Real Time Monitoring in Web Telerobotics

IONUT RESCEANU, MARIUS NICULESCU, CRISTINA PANA
Department of Mechatronics
University of Craiova
Address Bvd. Decebal Nr. 107
ROMANIA
resceanu@robotics.ucv.ro, toros@rdscv.ro, cristina@robotics.ucv.ro

Abstract: - There is a growing need for humans to perform complex remote operations and to extend the intelligence and experience of experts to distant applications. A blending of human intelligence, modern information technology, remote control, and intelligent autonomous systems is required. Telerobotics goes beyond autonomous control in that it blends in human intelligence and action as appropriate. It goes beyond teleoperation in that it incorporates as much autonomy as is possible or reasonable. A new approach for solving one of the fundamental problems facing teleautonomous systems is discussed: the need to overcome time delays due to telemetry and signal propagation. Optimized maintenance can reduce operating costs by demanding less resources and avoiding equipment breakdown, hence ensuring that daily manufacturing and production will not be interrupted. Web-based maintenance is one of the most efficient methods for optimizing maintenance. The Web-based monitoring system discussed provides remote sensing, monitoring, and on-line fault diagnosis for equipment, together with a collaborative maintenance platform for international experts to interactively share their experiences in maintenance.

Key-Words: - **Telerobotics, Web, Internet, Control, TCP/IP, UDP, Real-Time, Monitoring**

1 Introduction

In order to overcome the limitations of net-bandwidth in real-time monitoring tasks several technologies can be used.

In a remote monitoring process, the signal data obtained from real machines are transferred to and processed at a remote host in the network through long distances. In a complex monitoring system, in order to make a precise prediction, large volumes of signals are measured and transferred. In reality, however, the network transportation speed is not satisfactory. Due to the latency, congestion, and instability of network transfers occuring either in critical real-time systems or if the machines that have encountered some emergencies are located far away from the monitoring system, commands from the monitoring system might not be transmitted to the remote machines within the required time periods.

Teleoperation is the remote control of robot manipulators. Although commands can be sent from user to remote robot at different levels of abstraction, this section describes a kind of low-level teleoperation in which the human directly controls the motions and contact forces of the remote manipulator in real time. Perhaps the most common application of this technique is in construction equipment such as excavators in which the operator controls the velocity of the joints of the "robot" to accomplish the task. However construction equipment does not provide force feedback directly to the hand. When the user is located farther from the remote robot, considerable engineering effort must be applied to reproduce the sensory feedback information which allows accurate and efficient control. Both teleoperation and virtual environments require this rich and self-consistent sensory feedback. Haptic feedback devices were pioneered in teleoperation systems as far back as the 1940's. In both teleoperation and virtual environment applications of haptics, a loop is closed between the human operator's motion "inputs" and forces applied by the haptic device. In teleoperation this loop is closed via a communication link, robot manipulator, and the environment. In virtual environments, the loop is closed via a computer simulation.

Key issues for the advancement of teleoperation technology include:

Performance Evaluation: What quantitative measures can be developed with which to quantify the quality of a teleoperation system (including haptic displays)? Control: How can stable, high performance, control be obtained in spite of highly variable human operator and environment dynamics, time delays in communication channel, and kinematic effects such as singularities?

Scaling: What are the requirements for effective user interfaces when there is a large difference in scale (either up or down) between the master (human operator) and slave (remote robot)?

Mechanization: High quality teleoperation and haptic interaction depends critically on advanced mechanism designs for both master and slave sides. Key issues are light stiff structures and linkages, actuators with high torque/mass ratios and high linearity, compact, high resolution sensors for position velocity and force/torque.

Kinematics of Teleoperation: How can effective use be made of redundant degrees of freedom in teleoperation systems (i.e. when the number of slave DOF > number of master DOF)?

The field of telerobotics grew out of the need to perform operations where it is difficult to place a human being, due to constraints such as cost, safety or time. Telerobotic systems need to be able to perform tasks that a human would normally do. Due to limitations in robot autonomy, this often has to be achieved by using human operators to control the remote robot (via a communication link). Such a system is a telerobot. The human operator is responsible for high level control such as planning and perception, while the robot performs the low level instructions at the remote site.

An aspect of web telerobot systems that affects the choice of control scheme is time delay. Shared continuous control is less sensitive to these problems and has been demonstrated over the Internet, but only on short, high bandwidth Internet links. Discrete command control schemes and above are free of any time delay based instability problems as all closed loop control is performed locally. They are therefore the most appropriate choice for web telerobotic systems.

In order to overcome the limitations of net-bandwidth in real-time monitoring tasks, mobile agent technology can be used. Within the scope of the work described in current research on mobile agent's technology, a mobile agent can be defined as ''a software agent that is not bound to the system where it begins execution.'' A monitoring program can be dispatched as a mobile agent to the host that has sent the requirements to request for this monitoring program.

Operational failures of equipment may not only lead to a loss of production, but also, in some serious situations, may cause human casualties. Hence, equipment condition monitoring and fault diagnosis are often employed in maintenance to prevent operational failure of services and the fatal breakdown of manufacturing equipment.

The technology for equipment fault diagnosis has been developed in response to the demands of modern industry that has been concerned with such aspects as human safety, economic productivity and effectiveness. Machine fault diagnosis refers to the process of identifying a machine's operating condition and investigating its possible source of fault [1]. This is usually done in a way that is similar to medical diagnosis. Both of the diagnoses have to observe symptoms by sensing and analyzing signals collected either from a human body or a piece of equipment.

For machine-based fault diagnosis, the collected signals may be operating force, vibration, temperature, voltage, pressure, etc., any of which could be related to the inherent conditions of the machine [1]. Good diagnostic systems can provide managers and operators with the necessary information to determine the running condition of the equipment.

The concept of Web-based remote monitoring and a collaborative diagnostic system was initially proposed for medical care in 1988.

The system was designed to enable doctors or small clinics in rural areas to obtain instantaneous consultations from specialists in urban hospitals [2]. It was adopted a similar concept for industry and it was [4] laid out a framework for remote diagnostics and maintenance for manufacturing equipment. It has become a standard for others to follow. An ideal remote diagnostic system should include video-conferencing and remote measurements delivered as close to real-time as possible. It should also provide on-line fault diagnosis and support a multi-user kind of collaborative consultation. Hence, the system must install an operating system that supports a multi-tasking and multi-user operating environment. An eligible operating system must support distributive computing, cope with common servers and major communication protocols, and be easy to adapt to a variety of popular Web browser applets.

The Web allows universal access by having independent connectivity for different kinds of platforms using open standards for publishing (HTML, XML), messaging (HTTP), and networking (TCP/IP) [5]. Internet browser plug-ins should be able to handle new data types and allow different applets to be downloaded and run on any browser. A variety of software-based VIs should exist for performing the required signal processing, features extraction, and analysis. These VIs are, preferably, to be compatible with ActiveX standards. Therefore, Web-enabled VIs can be operated as browser applets/ActiveX in a multi-user environment.

Since the Web enables multi-media support, both interactivity and extensibility [6], it can seamlessly include new forms of content and media [7]. Therefore, Web-based maintenance should employ multi-media to a large extent. Nowadays, broad bandwidth communication is available in many countries for use by Web-based multi-media. The developments in database and object technologies, such as CORBA, IIOP and component-ware concepts, enable users to connect to back-end databases and legacy applications via user-friendly Web interfaces [8]. All of these features make the development of Web-based maintenance and an interactive type of collaborative maintenance feasible.

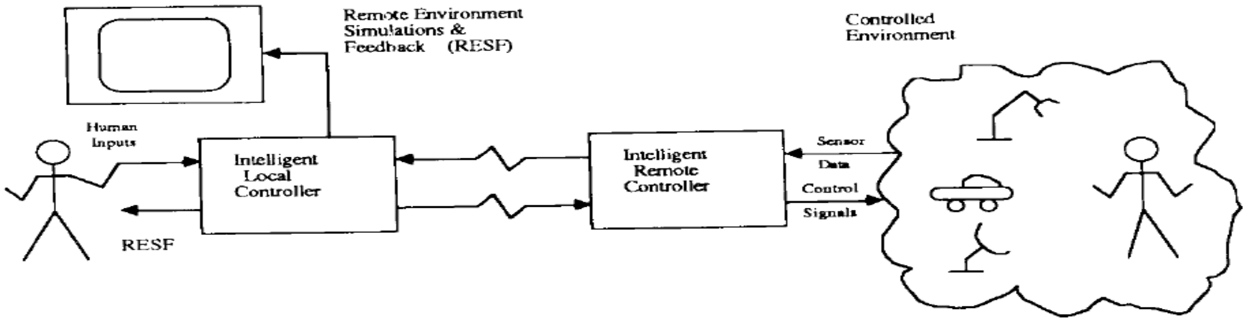
To ensure future compatibility and allow for expansion of the sensors that will be used on the Internet, the standard for smart transducers, IEEE 1451.2, has been formulated for the design of next generation Web-ready sensors. The future smart transducer will have a built-in Ethernet module and support direct plug-and-play on the Internet without the need for a connection to a PC or having a separate Ethernet card, as is the case with today's systems. Sensor manufacturers, such as Hewlett Packard and Bruel & Kjaer, have already proclaimed that their new directions in designing sensors are based on the standard for smart transducers [9]. With the help of these Web-ready sensors, Web based maintenance becomes easier to implement, using fewer resources and involving less capital cost.

Even though the research on Web-based services and systems is advancing every day, progress in research on Web-based maintenance is lagging behind. As has already been mentioned, the three crucial functions for Web-based maintenance are, remote data sensing with the help of the mini-server, signal processing and fault diagnosis using Web-enabled VIs, and collaborative maintenance platforms for multiple users. Research has been conducted to target one or two of the three functions. However, there is a deficiency in developing a Web-based maintenance system that has all three important functions combined together.

2 Overview of a Teleoperation System

Fig. 1 shows, at a conceptual level, the structure of a single local-remote pair in a basic teleautonomous system. The spatial reference frame is taken to be that of a human controller at left, that is, the controlled environment is remote.

Fig. 1 Basic components of a Teleoperation System



The controlled environment can include humans or any manner of device or both. The remote intelligent controller receives data from multiple sensors and provides outputs encompassing anything from servo-level control signals to a robot joint, to video signals, to a heads-up display worn by a remote human.

The inputs on the local side of the system may be any form of input control by the human from simple joystick control to complex cockpits with many inputs to discrete commands for the remote controller to perform complex tasks. The local display represents any kind of feedback to the human about the remote environment. This will include both simulated information and actual feedback signals and may be composed of television images, complex graphics, force reflection on input devices, or even high-speed data analysis. The distance between local and remote sites can produce substantial time delays in the signal transmission between them.

Telerobotic control of even a single local-remote controller pair provides many operating modes, including the following :

1) Direct continuous teleoperator control of a remote device. The remote controller merely follows its inputs. This is currently the most common form of operation.

2) Shared continuous teleoperator control of a remote device. The remote controller performs higher level control than position servoing. For example, it might treat received inputs as being

relative to an object to be manipulated and perform appropriate transformations before following them [17], or it might treat received inputs as a nominal path and perform some local sensing and replanning to reach the goals of the nominal plan.

3) Discrete command control by the human operator of the remote device. This implies a higher level of capability in the remote portion of the controller, which can vary from simple set-point control of a number of satellite antenna positioning servos to complex task analysis, planning, and execution. At this level, the commands become highly task specific, though the lower level primitives utilized may be more generic.

4) Supervisory control. The remote device operates in a largely autonomous mode and only interacts with the human when it encounters a situation it cannot handle, that is, management by exception, or a situation in which the human notices an opportunity to improve performance, that is, opportunistic management. It differs from the discrete command mode principally in the frequency of interaction with the human controller and the philosophy of being largely autonomous.

One local human operator might supervise a fleet of remote devices.

5) Learning control. The remote controller is given an intelligence that allows it to learn from human inputs and sensor information and subsequently to generate behavior in similar situations without human intervention.

6) Guidance of remote nonexpert humans by local experts.

In this mode, a variety of media, such as visual displays, graphics, touching, and pointing, are used to achieve a collaboration between the local expert and the remote non expert. Groups of such basic systems, possibly with local controllers in different locations, will make up larger scale teleautonomous systems. Many kinds of interactions will be possible, from handoffs of control between different local control agents (even if in different physical locations) to shared cooperative action of the remote devices.

We present here a sequence of interface control concepts that collectively underlie efficient control of manipulation tasks and also enable simple protocols for an exchange of such tasks among control agents.

Fully general teleautonomous systems with all of the capabilities described in the previous sections do not yet exist and will be the subject of considerable future research. In this section, we present a conceptual overview of basic human interface and system architectural concepts that we believe to be suitable for incorporation into almost any general teleautonomous system. We introduce the specific teleautonomous problems that these concepts address as well as methods for measuring the effectiveness of their implementations.

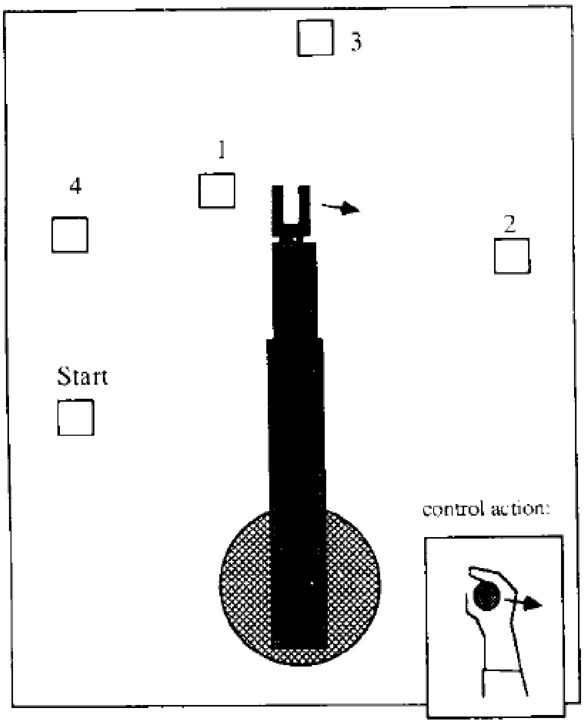


Fig.2 Visualizing **a** remote manipulation task

One of the most fundamental problems facing teleautonomous systems is time delay due to telemetry and signal propagation delays. Even modest time delays have long been known to cause instabilities in control systems such as robots [181. In addition, the time delays present in space applications are anything but modest. They are currently handled by a very inefficient "move a little and wait" mode of operation.

To provide a measure for the effectiver.ess of the control methods we develop, we introduce a simple experiment similar to accepted tests for human performance on direct manipulation tasks. Suppose that we are looking via video link over the shoulder of a telerobotic manipulator and controlling the manipulator via a joystick, as shown in Fig. 2. We are to perform the simple task of touching in sequence each of a series of boxes.

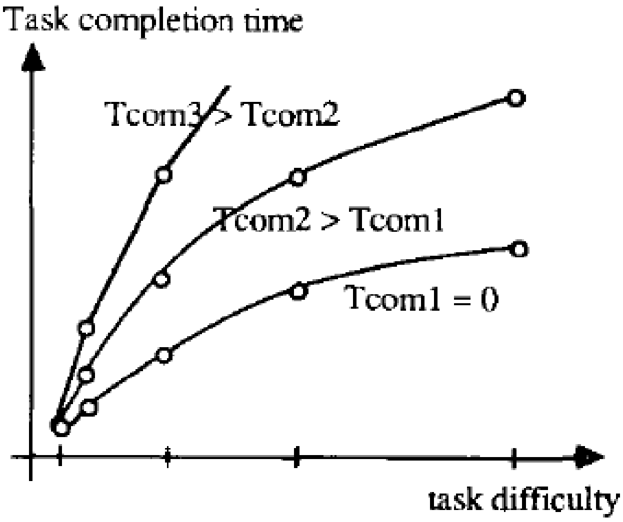


Fig 3. Task completion time as a function of task difficulty and communication delay *T,,,,* .

This is a standard test of human manipulation capability, and the task's difficulty, that is, the time to complete the task, has been shown to be a function of the ratio of the distance between consecutive boxes and the sizes of the boxes [19]. The difficulty can be varied easily, and we can undertake various trials of performance as a function of system parameters. For example, we could do some simple trials to see whether the time to complete the task is a logarithmic function of the ratio *D/S,* as in Fitts' law [19].

To demonstrate the consequences of time delay, we visualize the same manipulation experiment with a time delay inserted into the communications path. We find that the telerobot's motions then tend to be rather slow and jerky. The operator must move a little and then wait through the time delay to see what happened. The difficulties introduced by the time delay are quite noticeable, and the task completion time may be greatly extended as a function of the delay, as shown in Fig. *3*.

Coping with Time Delay

Over the past several decades, there have been numerous attempts to overcome the difficulties of time delay in control systems. In one way or another, all approaches have been based upon some form of prediction with respect to the time delayed sensed signal. At the servo level, feed forward control that does not encounter the time delay has been used. Kelly [20] found that a predictive display of one or more system variables under operator control was useful. It was noted that substantial improvement in submarine operation had been achieved through prediction. Bernotat and Widlok describe the use of mechanical prediction [21]. Another use of predictor displays was for orbital rendezvous [22]. A more recent study of similar nature [23] has shown that performance degradation due to time delays in rendezvous and docking maneuvers can be reduced via use of prediction.

In an attempt at overcoming time delays in controlling remote manipulators, Ferrell and Sheridan [24] built a local manipulator that has similar dynamics as the remote manipulator.

The operator controlled the local manipulator, where no time delays were present, with the control signals being sent to the remote manipulators as well.

From the telemanipulator perspective, remote operations have been studied for years, with little emphasis on time delays, except for the predictive manipulator cited before. More recently, it was extended the predictive manipulator idea by using a predictive display. The operator controls a local simulation of the telerobot with the control signals then sent in parallel to the simulation and the remote telerobot. The simulation is then displayed superimposed over the return video. In this way, the operator can "see" the effects of the control immediately without having to fully wait for the return signal from the telerobot .

3 Real-time machine monitoring

Neural networks and expert systems techniques are finding increasing application in the field of machine tool diagnostics. Hu et al. [7] proposed an intelligent diagnostic system based on a combination of the neural networks and expert systems techniques. The integration of the two artificial intelligence techniques takes full advantage of the low-level processing capability of neural networks and the high-level processing capability of expert systems. This combined technique is particularly suitable for real-time machine fault diagnosis.

A diagnostic system based on this combined technique has been developed and good results have been achieved.

It was developed a tool monitoring algorithm that can detect changes in the cutting process which may arise due to a broken tooth. The approach was shown to be able to distinguish between entry or exit transient conditions and actual tooth breakage. The limiting factor to the use of this method relates to the establishment of thresholds against which signals can be measured in order to identify tooth breakage. Such thresholds need to be sensitive to tool breakage, whilst remaining robust to process variations. However, they are not universally applicable across all cutting conditions.

3.1 Mobile agents applications

The notion of mobile agents was established in 1994 with the release of a white paper that described a computational environment known as ‘‘Telescript’’ [10]. In this environment, running programs were able to transport themselves from host to host in a computer network [11]. The ability to travel allows a mobile agent both to move to a system that contains an object with which the agent wants to interact and to take advantage of being in the same host or network as the object.

Typical mobile agents applications involve the following:

– Workflow management. A typical organization usually has its own workflow processes defining business activities and the overall control flow of the work.

Mobile agents have process information embedded within themselves in order to find a target destination once an activity has been initiated; hence, they can provide an elegant way of executing processes across spatial and temporal fields and offer suitable coordination structures for a workflow management system [12, 13].

– Electronic commerce. Electronic commerce is another application of the mobile agent technology [14]. A buyer agent could do shopping for a user, including making orders, and potentially even paying. Electronic commerce can also take place between agents.

A buyer agent can be given knowledge of a user’s preferences and sent to the dedicated host, where it would mingle and haggle with seller agents. If a potential match was found, the buyer agent could report back to the user, or potentially consummate the deal on behalf of the user.

– Parallel processing. Given that mobile agents can move from node to node and spawn sub-agents, one potential use of mobile agent technology is as a way to administer a parallel processing job. If a computation requires so much CPU time as to require allocation across multiple processors, an infrastructure of mobile agent hosts could be an easy way distribute the processes [14].

– Information retrieval. Information retrieval in a heterogeneous distributed software environment is another area where mobile agents can be employed [15]. A searching agent could be sent to a site, locally analyze the stored documents, and follow the interesting links by cloning itself. Mobile agents can avoid the transfer of a large amount of data over the network and even operate in situations of discontinuous network connections. This is made possible by their exploiting the availability of the connection to move themselves to the information site and, after elaboration in the remote site, being able to come back when the connection is available again.

– Database access. Due to the characteristics of agents, it is quite feasible for agents to interact with large databases distributed over several nodes in a network.

When an agent is immobile as in ‘‘standard’’ client/server architectures, this may give rise to substantial data-traffic. Therefore, when an agent is relatively small, it is more effective to transport the agent instead of the data [16]. In this way, mobile agents can be effective in some time critical systems.

– Management of distributed resources. By their nature, mobile agents are inherently distributed. As such, they must be executable across a variety of platforms and operating systems to achieve their full potential.

Mobile agents can be seen as effective alternatives to traditional client/server architectures for distributed systems [17]. Mobile agent technology offers important advantages, such as flexibility and scalability of the system, load balancing, on-demand service, and low traffic in the network. These advantages are due to the way in which mobile agents treat distribution problems by using local interaction and mobile logic.

Mobile agents are best suited to those applications that are characterized by asynchronous transactions, low bandwidth, high latency, remote information retrieval, multi-processing, or distributed task processing features [17]. However, most of the mobile agent applications have so far been concentrated on network management, mobile computing, and personal assistants, but rarely on manufacturing or machining condition monitoring.

4 Transmission Delay

Transmission delay, sometimes also termed as latency, of a communication line is the time from the start of data packet transmission at the source to the start of data packet reception at the destination. The source of the latency can vary from the speed of the signal to how the signal is relayed among various gateways. In many instances this delay can become significant enough to become noticeable by a human.

Dedicated algorithms are specially designed for and tested with the Internet as the communication medium, although in principle they are applicable to any sort of transmission delay. The Internet is a complex network of servers and clients where data transmission is not direct but is forwarded over many links via many gateways. This can produce significant latency especially at certain times of the day with heavy network congestion and in areas with poor network infrastructure. The unpredictability of the Internet can result in variations in latency as well as lost data packets. To perform bilateral teleoperation over the Internet, the system must solve the problems posed by these pure delays in what is effectively a closed-loop system.

4.1. Internet Delays

Propagation delay is unavoidable in a tele-operated system (due to the limitations of the speed of electricity, the speed of light). Preliminary studies have shown that a transmission delay of 200mS or more leads to problems with surgical accuracy and precision.

4.1.1 Fixed Delay

Propagation delay is unavoidable in a tele-operated system (due to the limitations of the speed of electricity, the speed of light). Preliminary studies have shown that a transmission delay of 200mS or more leads to problems with surgical accuracy and precision.

The International Telecommunication Union Telecommunications Standardization Sector (ITU-T) noted that with voice calls, most callers notice round trip delays when they exceed 250mS. As a result the ITU-T G.114 recommend the maximum desired one way latency to achieve high quality voice is 150mS. With Round Trip Time (RTT) delays of 500mS or more, a ‘natural’ phone conversation becomes very difficult. These figures for voice traffic will be used as an initial starting guide for our implementation of timely and realistic force feedback.

4.1.2 Random Time Varying Delay

The Internet is a best effort service that offers no upper bound to response time or bandwidth guarantees. The result is a service that is time varying in a random nature. This fact introduces an extra level of complexity in the teleoperation of a system. A control engineer can deal with the problem of compensating of a constant delay with relative ease. However, a random time varying delay is very difficult to compensate for. Such a situation can often result in destabilising the overall system. The key to timely and stable control of a closed loop system over the Internet is to effectively reduce the variance of the delay.

The problem of controlling a real time tele-system using the Internet as the link has been studied extensively over the past few years. Most researchers have tended to use TCP/IP (with its inherent short comings in ability to deliver data in a timely fashion), seemingly without firstly looking deeply into the IP protocols options available.

Generally this past research has concentrated on a variety of complex control methods in order help stabilise a telecontrol system in the presence of TCP/IP delay. Essentially the results have traded off a large amount of system response (delays of 5-6 seconds are not uncommon) in order to achieve stability.

4.2 TCP/IP Delays

TCP is a connection orientated protocol. Possible congestion due to TCP traffic flows is controlled by the congestion control mechanism that is native to TCP. This congestion control can inflict serious problems on real time applications. In addition to this, TCP has an error correction arrangement in the forms of:

- Ordered delivery
- Duplication detection
- Crash recovery
- Retransmission strategy

By TCP addressing these above issues, TCP offers a guarantee for the reliable transport of packets to destination, thus, shielding the data users from the unreliable nature of the underlying IP network. The downside is the fact that these flow and error control techniques employed by TCP present a major obstacle to achieving time guarantees over the Internet. For example, the TCP slow start mechanism is used to discover the channel throughput during the initial connection setup and for resumption of a broken connection. This is done by first sending a packet across the channel and waiting for a response. If a response is received, the next packet is sent a bit faster. This procedure is repeated until the speed of the link is discovered. With the half-second delay between responses, throughput is significantly slowed.

Since this process can take 7-15 Round Trip Times, for a link with a propagation delay of 500ms this can mean that for 3-7 seconds, the link is underutilised. (See figure 1).

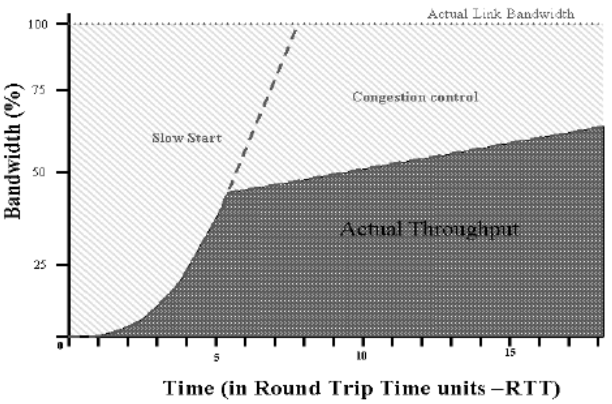


Fig 4. TCP/IP True Throughput

To further add to overhead, every TCP connection is established by a "3-Way Handshake" between the Receiver and Sender. On links with long propagation delays, this fixed overhead means that even very short data exchanges take at least a few seconds to be completed. Data links can be noisy, and this has profound effects on the performance of TCP/IP throughput because the slow start congestion control mechanism wrongly detects the noise as network congestion. Hence, from a real time viewpoint, TCP fails to provide an adequate solution, largely due to the enormous processing overhead it employs in order to provide a reliable path for data.

4.3 UDP/IP and RTP/UDP/IP Delays

Seemingly, none of the present online tele-operated systems to date have used Real Time Protocol (RTP) running over User Datagram Protocol/ Internet Protocol (UDP/IP). This is probably largely due to the fact UPD/IP is seen an unreliable data medium, whereby data could arrive out of order or not at all. Even so, RTP/UDP/IP is fast becoming the popular protocol arrangement for streaming data in real time over the Internet. UDP is a connectionless protocol. This fact gives it very different characteristics to TCP. UDP is an unreliable service due to the fact that delivery and duplication of packets cannot be guaranteed. In addition it is likely that packets will arrive at the destination out of order. Even so, UDP with RTP is a far better option than TCP for realtime applications such as voice or video. Retransmission of a packet 1-2 seconds after it was sent when it contains a 20mS sample (as is the common case for voice) would produce disastrous implications to the real-time voice stream. In addition the cost in time for TCP to detect a packet loss, stop the data stream, request a resend from the point of loss and then finally receive the lost packet can be in the order of several seconds. As stated, packet loss is unavoidable with UDP/IP, but it can be compensated for in voice streaming by codec loss-concealment schemes. One such codec is G.723.1, which has the ability to interpolate a lost frame by simulating the vocal characteristics of the previous frame and slowly damping the signal. It has been shown that packet loss rates up to the order of 10 percent have little noticeable impact on the audible quality of the speech. It should be noted also that the connectionless quality of UDP/IP reduces the overhead of the protocol (from TCP/IP 40bytes to UDP/IP 28bytes) and this makes UDP a further preferred choice for constant flow applications such as multimedia and control sessions. Even though a UDP/IP implementation has a lesser header overhead than that of TCP/IP, the RTP/UDP/IP implementation returns the header overhead back to 40bytes since the RTP component adds an additional 12bytes to the header. Now 40-45 bytes of overhead would not be an issue if the data packet were in the order of 1500 bytes. The problem is that our implementation only involves packets with a data size in the order of 10-20bytes (due to the sampling rate). Hence a whopping total of 40-45bytes of overhead to transmit a 10-20byte payload. There are two possible solutions to this problem:

1. Increase packet size, at the expense of sample rate and potential delay jitter.
2. Use header compression. In the case of voice packets it has been shown that the increased delay incurred from increasing the packet size is unacceptable. For this reason a great amount of research is being undertaken into optimizing header compression.

In summary, utilising UDP/IP in place of TCP/IP will greatly increase network efficacy by:

- Removing the need for having a connection setup before data can start to flow.
- Removing the slow ramping up of
- Low rate packet loss does not halt transmission of the streaming data.

In real time operations such as online gaming, some programmers would prefer to use user datagram protocol (UDP). This protocol eliminates the need for confirmation where the transmitting computer keeps sending the data packets with no regard as to whether the receiving computer has received the data. This means that all the data are sent in a timely fashion, an important feature for real time operations. But the lack of confirmation also means that it is less reliable. The transmission delay in UDP protocol is more stable than TCP.

Currently computers running on the internet use either the Transport Control Protocol (TCP/IP, where IP stands for Internet Protocol) or the User

Datagram Protocol (UDP). TCP provides a point-to-point channel for applications that require reliable communication. It is a higher-level protocol that manages to robustly string together data packets, sorting them and retransmitting them as necessary to reliably retransmit data.

Further, TCP/IP is confirmation based, meaning it transmits data and waits for confirmation from the other side. If not, it retransmits. With TCP/IP there is no data loss. The top of figure 1 shows the cross Atlantic round trip time delay between two sites, using TCP/IP protocol over a period of 100 seconds.

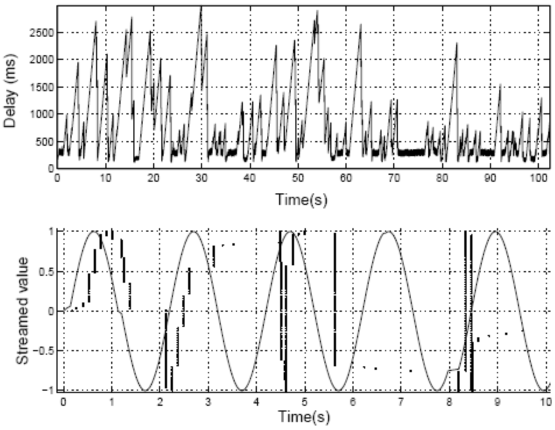


Figure 5. Top: The round trip time delay between two sites using TCP protocol. Bottom: Transmitted and received sine wave, sampled at 10 milliseconds.

Data points are generated 10 milliseconds apart. The experiment was carried out on a typical work day during mid after noon. The software was developed in Java 2 platform and ran on two PC running the windows NT operating system. It is apparent that the delay varies substantially, ranging from a minimum value of 100 milliseconds to as much as 3000 milliseconds [17].

The bottom portion of figure 1 shows the transmitted and received sine wave for a period of 4 seconds, sampled at 10 milliseconds intervals. Although no information is lost in TCP/IP based communication, it is evident from the figure that data sampled at different points in time gets lumped together along the way and arrives simultaneously at the destination. Hence the shape of the sine wave in not preserved. Thus making TCP/IP based communication rather unfavorable for real-time control.

The UDP protocol provides communication that is not guaranteed between two applications on the network. Unlike TCP/IP which is connection based, UDP is not. Rather, it sends independent packets of data, called Datagrams, from one application to another. In UDP based connections data is packed into packets called datagrams, addressed (like an envelope), and then transmitted. Much like sending a letter through mail, UDP is not confirmation based and the order of arrival is not guaranteed either. The over head of retransmitting data is eliminated, which comes at the expense of some data getting lost or not arriving at the destination at all. The top of Fig. 2 shows the round trip delay between the two sites, using UDP protocol. Again the sampling period was 10 milliseconds. Notice that the fluctuations are a lot less, ranging from a minimum value of 100 milliseconds to a maximum of 250 milliseconds with the average being 116 milliseconds.

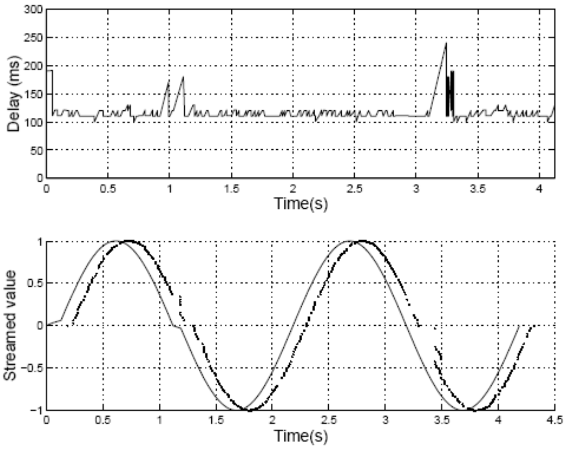


Figure 6. Top: Cross Atlantic round trip time delay between two sites using UDP protocol. Bottom: Transmitted and received sine wave, sampled at 10 milliseconds.

Although only 4 seconds of streaming results are shown, this trend was also confirmed over a much longer period of time. The bottom portion of figure 2 shows the transmitted and received sine wave over the same 4 second period. Notice this time the shape of the signal is preserved and the received wave closely tracks the transmitted wave with a lag time equaling the delay observed in the top graph.

However notice that some data arrives out of order as can be seen around the 3.5 second mark where the top graph shows a sudden spike in the delay. A few datagrams are also observed to arrive simultaneously and some 12 to 16 percent of information is lost along the way.

Given what is known about TCP/IP and UPD protocols, the protocol of choice for real-time control is UDP. This is because a consistent sample rate with lower fluctuations can be maintained with UDP. A feature essential in digital control. The problems stemming from data loss and data getting out of order during transmission will be addressed in this research.

There are many internet nodes between a local site and a remote site. Assume that the physical distance between the two sites is D. In order to examine the distribution of internet time delay, the local site and the remote site were set in the same computer and a reflector was provided at a place of distance D=2. The reflector returns the received data immediately. Internet time delay was measured in two cases. One was for a connection in the same domain, and the other was for a connection in the same country. The parameters for the measurements of the internet time delay were as follows: in the same domain, D =300 m, number of nodes=11, bandwidth(b_0,b_n)= D10 Mbps, and bandwidth ($b_{n/2O1}$,$b_{n/}$2)=10 Mbps, and in the same country, D=300 Km, number of nodes=29, bandwidth (b_0,b_n)= 10Mbps, and bandwidth ($b_{n/2O1}$,$b_{n/}$2)= 56 Kbps, where n is the number of nodes. Tables 1 and 2 show the regional and weekly variations of the delay measured every one minute for 24 hours each. According to the measurements, the internet time delay increases with distance, but the delay depends also on the number of nodes traversed. Also the delay strongly depends on the internet load so that it cannot be modeled for prediction [19].

TABLE 1
REGIONAL VARIATIONS OF THE
INTERNET TIME DELAY

	Mean	Standard dev.	Min. value	Max. value
Same domain	20 m sec	51.7	6 m sec	0.89 sec
Same country	4194 m sec	$2.25\cdot10^4$	39 m sec	343 m sec

TABLE 2
WEEKLY VARIATIONS OF THE
INTERNET TIME DELAY

	Mean	Standard dev.	Min. value	Max. value
Mon.	531 m sec	$1.42\cdot10^3$	48 m sec	15 sec
Tue.	6880 m sec	$2.89\cdot10^4$	45 m sec	343 sec
Wed.	1480 m sec	$7.77\cdot10^3$	44 m sec	154 sec
Thu.	1030 m sec	$4.16\cdot10^3$	45 m sec	68 sec
Fri.	779 m sec	$2.59\cdot10^3$	49 m sec	35 sec

The internet time delay is characterized by the processing speed of nodes, the load of nodes, the connection bandwidth, the amount of data, the transmission speed, etc. Especially the dominating factors are the processing speed and the load of nodes. The internet time delay T_d (k) can be described as follows:

$$T_d(k)=\sum_{i=0}^{n}\left[\frac{l_i}{C}+t_i^R+t_i^L(k)+\frac{M}{b_i}\right]=\sum_{i=0}^{n}\left[\frac{l_i}{C}+t_i^R+\frac{M}{b_i}\right]+ +\sum_{i=1}^{n}t_i^L(k)=d_N+d_L(k) \tag{1}$$

where l_i is the ith length of link, C the speed of light, t_i^R the routing speed of the ith node, $t_i^L(k)$ the delay caused by the ith node's load, M the amount of data, and b_i the bandwidth of the ith link. d_N is a term which is independent of time, and $d_L(k)$ is a time-dependent term. Because of the term $d_L(k)$ it is impossible to predict the internet time delay at every instant.

Since the internet time delay is affected by the number of nodes and the internet loads, it is variable and unpredictable. Also, a large internet time delay disturbs some control inputs. Fig. 3 shows the influence of the internet time delay on the control information. The received data at the remote site was distorted severely, and the information of the sine function y(t)=sin(0.2πt)+5, which was used as a test function, was almost lost. Fig. 4 shows the information loss of command signals when the time delay is T_d , where a command is given to the IPR via internet every sampling time T [18].

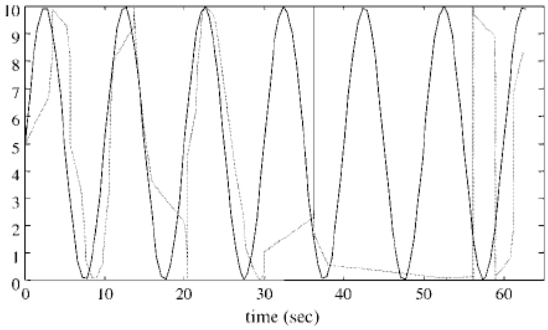


Fig. 7 Influence of the internet time delay.

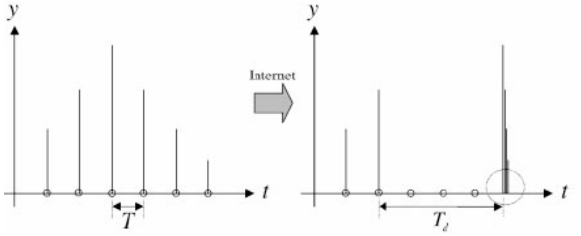


Fig. 8 Information loss of command signals.

N control commands input to the remote site at the same time after the delay T_d so that their informations is lost, where N=INTEGER(T_d/T) Thus, a novel internet control architecture is needed to control the IPR, which is insensitive to the internet time delay.

5 Control of web-based Telerobotics equipment

From the previous discussion, it is clear that an Internet-based control system must face the variable time delay and the packet losses introduced by the computer network. We are interested in evaluating the feasibility of Internet-based, force-feedback telerobotics equipment, in which the control loop between master and slave robots is closed across Internet. This is equivalent to deal with the stability of telerobotic equipment with a variable communication delay and data losses.

Types of MPC

a) Linear MPC

1. Uses linear model: $\dot{x} = Ax + Bu$
2. Quadratic cost function: $F = x^TQx + u^TRu$
3. Linear constraints: $Hx + Gu < 0$
4. Quadratic program

b) Nonlinear MPC

1. Uses nonlinear model: $\dot{x} = f(x,u)$
2. Cost function can be non-quadratic: $F(x,u)$
3. Nonlinear constraints: $h(x,u) < 0$
4. Nonlinear program

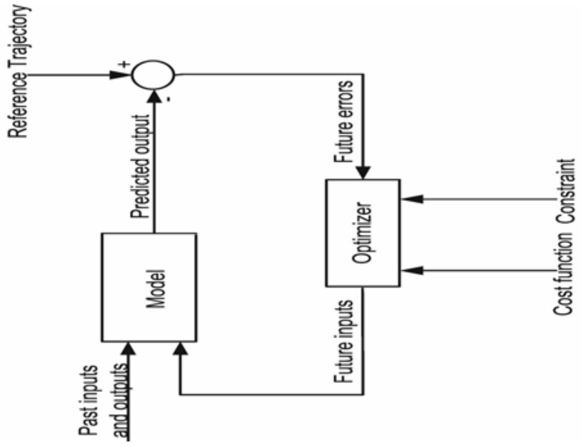


Fig. 9 Basic Structure of MPC

5.1 Predictive Control

Predictive Control (BGPC) is base don an extension of Model Predictive Control.

Model Predictive Control (MPC) is an advanced method for process control that has been used in several process industries such as chemical plants, oil refineries and in robotics area. The major advantages of MPC are the possibility to handle constraints and the intrinsic ability to compensate large or poorly known time-delays. The main idea of MPC is to rely on dynamic models of the process in order to predict the future process behavior on a receding horizon and, accordingly, to select command input w.r.t the future reference behavior. Motivated by all the advantages of this method, the MPC was applied to teleoperation systems. The originality of the approach lies in an extension of the general MPC, so-called Bilateral MPC (BMPC), allowing to take into account the case where the reference trajectory is not a priori known in advance due to the slave force feedback. The bilateral term is employed to specify the use of the signal feedback, which alters the reference system dynamic in the controller.

5.1.1 Generalized Predictive Control

Generalized Predictive Control (GPC), is one of the most popular predictive control strategies. GPC is based on the minimization of a quadratic cost function of the form (1) including a future control sequence on a receding horizon.

$$J = \frac{1}{2}\left(\begin{array}{l} \sum_{j=H_w}^{H_p} \left\| \hat{y}(k+j|k,W) - (k+j \right\|_{Q(j)}^{2}) \\ + \sum_{j=1}^{H_u} \left\| \Delta w(k+j-1) \right\|_{R(j)}^{2} \end{array} \right) \quad (2)$$

Predictive control, commonly grouped as model predictive control (MPC), uses a model of the plant to predict the output in the future $y\hat{}(k + j\ k)$. The GPC uses the Controlled Auto-Regressive and Integrated Moving Average (CARIMA) structure which is an input-output formalism taking into account the noise influence on the system through the *C* polynomial:

$$A(z^{-1})\Delta y(k) = z^{-\tau}B(Z^{-1})\Delta w(k-1) + C(z^{-1})\xi(k) \quad (3)$$

where *y(k)* and *w(k)* are respectively the output and the control of the system. $\Delta(z^{-1}) = 1 - z^{-1}$ is the differencing operator. The τ parameter, a multiple of the sampling period, is the pure system delay and *ξ(k)* is an uncorrelated random sequence. *A*, *B*, *C* are polynomials of the backward-shift operator *z-1* with respectively the following degrees *nA*, *nB* and *nC*. *A* and *C* have unit-leading coefficients. The *C* polynomial may be used as a tuning parameter, since its identification is usually avoided. It has been shown by that the *C* polynomial plays a crucial role in the robustness and disturbance rejection of the control law. More generally, this polynomial influences the robustness and disturbance rejection.

Bilateral Generalized Predictive Control Design. Due to the slave force feedback, the master trajectory is not a priori known in the future.

Therefore, we cannot determine a control sequence that minimizes the (1) cost function. To overcomethis difficulty, the Bilateral GPC (BGPC) approach proposes to rewrite the master model according to the slave control via the slave force feedback in order to determine the master output optimal prediction.

Having determined the master and slave CARIMA models for the BGPC, the minimization problem (3) is solved, where *ym* and *ys* are respectively the positions of the master system and of the slave robot end-effector.

$$J = \frac{1}{2}\left(\begin{array}{l} \sum_{j=H_w}^{H_p} \left\| \hat{y}_s(k+j|k,W_{ms}) - \hat{y}_m(k+j|k,W_{ms}) \right\|_{Q(j)}^{2} \\ + \sum_{j=1}^{H_u} \left\| \Delta w_{ms}(k+j-1) \right\|_{R(j)}^{2} \end{array} \right) \quad (4)$$

The objective is to determine the control sequence ***Wms*** minimizing the quadratic error between the future predictions of the master system output and the future predictions of the slave system output; both of these two outputs depend both on the control sequence. The plant output predictions *ŷm(k+j)* and *ŷs(k+j)* are obtained by solving two Diophantine equations for each incremental models.

Control law.

The receding horizon principle assumes that only the first value of the optimal control sequence resulting from the minimization of (3) is applied. At the next sampling period, the same procedure is repeated. This control strategy leads to a 2-DOF predictive RST controller, implemented through a difference equation:

$$R(z^{-1})\Delta w_{ms}(k) = T(z^{-1})y_m(k) - S(z^{-1})\hat{y}_s(k+\tau_g) \quad (5)$$

By appropriate choices of the horizon lengths *Hw*, *Hp, Hu* and of the weighting matrices **Q**, **R** in BGPC, an excellent master reference trajectory tracking may be obtained for the slave system. It is interesting to note that *T(1) = S(1)* to guarantee offset-free response and that the polynomial *T(z-1)* does not contain a non-causal structure generally inherent in the polynomial predictive control. This major difference, in comparison to the standard GPC, is due to the future reference trajectory, which is not a known priori. The experimental validation of the proposed BGPC approach is presented in the next section.

A robust approach. Stability conditions for constant and time-varying transmission delays of the nominal overall transfer function from the input force of the operator to the environment contact force have been determined on a frequency-domain. These conditions are derived by the small-gain theorem. Moreover, the proposed BGPC approach, which has taken into account the slave force feedback, introduces a new prefilter polynomial *Csem.* This *Csem* polynomial plays a role in robustness and disturbance rejection of the overall system. The advantage about of the proposed approach is to impose the desired behavior at remote system, to ensure a robust stability of teleoperation in the presence of environment and transmission timedelays uncertainties.

Delay jitter compensation. A different solution, is to even out the delay jitter by storing the incoming packets in a memory buffer. Given the standard deviation σ of the delay, a queue capable of absorbing a +/- 3σ variation of the delay is set up on both sides of the communication channel. This is realized with a FIFO queue with a length N=6σ/T, where 1/T is the transmission rate, as shown in fig.4.

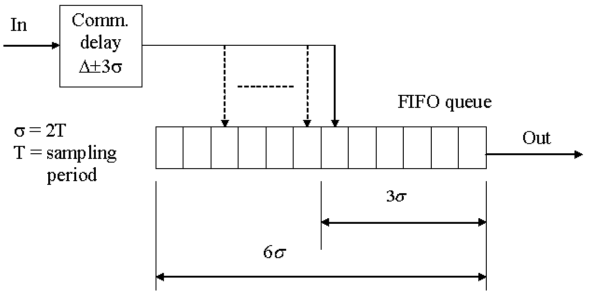


Fig.10 Delay jitter compensation via buffering Data extraction begins when the queue is filled up to half of its length. This mechanism introduces an additional delay of 3σ to the transmission delay, but this can be easily handled by simply designing the control algorithm considering an augmented delay or by using an IOD control technique.

With this solution, the connection results to have a constant delay, for which one of the standard control techniques for time-delay teleoperators can be used.

6 Applications

There exist many other Web robots on the net, performing a variety of tasks such as those described in [13]. The NASA Space Telerobotics program website (http://ranier.oact.hq.nasa.gov/telerobotics page/realrobots.html) currently lists over 20 Real Robots on the Web. Reviewing all those web-based teleoperation systems, it is clear that the main problem is of course the unpredictable and variable time delay for communication over the Internet, which calls for the use of some form of supervisory control or on-line teleprogramming scheme to ensure stability.

Most of the systems currently available on the web incorporate user interfaces, which implement basic functionalities, such as enabling the user to choose from a pre-specified set of tasks (e.g. target locations). These interfaces use some combination of HTML forms or Java consoles to enter data and issue simple commands for immediate or future execution (the requests issued by different client sites are scheduled by the robot server). Sensory feedback is usually limited to the display of images that are captured at the remote site, and the presentation of some status information in text form. It is obvious that this separation between the actions of the human operator (user) and the response of system fed back by the remote/slave robot deteriorates the transparency and telepresence characteristics of the teleoperation system. In other words, the user feels distant from the teleoperated system, and is forced to employ some form of move and wait strategy.

7 Conclusion

The Internet protocols do no guarantee a maximum delay for a message to be carried across a network link, which means that the control scheme must work under variable (and possibly large) time delays. Continuous control is not well suited, as it is prone to instability problems under time delay. In spite to all limitations, however, it is possible to realize reliable systems that in future will help in improving everyone's quality of life. In fact, remote diagnosis and rehabilitation, access to dangerous and/or remote sites will be more and more accessible and more applications are going to appear, all aimed at easing the interaction between distant worlds.

References:

[1] Allman, Hayes, Kruse and Ostermann, 1997 M. Allman, C. Hayes, H. Kruse, S. Ostermann, 'TCP Performance Over Satellite Links', '*The Fifth International Conference on Telecommunications Systems, Nashville,TN'*, March, 1997

[2] Anderson, R. J. and M. W. Spong, (1989) *"Bilateral control of teleoperators with time delay,"* IEEE Trans. On Automatic Control, vol.34, n.5, pp. 494-501

[3] K. Goldberg, \Introduction: The Unique Phenomenon of a Distance", in: *The Robot in the Garden. Telerobotics and Telepistemology in the Age of the Internet*, K. Goldberg (ed.), MIT Press 2000.

[4] Munir, S. and Book, W. J., *Wave-Based Teleoperation with Prediction,* Proceedings of the American Control Conference, Vol. 6, pp. 4605-4611, June 2001.

[5] Niemeyer G. and J. J. E. Slotine, (1991*) "Stable adaptive teleoperation,"* IEEE Journal of Oceanic Engineering, vol.16, n.1, pp.152-162.

[6] Niemeyer, G. and J. J. E. Slotine, (1998), "*Towards force-reflection teleoperation over the Internet,"* in Proc. IEEE Conf. Robotics and Automation (ICRA), Leuven, Belgium, pp.1909-1915

[7] Oboe, R. (2003). *Force-reflecting teleoperation over the internet: The JBIT project*. Proceedings of the IEEE, 91(3), 449–462.

[8] Oboe, R. and P. Fiorini, (1988) *"A design environment for Internet-based telerobotics*,"The International Journal of Robotics Research, vol.17,n.4, pp.433-449.

[9] Park J.H.,and H.C. Cho, (1999) *"Sliding-mode controller for bilateral teleoperation with varying time delay,*" in Proc. Advanced Intelligent Mechatronics Conference, Atlanta, USA, pp. 311- 316.

[10] Secchi, C., Stramigioli, S., & Fantuzzi, C. (2003).*Dealing with unreliabilities in digital passive geometric telemanipulation.* In Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (Vol.3, pp. 2823–2828).

[11] Secchi, C., Stramigioli, S., & Fantuzzi, C. (2003b). Digital passive geometric telemanipulation. In Proceedings of the IEEE international conference on robotics and automation (Vol. 3, pp. 3290–3295).

[12] Sheng J. and M. W. Spong,, (2004) *"Model predictive control for bilateral teleoperation systems with time delays",* Canadian Conference on Electrical and Computer Engineering, , v.4, p. 1877--1880, Tampa, Florida, U.S.A.

[13] Slama, T., D. Aubry, A. Trevisani, R. Oboe and F Kratz (2007a) *"Bilateral Teleoperation over theInternet: Experimental Validation of a*

Generalized Predictive Controller", in Proc. European Control Conference'07, Kos, Greece.

[14] Slama,T., D. Aubry, R. Oboe, and F. Kratz, (2007b*) "Robust bilateral generalized predictive control for teleoperation systems"*, in Proc 15th. Of IEEE Mediterraneanl Conference on Control and Automotion '07, Athens, Greece.

[15] Lynn Conway, Richad Volz, Michael W. Walker Teleautonomous Systems: Projecting and Coordinating Intelligent Action at a Distance IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION. VOL 6. NO 2. APRIL 1990

[16] Venners B (1999) Under the hood: Solve real problems with aglets, a type of mobile agent. JavaWorld. http://www.javaworld.com/javaworld/jw-05-997/jw-05-hood.html

[17] Raibulet C, Demartini C (2000) Mobile agent technology for the management of distributed systems–a case study. Comput Netw 34:823–830

[18] *Yulia V. Fetisova , Alexander N. Sesekin* Discontinuous Solutions of Differential Equations with Time Delay, *, pag 487-493, WSEAS TRANSACTIONS on SYSTEMS, Issue 5, Volume 4, May 2005, ISSN 1109-2777*

[19] Real-Time Systems Modeling and Scheduling with Hyperperiodic Tasks, *Chakib Chraibi,* WSEAS TRANSACTIONS on CIRCUITS AND SYSTEMS, Issue 3, Volume 3, May 2004, ISSN 1109-2734,pag.486-pag.492