Remote Interaction with Mobile Robots

ALAA M. KHAMIS, FRANCISCO J. RODRÍGUEZ AND MIGUEL A. SALICHS
Department of Systems Engineering and Automation, Carlos III University of Madrid,
Avda. Universidad, 30, 28911 Leganes, Madrid, Spain
akhamis@ing.uc3m.es
urbano@ing.uc3m.es
salichs@ing.uc3m.es

Abstract. This paper describes an architecture, which can be used to build remote laboratories to interact remotely via Internet with mobile robots using different interaction devices. A supervisory control strategy has been used to develop the remote laboratory in order to alleviate high communication data rates and system sensitivity to network delays. The users interact with the remote system at a more abstract level using high level commands. The local robot's autonomy has been increased by encapsulating all the robot's behaviors in different types of skills. User interfaces have been designed using visual proxy pattern to facilitate any future extension or code reuse. The developed remote laboratory has been integrated into an educational environment in the field of indoor mobile robotics. This environment is currently being used as a part of an international project to develop a distributed laboratory for autonomous and teleoperated systems (IECAT, 2003).

Keywords: mobile robotics, remote interaction, online robots, internet robotics, remote laboratories

1. Introduction

The new trends in robotics and automation systems have helped to develop many robotic systems, which aim to spread the use of the robot in human daily life. Most of these new trends have been made possible by the evolution of the personal computer (in terms of cost, power, and robustness) and the Internet (in terms of security, speed, and reliability) (Brugali and Fayad, 2002). The Internet has become a major global tool for communication and information sharing. It provides a global, integrated communication infrastructure that enables an easy implementation of distributed systems. Recently, considerable research efforts in the field of mobile robotics are addressing the use of the Internet as a communication medium to facilitate remote interaction with mobile robots. Remote interaction is a special type of human-robot interaction, where the human and the robot are separated by physical barriers but linked via telematic technologies. Such type of interaction can be used in many useful applications such as remote experimentation, teleoperation, teleperception, teleprogramming, etc.

Remote laboratories can be considered innovative environments, which can be used to provide remote interaction with mobile robots for educational and research purposes. Remote laboratories can be defined as network-based laboratories where the user and the real laboratory equipment are geographically separated and where telecommunication technologies are used to give users access to laboratory equipment (Khamis et al., 2003). Such laboratories have the advantage that they are not restricted to synchronized attendance by instructors and students: thus they have the potential to provide constant access whenever needed by students.

In recent years, many researchers have built remote laboratories for mobile robots using similar architectures and implementation tools (Archive, 2003). Many such remote facilities can be put together to form a framework or a distributed laboratory that can be used to provide a coordinated set of experiments for students with hardware facilities physically spread over different locations, but accessible via the Internet. The project IECAT (Innovative Educational Concepts for Autonomous and Teleoperated Systems) in which we are participating is an example of such frameworks in the field of mechatronics (IECAT, 2003). Such frameworks aim to be an electronic workspace for distance collaboration and experimentation in research or in another creative activity, to generate and deliver results using distributed information and communication technologies. They assist in the exchange of existing hardware resources and educational materials between the partners.

In the designing of these distance laboratories for robotic systems, a number of challenges must be addressed, particularly the telematics infrastructure which gives access to experiments, as well as the user interface which provides the necessary interactivity with the remote hardware supporting the learning process of students through appropriate feedback. In this paper, these considerations are discussed.

This paper is divided into seven sections. The first section gives a brief description of the remote laboratories. The advantages and the disadvantages of using Internet as communication medium are presented in Section 2. Section 3 discusses the remote interaction with a mobile robot with a special focus on the selection of control strategy. The system architecture is described in Section 4 followed in Section 5 by the implemented pedagogical scenarios, which incorporate the proposed remote laboratory in an educational environment for mobile robotics. Section 6 presents user feedback and finally the paper is concluded in Section 7.

2. Internet as Communication Medium

Although the Internet provides a cheap and readily available communication medium for remote interaction systems, there are still many challenges that need to be solved before successful real-world applications can be achieved. These problems include restricted bandwidth, random time delay and data loss, which influence system performance. Internet performance as a communication medium has nondeterministic characteristic, which depends mainly on the network load. The performance of a computer connection is based on the speed and reliability with which data are transmitted over that connection. The speed and reliability of the entire Internet cannot be measured at present. Very little quantitative performance data is available, but a number of projects have been created recently to analyze performance. Currently, the Internet Traffic Report (ITR, 2003) and Internet Weather Report (IWR, 2003) provide an approximate measure of Internet performance by measuring the performance of connections among a small group of monitored sites distributed throughout the world. Delay and packet losses are monitored as measures of connection speed and reliability respectively. Other projects are trying to develop metrics to analyze overall Internet delay and packet loss such as the Internet Performance Measurement and Analysis Project (IPMA, 2003) and the Internet Protocol Performance Metrics (IPPM, 2003).

Many researchers have studied the use of Internet as a communication medium from different points of view. Brady and Tarn (1998) discuss Internet-based remote teleoperation including a theoretical foundation for modeling communications delays. Han et al. (2001) have proposed a control architecture, which guarantees that a personal robot can avoid obstacles and reduce path error and time difference between a virtual robot at the remote site and a real robot at the local site. The proposed architecture is insensitive to the inherent time delay where the personal robot is controlled using a simulator provided at a remote site.

Others try to use the Quality-of-Service (QoS) model to improve the efficiency of Internet-based systems (Fung et al., 2002; Gillet and Salzman, 2002). On the Internet and in other networks, QoS is the idea that transmission rates, error rates, and other characteristics can be measured, improved, and, to some extent, guaranteed in advance (Whatis, 2003). QoS is usually described by four parameters, time delay, bandwidth, jitter and packet loss. The implementation of QoS model in today's networks is still in process, thus giving the opportunity to propose new dynamic tuning parameters (Hirche, 2002). Internet-based systems can take advantage of the implementation of the QoS model not only through the guaranteed end-to-end behaviour, but also by the control of network parameters through a performance criterion in order to adapt the performance to the task requirements.

Other researchers study the Internet effects from user perspectives. A quantitative evaluation of operability that depends on communication time delay in teleoperation and time perceptions of human operator presented in Ando et al. (1999). The experimental results of this study showed some tendency that from 0.6 s to 1 s in delay there was some change of operator's operation time. Other qualitative studies show that people seem to be able to compensate for (learn) small added delays, but cannot learn large ones (>100 ms.) and the operator performance decreases significantly as the time delay increases beyond 0.2 s (Rogers and Murphy, 2001).

A survey was conducted during the period from 7/1/02 to 5/2/02 to evaluate the network performance between the University of Applied Sciences (FH-Weingarten) in Germany, and the University Carlos III in Madrid (UC3M), which are partners in the IECAT project. The results of the study have been used as guidelines for selecting the suitable control strategy for the remote laboratory. Many tools can be used to measure the network performance such as ttcp (2003), pingER (2003) or hp netperf (2003). In this study the Netperf tool was used, which consists of two different executables: the netperf client in a 200MHz Linux Machine at FH-Weingarten, and the netperf server installed in a 300 MHz PIII Linux Machine at UC3M. The test was re-run several times using different test durations and it was being run two times per day (at 11 AM and at 6 PM) and the average was calculated to insure high accuracy.

2.1. *Time Delay*

TCP Connect/Request/Response Test has been used to measure the round trip time between the two sites. This test mimics the http protocol used by most web browsers. Instead of simply measuring the performance of request/response in the same connection, it establishes a new connection for each request/response pair. The measured round trip time has been used to calculate the total time delay between the client and server based on the TCP analysis model described in Kurose and Ross (2001). Figure 1 shows a comparison between the local and remote host time delay, which is the time elapsed from when the client (at Fh-Weingarten) initiates a TCP connection until the client receives a requested 10 kB web page stored in UC3M server.

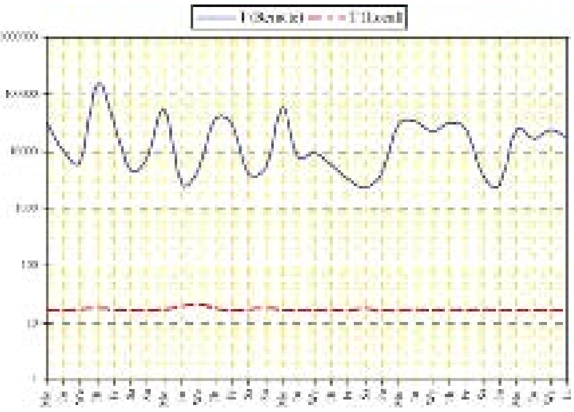

Figure 1. Time delay variation in ms.

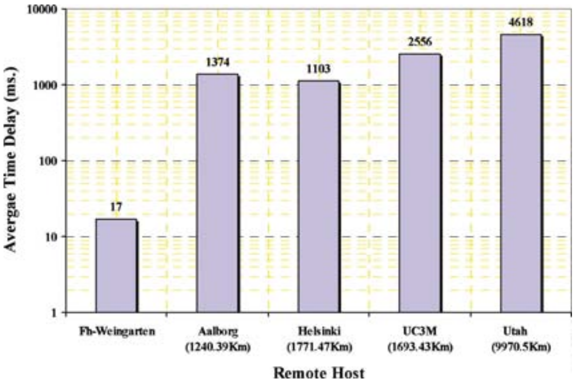

Figure 2. Average time delay measured from different sites.

The results show that the Internet imposes serious time delays, which are highly unpredictable and inevitable, unlike traditional teleoperation systems where a dedicated transmission link is used to guarantee the time delay and the bandwidth.

Another study has been carried out to compare the response time from different hosts. Figure 2 shows the results of the comparison between the average response time calculated during one week from FH-Weingarten, Germany as local host and four remote hosts, which are UC3M, Spain, University of Aalborg, Denmark, Helsinki University of Technology, Finland and Utah State University, USA. The used model supposes that a 10 Kbytes html file with one referenced object be transferred from the remote host to the local host.

The results show that the physical distances between the hosts have not dominant effect on the time delay. The Internet time delay is caused by the queuing delay, the processing delay and the transmission delay in the switches and the propagation delay in the links. The queuing delay defines the time; a packet waits in the buffer of a switch for transmission onto the next link. Therefore its value varies with the network load. The propagation delay depends on the physical distance due to the speed of light. With regard to all other delays adding to the total communication delay, the queuing delay represents the major portion. The queuing time depends mainly on the network load, which causes the arbitrary feature of the delay. The Internet delay variation is the most important problem to handle when implementing remote interaction systems. Depending on the network load, the delay can vary from a few milliseconds to hundreds. This delay affects the reliability of the remote operation. Beyond a certain delay, manual control of a vehicle may become highly error prone or impractical (McGovern, 1990). A maximum time delay of 1 s is usually taken as a reference of operability in remote control systems.

2.2. *Bandwidth*

Bandwidth has a general meaning of how much information can be carried in a given time period (usually a second) over a wired or wireless communications link. In data transmission, *throughput* is the amount of data moved successfully from one place to another in a given time period (Whatis, 2003). To evaluate the bandwidth between UC3M and FH-Weingarten, the Netperf Stream test has been used. In TCP Stream Test netperf client sends messages of selected size to the netserver daemon which receives them. There is no protocol beside the TCP/IP protocol but all messages sent are also considered as correctly received hence the term throughput can be used for the bandwidth. A timer makes the test stop and the throughput is calculated in both sides by using the message size, number of messages and elapsed time. The test was re-run several times using different test durations and the average throughput was calculated. Figure 3 shows the results of this test.

In Bapna et al. (1998) a teleoperation system for the mobile robot Nomad has been presented, which gives us the opportunity to operate Nomad safely from the distant control centers. Images and data from Nomad were also immediately available on the Internet. Bapna et al. have proved that the 1.4 Mbps bandwidth is sufficient enough to transport real imagery from the rover to a local control station and then to remote mission control sites. The obtained results (average = 1.61 Mbps) can be considered satisfactory but it is necessary to take into account the nondeterministic feature of network performance.

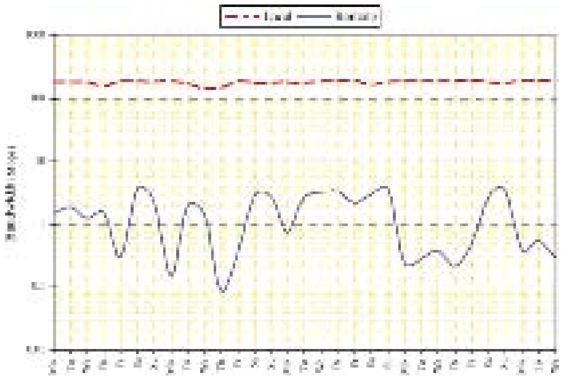

Figure 3. Bandwidth in Mbps.

In Internet-based systems, high-level commands are recommended to be used in remote interaction with the mobile robot because these commands require less bandwidth. The idea -of overcoming communication constraints by communicating at more abstract level and increasing the robot’s autonomy- is fundamental to remote control via constrained communications. The limitation in bandwidth also limits the refreshing rate of the video images, which prohibits experiments in which the scene might vary at high speeds. Otherwise abrupt breaks and jumps in the visualization of the events will occur. Virtual reality images of a rendered simulation have much lower needs in throughput; therefore they pose a better alternative to real camera images in these kinds of situations (Khamis et al., 2002). If real time images must be provided from remote sites, the system should react dynamically to changing bandwidth and computational resources and only transmit those pixels that are actually needed by using intelligent techniques (intelligent fragmentation, intelligent frame rate, intelligent task rate and brute force compression) described in Sayers (1999). In multi-user systems, system resources have to be shared and some types of interaction should be detrimental or prohibitive, particularly those that require a high and dedicated communication bandwidth.

2.3. *Jitter*

A crucial component of end-to-end delay is the random queuing delays in the network devices. Because of these varying delays within the network, the time elapsed from generation of a packet at the source until it is received can fluctuate from packet to packet. This phenomenon is called instantaneous variability or jitter. Jitter is the variance in one-way latency. The previous round trip measurements can be used to measure the jitter. Let the i-th measurement of the round trip time (RTT) be R_i, then the “jitter” is taken as being the Inter Quartile Range (IQR) of the frequency distribution of R (Cottrell, 2003). From descriptive statistics of the measured round trip time, jitter can be calculated as following:

$$\text{Jitter} = \text{Inter Quartile Range (IQR)} = \text{third quartile (Q3)} - \text{first quartile (Q1)} = 1911 - 340 = 1571\ \text{ms}.$$

This value represents high jitter, which can negatively affect the performance of data transmission especially in speech applications. Although speech transmission does not need a high bandwidth, it is very sensitive to delay changes. Gateways can be used to delete jitter in such applications. Many mechanisms have been proposed to remove the jitter in some special applications such as voice applications or Internet telephony (Kurose and Ross, 2001).

2.4. *Packet Loss*

Package loss is caused by exceeding the network capacity causing a network device to drop a packet. Probably the greatest concern of any Internet-based systems is the nondeterministic system behaviour that would result during packet loss or total dropout. One possibility in order to prevent package loss is implemented in TCP, where, as soon as package loss is detected, a resend is requested by the receiver. That results in higher latency regarding TCP compared to UDP, so there is a tradeoff between the parameters package loss ratio and time delay (Hirche, 2002).

3. Remote Interaction

The remote interaction is a special type of human robot interaction, where the human and the robot are separated by physical barriers but linked via telematic technologies. Such type of interaction can be used in many useful applications such as remote experimentation, teleoperation, teleperception, teleprogramming, etc. However, regardless of the application, most of the remote interaction systems consist of the following basic components as shown in Fig. 4.

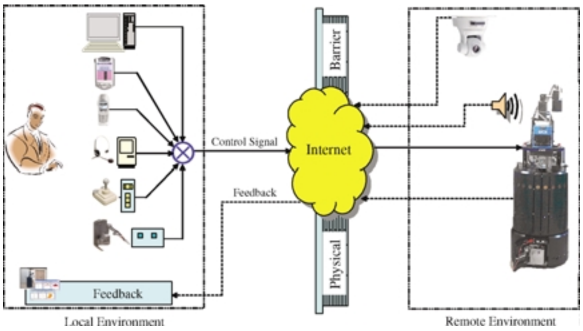

Figure 4. Remote interaction.

- An operator interface, incorporating an interaction device that the operator uses to send control commands to the remote system. There are many types of interaction devices such as PCs, mobile devices such as Personal Digital Assistants (PDA) and mobile phones, speech-based communication, joysticks and haptic interfaces.
- A mobile robot that performs the operator's commanded actions at the remote site.
- A communication scheme between sites. It is recommended that the communication scheme have a robust signal communication link with an acceptable time delay; dedicated data links with sufficient throughput; and an effective data loss-recovery approach. As mentioned previously, although Internet is a cheap, readily accessible communication medium, its performance has nondeterministic characteristics.
- Feedback interfaces. Video transmission is commonly used to provide visual feedback for the operator. Video transmission demands high bandwidth availability. When this is not possible, computer-generated imagery supplies the operator with a virtual interface that combines low bandwidth sensory data to form a realistic image. These virtual interfaces are also useful to overlaying computer predictions or visual clues onto video images. Auditory feedback can also be used to enhance visual experience and human robot interaction. Although audio transmission requires low bandwidth, it is very sensitive to time delay and jitter. Other type of feedback can be provided using kinesthetic aids. Using haptic systems the operator can sense the response of its commands directly in the control interface.

The control strategy of the remote interaction system plays an important role in the performance and stability of the system. When there is significant time delay in the communication link, instability occurs and manual closed-loop control is no longer suitable (Sheridan, 1992). Two control strategies are discussed here, which can be used to develop Internet-based systems.

3.1. *Manual Closed Loop Control*

Most of the Internet-based systems use the manual closed loop control model because of its simplicity. These systems tolerate delays by using the human operator as an intelligent link to the control system. An error in the position of the remote device is visible in the

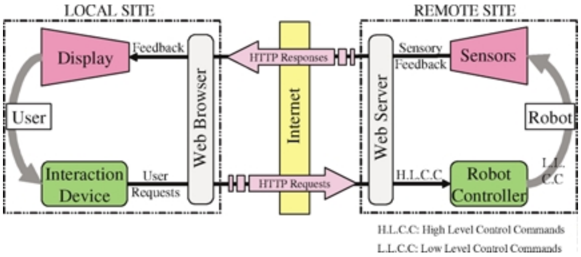

Figure 5. Manual closed loop control model.

returned imagery, and the operator sends commands to the remote device to compensate. This model has the components shown in Fig. 5.

It is based on a simple protocol commonly used in distributed computation "The Request/Response Protocol". The client interacts with the system using any Web browser to make the request. Client requests are translated into HTTP requests, which are satisfied by the Web server. These requests are converted to high-level control requests that are received by the robot controller, which transmits them as low level control requests to execute the required task. Sensory feedback is required to give the user information about the remote robot's environment and the consequence of his/her commands. While these systems are not certainly optimal, they are simple and they work. Predictive displays can be used to improve this model. Because of the communications delay, immediate feedback cannot come from the remote site, and instead it must be generated at the operator station (Sayers, 1999).

3.2. Supervisory Control

One technique that can improve operator performance and avoid the instability problem is a shared or supervisory control scheme, where control of the robot is shared between a remote control loop and the local human operator (Sheridan, 1992). The goal is not to make a robot that can realise the full task autonomously but to enable the robot to perform some simple subtasks that the operator can sequence. By supervisory control, the user can communicate with the remote system at a more abstract level using high or task-level commands sent to a robot, which has an increased level of autonomy in order to alleviate high communications data rates. Limiting the remote interaction to high level commands helps to decrease the bandwidth needs and increasing robot's autonomy can help to decrease teleoperated task sensitivity to the delay. Figure 6 shows the main components of the control model used to develop the remote laboratory. The following subsections describe the developed model in more details:

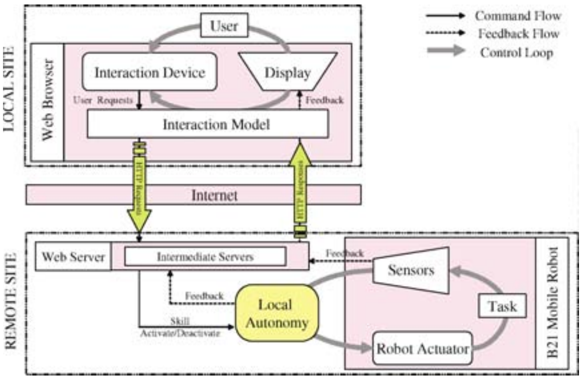

Figure 6. Supervisory control.

3.2.1. Robot Autonomy. To increase the local autonomy of the robot the concept of skill has been used. A skill represents the robot's ability to perform a particular task. They are all built-in robot action and perception capacities (Alami et al., 1998). The AD (Automatic Deliberative) control architecture has been proposed to develop different robotic skills (Barber and Salichs, 2001). This architecture has two main levels. The automatic level contains low-level control modules, which act directly upon the actuators, as well as the modules that collect data from the different sensors of the system. The automatic skills have been classified into motor, perceptive and sensorimotor skills. Motor Skills are those skills which generate actions for the robot's actuator such as direct control skill. Perceptive Skills are those skills which interpret the information obtained from sensors or other perceptive or sensorimotor skills. Sensorimotor Skills obtain as input the information provided by sensors or other perceptive skills, and based on this information choose the most adequate actions for the actuators.

The other level is deliberative, which contains modules that require reasoning capacity. Those modules do not produce immediate responses. They need a longer time to process the information they work with. Those modules form the deliberative skills such as path planner, the environment modeler and the task supervisor. Skills have been implemented in form of client-server modules (Boada, 2002; Boada et al., 2002). As shown in Fig. 7, each module contains an active object, an event manager object and data objects.

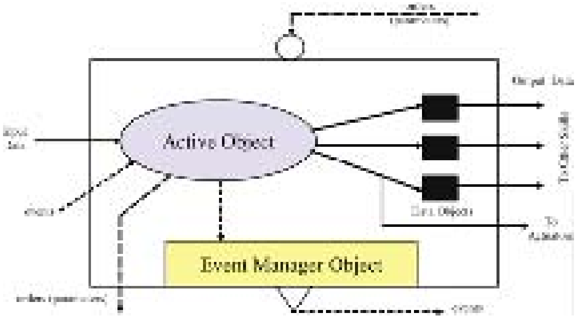

Figure 7. Skill structure.

Each skill is a distributed object that has an identity, interfaces and state. The active object of the skill has its own thread of control and it is in charge of processing. The processing results are stored in the data objects. These objects contain different data structures depending on type of data stored but the interfaces are similar. During the processing, the active object can generate events. Events are sent to the event manager object, which is in charge of notifying them to other skills, which have registered on it. In order to communicate among objects of the same module or different modules Common Object Request Broker Architecture (CORBA) is used. When a skill is activated, it connects to data objects of other skills or to sensors' servers as required by the skill. Then, it processes the received input information, and finally, it stores the output results in its data objects. If the skill is sensorimotor, it can connect to actuators' servers in order to send them movement commands. A skill can send a report about its state while it is active or when it is deactivated. For example, the skill called go-to-goal can inform on whether the robot has achieved the goal or not. When this skill is deactivated it might inform about the error between the current robot position and the goal (Boada et al., 2002).

3.2.2. User Intervention. In this model, the user has a supervisory role by which he can activate or deactivate the robotic skill. The error detection and recovery is perhaps the most significant challenge to time-delayed telerobotics. Errors can be handled by using a three-stage process: autonomous detection, shared diagnosis, and manual recovery. Errors are detected by using a visualization mean such as streaming video, graphical models, sensory data or connection status panels. The diagnosis task is shared by the user and the system. In recovering from the error, the user can telecollaborate with a human at the remote site or can ask the necessary privileges to be able to telnet the remote servers to reboot them.

Table 1. Total response time in ms.

Host	Min.	Av.	Max.
UC3M	131.89	138	148.65
FH-Weingarten	283.89	324.89	264.89

3.2.3. Response Time. By limiting the remote interaction to high-level commands and by increasing the local autonomy of the robot using the concept of skills, the system sensitivity to the communication delay can be decreased dramatically. The total response time, which is the time elapsed between sending the skill activation command until the start of skill execution, has been obtained by measuring the time delay from client to middleware server and then from middleware server to the skill server by using software clocks. The total response time is the sum of the two round trip delays.

The results showed that the connection between the middleware and the server (radio connection 1 Mbits/s) doesn't impose significant delay. The average value (10.89 ms) can be considered as a constant round trip delay between the middleware and the server. The major part of the delay was the client-middleware delay due to the Internet. Table 1 shows the total response time.

The measured values (138 ms from UC3M and 324.89 from Weingarten) represented acceptable values for remote laboratories applications taking into account that a maximum communication delay of 1 sec is usually taken as a reference of operability in teleoperation systems. These delays will be noticed by the users but they are accepted for an educative application as remote laboratory.

4. System Architecture

A remote laboratory for mobile robotics can be built using two kinds of distributed systems models. The first choice is to use a two-layer client/server structure, and the second option includes a three-layer model that includes a middleware. Two layer architectures are very easy to implement at first, but it is very difficult to maintain the application up to date with them. In this case, the client side is responsible for the data access, the business logic implementation, formatting the data,

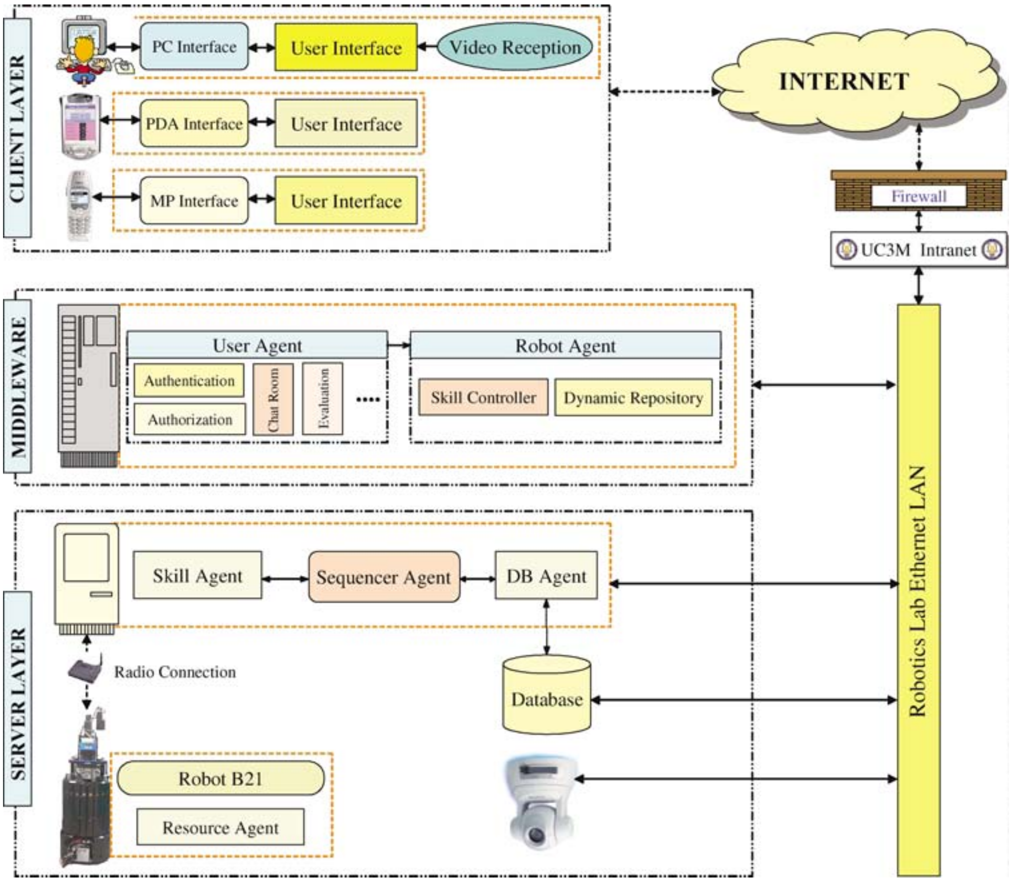

Figure 8. System architecture.

the user interface and data input. Because all the details involved on the client side, programs are very difficult to adapt to Internet, as they do not scale well. As shown in Fig. 8, the proposed architecture is three-layer architecture based on the concept of supervisory control. In this architecture, the client layer is responsible for the presentation of system interfaces, the middleware layer serves as controller to communicate the client with the remote robot servers and the server layer, which contains the abstraction model of the system represented by robot' servers. The following subsections describe in more details the architecture layers.

4.1. *Client Layer*

The visual-proxy pattern has been used to develop system interfaces (Houlb, 1999). This pattern is in some ways a specialization of the Presentation/Abstraction/Control (PAC) architecture (Rohbert et al., 1996), which can be used to build user interfaces for object-oriented systems and can guarantee high degree of extensibility and reusability of the software components). The objective of this pattern is to separate the generation of the user interface entirely from the abstraction layer objects to provide the reusability and extensibility facilitates (Houlb, 1999). The control object is passive with respect to message flow. The messages go directly from the visual proxy (presentation layer) to the abstraction-layer object that manufactured the proxy. In the visual proxy architecture, the encapsulation is still intact in the sense that the implementation of the abstraction-layer object can change without the outside world knowing about it.

As shown in Fig. 9, the *ExperimentTool* class implements *UserInterface* so it can produce visual proxies when asked. It asks the other *SkillController* and the *DynamicRepository* classes for visual proxies as simple *JComponents*. This class contains a constructor

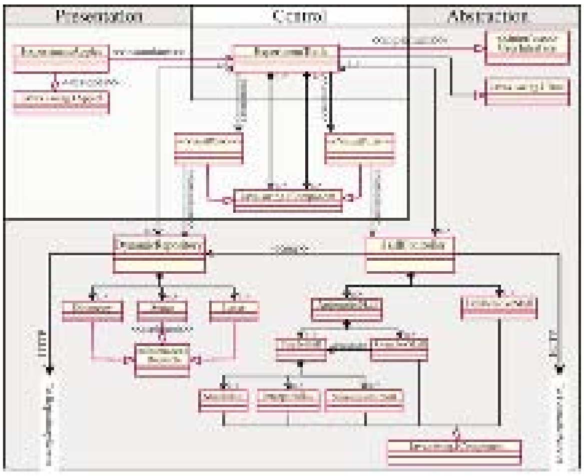

Figure 9. Client layer.

and implementation of all methods required by the interface but it is not a God-like class that controls everything from above. It positions the solicited proxies within the panel but does absolutely nothing else with them. This class is simply a passive vehicle for holding visual proxies. The proxies communicate directly with the abstraction-level objects that creates them and these abstraction layer objects can communicate with each other. If the state of an abstraction-level class changes as a result of some user input, it sends a message to another of the abstraction-level classes, which may or may not choose to update its own user interface (its proxy) as a sequence.

ExperimentApplet instantiates the *ExperimentTools* to form the experiment interface seen by the user via any Web browser. *SkillController* class is used to activate/deactivate the skill and it can be an automatic or a deliberative skill or a combination of different types of control classes in the same panel. The *DynamicRepository* class provides information about a sensor's status and can be used to remotely acquire the sensory data. This class is implemented as a thread by implementing the *Runnable* interface to provide updated sensory data in real time. This data may be odometry, which indicates the actual robot position and its translational and rotational velocities or ultrasonic sensor data or laser sensor data.

SkillController may use the *DynamicRepository* class to invoke sensory data, which may be necessary to complete the control task, such as in the case of obstacle avoidance skill by using sensory data. Both *SkillController* and *DynamicRepository* classes interact with the middleware using http protocol to send the user requests to the robot servers.

As shown in Fig. 8, different interface agents can be developed for different interaction devices by customizing the proposed design according to the capabilities of the interaction device. For example, in the case of mobile devices such as PDAs and Mobile Phones (MP), the same design can be used with some modifications necessary to deal with specific problems related to the limitations of handheld devices (slow processors, limited memory/storage, and small displays) and wireless network (restricted bandwidth, high latency, low connection stability, and low predictable availability).

4.2. *Middleware Layer*

In this layer, there is a PC linked internally to an Ethernet LAN and externally to the university's Intranet which runs an Apache Web server, which hosts two agents:

- *User agent*: contains two main servlets to manage user access. The user authentication servlet collects and verifies the user registry. The authorization servlet manages the authenticated user access and notifies the user that the access to the robot agent is granted or rejected based on round robin approach. When only a single user is connected, no restrictions on usage are given. However, if multiple users are simultaneously connected, only one is given access for a specified amount of time. Once the time has expired, the next user in line will be given opportunity to control the robot and the first user will be moved to the end of the waiting list. A chat room is provided as a communication mechanism between the different users. Also this agent holds other servlets necessary for the teleeducation environment as simulation servlet, system evaluation servlet, online test corrector servlet, etc. These services do not require connection with the real robot and they are accessible to all connected users.
- *Robot agent*: deals with the real robot when the robot is actually running. It provides the required information to process a certain skill selected by the user or required in step of an experiment. The agent contains two groups of java servlets, dynamic repository servlets to invoke the sensory data from a resource agent in the server layer and skill controller servlets that send control commands to the

skill agent in the server layer. The communication between these servlets and the remote robot servers is done via the Object Request Broker (ORB) of the Common Object Request Broker Architecture (CORBA) where the Java servlet acts as a client to the robot server. The ORB provides the communication via the unified interface language Interface Definition Language (IDL) and based on the Internet Interoperable ORB protocol (IIOP). The decision to use CORBA as the distributed object architecture of the remote laboratory is based on a qualitative and quantitative comparison between the two most commonly used architectures, CORBA and RMI (Juric et al., 2000). This study concluded that CORBA is suitable for large scale or partially Web-enabled applications where legacy support is needed and good performance under heavy client load is expected. Moreover, CORBA servers can be located at any Internet site. RMI, on the other hand, is suitable for small-scale fully Web-enabled applications where legacy support can be managed with custom build or pre-built bridges, where ease of learning and ease of use is more critical than performance. Integrating Java RMI with IIOP can provide another solution that combines RMI easy programming features with CORBA interoperability (Schaaf and Maurer, 2001).

4.3. *Server Layer*

This layer contains four main agents. Resource agent represents the servers of robot's resources implemented in C++ such as base server (actuator server, odometry server and sonar server), laser server, etc. This agent is also responsible for managing the users' requests to avoid any data inconsistency. The skill agent represents skill server, which may be simple automatic skill or deliberative skill server. The simple skills can be combined to form complex skill by using the sequencer agent (Rivero et al., 2003). The sequencer is responsible for deciding which skills have to be activated in each moment to avoid simultaneous activation of skills which act upon the same actuator. The data base agent provides access to a database, which contains plans of sequences. This layer also contains a commercial network camera (Sony SNC-RZ30P) with embedded streaming video and pan tilt zoom control capabilities to provide real imagery transmission from the remote environment to the local site of the operator.

5. Pedagogical Scenarios

Robotics education provides an ideal field for teleeducation systems because of its flexibility. Unlike traditional fields, robotics is still an emerging area. Relatively few programs exist at the graduate level, and even fewer at the undergraduate level. The courses in existence are still new and are open to rapid change and new approaches. Different educational activities can be combined in order to develop an innovative teaching environment in the field of indoors mobile robotics. These activities are classified into instructional activities and constructional activities. The instructional activities are used to map traditional teaching and learning activities and to solve traditional teaching constraints of curriculum, class size and limited resources. The constructional activities give the students the opportunity to physically construct and implement the ideas derived from the course. The following subsections describe these activities in more details.

5.1. *Instructional Activities*

The instructional activities of the course are implemented based on the Web-based education model. In the traditional education model, lectures and laboratories are the ways commonly used in any education system. This model of education is quite entrenched and the flow of knowledge is largely unidirectional, especially in large classes, with the exception of occasional questions and discussion. The amount of questions and discussion is usually inversely proportional with the class size, resulting in large classes often becoming the academic analog of watching an informational video (Bourne et al., 1996). Many instructional activities such as following can be implemented based on a web-based education model to map traditional teaching and learning activities and to solve the traditional teaching constraints of curriculum, class size and limited resources.

5.1.1. Online Classes. Online lecture is used to map the traditional classroom. Bourne et al. proposed the organization of online lecture material according to well-known taxonomies in education (Bourne et al., 1996). Barrett's Taxonomy proposed that learning should be divided into four categories: literal, inferential, applicative and evaluative. Merrill's Taxonomy uses a performance or remember, use, or find (create). The content is classified as fact, concept (classification), procedure or principle. Many online classes have been designed based on these taxonomies to present the basic knowledge of robots as a general issue and indoor mobile robotics as main subject. This knowledge includes introductory concepts, robots classifications, mobile robot anatomy, control architectures of mobile robots, sensors commonly used in mobile robots and mobile robots applications. The class also has been supplemented by FAQ pages, a search engine, a digital library and downloadable materials.

5.1.2. Remote Laboratory. An online laboratory in a field such as mobile robotics must have live performance characteristic, not just virtual reality or simulation programs. The multi-layered architecture, described in the previous section, has been implemented in order to reach this goal. The knowledge is presented to the user by using three types of tutoring tours. These tours are classified according to the level of guidance of the tutor into: fully guided, unguided, and guided tour. The fully guided tour is a demonstration tour, which demonstrates basic concepts without any intervention on the part of the student. The unguided tour or free tour does not determine any order and tasks at all. This tour provides generic tools to the user and let him/her customize the experiment according to his/her needs. These generic tools include 2D model for the robot and the lab, odometry data panel, sonar data panel, laser data panel, motion controller and low-level teleprogramming editor. The guided tour aims to present different specified courses with direct interaction between the student and the tutor. The experimental issues of the courses are provided by using interfaces of different types of skills as shown in Fig. 10.

The following examples are some of the experiments that have been implemented and are presented through the guided tour.

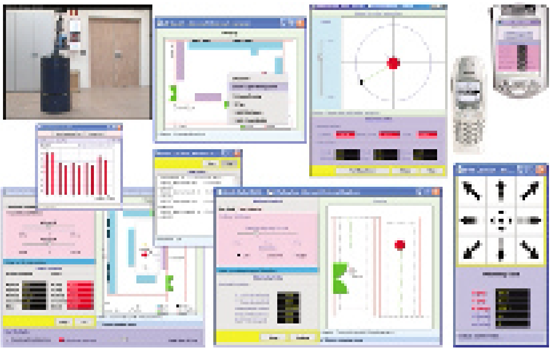

Figure 10. Screenshot of skill interfaces.

- *Direct Control.* This experiment aims at familiarizing the user with the mobile robot motion control and positioning by using different interaction elements such as PC or any MIDP (Mobile Information Device Profile)-enabled device such as handheld PDA or cellular phones. The remote user can send direct control commands to move the robot B21 from RWI forward, backward or to turn it left or right. Using a network camera, a 2D graphical model, and the odometry data (actual location with respect to the initial point, translational and rotational robot velocities), the remote user will be able to view the effect of the sent commands.
- *Go-to-Goal Skill.* By using this automatic skill, the user can send the robot to a certain point in the lab. To activate this skill the user has to set the skill parameters as goal point, robot velocity and maximum error. The skill can provide information as to whether the robot has achieved the goal or not. When this skill is deactivated it might supply information about the error between the current robot position and the goal. This skill also has been combined with a perceptive skill called detect obstacle to avoid the obstacle during motion.
- *Wall Following.* By this skill, the robot can follow certain contour using sonar data. The distance between the robot and the contour and the robot's velocity are controlled by the user.
- *Round Trip Skill*. This skill is a complex skill formed by combining go-to-goal skill and orientation control skill. It commands the robot to travel to a certain position and then return to the initial point.
- *Environment Modelling and Robot Localization*. The skill modelling skill can be used to construct an environment map using sensory data. The constructed map is then used to remotely estimate the robot's position by computing sets of poses which provide a maximal-quality match between a set of current sensor data and the map. The remote user can compare the result of the localization algorithm with the odometry data to determine the error. Figure 11 shows the experiment interface.

5.1.3. Assessment Tools. Methods of assessment are important aspect of every course. Using telematic-based model, it will be easy to construct systems that automatically correct and handle quizzes for the teaching staff and that reduce the time taken to carry out other forms of assessment. Evaluative tests can be used to provide online assessments. A student first takes

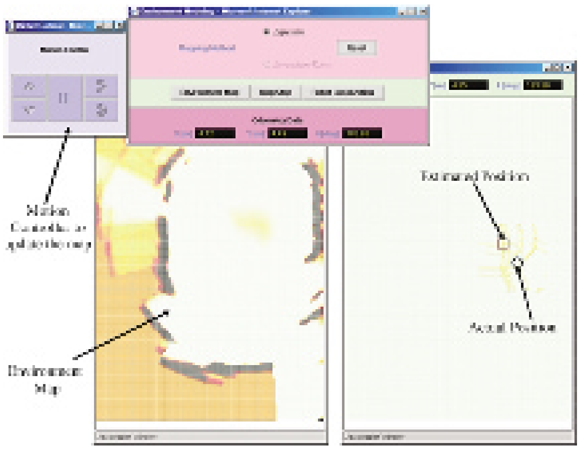

Figure 11. Modelling & localization experiment.

an exam and then, based on the results of the exam, customized learning activities can be generated to reinforce the areas in which the student scored poorly. Moreover practice tests can be designed to cover the experimental issues as a simulation to traditional laboratories. Three types of tests are designed as online assessment tools. Online quizzes are used to evaluate the student's background in mobile robotics in general. Time restricted theoretical tests are used to evaluate the student's knowledge in the theoretical topics of the course. Time restricted experimental tests are designed to help the student to understand practical problems.

5.1.4. Communication Tools. An essential part of any learning experience is communication. Two communication mechanisms (synchronous such as text/audio/video chat, Internet telephony and videoconference and asynchronous as E-mail, newsgroups and mailing lists) are in plans for implementation to provide communication between students and teaching staff.

5.2. Constructional Activities

Constructionism is an active learning process in which students construct things that are personally meaningful to themselves or others around them (Papert and Harel, 1991). Instead of being served information in the traditional one-way setting, students develop their own knowledge and understandings of a subject through physical construction and implementation of their ideas. The construction activities give the students the opportunity to physically construct and implement the ideas derived from the course. They consist of the following activities:

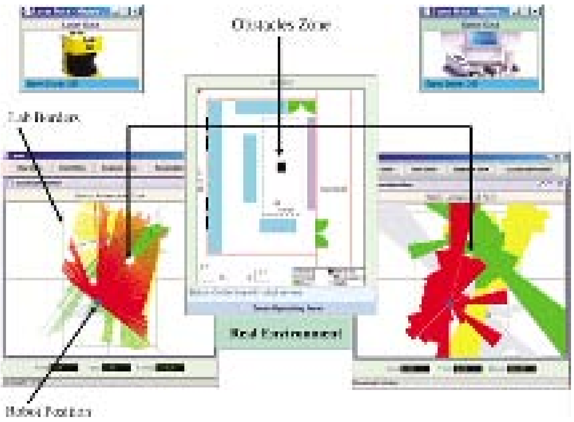

Figure 12. Environment perception experiment.

5.2.1. Off-Campus Activities. The off-campus constructional activities are active learning activities as teleprogramming, telemonitoring, telediagnostic, telemaintenance and teleperception, etc. . . , which can be incorporated into the remote laboratory.

- *Teleperception.* The objective of this experiment is the perception of environment using multi-sensor data (sonar and laser). The experiment is divided into two parts: without robot motion and with robot motion. The objective of the first part is to understand the operation of sonar and laser sensors and to be familiar with these readings. The second part aims at recognizing the real environment using sensor data. Figure 12 shows the use of accumulative readings of sonar and laser sensors in determining the obstacle zone remotely.
- *Teleprogramming.* A low level commands editor is now under development to facilitate interacting with the remote robot using low level commands or a group of commands in the form of a programming script. The user will be able to teleprogram the robot or to send low level commands via Internet to the robot's base server to perform many tasks such as adjusting the robot control parameters, increasing/decreasing robot velocity, informing about battery state, setting the Watch Dog timer, which can be used for safety or diagnostic purposes, etc. Also a graphic user interface used to define plans of sequences and store them in a database is under development to develop complex skills by combining simple ones.

Table 2. Student feedback.

Question	Agree (%)	Strongly agree (%)
The learning objectives were clear	43	57
The course was well presented	29	71
The lab was easily accessible	29	71
The corrected exercises contained helpful comments	43	57
The system was essential to the course	14	86
The system was reliable	43	57
The system was easy to use	29	71

5.2.2. On-Campus Activities. On-campus activities such as the design and construction of new prototypes or reverse-engineering of existed systems can be used to provide face-to-face interaction between the students and teaching staff and between the student and the machine, and therefore increase student motivation and decrease students' feeling of isolation. These constructional activities also help to increase student creativity and team work. Students are asked to build robotic prototypes using lab kits such as Lego or Fischertechnik parts (including motors) as hardware components, the handy board (Motorola 88HC11 based or 16-bit Siemens 80C167) micro controller board for control, commonly used sensors and other components. A teleoperated rover also can be built and programmed by the students.

6. User Feedback

The proposed teaching environment was used during the academic year 2001–2002 to update a postgraduate course on intelligent autonomous robots. Student feedback was gathered using an online questionnaire. The student responses were uniformly positive as to the use of the different proposed teaching activities specially the use of the remote laboratory. Most of the students felt that the online experiments helped them to achieve a deeper understanding of the subject material. Table 2 shows examples of students' feedback.

7. Conclusions

A three-tier architecture was proposed to build mobile robotics remote laboratories to facilitate remote interaction with mobile robots using different interaction devices for educational and research purposes. The supervisory control strategy used to implement the system helps to alleviate high communications data rates and to decrease system sensitivity to network delays. The visual proxy pattern used to build the user interfaces of the remote laboratory provides flexible user interfaces with minimal coupling relationships between subsystems. The generation of the user interface is entirely separated from the abstraction layer object to provide the reusability and extensibility facilitates. Based on the described architecture, many interfaces have been implemented for simple automatic skills such as direct control (using PC or PDA or Mobile Phone), Go to goal, orientation control and wall following skills. A sequencer has been used to combine simple skills to obtain complex skills such as go to point with obstacle avoidance and round trip skill. The developed remote laboratory has been integrated in an educational environment for mobile robotics. As a further work, more interfaces for other interaction means such as speech-based interaction, joystick or haptic-based interaction will be developed.

References

Alami, R., Chatila, R., Fleury, S., Ghallab, M., and Ingrand, F. 1998. An architecture for autonomy. *The International Journal of Robotics Research*, 17(4):315–337.

Ando, N., Lee, J., and Hashimoto, H. 1999. A study on influence of time delay in teleoperation. In *Proceeding of the 1999 IEEE/ASME Internacional Conference on Advanced Intelligent Mechatronics*, Atlanta, USA, pp. 317–322.

Archive of Online and Internet Robots. 2003. Available in: http://ford.ieor.berkeley.edu/ir/

Bapna, D., Rollins, E., Foessel, A., and Whittaker, R. 1998. Antenna pointing for high bandwidth communications from mobile robots. In *Proceedings of the 1998 IEEE International Conference on Robotics & Automation, ICRA98*, pp. 65–70, Leuven, Belgium, vol. 4, pp. 3468–3473.

Barber, R. and Salichs, M.A. 2001. A new human based architecture for intelligent autonomous robots. In *The Fourth IFAC Symposium on Intelligent Autonomous Vehicles*, Sapporo, Japan, pp. 85–90.

Boada, M. 2002. Control system for autonomous mobile robots based on reactive skills. Ph.D. Thesis (In Spanish), Universidad Carlos III de Madrid.

Boada, M., Barber, R., and Salichs, M. 2002. Visual approach skill for a mobile robot using learning and fusion of simple skills. *Robotics and Autonomous Systems*, 38:157–170.

Bourne, J. et al. 1996. A model for on-line learning networks in engineering education. *Journal of Engineering Education*, 85(3):253–262.

Brady, K. and Tarn, T. 1998. Internet-based remote teleoperation. In *Proceedings of the 1998 IEEE International Conference on Robotics & Automation, ICRA98*, Leuven, Belgium, pp. 65–70.

Brugali, D. and Fayad, M. 2002. Distributed computing in robotics and automation. *IEEE Transactions on Robotics and Automation*, 18(4):409–420.

Cottrell, L., Matthews, W., and Logg, C. 2003. *Tutorial on Internet Monitoring & PingER at SLAC*. Available in: http://www.slac.stanford.edu/comp/net/wan-mon/tutorial.html#jitter

Fung, W., Xi, N., Lo, W., and Liu, Y. 2002. Improving efficiency of internet-based teleoperation using network QoS. In *Proceedings of the 2002 IEEE International Conference on Robotics & Automation, ICRA02*, Washington, DC, pp. 2707–2712.

Gillet, D. and Salzman, C. 2002. Real-time interaction over the Internet. In *Proceedings, the 15th IFAC World Congress on Automatic Control (b'02)*, Barcelona, Spain.

Han, K., Kim, S., Kin, Y., and Kim, J. 2001. Internet control architecture for Internet-based personal robot. *Journal of Autonomous Robots*, 10:135–147.

Hireche, S. 2002. Control of teleoperation systems in QoS communication networks. Thesis, Technische Universitat Berlin.

Houlb, A. 1999. *Building User Interfaces for Object-Oriented Systems.* JavaWorld. Available in: http://www.javaworld.com/javaworld/jw-07-1999/jw-07-toolbox.html

IECAT (Innovative Educational Concepts for Autonomous and Teleoperated System) Web Page. 2003. Available in: http://www.ars.fh-weingarten.de/iecat/index.html

IPMA, Internet Performance and Measurement Analysis Project. 2003. Available in: http://nic.merit.edu/ipma/

IPPM, Internet Protocol Performance Metrics. 2003. Available in: http://www.advanced.org/ippm.html

ITR, Internet Traffic Report. 2003. Available in: http://www.internettrafficreport.com/

IWR, Internet Weather Report. 2003. Available in: http://www.mids.org/weather

Juric, M., Rozman, I., and Hericko, M. 2000. Performance comparison of CORBA and RMI. *Information and Software Technology*, 42:915–933.

Khamis, A., Pérez Vernet, M., and Schilling, K. 2002. A remote experiment on motor control of mobile robots. In *the 10th Mediterranean Conference on Control and Automation, MED 2002,* Lisbon, Portugal.

Khamis, A., Rivero, D.M., Rodriguez, F., and Salichs, M. 2003. Pattern-based architecture for building mobile robotics remote laboratories. In *IEEE International Conference on Robotics and Automation (ICRA'03),* Taiwan.

Kurose, J. and Ross, K. 2001. *Computer Networking*. Addison-Wesley.

McGovern, D. 1990. Experiences and results in teleoperation of land vehicles. Report SAND87-0646, Sandia National Lab, Albuquerque, PA.

netperf Web Page. 2003. Available in: http://www.netperf.org/netperf/NetperfPage.html

Papert, S. and Harel, I. 1991. *Constructionism*. Cambridge, MA: MIT Press.

pingER Web Page. 2003. Available in: http://www-iepm.slac.stanford.edu/pinger/

Rivero, D.M., Khamis, A., Rodriguez, F.J., and Salichs, M.A. 2002. A patterns-oriented framework for the development of automatic and deliberative skills for mobile robots. In *The 11th International Conference on Advanced Robotics, ICAR03,* University of Coimbra, Portugal.

Rogers, E. and Murphy, R. 2001, Human-robot interaction. Final Report for DARPA/NSF Study on Human-Robot Interaction. Available in: http://www.csc.calpoly.edu/∼erogers/HRI/HRI-report-final.html.

Rohbert, H., Sommerlad, P., and Stal, M. 1996. *Pattern Oriented Software Architecture: A System of Patterns*. John Wiley & Sons.

Sayers, C. 1999. *Remote Control Robotics*, 1st ed. Springer Verlag.

Schaaf, M. and Maurer, F. 2001. Integrating Java and CORBA: A programmer's perspective. *IEEE Internet Computing*, 5:72–78.

Sheridan, T. 1992. *Telerobotics, Automation, and Human Supervisory Control*. Cambridge, USA: MIT Press.

ttcp Web Page. 2003. Available in. http://ftp.arl.mil/∼mike/ttcp.html

Whatis Web Page. 2003. Available in: http://whatis.techtarget.com/

Alaa M. Khamis received his B.Sc. degree with "Distinction with Honor's Degree" in Electrical Engineering from Alexandria University, Egypt, in 1993, the M.Sc. degree in Electrical Engineering from Suez Canal University, Egypt in 1998, and is currently pursuing a Ph.D. degree in Industrial Technology, at Carlos III University of Madrid, Spain. From 1995 to 1999, he was a teaching assistant in the Engineering Sciences Department at Faculty of Petroleum and Mining Engineering, Suez Canal University. In 2001, he was selected as distinguished scholar in the DAAD-Programme "International Quality Network" (IQN) to share in research projects related to remotely controlled experiments at Faculty of Applied Science, FH Ravensburg-Weingarten, Germany. His present research is primarily focused on Internet robotics.

Francisco J. Rodríguez is an Associate Professor of System Engineering and Automation at Carlos III University of Madrid, Spain. He received his Ph.D. degree in Artificial Intelligence and Robotics from the Polytechnic University of Madrid, Spain, in 1993. His current research interests are in robotics, online robots and intelligent systems.

Miguel A. Salichs is a full professor and head of the Systems Engineering and Automation Department at the Carlos III University of Madrid. He received Electrical Engineering and Ph.D. degrees from Polytechnic University of Madrid. His research interests include cognitive architectures, personal robots and multimodal human-robot interaction. He is a member of the Policy Committee of the International Federation of Automatic Control (IFAC).