CASE STUDIES IN WEB-CONTROLLED DEVICES AND REMOTE MANIPULATION

T. Sobh, R. Mihali, A. Rosca, B. Ghimire, K. Vovk, G. Gosine, P. Batra, A. Singh, S. Pathak

Tarek M. Sobh, University of Bridgeport, School of Engineering and Design
169 University Avenue, Bridgeport, CT 06601, U.S.A.
Phone: (203) 576-4116, Fax: (203) 576-4766
sobh@bridgeport.edu, http://www.bridgeport.edu/~sobh

Abstract: *The concept of distance learning has been more and more articulated during the past few years and is expected to shortly turn into a practical education system within current high level learning institutions. The chances are that distance learning would transparently extend colleges and institutes of education, and could plausibly overtake and turn into a preferred choice of higher education, especially for adult and working students. In a supportive effort towards faster distance learning implementation and consideration, we are presenting through this material a sequence of projects that have been developed at University of Bridgeport and can serve very well in the process of distance learning education ranging from simple "hobby" style training to professional guidance material. The projects have an engineering / laboratory flavor, are part of an ongoing work of the faculty and students of the Computer Science and Engineering Department of the university, and are being presented in an arbitrary order, topics ranging from vision and sensing to engineering design, scheduling, remote control and operation.*

Keywords: *- remote automation, manipulation, web controlled devices*

1. INTRODUCTION

Distance learning is being envisioned as an education mechanism that is more accessible, comfortable and closer to the learning individual.

The main idea in *e-learning* is to build adequate solutions that could assure educational training over the Internet, without requiring a personal presence at the degree offering institution. For example, being able to obtain a Bachelor's degree in Computer Engineering from an accredited institution while residing thousands of miles away from it and actually never seeing it, except maybe for the graduation ceremony.

The advantages are immediate and of unique importance, to enumerate a few: Scholarship / education costs can be reduced dramatically, both from a student's perspective and the institution's (no need for room and board, for example); The usually tedious immigration and naturalization issues that are common with international students are eliminated; The limited campus facilities, faculty members and course schedules an institution can offer are no longer a boundary; Working adults can consider upgrading skills without changing lifestyle

The idea is to use the large information band-width that the Internet can offer and expose a variety of educational materials online. A course could for example be offered online through a comprehensive list of multimedia solutions ranging from an online course book and online homework submissions to live video conferencing, auditing and examinations. The methods of creating an online course are endless and would have to be chosen adequately, considering issues such as security/reliability of training, educational expectations such as work speed improvement (e.g. math and programming courses), physical dexterity (e.g. mechanical, medical courses), etc. Some courses might require specialized hardware input and output solutions, such as haptic devices combined with virtual reality gear, etc.

We are concentrating on some of the aspects of the mentioned problems and present a few distinct implementations that could help their solution.

2. MOBILE ROBOT CONTROLLED BY A PHONE

Mobile Robots have numerous applications: unmanned exploration, land mine removal, energy plants and manufacturing factories. We introduce

a cost-effective robot. With the introduction of video cell phones it will be possible for the user to see the robotic movement in real-time and possibly perform educational exercises using a simple interface at a distance. Examples include "calling" a robot on the way home from work and have it do various jobs like vacuuming the home and sprinkling the garden. This could also be done by logging on to a web server via an internet connected device and sending signals to the robot [1], or directly sending specific commands to the robot through cellular phones to a wireless (or cellular) server at home (with or without a web connection) (figure 1).

Figure 1: *An application of a Robot with a phonechip*

We focus on communication with the robot. The main objective was to make the robots more user friendly and be able to communicate with them as we can communicate with other people around the globe. Thus we decided to design a simple prototype of a robot that can be controlled via phone - Phonebot. We considered a design based on a phone chip

Hardware/Software and Test Equipment Used

Hardware Requirements: Flex 10K70 Chipset, Talrik II Robot (including the Servo motors and Sensors), TelePhone Set, Teltone's M-8870-01 Chip, SPST 90-2323, 5V relay switch from RadioShack, 3.58MHz Crystal Oscillators, Variable Power Supply, Resistors, Capacitors and Diodes

Software Requirements: Altera's MaxPlus II, OrCAD, HTML, CGI/ASP, Java, Matlab,

Test Equipment: Oscilloscope, Logic Analyzer, Multimeters, Nerd-Kit

Implementation

A call is placed to the Phonebot using either Plain Old Telephony System (POTS) or a Personal Communication System (PCS), or by using the telephony server via Internet. Once the Phonebot receives the ring, a ring detector circuit detects it and the call is completed by establishing a connection between the phone chip and the FPGA chip (figure 2).

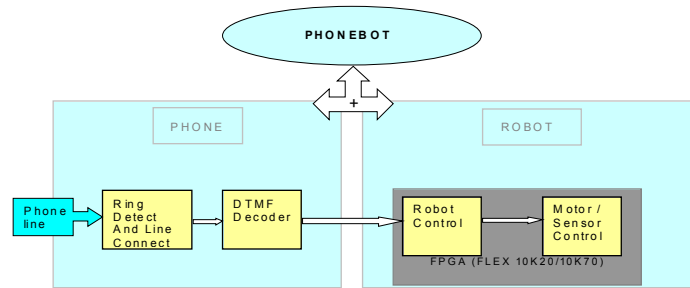The project was implemented using a top-down process:

Figure 2: *PHONEBOT – Basic Block Diagram*

- sending a signal through the phone line a) Ring Detect and connect phone line; b) DTMF decoder

- robot control with the FPGA device: c) Clock Division (VHDL program); d) Ring Detect (VHDL program); e) Motor Control (VHDL program); f) Robot Control (VHDL program)

The final part of the project involved assembling various parts of Phonebot and combining the different VHDL modules.

a) Ring Detect and Phone Line Connection

When the phone rings, the telephone company is sending a ringing signal, which is an AC waveform. The common frequency used in the United States is 20 HZ and in Europe it is typically 25 Hz and it can be any frequency between 15 and 68 Hz. Most of the world uses frequencies between 20 and 40 Hz. The voltage at the subscribers end depends upon the loop length and number of ringers attached to the line; it could be between 40 and 150 Volts.

The telephone line has only DC (-48V) and/or small signal AC (audio). In the circuit shown in Figure 3, capacitor C1 blocks the DC and the voltage divider circuit obtained from the R3 and R2 resistors prevent the low level AC from having any effect on the circuit.

Specifications

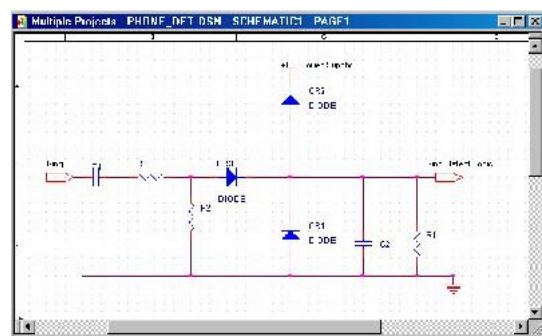C1 = 1 uf, CR1, CR2, CR3 = 1N914, C2 = 10 uF, R1, R3 = 100K, R2 = 10K

Figure 3: *Phone detect circuit*

When the telephone rings, it brings about 90V RMS of AC at 20Hz. When the telephone rings, the capacitor C2 is charged. Diodes CR1 and CR2

guarantee that the output (ring detect logic) does not exceed the power supply levels and prevent any damages to other circuits driven from its output. Since C2 and R1 have the time constants of 1s, the output goes low for 1second after the ring stops. This pulse is to be detected and used for connecting the telephone circuit by the FPGA chip [2].

b) DTMF Decoder

Dual Tone Multi Frequency (DTMF) signals are used for speed dialing and replace the conventional rotary dialing system. These signals correspond to the digits on the dial pad of any modern touch-tone phone. Each of those touch-tones is constructed with the combination of two different frequencies. This information can be utilized to find out which button was pressed on the keypad and can be used for various applications. The construction of the signals according to different frequencies is shown in table 1:

Table 1: Signals related to different frequencis

		High Frequency Values (Hz)			
Low Frequency Values (Hz)		1209	1336	1477	1633
	697	1	2	3	A
	770	4	5	6	B
	852	7	8	9	C
	941	*	0	#	D

These frequencies can be decoded using precise filters and then we can decode which digit was pressed depending upon these decoded frequencies. We used the M-8870-01 DTMF decoder chip to decode this information. The chip uses a series of low pass and high pass filters to decode the frequencies. Then it uses a digital detection algorithm and a code converter to provide its output in the form of four binary output data [3]. This data is supplied to the FPGA chip for further use.

FPGA Control

For the controller we used the Altera's Flex 10K70 chip. The FPGAs contain arrays of logic cells, and enable designing real systems to operate at increasingly higher frequencies [4]. They have the ability to increase integration, to place more and more electronics in a chip and use all available gates within the FPGA, thereby providing cost-effective solutions. The most important factor for selecting this chip is that we can program and re-program the device while it is in a system. This provides us with a great flexibility for using the Phonebot. The Phonebot can be programmed within a few minutes to do many tasks. Once programmed the robot will have the "intelligence" to complete the requested task.

c) Clock Division Module

This module divides the 25MHz clock of Altera's University Board into slower clocks so that we can provide the precise timing for the servo-motors because these motors work at pulses in the order of milliseconds. The 25MHz clock is divided into 1Mhz, 100Khz, 10KHz, 10Hz and 1Hz.

d) Ring Detect Module

This module takes its input from the ring detector circuit and the DTMF decoder, provides output to the relay for closing or opening the telephone circuit. As soon as a ring is detected this module provides a 0V signal to the relay and thus closes the circuit. When it senses a DTMF code for digit 0, it opens the switch again. We had a small problem while disconnecting the phone. The FPGA device stored the code for digit 0 even after it was disconnected and for the same reason next time we dialed up, it would connect and disconnect on its own. This problem was resolved by resetting the signals in the FPGA device on the falling edge of the SB signal (SB signal goes high when there is a valid code detected by the DTMF decoder).

e) Motor Control Module

The Phonebot has two servo motors to aid its movement. First we had to hack these motors to create a DC gearhead motor. The information on hacking the servos was obtained from Mekatronix manual for the Talrik II robot [reference!]. The hacked servos work on Pulse Width Modulation (PWM). This module provides the robot with pulses at specific times. If we provide a pulse for 1 to 1.5ms, then the motor spins in one direction and if we provide a pulse for 1.5 to 2ms, then the motor spins in another direction. The pulses have to be issued in an interval of about 20ms. This module implements the generation of this timing.

f) Robot Control Module

This module programs the movement of the robot according to the detected DTMF signals. It determines the path of the robot and provides a signal for the servo motor control module controlling the direction and the duration of running the motors. For example, if we press 7, it moves the robot in forward direction for 4 seconds; if we press 8, the Phonebot moves in reverse direction for 4 seconds; and if we press #, Phonebot comes to a complete stop. We also added few bump switches to the Phonebot. These bump switches send a high signal to the FPGA when they are bumped [5]. Depending upon which bumper detects switch closures, the FPGA device determines where the collision occurred and issues a signal to the Phonebot to change the direction of its path. The final task in

the project involved assembling the different modules together. We had to be make a common ground for the FPGA device, the circuit we designed for the ring detection and the DTMF decoder [6]. The Altera board was not detecting the decoded signals produced by the DTMF decoder until a common ground was implemented. The main controller program was implemented in structural format combining the various components. There are several features that could be added to the Phonebot. Instead of the DTMF decoder, a voice encoder/decoder can be used. Similar to a DTMF decoder, the voice decoder would recognize certain frequencies and depending upon those frequencies, the Phonebot can be programmed to do certain tasks. This feature could also be used for security purposes. Using more sensors and a visual feedback would make the Phonebot more animated and will be able to carry out more tasks. Integrating with the Internet is another possibility that would add more access to the Phonebot. A movie showing the Phonebot can be seen at http://www.bridgeport.edu/cse/projects/phonebot/index.html, as well as the controlling software of the robot

3. INTERNET BASED SOFTWARE LIBRARY FOR THE SIR-1 SERIAL PORT CONTROLLED ROBOT

The idea of web based control has been always envision from the first days of networked computing. Being able to execute operations from remote locations [7], with only feedback of some sort, it is an active and desired choice in many fields, such as robotic manipulators. This project presents a complete web based control solution for a manipulator, thus completely exemplifying one of the tons of possibilities that remote automation encompasses.

A complex API for the control of the SIR-1 robot (figure 4) has been developed. Available functions include direct/inverse kinematics computation, serial port communication interfacing, and link speed control.
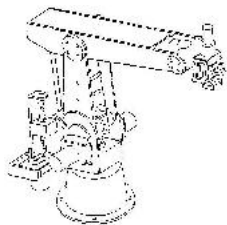

Figure 4: *The SIR-1 Robot*

The API can support an indefinite number of port connections and thus control a theoretically indefinite number of SIR-1 robots. For the purpose of this project, two robots with their respective controllers have been used.

Due to the high availability of serial ports on standard PCs, this API can be deployed virtually anywhere and in any environment, including the Internet, for almost any application.

Programming Generalities

The SIR-1 controller accepts commands from the serial interface as CRLF-terminated ASCII strings. Prior to issuing any command, a handshaking sequence has to be performed. Once the character ESC has been sent, the controller will respond with an 8-bit integer. The respective bits indicate the status of each link (1 for active and 0 for inactive). This is the only status update the controller will send throughout the process.

Commands are in the form: *<command char> <steps for link 1>, <steps for link 2>, ... <steps for link 6>*

For example, in order to move the first link 100 steps, the following string would be sent to the serial port:

M 100, 0, 0, 0, 0, 0 <CR><LF>. *In this case, the trailing zeros can be left out. Programming the controller via the serial port is thus reduced to assembling a command string in a buffer and sending it to the port character by character.*

Web Usable Enhancements Library Set
Steps-degrees conversion

In order to control link movement by degrees rather than steps, a simple proportional correspondence between the number of steps each link can move and the angle it covers has been implemented.

Inverse Kinematics

Inverse kinematics functionality has been implemented. Link movement is no longer controlled only by angle, but also by absolute rectangular coordinates.

Programming Platform

A Visual Basic port of the library is implemented. This allows for web deployment and robot control over the Internet.

Function Library Abstract

Below is a summary of all functions present in the library.

int SIR1_Handshake();

Initiates communication with robot. This function must be called before any other command can be issued.

int SIR1_MoveLink(link L, int degrees);

Moves one link a specified number of degrees. If L = G (the gripper is moved), the integer degrees specifies millimeters of gripper opening.

int SIR1_MoveRobot(int AlphaBase, int

AlphaShoulder, int AlphaElbow, int AlphaPitch, int AlphaRoll, int PercentGripper);

Moves all links a specified number if degrees. *void SIR1_HomeRobot();*

Sends robot to HOME position. No return value *int SIR1_GotoXYZ(int X, int Y, int Z, int PITCH, int ROLL);*

Moves the center of the gripper fingers to coordinates X, Y, Z, with a specified roll and pitch of the gripper segment. *void SIR1_SetSpeedLink(link L, SPEED S);*

Sets the speed of a link to a specified value (valid values are 0 to 7, of type SPEED defined above).

void SIR1_SetSpeedLinks(SPEED B, SPEED S, SPEED E, SPEED P, SPEED R, SPEED G);

Sets the speed of each link to a specified value (values are 0 to 7, of type SPEED defined at the top of **sir.h**).

int SIR_SetSpeedRobot(SPEED S);

Sets the speeds of all links to the specified value. *void SIR1_Pause(int TIME);*

Pauses robot for 1/100 * TIME seconds. No return value

int SIR1_LinkPosition(link L);

Returns the position of each link (in degrees) relative to the absolute system of coordinates defined for the robot. For the gripper the percentage of opening is returned.

void SIR1_SetPort(int port);

Sets the address of the port we are talking to *int SIR1_IsActive(link L);*

Checks to see if link L is active

Although the SIR-1 is a relatively simple robot, it can accomplish complex tasks, due to its high repeatability (0.6 mm according to the specifications). During tests, the robot was able to pick up and deposit a 9V battery, back and fourth, ten times in a row.

The project clearly demonstrates the endless possibilities of using such a robot as SIR-1 for web/ internet based remote automation through the implemented API, from pressing buttons, flipping switches or remotely controlling any other similar interfaces, to distance learning applications.

4. INTERNET CONTROLLED ROBOTIC VEHICLE WITH VIDEO FEEDBACK

The Internet-Controlled Robot is a robust and repeatable telerobotic vehicle that can be controlled from any Internet-enabled computer. In supervised telerobotics, an operator at a local site utilizes input devices and graphical visualization tools to command execution of a task at a remote site using a telerobotic vehicle.

The robot is controlled from a web page, where the operator handles the robot by using a graphical user interface. The operator can control each link of the robot separately, or move the robot to a given point in 3-D space be inputting the corresponding coordinates. The operator sees the movement of the robot on the computer screen in real-time via a live video camera broadcast. (Figure 5)
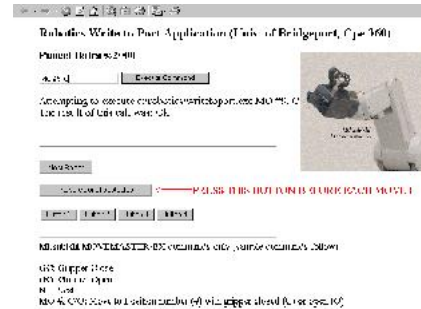

Figure 5: *Screenshot of the main interface page looks like (http://216.87.101.211:8080/satcontrol/aspexec/ commandrobot.asp)*

Such a telerobotic vehicle can be used for controlling a robot that operates in a hazardous or inaccessible environment. In addition, this vehicle offers the benefit that the operator does not have to be in the vicinity of the robot, but can be located anywhere throughout the globe. Practical applications of the system include remote-controlled operations in radioactive or toxic areas, which are dangerous for human beings, as well as remote reconnaissance and sample collection.

The designed robot features base rotation, shoulder, and elbow and wrist motion with a functional gripper along with two additional servos to provide different steering. The electronics and the servo controller are completely assembled. A host PC is used to issue positioning commands for the movement.

For the Internet control of the robot, we have created a distributed high performance client/server architecture [8]. In this architecture the client provides an easy control interface for the user and transmits control commands to the server over TCP/ IP, encoding the data before the transmission. The system architecture is based on the thin client model where the client performs only data validation and data transmission routines. The server receives and decodes control commands from the client. Once the transmitted command has been decoded, the server then performs inverse kinematics calculations on the data. This is necessary since the user inputs values of x, y, z coordinates via the client and they have to be transformed into angu-

lar values, which the robot's micro-controller can understand.

The server then sends robot control signals over RS-232 serial port to the robot's built-in microcontroller, which performs the required action as long as the requested position has been validated and confirmed to be within the robot's workspace. The user receives nearly instantaneous visual feedback from a video camera that is focused on the robot. The video signal from comes from a CCD Camera connected to the server via a Universal Serial Bus interface and has a resolution of 640 x 480 pixels. The signal is converted into streaming *RealVideo* format using RealProducer software, encoded at bitrate of 150 Kbps and streamed to the client real video player. Telerobots have the benefit of human cognitive and perceptual abilities, since they are operated by a human being, they can perform more efficiently in unfamiliar and dynamic environments. Telerobots such as the Internet-Controlled Robot can be thought of as a robotic device that synergistically combines human being and machine, and such systems will undoubtedly have a great contribution towards the progress of humanity in the future.

5. REMOTELY CONTROLLED MONITORING VEHICLE

The objective was to come up with a prototype for a low-cost monitoring vehicle that can be controlled from a remote location providing visual feedback to the remote operator A small-scale prototype of the described product can demonstrate the low cost and ease of setting up a device with an enormous number of potential applications including security surveillance, child monitoring, hostile territory exploration and even as a toy.

Hardware used: PCs (IBM compatible), Mitsubishi Movemaster EX robot, Generic RC car, Radio Frequency (RF) transmitter, Cables and connectors, Controller box for Mistubishi Movemaster EX, Wireless camera (for visual feedback),

Software Used: Operating Systems (Windows NT, Linux), Programming Languages (C++, Java, VB)

Description:

The two major issues included: 1. Issuing commands to the server from a remote machine; 2. Interpreting these commands and sending the appropriate signal to the car through the RF chip

Issuing Commands to the server from a remote machine

A java-based approach was approached. We wrote an RMI-based server that accepted a maximum of ten connection from the client software that was also written in Java. This design also allowed IP screening and other security features. The Java Server also had direct access to the local machine's serial port thus avoiding the need to write a separate DLL which proved to be a security hole by itself.

Interpreting the commands and sending the appropriate signal to the car through the RF chip

The major issue that arose at this point involved deciphering the commands and having the RF chip transmit the appropriate signals. To resolve this issue, we needed an FPGA controller, which would send a unique signal to the RF chip depending on the signal it received from the serial port. This approach was the most efficient but involved an expensive FPGA chip and expensive software, which was unavailable to us at development time.

The alternative solution we chose was to use a robot to decipher the signals from the server and manually manipulating buttons on the RF chip to send one of four signals (back, forward, left, right) to the vehicle. The availability of all the resources made us pick this as our option for the prototype.

The controller box for the Mitsubishi Movemaster-EX robot accepted commands as strings from a PC's serial port (RS-232). Since our client already had this capability we simply had to program the robot to teach it certain positions and send it commands to move to these positions in order to manipulate the controls on the RF-chip.

Potential applications of our web based system include: Security surveillance vehicle, Exploration of hazardous environments, Remote transportation of objects, Support for disabled individuals

6. INTERNET CONTROLLED SATELLITE TRANSPONDER (USING A REMOTELY CONTROLLED ROBOTIC MANIPULATOR)

The objective is to be able to control a satellite transponder from a remote location by using a robotic manipulator to mechanically change controls on the receiver or an UHF/infrared remote control of the receiver.

The reasons a robot was used to mechanically use the controller unit instead of hard-wiring the controller unit (or the UHF remote control) to the server were: (1) The process of hard-wiring can permanently damage the expensive controller unit and subsequently render the transponder useless; (2) The robot can be easily reprogrammed to adapt to different controller equipment, such as working with a remote controller.

To establish communication between the remote

client and the robot via the server and enable the robot to carry out desired commands without exceeding its limited workspace or without running into obstacles, the following tasks need to be taken care of (figure 6):
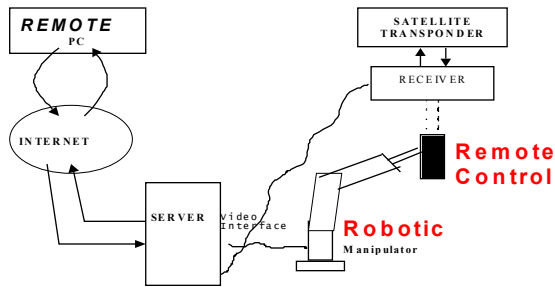

Figure 6: *System Schematics*

Interfacing between robot and server (RS-232); Interfacing between receiver and server; Interfacing between server and Internet; Teaching the robot (trajectory planning and generation); Bringing it all together.

Interfacing between the robot and the server is carried out via the RS-232 serial port that is a standard on the main board of any IBM-compatible PC. The robot's controller box simply receives commands as ASCII strings through the RS-232 connection. An important issue that arises is that there is only one-way communication. i.e. Once the computer issues commands to the robot the controller box does not send back a signal of any kind to indicate a successful or unsuccessful command interpretation and application [9]. To overcome this it is important that a flawless trajectory be planned during initialization of the robot and implemented in a manner such that obstacles do not pose a problem.

Interfacing between the receiver and the server is implemented through a special interface card which takes the analog feed from the transponder's receiver and converts it into a digital format ready to be broadcast over the internet. The PC card receives a feed through a standard coaxial cable such as that used for Cable television. The transponder can point at different 'look-angles' and at each look-angle it can receive feeds at numerous frequencies.

Interfacing between the server and the internet is relatively simple since it involves setting up and configuring a web server to serve HTTP requests from clients [10]. Security is an important issue here and user-authentication is vital since it is undesirable to have unauthorized individuals controlling the robot remotely and causing damage to equipment. It is also important to ensure that only one individual is controlling the robot at the same time. More than one person controlling the same robot can have disastrous results. It is also important to initialize the robot so that it starts at a posi-

tion known to the software. Initialization also involves defining key positions in the robot's workspace and defining trajectories that the robot should travel along.

The robot must be 'taught' about its surroundings and the environment that it will be operating in. This can be done by fixing the position of the remote control unit and defining it's position relative to that of the robot. This allows the robot to 'know' where the controller unit is and where particular controls on the unit are. The robot must be nested prior to operation to ensure that it always starts from the same point thus allowing error-free software control.

Constraints and limitations:

Limited set of commands that the robot's controller box can understand; Only one-way communication between the server and the robot's controller box; The web server's clients have no access to the serial port which is the only means of talking to the robot; Time constraints; Compatibility issues force use of a certain operating system

How were the above limitations overcome to achieve the objective?

The following tasks had to be carried out: Configure the server-to-server HTTP requests; Write software allowing HTTP clients to issue commands to the robot via the RS-232 serial port; Write software to initialize the robot and define the work area.

Server configuration was relatively less complicated and setting up an NT server with an IIS web server to process requests and enabling security authentication and control was smooth.

An *ActiveX control* needed to be written in Visual C++. This component allows Active Server Pages to run executables on the server itself.

An *executable* called 'writetoport.exe' did the simple task of taking command-line arguments and sending these arguments to the robot as commands via the serial port [11]. This application was first implemented in C++ under a DOS environment but when ported to Windows NT, the operating system's security management features did not allow direct access to the computer's serial ports.

Robot initialization

Robot-initialization is vital to ensure error-free operation, especially since the robot's controller unit sends back no signals to the server itself. The position of the remote control needs to be fixed and positions within the robot's work area need to be defined.

There are two basic types of movements: Point to Point (PTP) movement and XYZ movement

PTP movement involves specifying the joint variables (the joint angles in this case) and changing them according to the desired position of the robot.

XYZ movement involves moving the end-ef-

fector in an x,y,or z plane. The joint variables are calculated based upon position and desired motion and accordingly changed and recalculated at every point during the trajectory.

XYZ movement is much slower and inaccurate than PTP movement since the robot needs to perform repeated calculations to determine the desired position. PTP movement was used to minimize error between the desired trajectory and the actual trajectory. In order to use PTP movement the trajectories need to be well-defined; each of the joint angles need to be specified and effected simultaneously to allow smooth trajectories.

Advantages and potential uses

Remote satellite feed for Internet users; Remote communication; Distance learning

The robot was set up in the actual work environment. Communications were established and rigorous testing was conducted to avoid unnecessary damage.

This project demonstrates the synergy created by combining robotic with specific software computing. On a larger scale this concept can be ported to pragmatic and useful applications using robots.

BIBLIOGRAPHY

[1] Tanenbaum, Andrew S. Computer Networks, New Jersey, Prentice Hall.

[2] Meyers, Robert A. Encyclopedia of Telecommunications, Academic Press Inc.

[3] Skahill, Kevin. VHDL for Programmable Logic, Addition-Wesley, 1996.

[4] Yalamanchili, Sudhakar. VHDL Starter's Guide, Prentice Hall, 1998.

[5] Doty, Keith L. TALRIK[II] Assembly Manual, Mekatronix[TM], 1999.

[6] Internet: ftp://ftp.teltone.com/pub/8870.pdf

[7] H. Latchman, Sanjeev Tothapilly, C. Salzmann and D. Gillet, "Hybrid Asynchronous and Synchronous Learning Networks in Distance and Local Education", Proceedings of the 1998 International Conference on Engineering Education, Rio de Janiero, August, 1998.

[8] D. Gillet, G. F. Franklin, R. Longchamp, and D. Bonvin, "Introduction to Automatic Control via an Integrated Instruction Approach", The 3rd IFAC Symposium on Advances in Control Education, Tokyo, Japan, August 1994.

[9] F. Michau and D. Munteanu, "Web Server of Pedagogical Documents in Control Engineering", 7th Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEEIE) on Telematics for Future Education and Training, Oulu, Finland, 1996.

[10] Foley J. D., Dam A. V., Feiner S. K., Hughes J. F.; Computer Graphics: Principles and Practice; Addison Wesley Publishing Company; 1990

[11] E. L. Hutchins, J. D. Hollan, and D. A. Norman. "Direct Manipulation Interfaces." Human-Computer Interaction. 1, 1985, 311-338.

Professor Tarek M. Sobh received the B.Sc. in Engineering degree with honors in Computer Science and Automatic Control from the Faculty of Engineering, Alexandria University, Egypt in 1988, and M.S. and Ph.D. degrees in Computer and Information Science from the School of Engineering, University of Pennsylvania in 1989 and 1991, respectively.

He is currently the Dean of the School of Engineering at the University of Bridgeport, Connecticut; the Founding Director of the Interdisciplinary Robotics, Intelligent Sensing, and Control (RISC) laboratory; a Professor of Computer Science, Computer Engineering, Electrical and Mechanical Engineering; and the Chairman of the Prototyping Technical Committee of the IEEE Robotics and Automation Society. He was the Interim Chairman of Computer Science and Computer Engineering and the Director of External Engineering Programs at the University of Bridgeport. He was an Associate Professor of Computer Science and Computer Engineering at the University of Bridgeport from 1995 -- 1999, a Research Assistant Professor of Computer Science at the Department of Computer Science, University of Utah from 1992 -- 1995, and a Research Fellow at the General Robotics and Active Sensory Perception (GRASP) Laboratory of the University of Pennsylvania from 1989 -- 1991.

He was the Chairman of the Discrete Event and Hybrid Systems Technical Committee of the IEEE Robotics and Automation Society from 1992- 1999.

His background is in the fields of computer science and engineering, control theory, robotics, automation, manufacturing, AI, computer vision and signal processing. He has published over 100 journal and conference papers, and book chapters in these and other areas.

Dr. Sobh has been awarded many grants to pursue his research.

Dr. Sobh is a Licensed Professional Electrical Engineer (P.E.) and a Certified Manufacturing Engineer (CMfgE) by the Society of Manufacturing Engineers, a member of Tau Beta Pi (The Engineering Honor Society), Sigma Xi (The Scientific Research Society), Phi Beta Delta (The International Honor Society), Upsilon Pi Epsilon (The Computing Honor Society) and Delta Mu Delta (The Business Adminstration Honor Society).

Dr. Sobh was the recipient of the Best Paper Award at the World Automation Congress Conference (WAC 98) in Anchorage, Alaska.