2006-1424: LABORATORY DEVELOPMENT FOR ROBOTICS AND AUTOMATION EDUCATION USING INTERNET BASED TECHNOLOGY

Richard Chiou, Drexel University
Dr. Richard Chiou is currently Associate Professor of Applied Engineering Technology at Drexel University in Philadelphia. Dr. Chiou received his Ph.D. degree in Mechanical Engineering from Georgia Institute of Technology in 1995. His areas of education and research emphasis include machining, mechatronics, and Internet based robotics and automation. Dr. Chiou incorporates real-world problems into his research and teaching. He has secured many research and education grants from the NSF, the SME Education Foundation, and industries.

Yongjin Kwon, Drexel University
Dr. Yongjin Kwon has over 12 years of engineering experience in industrial and academic settings. He has extensive experience & practical knowledge in current design, manufacturing and quality control. His work has been cited a number of times in high profile journals. He is currently developing Internet-based manufacturing systems.

Shreepud Rauniar, Drexel University
Mr. Shreepud Rauniar was born in Kathmandu, Nepal in 1980. He received B.S. in Mechanical Engineering from National Institute of Technology, Jamshedpur, India in 2003. Since 2004 he has been pursuing M.S. in Mechanical Engineering at Drexel University. His research interests are in designing web based applications & control systems for robotics and automation.

Horacio Sosa, Drexel University
Dr. Horacio Sosa is a professor and the senior associate dean of the Goodwin College at Drexel University. He received his M.S. and Ph.D. degrees in applied mechanics from Stanford University. His research interests include theoretical mechanics, advanced materials, and engineering education. He was a visiting professor at universities in Japan, Spain and Argentina, and was awarded with Fellowships from Argentina’s National Commission for Atomic Energy, Spain’s Ministry of Education, NATO’s Collaborative Council, and FULBRIGHT.
Laboratory Development for Robotics and Automation Education
Using Internet Based Technology

Abstract

This paper describes laboratory and curriculum development integrated with Internet based robotics and automation for engineering technology education. Drexel University’s Applied Engineering Technology Program received a NSF CCLI grant to develop a series of laboratory courses in the area of internet based robotics and automation in manufacturing. The development efforts include industrial partnership with Yamaha Robotics, restructured and advanced courses in applied engineering program curriculum, and laboratory activities integrated with network technologies. The instructional materials for Internet based robotics and automation education utilized Robotics and Mechatronics lab as the experiments of choice. The application-oriented curriculum, which is validated by the industry, has been developed with the goal of producing graduates ready to work in engineering technology. The new Internet based techniques allow the remotely situated students to program, control, and monitor the robotic operations through the Internet using the Windows-based graphical user interface. This also allows the integration of robots and automation into the information networks and easy access through the Internet for design and manufacturing. The paper also covers the new learning technology through the lab development for teaching of the remote monitoring/control of robots and programmable logic controllers (PLCs) and how to effectively deliver internet based robotics and automation education through the Internet.

1. Introduction

Robotics and automation are essential components in automotive, electronic, appliance and other industries. Over the past few years, Web-based approaches have been widely used in collaborative product design and manufacturing. In the areas of robotics and automation, Web-based technology is regarded as a new promising approach for manufacturing. In tomorrow’s factory, design, manufacturing, and business are integrated into the Internet. It is a trend that web based robotics and automation have become critical issues in the integration with e-manufacturing systems and management. Internet has improved technology tremendously over the past few years. The improvement in data transfer speed, data security, technology to transfer these data has opened new frontiers. The internet technology can also create online educational tools for teaching and demonstration of automated manufacturing processes with robotics. Those Web-based systems allow robotics and automation to communicate, share design data, information and knowledge through the Internet1-13.

To enhance the workforce skills in the product development cycle, an Internet-based approach for lab development is introduced to develop web-enabled robotics and automation. This laboratory development component in the NSF project deals with integrating various AET (applied engineering technology) tools such as robots, pneumatic actuators, sensors, web-cameras, conveyors, programmable logic controllers
(PLCs) and use Internet to connect these cross-platform systems. Visual Basic.Net has been used to create a graphic user interface (GUI) for performing offline analysis and to interact remotely with the system. These tools can be used by students to develop their ideas in the field of robotics, obtain industrial experience in interacting with the latest equipments, deal with different sensors, and learn different ways to make them work with various equipments\textsuperscript{14-18}.

2. System Architecture

The core of our work is to provide a server side system that can communicate with laboratory instruments and mechanical devices so that Internet based robotics and automation can be remotely controlled on the client side. The experimental setup includes the following items: Yamaha SCARA robot, RCX40 robot controller with optional on-board Ethernet card, Yamaha I/O checker, DLink DCS-5300, HP m1050e PCs, and Allen Bradley PLC 1756 Series. The system also consists of power supplies, DC motors, fans, buzzers, limit switches, relays, and lights. The RCX40 controller can be connected to the Ethernet and controlled using a PC/Server. A web-camera is used for constantly viewing the robot movement.

![Figure 1. (a) A workstation with a Yamaha robot, a computer, and a controller, and mechanical devices. (b) Allen Bradley PLC 1756 Series for Internet based automation.](image)

3. Web Based Control Architecture

All the devices, such as robot, web-camera, PLC, are connected to Ethernet. This reduces the wire maze needed to link every device and enables users (students) to operate/control the equipment remotely. All the devices mentioned are connected to a LAN network as shown in Figure 2. These devices are connected by Internet Protocol (IP) address over the network. Network means the connection between the devices in the work area as well as with the users. Network used here is local area network (LAN), which is connected to a Drexel University server. The web server is connected to this LAN and also from outside, the LAN can have access to the devices on the network. Every device connected to the network has a unique IP address, which is used to connect to them and also recognize them on the network. The IP address is separated into network...
address and host address sections. The network address section is extracted from the IP address by *AND* processing with the subnet mask. The remaining portion is the host address section. Devices belonging to the same network must be set to have the same network address. The host address, however, should be different for every device so that no two devices have the same number. The first and the last host address numbers are reserved for the system (0 and 255), so it should be made sure not to set these as the IP address.

In the process of communication, the IP address of the device must first be specified for the connection. After the connection is made, the actual data is exchanged between the devices. For security reasons all those devices are password protected. This paper focuses on the Internet based automation technology in communication between a PLC and a Yamaha SCARA robot with an RCX40 controller. The controller of the robot (RCX40) can be connected to Internet directly or to a computer using RS 232C cable on COM 1 port. The software used for communication is VIP for Windows Version 1.6.0 developed by the Yamaha Co. Ltd. This application can be used in the following ways:

- **Backup current data from robot controller**: This can be used to download the program from the robot controller and save it as a file in the computer.
- **Restore previously saved data to robot controller**: This can be used to upload the saved program from computer to the robot.
- **Operate robot controller and robots directly**.

Robot can be controlled manually or automatically. While controlled manually, the arm movement can be controlled by pressing jog buttons which changes the coordinates along axes. Automatic control is performed by writing a program, which defines and sets coordinate, speed & path of the robot. After uploading data from the computer to the robot controller, it can also be modified. This feature eliminates the hassle of
downloading the data to computer each time if any changes need to be made. The RCX40 controller can be connected to the Internet in two ways: (a) through a computer connected to the Internet, and (b) directly through an on-board Ethernet Unit. In the first method (a), the controller can be connected to the computer (which would now be called a server as it would be acting as a media between an operator and the controller) using RS 232C cable. This server is connected to the internet. It has an unique IP Address which would be used to locate this computer remotely. Once this computer is called using this IP, it gets connected to the user. It would then accept the commands from the operator’s computer and send to robot and vice versa. For security reasons, before connecting to the server, the user needs to provide username and password. Only the users registered/configured in the server can access the port for transferring data. This is called TCP (Transmission Control Protocol) port number. TCP port number 80 is usually used and is open to all. But, this can be changed to any number between 1000 and 65535. If this number is set to number other than 80, then it needs to be provided along with IP while connecting to the server.

In the second method (b), the communications protocol utilizes TCP/IP (Transmission Control Protocol/Internet Protocol) which is a standard Internet Protocol so PCs with Internet Access can exchange data with the robot controller. This unit uses 10BASE-T specifications, so UTP cables (unshielded twisted-pair) or STP cables (shielded twisted-pair) can be used. The command is sent to the controller using Telnet. Telnet is a protocol (protocols allow data to be taken apart for faster transmission, transmitted, and then reassembled at the destination in the correct order) for remote computing on the Internet. It allows a computer to act as a remote terminal on another machine, anywhere on the Internet. This means that when Telnet is completed to a particular host and port, controller in our case, the remote computer (which must have a telnet server) accepts input directly from the user’s computer (which must have a Telnet client) and output for the session is directed to user’s screen.

**4. Robot Manipulation over the Internet**

This section provides a brief explanation on how information is exchanged over the Internet between the robot controller and a PC. The RCX40 series robot controllers equipped with the Ethernet card operate as a server and constantly await a connection request from the client (other party’s device such as a PC). Specific actions are then carried out when a request arrives from a client. So the robot controller does not connect to another server on its own. Specific actions are then carried out when a request arrives from a client. The connection to the controller is made using Telnet. Once the connection is established, following are some commands that can be sent to the controller to achieve desired results (Table 1):

RCX40 controller is designed for use with SCARA robots or Cartesian robots. It can control a maximum of 4 axes: X, Y, Z & R (for rotation). It is programmed using BASIC like-high level robot language that conforms to the industrial robot programming language. There are two links present in the SCARA robot which are of open type, i.e. they are coupled with one another only at one end. Thus, the motion between them is
Table 1. List of commands for remote operation of robot.

<table>
<thead>
<tr>
<th>No.</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>@?CONFIG</td>
<td>Check the type of robot connected to controller</td>
</tr>
<tr>
<td>2</td>
<td>@?VER</td>
<td>Check the version of controller</td>
</tr>
<tr>
<td>3</td>
<td>VER</td>
<td>Check the version of Ethernet Card</td>
</tr>
<tr>
<td>4</td>
<td>@?READ PGM</td>
<td>Read the currently loaded program</td>
</tr>
<tr>
<td>5</td>
<td>@?READ PNT</td>
<td>Read the currently loaded points</td>
</tr>
<tr>
<td>6</td>
<td>@?SERVO</td>
<td>Check the status of Robot Servo Motors</td>
</tr>
<tr>
<td>7</td>
<td>@SERVO ON</td>
<td>Power ON the servo motors</td>
</tr>
<tr>
<td>8</td>
<td>@SERVO OFF</td>
<td>Power OFF the servo motors</td>
</tr>
<tr>
<td>9</td>
<td>@?SPEED</td>
<td>Check the current speed</td>
</tr>
<tr>
<td>10</td>
<td>MOVE P, P(x)</td>
<td>Move the robot to the point P(x)</td>
</tr>
<tr>
<td>11</td>
<td>LOGOUT</td>
<td>Terminate the connection</td>
</tr>
<tr>
<td>12</td>
<td>QUIT</td>
<td>Exit Telnet</td>
</tr>
</tbody>
</table>

relative. The first link is a manipulator base and the last link is an end-effector. The movement is in two dimensional planes (xy-plane), whereas the end effector moves in z-axis (vertical plane) and rotation axis (r-axis). Clockwise rotation from the central axis is positive and anti-clockwise rotation from central axis is negative (Figure 3). Thus, calculation for finding the position of the end-effector is focused on 2D plane where rotational movement of arms exists. Actual position of the end effector after knowing the end position of the arm in 2D plane can be found by changing the coordinate only in z-direction and r-direction. Since in serial manipulator, all links are connected to one another, if the base link is displaced. Then, it affects all the other links. Similarly, in SCARA robot arm, if a base link is displaced, the end-effector moves. The rotation movement of the link can be calculated using a transformation matrix.

Figure 3. SCARA robotic arm notation and its coordinate frame

The transformation matrix for a SCARA robot is:

\[
\begin{bmatrix}
  x' \\
  y'
\end{bmatrix} =
\begin{bmatrix}
  \cos \alpha & -\sin \alpha \\
  \sin \alpha & \cos \alpha
\end{bmatrix}
\begin{bmatrix}
  x \\
  y
\end{bmatrix}
\]  

(1)

where \( x' \) and \( y' \) are transformed coordinates. Thus, first considering angle \( \alpha \) to be the rotation of the first arm, one can calculate the coordinate transformation of its tip. Due to
the displacement of the first arm, the second arm also gets displaced by the same amount with respect to the axes. Thus, the new coordinate of the tip of the second link becomes:

\[
\begin{align*}
x_{\text{New}} &= x_{\text{Old}} + (x' - x) \\
y_{\text{New}} &= y_{\text{Old}} + (y' - y)
\end{align*}
\]

(2)

If the second arm also rotates, then the above transformation matrix can be used to calculate its final position. The transformation for the second arm that rotates by an angle \( \beta \) and has initial and final positions \((x_{\text{New}}, y_{\text{New}})\), denoted as:

\[
\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \beta & -\sin \beta \\ \sin \beta & \cos \beta \end{bmatrix} \begin{bmatrix} x_{\text{New}} \\ y_{\text{New}} \end{bmatrix}
\]

(3)

The method to analyze the robot manipulator is to use reverse or inverse kinematics. After analyzing the arm configurations, this information can be sent to the robot to move to that position. This helps in doing offline path planning of the robot to do the pick and place operation.

5. Robotic System Communications

Yamaha I/O checker is used for communicating robot input/output signals to and from RCX40 controller as shown in Figure 4. It is a collection of switches, each with specific function. The operating voltage is 24V and is provided to pin numbers 47-48 (+24V DC) and 49-50 (GND). The connection to the controller is done using MR-50LM (50 pin) connector. Notation used for input data is DI and for output DO followed by two digit numbers, e.g. DI01, DI11, DO33. Pin 11 (DI20) through 26 (DI37) are used for general purpose input signals. Pin 39 (DO20) through 46 (DO27) are used for general purpose output signals.

![Figure 4. (a) Yamaha robot and Yamaha I/O checker, (b) The pin structure of I/O checker, and (c) MR-50LM connector.](image-url)
6. Programming and Control Aspect of the Experiment

In a hands-on experiment, students were asked to use the PLC flip-flop timer sequencing technique for stepping a program they design through a typical process industry cycle, using the electrical test bench for practical demonstrations. This was a combined PLC and robotic control experiment. In the lab, the students should efficiently decide who is to do the hookups, who is to keep the data, who is to be preparing the next section, etc, and should press forward to complete it in the allotted 2-hour time period. Industries which consist of the processing of liquids and/or gases into useful products are generally known as the Process Industries. Petrochemicals (Oil and gasoline), food manufacture (soups and sauces), pharmaceuticals (cough syrups), chemicals (polyester resins) and many other products are made in this fashion, requiring large numbers of engineers from many backgrounds.

Not only does the PLC perform the sequencing from one step to another, but with the use of Yamaha Robotics digital I/O, it also can control temperature, pressure and the flow rate of fluids. Because of the precise nature of this type of control, a batch processing requires highly sophisticated PLC programming and engineering. This experiment deals with the simulation of a small batch processing operation typical of what might be used in the pharmaceutical, food or organic chemical industries. Processing involves various mixing, pressurizing, heating, cooling or curing operations, any of which may be used to optimize chemical reactions.

Figure 5 shows the control circuits for integration of PLC, a robot, and mechanical systems for a simulated process. The control is planned out for a logical system by labeling inputs and outputs. The students have to make sure that the PLC programming is in

![Diagram](image-url)

Figure 5. Integration of programmable logic control with robots through Internet.
agreement with the logical system and the required electric circuit is connected with the interconnect wiring to the Yamaha I/O checker as well as the PLC input and output modules. It is designed that a lamp on the electric control module to simulate the heater, an air motor to simulate the pump motor, and a 110 VAC fan motor to simulate the mixer motor. Students also need to check out the proper identification and action of the switches by manually overriding the solenoids. The detailed list, which cross-references logic, flip-flops, I/O numbers, cylinder position, is absolutely essential as students get into this experiment.

**Programming using I/O:**

Using the Input/Output commands, the program is written in a way that it waits for an input signal (DI). As soon as the input signal is received (input signal for the controller can be sent using I/O checker by flipping a switch), the command in the program tells the robot what operation needs to be performed. The following is an example:

```
001 WHILE -1
002 WAIT DI(20)=1
003 DELAY 100
004 SERVO ON
005 DELAY 100
006 MOVE P,P0,P1,P0
007 DELAY 100
008 SERVO OFF
009 WEND
010 HALT
```

In the above example, the controller waits for input signal from DI20 (pin No. 11). The signal can be provided using the I/O checker by flipping the switch labeled DI20 (when the switch is flipped, LED next to the switch should light up). Open state of the switch/signal is denoted by ‘0’ and closed stated by ‘1’. In other words, the program works step by step:

- **Line 001:** starts an infinite loop i.e. the program runs in continuously over and over again.
- **Line 002:** this instructs the controller to wait and watch DI(20) till it’s value is 1 (closed state)
- **Line 003:** pause for 0.1 sec (1 sec = 1000)
- **Line 004:** turn on the servo
- **Line 005:** pause for 0.1 sec
- **Line 006:** move to points P0, P1 and P0
- **Line 007:** pause for 0.1 sec, turn the servo off, end while loop and halt operation.

It should be noted that there is only one input signal in the program. In general, a program can be written for 16 general purpose inputs and 8 outputs. (Note: When I/O checker is connected, switch DI11 (Interlock) needs to be flipped ON.)

7. **Integration of Robotics with PLC over the Internet**

To integrate robots with the PLC over the Internet, a PLC program needs to be created offline and then downloaded into the controller. There are a 16-point DC input module in each slot (start counting with slot 0) of PLC station. Notice the rung that appears in the right window in Figure 6. Also note that the rung is in edit mode as
signified by the “e’s” to the left of it. It is now ready to add the rungs. The steps are the same, only the addressing will be different. Notice the tag names on the left identify the tag as referring to a local (vs remote) module, in slot 0 or 1, and data type of input (I), output (O), or configuration (C). These tags all refer to the Input and Output modules. Figure 7 shows the ladder diagram corresponding to the production cycle in the experiment. In the Figure, the local 0:I Data 0 is closed by a limit switch after a robotic operation is completed and a Timer (TON) is activated for heating process (10 seconds). The internal normally open contact mytimer DN in rung 1 is closed after 10 seconds and then the counter CTU in rung 2 is triggered for counting the number of products from a light sensor in the system. After 20 products are packed in a case in the production, a motor for packaging is energized by the output Local 2: O Data 1 in rung 3.

Figure 6. Rockwell Automation RSLogix 5000 software for PLC programming

Figure 7. Ladder diagram segment for a production cycle.
8. A Case Study in a Robotic and Automation Experiment Through Internet

The system integration enables the computer to perform the sequence of tasks outlined by a user, automatically, when the user starts the equipment from a remote site. The sequence control is primarily accomplished by the use of Rockwell Automation RSLogix 5000 programming software and Yamaha VIP programming. This application of Yamaha VIP Windows is assistant software for multiple axes robot controller and robots. It can be used to create and edit program, point, parameter, shift and/or hand data directory used with robot controller. The RSLogix 5000 software along with PLC and Yamaha VIP software for robotic programs enables information exchange between the various levels of the control architecture.

The controller (RCX40) is connected to a computer (we would treat it as a server as it awaits for commands from user/client and sends it to the controller). The connection between the controller and the server is made by using RS 232C cable connected to the COM 1 port of the server. For viewing the workspace a D-Link webcam which has Pan/Zoom/Tilt functions has been used as shown in Figure 8. The software used for controlling the robot is VIP Window software developed by Yamaha Robotics. This software enables to run the robot manually by typing specific destination points, or automatically by writing a program. The DLink webcam can be accessed using Microsoft

![Figure 8](image)

(a) Connection to the server is established using RDC and VIP Windows software is used for controlling the robot. (b) The robot is viewed simultaneously using the Dlink webcam.
Internet Explorer and typing in the IP address of the camera. For security reason the camera is password protected. It can be accessed in two modes: demonstration mode and complete access mode. In demonstration mode features like Pan/Tilt/Zoom are disabled. The VIP Software on the server can be accessed from anywhere using Remote Desktop.

9. Conclusions
The development of the internet based robotics and automation laboratory through NSF/Industrial partnership has significant, positive impacts upon education, research, and service at the Drexel’s new MET curriculum. The authors are building a network e-robotic and e-automation system towards real-world applications so that Internet users, such as students, can control the industrial robots in a dynamic environment remotely from any place in the world. Such high technology curriculum will offer a unique career pathway to students interested in advanced engineering technology through the full time program at Drexel, Dual Programs with Community Colleges, and 2+2+2 Programs incorporating also high schools in the Greater Philadelphia Region. Furthermore, the project will reinforce our ongoing initiatives to revitalize the regional manufacturing sector, by providing highly skilled graduates to meet the demands of new advanced technologies.

Acknowledgement
The authors would like to thank the National Science Foundation (Grant No. NSF-DUE-CCLI-A&I 0410719) for its financial support of the project and Yamaha Robotics for its in-kind gift of robot equipment.

References