AC 2011-2037: A REMOTE LABORATORY FOR ROBOTICS ACCURACY AND RELIABILITY STUDIES

Richard Chiou, Drexel University

Dr. Richard Chiou's background is in mechanical engineering with an emphasis on manufacturing. Dr. Chiou is currently an associate professor in the Goodwin School of Technology and Professional Studies at Drexel University. 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.

Robin Kizirian, Drexel University

Robin Kizirian completed his M.S. degree in Computer Engineering at Drexel University in Philadelphia and his B.S. degree in Computer Engineering with a minor in Computer Science at the University of Hartford in Connecticut. He is currently serving as a research assistant at the Engineering Technology department at Drexel University. Robin has been involved in various projects funded by Pfizer, NASA, NSF and Department of Education. His areas of research include Embedded Systems, Mechatronics, Efficient Solar Energy Systems, Internet-based Quality Control and 3-D Online Education.

Brittany Killen

A Remote Laboratory for Robotics Accuracy and Reliability Studies

Abstract

Accuracy and reliability studies for remote laboratories are utilized in a wide range of curricula in both engineering and engineering technology programs. In a typical undergraduate class, learning of a theoretical method is often reinforced by analysis and practical experiments. Using a remote measurement technology, theoretical learning is complemented by online experimental verifications. The paper addresses the development of a remotely controlled laboratory setup that allows monitoring the resulting reliability and accuracy. The setup incorporates modern sensors, data acquisition instrumentation, and programs to monitor and control such an application. These tools are beneficial for laboratory practices in undergraduate level quality control or instrumentation classes. The entire process shows results based on the data that is collected in real-time through the Internet. The system demonstrates the effectiveness of an online calibration tool to test for position accuracy and repeatability of a robotic device. The results provide students with a “hands-on” approach to learning, allowing them to thoroughly understand the measurement research process.

Introduction

The goal of this experiment is to ask the students to take a sensor system and incorporate it in a robotic system so that it tests a robot for accuracy and repeatability on an Internet basis. Using a group of complex systems, they had to design and integrate the systems[1-5]. As an initial step, the MP2000 LVDT Schaevitz sensor system was used as the base for testing the system. The LVDT sensor system is a system that allows you to test and control the settings of the LVDT sensors. The second component that was integrated into the system was the YK220X Yamaha Robot. This robot was used as a tool to test its accuracy and repeatability. In order to complete the integration process, the whole system had to be linked to a computer (either through a RS232 or an Ethernet connection). This procedure was performed by linking the system to a graphical user interface that is programmed in Visual Basic 6 and uses Microsoft Excel 2007. Visual Basic is our command central in this project which links the robot and the sensors together. Additionally, the Visual Basic program accesses the Microsoft Excel drivers, opening a worksheet and recording the collected data in it. In order to effectively demonstrate this program, many tests using different variable types had been run to show the accuracy and repeatability of the robot through offsets[6-13]. This test is very useful for calibrating the robot on an Internet basis and being able to record data in real-time without having to be in the same location as the robot or the MP2000 Readout Controller. Furthermore, it helps with the improvement of Internet-based controls with the Yamaha Robot (Figure 1). This project enhances the students’ programming capabilities and understanding of the accuracy parameters in the Robotics and Mechatronics fields. The students learned the basic steps of connection-oriented programming and implemented it for interfacing with external devices. By designing this program, they got educated about remote data acquisition methods and Internet-based robotic engineering applications.

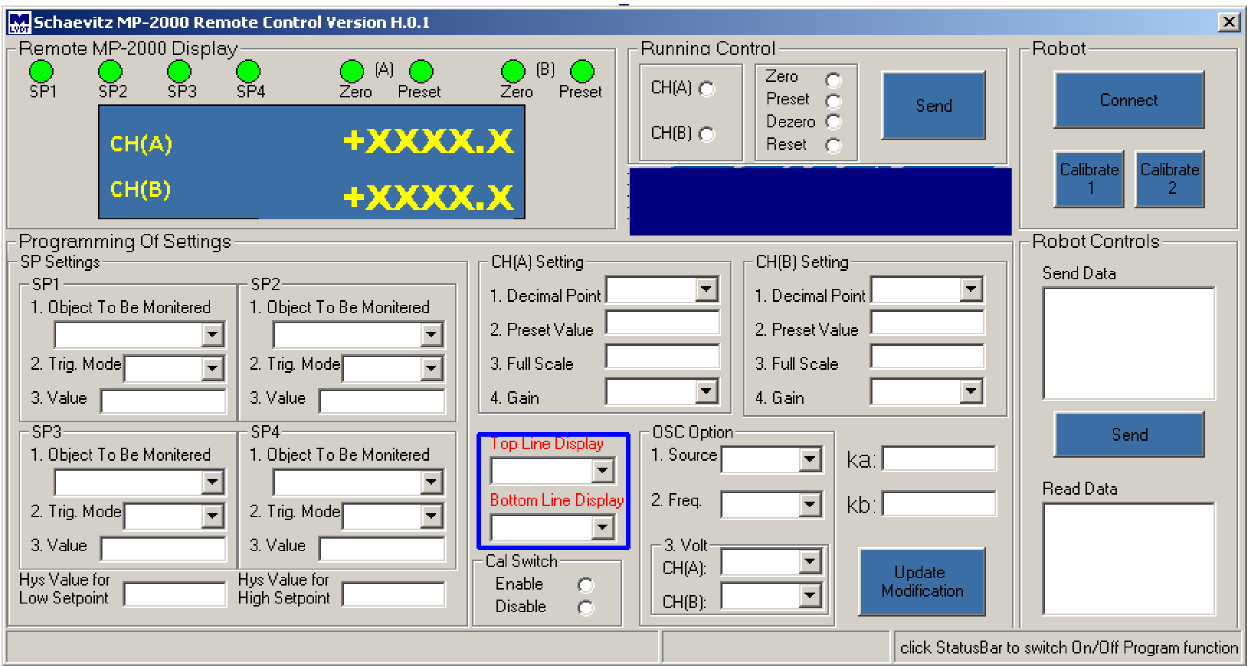


Figure 1. The Schaevitz MP-2000 Remote Control Version H.0.1 with YK220X Yamaha Robot controls.

Experimental Details

The experiment involved asking the students to setup the physical system in addition to teaching them about the different control functions. The physical system was setup so that the user has to only interact with the Visual Basic software to control the Internet-based system. This means that the students had to incorporate controls that allowed for a user to run the device without altering the physical model during testing. The way the system works is that the robot and the sensor system are setup on a metal surface. The robot is attached through the use of screws while the sensors are attached to an adjustable stand that allows the stand to stay sturdy through using the metal surface as a magnetic field.

The system has been designed so that the robot can return to its original location at the beginning of the experiment (which is programmed using the robot controls). Once this origin point is configured, the stand with the sensor is setup so that the sensor is touching the robot's axle. The height of this stand is adjustable so that you can get accurate readings. The two sensors are placed such as one is perpendicular to the other so that the sensors test both axes (x and y-axis). When the tests are run, the axle of the robot moves back and forth from the sensors. Each time the axle of the robot touches the sensor, the data is recorded to see how far the offset of the robot is from its calibrated zero value.

The Visual Basic interface is connected to a number of different components that control the entire system. These components include the Microsoft Excel worksheet that updates the data being recorded in real-time, the robot, the sensor system, the Ethernet connection or serial connection, along with the actual sensor readout controller. By setting up this model, the students have demonstrated their understanding of the basic components of the system such as the sensor and robot controls (Figures 2 and 3).

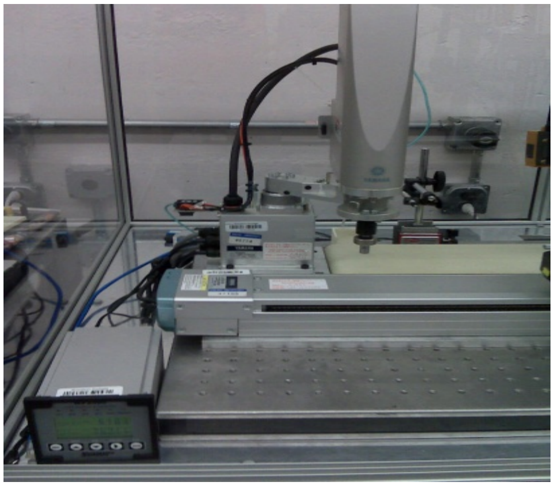


Figure 2. The setup for the sensor system using the Yamaha Robot.

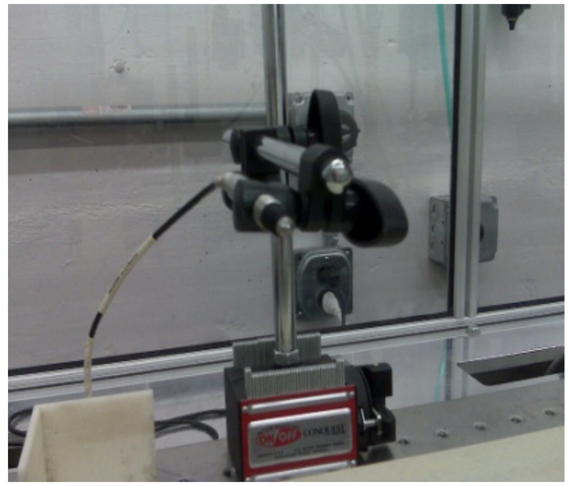


Figure 3. The setup of one of the sensors that is used to record data.

System Interface

The students were able to program the Visual Basic interface used to control the system. The stock sensor program that controls the system interface was updated. Through the use of Winsock, the Ethernet functionality was possible, allowing a remote connection to the MP2000 device and the robot. Robot controlling functions and buttons were also added to the interface as shown in Figure 4. The students also gained the knowledge of embedding the Microsoft Excel element into the GUI.

The program, itself, contains various interfaces. This includes different types of sensor tests and calibrators along with robot calibrators. On the left side, different types of sensor calibration and system settings are shown. On the right side, different calibration tests are displayed. The read out and send out boxes show the tasks the user commands the robot to perform. The user is also capable of connecting to the robot remotely through the “Robot Controls” frame and sending it online commands. Disconnecting from the program is performed by clicking the “Disconnect” button and then pressing the “Stop” button located on the graphical user interface.

To run a test, the user is required to select the correct channels on the MP2000 device and calibrate them through using the zero command. Once the calibration process is complete, the user has to choose the type of test to perform. Every time a new test is to be conducted, the entire program has to be restarted. Setting up this system shows that the students have demonstrated knowledge of the Visual Basic Software. This allows the students to improve their programming knowledge and apply it in a real-world application.

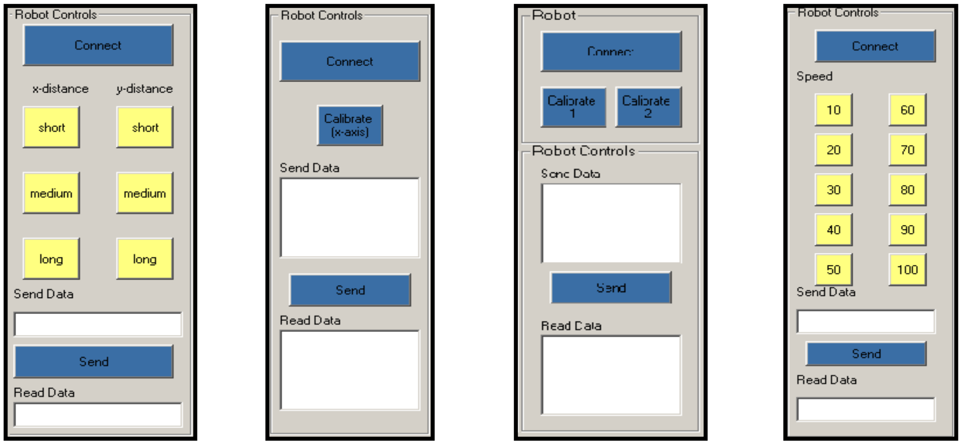


Figure 4. The interfaces used for the robot testing.

Connections to the Visual Basic Software

This section describes the connections made from the Visual Basic software to the different system devices. The first connection is established to the Yamaha Robot through the use of a TCP/IP connection. This connection allows the use of the port 23 and telnet to connect to the YK220X Yamaha Robot. The Internet capability is made possible through the use of the Winsock control library. Through using the Winsock API, the TCP commands are used automatically to connect to the robot and send and receive data from it. The commands being sent are in the form of the Yamaha online commands. The user has the choice of either controlling the robot manually through the online robotic command button or through running the calibration tests that automatically send and receive the required data to and from the robot. The other connection is to the MP2000 device that also uses the TCP/IP connection from the Visual Basic software through port 4660. The commands are sent and received from the Visual Basic application through TCP Packets. Since the MP2000 device communicates only through a serial RS-232 connection, special adjustments had to be made to make it Internet-based. Through using a GW212 Serial/Ethernet and Ethernet/Serial converter box, the remote control of the device was possible. The communication layout is described and displayed in Figure 5.

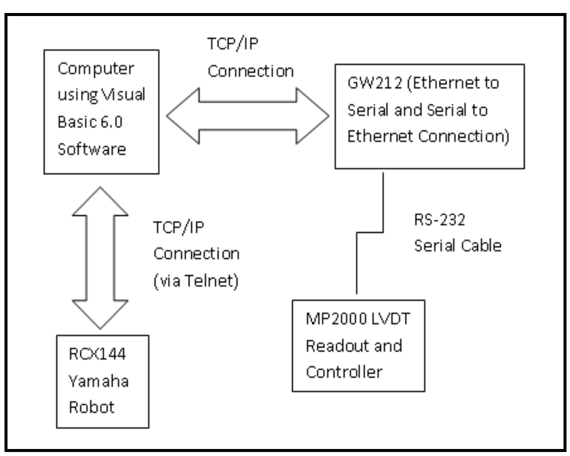


Figure 5. A system diagram of all of the connections made to the Visual Basic software.

The MP2000 device requires the Visual Basic program to send certain commands periodically. In the program, the PC sends a command of 0x80 and expects to receive a 0x80 back. This byte can be used in two different places, during the baud rate setup and handshaking. During baud rate auto detection, this byte can be sent and received. In this code, a baud rate of 9600 was set. If the baud rate is set correctly on the MP2000 Readout and Controller, this command should be able to work properly. For the purposes of handshaking, the 0x80 command is sent out periodically (minimum of 30 seconds) to ensure that the keyboard is locked and the connection is still active. If this command is not sent, the MP2000 will unlock its keyboard continuously for approximately 30 seconds.

In addition, the Visual Basic program is required to send an update packet every 300 milliseconds. This packet is sent in the format: obj1value + obj2value + Sp byte + verify byte + 0x0d. The Verify Byte is a result of byte calculation of the bytes before the Verify Byte. In this case, these bytes consist of obj1value, obj2value and Sp byte. In order to calculate the value of the Verify Byte, the sum of the previous bytes is found. Then, the 2's Complement Rule is applied to the sum of the bytes which is then passed through a modulus of 128. The formula calculates the correct value for the Verify Byte which is used by the MP2000 device or the Visual Basic program to check whether the correct data has been sent or received. 0x0d stands for the completion of an entire command. The period at which these packets are sent are controlled through timers within the code. With the correct timer configuration, the packets can be sent at the requested rates.

Other types of packets include the "Request for Data" packet, the "Receive Data" packet, and the "Change Setting" packet. The requests for data are sent in the format: command ID+ 0x7F + verify byte + 0x0d (with 0x7F as a fixed byte). After sending this command, the Visual Basic program expects to receive the following data packet back: Command ID + Data String + Verify Byte + 0x0d. This code is used to check the values and settings of the MP2000 Readout and Controller. In a case of an error, the PC will repeat the last sent command as many times necessary until the command goes through.

The "Change Setting" command is sent in the following format: Command ID + Command String + Verify Byte + 0x0d. After sending this command, the Visual Basic program expects to receive: Command ID + 0x0d within approximately one second or it will send the command again. This occurs during the enabling or disabling of a setting. With this setting, there are three different ways the code can write and one way it can read. The three different types of write controls are known as float, k factor, and char. They all have different send out formats for the MP2000 Readout Controller. Some of the write controls use commands such as hex bytes or just bytes to write data, depending on what kind of data the user is sending. For instance, if one is sending a command to monitor object SP1, the code would be sent out in the format of "n." This means that it will be sent in a hex byte format for the MP2000 Menu Display Item. The read control only needs one code because it is only sending data of one type. Figure 6 displays the different types of packets that are sent back and forth between the MP2000 device and the Visual Basic program. By understanding these concepts, the students were able to learn about the different communication protocols and data formats used in RS232 and Ethernet. They also learned how to send certain instructions to control a microcontroller.

Command Update Packet Format:

Obj1value	Obj2value	Sp byte	Verify byte	0x0d

Request for Data Packet:

Command ID	0x7F	Verify byte	0x0d

Receive Data Packet:

Command ID	Data String	Verify Byte	0x0d

Change Setting Packet:

Command ID	Command String	Verify Byte	0x0d

Change Setting Confirmation Packet:

Command ID	0x0d

Figure 6. The packets used in the program.

Embedding Microsoft Excel for Reporting

For this software, the students learned how to incorporate remote real-time data into Microsoft Excel using the Visual Basic environment. They had to learn about the different drivers that Excel uses to open a new worksheet and input the data into specific cells. As shown in Figure 7, the code for the Microsoft Excel program is incorporated into the Visual Basic program. The code shows how Microsoft Excel is embedded into the program by adding the corresponding Visual Basic code. In the first part of the code, the Excel chart is embedded so that every time one of the test buttons is clicked, the Excel spreadsheet will open. A timer has been embedded to record a new piece of data into Excel from the sensor system every 1000 milliseconds. Sample data recorded from this system is displayed in Figure 8.

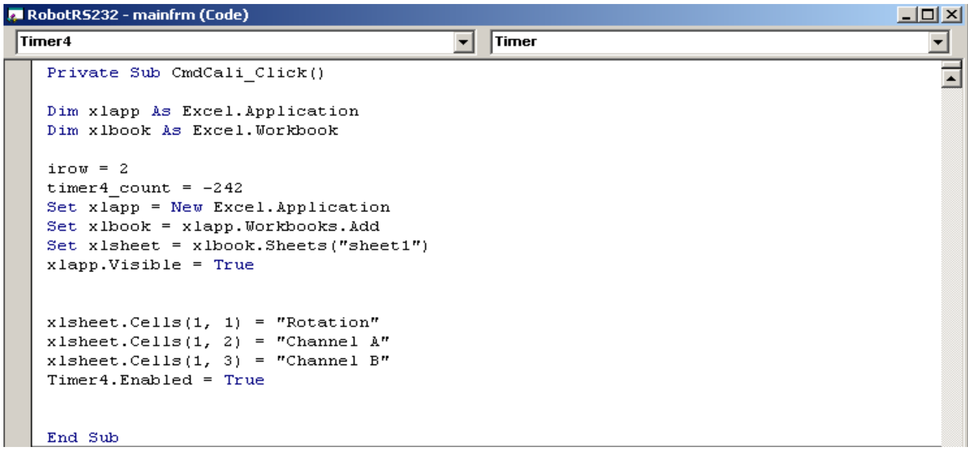
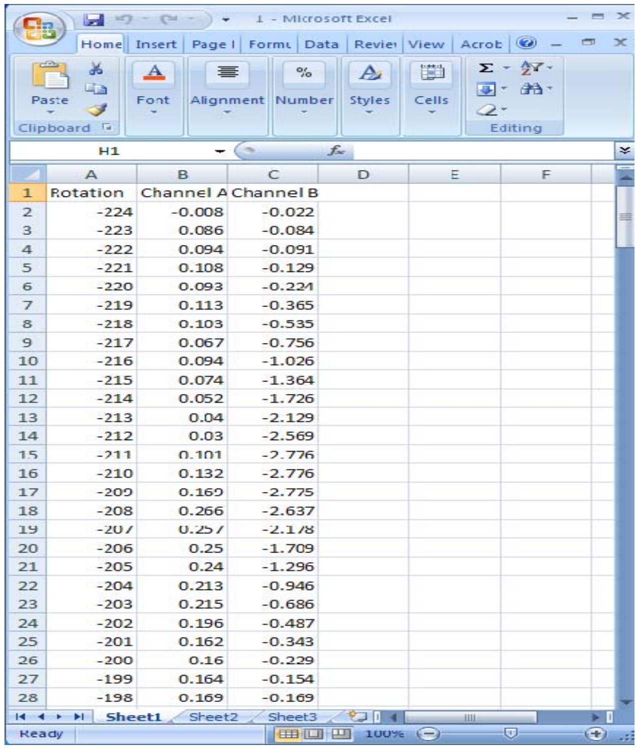


Figure 7. The embedding of the Microsoft Excel code into the Visual Basic Code for the sensors.



	A	B	C
1	Rotation	Channel A	Channel B
2	-224	-0.008	-0.022
3	-223	0.086	-0.084
4	-222	0.094	-0.091
5	-221	0.108	-0.129
6	-220	0.093	-0.224
7	-219	0.113	-0.365
8	-218	0.103	-0.535
9	-217	0.067	-0.756
10	-216	0.094	-1.026
11	-215	0.074	-1.364
12	-214	0.052	-1.726
13	-213	0.04	-2.129
14	-212	0.03	-2.569
15	-211	0.101	-2.776
16	-210	0.132	-2.776
17	-209	0.169	-2.775
18	-208	0.266	-2.637
19	-207	0.257	-2.178
20	-206	0.25	-1.709
21	-205	0.24	-1.296
22	-204	0.213	-0.946
23	-203	0.215	-0.686
24	-202	0.196	-0.487
25	-201	0.162	-0.343
26	-200	0.16	-0.229
27	-199	0.164	-0.154
28	-198	0.169	-0.169

Figure 8. A sample of what the data being recorded in real-time via the Microsoft Excel 2007 application looks like while running a calibration test.

Robot Calibration

This section describes how the user is able to control the YK220X Yamaha Robot and the MP2000 LVDT Readout and Controller on an Internet basis. As shown in Figure 1, the Visual Basic system consists of one display screen that allows the user to control all the elements that go into this system. This display screen consists of many different elements. First of all, in the top left hand corner the user will see a frame labeled as “Remote MP-2000 Display.” This screen allows the user to view the values of the MP2000 LVDT Readout and Controller itself. Furthermore, the user is able to see if different functions are enabled or disabled based on if the green light is on (for example SP1). Along with this, there is a frame labeled “Programming of Settings.” In this box, these functions change the different settings of the MP2000. The user is able to change any of these functions by pressing the “Update Modification” button. Moreover, there is one more control box for the MP2000. It is labeled as “Running Control.” This box allows the user to switch the values of the two channels by clicking on the “Send” button. Along with these frames, there are two more frames that appear in this display screen. The first frame is labeled as “Robot.” With this frame, the user is able to connect to the robot and is able to perform both calibration tests with the robot through the Internet. The “Robot Controls” frame allows the user to send commands to the robot (via the “Send Data” box) and allows the user to see which commands the robot is sending (via the “Read Data” box).

When the Visual Basic program is first opened, the user should press the “Start” button located in the top left hand corner of the screen (this will turn the system on). The user must then wait until the system is done checking to see if there are any errors with its connection state. The user is then able to perform the calibration tests. The MP2000 LVDT Readout and Controller automatically connects itself to the PC when the program is turned on. Updated values that appear on the screen signify that the program is running correctly. However, in order to connect to the robot, the user must press the “Connect” button. The user knows that the Yamaha Robot is connected if an “OK” message in the “Read Data” box is displayed. Once this process is complete the user is ready to run the calibration tests.

Data Analysis

Travel Distance Testing

Students collected data to measure the traveling accuracy in a robotic cell as shown in Figure 9. Students also had to learn how to properly use the system they had created. In order to do this, programming was done to the Visual Basic software to perform tests in which they can travel different distances on the x-axis and y-axis. For each of these two tests, there is a set “short” distance, “medium” distance, and “long” distance. The difference between each of the short, medium, and long distances is about 10mm. This would help determine if any of these settings affect our data results. It was also decided to perform these tests using holders and no holders. This would help determine if payload (weight) was a factor that needed to be looked into. The test was also performed using the 3V and 1V settings of the sensor system to help determine if the settings of the voltage altered the offset or the sensor values. The results of these experiments are shown in Figures 10 and 11. It should be noted that all the results are in mm units.

Figure 9: Student conducting traveling distance testing and calibrating robotic accuracy

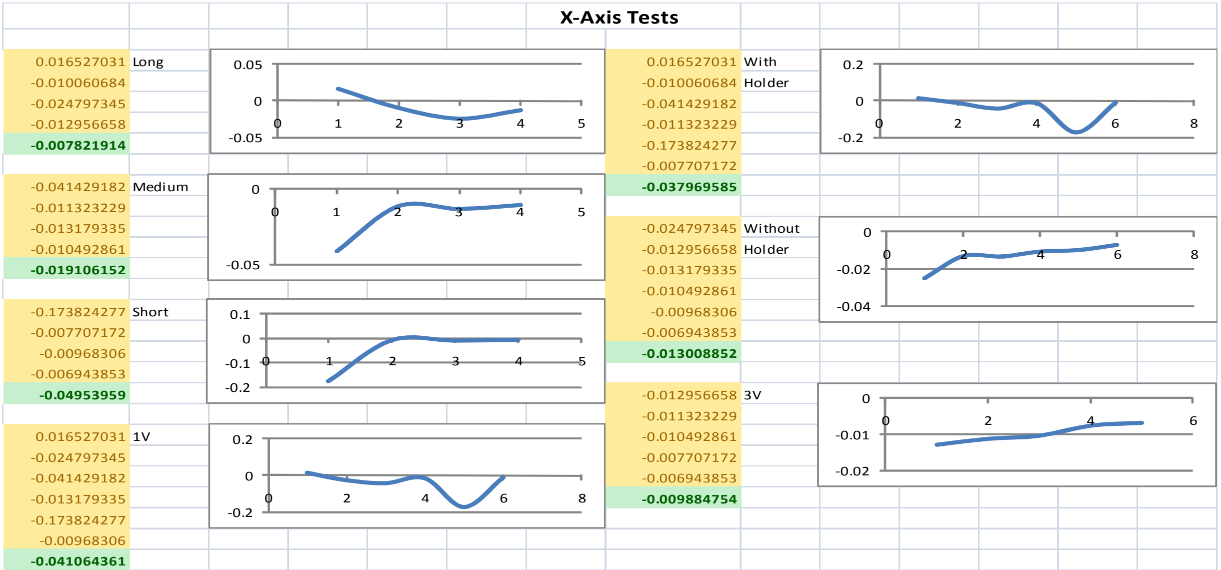


Figure 10. The x-axis tests run during the travel distance testing in mm.

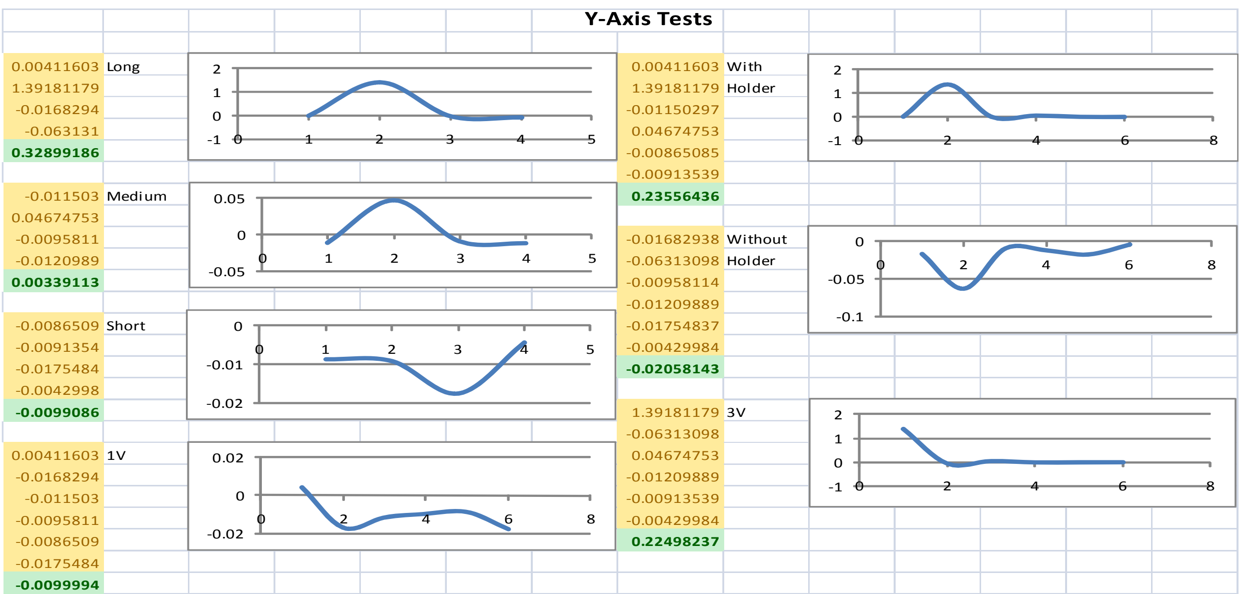


Figure 11. The y-axis tests run during the travel distance testing in mm.

The results for the variation in the x-axis were the following. When comparing the travel distances it was found that the long distance has an average offset of -0.008, medium has an average of -0.019, and short has an average of -0.050. As shown, when the robot moves a longer distance its offset is not as high. When comparing the payloads, the tests with the holders had an average offset of -0.037 (compared to the test without the holder averaging at -0.013). Overall, as one can see, the lower the payload the less offset the robot is. Lastly, voltage was compared during these tests. The 1V had an average offset of -0.041 compared to the 3V at -0.010. As one can see, the 3V has a lower offset during these tests.

The other axis tested in this experiment for this variable was the y-axis. This was tested to determine not only if the travel distance affected the robot, but also to see if one axis is more precise than the other. When comparing distances the short distances had an average of -0.01, the medium had an average of 0.003, and long had an average of 0.33. As one can see for this test the medium distance test was the most accurate. When comparing the payload it was found that with the holder had an average of 0.24 and without a holder had an average of -0.02 making without the holder the more accurate option. When comparing voltages the test showed that the 1V tests had an average of -0.01 and the 3V had an average of 0.22. This means that the 1V test is more accurate.

Discussion

In order to find these results, students had to make sure their data was accurate. For this reason, Microsoft Excel was connected to the program to record the sensor values in real-time. Once the data is recorded, Excel's built-in tools were used to calculate the slope of the recorded data. In order to calculate slope, the following equations were used.

$$a = \frac{\Sigma y - m\Sigma x}{N} \tag{1}$$

$$b = \frac{N\Sigma(xy) - \Sigma x \cdot \Sigma y}{N\Sigma(x^2) - (\Sigma x)^2} \tag{2}$$

$$r = \frac{N\Sigma(xy) - \Sigma x \cdot \Sigma y}{\sqrt{[N\Sigma(x^2) - (\Sigma x)^2] \cdot [N\Sigma(y^2) - (\Sigma y)^2]}} \tag{3}$$

where **x** and **y** are the variables
N is the number of values
a is the intercept point of the regression line
b is the slope of the regression line
r is the correlation coefficient

Furthermore, the average of the slopes was found by using the equation:

$$\text{Average} = \frac{b_1 + b_2 + b_3 + \dots}{\text{Total number of slopes}} \tag{4}$$

From these averages, the students were able to graph the average of the slopes to find their final results.

Conclusion

Once all of these tests were recorded, it was decided to compare the x axis to the y axis tests based on the different variables. It was found that the long distance test with the x-axis was more accurate. Additionally, for the medium and short distance tests, the y-axis was more accurate. When looking at payload, it was found that the x-axis was better for both. When looking at voltage, however, it was found that at 1V, the y-axis is better and at 3V, the x-axis is more accurate. As you can see the accuracies of the movement are different as shown in these tests.

This paper demonstrates how an undergraduate student's knowledge of a research process is enhanced through designing an Internet-based robotic position accuracy calculator. It shows how the use of online calibration can be useful when used for testing position accuracy. This can be a very helpful tool that will allow the user to test for accuracy without being in the same remote location as the robot or the MP2000 system. It will also help with the accuracy of data results for different types of tests involving the use of a robot. Overall, this system can be a very beneficial tool in demonstrating the use of Internet-based calibration tests.

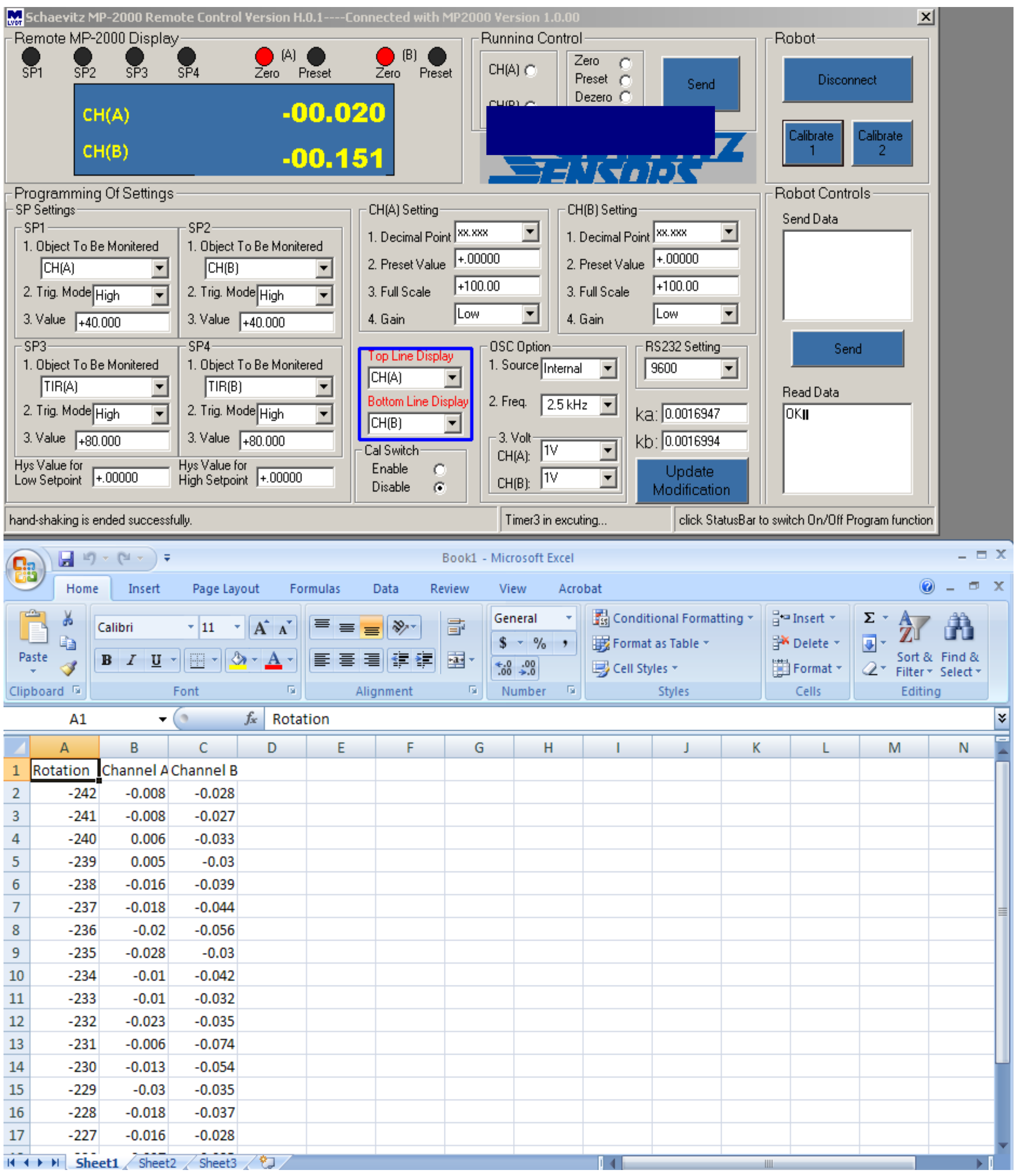


Figure 11. The Visual Basic System running during testing.

Acknowledgement

The authors would like to thank the National Science Foundation (Grant No. NSF-DUE-CCLI-0618665) for its financial support of the project.

Bibliography

1. Richard Chiou, Yongjin Kwon, and Prathaban Mookiah, "Manufacturing E-Quality Through Integrated Web-enabled Computer Vision and Robotics," *The International Journal of Advanced Manufacturing Technology*, Volume 43, Numbers 7-8, pp. 720-730, August, 2009.
2. Richard Chiou**,** Michael Mauk, Sweety Agarwal**,** and Yueh-Ting Yang, "Development of E-quality Laboratory Modules for use in Engineering Quality Control Courses," ASEE Annual Conference & Exposition, Austin, TX, June 14-17, 2009.
3. Kwon, Y., Rauniar, S., Chiou, R. & Sosa, H., "Remote Control of Quality Using Ethernet Vision and Web-enabled Robotic System" *J. of Concurrent Engineering: Research and Applications,* Vol. 14, No. 1, pp. 35-42, 2006.
4. Greenway B. "Tutorial robot accuracy", *Industrial Robot: An International Journal*, 27(4) 257-265, 2000.
5. Dagalakis, N.G. "Industrial robotics standards", in Nof, S.Y. (Eds), Handbook of Industrial Robotics, 2nd ed., Wiley, NY, USA, pp.449–59,1999.
6. Brethe, J.-F., Vasselin, E., Lefebvre, D. Dakyo, B. "Determination of the Repeatability of a Kuka Robot Using the Stochastic Ellipsoid Approach", Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 4339–4344, 2005.
7. Riemer R., Edan Y. "Evaluation of influence of target location on robot repeatability" *Robotica*, 18(4) 443-449, 2000.
8. Mehrez, A., Hu, M., Offodile, O.F. "Multivariate Economic Analysis of Robot Performance Repeatability and Accuracy", *Journal of Manufacturing Systems*, 15(4) 215-226, 1996.
9. Offodile O.F., Ugwu K.O., Hayduk L. "Analysis of the Causal Structures Linking Process Variables to Robot Repeatability and Accuracy", *Technometrics*, 35(4) 421-435, 1993.
10. Shiakolas P.S., Conrad K.L., Yih T.C. "On the accuracy, repeatability, and degree of influence of kinematics parameters for industrial robots", *International Journal of Modelling and Simulation*, 22(3) 1-10, 2002.
11. ISO 9283 – "Manipulating industrial robots. Performance criteria and related test methods". ISO, Geneva 1998.
12. ANSI/RIA R15.05-1-1990 (R1999) "American National Standard for Industrial Robots and Robot Systems - Point-to-Point and Static Performance Characteristics – Evaluation", American National Standards Institute, 1990.
13. Montgomery, D.C. *Design and Analysis of Experiments*, 6th edition, Wiley, 2005.