Wireless Control of a Robotic Arm Using 3D Motion Tracking Sensors and Artificial Neural Networks

Fernando Rios, Imani Augusma, and Rocio Alba-Flores
Georgia Southern University, Statesboro, GA, U.S.

frios@georgiasouthern.edu, ia00336@georgiasouthern.edu, ralba@georgiasouthern.edu

Abstract

This paper describes the hardware and software components of an intelligent system that is able to wirelessly control the movements of a robotic arm for mimicking human arm gestures. For the implementation of the system, a laptop computer, 3D wireless motion tracking sensors, an Artificial Neural Network (ANN) classifier, and a microcontroller were used to drive the six degrees of freedom robotic arm. Results demonstrated that the robotic arm is capable of mimicking the human arm motions. The overall accuracy of the ANN classification system was 88.8%. Due to limitations of non-continuous rotation servos, some movements have to be limited or changed in order for the robotic arm to perform the equivalent to the human arm movement.

Introduction

Robotic technologies have played and will continue playing important roles in helping to solve real life problems. One of the most important fields in the development of successful robotic systems is the Human-Machine Interaction. This paper describes the development of a system that uses an ANN classifier to control a robotic arm that is able to mimic the movements of a human arm. In this work the user can directly control a six degrees of freedom (6-DOF) robotic arm by performing arm motions with his/her own arm. The system uses inertial measurement units to sense the movements of the human arm.

Alternative approaches that have been used to develop human-machine interaction include the use of Electromyography (EMG) signals to capture and analyze the electrical activity in human muscle tissue [1, 2]. However, due to the electrical signals being minuscule, it makes difficult to process the data using this method. Other techniques that have been used include using gyroscopes and accelerometers, for example in [3] the authors developed a low cost wireless motion sensing control unit using three separated sensors: accelerometer, gyroscope, and magnetometer. They used a three degree of freedom robotic arm to control the elbow and wrist positions. Matlab software was used to process the signals coming from the sensors and generate the PWM signals to control the servomotors; the accuracy of the developed system was not specified. An alternate approach that recently has started to gain popularity among researchers is to track muscle activity using inertial measurement units (IMUs) and air pressure sensors [4, 5]. IMUs integrate an accelerometer, a gyroscope, and a magnetometer together to measure three directional static and dynamic movements. In [6] the researchers developed a mimicking robotic hand-arm using flex sensors for individual fingers and multiple three-axis accelerometers. By using four encoders, they divided individual processing units for the fingers and arm to increase the processing speed. They used a high speed microcontroller to control the input and output processing. The researchers developed a glove to house all of the components and for a user to wear.

The research presented in this paper describes the design and development of a wireless control system that gives commands to a robotic arm. The commands are given by a human subject wearing two inertial measurement units (IMU) on his/her arm. The IMU contains an accelerometer, a gyroscope and a filter in a small unit [7]. The robotic arm used has six degrees of freedom and can perform elbow, wrist, and shoulder joint movements. Kalman filtering is also integrated in the IMU to reduce potential noise and smooth signal data. After obtaining muscle activity information from the IMUs, the data is then fed into a trained Artificial Neural Network (ANN). An ANN is an adaptive and powerful system that replicates the biological system. The system has the ability to recognize linear and nonlinear relationships between input and output data, similar to the human brain. ANNs are used for data classification and pattern recognition. The ANN processes information using various layers that are linked together: the input layer, the hidden layer, and the output layer. Each layer is composed of interconnected 'nodes' which represent neurons. Data is fed into the input layer which communicates with the hidden layer and the hidden layer then provides the corresponding output to the input information to the output layer. The system learns by example using the algorithm called backpropagation, also known as the backwards propagation error. The ANN sees input data repeatedly and then makes a 'guess' to the corresponding output and then compares it to the actual output. The hidden layer computes an error which will be fed back into the network to adjust the weights. Each input and hidden is multiplied by a predetermined weight. The weights are meant to minimize the error as much as possible to minimize misclassifications. The weighted input layer and the weighted hidden layer are then summed together. If the summation does not equal one, then adjustments will occur during each cycle or 'epoch' until the summation is as close to one as possible which means the error cannot be minimized anymore and this input corresponds to an output. This is called training the Artificial Neural Network. Rote memorization can occur if the network is trained to recognize only one type of input. This is called over training the system. For an ANN to work properly, it must be trained with various types of input data to a desired output. Figure 1 is a basic structure of an Artificial Neural Network. After training the system, it will be able to see new input data and adjust the weights accordingly to produce an accurate output [8-11]. The ANN shall then determine the corresponding movement that is performed by the user based on the test set that was used to train the ANN. After the network decides the movement, this information is sent to the robotic arm to emulate the human arm motion. In the next sections more detailed description of the developed system is presented.

Main System Components and Methods

The 3D wireless motion tracking sensors (inertia measurement units, IMU) [7] were placed on a human arm at two locations - the wrist and the upper arm (see fig.12). At a sampling rate of 100 Hz, the sensors track the XYZ-coordinate inertial data and the Euler angles of the subject's arm as they perform a specific movement. Nine pre-defined arm motions were selected in this work (refer to the Arm Movement Descriptions section below). The raw data (XYZ-coordinates from the IMUs) is processed computing the Root Mean Square (RMS) and the Rectified Average. Normalized data is then used to train a multi-layer, feed-forward Artificial Neural Network to classify the arm motions.

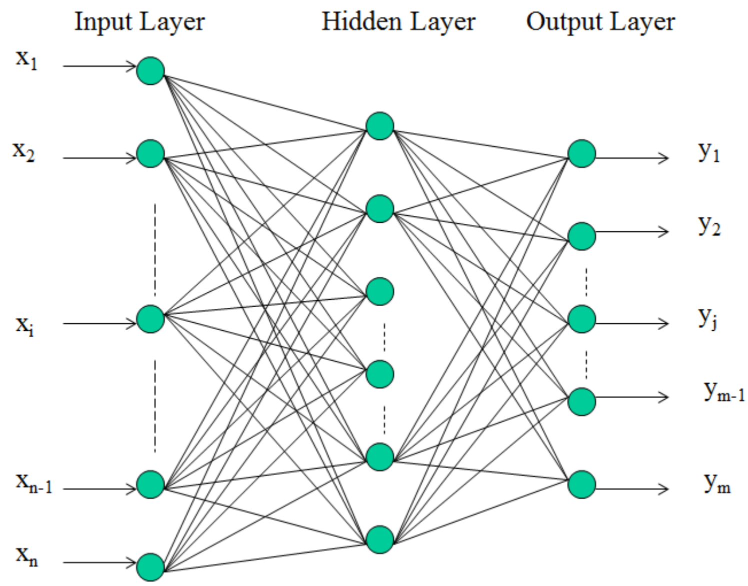

Figure 1. Schematic Diagram of a Multilayer Feed Forward Neural Network

Matlab software was used for the design and implementation of the ANN classifier. An Arduino microcontroller was used to control the servo motors in the robotic arm. The Arduino was directly interfaced to the laptop computer running the ANN classifier that uses the ANN toolbox of Matlab. The data set used in this research consisted of 180 data vectors (from 4 different subjects, each performed each motion 5 times). 70% of the data was used for training the ANN, 10% for validation, and 20% for testing. A totally independent set of arm motions (from a 5th subject) were used to determine the accuracy of the ANN classification system.

The main components used for the capture and signal processing were: a) an IMU board composed of a digital three-axis accelerometer and a digital three-axis gyroscope [7]; b) a Zigbee RF wireless communication modules [12] to transmit and receive data; c) a low cost microcontroller, Arduino Mega [13], to control the input and output processing; d) a (6-DOF) robotic arm that used servo motors to control the joints (see fig. 11), the servos were controlled using pulse width modulated signals; e) a Kalman filter that was used to reduce noise and have smooth signal data from the accelerometers and the gyroscope. The interaction of these components is shown in (fig. 2). All the information was processed in real time using Matlab. There robotic arm mimicking system was successful, but currently the system is unilateral.

The user performs one of the pre-defined arm movements and Xsens Technologies' Software captures the waveform of the acceleration data. This data is then exported into an Excel spreadsheet. Matlab imports this data. Due to the sensors sampling at a rate of 100 Hz and each arm movement taking approximately three seconds to perform, Matlab averages the data by computing the Average Rectified Value (ARV) and the Root Mean Squared (RMS) value. These averages are then fed into the trained Artificial Neural Network. The Artificial Neural

ISBN 978-1-60643-379-9

Network then determines the corresponding arm movement that was performed by the human subject. The corresponding arm movement is then performed by the robotic arm.

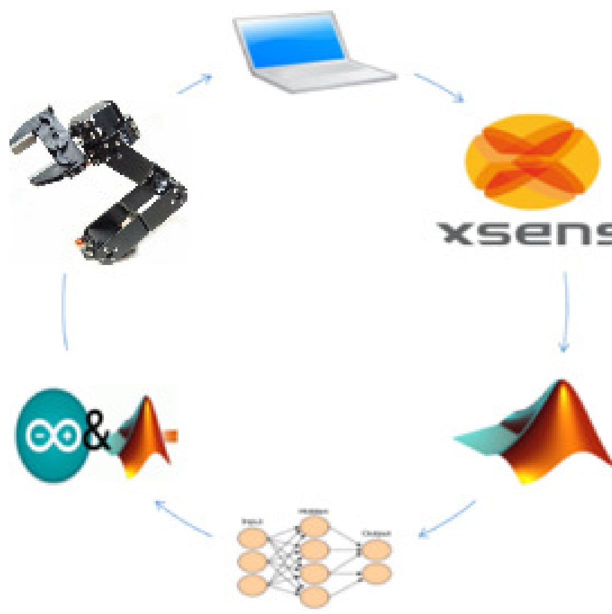

Figure 2. Main Components of the Wireless Robotic Control System.

Arm Movement Descriptions

For capturing the arm's movements two inertial measurement units were used. Sensor 1 was placed on the user's upper arm and Sensor 2 was closest to the wrist as shown in fig. 12. Sensor 1 indicates the unit on the user's upper arm and Sensor 2 indicates the unit on the user's wrist. There are a total of nine pre-defined arm movements that the ANN can identify and the robotic arm can mimic: Arm Extension, Arm Raise, Arm Raise Elbow Bend, Clockwise Windmill, Counterclockwise Windmill, Shoulder Touch, Side Arm Raise, Wipe Right, and Wrist Rotation. Each arm movements use the same initial position, this is straight down by the user's side, fingers pointing to the floor (fig. 12). The following paragraphs explain each arm movement and show sample of the data obtained from each IMU.

Motion 1. Arm Extension.

When performing arm extension, the user bends the elbow until the arm is parallel to the floor, extends the arm all the way forward and then returns to the initial position.

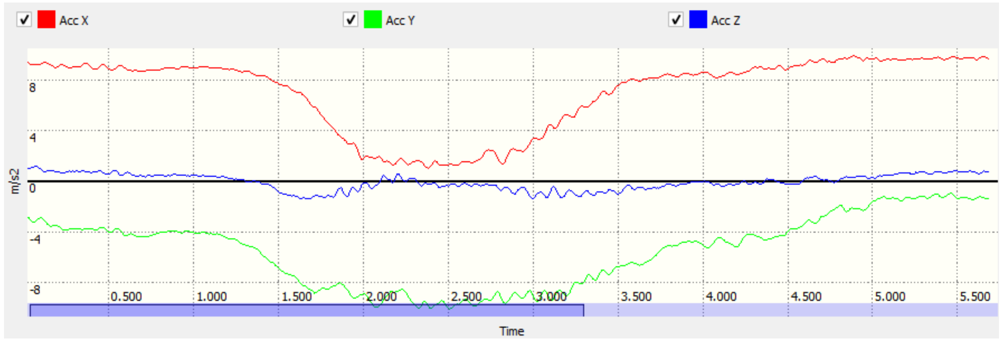

Figure 3a). Sensor 1 Acceleration Waveform for Arm Extension

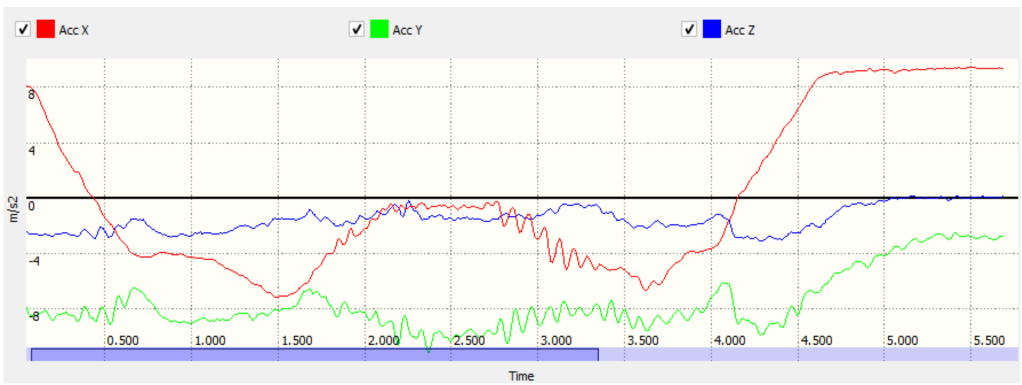

Figure 3b). Sensor 2 Acceleration Waveform for Arm Extension

Motion 2. Arm Raise.

When performing arm raise, the user rotates the wrist until palm is facing the body, raises the arm to shoulder height, and then returns to the initial position.

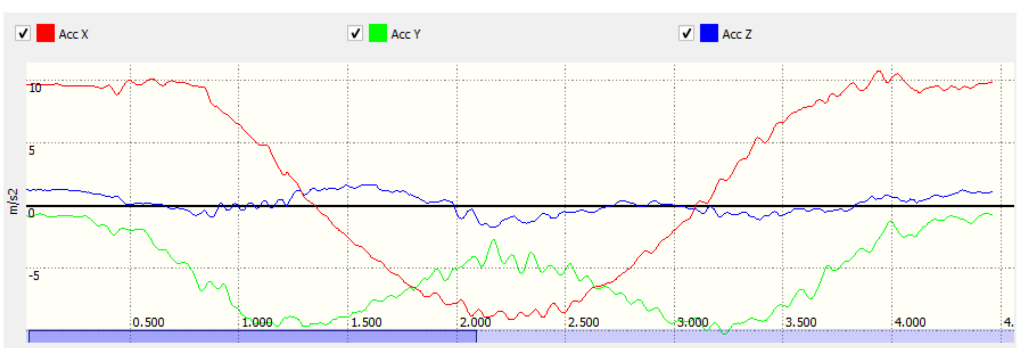

Figure 4a). Sensor 1 Acceleration Waveform for Arm Raise

ISBN 978-1-60643-379-9

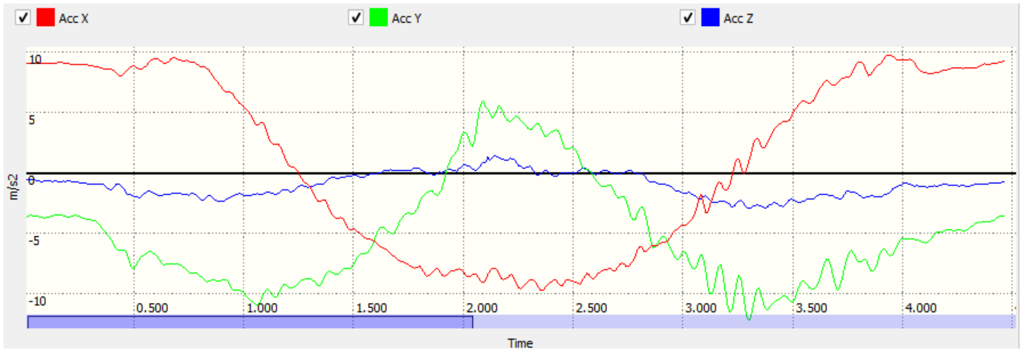

Figure 4b). Sensor 2 Acceleration Waveform for Arm Raise

Motion 3. Arm Raise Elbow Bend.

When performing Arm Raise Elbow Bend, the user rotates the wrist until palm is facing the body, raise arm to shoulder height, bend elbow inwards across the body and then returns to initial position.

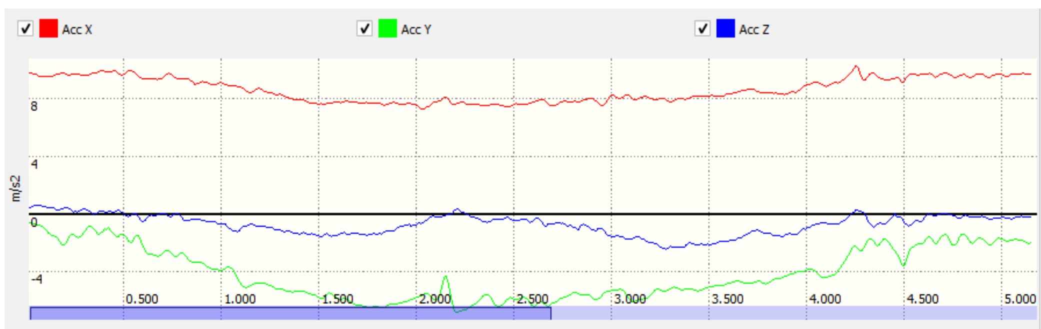

Figure 5a). Sensor 1 Acceleration Waveform for Arm Raise Elbow Bend

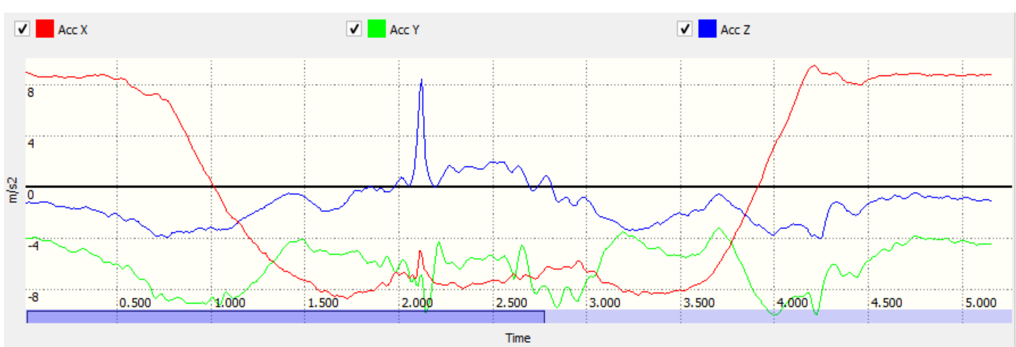

Figure 5b). Sensor 2 Acceleration Waveform for Arm Raise Elbow Bend

Motion 4. Clockwise Windmill.

When performing Clockwise Windmill, the user rotates shoulder 360 degrees clockwise and then returns to initial position.

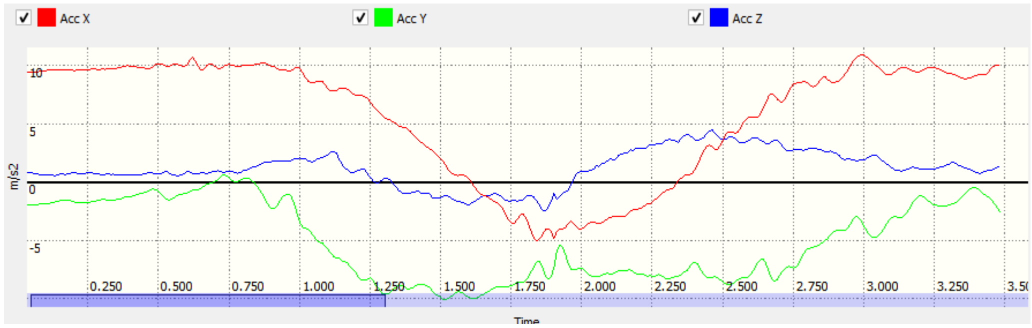

Figure 6a). Sensor 1 Acceleration Waveform for Clockwise Windmill

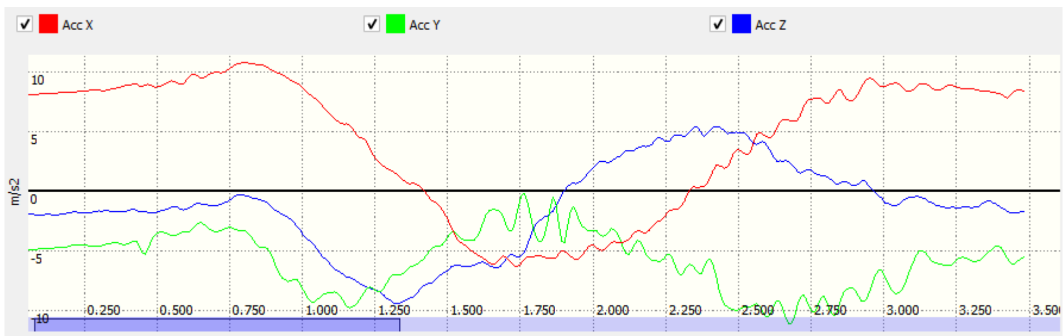

Figure 6b). Sensor 2 Acceleration Waveform for Clockwise Windmill

Motion 5. Counterclockwise Windmill.

When performing Counterclockwise Windmill, the user rotates shoulder 360 degrees counterclockwise and then returns to initial position.

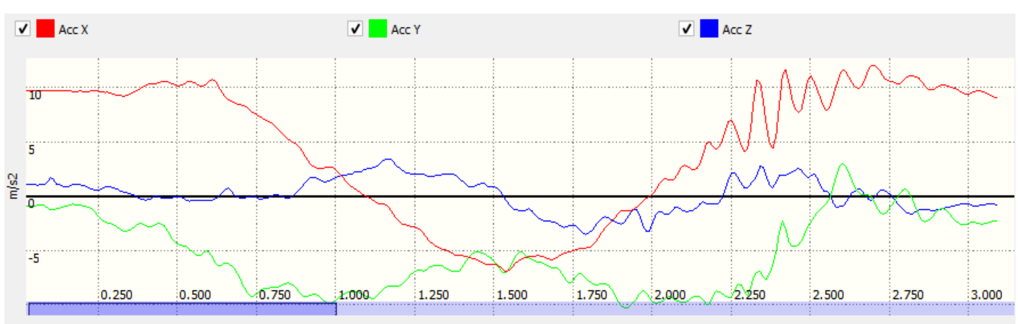

Figure 7a). Sensor 1 Acceleration Waveform for Counterclockwise Windmill

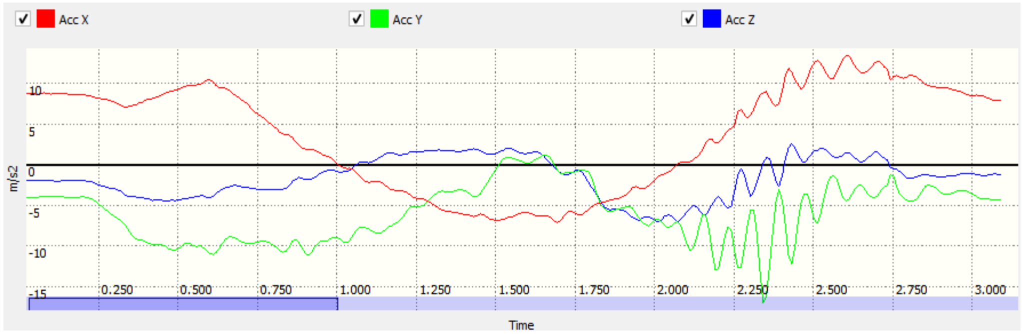

Figure 7b). Sensor 2 Acceleration Waveform for Counterclockwise Windmill

Motion 6. Shoulder Touch.

When performing Shoulder Touch, the user bends elbow until the user's hand is able to touch the shoulder and then returns to the initial position.

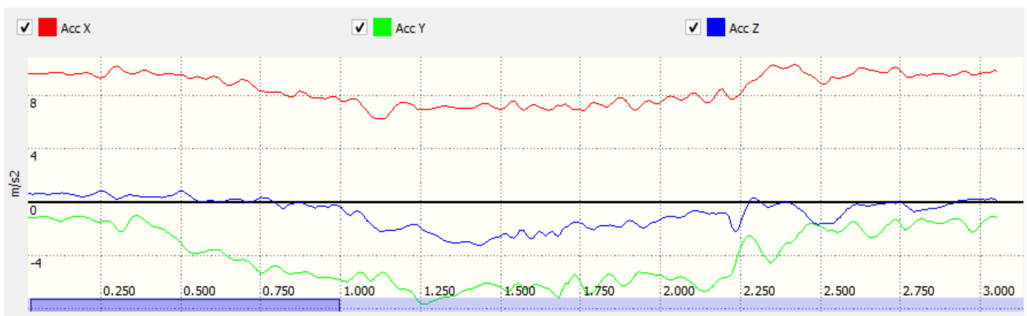

Figure 8a). Sensor 1 Acceleration Waveform for Shoulder Touch

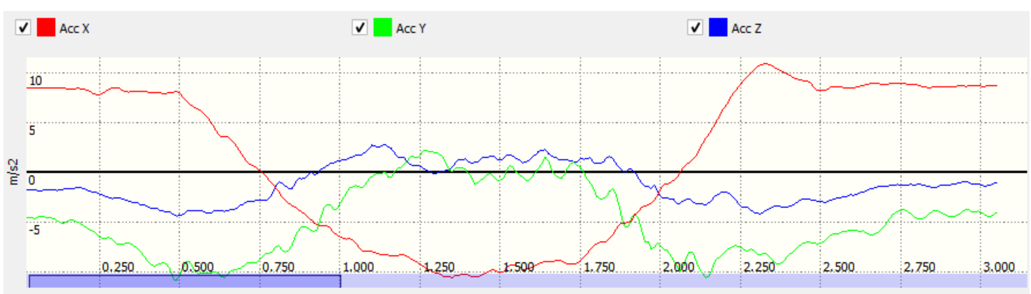

Figure 8b). Sensor 2 Acceleration Waveform for Shoulder Touch

Motion 7. Side Arm Raise.

When performing Side Arm Raise, the user lifts arm straight to the side and up 90 degrees until parallel to the floor and then returns to the initial position.

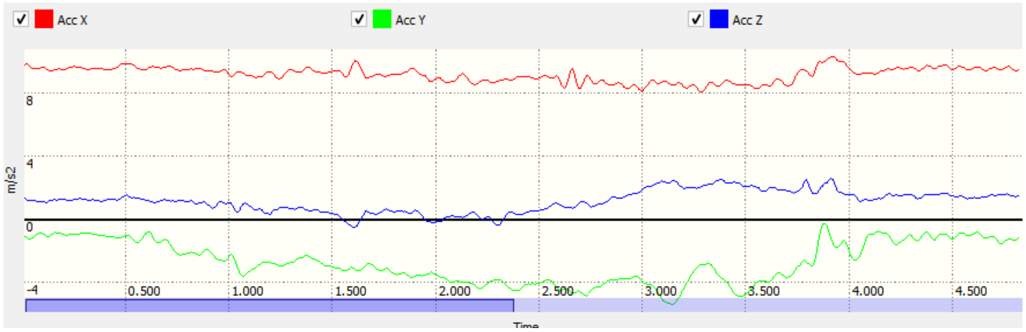

Figure 9a). Sensor 1 Acceleration Waveform for Side Arm Raise

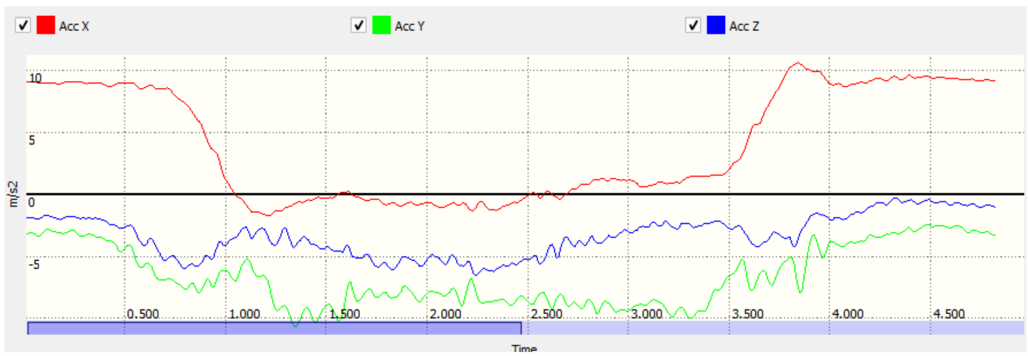

Figure 9b). Sensor 2 Acceleration Waveform for Side Arm Raise

Motion 8. Wipe Right.

When performing Wipe Right, the user bends elbow until arm is parallel to the floor, rotates forearm to the right as far as possible and then returns to the initial position.

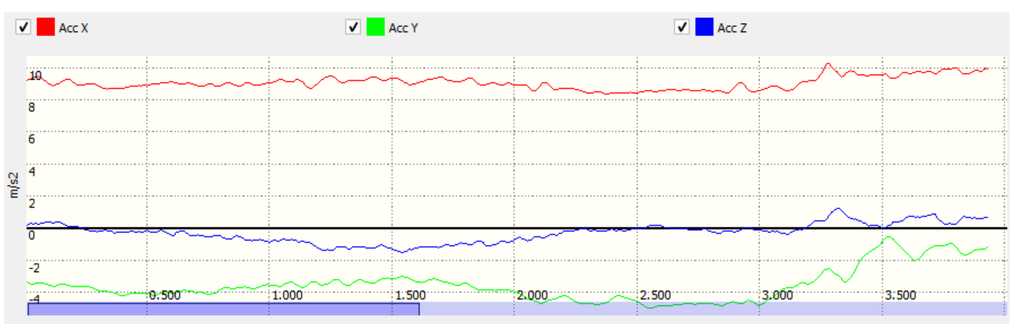

Figure 10a). Sensor 1 Acceleration Waveform for Wipe Right

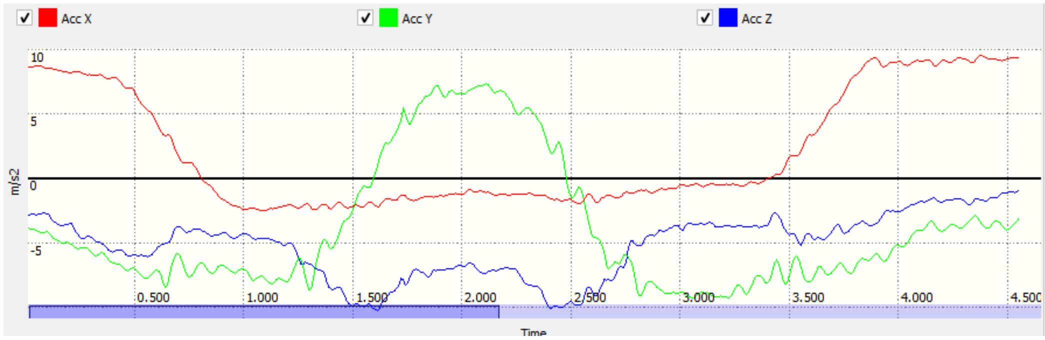

Figure 10b). Sensor 2 Acceleration Waveform for Wipe Right

Motion 9. Wrist Rotation.

When performing Wrist Rotation, the user bends elbow until parallel to the floor, rotates wrist inward and back again and then returns to the initial position.

Artificial Neural Network

The robotic arm had to be trained to distinguish among varieties of arm movements. To make this possible, an ANN was trained using the backpropagation algorithm. The network was trained using the triaxle acceleration sampling data obtained from the inertial measurement units. There were a total of 20 data sets for each movement. The inertial measurement units sampling at a rate of 100 Hz and each arm movement took approximately 3 to 5 seconds to complete, then each trail contained hundreds of individual data points. To be able to utilize this data to train the ANN, the data was condensed into two statistical measurements: the Average Rectified Value (ARV) and the Root Mean Square (RMS) Value. The ARV, is the average of the absolute values in the data set whereas the RMS is the square root of the average. This would create a characteristic value that could be used as an input for the ANN.

$$x_{ARV} = \frac{1}{n}(|x_1| + |x_2| + \cdots + |x_n|) \qquad (1)$$

$$x_{RMS} = \sqrt{\frac{1}{n}(x_1^2 + x_2^2 + \cdots + x_n^2)} \qquad (2)$$

The size of the training matrix was 12 x 265. This corresponds to having 12 input nodes that are signals coming from the x-acc, y-acc, z-acc coordinates, from sensor 1 and sensor 2, each having the computed ARV and RMS values (3x2x2). The length of each vector was 265 samples, this was selected by looking at the 3-5 sec signals and selecting enough samples to have reliable information from the IMU. A target set matrix (output) with 9 rows by 265 columns was constructed to train the ANN to the correct output values as it leaned. Each of the 9 rows corresponded to a particular motion while the length of the matrix mirrored the training set. Each movement's row had a 1 in their output cell if the current trail matched, otherwise, it had a 0.

The data set was divided in three batches: training, validation, and testing. During the training stage the network runs through epochs, or iterations, and attempts to minimize the error until it cannot progress further. The training set uses 70% of the total database, 20 % is used in the validation process and 10% is left for the testing stage.

The validation stage is used to test the network progress and signal when to stop the training, while the testing stage is used to measure the accuracy of the trained network. The database for the construction of the ANN consisted of 180 arm movements collected from 4 different subjects. Each subject performed each motion 5 times.

The network architecture was constructed by selecting the amount of hidden neurons to place in the hidden layer of the network. This number can be tuned, but the general rule of thumb that was followed was to select a number somewhere between the number of inputs and outputs. For this research there were 12 inputs and 9 outputs, so 10 neurons were placed in the hidden layer. The ANN was trained until the Mean Squared Error of the output vector was minimized.

Robotic Arm

The robotic arm used in this research has six degrees of freedom and uses 7 servomotors (fig. 11). Each servo has an individual signal port with the exception of the two shoulder joint servos, which were driven in tandem. The six servos are: base rotation, shoulder, elbow, writs rotation, wrist pitch and grabber.

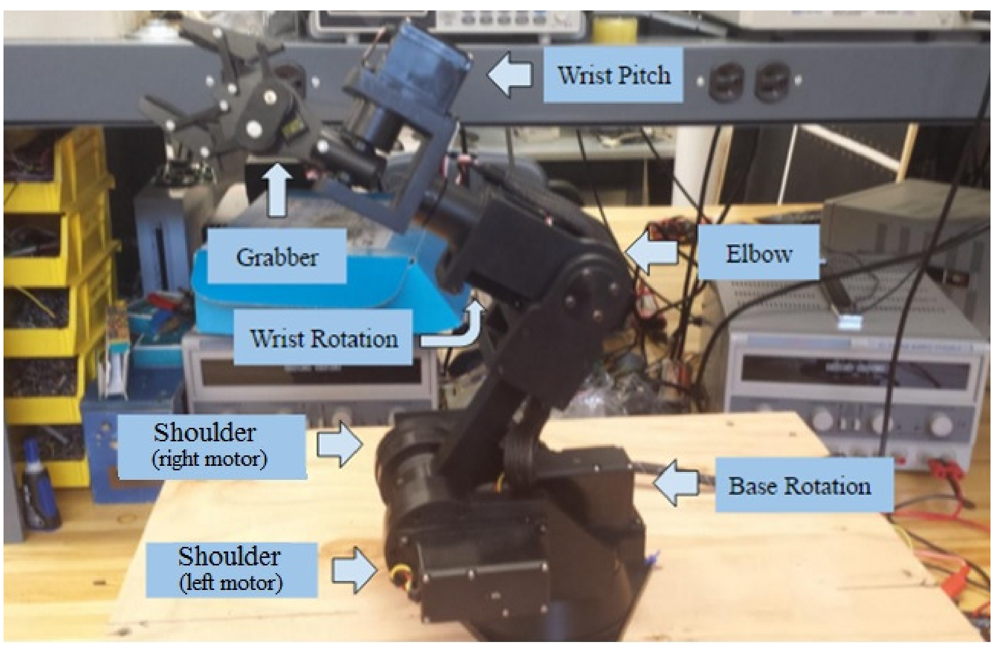

Figure 11. Six DOF Robotic Manipulator

Initially, a separate program was developed to model the desired movements using user-input angle writes. This allowed us to visually see how the arm would need to move to execute the proper movement. When the robotic arm is powered up, the arm assumes an initial position (fig. 16) and holds until further input. The robotic arm was programmed to receive input voltages (PWM signal) to each servo, corresponding to the particular motion classified by the ANN system. Then the robotic arm executes the movements in a predetermined sequence, when the commanded motion has been completed, the robotic arm pauses for about 5 seconds, and then return to the starting position. The movements were set up so that the motors were stepped through a range of angles until reaching the desired position. This was accomplished using "for" loops and time delays to make the movements smooth rather than abrupt. The implementation of the Artificial Neural Network was performed using the ANN toolbox from Matlab, once the ANN identify the specific arm movement, it would automatically transition to a program where each servo motor is assigned a control signal (PWM) accordingly. This was possible by interfacing Matlab software with an Arduino microcontroller.

Testing the Arm Extension Motion

The robotic arm was capable of mimicking the human arm motions in real time. As an example of results, consider the "Arm Extension" motion which entails bending the elbow until the arm is parallel to the floor, then extending the arm all the way forward. The user begins to bend his elbow until the arm is parallel to the floor. After the user's elbow is parallel with the floor (fig. 13), then he begins to slowly extend the arm (fig. 14), until the arm is fully extended (fig. 15). While the user performs these movements, the XYZ acceleration data is acquired and imported into Matlab which computes the Root Mean Square (RMS) value and the Averaged Rectified Value (ARV) for each coordinate. This information is then fed into the ANN. The network outputs that the robotic arm should perform the "Arm Extension" movements. Figures 16, 17, 18, and 19 show the "Arm Extension" motion performed by the robotic arm, mimicking the human arm movement. In particular for the "Arm Extension" motion, the robotic arm begins in the initial position (Fig. 16) that corresponds to a 10 degrees position on the shoulder servo motor and a 180 degrees in the elbow servo motor. Then, the robotic arm begins to bend its elbow until parallel to the floor (90 degrees to the elbow servo motor). After the robotic arm is parallel to the floor, it begin to extend itself my moving the shoulder servo approximately 55 degrees. As the arm extends the angles of the shoulder and elbow have to be adjusted so that the arm maintains a horizontal position with reference to the floor. Once that the robotic arm has performed the complete set of movements for the "Arm Extension" motion, the robotic arm returns to the initial position waiting for the next commands.

Results

To determine the accuracy of the ANN classification system, an independent volunteer was asked to perform each of the nine motions. The accuracy of the classification was 88.8% (one motion was misclassified by the ANN).

Due to mechanical limitations of the robotic arm used for this project, not all the nine motions were able to perform exactly as originally intended. The main reason for this was that the robotic arm was designed to be used in the horizontal position (fig. 19), and in order to mimic better the human arm movements, it is necessary to use a robotic arm that can be mounted in vertical position instead. Due to this limitation, there were three human arm movements that the robotic arm was not able to perform correctly: the Wipe Right, the Counterclockwise Windmill, and the Clockwise Windmill.

Figure 13. User Bends Elbow until Parallel to the Floor

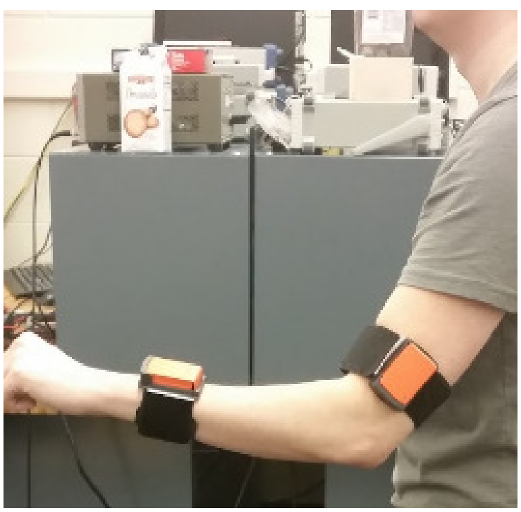

Figure 14. User Slowly Begins to Extend Arm

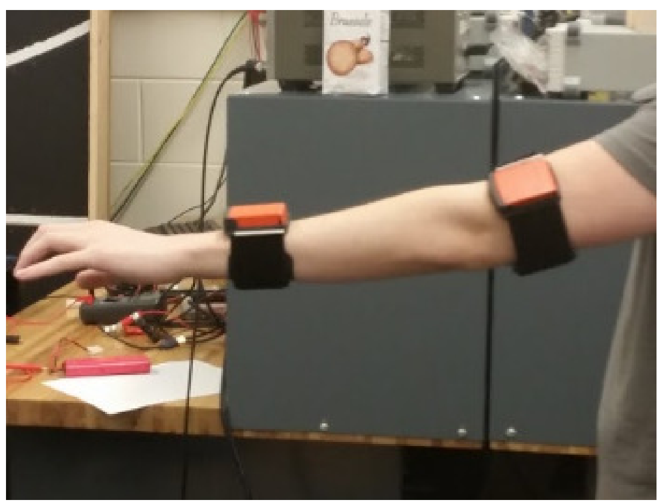

Figure 15. User Extends Arm

Figure 16. Robotic Arm in Initial Position

Figure 17. Robotic Arm Bends Elbow Until Parallel to the Floor

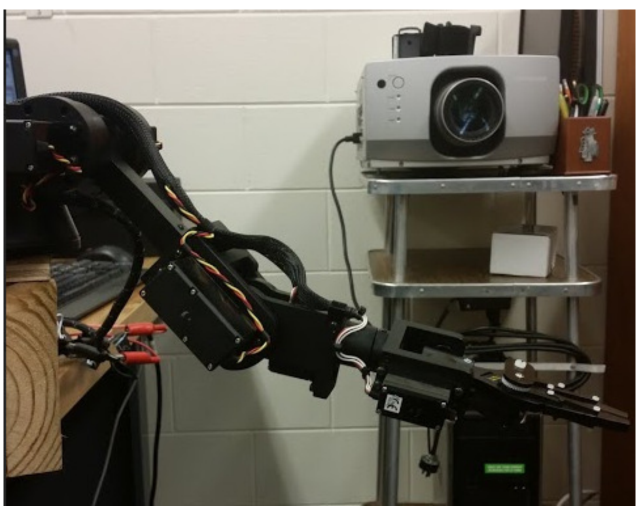

Figure 18. Robotic Arm Slowly Begins to Extend its Arm

Figure 19. Robotic Arm Fully Extended

Conclusions

The overall system was able to perform the commanded movements in real time, with a small delay of about 3 seconds, due to the signal processing time on the computer. This delay can be reduced if we can interface the Xsens Technologies' software directly with Matlab, so that the intermediate step that consists on importing the signals captured by the Xsens technology into Matlab.

The Artificial Neural Network performed very well, in the test with the independent subject, the ANN was able to identify correctly 8 of the 9 motions (88.8%). This accuracy can be improved by expanding the database that was used to train, validate, and test the ANN. Only motions from 4 subjects were used to train, validate, and test the ANN, and motions from subject 5 were used as independent motions to compute the accuracy of the system. Currently the system is unilateral, the human subject is the one that sends signals to the robot, but adding haptic feedback, the robotic arm could be able to send signals to the human subject, making a bilateral system that could expand the possible applications.

Potential applications of robotic mimicking include the manufacturing and medical industries. Manufacturing companies can use such a system such that employees can teach a robotic structure certain actions to perform without having an expert programmer. Robotic limbs can be integrated with this system for amputees and this system can help individuals in physical therapy to repair motor skills.

References

[1] Lobov, S., Mironov, V., Kastalskiy, I. and Kazantsev, V. "Combined Use of Command-Proportional Control of External Robotic Devices Based on Electromyography Signals", *Medical Technologies in Medicine*, 2015, Vol. 7 Issue 4, pp. 30-37.

[2] Hickman, S., Mirzakhani, A., Pabon, J. and Alba-Flores, R. "A Case Study on Tuning Artificial Neural Networks to Recognize Signal Patterns of Hand Motions", *Proceedings of the IEEE SoutheastCon 2015*, April 9-12, 2015, Fort Lauderdale, FL, pp. 1-4.

[3] Sekhar, R., Musalay, R., Krishnamurthy, Y. and Shreenivas, B. "Inertial Sensor Based Wireless Control of a Robotic Arm", *2012 IEEE International Conference on Emerging Signal Processing Applications*, Jan. 12-14, 2012, Las Vegas, NV, pp. 87-90.

[4] Georgi, M., Amma, C. and Schultz, T. "Recognizing Hand and Finger Gestures with IMU based Motion and EMG based Muscle Activity Sensing", *Proceedings of the 2015 International Conference on Bio-inspired Systems and Signal Processing*, Jan. 12-15, 2015, Lisbon, Portugal, pp. 99-108.

[5] Jung, P., Lim, G., Kim, S. and Kong, K. "A Wearable Gesture Recognition Device for Detecting Muscular Activities Based on Air-Pressure Sensors", *IEEE Transactions on Industrial Informatics,* Vol. 11, No. 2, April 2015, pp. 485-494.

[6] Malegam, M. and D'Silva, M. "Mimicking Robotic Hand-Arm", *2011 Annual IEEE India Conference,* Dec. 16-18, 2011, pp. 1-5.

[7] Wireless Motion Tracker, XSENS, [Online]. Retrieved April 2016 from https://www.xsens.com/products/mtw-development-kit/

[8] Ripley, B.D., *Pattern Recognition and Neural Networks*, Cambridge University Press, 1996.

[9] Stergiou, C. and Siganos, D. "Neural Networks. [Online]. Retrieved April 2016 from http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html
[10] Shiffman, D. "The Nature of Code: Chapter 10. Neural Networks". [Online]. Retrieved April 2016 from http://natureofcode.com/book/chapter-10-neural-networks/
[11] University of Toronto, Department of Psychology, "Artificial Neural Networks Technology". [Online]. Retrieved April 2016 from http://www.psych.utoronto.ca/users/reingold/courses/ai/cache/neural2.html
[12] XBee Pro, 2.4GHz, XBee 802.15.4 RF modules from Digi, [Online]. Retrieved April 2016 from http://www.digi.com/pdf/ds_xbeemultipointmodules.pdf
[13] Arduino mega Microcontroller, [Online]. Retrieved April 2016 from https://www.arduino.cc/en/Main/arduinoBoardMega

Biographies

FERNANDO RIOS is currently an Associate Professor in the Electrical Engineering Department at Georgia Southern University. His area of research includes, robotics, neural networks, fuzzy logic and embedded systems. He has over 20 years of experience as a faculty in higher education institutions. Dr. Rios may be reached at frios@georgiasouthern.edu

IMANI AUGUSMA is currently a graduate student in the Master of Science in Applied Engineering Program at Georgia Southern University. Her areas of interest include intelligent vehicles, mechatronics, and engines. Miss Augusma may be reached at ia00336@georgiasouthern.edu

ROCIO ALBA-FLORES is currently an Associate Professor in the Electrical Engineering Department at Georgia Southern University. Her areas of research includes, control systems, biomedical applications, and robotics. Dr. Alba has over 20 years of experience as a faculty in higher education institutions. Dr. Alba may be reached at ralba@georgiasouthern.edu