

VISUAL BASIC PROGRAMMING FOR INTERNET BASED ROBOTIC CONTROL

Richard Chiou, Yongjin Kwon, Shreepud Rauniar, and Horacio Sosa
Applied Engineering Technology
Goodwin College of Professional Studies
Drexel University
Philadelphia, PA 19104

Abstract

This paper introduces the concept of learning real time internet based robotic control for manufacturing by using Visual Basic computer programming. The new internet based techniques allow the remotely situated students to program, monitor, and control the robotic operations. The development efforts include industrial partnership with Yamaha Robotics in laboratory activities integrated with network technology. This also allows the integration of robotics into the information networks through the internet for design and manufacturing. Internet based robotic control has been implemented as an integral part of laboratory activity at Drexel University that combines mechanical, electrical, computer engineering, and information technology principles. Visual Basic programming is cost effective in the environment for teaching internet based robotics.

Introduction

Robotics and automation are essential components in automotive, electronic, appliance and other industries. Over the past few years, internet-based approaches have been widely used in collaborative product design and manufacturing. In the areas of robotics and automation, internet 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 internet based robotics and automation have become critical issues in the integration with manufacturing systems and management. Internet has improved technology tremendously over the past few years. The improvement in data transfer speed,

data security, and technology to transfer data has opened new frontiers that allow robotics and automation to communicate, share design data, information and knowledge through the internet[1-8].

The use of Java and HTML program has been adopted for remote control in accessible worldwide internet. The internet based robotic systems usually have graphical or textual interface so that the user submits the request by filling up the fields of a web form and receives the textual and graphical information of the current state of the robot at the remote site. The internet based technology has also created online educational tools for teaching and demonstration of automated manufacturing processes with robotics. However, over the years of teaching and training in the industry, there is still a considerable lack of knowledge for proper training of robotic control in both academia and the industrial communities. The specifications, sequences, and planning of robotic assembly jobs that require industrial standard and approximate location from the feedback image have been the challenging tasks [9-13]. In addition, the issue of writing clear and efficient code with the fundamental Visual Basic programming for internet based robotic control is rarely addressed in the many papers or textbooks used in schools and colleges.

Visual Basic (VB) programming language is a quick solution for developing internet based control technology. The programming structure and excellent compatibility with windows operation system makes VB a feasible tool for developing internet based control and monitoring applications for educational and industrial purposes. Most of the Ethernet based

devices, such as webcam, operate using a library function ActiveX control. It can be quickly embedded in a webpage or in a VB executable application. Also, major companies recommend the use of VB for developing control applications. Yamaha Robotics' controller (RCX) is programmed with VB library. The library files are recommended to be used with VB to make the executable files for robotic control and automation [14-19]. VB can increase the flexibility in modifying the program in least amount of time to accommodate the changes necessary. 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 internet based technology.

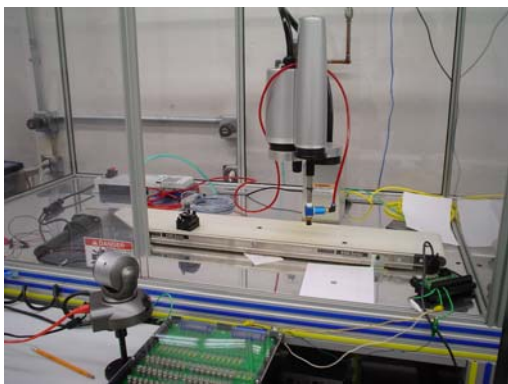


Figure 1. A workstation with a Yamaha robot, a controller, a conveyor, and mechanical devices for internet based robotic control.

Development of Web Based Robotic Control Systems

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 as shown in Figure 1 includes the following items: Yamaha SCARA robot, RCX40 robot controller with optional on-board Ethernet card, Yamaha I/O checker,

DLink DCS-5300, and HP m1050e PCs. 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. Two web-cameras are used for constantly viewing the robot movement. 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 development of Visual Basic software 6.0 program for internet based communication with a Yamaha SCARA robot. The RCX40 controller can be connected to the internet directly through an on-board Ethernet Unit. The communications protocol utilizes TCP/IP (*Transmission Control Protocol/Internet Protocol*) which is a standard internet Protocol

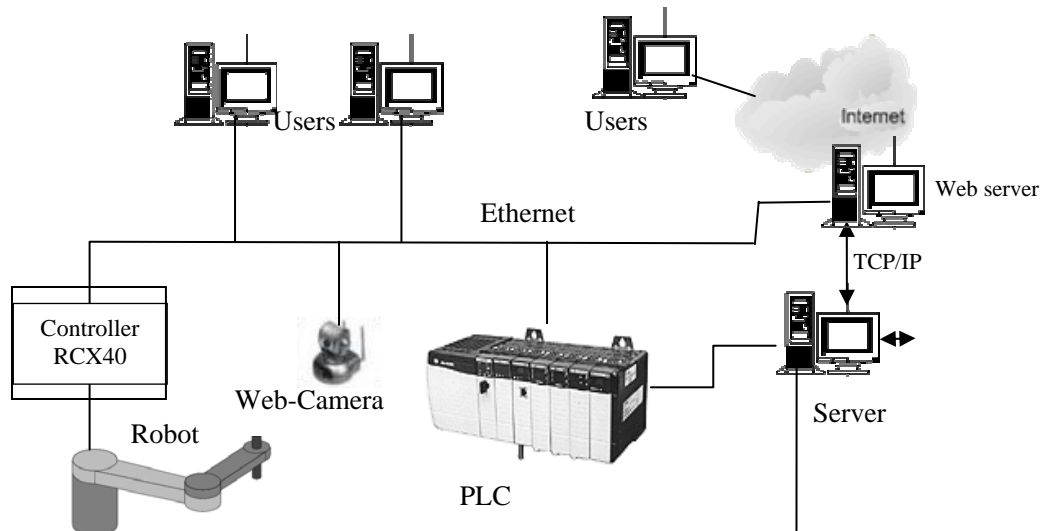


Figure 2. Web based control architecture.

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.

Robotic Operation Systems

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. Therefore, 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. 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). 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 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 based 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 [16 - 19].

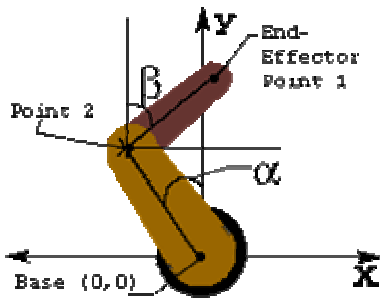


Figure 3. SCARA robotic arm notation and its coordinate frame.

Visual Basic Programming

Visual Basic programming for applications in internet based robotic control is an attractive alternative. VB, with its control objectives and internet library files make computer programming a very interesting subject. The

availability of these functions in VB library for internet based control enables students to perform sophisticated internet based robotics. The TCP/IP is a two way communication protocol. For every instruction issued by the user, the controller sends back an acknowledgement. This ensures that the issued command is executed. Visual Basic programming language provides Winsock library (winsck.dll) for internet programming. Basic methods provided by this library that are mainly required for a successful control and monitoring of the robot are connect(), state(), getData(), sendData(), and disconnect(). The system integration with the methods enables the internet based robotic control to perform the sequence of tasks outlined in Figure 4, automatically, when the users starts the experiment from a remote site.

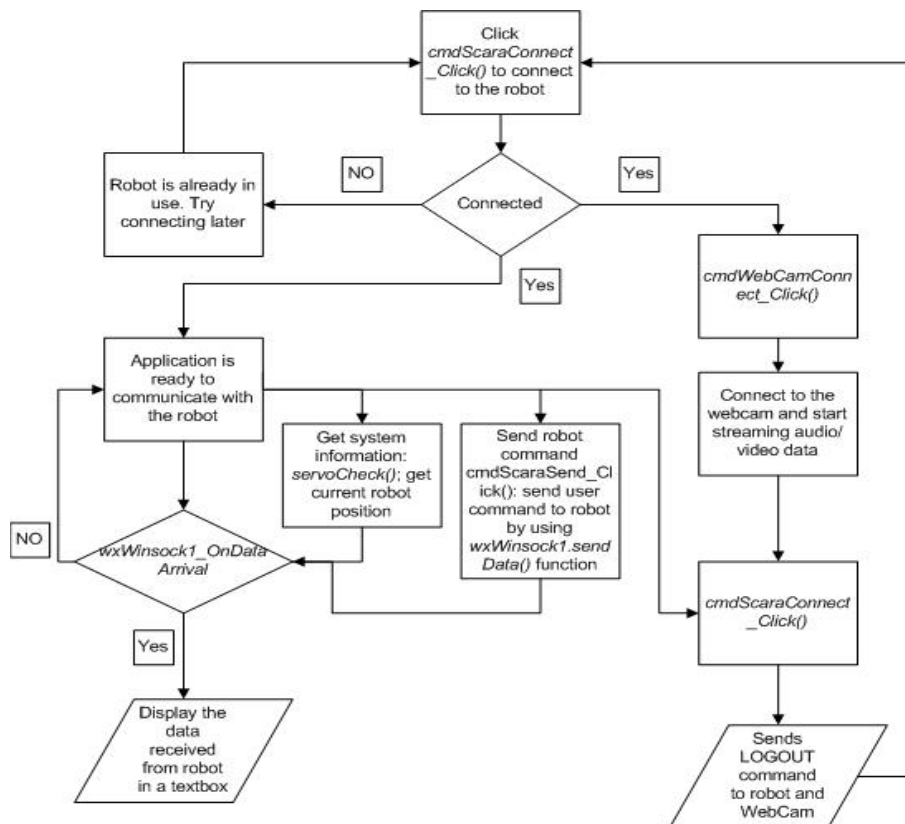


Figure 4. Visual Basic Programming flowchart for internet based robotic control.

Connection With Robotic Controller

The first step is to establish a connection with the controller when the application starts. The IP address, port number, username and password need to be provided for the communication channel. This is completed by clicking the *connect* button which is named *cmdScaraConnect*. The code is given below:

```
Private Sub cmdScaraConnect_Click()  
Dim sServer, nPort As Long  
wxWinsock1.Connect sServer, nPort  
End Sub
```

Success of connection is determined by *onConnect()* method. Once the connection is established, the controller asks for login information. The request of the controller needs to be handled by the program in a separate event. This is completed by using the *onDataArrival()* method. This method constantly looks for arrival of any data from the controller. Each different set of instruction for the controller has a unique word. Thus the VB application can recognize it by transferring the latest arrived data and comparing the word with the command list presented in the codes. This can be more clearly shown by interpreting the authenticating process. The following code describes the authenticating event:

```
Private Sub wxWinsock1_OnDataArrival(ByVal bytesTotal As Long)  
Dim sBuffer, sBufferTemp As String  
Dim strPos As Long  
Dim i As Integer  
wxWinsock1.GetData sBuffer  
sBuffer = Trim(StrConv(sBuffer, VbStrConv.vbUpperCase))  
If (InStr(sBuffer, "LOGIN")) Then  
    wxWinsock1.SendData (txtScaraUser.Text & vbCrLf)  
End If  
If (InStr(sBuffer, "PASSWORD:")) Then  
    wxWinsock1.SendData (txtScaraPassword.Text & vbCrLf)  
    authenticate = True  
    wxWinsock1.SendData ("@?SERVO" & vbCrLf)  
    wxWinsock1.SendData ("@?MOD" & vbCrLf)  
    wxWinsock1.SendData ("@?WHRXY" & vbCrLf)  
End If  
.....  
.....  
End Sub
```

When the connection is established and *onConnect()* event is successful, the controller sends the word "LOGIN". This is captured by using *GetData()* and storing the data in a variable *sBuffer*. This stored data is then compared with the word "LOGIN". If the comparison is true, then the username is sent to the controller by using *SendData()* method. Once the login is successful, it is stored in a session variable (Boolean) as *authenticate = TRUE*.

Command Codes for Web Communications

After the authentication is completed, the next step is to acquire the robot system information such as current arm position, status of motors, mode, loaded program, etc. This information is acquired by sending the respective commands and receiving the responses from the controller. The following code illustrates how to recognize the current servo motor status:

```

Public Sub checkServo()
    Dim sBuffer As String
    wxWinsock1.SendData "@?SERVO" & vbCrLf
    wxWinsock1.GetData sBuffer
    If sBuffer = "ON" Then
        cmdServoOn.Caption = "Turn-Off Servo"
    End If
    If sBuffer = "OFF" Then
        cmdServoOn.Caption = "Turn-On Servo"
    End If
    Return
End Sub

```

This function sends the command “@?SERVO & vbCrLf” to the controller and receives the response using wxWinsock1.GetData. The sent command requests the servo status in the controller. This command ends with “vbCrLf” which stands for Visual Basic Carriage Return Line Feed. It is used as the shortcut for the “Return” key. The received data is stored in a variable sBuffer. There can be two responses for the sent command “ON” or “OFF”. Thus, the required operation can be completed by comparing the responses. If the response is “ON”, the caption on the button “cmdServoOn” is set to “Turn-Off Servo.” Similarly, if the response is “OFF”, the caption on the button “cmdServoOn” is set to “Turn-ON Servo.” The other commands can also be created to acquire the various status of the robot as shown in Table 1.

Table 1. Robotic operation commands.

Command	Description
@?SPEED	Check the current speed of the robot
@?WHRXY	Check the current robot position
@?MOD	Check the status Auto/Manual
@?EMG	Emergency Stop Status
@?MEM	Remaining Memory Capacity
@?DI(22)	Check the input state of data-input pin 22
@READ DIR	Read the program directory of the controller

Robotic Position Control

The response to the current robot position starts with “[POS]” followed by the coordinates of the end-effector. When the response is received, the string is parsed and the coordinates are separated to be displayed accordingly. This block of code is placed inside the event procedure wxWinsock1_OnDataArrival which was described earlier. The following code shows how the robotic operation is programmed:

```

'check for current position
If (InStr(sBuffer, "[POS]")) Then
    strPos = InStr(sBuffer, "]")
    valX.Text = Trim(Mid(sBuffer, strPos + 1, 8))
    valY.Text = Trim(Mid(sBuffer, strPos + 9, 8))
    valZ.Text = Trim(Mid(sBuffer, strPos + 17, 8))
    valR.Text = Trim(Mid(sBuffer, strPos + 25, 8))
End If

```

InStr() function checks the presence of a particular string in a variable. In this case this function is used to check if [POS] is present in the data received from the controller. If it is found then the variable is parsed and X, Y, Z and R coordinates are stored in the respective variables. These variables are then used to move the robot arm via the internet as shown in the following program:

```

Private Sub cmdXminus_Click()
    wxWinsock1.SendData ("@MOVE P," & CInt(valX) - CInt(txtIncrement) & " " & valY & " " & valZ & " " & valR & " 0.00 0.00" & vbCrLf)
    wxWinsock1.SendData ("@?WHRXY" & vbCrLf)
End Sub

```

When the button “cmdXminus” is clicked this event procedure is executed. This function moves the x-axis of the robot in the negative direction by the increment (in mm) provided. Thus, the robot would move in x-direction which would be at a distance given by “txtIncrement” along negative direction. “txtIncrement” is a textbox which stores the value of increment the robot needs to move when it is being controller manually. Similarly, other axes are also controlled in the same way.

Finally, the connection can be closed by sending “LOGOUT & vbCrLf” to the robot.

Web Camera Image Capturing

The command “WebRauto” application enables streaming of video images from web cameras installed in the robot work cell. Most of the web cameras are provided with the web based interface for its control. The images from this camera can be viewed using a web browser by entering the IP address in the browser. Since the command “WebRauto” is a complete control package, it is necessary to have a video feedback embedded with it. For most of web cameras, ActiveX component is used for controlling it. This can be incorporated into a VB application for displaying image by using the browser library Microsoft Internet Control component. Below is the part of code used in “WebRauto” application for creating a webpage and loading the video image:

```
Private Sub cmdWebcamConnect_Click()  
    Dim fso, txtfile, myPath  
        If cmdWebcamConnect.Caption = "Connect" Then  
            Set fso = CreateObject("Scripting.FileSystemObject")  
            Set txtfile = fso.createtextfile("dLinkConnect.htm", True)  
            txtfile.write Trim(dlinkTxtStart) & dlinkTxtMiddle & "http://" &  
            & txtWebcamUser & ":" & txtWebcamPassword & "@" & txtWebcamIp  
            & Trim(dlinktxtEnd)  
            txtfile.Close  
            myPath = CurDir()  
            axBrowser1.Navigate (myPath & "\dLinkConnect.htm")  
            controlButton cmdWebcamConnect, "Disconnect"  
            .....  
End Sub
```

The button “cmdWebcamConnect” is used for loading the video image from the camera. When it is pressed, a webpage is written using the “File system object” provided in VB. The commands *dlinkTxtStart*, *dlinkTxtMiddle* and *dlinktxtEnd* are the variables that contain the script to be written to the webpage. The name of the webpage created is *dLinkConnect.htm*. This webpage consists of commands that load the ActiveX component by providing the object ID

and class ID, IP address, port address, username, and password.

Internet Based Robotic Control

Figure 5 shows remote controlling and monitoring of a SCARA robot connected to the YAMAHA RCX40 controller via internet. When a series of buttons in the internet based control system are pressed, they send commands via Ethernet using the methods provided by Microsoft Winsock component library. This program is designed to connect to the controller remotely, receive the system information, send the user input to the robot, and disconnect once the operation is finished. It also provides the users (students) to view the operations at the robot area using two video cameras which can be controlled for pan and tilt motions. Once the controller (RCX40) is connected to internet (it is treated as a server as it awaits for commands from user/client and sends it to the controller).

For viewing the workspace two webcams are used as shown in the figure. This program enables to run the robot manually by typing specific destination points, or automatically by writing a program. The webcams can be accessed by 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.

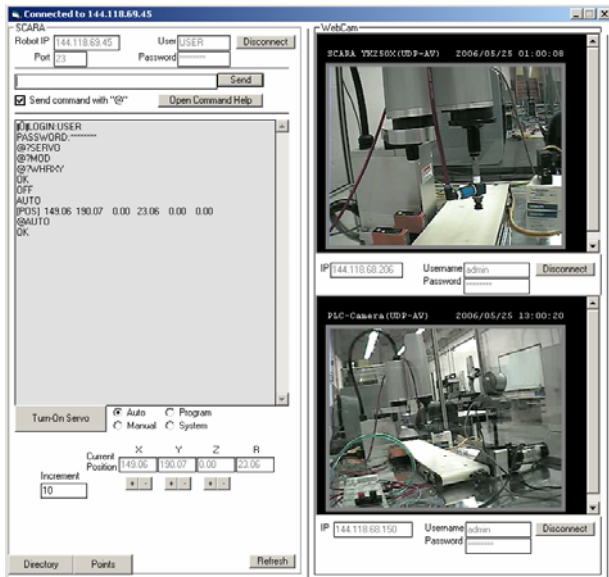


Figure 5. Connection to the server is established using Visual Basic Software and the robot is viewed simultaneously using two webcams.

Conclusions

Visual Basic programming allows students to learn the online experimentation for internet based technology. The internet based robotics laboratory activities maximize the students' learning in internet robotic control integrated with web-enabled manufacturing. The internet based technology tool can also be used for distance education and laboratory practices in mechatronics 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. The development of the internet based robotics control laboratory using Visual Basic programming has a significant and positive impact upon education, research, and service.

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

1. SME Manufacturing Engineering – Automation Technology: Robotic Automation Can Cut Costs, pp. 65 - 72, December 2005.
2. SME Manufacturing Engineering – Tomorrow's Factory: Manufacturing Industry Takes First Steps Toward Implementing Collaborative E-Manufacturing Systems, pp. 43-60, Nov. 2001.
3. SME – Machine tools begin connecting to the Internet Manufacturing, 9/2001.
4. Koc, M., Ni, J. and Lee, J., "Introduction of e-manufacturing," Proceedings of the International Conference on Frontiers on Design and Manufacturing, Dalian, China, July 2002.
5. Lee, J., 2003, "E-manufacturing—fundamental, tools, and transformation," Robotics and Computer-Integrated Manufacturing, Volume 19, Issue 6, pp. 501-507.
6. Ahn, S. H., Bharadwaj, B., Khalid, H., Liou, S. Y. and Wright, P. K., "Web-Based Design and Manufacturing Systems for Automobile Components: Architectures and Usability Studies," International Journal of Computer Integrated Manufacturing, 15, pp.555–563, 2002.
7. Feng, J., "Internet-Based Reverse Engineering," International Journal of Advanced Manufacturing Technology, January, 2002.
8. Potgieter, J and Bright, G., "Modular Mechatronics Control System for Internet Manufacturing," Proceedings of the 18th International Conference on CAD/CAM, Robotics and Factories of the Future, Porto, Portugal, pp. 529-536, July 2002.

9. Hu, H., Yu, L., Tsui, P. W. and Zhou, Q, "Internet Based Robotic System for Teleoperation, Assembly, and Automation," *International Journal of Assembly Automation*, Vol.21, No.2, pp.143-151, 2001.
10. Huang, G. Q. and Mak, K. L., "Web-integrated Manufacturing: Recent Developments and Emerging Issues," *International Journal of Computer Integrated Manufacturing*, 14, pp.3-13, 2001.
11. Shaw, M. J., "Information-Based Manufacturing with the Web," *International Journal of Flexible Manufacturing Systems*, 12, pp.115-129, 2000.
12. Taylor, K. and Dalton, B., "Internet Robotics: A New Robotics Niche," *IEEE Robotics and Automation*, Vol.7, No.1, pp.27-34, March 2000.
13. Ravindra Thamma, Luke H. Huang, Shi-Jer Lou, and C. Ray Dies, "Controlling Robot Through Internet Using Java," *Journal of Industrial Technology*, Vol. 20, No. 3, June-August, 2004.
14. Hsieh, Sheng-Jen and Patricia Yee Hsieh, "Web Based Programmable Logic Controller Learning System," 32nd ASEE/IEEE Frontiers in Education Conference, November 6 - 9, 2002, Boston, MA.
15. Richard Chiou, Yongjin Kwon, Shreepud Rauniar, and Horacio Sosa, "Laboratory Development for Robotics and Automation Education Using Internet Based Technology," *Proceedings of the American Society for Engineering Education Annual Conference & Exposition*, June 18-21, 2006, Chicago, Illinois.
16. Kwon, Y., Chiou, R., Rauniar, S. & Sosa, H., 2006, "Positioning Accuracy Characterization of Precision Micro Robot over the Internet," *Journal of Advanced Manufacturing Systems*, Vol. 5, Number 1, June 2006.
17. Kwon, Y., Chiou, R., Rauniar, S. & Sosa, H., 2006, "Web-enabled Micro Robotic System & its Positioning Characteristics," *Journal of Robotics and Computer-Integrated Manufacturing* (to appear)
18. Paul I-Hai Lin, Hal Broberg, and Aik Mon, "A Web-based Lab for Distance Learning," *Proceedings of the 2002 American Society for Engineering Education Annual Conference & Exposition*.
19. Yamaha Robots Operation Manual- Robot, Controller, and Software, Version 1.7.0.

Biographical Information

Dr. Richard Chiou's background is in mechanical engineering with an emphasis on manufacturing. His areas of research include machining, mechatronics, and internet based robotics and automation. He has secured many research and education grants from the NSF, the SME Education Foundation, and industries.

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.

Mr. Shreepud Rauniar was born in Kathmandu, Nepal in 1980. He received a B.S. in Mechanical Engineering from National Institute of Technology, Jamshedpur, India in 2003. He received his M.S. in Mechanical Engineering at Drexel University in 2006. His research interests are in designing web based applications & control systems for robotics and automation.

Dr. Horacio Sosa was a professor and the senior associate dean of the Goodwin College at Drexel University. Currently, he is the dean of College of Professional & Continuing Education in Rowan University. His research interests include theoretical mechanics, advanced materials, and engineering education.