HAL
open science

Modeling and simulation of robotic tasks teleoperated through the internet

Arnaud Lelevé, Philippe Fraisse, Pierre Dauchez, François Pierrot

To cite this version:

Arnaud Lelevé, Philippe Fraisse, Pierre Dauchez, François Pierrot. Modeling and simulation of robotic tasks teleoperated through the internet. AIM: Advanced Intelligent Mechatronics, Sep 1999, Atlanta, United States. pp.299-304, 10.1109/AIM.1999.803182 . hal-00189549

HAL Id: hal-00189549

https://hal.science/hal-00189549v1

Submitted on 8 Mar 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Modeling and Simulation of Robotic Tasks Teleoperated through the Internet

A. Lelevé, P. Fraisse, P. Dauchez, F. Pierrot
LIRMM - UMR 5506 CNRS / Université Montpellier II
161 rue Ada - 34392 Montpellier CEDEX 5 - France

***Abstract*— According to their research activities, some laboratories have to set up expensive full-scale experiments requiring the use of teleoperated robots. In order to limit these costs, it is of interest to share the same equipment between laboratories and to teleoperate it directly from each laboratory through the Internet. In this context, we have initiated a study on a basic teleoperation diagram. Starting from experiments with a land mobile manipulator, we have built a generic simulation model that has been used to find improvements for later teleoperation experiments. The results presented in this paper are mainly based on simulations. Nevertheless, they will be very useful for the real experiments we are currently preparing.**

***Index terms*—Teleoperation, Internet**

1. PRELIMINARIES AND PROBLEM DESCRIPTION

1.1 Experiment sharing

Nowadays, not every laboratory can afford running full-scale experimentations. A typical example is underwater data collecting by means of an underwater vehicle in the middle of the Atlantic Ocean [1]. This kind of experimentation requires heavy equipment, technical staff, traveling to the experimentation site, several days or weeks of preparation and at last a lot of money.

As some of these experiments have to be locally teleoperated because a human cannot have access to the work place, it seems logical to extend the frame of this kind of short distance teleoperation (from the boat to the underwater vehicle) to a long distance teleoperation (from the laboratory to the underwater vehicle, via the boat) where qualified people could remain in their laboratories to control the experiment, using a bare PC computer. This would decrease a little the cost of the experiment and would save time for the people who need the results.
In the same manner, we can consider other laboratories, located anywhere around the world, using the same experiment, each one in its turn, to get their own data. This way, one could, as a first approach, divide the cost of such experimentations by the number of laboratories using it.
Still in the prospect of limiting the costs, each laboratory would just have to own a standard equipment to control the remote experiment. Yet, the most standard equipment that every laboratory has is a PC computer linked to the Internet. This involves creating a software which an operator could use to easily control the remote system.
It is in this outlook that we are studying a way to improve the long distance teleoperation diagram through the Internet.

1.2 Where teleoperation acts

One has to use a teleoperation diagram when it is not possible for anybody to be in the operation site. This because the place may be hazardous for a human (dangerous rooms of a nuclear plant, explosive area, in space [2], under water, …) or even impossible to reach (far planets, very deep underwater areas).
This condition does not prevent an autonomous robot to perform the task. So the second condition is that the task be too complex for an autonomous robot to accomplish it and/or the environment evolve too much (at worst, it is unknown) regarding the intelligence of the robot (once again, space and deep underwater examples are appropriate).

1.3 Technical problems

The long distance control of a remote system requires the use of different means of information flow.
For instance, the control of an underwater vehicle via the Internet from any laboratory around the world requires the data to go at first through the local network (Ethernet, Token Ring, ...) of the master laboratory. Then through the Internet and its numerous links and switchers. Possibly through a wireless network from the ground to a boat and, at last, through a wireless acoustic link (if there is not any tether) from the boat to the underwater vehicle.

This kind of association causes two main technical problems in teleoperation : limited bandwidth and sometimes high communication delays due to the propagation, packetisation and the many other events the Internet relays may inflict on the data [3]. Moreover the bandwidth and the delays may vary according to the events that occur all along the transmission line. In acoustic transmission, round-trip delays greater than 10s and rates smaller than 10kbits/s are common.

1.4 Logistics problems

These technical restraints result in difficulties for the operator to securely control the remote system. If he has to wait at least 10 seconds before being able to see the result of his action, it is very difficult for him to perform his task and in case of trouble, he may notice it too late. It seems necessary to add a minimum of intelligence in the teleoperated system to avoid critical situations.

Moreover, the bandwidth can prevent the use of real-time video views of the experimentation site. This requires the man-machine interface to display a 2D/3D model of the experimentation site, which cannot be accurate when the operating site is ill-known or evolves.

2. USUAL APPROACHES

2.1 Autonomous robot

We consider in this paper tasks that cannot be performed autonomously by a robot. Some kind of teleoperation is mandatory.

2.2 Master/Slave control

This bilateral structure is well-known in telerobotics systems. The force reflection allows the operator to handle very efficiently a remote system. This architecture is very common for teleoperation cases when communication delays do not exceed ¼ second. Last decade researches have brought diagrams to hold higher variable delays (up to a few seconds) so that a force reflection can be used. However this requires a large bandwidth [4].

2.3 Macro-instructions and teleprogramming

When the bandwidth is small and the delays are prohibitive, the common solution is to send a macro-instruction to the remote system and wait for its completion. As the power of on-board computers increases, the intelligence of the remote system enhances; in the first experiments, the robot could only perform the task it was programmed for. More recent experiments showed remote robots able to follow a pre-loaded (possibly dynamic) trajectory, grip a cubic piece with force control, … [5]. Briefly speaking, the robot becomes momentarily autonomous with respect to the goal commands it continuously receives. Bare or mature security procedures may take place in case of wrong maneuvers. It is for instance possible to specify areas or volumes of tolerance for every motion; if the system goes outside theses limits, a security procedure takes place.

Efforts in improving the man-machine interface [6] have provided the ability for the operator to previously simulate the macro-instructions before sending them and to match a virtual reality model (regularly refreshed by the returning data) of the performing robot with a video stream [7] (the available bandwidth may limit the stream to periodic pictures of the experiment site).

Virtual Reality instruments as 3D helmets and glove sensors magnify the handiness of the man-machine interface.

This kind of teleoperation diagram is often called *Teleassistance* or *shared control* because sometimes the operator directly controls the remote system and at other times the system is momentarily autonomous with goals to achieve. Making the remote system more autonomous is the common way researchers are working on. It helps the handiness of the whole control and prevents from delay drawbacks. This is the framework we are interested in.

3. TELEOPERATION ARCHITECTURE

We aim at teleoperating a mobile manipulator through the Internet. The next paragraph introduces the teleoperation architecture we plan to implement.

In order to build this general architecture, we have started from a few basic teleoperation experimentations with a terrestrial vehicle. Section 4 details our experimental set-up and some of the results of these preliminary experimentations.

These results helped us to create a generic simulation model of a monodimensional teleoperation system (see section 5). This model allows us to test some improvements of the teleoperation diagram. These improvements will be tested in the future with our experimental set-up.

3.1 Teleoperation architecture

Teleoperating a mobile manipulator through the Internet involves high and highly variable delays, small and varying bandwidth and also a remote environment that may be evolving or hardly known.

We will use the term "*base*" to refer to the host computer that features the distant control and the man-machine interface. The "*remote system*" will point out the system to be teleoperated. To simplify the reasoning, we will consider a mono-dimensional system. Figure 1 gives a global view of the teleoperation diagram and the next sections give details on each block.

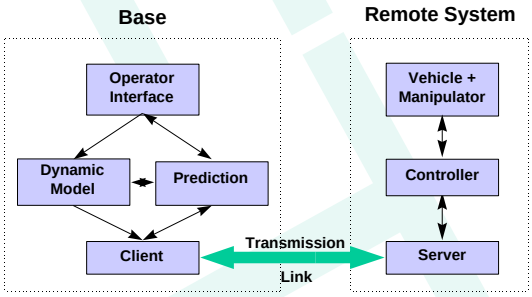

Figure 1 - Summary of the functions of the base and the remote system

3.2 The remote system

The remote system includes the communication with the

base (*server block*) and the local control of its state variables: velocity, direction, axes of the manipulator (*controller block*). It also features security systems in case of breakdown or loss of communication.
We will see later in this paper how we improved the whole system by adding a transmission delay compensation system. We also plan to add a video stream and to implement high-level maneuvers.

3.3 The base

The *client* block communicates with the remote site through the *transmission link*. The *operator interface* block acts as a virtual reality man-machine interface. It detects and displays security warnings in case of communication trouble. The association of the *prediction* and *dynamic model* blocks allows us to make pre-simulations and display a prediction of the remote system state before receiving it by the transmission link. Details are given in the simulation results of section 5.

4. INITIAL EXPERIMENTATIONS

In order to build a simulation model, we have performed preliminary experiments using a terrestrial vehicle equipped with a 6 degree-of-freedom manipulator (d.o.f.) [8]. The set-up includes the base, the mobile manipulator and the communication between both of them.

4.1 The remote system

It consists of a terrestrial 6x6 vehicle fitted with a *PUMA* manipulator, visible in Figure 2. For these experiments, we just send desired values to a DSP-based controller that computes *PID* control laws on every axis of the mobile manipulator. A *TCP* server initiates the communication with the base. For security reasons, a low-pass filter limits the desired evolution rate of the values. A *PC* laptop, equipped with a wireless *Ethernet* network board exchanges information back and forth between the base and the remote system.

Figure 2- Picture of our mobile manipulator

4.2 The base

We have first used a *Silicon Graphics Indigo*^{TM} connected to the local *Ethernet* network. We have implemented in this machine a virtual environment built up from the kinematic models of the mobile robot, including the manipulator and the environment.
The operator interface consists of a 3D perspective representation of the vehicle simulated in a structured environment. The operator handles a 6 d.o.f. mouse (? *a space mouse* ?) to move either the vehicle or the manipulator. The motions are analyzed and transmitted to the remote system.
A *TCP* client is connected to the remote system server and translates data to and from the interface. A wireless router makes the connection between the wireless network and the lab *Ethernet* network.

4.3 Results of the preliminary experiments

The various experiments we have carried out have led to a model of the transmission line and of the remote system. As long as the controller works properly, we can model it as a monodimensional 2nd order discrete filter $H_r(z)$ without making strong assumptions.

$H_r(z)$ = ***Erreur !*** (sampling period = *20ms*)

Because of the low distance between our base and our remote system (less than 300 meters), the round-trip delay statistical law looks like a *Poisson* one which usually rules the probabilities of arriving at a queue.
If we had a longer distance, we would probably have had a Gaussian distribution centered around the mean round trip delay [9].

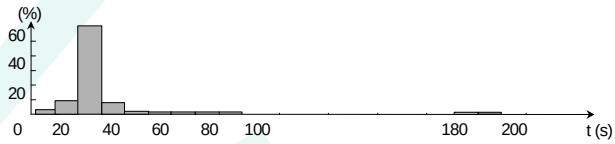

Figure 3 – Delay distribution

4.4 Improving the experimentation platform

In our experiments, the information perception from the vehicle can be improved by the adjunction of different transducers as accelerometers, magnetic compass, force sensors, CCD cameras, laser telemeters and ultrasonic sensors. The simulator will then be more accurate since it will receive richer information and the operator will be able to view simultaneously both the simulation and the reality on his screen.

5. SIMULATIONS

5.1 Organization

Consequently to the results of section 4.3, we have used a discrete monodimensional system with a transmission sampling period T_e of 500ms except for the *remote system* internal model. Globally, the system has 4 parts as showed in Figure 4.

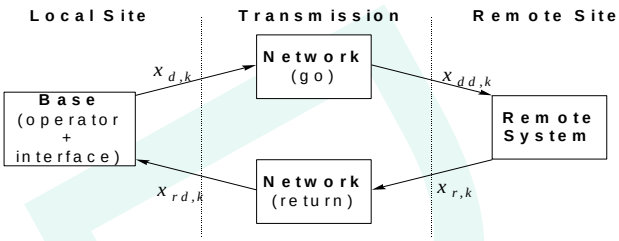

Figure 4 - General organization of the simulation

At time $t = k.T_e$, $x_{d,k}$ stands for *"desired signal"*, $x_{dd,k}$ for *"delayed desired signal"*, $x_{r,k}$ for *"return signal"* and $x_{rd,k}$ for *"delayed return signal"*.

5.2 Basic simulation models

?? *Remote system model*

We have used the generic model derived in section 4.3, assuming that the controller perfectly works.

$$H_r(z) = \textbf{Erreur !} \text{ (sampling period = } 20ms\text{)}$$

In order to make the model a little more realistic, we have added a random signal generator that produces noise at the output of the low-pass filter (visible in Figure 5).

$$x_{r,k} = b_1.x_{r,k-1} + b_2.x_{r,k-2} + a_1.x_{dd,k-1} + a_2.x_{dd,k-2} + ?_k$$

where $?_k$ is a flat random discrete function of which amplitude is about 3% of the amplitude of $x_{r,k}$.

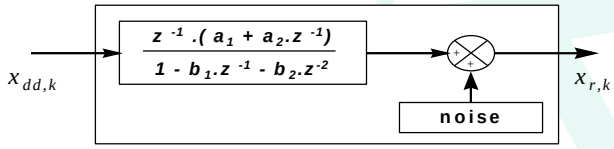

Figure 5 - Remote system simulation model

The response of the remote system simulation model to step values is showed in Figure 6.

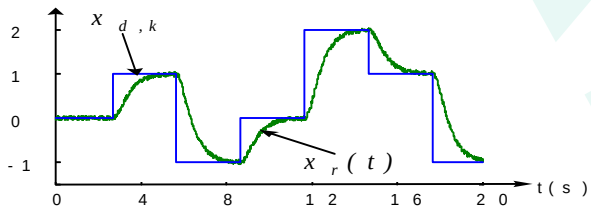

Figure 6 – Step responses of the mobile manipulator simulation model

?? *Telecommunication model*

Two identical blocks (one is shown in Figure 7) delay the signals going to and coming from the remote system. Every delay is chosen according to a statistical law (currently *Poisson*, given the mean delay <? ?>) and so that the samples go out of the network block in the same order as they went in (because the *TCP* network protocol used for the experiments ensures that the packets of data arrive in the correct order). A *10* second round-trip delay is chosen as reference.

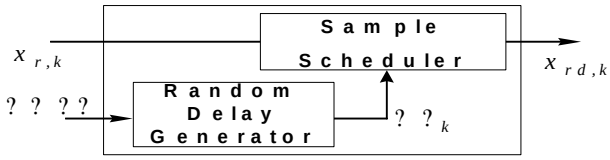

Figure 7 - Network simulation model

(depicted here for return signal)

Figure 8 shows how this block acts on the data going through it. $x_r(t)$ points out the data going out from the remote system block. It is sampled at a constant period of *500ms*. $x_{rd,k}$ shows the data going out of the network block. We can easily see that this latter signal has a varying sampling period and the signal does not even represent a delayed sampled version of the first signal.

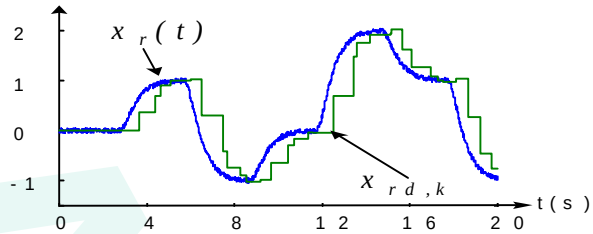

Figure 8 - Network block effect on return signal $x_{r,k}$

?? *Base simulation model*

On one hand, the base simulates the action of the operator by sending periodic sequences of values. These values are sampled at a rate of *500 ms*: $x_{d,k}$.

On the other hand, it retrieves the values sent back by the remote system ($x_{rd,k}$). Figure 9 shows the periodic sequences and the data back from the remote system after passing through the return network block.

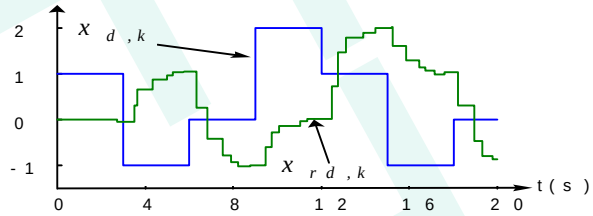

Figure 9- Desired signal and return signal

5.3 Conclusion

This model allowed us to emphasize some problems about long distance teleoperation.

The main impediment is the varying of delays that distort the data from and to the remote system. Also, the fact that

the operator watches the response to his action a long time after the action disturbs him a lot.

5.4 Improving the teleoperation diagram

?? *Delay variation compensation*

In order to eliminate these variations, we added a block which stacks incoming samples as they asynchronously arrive. This block is located at the input of the *remote system* and at the input of the *base*.

Meanwhile it unstacks these samples (in the same order : *First-In, First-Out*) with their very initial constant sampling rate.

The signal finds again the initial shape it had before being distorted by the network block; it differs just from a constant delay corresponding to the number of samples stacked in the block. This number of samples is to be calculated according to the difference between the maximum delay and the mean delay. If a very long delay occurs, the stack empties as long as there remains samples inside. When it is empty, an alarm signal is set.

This involves that the final delay is greater than the initial mean delay.

It is necessary to regularly watch for the initial mean delay and to adapt the size of the stack to it. The changes of size have to be signaled to the following blocks so that they do not use a distorted signal.

Figure 10 illustrates the effect of this block on return signals: delaying sampled $x_r(t)$ ($x_{r,k}$) gives $x_{rd,k}$ which becomes $x_{rc,k}$ (*c* as ?*corrected*?) through this block.

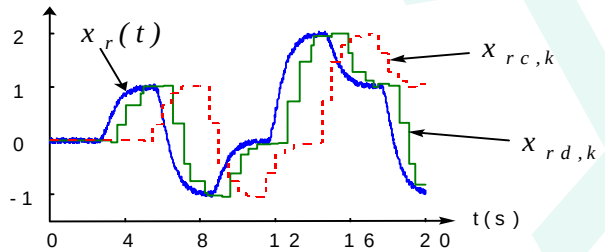

Figure 10 - Effects of the delay variation balancer

?? *Estimation of the delays*

Every sample sent from the *base* is fitted with its emission date. Therefore, it is easy to compute the round-trip delay by comparing the date of arrival of a sample with its date of departure. In a first approach, we have used a meaning filter with an horizon of 10 samples to estimate the next round-trip delay : *ERTD*. To estimate the go or return delay, we have to make the assumption that they are equal.

?? *Estimation of the state of the remote system*

The difficulty to estimate the state of the remote system is that we have to compare the result of our estimation and the real values. But in our case, the real values are available after the return communication delay. The solution we have provided is showed in Figure 11.

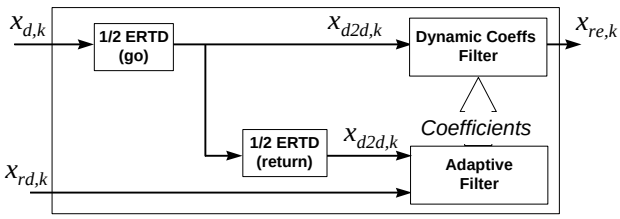

Figure 11 - Remote system state estimation

As the variations of the *ERTD* are slower than the sampling period, we assume that the delay block is a discrete linear function. This is why it is possible to swap it with the adaptive (discrete linear) filter.

The whole estimator features, on one hand, a discrete linear filter with dynamic coefficients of which recursive equation and transfer function are the following :

$$x_{re,k} = a_{1e}.x_{d1d,k-1} + a_{2e}.x_{d1d,k-2} + b_{1e}.x_{re,k-1}$$

$$H_a(z) = \textbf{Erreur != Erreur !}$$

where $x_{re,k}$ is the estimation of $x_{r,k}$ (introduced in Figure 4). a_{1e}, a_{2e} and b_{1e}. are the latest estimated coefficients, and $x_{d1d,k}$ is the *desired signal* $x_{d,k}$ delayed by ½ *ERTD*.

On the other hand, a normalized 1st order adaptive filter deduces the more appropriate coefficients a_{1e}, a_{2e} and b_{1e} by internally comparing the computation of $x_{d2d,k}$ through $H_a(z)$ with $x_{rd,k}$. The process is detailed in Figure 12.

The coefficients are computed as follows:

$$?_k = x_{rd,k} - x_{rde,k}$$

$$a_{1e,k} = a_{1e,k-1} + ?_k * ?_k * e_{k-1}$$
$$a_{2e,k} = a_{2e,k-1} + ?_k * ?_k * e_{k-2}$$
$$b_{1e,k} = b_{1e,k-1} + ?_k * ?_k * x_{rde,k-1}$$

$?_k$ is a dynamic parameter; the *Normalization* block computes the sum of the squares of *N* last samples and divides the initial $?_i$ by this sum (approaching the squared mean power).

$$?_k = \textbf{\textit{Erreur !}}$$

A loop checks the stability of the new coefficients and in case of instability, it re-computes up to *m* times the coefficients with smaller and smaller $?_k$. A signal warns if the computation fails to get stable coefficients.

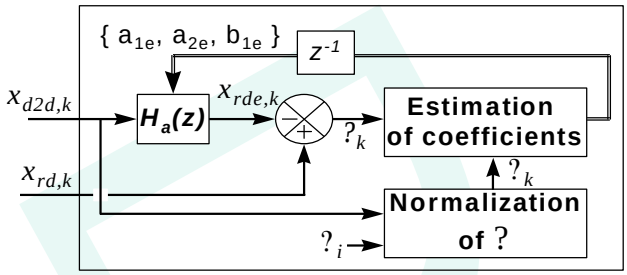

Figure 12 - Details of the Adaptive Filter Block

The *Dynamic Coefficients Filter* block (Figure 11) retrieves the estimation ($x_{rde,k}$) of the signal arriving to the remote site ($x_{rd,k}$) by delaying $x_{d,k}$ with half the round-trip time. It computes it with the last estimated coefficients. Whenever this block detects unstable coefficients, the output signal is set to zero.

According to Figure 11, in case of a 10s round-trip delay, the operator will be disturbed by the 5s reaction delay of the simulated model. We have put it this way in order to compare directly $x_{r,k}$ and $x_{re,k}$; its next position will be directly after $x_{d,k}$ so that the simulated model instantaneously reacts to the operator commands (it is a pre-estimation because at this time the remote system has not received this order yet).

Figure 13 shows that the estimation during permanent pattern is reliable, but the transient period exhibits several problems:

✍ 0 to 10s: the input signal for the *estimator* block equals 0; the adaptive filter cannot adapt itself with constant null values.

✍ 10 to 20s: the coefficients are stable but varying very quickly. It is due to the normalization block which is not yet in steady state because of the last 0. The result is a too high ? that makes the coefficients evolve too quickly. Figure 14 shows the evolution of the coefficients.

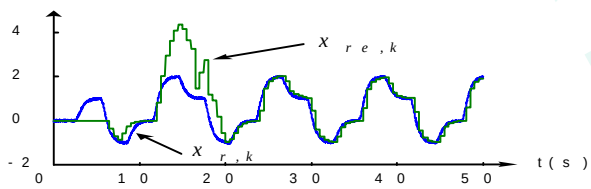

Figure 13 - Remote system real state and estimation

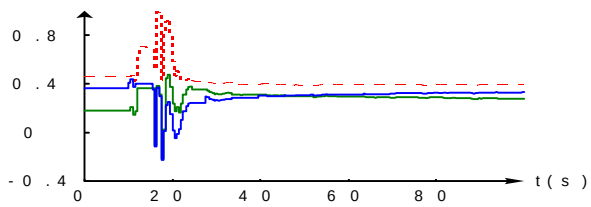

Figure 14 - Evolution of the estimated coefficients

6. CONCLUSION

Preliminary teleoperation experiments with a terrestrial mobile manipulator and the simulation model built from the results of these experiments have allowed us to find some approaches to improve the teleoperation control scheme of robots through the Internet. We are now preparing experiments to validate these approaches. These future experiments will also allow us to refine thee solutions in the case of our terrestrial mobile manipulator. We expect to be able to present some of these real experiments at the conference.

7. REFERENCES

[1] C. P. Sayers, D. R. Yoerger, R. P. Paul and J. S. Lisiewicz, *"A Manipulator Work Package for Teleoperation from Unmanned Untethered Vehicles"*, Proc. of the IARP Workshop on Subsea Robotics, Toulon, France, 1996.

[2] G. Hirzinger, K. Landzettel, CH. Fagerer, *"Telerobotics with large time delays - the ROTEX experience"*, Proc. of the Intl. Conf. on Intelligent Robots and Systems (IROS' 94), Munich, Germany, 1994, pp. 571-578.

[3] K. Taylor and B. Dalton, *"Issues in Internet Telerobotics"*, Proc. of the Intl. Conf. on Field & Service Robotics (FSR' 97), Canberra, Australia, 1997.

[4] M. Otsuka, N. Matsumoto, T. Idogaki, K. Koguse, T. Itoh *"Bilateral Telemanipulator System with Communication Time Delay Based on Force-Sum-Driven Virtual Internal Models"*, Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'95), Nagoya, Aichi, Japan, 1995, pp. 344-350.

[5] P. K. Pool & D. H. Ballard *"Remote Teleassistance"*, Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'95), Nagoya, Aichi, Japan, 1995, pp. 944-949.

[6] H. Turchi, A. Crosnier, P. Fraisse, *"Realtime Environment for Mission Programming of Telerobotics Systems"*, Proc. of the SPIE Conf. on Intelligent Systems, Pittsburgh, USA, 1997, pp. 22-24.

[7] A. Rastogi & P. Migram *"Augmented Telerobotic Control"*, University of Toronto, Canada.

[8] A. Lelevé, P. Fraisse, A. Crosnier, P. Dauchez & F. Pierrot, *"Towards Virtual Control of Mobile Manipulators"*, Proc. of the 3rd World Automation Congress (WAC'98), Anchorage, USA, 1998.

[9] R. Oboe and P. Fiorini, *"Issues on Internet-based Teleoperation"*, Proc. of the 5th IFAC Symp. On Robot Control (SY.RO.CO.), Nantes, France, 1997, pp 611-617.

DRAFT

DRAFT