Research article

Controlling an Internet-enabled arm robot in an open control laboratory

S.H. Yang
X. Zuo and
L. Yang

The authors

S.H. Yang is based at the Computer Science Department, Loughborough University, Loughborough, Leicestershire, UK.
X. Zuo is based at the Automation Research Institute, Petroleum University, Beijing, People's Republic of China.
L. Yang is based at the Computing Department, Business School, University of Derby, Derby, UK.

Keywords

Internet, Robotics, Control systems

Abstract

Internet-based robotic systems have received much attention in recent years. A number of design issues are essential for designing this new type of robotic systems. This paper addresses the Internet time delay, the user interface design and concurrent user access for an Internet-enabled arm robot. The implementation and application of the Internet-enabled arm robot in an open control laboratory has been illustrated as a case study.

Electronic access

The Emerald Research Register for this journal is available at
www.emeraldinsight.com/researchregister

The current issue and full text archive of this journal is available at
www.emeraldinsight.com/0144-5154.htm

Assembly Automation
Volume 24 · Number 3 · 2004 · pp. 280–288
© Emerald Group Publishing Limited · ISSN 0144-5154
DOI 10.1108/01445150410549773

1. Introduction

Internet-based robotic systems have received much attention recently. When we see the Internet as an infrastructure on which to build a robotic system, its attraction is three-fold. First, Web browsers can provide a nice human interface for a robotic system because the browsers can display various media including hypertext, moving images, sounds, and three-dimensional graphics as well as handling interactive operations of the media. Second, the hypertext transfer protocol (HTTP) can be a standard communication protocol of a tele-robotic system since robots connected to the Internet can be accessed from any Internet site via the protocol. Third, it becomes possible to use various robotic hardware/software components distributed over the Internet together to accomplish a single mission.

Therefore, it is natural to consider developing robotic systems on the Internet. Many robotic systems have been developed on the Web, despite the fact that the latency of the Internet is unpredictable. In the short history of Internet-based tele-operation, a number of experiments have been conducted: the Mercury Project (Goldberg *et al.*, 1999) was presented in 1994, the TeleGarden (Goldberg and Santarromana, 1997) was developed in June 1995 and the mobile robot system (Chen and Luo, 1997), which can be controlled via a Web browser, was reported in 1997. Another example is the Bradford Robotic Tele-scope (Baruch and Cox, 1996), through which the WWW users can look at an image taken from an observation with the telescope and compare it with one taken from a star database held at NASA. The University of Essex (Yu *et al.*, 2000) built a Web-based mobile robot. An arm robot has been put on the Internet for students to demonstrate remote control in our laboratory (Yang *et al.*, 2002). However, most of these projects rarely address the problems of interface design, Internet latency, and concurrent user access for the Internet-enabled robot.

This paper describes our experiences in dealing with the Internet latency and concurrent user access and in building an interactive Web interface for operating and managing an Internet-enabled arm robot in an open control laboratory. This paper is structured as follows. Section 2 presents the challenges in the design of Internet based control systems. Experimental system architecture is given in Section 3. The selections of an open-loop control structure and a dual-rate control structure to deal with the Internet latency problem are presented in Section 4. A multimedia user interface is explained in Section 5. The method of dealing with the concurrent user access is described in Section 6. The implementation of the Internet-based control system for our arm robot and its application in our open control laboratory are introduced in Sections 7 and 8. Finally Section 9 concludes this paper.

2. Challenges of Internet-based control

The introduction of the Internet into control systems has introduced a number of challenges. Three of them are discussed in this section.

2.1 Internet transmission delay

The transmission latency is one of the main differences between Internet-based control and other tele-operation. Most tele-operating systems are based on private media, by which the transmission delay can be well modeled.
The Internet in contrast is a public and shared resource in which various end users transmit data via the network simultaneously. The route for transmission between two end points in a wide area is not fixed and traffic congestion may be caused when too many users traverse the same route simultaneously. The transmission latency of the Internet is difficult to model and predict. The reasons why the variable time delay occurs are as follows:

- network traffic changes all the time because multiple users share the same computer network;
- routes or paths of data transmission decided by the Internet Protocol (IP) are not certain. Data are delivered through different paths, gateways, and networks whose distances vary;
- large data are separated into smaller units such as packets. Moreover, data may also be compressed and extracted before sending and after receiving; and
- using TCP/IP protocols, when error in data transmission occurs, data will be retransmitted until the correct data is received.

To reduce the effect of the time delay, some systems try to extrapolate forward environmental information and manipulate states in time. Some works use local simulated manipulations or simulators (Han *et al.*, 2001) to assist in controlling the remote devices. Selecting a proper control structure promises another way to overcome the time delay problem (Yang *et al.*, 2004a).

2.2 Web user interface design

The central design objective for a Web-based user interface in Internet-based control is to enable the operator to appreciate more rapidly what is happening in the robots and to provide a more stimulating problem-solving environment outside the central control room. It should be borne in mind that media available in the Internet environment, outside the central control room, will be very much limited as compared to the central control room.

Technologies from the areas of "multimedia" and "Virtual Reality" show considerable potential for further improving the human-computer interfaces used in control technology (Hori and Shimizu, 1999). Different media can transmit certain types of information more effectively than others and hence, if carefully chosen, can improve operator performance (Alty, 1999). Figure 1 shows the features of various media. Choosing the best media for different interface tasks and minimizing the amount of irrelevant information in the interface are two main guidelines in the user interface design.

2.3 Concurrent user access

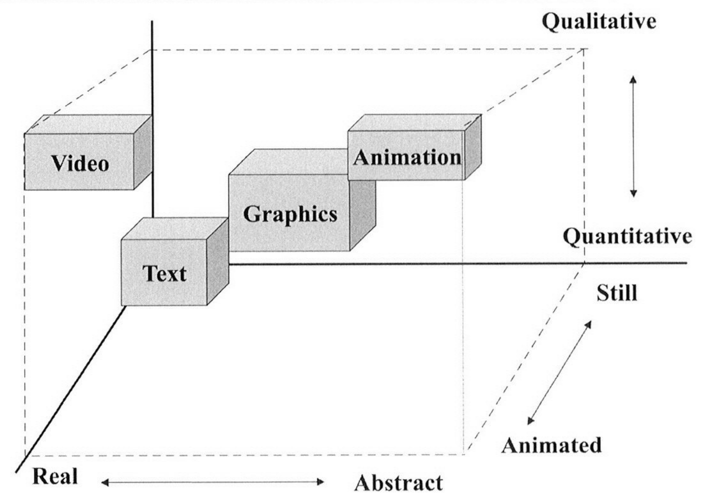
Compared with the traditional local control system, the special features of the Internet-based control system are multiple users and the uncertainty about who the users are, how many users are there, and where they are. In the Internet-based control system, the users cannot see each other, or may never have met. It is likely that multiple users may try to simultaneously control a robot. Coordination among multiple users becomes very important. Some mechanism is required to solve control conflict problems between multiple users, and coordinate their operations.

3. Experimental system architecture

3.1 System layout

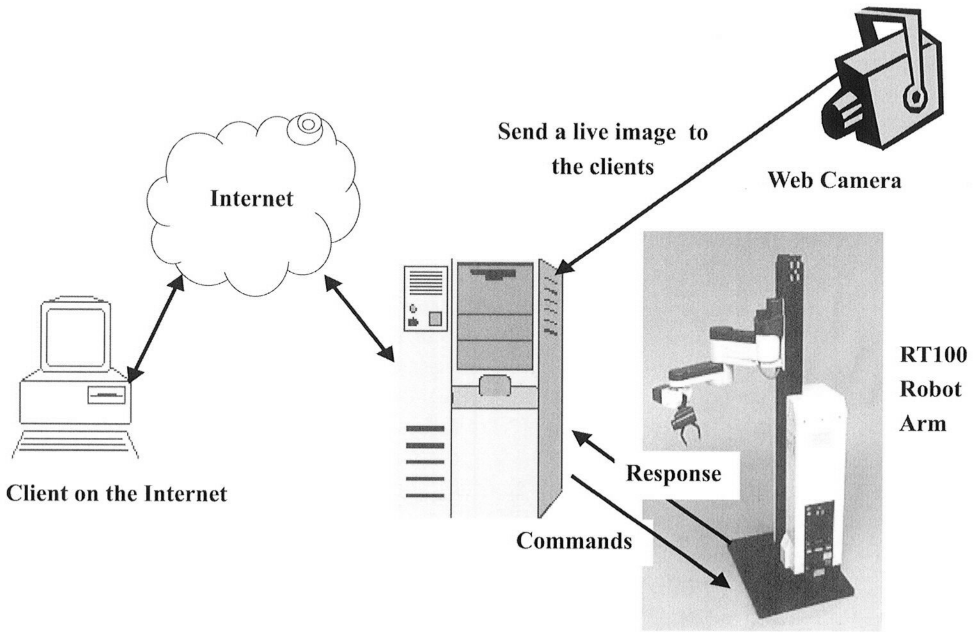
The hardware structure of the Internet-enabled arm robot system is depicted as Figure 2. On the

Figure 1 Web user interface design



Assembly Automation
Volume 24 · Number 3 · 2004 · 280–288

Figure 2 Experimental system layout



local side, it consists of a physical arm robot, a Web camera, and a server machine The physical arm robot is connected with the server machine through the RS232 link. The server machine comprises not only the Internet server, but also functions as a local control system for the arm robot. The server implementation is mainly based on the Apache HTTP server. The camera is connected to the server in order to provide a visual feedback on the robot status. On the remote side, Web clients use a 3D interactive interface to operate the arm robot. An arm robot simulator has been employed in the interface to provide a highly flexible and direct way to manage the robot operation. Macromedia Flash, a multimedia technology, is used in the development of the simulator. Details are given in the next section.

3.2 Three-tier architecture

A three-tier architecture is used to implement the above system, as shown in Figure 3. On top (left-hand side of figure) is the client tier, using a zero-cost Web browser to interact with the rest of the system. The middle tier is the Apache HTTP server, which is in charge of communication between the other tiers and provides Internet services, such as processing HTTP requests and formulating responses. The PHP scripting language has been chosen as the middle-tier scripting language. PHP is particularly suited for Web database applications because of its integration tools for the Web and database environments. Furthermore, the flexibility of embedding scripts in HTML pages enables easy integration with the client tier. The application tier is at the lowest level, which consists of a Microsoft Access database for user management and a real-time Java program for controlling the RT100 arm robot. The Java program communicates with the Web server through TCP/IP sockets, and works independently. The Access database is used to log the users' details such as logging-in and logging-out times, user Web addresses etc.

4. Control structure and Internet latency

As discussed in our previous work (Yang *et al.*, 2004a), selecting a proper control structure can make a great contribution towards overcoming the time delay problem. The straightforward control structure over the Internet allows the operator located in the remote site to send control commands (desired input) to the controller and robot located in the local site. The structure is shown in Figure 4. In order to monitor both the performance of the controller and the status of the robot, the measured output and/or some visual information are required to feedback to the operator at the remote site. Because the Internet is excluded from the closed loop and the controller is located at the same location as the robot, the Internet transmission delay will not affect the local performance of the control system. Obviously the Internet transmission delay can affect the transfer of the desired input and output between the operator site and the controller site.

Figure 3 The three-tier architecture

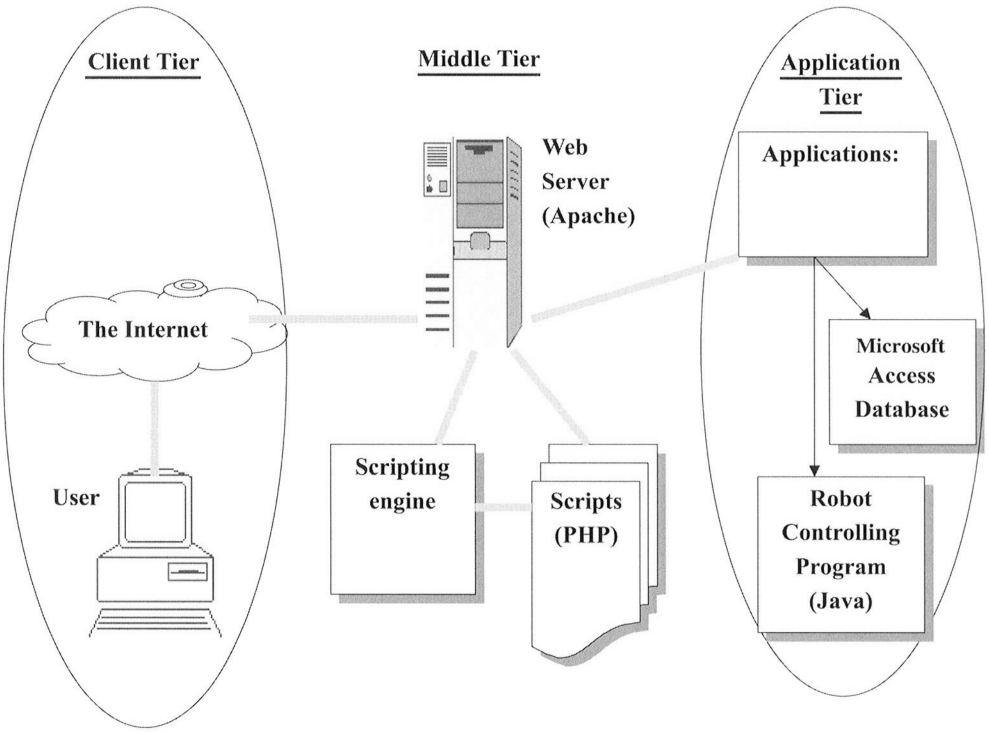


Figure 4 Control structure with the operator located remotely

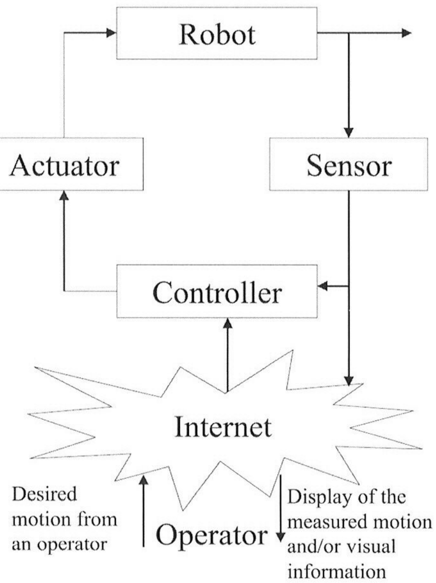
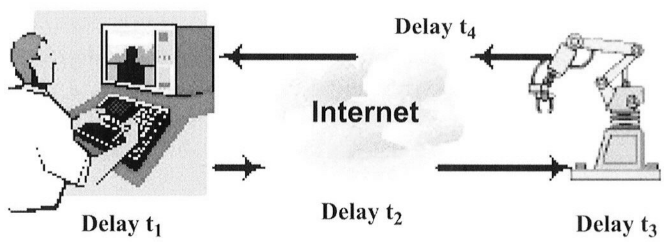


Figure 5 The block diagram of the Internet-based robot control



Some measures must be taken to compensate for these effects.

The block diagram of the Internet-based robot control system, shown in Figure 4, can be represented as shown in Figure 5. The total time to perform an operation, per cycle, is $t_1 + t_2 + t_3 + t_4$, where the four types of time delay are:

(1) t_1 – time delay for making a control decision by the operator;
(2) t_2 – time delay for transmitting the control command from the operator to the robot;
(3) t_3 – execution time for the robot and the local controller to perform a primitive action; and
(4) t_4 – time delay for transmitting the information from the robot back to the operator.

If each of the four time delays is a constant, robot control over the Internet has a constant time delay. Unfortunately, t_2 and t_4 are usually unpredictable. Luo and Chen (2000) have repeatedly tested the transmitting efficiency of the Internet by sending 64 bytes data continuously from their Web server to different remote servers. The resulting statistics show that the Internet not only contains serious and uncertain time delays but also causes data loss. The long transmission delay may result in remote control failures in a complex task or, more seriously, endanger the robot and its surroundings.

Under the open-loop control shown in Figures 4 and 5, the operator makes the control decision based on the video feedback from the robot. The robot is designed to reject any further control command, before the ongoing task is completed. Therefore, the time delays t_2 to t_4, can slow down the execution of the operation cycle, but they do not endanger the robot and its surroundings.

The bilateral control structure shown in Figure 6 is required if a remote controller is essential to the robot system for achieving a better control performance. There is a local controller in the robot side linked with a remote controller in the operator side through the Internet in the bilateral control structure. Usually the local controller is running at a higher frequency and responsible for the regulation of the normal situations. Once the performance of the local controller is degraded due to the disturbance from the environment and/or other reasons, the controller in the remote site is put in use for tuning the parameters and/or changing the desired input for the local controller. The remote controller is running at a lower frequency in order to reduce the communication load and increase the possibility of receiving data on time from the Internet. We denoted the local controller as the fast controller and the remote controller as the slow controller. A dual-rate control loop is shown in Figure 7.

The two sampling intervals for the fast and slow controllers, T_{remote} and T_{local}, are chosen as:

$$T_{\text{remote}} = nT_{\text{local}} \quad n \in \{2, 3, 4, \ldots\}$$

Figure 6 Bilateral control structure

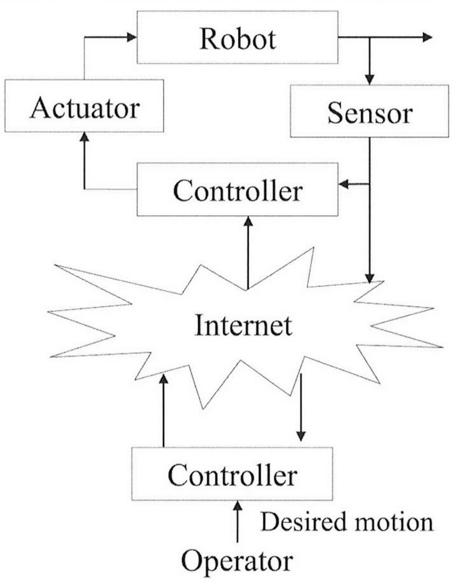
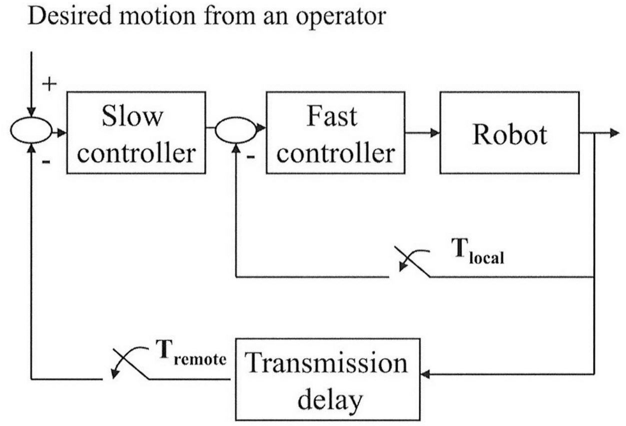


Figure 7 Dual-rate control loop

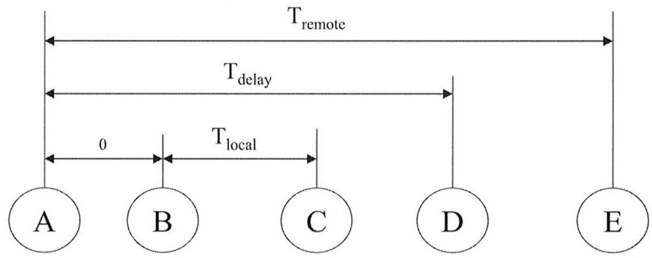


There are two cases that are involved in the dual-rate control scheme.

Case 1: $T_{\text{delay}} < T_{\text{remote}}$. The time scheme of the case 1 is shown in Figure 8. Assume that the time delay between the nodes A and B being 0 because the fast controller is located in the local site and the local network transmission delay can be ignored. T_{local} represents the time taken by the fast controller at the local site. If the transmission delay T_{delay} is less than the control period of the slow controller T_{remote}, as shown in Figure 8, there will be no data loss during each sampling interval of the slow controller in the remote site. Therefore the transmission delay has no influence on the remote controller.

Case 2: $T_{\text{delay}} \geq T_{\text{remote}}$. The time scheme of the case 2 is shown in Figure 9. Since the transmission delay is greater than the sampling interval the sample will be delayed to arrive at the slow controller in the remote site after the next control instant. A compensator must be employed in this

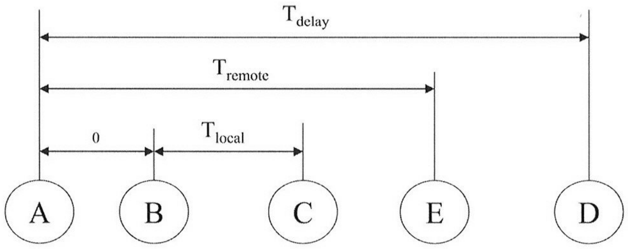
Figure 8 Time scheme of dual-rate control for case 1



A: the instant at which the sensor sends data to the controllers;
B: the instant at which the fast controller receives the data;
C: the instant at which the fast controller sends out the control action;
D: the instant at which the slow controller receives the data;
E: the instant at which the control action is calculated and sent out by the slow controller.

Assembly Automation
Volume 24 · Number 3 · 2004 · 280–288

Figure 9 Time scheme of dual-rate control for case 2



A: the instant at which the sensor sends data to the controllers;
B: the instant at which the fast controller receives the data;
C: the instant at which the fast controller sends out the control action;
D: the instant at which the slow controller receives the data;
E: the instant at which the control action is calculated and sent out by the slow controller.

case. A model based predictive control compensator for an Internet based control system can be found from our recent publication (Yang *et al.*, 2004b).

5. Design of the multimedia-based user interface

The central design objective for the Internet-enabled arm robot control system is to enable Web clients to directly manipulate the arm robot and to appreciate more rapidly what is happening in the physical arm robot located remotely from the users. There are a number of general user-centred interface design principles available (Alty, 1999; Hori and Shimizu, 1999) such as "user in control", directness, consistency, feedback and simplicity. This project fully follows these general principles and focuses on providing a flexible, direct and easy-controlled 3D interface for the Internet-enabled arm robot.

Macromedia Flash has been chosen for the development of the interface. Flash allows users to create full screen animation (flash movie) and interactive graphics. The flash file is saved in a vector-based format, which results in extremely compact files. This feature enables the flash-based animation to be embedded and quickly downloaded in a Web page.

The interface for our arm robot is shown in Figure 10, which provides not only individual control buttons to initiate various actions, like starting, stopping, initialising, and logging-out, but also a dynamic simulator for virtually operating the physical arm robot. With this simulator users can control the arm robot off-line, choose the best position for the arm, and then send the automatically generated control command to the physical arm robot by pressing the "send data" button. The actual position of the arm robot can be viewed from the Web camera visual feedback and the graphical feedback is shown on the left hand side in the interface.

6. Concurrent user access

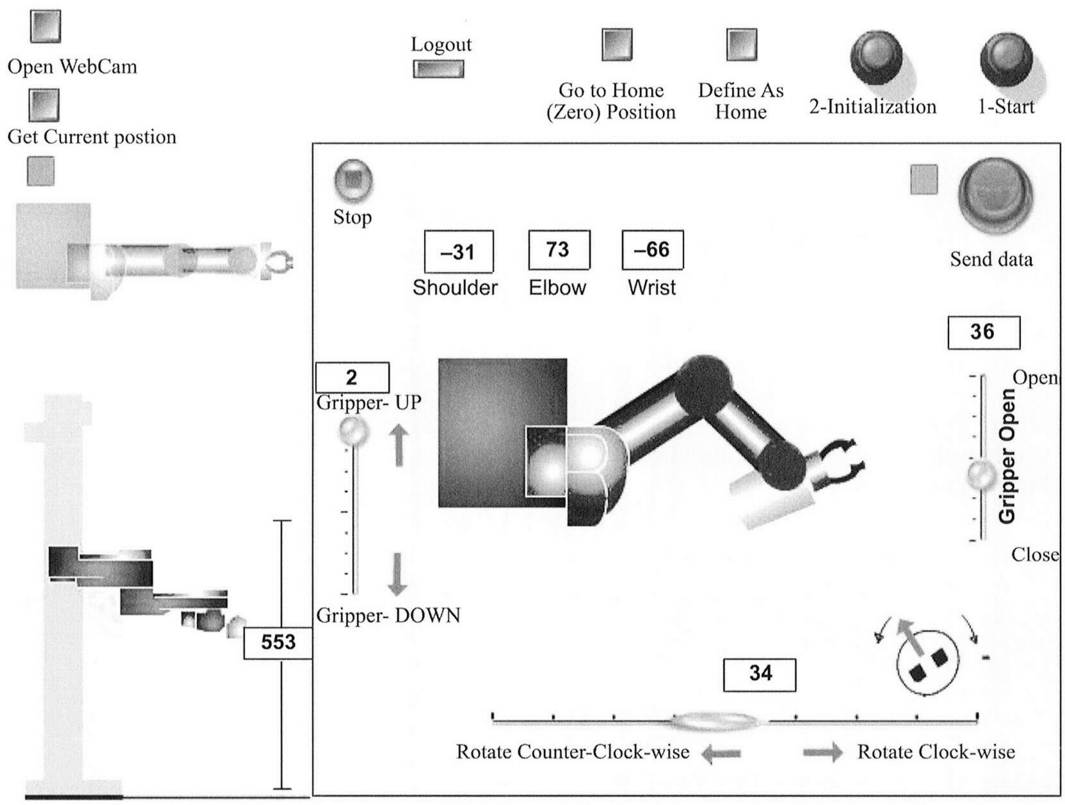
If authorized users have the same opportunity to fully control the robot some problems could arise Assume that user 1 wants to move the robot arm to the left, and then the corresponding command is sent to the local controller through the Internet. If user 2 sends another control command to the robot before the first control action is completed, the second control command will overwrite the first control command, even if it is unsuitable for the current situation. The arm of the robot might fluctuate from point to point, due to the inconsistent multiple commands.

There are three proposed steps for dealing with multi-user concurrent access:

- Assign users with different priorities, for example the integer values 0 to 6. The user with a high priority can immediately overwrite the commands issued by users with lower priorities. Generally, the user account provides an easy way to identify a user. When the user logs in, a dynamic identity (ID) will be generated for the client. The control command issued by the user will be combined with this ID, so that the local controller can identify the command owner, and can work out the priority of the owner, and decide whether the command is acceptable or not.
- After a new command is accepted, the robot will be blocked for a certain period of time and will refuse to accept any further command from users with equal or lower priorities. The time will vary from case to case. Normally the time could be chosen as the time constant of the system, to ensure that the command has been fully executed before the new command is carried out.
- Allow only a single user to operate the robot in the tuning operation. There are two typical operation scenarios. One is the tuning operation, the other is the normal operation. The tuning operation covers maintenance and fault recovery. Normally, the tuning operation is handled by a senior user. In this case, the operation should be continuous, exclusive, and should not be interrupted. Therefore, the command service will hold for a single user only – until this user logs out or releases the right to another. The other users only can monitor operations in this case.

Assembly Automation
Volume 24 · Number 3 · 2004 · 280–288

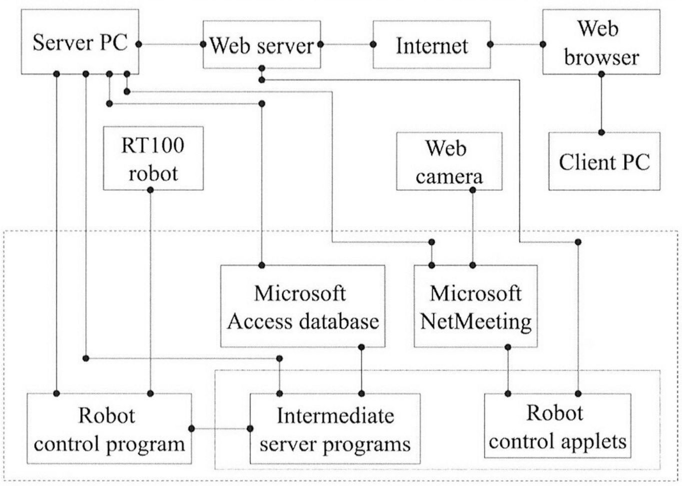
Figure 10 Web interface for the arm robot designed with flash



7. Implementation

The implementation of the experimental system is shown in Figure 11, which shows the interaction between the system components. For example, a remote operator uses a Web browser at the client PC to invoke the robot control applets from the Web server through the Internet. The robot is controlled by the robot control programs. The control commands are received from the robot control applets through the intermediate server programs. A number of Web technologies including HTML, HTTP, PHP, TCP/IP, and Java have been used in the implementation of the Internet-enabled arm robot control. HTML is used to build a front-end including a welcome page, a logging-in page, and an interface frame. The interface was established by using Macromedia Flash and embedded in the HTML interface page.

Figure 11 Implementation of the experimental system

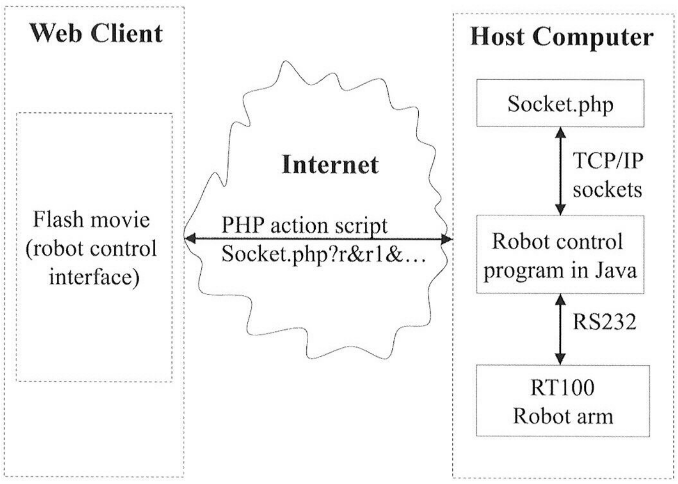


The communication between Web clients and the Web server follows the HTTP communication protocol. A Java program located in the server machine directly controls the arm robot. TCP/IP sockets are used to build communication between the Java program and the Web server through a PHP program Socket.php. The PHP program is also used to operate the Microsoft Access database for user management. The flow of information is shown in Figure 12. The control commands collected from the Web client interface are sent to the PHP server by using the following PHP action script:

```
LoadVariableNum (
"http:// pc-robot.lboro.ac.uk/
Socket.php?Para1 =" +Para1 +
"&Para2=" +Para2 + ...);
```

Assembly Automation
Volume 24 · Number 3 · 2004 · 280–288

Figure 12 Information flow of the arm robot control



where "hhtp://pc-robot.ac.uk" is the Web address of the PHP file Socket.php, Para1 and Para2 are two parameters that are required to be transferred to the PHP server and then to Socket.php.

In the Socket.php program the TCP/IP sockets are used to communicate with the Java program, which controls the arm robot. The PHP action scripts for TCP/IP communication includes:

```
$connection = fsocketopen(host,
  port, timeout);
fputs($connection, "move
  $Shoulder");
...
fclose($connection);
```

The first line is used to build the TCP connection to the remote host on the specific port. The "timeout" indicates the length of time in seconds to wait before timing out. The second line transfers the control command "move $Shoulder" to the host. The third line closes the connection. User information, including who they are, what they did and when, have been automatically logged in a database for management purposes. In order to display the visual feedback from the Web camera the NetMeeting ActiveX control has been added to the client interface. The visual window is popped out/off by pressing the "OpenCam" button in the interface, as shown in Figure 10.

8. Application in an open control laboratory

Hands-on laboratory exercises are playing an extremely important role in the further and higher education. The Open Control Laboratory allows more students in further and higher education institutions to remotely access limited experimental equipment and instruments over the Internet. Students are able to carry out various experimental projects from simple operations to comprehensive problems such as control system design problems. In detail, an open control laboratory can bring the following benefits to educational communities:

- enable multiple higher education institutions to share the same experimental rig if it is too expensive to own, and allow all students access to the latest technologies and resources;
- enable distance-learning students to remotely carry out laboratory work;
- enable students in further education institution to remotely use university laboratory resources, which would never be available in their own institute due to financial reasons; and
- overcome accessibility issues for students with disabilities and provide an alternative, efficient and safe method to carry out laboratory work.

Applying the Internet-enabled arm robot in the open control laboratory is straightforward (He, 2003). The following functions have been implemented for the remote users:

- initializing the robot: move the robot to the initial state;
- reading the robot: display the current state of the robot;
- going home: move the robot to a set of pre-defined position;
- moving the robot: move the robot to a particular position; and
- stopping the robot.

The following functions have been implemented for the local users:

- directly controlling the robot;
- enabling remote control;
- disabling remote control;
- re-arranging the user priorities; and
- monitoring the actions of the remote users.

An open control laboratory is required to open 24 h a day and 7 days a week. Obviously the safety issues and the fault auto-recovery features should be highly emphasized in the design of an open control laboratory. A safety interlock system has been added into the local control server but it stays independent from the local control system in case that the local control system goes wrong. The safety interlock system tries to keep the arm robot in a safe status under any circumstance. It allows students to make mistakes during the experiment, including switching off the Internet connection without any advance notice. The fault auto-recovery is implemented by triggering the "reset" button after the safety interlock system is activated.

Assembly Automation
Volume 24 · Number 3 · 2004 · 280–288

9. Conclusions

This paper has presented the challenges in the design of Internet-based control systems for an Internet-enabled arm robot. Internet latency, user interface design and concurrent user access have been discussed, respectively, for the Internet-enabled arm robot. The open control structure and the dual-rate control structure are presented to overcome the Internet time delay problem. Various Web technologies have been used in the arm robot case study, including HTML, HTTP, PHP, TCP/IP, and Java. Communication is based on PHP pages and TCP/IP sockets. In order to provide a flexible, direct and easy-use interface for Web clients, a multimedia based dynamic simulator of the arm robot is embedded in the interface. An open loop control structure is implemented in the case study. Users control the arm robot off-line first through operating the simulator, and then send the automatically generated control commands to the physical arm robot by pressing a confirmation button. Our experimental work shows that operating the arm robot through the simulator at the client side offers a greater opportunity for the users to improve the operability of the physical arm robot over the Internet. The Internet-enabled arm robot control system represents a straightforward application in an open control laboratory for higher and further education institutions.

References

Alty, J.L. (1999), "Multimedia and process control interfaces: signals or noise?", *Transaction of Inst MC*, Vol. 21, pp. 181-90.

Baruch, J.E.F. and Cox, M.J. (1996), "Remote control and robots: an Internet solution", *IEEE Computing Control Engineering Journal*, pp. 39-44.

Chen, T.M. and Luo, R.C. (1997), "Remote supervisory control of an autonomous mobile robot via world wide web", *Proc. IEEE Int. Symposium on Industrial Electronics*, Vol. 1, pp. 60-4.

Goldberg, K. and Santarromana, J. (1997), *About the Tele-Garden*, available at: http://telegarden.aec.at/html/intro.html

Goldberg, K., Gentner, S., Sutter, C. and Wiegley, J. (1999), "The mercury project: a feasibility study for Internet robots", *IEEE Robotics and Automation.*

Han, K.H., Kim, S., Kim, Y.J. and Kim, J.H. (2001), "Internet control architecture for Internet-based personal robot", *Autonomous robots*, Vol. 10, pp. 135-47.

He, H. (2003), "Control robot over the Internet by wireless PDA", MSc dissertation, Loughborough University.

Hori, S. and Shimizu, Y. (1999), "Designing methods of human interface for supervisory control systems", *Control Engineering Practice*, Vol. 7, pp. 1413-19.

Luo, R.C. and Chen, T.M. (2000), "Development of a multibehaviour-based mobile robot for remote supervisory control through the Internet", *IEEE Transactions on Mechatronics*, Vol. 5, pp. 376-85.

Yang, S.H., Chen, X. and Hinde, C.J. (2002), "Development of multi-client robot control through the Internet", *Proceedings of the 4th Asian Control Conference*, Singapore, pp. 549-54.

Yang, S.H., Zuo, X. and Yang, L. (2004a), "Control system design for Internet-enabled arm robots", *Lecture Notes in Computer Science*, Vol. 3029, pp. 663-72.

Yang, S.H., Chen, X., Tan, L.S. and Zuo, X. (2004b), "Time delay and data loss compensation for Internet-based process control systems", *Transactions of the Institute of Measurement and Control* (in press).

Yu, L., Zhou, Q. and Hu, H. (2000), "A framework for Web-based telerobotic systems", in Hu, H. and Wu, M.H. (Eds), *Proceedings of the 6th Chinese Automation & Computer Science Conference in the UK*, 23 September, Loughborough, pp. 119-24.