Available online at www.elixirpublishers.com (Elixir International Journal)

Information Technology

Elixir Inform. Tech. 80 (2015) 31245-31249

ISSN: 2229-712X

Remote Learning: Android Operated Educational Robot Arm with 6 DOF

Neerparaj Rai, Bijay Rai and Pooja Rai
SMIT, Sikkim and India.

ARTICLE INFO

Article history:
Received: 9 October 2014;
Received in revised form:
28 February 2015;
Accepted: 23 March 2015;

Keywords
Robotic arm
Android Smartphone
Bluetooth
Project-based learning.

ABSTRACT

This paper addresses an intelligent optimal control for a flexible robot arm that is driven by a permanent-magnet synchronous servo motor. An intelligent optimal control of robot arm with 5 axes with servo gripper is proposed by the instructions given through Android device (android app on smart phone). This proposal was addressed to fourth-year engineering students and combines knowledge from android java programming and robotic control to complete an automation project task. The educational robotic arm initially learns to pick up a randomly placed target from a target surface and move it to a predefined destination after which it repeats the contouring process under the presence of the target. This paper presents an off-line trajectory generation algorithm and, therefore, it possesses significant industrial implications, as no hardware changes are needed for its implementation. The proposed method has been experimented with Dexter ER-1 Robotic Arm for optimum project based learning.

© 2015 Elixir All rights reserved.

Introduction

Doing laboratory experiments is a fundamental part of the learning process because it allows theoretical knowledge and concepts to be assimilated [1] and applied , increasing student's motivation [2] and collaboration. This paper proposes the development of a project-based learning example (PBLE) of controlling an educational robotic arm in a simulated industrial environment. This PBLE was proposed to fourth-year engineering students at Sikkim Manipal University, Sikkim. The learning objective was the development of an educational robotic arm and android device interface to pick up a randomly placed target on a target surface and move it to a predefined destination through Bluetooth communication. During the initial operation, the robotic arm is made to memorize the complete trajectory and continuously follows the same under the presence of the target on the target surface. The design of such systems is an excellent tool for reinforcing applied engineering concepts. It is important for instructors in robotics to understand, however, that robotics is not just a tool to teach other aspects of engineering. Rather, it is a robust and mature discipline in its own right, with important applications in a wide range of fields [3].

The origins of robot manipulators date back to the 1940s, when Walters invented “machina,” the first robotic manipulator. Then, Devol invented the first industrial robot, “Unimate” in the late 1950s. Since then, industrial robot arms have proliferated and in today's automated industries, robot arms are well employed in various tasks on the factory floor. There are two application categories where industrial robot arms are usually employed: 1) positioning and 2) contouring. In positioning, the end-effector of the robot arm is moved from a start-point to an end-point, whereas in general, there is no major concern with the course of motion it takes in between the two stated points. A typical positioning application is pick-and-place. In contouring, the end-effector is supposed to move along a given Cartesian trajectory, obeying a given course of motion. Some typical contouring applications are welding, cutting, and painting.

The field of robotics have influenced almost all areas and it reduces the work of humans. This easiness provided by robot has made it to become more widespread across various industries ranging from manufacturing to health care. One such advancement is shown in this project. The materials available in the laboratory for the PBLE were an automation cell (Fig. 1) with a fixed target on the target area, a force sensitive resistor(FSR), an educational robotic arm, a destination container, Bluelink module, Atmega 328 microcontroller and android device(smart phone).

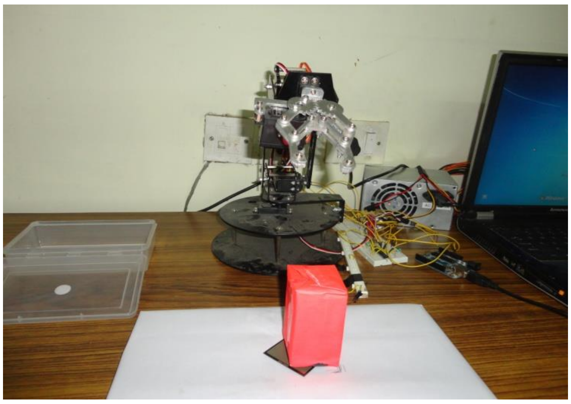


Figure 1. image of the complete setup used in this paper.

This paper is structured as follows: Section 2 describes the hardware and software used in this paper, Section 3 the proposed system used, Section 4 the results obtained and Section 5 the final conclusions.

Hardware and software used

Figure 1 shows the automation cell required for the development of the proposed PBLE. The most important elements are a fixed target on the target area, a force sensitive resistor(FSR), an educational robotic arm, a destination container, Bluelink module, Atmega 328 microcontroller and android device(smart phone). Eclipse and Arduino developing environments are used for creating applications for successful operation to control the robotic arm.

Educational robotic arm

The robotic arm used in this paper is the Dexter ER-1 Heavy Duty Robotic Arm (see Fig. 2) whose technical

Tele:
E-mail addresses: neerparaj_rai@yahoo.in

© 2015 Elixir All rights reserved

specifications and operational data are available in [17]. This educational robotic arm was designed to operate like an industrial robotic arm with 5 Axis robotic Arm + Servo Gripper and stepper motors in each joint. Application examples based on the use of educational robot can be found in [1] and [9].

The educational robot weighs 1.5 Kg with a theoretical load capacity of 50 g. Table I summarizes the displacement range of the six stepper motors included. The first motor drives the waist, the second the shoulder, the third the elbow, the fourth and fifth the wrist, and the sixth, the opening and closing of the gripper. . It uses four metal gear servo motors with 15Kg/cm torque and two servo motors with 7Kg/cm torque. Robot Arm has 5 degrees of freedom which includes Base rotation, Shoulder rotation, Elbow rotation, Wrist, pitch and roll.

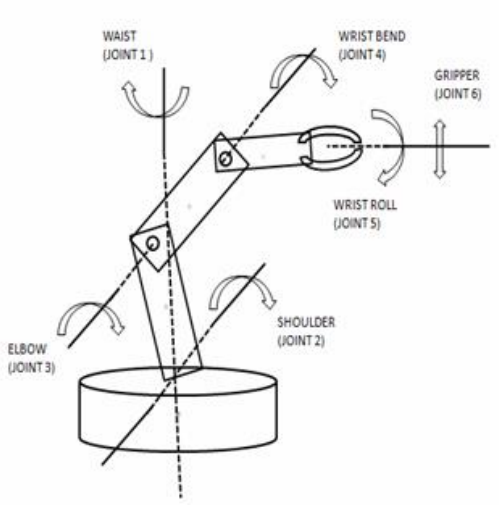


Figure 2. Schematic Representation Of Educational Robot Dexter Er-1

Servo control is done by sending each servo a PWM (pulse width modulation) signal, a series of repeating pulses of variable width. Servos are connected through a standard three-wire connection: two wires for a DC power supply and one for control, carrying the pulses. The parameters for this pulse are that it has a minimum pulse, a maximum pulse, and a repetition rate. Given the rotation constraints of the servo, neutral is defined to be the position where the servo has exactly the same amount of potential rotation in the clockwise direction as it does in the counter clockwise direction. It is important to note that different servos will have different constraints on their rotation but they all have a neutral position, and that position is always around 1.5 milliseconds (ms).

The angle is determined by the duration of a pulse that is applied to the control wire. This is a form of pulse-width modulation, however servo position is not defined by the PWM duty cycle (i.e., ON vs OFF time) but only by the duration of the pulse. The servo expects to see a pulse every 20 ms, however this can vary within a wide range that differs from servo to servo. The length of the pulse will determine how far the motor turns. For example, a 1.5 ms pulse will make the motor turn to the 90 degree position (neutral position).

Bluelink - bluetooth module

Bluelink is a compact Bluetooth Module (5V Serial TTL). The module has built-in Voltage regulator and 3V3 to 5V level converter that can be used to interface with 5V Microcontrollers. The module has only 5 pins (Standard 2.54mm berg strip) VCC, GND, TX, RX and RESET. The module is factory configured in Transparent Mode and hence there is no command required for normal operation. The technical details and its command formats are available in [16].

The Bluelink is a Drop-in replacement for wired serial connections, transparent usage. It can be used simply for serial port replacement to establish connection between MCU and GPS, PC for embedded project / Robot etc. Any serial stream from 9600 to 115200 bps can be passed seamlessly from PC/MOBILE to the target. The Bluelink should be connected to the serial pins of the microcontroller (Baud Rate 9600).

It is connected in receiver mode and is always ready to receive data (ASCII format) from the paired smart phone. The received data is then passed on to the microcontroller for further operation. Depending on the data (ASCII format) received from the Bluelink, microcontroller rotates the respective servos along with storing each instruction received in its EEPROM memory.

Force sensitive resistor

Force Sensing Resistors (FSR) are a polymer thick film (PTF) device which exhibits a decrease in resistance with an increase in the force applied to the active surface. Its force sensitivity is optimized for use in human touch control of electronic devices. FSRs are not a load cell or strain gauge, though they have similar properties. FSRs are not suitable for precision measurements.

A force sensitive resistor with a square sensing area of 1.75x1.5"is used. This FSR will vary its resistance depending on how much pressure is being applied to the sensing area. The harder the force, the lower the resistance. When no pressure is being applied to the FSR its resistance will be larger than 1MΩ. This FSR can sense applied force anywhere in the range of 100g-10kg. Two pins extend from the bottom of the sensor with 0.1" pitch making it bread board friendly.

The force vs. resistance characteristic shown in figure 3 provides an overview of FSR typical response behaviour. For interpretational convenience, the force vs. resistance data is plotted on a log/log format.

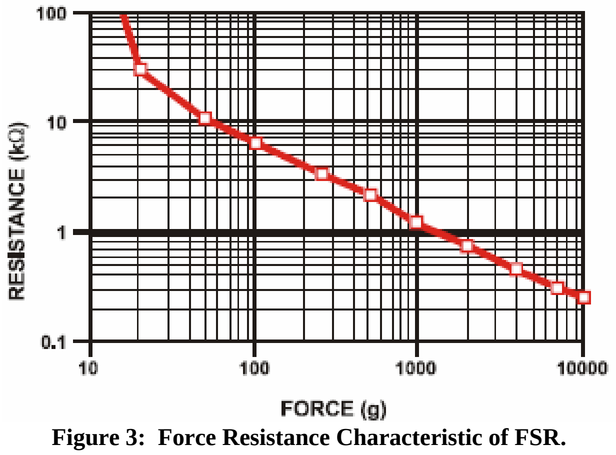


Figure 3: Force Resistance Characteristic of FSR.

Eclipse and Arduino

The Eclipse Project was originally created by IBM in November 2001 and supported by a consortium of software vendors. In computer programming, Eclipse is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop applications. By means of various plug-ins, Eclipse may also be used to develop applications in other programming languages: ABAP, C, C++ etc. The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.

The Eclipse (software) is used to create an android application for controlling an educational robot arm. This application developed in Eclipse makes the Bluetooth control via smart phone possible which runs on smart phone itself. This program sends from the phone the servo number and its required angle of rotation corresponding 2byte command to the Bluetooth module on the robot. Simple commands are then issued to the servo microcontroller from the Bluetooth module. This hierarchy allows easy and efficient changes in the implementation of the command scheme without requiring a redesign of the entire servo firmware package.

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software running on a computer (e.g. Flash, Processing, MaxMSP).

The panel arrange for robust communication to an Android device via a Bluetooth link and is entirely well-regulated from within an Android request via guileless and instinctive Java API - no fixed software development or outward programmer will ever be required [15]. The Arduino board holds only MCU that work as a USB crowd and infers guidelines from an Android application [14]. In accumulation, the board can network with marginal devices in the same way as most MCUs [1]. Arithmetical Input/Output, PWM, analog input, I2C, SPI, and UART mechanism are made available with the Arduino board.

The microcontroller executes all the controlling responsibilities. Atmega 328 microcontroller is operating at 16 MHz of frequency. The board transmit data on the pins of the Atmega 328 microcontroller. The controller then identifies at which pin the PWM signal has been sent and then it deals the servo motors of the arm accordingly. The microcontroller is also capable of serial transmission; the above frame is serially transmitted to the transmitter section. The microcontroller is used in 8 bit UART mode with 1 start bit, 8 data and 1 stop bit at 9600. Figure 4 explains the complete block diagram for the operation of the proposed system.

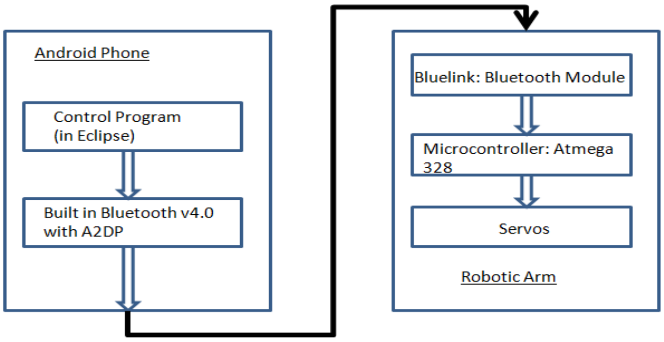

Android Phone
Control Program (in Eclipse)
Built in Bluetooth v4.0 with A2DP
Bluelink: Bluetooth Module
Microcontroller: Atmega 328
Servos
Robotic Arm

Figure 4. Robot Control System Block Diagram

The proposed system

A person operates the android device with an established connection to the MCU board. To develop it we use the eclipse for coding (java) and Android SDK for development of the apps. In this we write two set of coding. One is Bluetooth adapter coding to connect Bluetooth shield with mobile Bluetooth. And next is sending commands to the Robot to move in user desired direction. A signal is generated against each button which represents specific movements for a specific servo. These signals are received by pin 2(RX) on the MCU board from the Bluetooth module. The communication between the Android device and the MCU board is established with an ‘OPEN’ button in android application particularly developed for this project. Similarly, the established connection is disconnected using ‘CLOSE’ button (Fig.5). The current status can also be viewed on the ‘status’ edit bar. Simple commands are then issued to the servo microcontroller from the Bluetooth module.

Before the transfer of data from android smart phone to Bluetooth module, both needs to be paired. Pairing is a one-time process done per dongle/ module. Once paired, the Android device is going to send data to the paired dongle/ module and will not need pairing again and again.

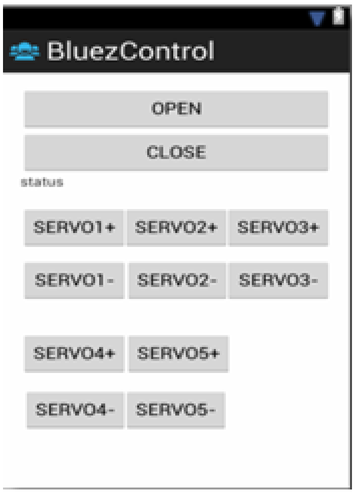


Figure 5. Android Application for Remote Control

The Bluetooth standard provides interfaces for a wide range of communications protocols, from a simple serial port to audio. Like many higher-level protocols such as OBEX file exchange, micro-controller sits on top of the serial port emulation layer of the Bluetooth protocol stack. It is not part of the “official” Bluetooth standard. However, the standard is relatively open in that anyone is free to create software for remote devices. The internally predefined values get fed into the program developed using BASIC 4.0 on the phone, and these values are then analyzed and the proper command value is then sent wirelessly to the Bluetooth module on the robot. The commands based on servo selection and angle of tilt for each servo. As shown in Figure 4, the Bluelink Bluetooth module allows the students to control the robot from certain Bluetooth capable devices, such as smart phones or laptops. This gives them hands-on experience with wireless control of embedded systems. A control program has also been implemented on an Android smart phone using Eclipse for Android. The program allows for remote control of the robot via Bluetooth.

The communication between Bluetooth module and microcontroller requires the submission of ASCII orders through the software serial port (2,3 pins). Unfortunately on the event of using multiple software serial ports, only one port can receive data at a time. The servo control function in MCU generates the different pwm signals to control 5 servos and gripper. The microcontroller interprets the commands issued by the Bluetooth module, and produces the proper sequence of timed pulses necessary to move the eight servos to the desired positions (increase or decrease the servo angles). The microcontroller produces a 50Hz sequence of pulses, the duty cycle of which controls the commanded angle of each servo. For example, a 600us pulse could command the servo to move to its full counter-clockwise position; a 1320us pulse could command it to the centre, or neutral, position, and a 2040us pulse could command it to move to the full-clockwise position.

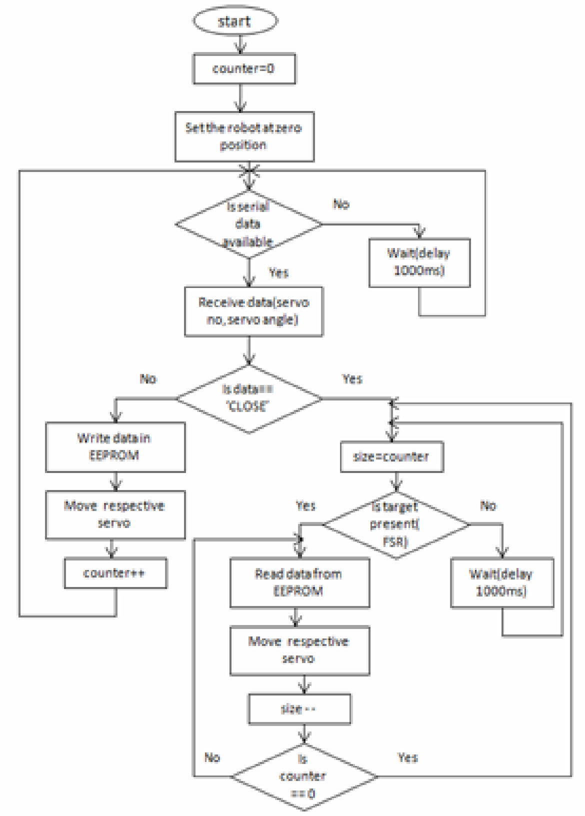


Figure 6. Diagram of a Visual Control Loop Based on a Trial and Error Approach.

The microcontroller on the circuit board receives these commands and implements them as canned timing signal sequences to be sent to the six servomotors and also writes the ASCII command received from Bluetooth module on its EEPROM memory. This data logging provides trajectory generation which is read from the memory for continuous operation which is useful for industrial purposes. This provides offline remote learning of an educational robot arm. The microcontroller on the Arduino board has EEPROM: memory whose values are kept when the board is turned off (like a tiny hard drive). The EEPROM library enables you to read and write those bytes. The microcontrollers on the various Arduino boards have different amounts of EEPROM: 1024 bytes on the ATmega328, 512 bytes on the ATmega168 and ATmega8, 4 KB (4096 bytes) on the ATmega1280 and ATmega2560.

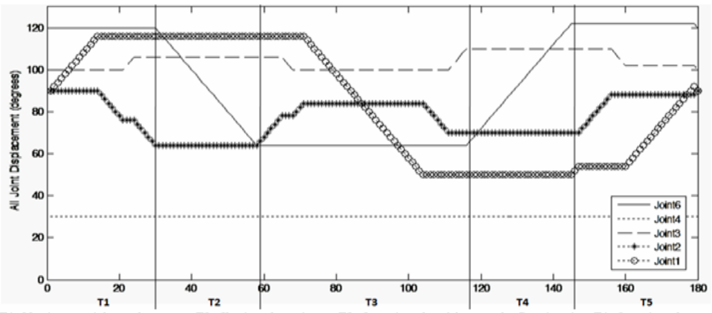
The microcontroller program reads the complete loop of data in EEPROM memory and the gripper is placed in the correct position. This makes the control of the robotic arm very simple following the initial complete operation of picking up a target and placing it inside the destination container. After the complete sequence of intermediate positions, the robotic arm is again set to its initial or zero position for the next operation. But the next operation is only executed under the presence of the target on the target surface which is continuously sensed by Force Sensing Resistors (FSR). The sensed value is continuously fed to the microcontroller through analog pin0 which initiates the complete cycle of operation. For example, the diagram of figure 6 details a visual control loop algorithm based on a trial and error approach.

Results

In industry, robot arms are modelled with linear servo model and controlled by PID servo controllers. Therefore, it is required that trajectory generation algorithms be devised on this basis so that the consistency between the proposed method and industrial setup is maintained and, therefore, the proposed method would bring about strong industrial implications. Even though this research is quiet in an initial step of improvement which uses 5 volt, 7kg cm torque servo but the proposed method is perfectly consistent with the modelling and control methods of most industrial robot arms. Even if the objective trajectory is modified with considerations to the constraints and desired performance, the learning of the robotic arm can be repeated and the complete operational loop can be stored in its internal memories at any moment. Though illustrated with a 6 DOF robot arm, the proposed trajectory generation method can be extended and applied to a higher number of links as well.

This is of utmost importance since in this configuration it is possible to plug this algorithm into the existing servo controllers straight away. For the same reason, it is expected that the proposed trajectory generation algorithm would bring about strong industrial implications. The proposed methodology permits a non-skilful person to automatically link with a variability of engineering robotic apparatus. Figures 7-9 show several images obtained during the practical development of the PBLE by the students. Further operational details can be viewed in a video available on YouTube [18].

As a summary of the results, Figure 8 shows an image obtained during the development of the visual feedback loop to pick up the target object. Figure 9 shows the robotic arm going to the predefined Position 1 and, finally, Figure 7 shows the robotic arm trajectories of opening the gripper and releasing the object in the destination container.



T1: Moving to pick up the target T2: Closing the gripper T3: Carrying the object to the Destination T4: Opening the Gripper T5: Going back to the home position.

Figure 7. Trajectories of Robot Joints While Picking up and Placing the Object.

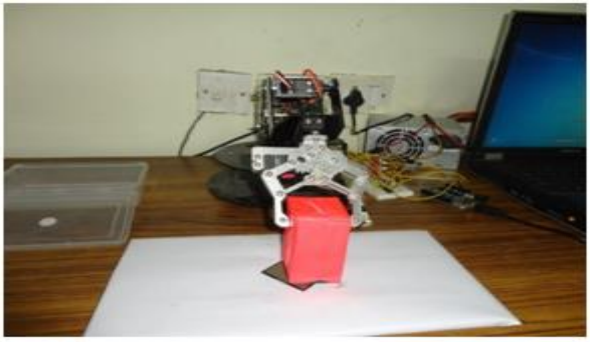


Figure 8. Image of the Robotic Arm Picking up the Target Object.

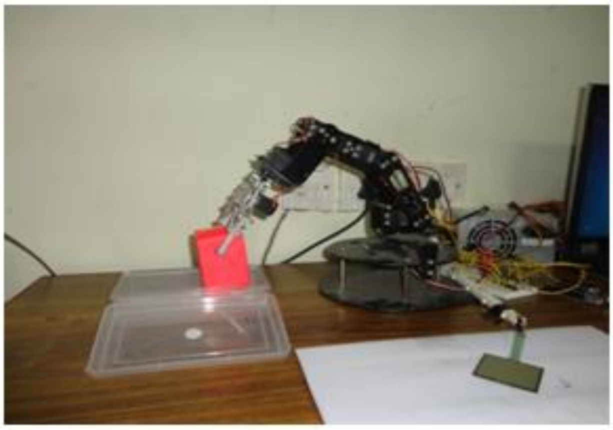


Figure 9. Image of the Robotic Arm Going to the Predefined Position.

Conclusion

This paper describes a project-based learning example addressed to fourth-year engineering students. This paper has successfully demonstrated the application of an intelligent robotic arm control system using an Android device. The proposition of the control scheme is to achieve good tracking performance with the flexibility of easy learning method. The design procedure of the proposed optimal control system was described in detail. Moreover, experimentation was carried out using different reference trajectories to verify the effectiveness of the proposed control system.

References

[1] A. Emin and G. Erdemir, "Development of a web-based control and robotic applications laboratory for control engineering education," Inf.Technol. Control, vol. 40, no. 4, pp. 352–357, 2011.

[2] E. Hyun, H. Yoon, and S. Son, "Relationships between user experiences and children's perceptions of the education robot," in Proc. HRI, 5th ACM/IEEE Int. Conf., Mar. 2010, pp. 199–200.

[3]V.B. Sunil & S.S. Pande 2003, "WebRobot: Internet based robotic assembly planning system," Computers in Industry 54 (2004) 191–207.

[4] Sotiris Makris , Panagiota Tsarouchi, "Intuitive dual arm robot programming for assembly operations," CIRP Annals - Manufacturing Technology, 2013.

[5] S. Rohan Munasinghe, Masatoshi Nakamura, "Optimum Contouring of Industrial Robot Arms Under Assigned Velocity and Torque Constraints," IEEE transactions on Systems, Man, And Cybernetics—part c: applications and reviews, vol. 31, no. 2, May 2001.

[6] Hwi-Su Kim and Jae-Bok Song, "Multi-DOF Counterbalance Mechanism for a Service Robot Arm," IEEE Asme Transactions On Mechatronics, 2014.

[7]http://www.researchgate.net/publication/229045477_Development of a Microcontroller Based Robotic Arm.

[8] Ill-Woo Park_, Jung-Yup Kim, Baek-Kyu Cho, Jun-Ho Oh, "Control hardware integration of a biped humanoid robot with an android head, " Robotics and Autonomous Systems 56 (2008) 95–103.

[9] Tomàs Pallejà Cabré Jordi Palacín Roca, "Project-based learning example: controlling An educational robotic arm with computer vision," IEEE Revista Iberoamericana De Tecnologias Del Aprendizaje, vol. 8, no. 3, August 2013.

[10] Zain Anwar Ali1, M.Tanveer, Habib Shaukat, Saad Anwar, "Android Operated Robotic Arm," Universal Journal of Control and Automation 2(1): 1-3, 2014.

[11] Rong-Jong Wai and Meng-Chang Lee, "Intelligent Optimal Control of Single-Link Flexible Robot Arm," IEEE Transactions On Industrial Electronics, vol. 51, no. 1, February 2004.

[12] Monika Jain, Aditi, Ashwani Lohiya, "Wireless Gesture Control Robot: An Analysis," International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 10, December 2012.

[13] Romeo Tatsambon Fomena, Omar Tahri, and Franc¸ois Chaumette, "Distance-Based and Orientation-Based Visual Servoing From Three Points," IEEE Transactions On Robotics, Vol. 27, No. 2, April 2011.

[14] "Arduino MCU", - www.arduino.com.

[15] http://microcontrollerkits.blogspot.com/2011/10/ioio-board-for-android-control.html.

[16] Web of the Bluetooth Module (5v Serial UART). Available:
http://www.rhydolabz.com/index.php?main_page=product_info&cPath=130_132&products_id=479.

[17] Web of the Robotic Arm Dexter ER-1 [Online]. Available:
http://www.nexrobotics.com/index.php?page=shop.product_details&flypage=flypage.tpl&product_id=667&category_id=124&option=com_virtuemart&Itemid=45.

[18] https://www.youtube.com/watch?v=U0lj4KQYRdE.