ISSN 2278-3091

Volume 8, No.5, September - October 2019

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse59852019.pdf

https://doi.org/10.30534/ijatcse/2019/59852019

Kinect-based Teleoperated Pseudo-Anthropomorphic Robotic Arm

Eduardo I. Cabral Jr., Christopher Lambert M. Flores, Geromin S. Nepomuceno III, Jesus Lorenzo A. Singson and Gerald P. Arada

Gokongwei College of Engineering, De La Salle University, Manila, Philippines, gerald.arada@dlsu.edu.ph

ABSTRACT

The objective of this study is to develop a pseudo-anthropomorphic robotic arm that is controlled through a motion capture device. The study aims to present an alternative method to robot control. Utilization of motion capture to control a pseudo-anthropomorphic arm can make the control intuitive and straightforward for its operator. This can contribute to robot applications wherein accurate control is necessary. This pseudo-anthropomorphic robotic arm has four degrees of freedom, specifically, shoulder yaw and pitch, elbow pitch and wrist pitch. The motion capture device used is the Microsoft Kinect which detects its operator's arm position. Data from the Kinect is fed through computer and a PIC microcontroller in order to control the motors of the robotic arm. The PIC microcontroller is responsible for generating pulse width modulation that will power the motors. Encoders are used to determine the position of the motors. A PI controller is used as the feedback controller and is implemented also through the PIC microcontroller. Testing was done in order to determine how well the robotic arm is able to follow the operator's arm position. Through this study the Kinect was shown to be a capable method to control a pseudo-anthropomorphic robotic arm with four degrees of freedom.

Key words: Microsoft Kinect, proportional integral controller, pseudo-anthropomorphic robotic arm, telerobotics.

1. INTRODUCTION

The term teleoperation is used in research, academic and technical fields to define the operation of a device or machine in a given distance. Teleoperation can also apply to robots that are not autonomous. One specific term used for robots controlled at a distance is telerobotics. There have already been various methods through which these robots are controlled such as computers, joysticks, gloves or remote controls. Cameras have also been used in order to track a robot operator's movements. This study aims to utilize a motion-sensing device in order to control a pseudo-anthropomorphic robotic arm.

The study intends to contribute to motion-capture based telerobotics. With motion-capture as the method of control, different applications of robotics may be improved. Easier and more precise control may be achieved with the use of motion capture since the operator is able to control the robot through his own movements without having to learn and depend on a different controller.

The system will utilize Microsoft Kinect as the method to detect the robot operator's movements. The Kinect has an RGB camera as well as a depth sensor [1]. It can be programmed in order to track movements of specific points on the operator's body. The data obtained from the Kinect will be used to control the position of a pseudo-anthropomorphic robotic arm [2].

2. DESIGN CONSIDERATION

2.1 Robotic Arm Description

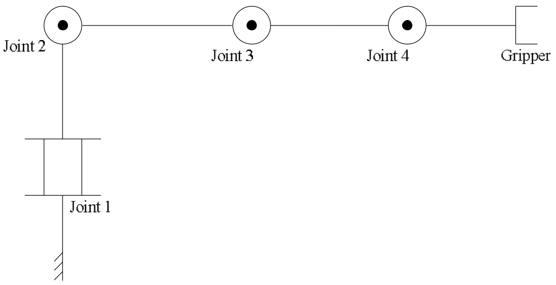


Figure 1: Diagram of the robotic Arm.

The mechanical structure of the robotic arm consists of 4 revolute joints as seen in Figure 1. Consequently, it has 4 DOF which are shoulder yaw (Joint 1), shoulder pitch (Joint 2), elbow pitch (Joint 3) and wrist pitch (Joint 4). It forms an open kinematic chain where one end of the chain is constrained to the base (Joint 1) while an end-effector (gripper) is connected to the other end allowing manipulation of objects in space. Furthermore, it has an anthropomorphic jointed arm configuration that will provide a work envelope similar to the work envelope of a human arm. It is also classified as a servo-controlled robot since the position of the gripper will be monitored by a controller operating in a close-loop mode. Lastly, the robotic arm's motion is powered by an electric drive system which is easy to control and provides an accurate control of position and speed [3] [4] [5] [6].

2.2 Kinematics of the Robotic Arm

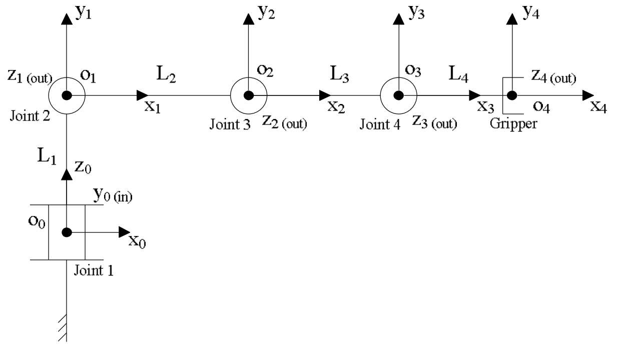


Figure 2: Diagram of the robotic arm with Denavit-Hartenberg (D-H) reference frame assignment

Table 1: D-H Parameters of the Robotic Arm

Joint i	Link Length (a_i)	Link Twist (α_i)	Joint Distance (d_i)	Joint Angle (θ_i)	Operating Range
1	0	90°	L_1	θ_1	-45° to 45°
2	L_2	0	0	θ_2	-120° to 120°
3	L_3	0	0	θ_3	0° to 160°
4	L_4	0	0	θ_4	-60° to 160°

The kinematics of the robotic arm was derived using Denavit-Hartenberg (D-H) algorithm [7] [8] [9]. The D-H reference frame assignment and the D-H parameters are depicted in Figure 2 and listed in Table I, respectively.

2.3 PIC Microcontroller

A PIC24FJ128GA010 microcontroller is used as the motor controller for this project. This microcontroller is necessary in order to implement a software based PI controller with motor control. It also includes peripherals that will be used for this implementation such as UART, SPI, timers, and PWM. The microcontroller receives data from the computer for the computed angles through its UART module. SPI is used on the microcontroller to receive data on the robotic arm's position through the encoders. Timers are used to ensure that the position data is gathered and processed by the software implemented PI controller at proper intervals. PWM or pulse width modulation, along with several general I/O pins for direction are used to control the H-bridges based on the output of the software based PI implementation. PWM duty cycle is varied with respect to the output of the PI controller. Direction is based on the sign of the controller output.

2.4 Proportional Integral (PI) Controller

A discrete proportional integral (PI) controller was implemented on the PIC microcontroller where the discrete PI controller equation would be executed. The PI controller output would then be translated into the correct PWM signals in order to correct the positions of the robotic arm's motors. There was no need for the derivative controller since the robotic arm will be controlled based on position and not speed [10].

2.5 Microsoft Kinect

The Kinect is a device jointly developed by PrimeSense and Microsoft. It has been used for many skeletal tracking purposes due to its capabilities. It not only has a regular camera, but it also has an infrared projector and an infrared camera which work regardless of the lighting conditions. Thus, it is not prone to error due to variations in lighting. The infrared projector and camera provide the ability to map out the objects in front of the Kinect in terms of depth. On the software side, the processing language is used. Processing is a multi-platform open source programming language and environment. It is one of the main languages in use for Kinect programming. A wide amount of support and online content is available for Kinect programming with processing [11]. Processing is widely used for Kinect programming is because of the Simple OpenNI library that is freely available for use. Simple OpenNI is an OpenNI and NITE wrapper for processing [12]. Another library that was tested and used was the Kinect for processing library. It is a java wrapper for the Kinect for Windows SDK provided by Microsoft. However, the Kinect for Windows SDK allows for tracking of the wrist joint, whereas the Simple OpenNI library does not [13]. Aside from the Kinect related libraries, another library is also used to aid in detecting open or closed hand states. The finger tracker library provided for processing is used. It has built in functions that make use of the depth image provided by the Kinect to create a contour map. A contour map creates traces around objects. Given the contour map, inflections are detected and assumed to be finger points [14].

2.6 Mechanical Design

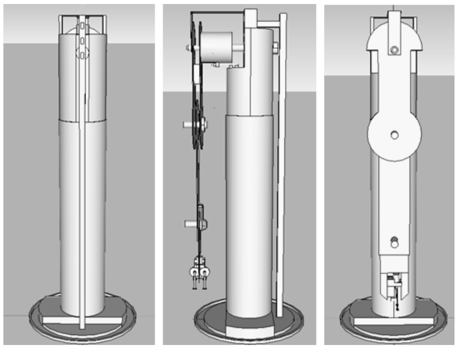


Figure 3: Computer model of the robotic arm.

The mechanical design of the robotic arm has six parts which are the base, shoulder, upper arm, lower arm, wrist and gripper. Since the robotic arm is a pseudo-anthropomorphic robotic arm, its design does not necessarily need to mimic the structure of an actual human arm. The design also considered the angle restrictions of each joint, the motor position for each movement, and the distributed weight of each part. The complete design of the robotic arm can be seen in Figure 3.

2.6 Operation of the Robotic Arm

Referring to Figure 4, the operation of the robotic arm begins with the detection of the operator in front of the Kinect. The infrared projector and infrared camera of the Kinect are used in order to detect the operator's joints. The detected xyz coordinates are translated to use coordinates of the operator's right hip as the origin. The computer makes use of the translated xyz coordinates of each of the joints together with inverse kinematics to compute for the angles of each of the joints. For the gripper, depth data around the operator's hand and detection of inflections are used to determine whether the operator's hand is open or closed. Upon obtaining both the angles for each of the joints and the state of the operator's hand, the data is transferred to the PIC24 microcontroller through UART. The PIC24 microcontroller is responsible for carrying out discrete proportional integral control, pulse width modulation, control of motor direction, and reading of the absolute magnetic encoders for feedback. Four absolute magnetic encoders are connected to each of the shafts of the motors being used at the joints. These absolute magnetic encoders can send twelve bits of binary data that correspond to specific positions of the motor shaft at the joint.

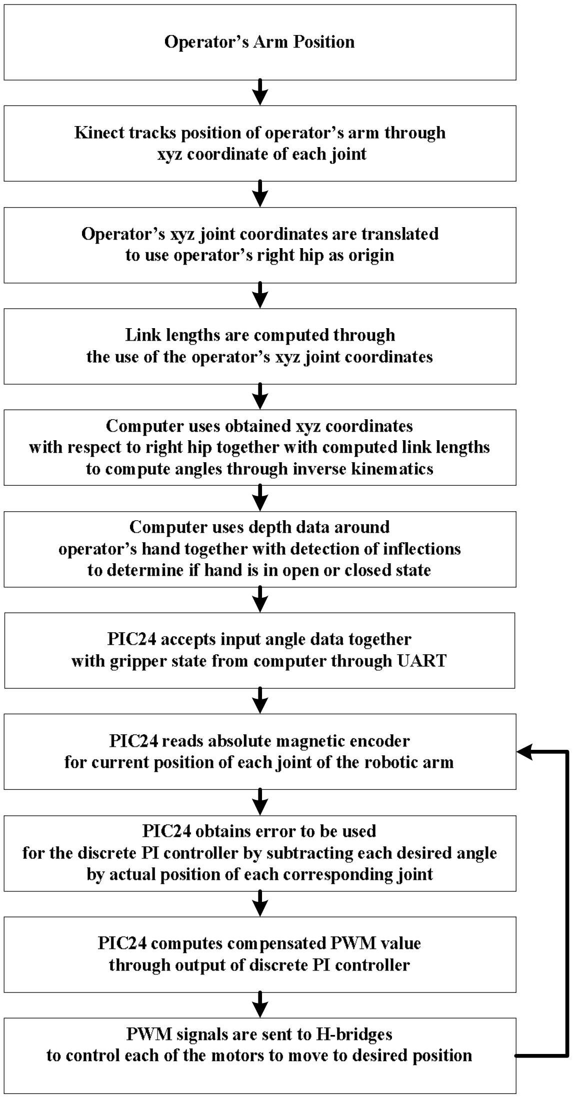

Operator's Arm Position
Kinect tracks position of operator's arm through xyz coordinate of each joint
Operator's xyz joint coordinates are translated to use operator's right hip as origin
Link lengths are computed through the use of the operator's xyz joint coordinates
Computer uses obtained xyz coordinates with respect to right hip together with computed link lengths to compute angles through inverse kinematics
Computer uses depth data around operator's hand together with detection of inflections to determine if hand is in open or closed state
PIC24 accepts input angle data together with gripper state from computer through UART
PIC24 reads absolute magnetic encoder for current position of each joint of the robotic arm
PIC24 obtains error to be used for the discrete PI controller by subtracting each desired angle by actual position of each corresponding joint
PIC24 computes compensated PWM value through output of discrete PI controller
PWM signals are sent to H-bridges to control each of the motors to move to desired position

Figure 4: Flowchart of the robotic arm's operation.

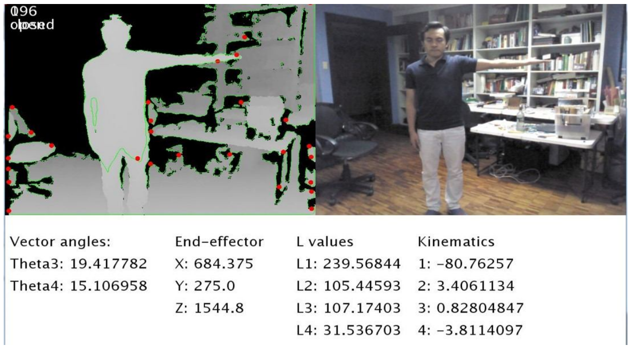
With the PI controller, to perform the proportional part, the error, which is the difference of the desired angle of the operator and the actual angle measured by the absolute magnetic encoders, is multiplied to a proportional constant. To perform integration, the sum of the current and previous error is taken and multiplied to the integral constant. The sum of the proportional and integral parts is obtained in order to output a value that will determine the duty cycle for pulse width modulation. Pulse width modulation is used to control the speed of the motors. The sign of the controller output on the other hand, whether negative or positive, determines the motor direction. The PIC24 outputs a PWM signal together with high or low outputs from general I/O pins which go into the H-bridges of each motor. The output of the general I/O pins dictates the direction. The motor then moves faster or slower depending on the PWM signal whose pulse width increases when there is larger error and decreases when the error is decreasing as well.

3. DATA AND RESULTS

3.1 Kinematics on Kinect

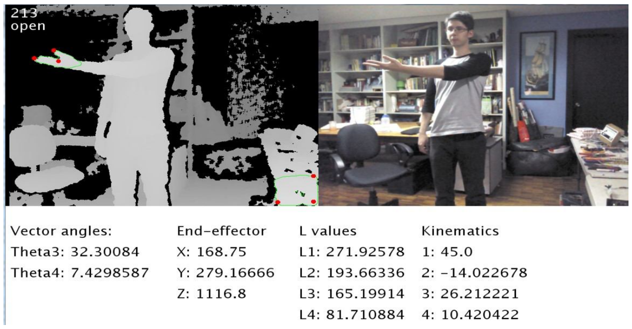
The ability of the Kinect to obtain the correct joint angles was tested. The ability to detect an open and closed hand was also tested. The testing was done by comparing actual angles on the operator and the obtained angles from the Kinect through kinematic equations. Two operators were used with different heights. The short operator has a height of 1.651 meters while the tall operator has a height of 1.8796 meters (See Figures 5 and 6). Two operators were used in order to determine how well the Kinect is able to obtain the angles despite changes in the arm length of the operator. The tracked joints used are the right hip, right shoulder, right elbow, right wrist, and right hand. The right hip is used as the base point and is treated as the origin of all coordinates. From the testing done with the gripper, it can be seen that the Kinect was able to differentiate an open palm from a closed one. This is true for both the short operator and tall operator (See Figures 7 and 8).

Thus, coordinates of all points are translated to give x, y, and z values based on the operator's right hip position. The axes of each point are also translated to ensure that they follow the same axes used for the kinematic equations. For the short operator trials, it can be seen under the Kinematics column that the Kinect is able to determine the angles for each of the joints on the operator being accurate within a few degrees. For the tall operator trials, the same trend can be seen wherein most of the error is within 10 degrees. It can be said that the difference in operator does not affect the ability of the Kinect to measure the angles. Relatively, the same amount of error can be seen for both operators. There are instances wherein the error became large for the taller operator but other trials have shown that it is not always the case since it is able to measure the other angles more accurately (See Figures 9 and 10).



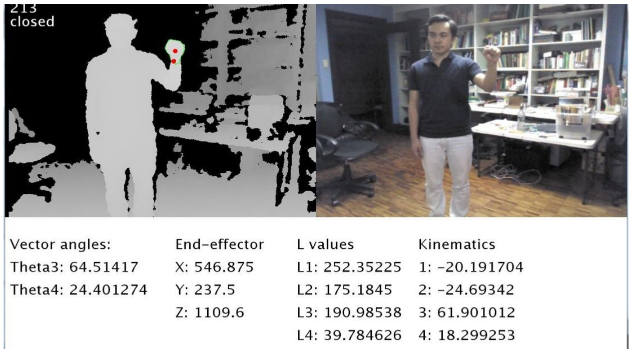
```
Vector angles:        End-effector   L values        Kinematics
Theta3: 19.417782     X: 684.375     L1: 239.56844   1: -80.76257
Theta4: 15.106958     Y: 275.0       L2: 105.44593   2: 3.4061134
                      Z: 1544.8      L3: 107.17403   3: 0.82804847
                                     L4: 31.536703   4: -3.8114097
```

Figure 5: Kinect Kinematics for Short Operator



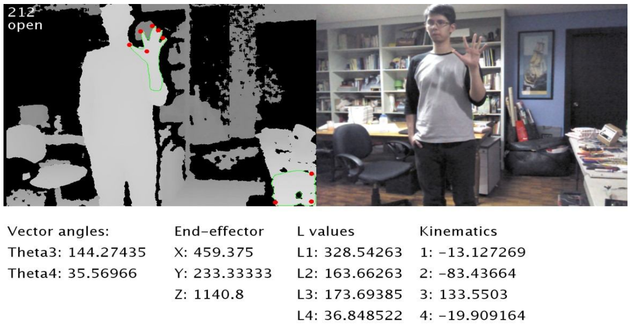
```
Vector angles:        End-effector   L values        Kinematics
Theta3: 32.30084      X: 168.75      L1: 271.92578   1: 45.0
Theta4: 7.4298587     Y: 279.16666   L2: 193.66336   2: -14.022678
                      Z: 1116.8      L3: 165.19914   3: 26.212221
                                     L4: 81.710884   4: 10.420422
```

Figure 6: Kinect Kinematics for Tall Operator



```
Vector angles:        End-effector   L values        Kinematics
Theta3: 64.51417      X: 546.875     L1: 252.35225   1: -20.191704
Theta4: 24.401274     Y: 237.5       L2: 175.1845    2: -24.69342
                      Z: 1109.6      L3: 190.98538   3: 61.901012
                                     L4: 39.784626   4: 18.299253
```

Figure 7: Kinect Closed Gripper



```
Vector angles:        End-effector   L values        Kinematics
Theta3: 144.27435     X: 459.375     L1: 328.54263   1: -13.127269
Theta4: 35.56966      Y: 233.33333   L2: 163.66263   2: -83.43664
                      Z: 1140.8      L3: 173.69385   3: 133.5503
                                     L4: 36.848522   4: -19.909164
```

Figure 8: Kinect Open Gripper

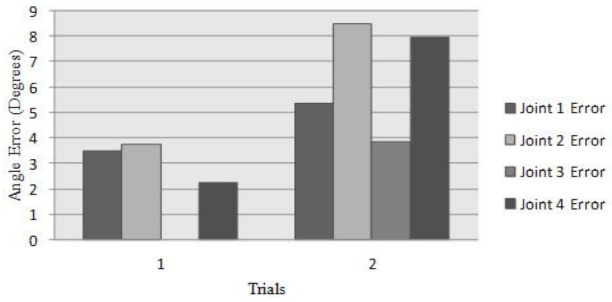


Figure 9: Short Operator's Arm and Kinect Measurement Comparison for Joints 1, 2, 3, and 4

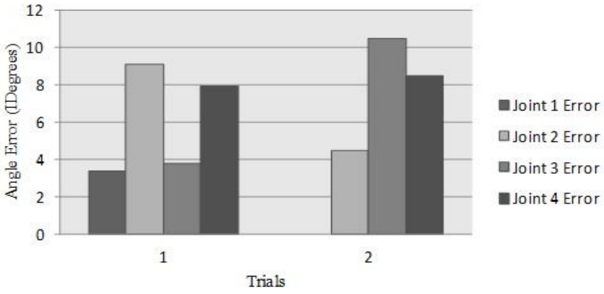


Figure 10: Tall Operator's Arm and Kinect Measurement Comparison for Joints 1, 2, 3, and 4

3.2 Operator's Arm Position vs. Robotic Arm Position

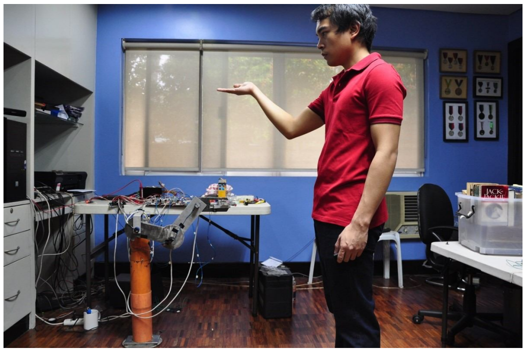


Figure 11: Short Operator's Arm and Robotic Arm Comparison

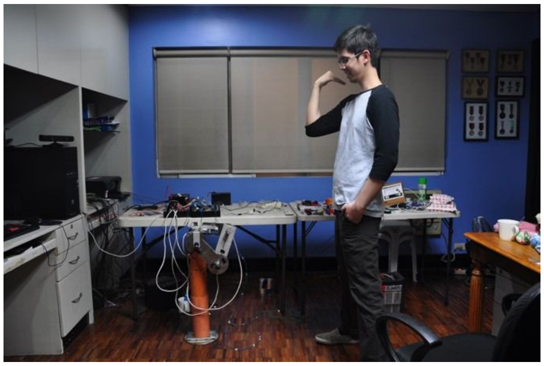


Figure 12: Tall Operator's Arm and Robotic Arm Comparison

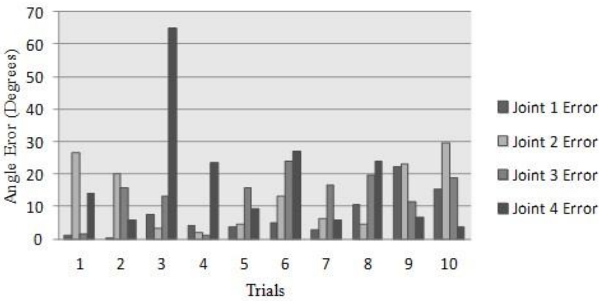


Figure 13: Short Operator's Arm Position vs. Robotic Arm Position Comparison

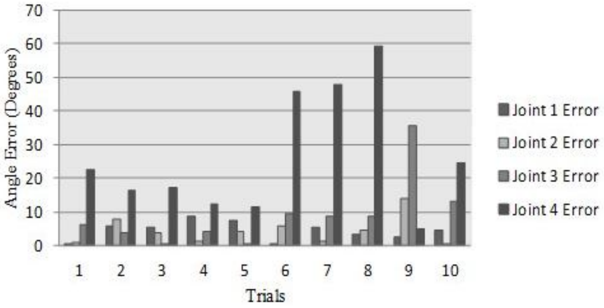


Figure 14: Tall Operator's Arm Position vs. Robotic Arm Position Comparison

The comparison between the position of the operator's arm and the robotic arm is shown in Figure 11 and 12. Again, two operators were used, one being taller than the other. From the figures it can be seen that the robotic arm is able to follow both of the operators' movements. It can be said that all four joints are able to obtain error lower than 10 degrees for both the short operator and the tall operator as seen on Fig 13 and 14. The changes in operator arm length do not vary the results of the testing drastically. Joint 4, however, can be seen to be inaccurate for both the short and tall operator. It is prone to large error and it is not also as precise as the other joints. Differences in operator's arm and robotic arm angles may be due to the previously mentioned problems encountered with noise and encoders together with the differences observed between operator angles and the computed angles based on the kinematics equations from the Kinect data. Joint 4 is seen to be the least accurate since the Kinect is not always able to track it as well as the other joints. The kinematic equations make use of wrist and gripper points for computation of Joint 4 angles. The Kinect together with the library in use, Kinect for Processing, is able to detect both wrist and gripper points. However, there are certain instances wherein it struggles to accurately obtain the points. Thus, this difficulty in differentiating or proper detection of wrist and gripper points may lead to the computation of incorrect angles through the use of the kinematic equations.

3.3 Operator's Arm Position vs. Robotic Arm Position using Encoder Readings

Figures 15 and 16 show the error between the operator's arm position and the robotic arm position based on the absolute magnetic encoder readings. Again, a short operator and a tall operator were used in order to determine whether the arm length difference would affect the accuracy of the robotic arm. All four joints were tested for three different angles. Each angle had three trials. Figure 15 shows one of the trials for one of the joints for the short operator. Fig 16 meanwhile is for one of the trials for the tall operator.

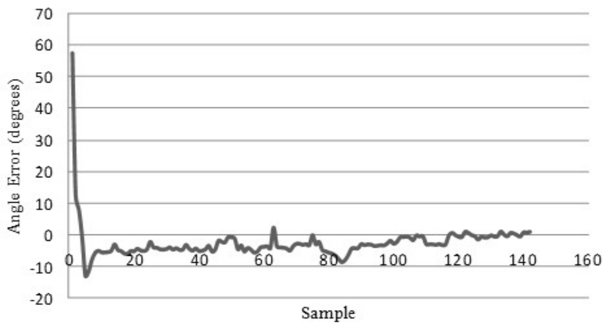


Figure 15: Angle error for short operator.

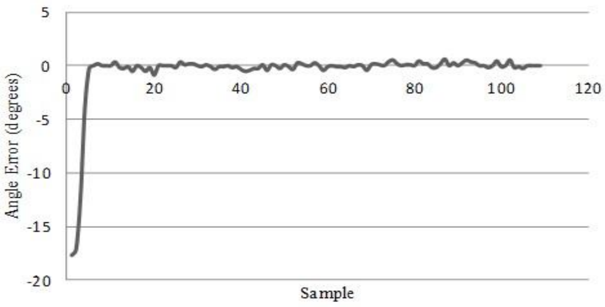


Figure 16: Angle error for tall operator.

4. CONCLUSION

An alternative method to control a robotic arm was done through this study. A pseudo-anthropomorphic robotic arm with four degrees of freedom was controlled through the use of the Microsoft Kinect. The kinematics used was able to output the correct values for all joints. The Kinect was able to detect the angles of the shoulder yaw, shoulder pitch, elbow pitch and wrist pitch of the operator The robotic arm was able to follow most positions of the operator's arm. There are instances where the position of the operator was not followed as well but this may be due to errors in detection of the Kinect, and absolute encoder angle detection errors due to noise. Overall, the robotic arm was able to follow the operator's arm position for all four degrees of freedom.

REFERENCES

1. "**Kinect for Windows Sensor Components and Specifications**." MSDN. Microsoft, n.d. Web. 1 April 2018.
2. A. Zoss, H. Kazerooni and A. Chu. **On the mechanical design of the Berkeley Lower Extremity Exoskeleton**, IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, pp. 3465-3472, 2005. https://doi.org/10.1109/IROS.2005.1545453
3. R. Manseur. A. ***Robot modeling and kinematics***. Boston, Ma: Charles River Media, 2006.
4. M. Zhihong. ***Robotics, 2nd ed.*** Jurong, SG: Prentice Hall, 2005.
5. L. Sciavicco & B. Siciliano. ***Modeling and control of robot manipulators***. Naples, IT: The McGraw-Hill Companies, Inc, 1996.
6. K. H. Low. ***Robotics: Principles and systems modeling.*** Jurong, SG: Prentice Hall, 2007.
7. V. Desphande & A. Verma. **End-effector position analysis of SCORBOT-ER Vplus robot**. International Journal of Advance Science and Technology, vol.29, April 2011.
8. Introduction to Robotics [Video file]. Retrieved from http://www.youtube.com/watch?v=0yD3uBshJB0
9. E. R. Magsino & J. R. Del Rosario. **Offline and Simplified Industrial Automation Processes Employing Robotic Manipulators.** Applied Mechanics and Materials, 446-447, 2013. https://doi.org/10.4028/www.scientific.net/AMM.446-447.1250
10. Z. Vukic Z. & O. Kuljaca. **Lectures on PID conrollers.** April 2018**.** Retrieved from https://pdfs. semanticscholar. org/55f8/145da00f16774e264f1b4d3f3b4d0caf21f5.pdf
11. "**Overview. A short introduction to the Processing software and projects from the community**." Processing. Media Temple, n.d. web. 1 April 2018.
12. "**Simple-openNI**." Google Project Hosting, n.d. Web. 1 April 2018.
13. **Magic & Love Interactive**. Retrieved 1 April 2018 from http://www.magicandlove.com/blog/research/ kinect-for-processing-library/
14. **Makematics**. Retrieved 2 April 2018 from http://makematics.com/code/FingerTracker/