

Modeling and Implementation of Wireless Embedded Robot Arm for Object Sorting

C. Chandra Mouli¹, P. Jyothi², K. Nagabhushan Raju³

¹Senior Research Fellow, Department of Instrumentation, Sri Krishnadevaraya University, Anantapur, INDIA

²Research Scholar, Department of Instrumentation, Sri Krishnadevaraya University, Anantapur, INDIA

³Professor, Department of Instrumentation, Sri Krishnadevaraya University, Anantapur, INDIA

Abstract: Inverse Kinematic (IK) model of Dexter ER2 Robotic Arm and its implementation using wireless embedded system for object sorting application was presented in this work. Dexter ER2 Robotic Arm is a vertical articulated serial robot arm built by using DC servo motors. IK modeling of robot arm was carried out in PC by using geometric method which determines the joint angles of the robot arm for required end-effector position. Determined joint angles are transmitted from PC to ARM microcontroller LPC2148 using long range Zigbee wireless communication. LPC2148 was programmed in embedded 'C' in such a way that it receives the joint angles through long range Zigbee wireless communication and converts into corresponding PWM signals to control the DC servo motors for robot arm end-effector position. The results were taken for different points, where the end-effector can reach in its workspace. LabVIEW software package was used for modeling and transmitting the joint angles to the LPC2148.

Keywords: IK Modeling, ARM microcontroller, Robot arm, LabVIEW and Zigbee.

I. Introduction

Wireless embedded systems and robotics are the most developing technologies in this modern epoch. Bare embedded technology demands a system that could easily connect data transfer devices over distances – without using wires that grew stronger [10]. Embedded wireless technology is anticipated to burst and touches every area from robotics, industrial automation and medical devices, to the transportation infrastructure and manufacturing. The core of an embedded system is its processor/controller. ARM processor is one of the embedded processor that was taken the world of embedded systems to the next level. It provides powerful information processing capability and execution. Junhua Yang in his paper explained about ARM processor [11]. Mo Guan proved that ARM processor is best suitable for wide variety of wired and wireless network applications [19].

Robot arm is an electro-mechanical device which is capable of performing several jobs that ranges from simple mechanical tasks to extremely difficult jobs [1]. Robot arm modeling and implementation implicates the study of its kinematic behavior [2]. Kinematics of the robot arm gives the motion of bodies without concern of the forces or moments that cause the motion. IK modeling is important for analyzing the actions of a robot manipulator. Robot arm kinematics has been divided into forward kinematics and inverse kinematics [3]. The present study will focus only on IK model. The process of computing the joint coordinates for a given set of end-effector coordinates is called IK [8]. IK problem is more complex than forward kinematic problem in case of serial robotic arm [3]. Many researchers have evaluated and executed these problems in different scenarios using different tools and devices. Researchers frequently use geometric methods for serial manipulators which are relatively simple geometry [4]. Geometric method was used for IK model in this work, due to its versatility and acceptability to solve the kinematic model of any number of joints and links of a serial manipulator regardless of complexity [8].

5-axes articulated robot arm kinematic model was designed with a homogenous 4 x 4 matrix calculation in [5]. A low cost 4 DOF robot arm was designed using LabVIEW where joint angles are calculated and transmitted to the microcontroller through LabVIEW using wired communication [6]. To optimize the inverse kinematics problem of robot arm, an optimization process was carried out using neural networks and LabVIEW only for simulation in [7]. Existing object sorting robot arm systems described in [12-18] generally works with CISC microcontrollers and they use wired embedded system to control robot arms for object sorting.

IK model of Dexter ER2 Robotic Arm and its implementation using wireless embedded system for object sorting application. LabVIEW installed PC is used for IK model of robot arm and transmitting the joint angles. Long range Zigbee high level communication protocol is used to establish communication between PC and ARM microcontroller. The proposed system introduces the configuration of robot arm and ARM processor based wireless embedded system for object sorting application. The software part of PC was developed by Graphical Programme (GP) using LabVIEW and software part of ARM microcontroller was developed by

embedded 'C'. Since the present work was focused on IK modeling and its implementation, object sorting system was not presented in this paper.

Table I Salient Features of Dexter ER-2 heavy duty robotic arm

Mechanical Structure	Vertical Articulated
Number of Axes	5 axes plus servo gripper
Axis Movement	
Axis 1: Waist rotation	180°
Axis 2: Shoulder rotation	180° (Dual servos)
Axis 3: Elbow rotation	180° (Dual servos)
Axis 4: Wrist pitch	180° (Dual servos)
Axis 5: Wrist roll	180°
Maximum Operating Radius	308mm
End Effector	DC servo motor based gripper with Parallel finger motion
Maximum Gripper Opening	55mm
Actuators	5VDC servo motors
Motor Capacity (axes 1-4) (7 motors)	15Kg/cm
Motor Capacity (axes 5)	7Kg/cm
Motor Capacity (gripper)	7Kg/cm
Weight	1.5Kg
Power	5V-10Amp; 12V-2Amp (SMPS)

II. Inverse Kinematic Model

Dexter ER2 robotic arm is a vertical articulated 5-axes robot arm that was designed by using DC servo motors. DC servo motors are controlled through Pulse Width Modulation (PWM) signals. DC servo motors comprise of encoders which automatically deliver feedback to the motors and change the position consequently. The drawback of these motors is the rotation angle range is from 0^0 to 180^0 . Servo motors were selected based on the maximum torque mandatory to the structure and loads.

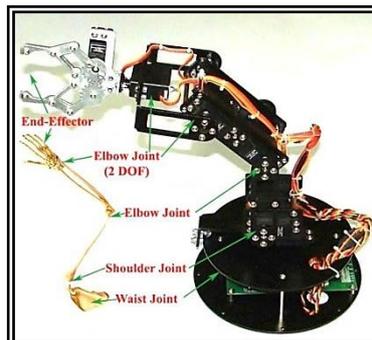


Fig. 1 Joint Configuration of Dexter ER-2 Robot Arm

Table I gives its salient features of the robot arm. Since the robot arm consists of six rotational joints from base to gripper it is a 6DOF robot arm. It has three links L1, L2 and L3 with lengths 9cm, 8cm and 13.8cm respectively. The gripper can open its tooth up to 5.5cm. Fig. 1 shows the joint configuration of the robot arm and its comparison with human arm. It uses 9 servo motors of which 7 are metal gear servo motors with 15Kg/cm torque and 2 servo motors with 7Kg/cm torque. It has 5 degrees of freedom which includes: Base rotation, Shoulder rotation, Elbow rotation, Wrist pitch and roll. Out of which Shoulder rotation, Elbow rotation, Wrist pitch has two 15Kg/cm torque servo motors in parallel for giving additional torque.

IK model involves solving the set of geometric equations using trigonometric functions. Generally the equations are complex and nonlinear, hence IK becomes more complicated.

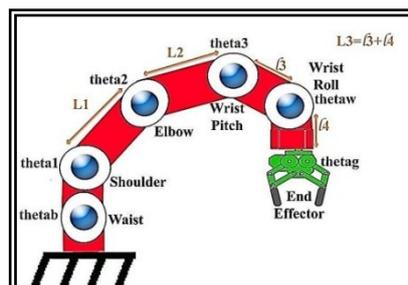


Fig. 2 Kinematic Model of Dexter ER-2 Robot Arm

Fig. 2 shows the kinematic model of the present robot arm. Basics of trigonometry give the joint coordinates of the robot arm for position and orientation of the end effector as follows

$$x = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) \quad (1)$$

$$y = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3) \quad (2)$$

$$\phi = \theta_1 + \theta_2 + \theta_3 \quad (3)$$

Eqn. (1), (2) and (3) gives the relationship between the effector coordinates and joint coordinates. To find the joint coordinates to the set of end-effector coordinates (x,y,ϕ), one needs to evaluate the nonlinear equations for θ₁, θ₂ and θ₃. Table II shows the link lengths of Dexter ER-2 robot arm. Link L₃ = 13+14 as shown in the Fig. 2.

Table II Link Lengths of Dexter ER-2 mrA toboR

Joint	Waist	Shoulder	Elbow
Symbol	L ₁	L ₂	L ₃
Link Length [mm]	90mm	80mm	138mm

Substituting (3) into (1) and (2), θ₃ can eliminate so that we have two equations in θ₁ and θ₂:

$$x - L_3 \cos\phi = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) \quad (4)$$

$$y - L_3 \sin\phi = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) \quad (5)$$

Rename the Eqn. (4) & (5) as x_p = x - L₃cosϕ, y_p = y - L₃sinϕ for ease. From Fig. 3 and the law of cosines we get Eqn. (6).

$$\cos\alpha = \frac{x^2 + y^2 - L_1^2 - L_2^2}{2L_1L_2}$$

$$\alpha = \text{Acos}\left(\frac{x^2 + y^2 - L_1^2 - L_2^2}{2L_1L_2}\right)$$

$$\theta_2 = 180 - \alpha \quad (6)$$

After intensive mathematical computations Eqn. (7) was yielded for θ₁.

$$\theta_1 = \text{Atan2}(y_p, x_p) + \text{Asin}\left(\frac{L_2 \sin\theta_2}{\sqrt{x_p^2 + y_p^2}}\right) \quad (7)$$

From Eqn. (3) $\theta_3 = \phi - \theta_1 - \theta_2$ (8)

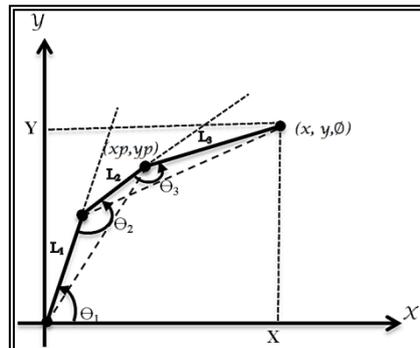


Fig. 3 x, y, ϕ plane

By executing the Eqn. (6), (7) & (8) using LabVIEW one can get the robot arm end-effector position. Since all the angles are measured in anti-clockwise therefore θ₂ & θ₃ are shown as negative angles and Eqn. (6) & (8) are modified as follows for positive angles. By executing Eqns. (7), (9) & (10) one can get the correct joint angles.

$$\theta_2 = \theta_2 - 270 \quad (9)$$

$$\theta_3 = 180 - (\theta_3 + 270) \quad (10)$$

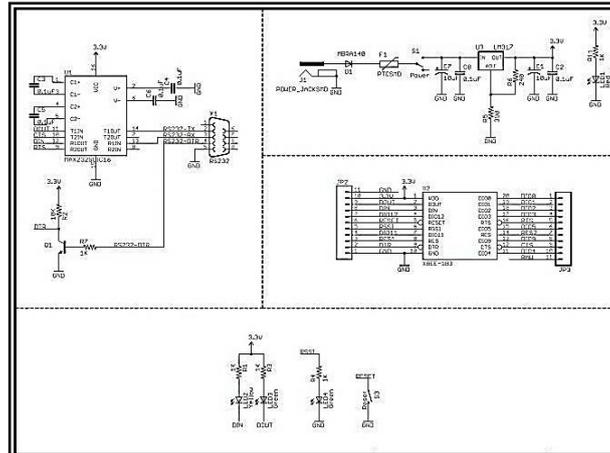


Fig. 4 Schematic Diagram of Zigbee Module at PC

III. Hardware Configuration

This section describes hardware configuration and components used. Hardware components used in the present work are PC, a pair of long range Zigbee wireless modules, ARM application specific board and Robot arm. PC used in the present work is a HP 430 Laptop. HP 430 comes with 2nd Gen Core i5 processor, 4GB for fast processing and 500GB HDD.

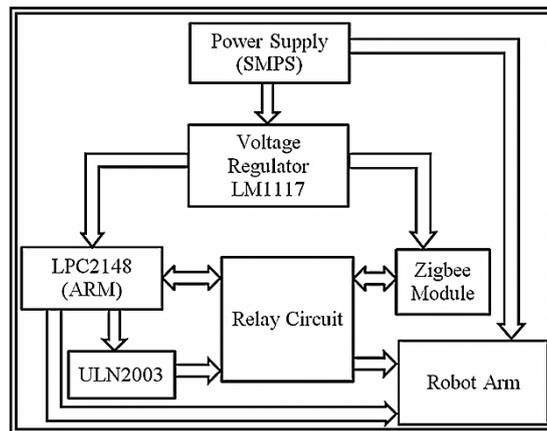


Fig. 5 Block Diagram of present work at microcontroller

Fig. 4 shows the schematic diagram of Zigbee module used to interface with PC. The Zigbee device used in the present work is XBee Pro 802.15.4 OEM RF module. It is used to establish the low power sensor networks which work with minimal power and produce reliable data between the nodes. It operates at ISM 2.4 GHz frequency band for long range communication. Since PC works with RS232 standard voltage levels and Zigbee works with TTL voltage levels, a converter is required to convert RS232 standard voltage levels to TTL voltage levels and vice versa. Standard serial interface RS232 of PC uses voltage levels in a range between -12V and +12V. For the serial signal it uses the voltage ranging between -3 and -12V stands for a logic one '1', whereas a voltage in a range between +3V and +12V stands for a logic zero '0'. In order to adjust this signal to voltage levels present on the XBee pins (TTL standard), it is necessary to use a voltage level converter. The MAX232 board features a built-in circuit MAX232 used to perform necessary adjustment. This circuit is powered with a single 3.3V voltage and used to convert a serial signal from TTL to RS232 standard and vice versa by means of a built-in voltage generator. The schematic diagram consists of MAX232 interfacing module schematic diagram at the top left corner, power supply schematic diagram for XBee module on the top right, XBee module schematic and the LED indicators used on XBee module. The schematic of MAX232 shows that pin no. 11 and 12 are used as transmission in and reception out respectively to the XBee end. Pin no. 14 and 13 are used as transmission out and reception in to the PC end.

Fig. 5 shows the block diagram of the present work at microcontroller end. The block diagram contains a 5V-10A Switch Mode Power Supply (SMPS), voltage regulator LM1117, Zigbee module, LPC2148 ARM microcontroller, robot arm, ULN2003 and relay circuit. 5V supply was regulated using LM1117 to 3.3V to provide power supply to LPC2148 and XBee module.

ARM microcontroller used in the present work is LPC2148 which comes with LPC2148 application specific board. This board is a powerful development platform based on LPC2148 ARM7TDMI microcontroller and ideal for developing embedded applications involving high speed wireless communication. The on board peripherals include XBee module interface, ULN2003 500mA current sinking driver. Since robot arm was designed by using nine 5VDC servo motors a 5V-10A SMPS was used.

6DOF robot arm requires six PWM signals to control the end-effector position. Because of the flexibility to configure six PWM channels, LPC2148 was chosen for the present work. Each PWM port pin of LPC2148 has multiple functions as shown in TABLE II. Since it requires total six PWM channels to control the robot arm, a relay circuit was used as shown in Fig. 3 to switch the control of port pins P0.8 and P0.9 from PWM4 and PWM6 to DI and DO of XBee Module respectively. The relay circuit consists of two 6VDC relays which are controlled by using port pins of LPC2148 through ULN2003. Port pins P1.20 and P1.21 were connected to the pins 1 and 3 of JP11 respectively. Pins 1 and 3 are shorted to 2 and 4 of JP11 to connect the control logic of P1.20 and P1.21 to ULN2003 pins I1 and I2 respectively. O1 and O2 are connected to ULN2003 pin1 and ULN2003 pin2 of relay circuit as shown in Fig. 3. When P1.20 and P1.21 are driven with '0' logic then P0.8 and P0.9 were connected to control the servo motors SM4 and SM5 of robot arm respectively. When P1.20 and P1.21 are driven with '1' logic then P0.8 and P0.9 were connected to DI and DO of XBee module to transmit/receive the signals.

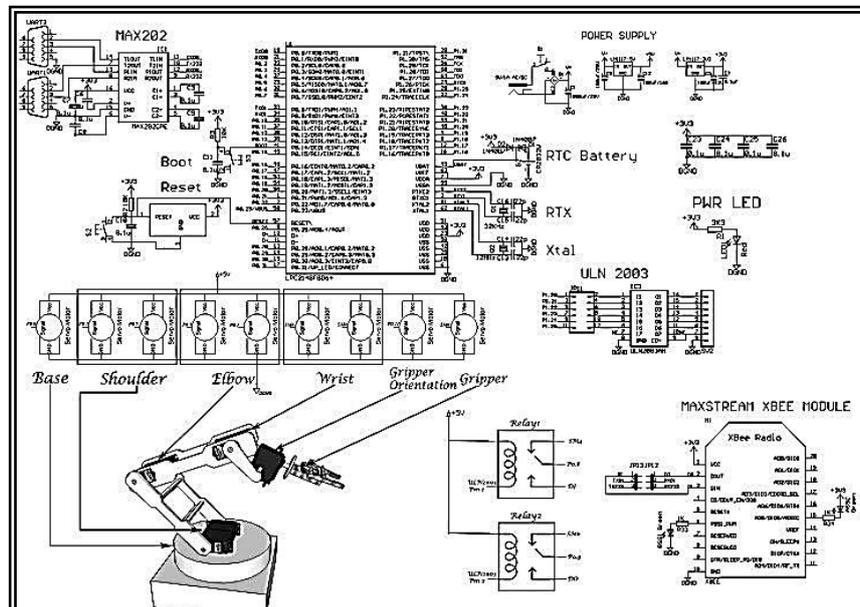


Fig. 6 Schematic Diagram of LPC2148 and Robot arm control

Table III Lpc2148 Pins Multiple Functions Configuration

Port No.	Functions
P0.0	PWM1/TXD0
P0.7	PWM2/SSEL0/EINT2
P0.1	PWM3/RXD0/EINT0
P0.8	PWM4/TXD1
P0.21	PWM5/CAPI.3
P0.9	PWM6/RXD1/EINT3

IV. Software Configuration

Software configuration of the present work was divided into two parts such as LabVIEW and embedded 'C'. LabVIEW program was developed on PC for inverse kinematics evaluation, simulation and transmission of joint angles to embedded system through XBee module. Embedded 'C' was used to program LPC2148 microcontroller, so as to receive the joint angles and convert the joint angles to respective PWM signals that controls the servo motors of robot arm.

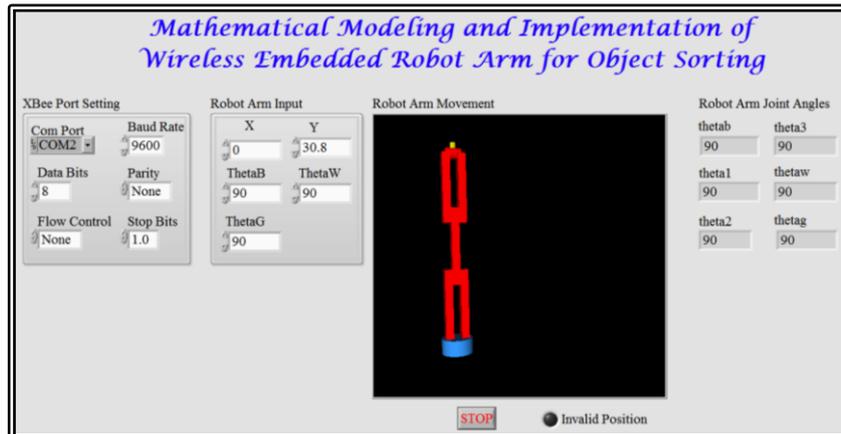


Fig. 7 LabVIEW Front Panel of the present work

A. Lab view

IK model and implementation of robot arm was presented in [20]. In this paper geometric method was followed to develop IK model and parallel communication was used to transmit the joint angles to LPC2148. IK model calculates only three joint angles i.e. shoulder (theta1), elbow (theta2), and wrist pitch (theta3) while the base (thetab) and end effector (thetag) is straight away given for desire position so as to control only five servo motors of the robot arm. But the present work provides the wrist roll (thetaw) control. Same as thetab and thetag, thetaw is also straight away given as input through front panel.

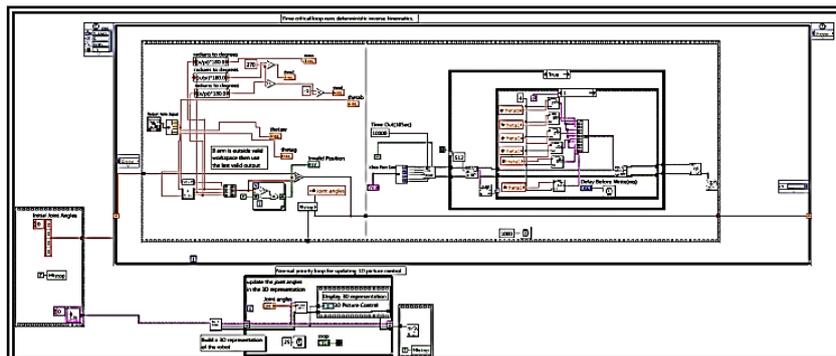


Fig. 8 LabVIEW Block Diagram of the present work

Fig. 7 shows the LabVIEW front panel of the present work. It consists of Xbee port setting, Robot Arm Input and Robot Arm Joint Angles. Xbee port settings is a cluster of six elements used to initialize the com port settings for Zigbee communication. It contains a VISA resource name VInamed as Com Port, one unsigned long 32-bit integer named as Baud Rate and four unsigned 16-bit integers named as Data Bits, Parity, Flow Control and Stop Bit. Com Port, Baud Rate, Data Bits, Parity, Flow Control and Stop Bit values were set to COM2 (since communication port was established in port2), 9600, 8, None and 1.0 respectively. Robot Arm Input is a cluster of five elements namely X, Y, ThetaB, ThetaW and ThetaG which are used as robot arm inputs. Robot Arm Movement is 3D Picture Control VI which is used to observe the robot arm transformation in simulation. Robot Arm Joint Angles are numeric displays thetab, theta1, theta2, theta3, thetaw and thetag are 64-bit numeric indicators used to display the joint angles of the robot arm. Invalid position and stop is the indicator and control Boolean variable respectively.

Fig.8 shows the block diagram of the present work using LabVIEW. Total block diagram has been divided into three parts such as IK model, Xbee communication and 3D picture control. IK model and Xbee communication was executed in a Time critical loop. It executes one or more sub diagrams, or frames, sequentially each iteration of the loop at a given period. To execute the IK program and Xbee module sequentially a flat sequence structure was used. It consists of two frames one for IK model and one for Xbee module program to execute sequentially.

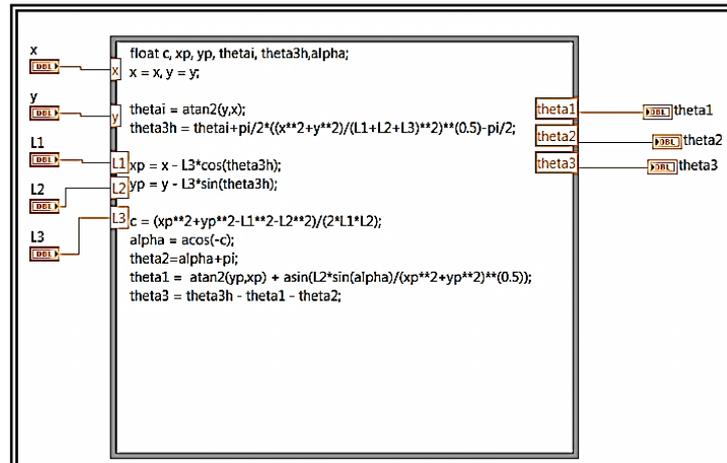


Fig. 9 Block Diagram of IK Program in LabVIEW

IK model was evaluated and Eqns. (7), (9) & (10) were yielded. These equations determine the correct joint angles for any object in the workspace of the robot arm. Fig. 9 shows the block diagram of IK program in LabVIEW. The VI calculates values for theta1, theta2, theta3 of Eqns. (6), (7) & (8). Eqns. (7), (9) & (10) are evaluated in the flat sequence structure. The sub VI was developed from the formula node that gives the detailed evaluation description of inverse kinematics for robot arm. Two input constraints x and y are used to determine three angles. thetai is the angle from the base to the end effector, theta3h is the position of the arm based on the distance from the base, xp and yp are the proximal joint of the last link and alpha is just a temporary variable. theta1 and theta2 are solved by using the cosine rule and theta3 solved from theta1, theta2 and theta3h.

Second part is the serial communication part which is used to transfer the joint angles of the robot arm from PC to LPC2148. VI named as VISA serial shows the VISA Configure Serial Port VI. It initializes the XBee port specified by Com Port and settings specified by the front panel controllers. The second VI is VISA Read VI reads ctrl_pin from microcontroller LPC2148. If ctrl_pin is equal to '0' then the third VI VISA Write VI writes the joint angles from write buffer to the device. Third part is the 3D picture display which is used to observe the simulation of the robot arm on the front panel of PC. Fig. 10 shows the flow chart of the total LabVIEW program.

B. Embedded 'C'

The most important phase in the present work is servo motor position control and software coding of the ARM controller LPC2148. Total 6 PWM signals were generated from LPC2148. After a deep investigation of servo motors, 0° angular position can be achieved with 600µs on-time in total time period of 20ms and 180° with 2.2ms on-time in total time period of 20ms.

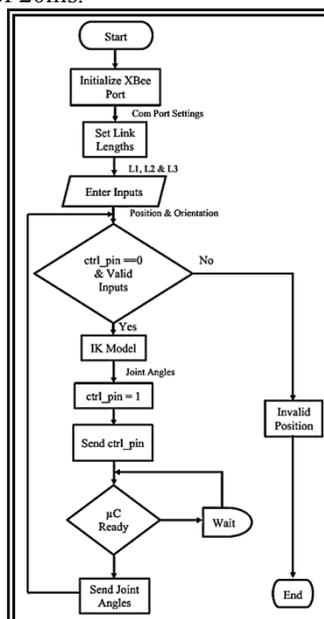


Fig. 10 Flow Chart of LabVIEW program

PWM signals were generated on the corresponding pins of LPC2148 by using the registers PINSEL, PWMPR, PWMPCR, PWMMCR, PWMMR0, PWMMR1, PWMMR2, PWMMR3, PWMMR4, PWMMR5, PWMLER and PWMTCR. The present work five PWM signals are generated. The PWM function for IO pins were selected by programming the control word to PINSEL registers. PWMPR is a PWM Prescale Register in which the value is incremented by PWMPR+1 for every phase locked loop clock cycle. PWMPCR is a PWM control register which enables the PWM output and controls the PWM channel type as either single edge or double edge. In the present work the PWM channel is controlled by using single edge. PWMMCR is a PWM Match Control Register and controls the PWM signal and if the PWMTC is reset when a Match occurs. PWMLER is PWM Latch Enable Register which enables the use of new PWM match values. PWMTCR is PWM Timer Control Register which is used to control the timer counter functions and timer counter can be disabled or reset. PWMMR x (where x is 0/1/2/3/4/5) are the match control registers which controls the total time period of the PWM and on -time of the pulse.

The total time period and on-time of one pulse depends on the values programmed to PWMMR0 and PWMMRx (where x is 1/2/3/4/5) respectively. In the present work 12MHz crystal is used for the controller crystal frequency. The crystal frequency is multiplied by a constant value 5 to get the processor frequency of 60MHz with the help of phased lock loop in the controller. The value in theregistersismodified using this processor frequency of 60MHz.

$$PWMMR0 = \frac{F_{cclk}}{83} \tag{11}$$

$$PWMMR1 = \frac{\left(\left(\frac{F_{cclk}}{83}\right) \times (100 - \text{count})\right)}{100} \tag{12}$$

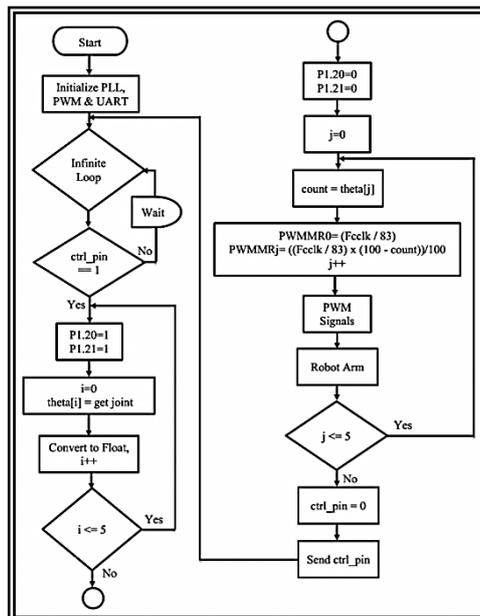


Fig. 11 Flow Chart of LPC2148 program

Eqn. (11) shows the formula to calculate the value to be programmed inPWMMR0 register to get total timeperiod of one pulse. Here FFclk is user defined macro which is used to represent the frequency of 60MHz. To get the 1Hz frequency pulse the value of the denominator should be 1.66 constant values. The denominator value of (11) is calibrated to 83 to get the 50Hz frequency pulse i.e. 20ms pulse. Eqn. (12) shows the formula to calculate the value to be stored in PWMMR1 register to get the on -time of the pulse. Here the value of PWMMR0 i.e. Fclk/83 is multiplied with 100-count value, where count is the percentage of on-time with respect to the duty cycle of the pulse. The product of these values is going to be divided with 100 to get the value into PWMMR1. The on-time of the pulse is decided by the variable, count and the formula is shown in (13). Fig. 11 shows the flow chart of LPC2148 embedded C program.

$$\text{count} = \left(\frac{\text{initialtime}}{\text{totaltimeperiod}}\right) \times 100 \tag{13}$$

V. Results

In this section, the experiments to verify the performance of the present design were presented by utilizing a 6-DOF robot arm. Inputs X, Y and Thetag determine the joint angles thetab, theta1, theta2 and theta3 whereas; Thetaw and Thetag inputs are straight away given. The intention of the experiment is same as in [22]. Primarily, locate the robot arm at its initial position (0, 30.8, 90). Then, it moves along x and y-axis depending upon the input position. Randomly few positions of simulation and actual motion have been presented below by keeping the values of thetab and thetag constant.

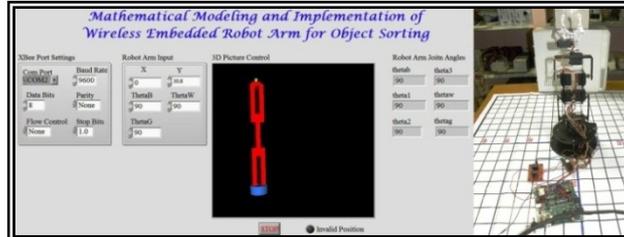


Fig. 12 Simulation and Actual Motion at (0, 30.8, 90)

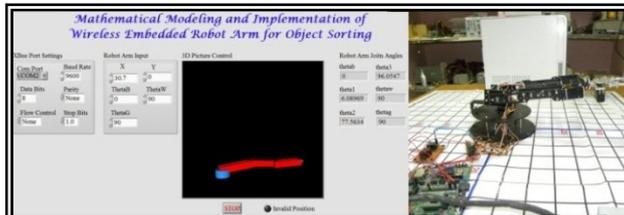


Fig. 13 Simulation and Actual Motion at (30.7, 0, 0)

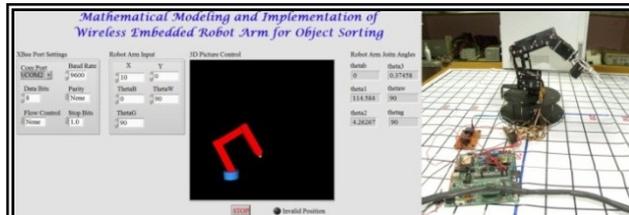


Fig. 14 Simulation and Actual Motion at (10, 0, 0)

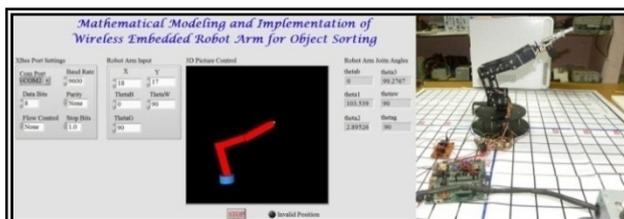


Fig. 15 Simulation and Actual Motion at (18, 17, 0)

- Fig. 12 shows the simulation and actual motion of the robot arm at (0, 30.8, 90). At this position robot arm all angles are at 90^0 .
- Fig. 13 shows the simulation and actual motion of the robot arm at (30.7, 0, 0). At this position robot arm angles thetab, theta1, theta2 and theta3 are at 0^0 , 6.08^0 , 77.56^0 and 96.05^0 respectively.
- Fig. 14 shows the simulation and actual motion of the robot arm at (10, 0, 0). At this position robot arm angles thetab, theta1, theta2 and theta3 are at 0^0 , 114.5^0 , 4.2^0 and 0.37^0 respectively.
- Fig. 15 shows the simulation and actual motion of the robot arm at (18, 17, 0). At this position robot arm angles thetab, theta1, theta2 and theta3 are at 0^0 , 103.53^0 , 2.9^0 and 99.27^0 respectively.

VI. Conclusion

IK model of Deter ER2 robot arm was developed and implemented using wireless embedded system. IK model developed has provided correct joint angles to place the robot arm end-effector in the desired position. The simulation results have been compared with actual motion of the robot arm. It was found that the robot arm end-effector position precision was with in ± 1 cm. This deviation is due to the mechanical coupling of the joints.

Acknowledgements

The authors acknowledge the help and support of University Grants Commission (UGC), Bahadur Shah Zafar Marg, New Delhi for providing the facilities for carrying out the research work.

References

- [1] AlenRajan, Aby K. Thomas, Rejin Mathew, A Comparative Performance Analysis of ARM based Web Servers with Integrated and External Ethernet Interfaces for Industrial Applications, *International Journal of Computer Applications* (0975 – 8887) Volume 44–No.21, April 2012.
- [2] JamsheedIqbal, Razaul Islam, and Hamza Khan, Modeling and Analysis of a 6 DOF Robotic Arm Manipulator, *Canadian Journal on Electrical and Electronics Engineering*, Vol 3, No 6, July 2012,pp: 300–306.
- [3] Mark S., Seth H. and VidyasagarM., *Robot modeling and control*, John Wiley & Sons, 2006.
- [4] Tsai L.W., *Robot Analysis: The mechanics of serial and parallel manipulators*, John Wiley & Sons, 1999.
- [5] Sanjay Lakshminarayan, ShwetaPatil, Position Control of Pick and Place Robotic Arm, *International Conference On Engineering Innovation and Technology*, ISBN : 978-93-81693-77-3, Nagpur, 1st, July, 2012.
- [6] Ashraf Elfasakhany, Eduardo Yanez, Karen Baylon, Ricardo Salgado, Design and Development of a Competitive Low-Cost Robot Arm with Four Degrees of Freedom, *Modern Mechanical Engineering*, 1, November 2011, 47-55.
- [7] Adrian Olaru and SerbanOlaru, Optimization of the robots inverse kinematics results by using the Neural Network and LabVIEW simulation, 2011 International Conference on Future Information Technology, IPCSIT vol.13 2011, IACSIT Press, Singapore, 40-45.
- [8] John J Craig, *Introduction to Robotics – Mechanics and Control*, Third Edition, Pearson Education, 2009.
- [9] AlenRajan, Aby K. Thomas, ARM Based Embedded Web Server for Industrial Applications, *International Conference on Computing and Control Engineering (ICCE 2012)*, 12 & 13 April, 2012.
- [10] MohdAshiqKamarilYusoff, Reza EzuanSamin, Babul Salam Kader Ibrahim, Wireless Mobile Robotic Arm, *International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012)*, *Procedia Engineering* 41 (2012) 1072 – 1078.
- [11] Junhua Yang, Zhien Shang and Tao XinG, Intelligence Monitoring System Based on ARM and Information Fusion International Conference on Electric Information and Control Engineering, pp.487-490, April 2011.
- [12] Vishnu R. Kale, V. A. Kulkarni, Object Sorting System Using Robotic Arm, *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, Vol. 2, Issue 7, July 2013.
- [13] DharmannagariVinay Kumar Reddy, Sorting of Objects Based on Colour By pick And Place Robotic Arm and with Conveyor Belt Arrangement, *Int. J. Mech. Eng. & Rob. Res.* 2014, Vol. 3, No. 1, January 2014.
- [14] Vishnu R. Kale, V. A. Kulkarni, Automation of Object Sorting System Using Pick & Place Robotic Arm & Image Processing, *Proceedings of 3rd IRAJ International Conference*, 5th January 2014, Mumbai, India. ISBN: 978-93-82702-51-1.
- [15] N.I. Giannoccaro, L. Spedicato, A. Lay-Ekuakille, A Smart Robotic Arm For Automatic Sorting of Objects With Different Tags, 4th Imeko TC19 Symposium on Environmental Instrumentation and Measurements Protecting Environment, Climate Changes and Pollution Control, June 3-4, Lecce, Italy, 2013.
- [16] Vikas S. Zilpe, ShahrukhPathan, Pranita Choudhary, A Smart Robotic Arm for Automatic Sorting Of Objects With Different RFID Tags, *Discovery*, 2014, 18(54), pp: 111-114, www.discovery.org.in.
- [17] Monal S. Telgote, Ganesh S. Sable, Object Sorting System in Matlab using Robotic Arm, *International Journal of Science and Engineering Investigations* vol. 2, issue 22, November 2013.
- [18] Vindhya Devalla, Dr. Rajesh Singh, Amit Kumar Mondal, VivekKaundal, Design and Development of Object Recognition and Sorting Robot for Material Handling in Packaging and Logistic Industries, *International Journal of Science and Advanced Technology* (ISSN 2221-8386) Volume 2 No 9 September 2012.
- [19] Mo Guan, MinghaiGu, Design And Implementation Of An Embedded Web Server Based On ARM, pp. 612-615, 2010.
- [20] C. Chandra Mouli, P. Jyothi, Prof. K. Nagabhushan Raju, C. Nagaraja, Design and Implementation of Robot Arm Control using LabVIEW and ARM Controller, *IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)*, e-ISSN: 2278-1676,p-ISSN: 2320-3331, Volume 6, Issue 5 (Jul. - Aug. 2013), PP 80-84.
- [21] Shouqian Yu; Weihai Chen; Li Li; Jianglei Qin, Development of ARM-based Embedded System for Robot Applications, *Robotics, Automation and Mechatronics*, 2006 IEEE Conference on , vol., no., pp.1,6, Dec. 2006.M Ozaki, Y. Adachi, Y. Iwahori, and N. Ishii, Application of fuzzy theory to writer recognition of Chinese characters, *International Journal of Modelling and Simulation*, 18(2), 1998, 112-116. (8)