# Robotic Arm Control through Mimicking of Miniature Robotic arm

**Rajesh Kannan Megalingam,**
*Dept. of ECE, Amrita School of Engg., Amritapuri*
*Amrita Vishwa Vidyapeetham, Amrita University, India*
*rajeshkannan@ieee.org,*

**Sricharan Boddupalli,**
*Dept. of ECE, Amrita School of Engg., Amritapuri*
*Amrita Vishwa Vidyapeetham, Amrita University, India*
*b.sricharan14116@gmail.com,*

**K G S Apuroop**
*Dept. of ECE, Amrita School of Engg., Amritapuri*
*Amrita Vishwa Vidyapeetham, Amrita University, India*
*apuroopkgs@gmail.com*

***Abstract*— In the recent technology advancements, employing robots for handling precise operations were seen in abundance like industrial, medical and rescue operations etc. Introducing human interaction with the robotic arm in the real-time scenario opens new doors in innovations and applications of the robotic arms. This paper reports the making and testing of a miniature prototype robotic controller arm using Bluetooth wireless technology which can be used to manually control an actual robotic arm that has 3 degrees of freedom. The main objective is to design and employ a miniature version of robotic arm which can make the main robotic arm imitate as it does when the miniature arm is operated by a human operator. This can be achieved with the help of simple potentiometers at the joints of the miniature arm. The data is exchanged by the miniature robotic arm and the main robotic arm with the help of Bluetooth wireless interface.**

***Index Terms*—Miniature robotic arm, Potentiometers, Servo motors, Bluetooth.** *(key words)*

## I.INTRODUCTION

A robot is an intelligent embedded machine which is employed to carry out the tasks that are monotonous or that are unsafe for human intervention. They fall under certain categories based on the nature of their controlling such as fully automatic, semi-automatic and manually controlled. They are built upon hardware which is interfaced with software that provides the significant intelligence to direct its mechanical movements for the accomplishment of a task. Automating robots is a perplexing task, yet it became easier over the years. A robotic arm is a programmable mechanical arm which can execute the functions of a human arm. The terminus of the robotic arm is called the 'end-effector'. The end effector can perform various tasks such as pick and place, welding, drilling, painting, watering etc. Robots can be operated remotely which helps the end user in manipulating the remote objects using the robotic arms, mainly in hostile environments. A robotic arm is a mechanical device which is interfaced through electronics to carry out precise operations. It is designed to execute to undertake many industrial and medical operations.

This research work gives an insight about the idea of making a miniature arm which can control a robotic arm of certain degrees of freedom. This technology can be used for various types of applications. For example, the miniature arm can be made a wearable one which can be worn by a therapist and the main arm can be an exoskeleton robotic arm which can be worn by a patient. Thus, this combination can be used in rehabilitation of stroke patients and other disabled people. This technology can also be used for operating manual pick and place robots. In a scenario where the pick and place robots needs to handle some identified hazardous substances, cannot be left without a human supervision. For controlling it manually there must be a device like the miniature arm which can control the robotic arm. It is a cost-effective method for manufacturing manual robotic arm controllers. This setup needs very less components for its implementation. This idea can be extended to measure other sensory feedback like vibrations from the robotic arm when it is involved in some operation. This idea with measuring necessary feedback parameters can be used for remote controlled medical operations such as surgeries.

While using the miniature arm, the only crucial parameter we need to read is the angle made by the joint. This angle can be read easily with the help of a potentiometer and any microcontroller based embedded system. This data is transmitted to the robotic arm using the Bluetooth Wireless Technology for short range operation of the miniature arm and robotic arm combination. In addition, control of the robotic arm can also be via Wi-Fi or through the Internet. Here, the data from the miniature arm is fed to the Bluetooth module for operating the servo motors in the robotic arm. Similarly, the feedback from the main arm is sent to the miniature arm utilizing the same wireless technology. This feedback from the robotic arm can be analyzed and processed for stability and the precision movements of the main arm. Presently, we are dealing with the angle feedback from the arm.

## II.MOTIVATION

The robotic arms are involved in many operations in various fields like medical, disaster management, industrial applications etc. Controlling the robotic arm using switches and joysticks is a tough task and it needs

978-1-5090-4559-4/17/$31.00©2017IEEE

good experience for the individual operating the mechanism. But to make things simple and automated, the miniature arm is employed. This miniature arm helps in establishing human machine interface with the robotic arm where it can be controlled manually. There are various places where the movement of humans are completely restricted including nuclear operations, specific areas in industries, rescue operations etc. In such scenarios, a manually controlled miniature arm which can control the robotic arm remotely will be extremely useful. With the vast advancements in medical technology, the robotic arms are employed in the medical field as well. The robotic arms are well programmed to undertake critical surgeries. There are scenarios where a surgeon needs to operate the robotic arm remotely. Then the miniature arm can be used to control the movements of the robotic arm. Usually a robotic arm is controlled using a terminal in the computer. Exact co-ordinates need to be mentioned in the terminal to make the robotic arm move which is a herculean task for the user. Such miniature arms will resolve the issues involving the control of robotic arms. The motivation behind this research work is to remotely control a robotic arm using a manually operated miniature arm.

## III. Related Works

The authors in the paper [1] states that a hand wearable can be used as an interface to control the robotic arm in real-time. This method may not completely address the problem of operating the robotic arm, as the user needs to put his arm in the same posture for a long time until the task is completed by the robotic arm. The miniature arm discussed in this paper will address the problem as the user simply need to position the miniature arm in the desired way that he/she might want the robotic arm to be in. An alternative method to control the robotic arm using a miniature arm is discussed in the paper [2] by using the accelerometers. The basic principle here is, the components of the acceleration parallel to the assumed axes (X, Y, Z) have relation among themselves with the total magnitude. Certain parameters like angles, velocity can be obtained accurately (to decimal values). Accelerometers are not cost effective solutions. Another way of controlling the robotic arm is to use joysticks which is discussed by the authors in the paper [3]. The advantage with this type of control is that we can control the arm as a whole with the help of one joystick itself. Simultaneous control of all the DOF of the robotic arm at a time might be a rigorous task and needs a lot of patience and judgement.

In paper [4], authors weigh importance to program the robotic arm, with latest software, that needs to execute a monotonous task for a considerable time period. In such method, the operator needs to have keen knowledge about programming and also the debugging of the program is quite tough and a time-consuming process. Authors in the paper [5] state that the virtual reality environment helps the operator to understand the situation of the remotely located robotic arm which in-turn helps in executing a particular task with precision. The present model uses the TCP/IP protocol for the data transmission. Implementation is not cost effective. Visual feedback is another method for implementing a remote connection with the robotic arm and is discussed by the authors in paper [6]. The method uses the MATLAB for programming which is a bit complex. The programming can be achieved using a simple Arduino sketch. In paper [7] the authors implement the idea using the inertial motion capture technique which is a very complex method. It uses accelerometers as in the paper [2] which turns out to be an expensive method. In one of our earlier works [8] we discuss about a robotic arm for a coconut tree climber robot.

## IV. System Architecture

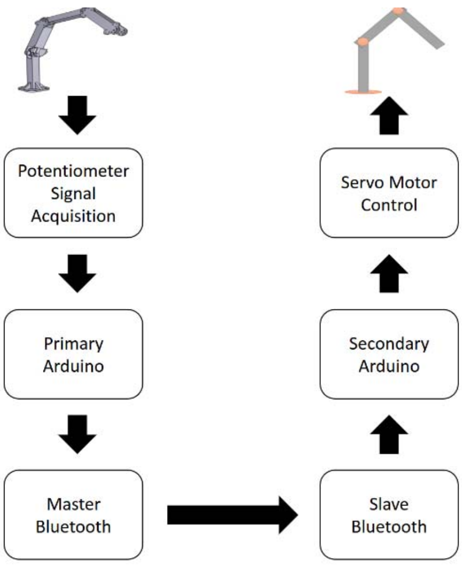


Figure 1. Flowchart showing the path of the data from its origin to end

### *Potentiometer Signal Acquisition*

As mentioned earlier, the angle made by the joint is the only parameter which we have to identify and analyze. In order to register this data, the easiest way is to use a potentiometer at that joint. A potentiometer is basically a practical voltage divider used for measuring the voltage. This voltage level can be read from the potentiometer as a function of the internal resistance times the total voltage applied on the potentiometer. When a rotary potentiometer is used, the internal resistance is a function of the angle turned by the knob from the pre-set zero condition. This is internal resistance can be used to find out the approximate angle turned by the knob. This knob is eventually connected to work with the joint of the exoskeleton.

### *Initial Data Processing using Arduino*

Arduino is a microcontroller based development board provided with set of digital and analog input/output pins that can be interfaced with the various expansion boards and other circuits. The system features serial communication interfaces. The programs can be uploaded into the board via Universal Serial Bus port. The Arduino project provides the Arduino integrated development environment, is a cross-platform application written in programming language Java. A program written with the help of Arduino IDE is called a "Sketch".

The data acquired from the potentiometers is now further carried over to an embedded system, Arduino, for processing. The data is acquired from the analog pins of the Arduino board. As the miniature arm is made to control the robotic arm of three degrees of freedom, the signals will be taken from the three potentiometers and are processed. Let us consider one potentiometer as an example. When a potentiometer is connected to the joint, according to the movement in the joint, the value of the resistance of the potentiometer changes according to the angle. When this internal gain is found out, this value is rendered by the embedded system (here, it is the Arduino). In the Arduino, this value is divided into 1024 levels of voltage (with 0 being the lowest and 1023, the highest). This is called "Mapping". The Arduino maps the values from 0 V - 5V to 0 - 1023. These values can, in turn, be mapped from 0 to 180. These values can be used as the angle turned by the knob from the zero level. This is for one potentiometer reading. With the help of Arduino programming, we can compile all the potentiometer readings (the angles) in a string of data. This data is eventually sent to the slave Bluetooth module (for transmission).

### *Transmitter and Receiver (Master-Slave Control)*

The processed data from the primary Arduino board is to be transmitted to the robotic arm via a master slave Bluetooth setup. In this model the master transmits the processed potentiometer signals to the slave device connected to the robotic arm. The slave module receives the transmitted data and sends it to the secondary Arduino board for further analysis. The board processes the data and the slave's servo motors are driven using the digital pins of the Arduino board. Here, the Arduino program is written to fetch the string data from the master Bluetooth module. This string is then sent to the slave module serially. The transmission rate is fast, and it can be employed in transmitting and receiving data from multiple potentiometers and servo motors respectively. The bit transfer rate through which the data is exchanged is called the Baud rate. The default baud rate of a typical Bluetooth module is 38400. This basically means that 38400 characters per second. This speed is well enough for implementing a master slave pair with negligible delay in data transmission. As mentioned earlier, the data from the potentiometers are stacked up on a string and passed on to the Bluetooth module for transmission. This string has a maximum length of 12 characters. Hence, the rate of transmission comes to 3200 packets of data per second. This value is well enough for implementing a real-time data transfer module pair.

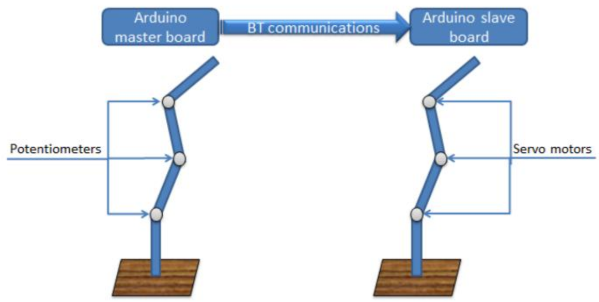


Figure 2. Basic illustration of the communication between the master and slave parts.

### *Final Data Processing and PWM Generation using Arduino*

When the serial data is received from the slave Bluetooth module, the Arduino decodes the data and assigns the angle values for respective joint motor. When these variables are fixed, the Arduino generates PWM signals for the servo motors to be operated properly. There are different PWM signals for different servo motors for different angles. This PWM signal can be determined using the Analog signal to discrete time interval convertor. This convertor basically encodes the analog value (the angle) into the time interval between two consecutive pulses. This helps the Servo motor to go to a specific angle (determined by the PWM) and stay there till the position is altered.

### *Servo Motor Control*

The servo motor takes the digital inputs from the Arduino board. The gyration of the servo motor is calibrated according to the position of the potentiometer. A Servo motor uses its own position feedback to control its motion and desired position. This is basically achieved with the help of (feedback) encoders. Without these, we need to calibrate the motors every time when we want to operate them. The encoders help is in changing the speed of the servo motor too. The feedback from the motor is input to the encoders which then creates an error if the present position is not the same as that of the desired position. This error help in the direction in which the motor should rotate in order to reach the final position. When the desired position is met, the encoder generates a zero error which stops the servo motor. This is also known as Bang-bang control of the motor.

## IV. IMPLEMENTATION

The miniature arm model shown in Figure 4 was developed with the help of a 3D printer. At each joint, the casing of the potentiometer is secured within the groove and the shaft is inserted and was secured by a potentiometer shaft coupler. The potentiometers are calibrated to an initial zero angle position and the 180-degree angle position. Hence, this helps the potentiometer to give out only a range of values which were eventually mapped from 0 to 180.

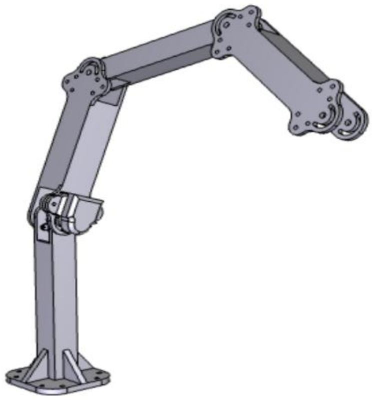

Figure 3. Model of the Miniature Arm

The data is later fetched and then fed to the Arduino for processing the raw data. The processed data is finally fed to the master Bluetooth which is fixed on the miniature robotic arm. The data is comprised of all the three potentiometers values which are in turn needed to run the servo motors of the robotic arm. The transmitted data is received by the slave Bluetooth interfaced to the robotic arm. The secondary Arduino connected on the robotic arm will be receiving the data from the slave Bluetooth and the data is decoded, processed and the robotic arm is made to function.

The robotic arm used for this is almost the same design as the miniature arm shown here. The miniature arm was designed using SolidWorks. This miniature arm is designed in such a way that it can be extended to control three or more degrees of freedom. The base of the robotic arm is robust and can stabilize the miniature arm on all regular surfaces. The robotic arm is fixed on an even base. Multi and single stranded wires are used for connecting the components. The accuracy of the arm rotated is measured using the Arduino Serial monitor.

Figure 4. 3D Printed Model of the Miniature Arm

TABLE I. . DIMENSIONS OF THE MINIATURE ARM BODY.

| | Length (in mm) | Breadth (in mm) | Height (in mm) |
|---|---|---|---|
| Support Arm Body | 30 | 30 | 155 |
| Support Arm Base | 60 | 60 | 5 |
| Arm 1 | 23 | 23 | 150 |
| Arm 2 | 20 | 20 | 150 |
| Arm 3 | 26 | 26 | 150 |

Table 1 is about the dimensions of the parts of the miniature arm as shown in Figures 3 and 4. The arm parts are given their names based on how close they are to the support arm. The length and breadth of the arm were given the same dimensional values in order to attain structural stability.

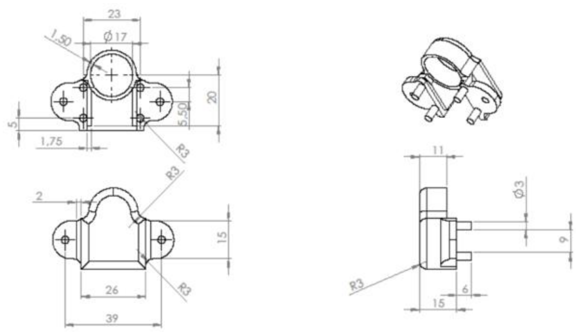

Figure 5. Potentiometer Body housing (in mm)

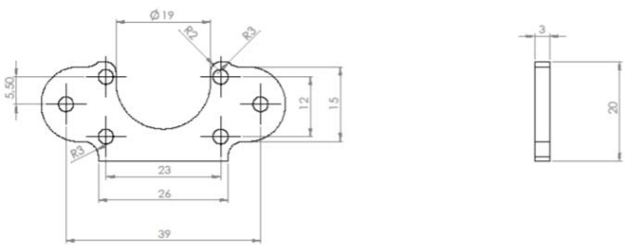

Figure 6. Potentiometer lower support (in mm)

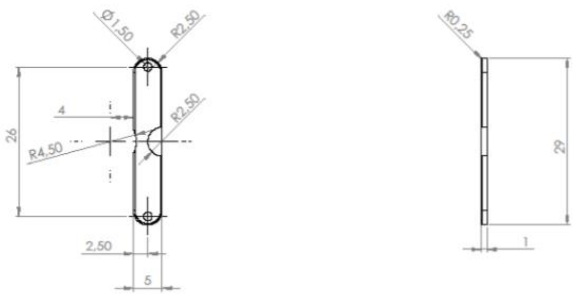

Figure 7. Potentiometer shaft coupling clip (in mm)

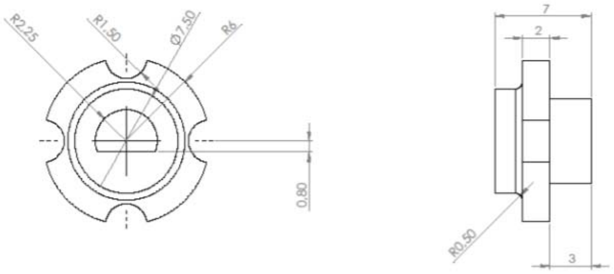


Figure 8. Potentiometer shaft coupling (in mm)

In the figures 5,6,7,8 the dimensions of the potentiometer body housing, lower support, shaft coupling clip and shaft coupling were depicted. These were made for housing the potentiometers along with fixing them in a particular position. All the measurements are in millimeters.

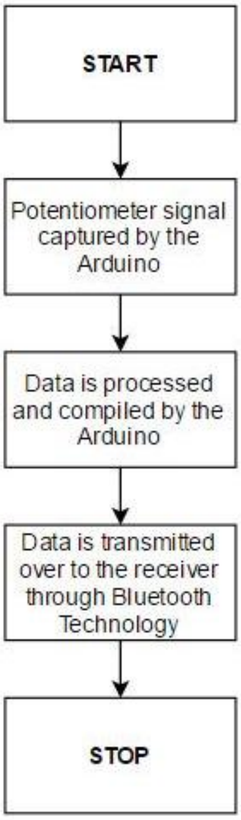


Figure 9. Flowchart showing the working of the miniature arm.

The data is taken from the potentiometers and then fed to Arduino interfaced on the miniature robotic arm. Post processing the data is fed to the master Bluetooth module for the transmission. The data is fetched from the three potentiometers and is transmitted. The baud rate is 9600 bits per second. This actually makes possible to send the three values simultaneously without negligible latency.

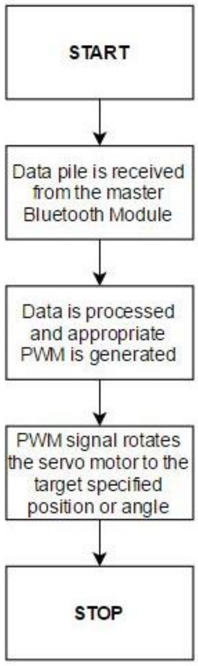


Figure 10. Flowchart showing the working on the robotic arm side.

The data string from the Miniature Arm is first passed on to the secondary Arduino where the stack is decomposed to individual angle values and those values are processed accordingly for the respective Servo motors which have to be controlled to achieve a certain target. This process of controlling the servo motors actually happens with the help of PWM signals. The values are given as PWM signals to the Servo and the motor rotates to the specified angle. This can be monitored by the Serial communication port monitor of the Arduino.

Figure 11. The 3 DOF robotic arm used for this experiment.

## V.EXPERIMENTS AND RESULTS

The values we take are from the Two Arduino boards via their Serial port. Connecting the Board to a Computer and the values can be seen through the Serial monitor in the Arduino IDE. The Values are initialized to be integers by default. In the Program, we can assign various types for the values, but we are taking in the integer values only. The reason behind this is that the Servo Motor always gives out an integer number feedback.

TABLE II. . SOME VALUES CONSIDERED FOR KNOWING THE ERROR. THE VALUES ARE TAKEN AT RANDOM AND THE ERROR IS FOUND BY SUBTRACTING THE OUTPUT VALUE FROM THE INPUT

| S No | Input via miniature arm (in degrees) | | | Output via Robotic arm (in degrees) | | | Error (in degrees) | | |
|---|---|---|---|---|---|---|---|---|---|
| | Pot. 1 | Pot 2 | Pot 3 | Servo 1 | Servo 2 | Servo 3 | | | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 30 | 30 | 30 | 31 | 32 | 30 | 1 | 2 | 0 |
| 3 | 60 | 60 | 60 | 59 | 60 | 59 | -1 | 0 | -1 |
| 4 | 90 | 90 | 90 | 90 | 91 | 91 | 0 | 1 | 1 |
| 5 | 135 | 135 | 135 | 132 | 134 | 133 | -3 | -1 | -2 |
| 6 | 180 | 180 | 180 | 180 | 180 | 180 | 0 | 0 | 0 |

These values were obtained with the help of the serial communication port in the Arduino. These serial ports can be connected to a computer and can be monitored any time. The angle at the miniature arm side is set to a specific angle which is constantly monitored from the serial monitor and the feedback can be measured with the help of the secondary Arduino. These values can be considered for various analysis purposes.

The error was calculated manually for selected values only. Since both the values were integer type, the error was obtained to be an integer only. The error here was the difference of the potentiometer value from the servo motor feedback value. Then the mean error can be calculated for these values. Considering the data in the table 2, we get a mean error of half a degree lag at the first joint. The second joint and the third joint had a relatively less error with one third of a degree and a zero respectively.

TABLE III. . THE MEASUREMENT OF THE DELAY IN GENERATING THE OUTPUT.

| **S No.** | Type of Operation | Avg. Time Lag measured |
|---|---|---|
| 1 | Joint 1 alone | 198 ms |
| 2 | Joint 2 alone | 196 ms |
| 3 | Joint 3 alone | 195 ms |
| 4 | Joints 1&2 | 216 ms |
| 5 | Joints 2&3 | 218 ms |
| 6 | Joints 1&3 | 217 ms |
| 7 | All the joints at once | 254 ms |

Table 3 shows the average response time difference between corresponding servo motor with a given joint movement. There are mainly 7 basic movements through which, the experiment was carried out.

## VI.CONCLUSION

The objective of designing a miniature arm which can control the robotic arm using potentiometers has been achieved. By using Arduino programming, a precise controlling system has been developed. An inference can be made from the observations that the robotic arm mimics the miniature arm movements accurately and precisely. Hence this miniature arm should resolve the issues that arises in the applications of the robotic arms. Due to the wear and tear of the robotic arm the error might have some negligible tolerance which can be corrected by calibrating the potentiometers.

## ACKNOWLEDGMENT

We wish to thank Almighty God who gave us the opportunity to successfully complete this venture. The authors wish to thank Amrita Vishwa Vidyapeetham University and the Humanitarian Technology (HuT) Labs for aiding us in this endeavor.

.

## REFERENCES

[1]. Gupta, G.S. Mukhopadhyay, S.C. Messom, C.H. and Demidenko,S. Master-Slave Control of a Teleoperated Anthropomorphic Robotic Arm with Gripping Force Sensing. In Proceedings of the IEEE Instrumentation and Measurement Technology Conference (2005), 2203-2208.

[2]. Bonny Varghese and Thilagavathi, B. Design and wireless control of anthropomorphic robotic arm. In Innovations in Information, Embedded and Communication Systems (ICIIECS) Conference (2015), 1-4.

[3]. Neo Ee Sian, Kazuhito Yokoi, Shuuji Kajita, Fumio Kanehiro and Kazuo Tanie. Whole Body Teleoperation of a Humanoid Robot-Development of a Simple Master Device using Joysticks. In Proceedings of the IEEE/RSJ Intl. Conference on Intelligent Robotic and Systems (2002), 2569-2574.

[4]. Muhammad Yaqoob, Sardar Rushan Qaisrani, Muhammad Waqas, Yasar Ayaz, Sohail Iqbal and

Sajid Nisar. Control of robotic arm manipulator with haptic feedback using programmable system on chip. In Proceedings of the Robotics and Emerging Allied Technologies in Engineering (iCREATE) (2014), 300-305.

[5]. Yamada, H. Muto, T. and Ohashi, G. Development Of A Telerobotics System For Construction Robot Using Virtual Reality. In Proceedings of the European Control Conference (ECC) (1999), 2975-2979.

[6]. NazreenBin Zainal Abidin, M. and Syed Saad Azhar Ali. 3-Degree-of-Freedon Real-Time Human Hand Following Robot Arm using Visual Feedback. In Proceedings of the International Conference on Emerging Technologies (ICET) (2015), 1-6.

[7]. Piotr Kopniak and Marek Kamiński. Natural interface for robotic arm controlling based on inertial motion capture. Proceedings of the 9th International Conference on Human System Interactions (HSI) (2016), 110-116.

[8]. Rajesh Kannan Megalingam; Thejus Pathmakumar; Trayesh Venugopal; Gokul Maruthiyodan; Allan Philip, “DTMF based robotic arm design and control for robotic coconut tree climber”, 2015 International Conference on Computer, Communication and Control (IC4) Pages: 1 - 6, DOI: 10.1109/IC4.2015.7375539