes the sequence of events that are detailed in the following: The first event that takes place is the determination of a pressed key of the PC keyboard. As a second step, the program runs one of the different types of possible communications. Finally, it proceeds to visualise the data transferred between both stations (local and remote).

4. EXPERIMENTATION

In the experimental phase, the developed hand controller performance was tested connecting it to an intelligent two-fingered robotic hand, developed at the INAUT, considering it as a remote device (remote robot). This structure allows to teleoperate a remote robotic hand with the local hand controller. The constructive characteristics and associated electronic of this robotic hand are described in detail in reference 14. The communication system developed for the hand controller was also implanted in the remote robotic hand.

The robotic bilateral teleoperation system works in the following way: the server PC receives the data from the hand controller through the serial port Com1 and from the remote robotic hand through the serial port Com2. The position and force signals backfed from the remote hand are sent towards the hand controller as sensed signals, and the signals coming from the hand controller are sent to the remote robotic hand as reference commands.

4.1 Experimental results

The robotic teleoperation system was tested at several communication speeds.[15] Figure 8 shows a picture of the developed system.

Some of the experiments that were carried out with the teleoperation system are discussed in the following section.

(i) *Experience 1: Different data transmission speeds*

In this experiment, the 3-axes hand controller was used as the local robot and, as the remote robot, the INAUT's intelligent robotic two-fingered hand was implanted. It should be noticed that though the hand controller has three

https://doi.org/10.1017/S0263574700002782 Published online by Cambridge University Press

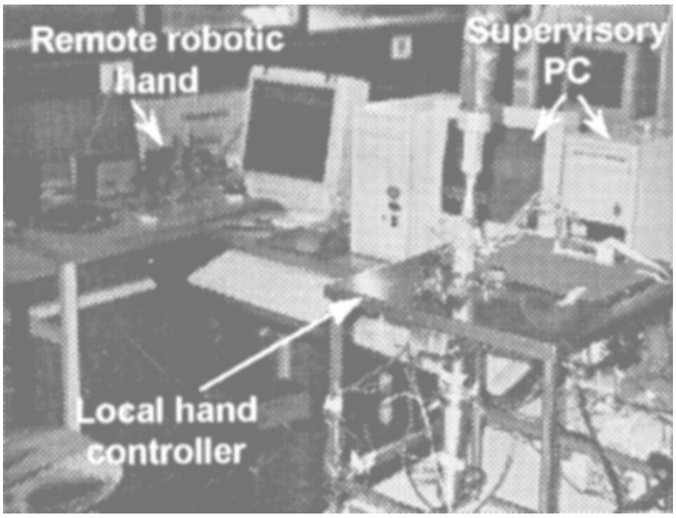


Fig. 8. Experimental robotic teleoperation system.

degrees of freedom, the remote hand will be able to control in only one space direction (since it only has one degree of freedom). Therefore, "y" direction of the hand controller was chosen to control the position and force in it. Besides, it was also proven that, with a greater proportional constant of the force algorithm, the teleoperation system answered to a smaller force applied by the human operator in the free motions. If the proportional constant was too high, in the moment of a crash with the environment, the system tends to be more unstable. This situation can be improved (as well as the answer of the teleoperation system) to give certain autonomy to the remote station, it included an impedance control loop.[6,16]

The impedance of a mechanical system is defined as the dynamic relationship of the applied force and the displacement velocity,

$$\mathbf{f}(t) = Z(p)\nu(t) \tag{1}$$

where $\mathbf{f}((t)$, $\nu(t)$ y $Z(p)$ represent the force, the velocity and the impedance of the mechanical system, respectively, and $p = d/dt$ is the derivative operator. In terms of position $\mathbf{x}(t)$, Eq. (1) becomes,

$$\mathbf{f}(t) = Z(p)p\mathbf{x}(t) \tag{2}$$

In this case, a desired motion trajectory $\mathbf{x}_d(t)$ is specified for the robot manipulator and the robot impedance is defined by,

$$\mathbf{f}(t) = Z(p)p\tilde{\mathbf{x}}(t); \quad \tilde{\mathbf{x}}(t) = \mathbf{x}_\mathbf{d}(t) - \mathbf{x}(t) \tag{3}$$

where $\mathbf{f}(t)$ is the force applied by the robot's end effector against the environment, and $\tilde{\mathbf{x}}(t) = \mathbf{x}_\mathbf{d}(t) - \mathbf{x}(t)$ is the motion error of the robot.

A desired impedance should be specified to establish the robot's behaviour in an impedance control structure. It is natural, for its simplicity, to establish a lineal relationship. Also, as the dynamic behaviour (model) of the robot is of second order, it is logical to specify a second order desired impedance by,

$$\mathbf{f}(t) = (\mathbf{M}p^2 + \mathbf{D}p + \mathbf{K})\tilde{x}(t) \tag{4}$$

where **f** represents the "applied" force by the robot against the environment. The matrix **M** is called inertia matrix, **D** is the damping matrix and **K** is the elasticity matrix. Matrices **M, D, K** are specified according to the desired dynamic behaviour of the robot.

In the proposed impedance loop, the elastic constant of the impedance term of Eq. (4), in the remote system was only considered (intelligent robotic hand). Figure 9 shows the results for a transmission speed of 57600 bauds in a) position and b) force control in "y" space direction. In Figure 10, the same experiment is shown, without varying the parameters of the controllers neither the remote impedance, but for a transmission speed of 300 bauds.

It should be noticed, for the correct understanding of the time scale of all the figures (from Figure 9 to Figure 14) that it is given in program cycles. The equivalence is the following:

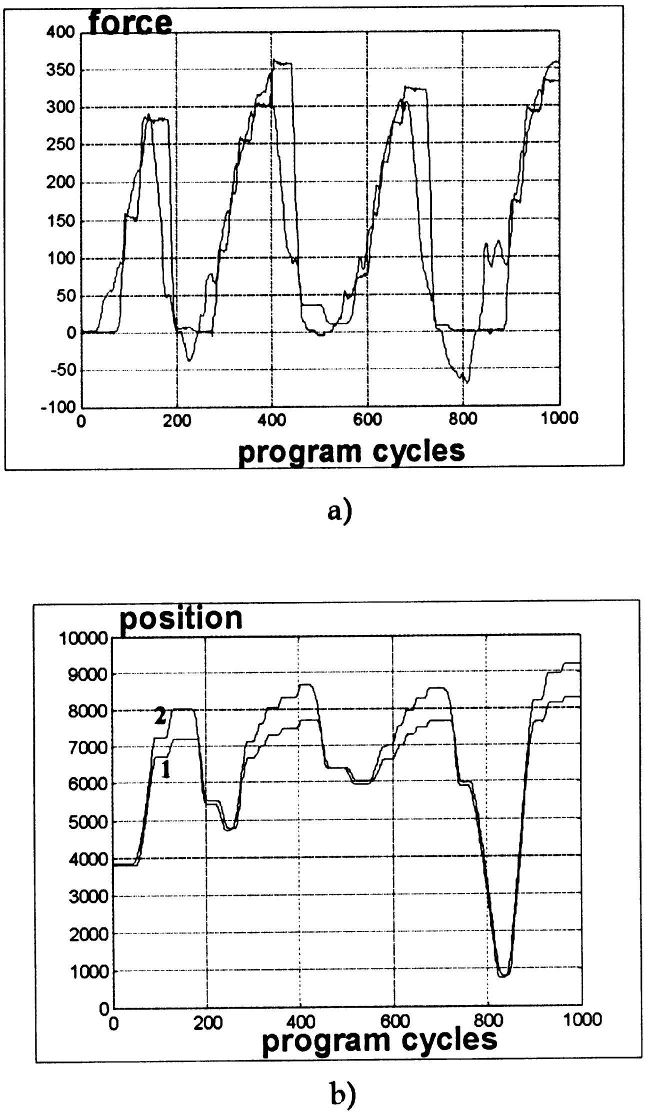


Fig. 9. a) **1**: Force measured in the fingers of the remote robotic hand, **2**: Force measured in the local hand controller (axis **y** only). b) **1**: Remote robotic hand position, **2**: Hand controller position (axid **y** only). Transmission speed: 57600 bauds.

https://doi.org/10.1017/S0263574700002782 Published online by Cambridge University Press

1 program cycle = 0.506 seconds, for a transmission speed of 300 bauds;
1 program cycle = 15.8 mseconds, for a transmission speed of 9600 bauds;
1 program cycle = 2.63 mseconds, for a transmission speed of 57600 bauds.

It also should be taken into account that in position figures (e.g. Fig. 9) b), 10000 pulses of the optical encoder are equivalent to 22.92° of change in the position of the hand controller (or 0.4 radians).

These conversions will be kept in mind for the comparison of all the mentioned figures. For example, the time scale of Figs. 9a) and b) represents a total period of 6 seconds.

If these lat two figures are analysed (Figs. 9a) and b) and 10a) and b)), it can be observed that for a lower transmission speed (300 bauds), as it was expected, the system responds in an uninterrupted way due to the great sampling time generated in the communication. When increasing the communication speeds, this allowed one to increase the system stability and to obtain this way a bigger fluency during free motions reflected in the teleoperation system. As for the constrained motions (contact of the remote system with an object or an environment), it was proven that the teleoperation system was unstable at low transmission speeds. This problem was solved when increasing the communication speed to 57600 bauds. For a reader's reference and comparison, experiments at a transmission speed of 9600 bauds are shown in Figure 11 (a) and b). It can be noticed that the system's performance is better than those at 300 bauds, but worse than the ones at 57600 bauds transmission speed. This better performance is emphasised in teleoperation with force reflection experiments.

The experimentation conditions were the same of the experience 1, but it was changed, for the same transmission speed between the local and remote stations (57600 bauds), the elasticity term of the fictitious impedance implanted in the remote robotic hand. In Figures 12 a) – b) and 13 a) – b) the experimental results are shown.

Experimentally it was proven that the events that take place in moments of crash of the remote system with some objects (or with the environment) were improved, regarding the systems' answer, allowing better manipulation of them. The bigger the K constant values, the more stable response of the teleoperation system (as shown in Figures 12 and 13).

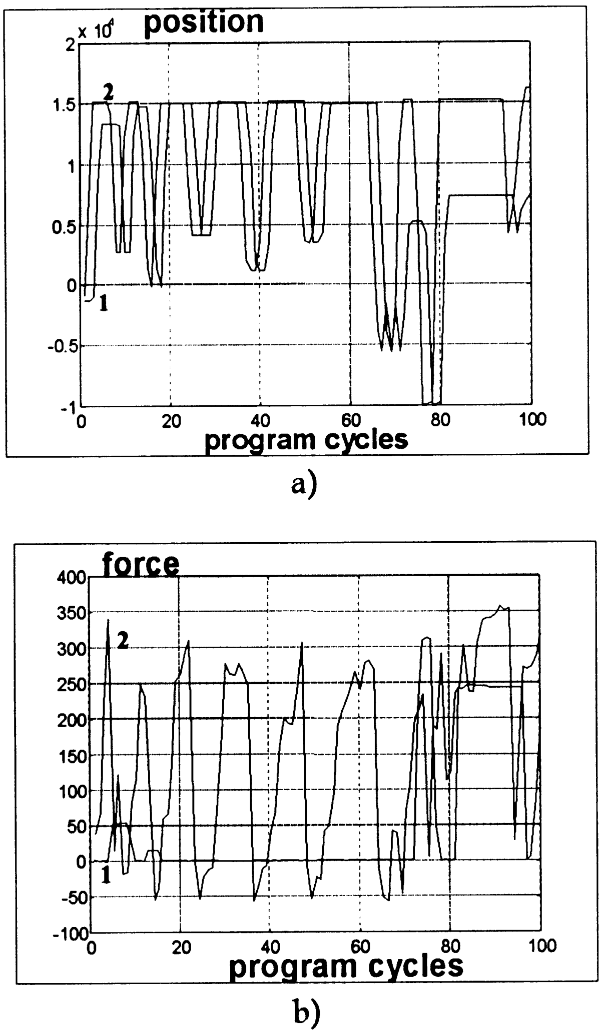


Fig. 10. a) Position and b) force, with the same parameters of Fig. 9 a) and b). Transmission speed: 300 bauds.

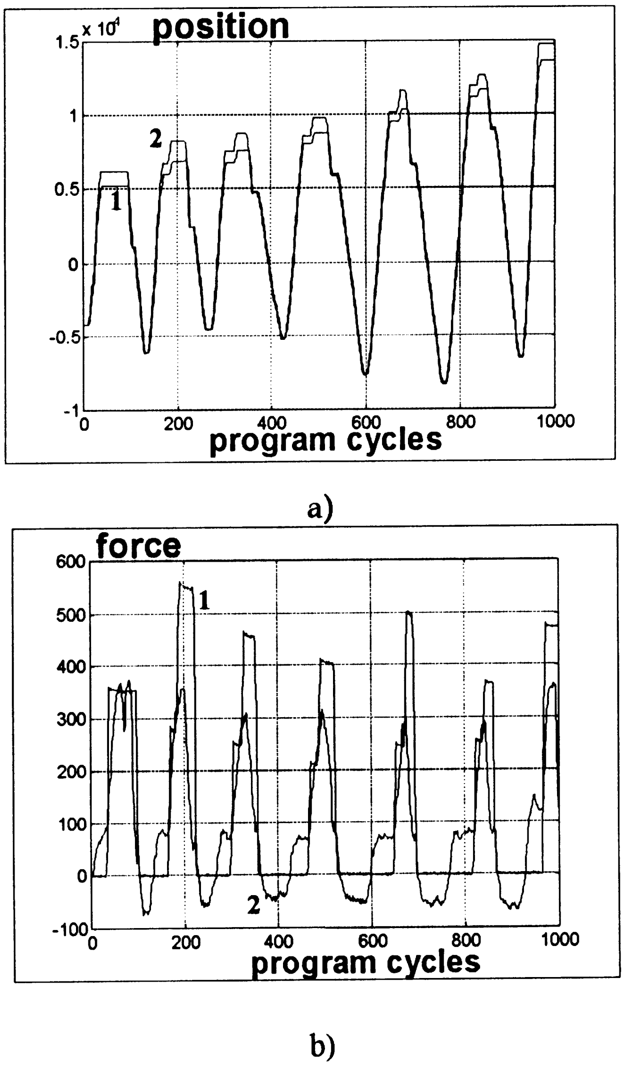


Fig. 11. a) position and b) force, with the same parameters of Fig. 9 a) and b). Transmission speed: 9600 bauds.

https://doi.org/10.1017/S0263574700002782 Published online by Cambridge University Press

Finally, Figures 14 and 15 depict the experimental results (for the same elasticity value) of the developed hand controller. It can be clearly seen that the behaviour and performance of the band controller worsens when the proportional constant K_p of the PID force controller is decreased (with an integrative constant $K_i = 0$ in both cases).

5. CONCLUSIONS

In this work, the development and experimentation of a three degree of freedom hand controller (force in x-y axes and torque in the z axis) for robot bilateral teleoperation systems, has been presented.

The control system for the bilateral teleoperation structure was designed in such a way that the human operator, acting on the hand controller, sends position and force commands to the remote system through a communication channel, and at the same time receives the position and force signals that the remote system reflects when it senses its interaction with the environment (bi-directional information in the communication channel) in constrained motion tasks.

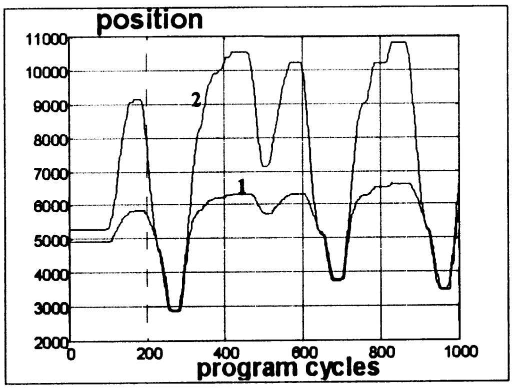
It was possible to design, to build and to implant the hardware and the associated electronics necessary for the correct operation of the manual controller's mechanical structure (three degrees of freedom hand-controller).

Also done was the assembly and the mounting of the D.C. motors (in the corresponding two axes of force reflection and in the remaining torque reflection axis).

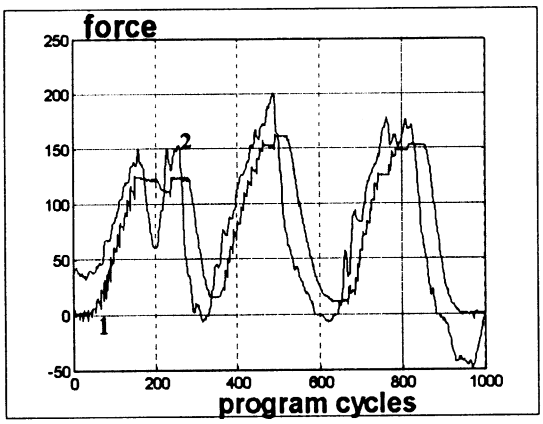
The control algorithms, for each of the three motors of the hand controller, were designed and programmed in the *assembler* language of the MC80196KC microcontroller. The software necessary to visualise the graphic environment of interface between the user, the developed hand controller and the remote environment where the teleoperation task is performed, was developed as well. The graphic information to the human operator added to the control loop (that is, the visual feedback of force and position of the remote robot), allows to set accurately the desired values of interaction force with the environment.

Finally, an experimental bilateral teleoperation setup was mounted, using an intelligent robotic hand as the remote system (developed at the INAUT). This setup allowed one to exhaustively test the teloperation system in order to show the force and torque reflection properties of the local station hand controller. These tests show an stable behaviour and good performance of the whole telerobotic system.

The substitution of the one dof remote robotic hand for an industrial robot, a 4 dof Bosch SR 800, is one of the future

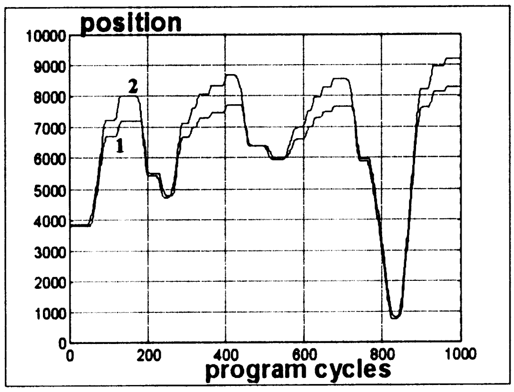


a)

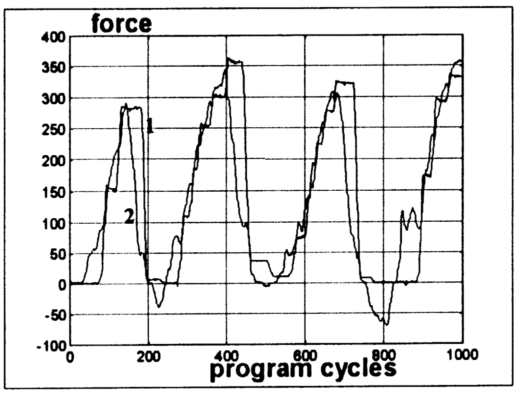


b)

Fig. 12. **1**: Remote robotic hand position, **2**: Hand controller position (axis **y** only). b) **1**: Measured force at the fingers of the remote robotic hand, **2**: Measured force of the local hand controller (axis **y**), elastic constant of the remote hand impedance K: 16. Transmission speed: 57600 bauds.



a)



b)

Fig. 13. a) **1**: Remote robotic hand position, **2**: Hand controller position (axis **y** only). b) **1**: Measured force at the fingers of the remote robotic hand, **2**: Measured force of the local hand controller (axis **y**), elastic constant of the remote hand impedance K: **2**: Transmission speed: 57600 bauds.

https://doi.org/10.1017/S0263574700002782 Published online by Cambridge University Press

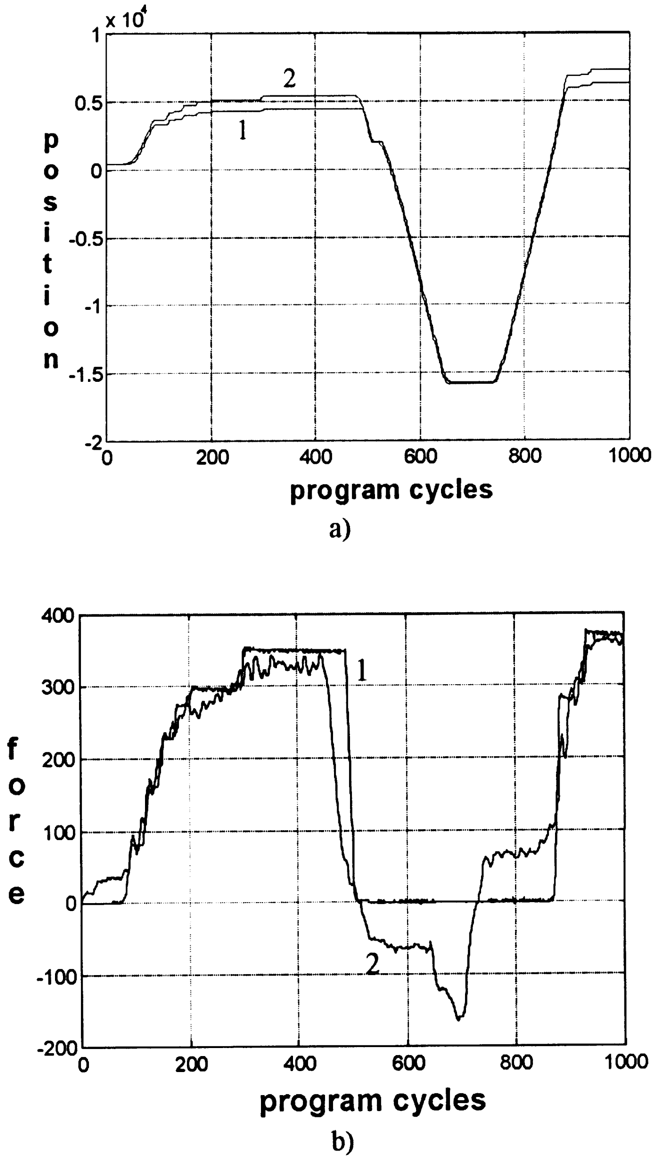


Fig. 14. a) **1**: Remote robotic hand position, **2**: Hand controller position (axis **y** only). b) **1**: Measured force at the fingers of the remote robotic hand, **2**: Measured force of the local hand controller (axis **y**), elastic constant of the remote hand impedance K: 2. Transmission speed: 57600 bauds. Proportional constant of the PID force controller: $K_{pf} = 80$.

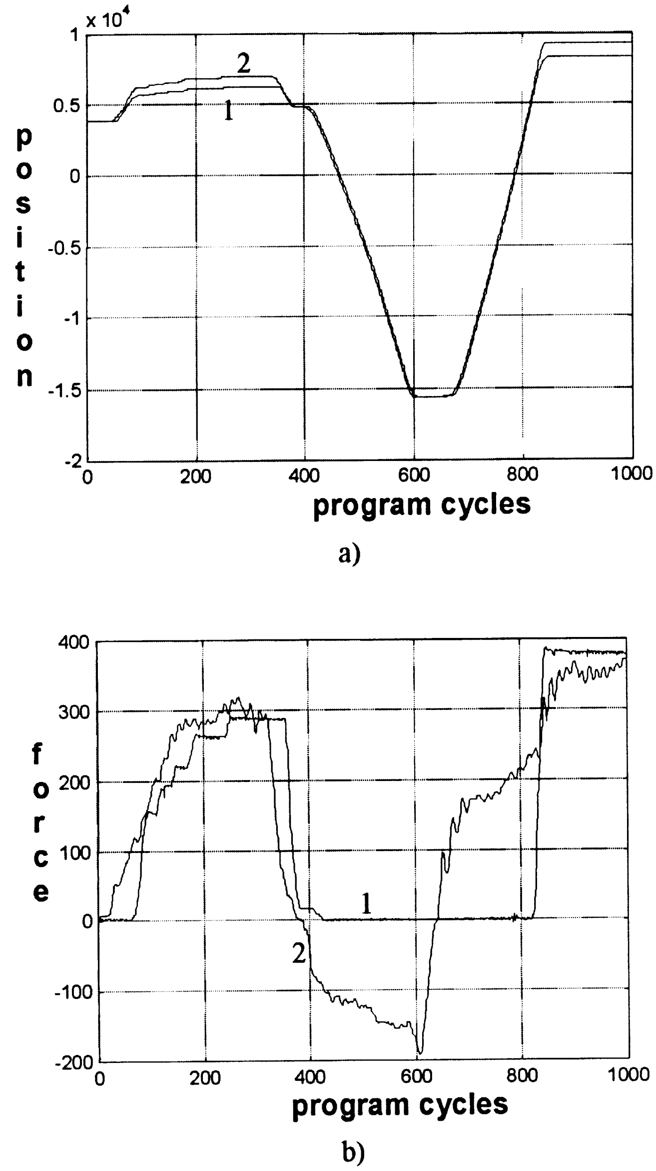


Fig. 15. a) **1**: Remote robotic hand position, **2**: Hand controller position (axis **y** only). b) **1**: Measured force at the fingers of the remote robotic hand, **2**: Measured force of the local hand controller (axis **y**), elastic constant of the remote hand impedance K: 2. Transmission speed: 57600 bauds. Proportional constant of the PID force controller: $K_{pf} = 20$.

tasks to cope with. The main idea is to teleoperate this industrial robot with the developed hand controller, while performing interaction tasks, applying advanced control structures. Furthermore, it is also a future challenge the teleoperation of a mobile robot with the developed hand controller, when performing obstacle avoidance tasks.

ACKNOWLEDGEMENTS

The authors would like to thank Carlos M. Schugurensky for the design of the sensing system and for his technological advices, as well as to Alberto Garcia Brizuela for his useful suggestions. This work was financed partially by the National Council of Scientific and Technical Research (CONICET), Argentina.

References

1. L.I. Slutski, *Remote Manipulation Systems: Quality Evaluation and Improvements* (Kluwer Academic Publishers, 1998).
2. P. Fiorini and R. Oboe, "Internet-Based Telerobotics: Problems and approaches, *1997 International Conference on Advanced Robotics (ICAR '97)*, Monterey (CA) (July 1997) pp. 765–770.
3. V.S. Kuleshov and N.A. Lakata, *Remotelly Controlled Robots and Manipulators* (Mir Publishres, Moscow, 1988).
4. H. Das, H. Zak, W.S. Kim, A.K. Bejczy and P.S. Schenker, "Operator performance with altrenative manual control modes in teleoperation), *Presence*, **1**, 2, 201–218 (1992).
5. C. Guo, T.J. Tarn, N. Xi and A.K. Bejczy, "Fusion of human and machine intelligence for telerobotic systems", *Proc. IEE Int. Conf. on Rob. and Automation* (1995), pp. 3110–3115.
6. J. Postigo and R. Carelli, "An Adaptive Impedance Control for a Bilateral Robot Teleoperation System", *Submitted to the 6th*

International Symposium on Robot Control – Syroco 2000, International Federation of Automatic Control (IFAC). Vienna, Austria (September 21–23, 2000 (to appear).

7. G.C. Burdea, *Force and Touch Feedback for Virtual Reality* (John Wiley & Sons, Inc., 1996).
8. W.R. Mark, S.C. Randolph, M. Finch, J.V. Van Verth and R.M. Taylor II, "Adding Force feedback to graphics systems: Issues and Solutions," *Proceedings of SIGGRAPH 96*. New Orleans, Louisiana (August 4–9, 1996). **In:** *Computer Graphics Proceedings, Annual Conference Series, 1996, ACM SIGGRAPH*, pp. 447–452.
9. T.B. Shreidan, *Telerobotis, Automation and Human Supervisory Controlf* (MIT Press, USA, 1992).
10. J.E. Colgate, M.C. Stanley and J.M. Brown, "Issues in the Haptic Display of Tool Use", *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Pittsburgh, PA (1995) pp. 140–145.
11. C.H. Pappas and William H. Murray III, *Manual de Borland, C++*, version 3.1 (Editorial Osborne/McGraw-Hill, 1993).
12. EV80C196KB, KS Microcontroller Evaluation Board, *User's Manual Intel. Corp.* (1989).
13. Microsoft Corporation, *Visual C++ Programmer's Guide* (Microsoft Corporation, 1997).
14. J. Tramontin, "Garra Robotica sensible – GRII (Sensitive robotic hand – GRII)," *Internal report No. IR-164/97-UNSJ* available at Biblioteca del Inaut, Universidad Nacional de San Juan, San Juan, Argentina, in Spanish, (1997).
15. J. Funda, *Teleprogramming: Overcoming communication delays in remote manipulation.* (Computer and Information Science Department, University of Pennsylvania, June, 1990).
16. N. Hogan, "Impedance control: An approach to manipulation," *Part 1: Theory. Journal of Dynamic Systems, Measurement and Control*, **107**, 1–7 (1985).