# Design and development of a robotic arm

**Conference Paper** · October 2016

DOI: 10.1109/CIMCA.2016.8053274



Kiran Kumar B M
M. S. Ramaiah University of Applied Sciences


Sanjay Lakshminarayanan
M. S. Ramaiah University of Applied Sciences

# Design and Development of a Robotic Arm


Kruthika,K
Department of Electrical Engineering
M.S.Ramaiah University Of Applied Sciences
Bangalore, India
Email: kruthi2805@gmail.com

Kiran Kumar B.M
Department of Electrical Engineering
M.S.R.U.A.S
Bangalore, India
Email: kirankumar.ee.et@msruas.ac.in

Dr. Sanjay Lakshminarayanan
Department of Electrical Engineering
M.S.R.U.A.S
Bangalore, India
Email:sanjay.ee.et@msruas.ac.in



***Abstract*—Demand for assistive robots are increasing as they allow individuals to work independently and perform tasks that are impossible without them. This paper presents the design and development of 5-Degree of Freedom (DOF) robotic arm which is used for feeding the elderly or specially challenged people, where the position of the joints are controlled by the user. Using principles of robotic Kinematics and MATLAB, the robotic arm is controlled. The alogrithms are verified using stick diagram. A Graphical user Interface is developed for controlling the actuators which in turn controls each joint of the robotic arm. Arduino MEGA2560 I/O board is the main heart of this project which interface with the Graphical User Interface, motors and sensors. A mimic of the developed robotic arm is designed in MATLAB and it is simulated using forward and inverse transformation to check the accuracy of the algorithm.**



***Keywords*—*Robotic arm, degrees of freedom, feeding, GUI, Arduino, MATLAB, forward and inverse kinematics***


## I. INTRODUCTION

Robots have been used for assisting individuals in critical environments, research trends are towards developing robotic arm for assisting specially challenged or elderly people in their daily routine. This paper presents the designing and developing a safe robotic arm which is used for feeding food for specially challenged or elderly people. This paper is an in depth understanding the issues related to gripper positioning and solving it with the help of kinematics.

In literature it is evident that for the robotic arm to interact with humans, it should have easy controlling techniques and should be light-weight. In [1] the author explains the design of the 7-DOF robotic arm with cable driven which are used as pick and place robotic arm, cable driven methods acts as a disadvantage as it is not feasible. In [2] the author uses slider-crank mechanism which reduces the gravitational torque, this design is also not feasible as the slider causes vibrations.

A 5-DOF articulated robotic arm is designed which consists of five axes, the position and orientation of the robotic arm is solved from the base to the gripper using kinematics. This kinematic model is implemented using MATLAB and the same is simulated to check if the designed model reaches the destination point. Arduino Mega2560 is used as the brain of the robotic arm, force sensors are placed at the gripper for finding the force applied on the object, and potentiometers are used at the joints for detecting the position of the motor shaft.

This paper is organized as follows: Section II discusses the brief description of the project including the mechanical configuration and the components used. Section III involves the hardware description of the project which includes software and hardware implementation. Section IV discusses the kinematic modelling for 5-DOF robotic arm and finally the results and future works are discussed in Section V of the paper.

## II. PROJECT DESCRIPTION

Mechanical configuration describes the components used for fabricating the robotic arm for feeding application.

### A. Mechanical configuration

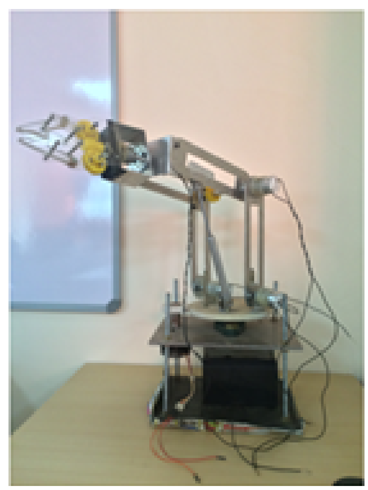

Fig. 1. Robotic Arm model

Fig 1. shows the exact model of the developed robotic arm built using Aluminum due to its characteristics such as light weight, do not wear out easily, cheaper and machining is easier. A stepper motor of 5 kg-cm torque is placed at the base which is used for rotating the entire arm clockwise or counter clockwise using pinon and gear mechanism. Other joints i,e., the shoulder, elbow, wrist and the gripper consists of DC geared motor of 3.5 rpm. DC motors and stepper motors are used as they are easy to control [9,10] giving HIGH or LOW pulses from the Arduino I/O board. To detect the position of the joint, potentiometers are placed, which acts as a sensor which feeds the output signal back to the Arduino I/O board for reducing the error signal. Force sensors are placed at the gripper for detecting the force applied on the object being held.

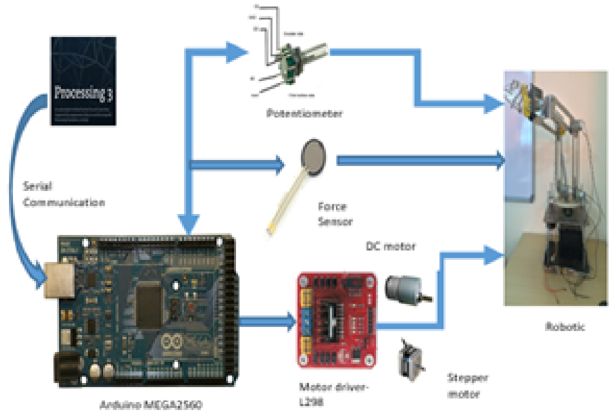


Fig. 2. Proposed block diagram for the robotic arm

## III. HARDWARE DESCRIPTION

### *A. Hardware description*

Fig 2 shows the complete block diagram for the robotic arm control. The commands from the user are given through the Graphical user interface which communicates with Arduino I/O board serially. Arduino I/O board on receiving the commands from the GUI controls the position of the motor shaft which in turn controls the movement of the joints.

### *B. Arduino Mega and L298*

Arduino Mega2560 consists of 54 digital I/O pins, 16 Analog pins [5], 14 PWM outputs and 4 serial pins. COMPIM is used for serial communication between Arduino I/O board and processing software which is used for developing Graphical user interface (GUI). Pins D2 to D16 are used for interfacing stepper motor, DC motor and sensors. L298 (H-bridge) are used for interfacing motors to Arduino I/O board, stepper motor consists of four wires, two wires for each coil. DC motors consists of two wires connected to H-bridge, H-bridge takes in 12V voltage supply for activating motors and gives +5V as output for Arduino I/O board.

### *C. Sensors*

The position of the motor shaft is detected using potentiometer which consists of 3 pins [4], one end of the potentiometer is connected to +5V supply, other end is connected to the ground, middle pin is connected to the analog pin on the Arduino board. Potentiometer are placed at the shoulder joint and the elbow joint. At the shoulder joint, gears are connected to the potentiometers shaft, these gears are connected to the gears placed on the stepper motor. Force sensors are placed at the gripper for detecting the amount of pressure applied on the object. Force sensor consists of 2 pins. A pulldown resistor of 10K [3,4] is used, one end of the sensor is connected to +5V and other end to the ground as well as the analog pin on the Arduino board. Programming the force sensor is similar to that of the potentiometer.

### *D. Graphical User Interface (GUI)*

The developed robotic arm is not completely closed loop system. An individual suffering from upper limb disorder will have less or limited limb functionalities to move their wrist or fingers. GUI takes the user inputs and move the joints accordingly. GUI was developed using Processing3 software which uses Java platform for programming [5], the main advantage of using this software is that it is an open source software and it can be programmed according to the user requirements. The developed GUI consists of 12 buttons, a couple of buttons are assigned to each motors which are attached to the joints of the robotic arm. Each button are programmed such a way that, till the button is pressed the motor rotates and stops rotating if the button are released.

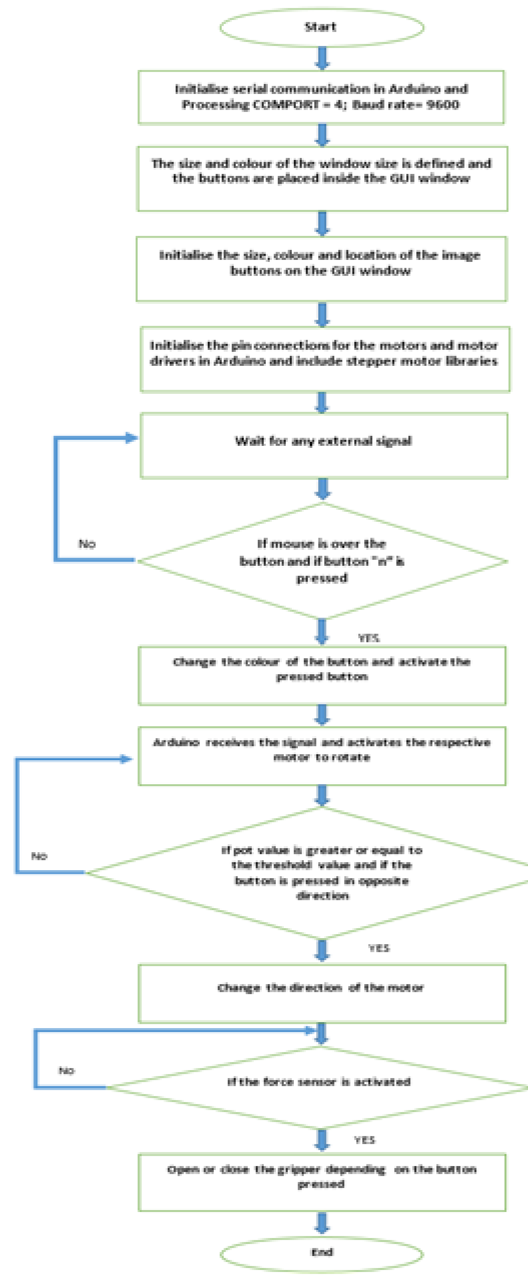


Fig. 3. Flowchart for Robotic arm operation

### E. Software implementation

Arduino IDE and Processing3 software platforms are used for automating the developed robotic arm, where both the softwares communicate with each other serially. Processing software are used for developing GUI as discussed above. Arduino IDE are used for initialing stepper motor, DC motors and sensors. Software flow diagram is as shown in the Fig 3, it starts from initializing the serial communication setting the baud rate as 9600 and communication port as 4, for transferring and receiving the information. Window size, color, the size and color of the buttons are specified, the location of the buttons are also defined in Processing. The pins for connecting the stepper motors, DC motors and the sensors are initialized in Arduino IDE. Buttons are assigned for each and every motor on the robotic arm.

## IV. KINEMATIC MODELLING FOR THE DEVELOPED ROBOTIC ARM

The geometrical configuration of the robotic arm consists of waist, shoulder, elbow, and wrist, all of these joints have single DOF.

### A. Forward kinematics model

Forward kinematics calculates the position and orientation of the end-effector with respect to the given reference frame, given the set of joint-link parameters. There are two different methods used are Denavit-Hartenberg (D-H) parameter [6] and successive screw displacements. D-H method and geometric methods are used in this paper for implementing forward kinematics. [8] D-H parameters works using four parameters twist angle, link length, link offset and joint angle $(\alpha_i, a_i, d_i and \theta_i)$-orthonormal coordinate frames are attached to each of the link of the robotic arm as shown in Fig 4 and the length of each link is as shown in the table-1.
Transformation matrix [7] describes the position and orientation of the frames which are assigned for each link.

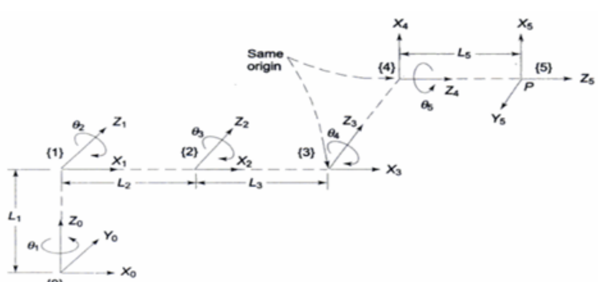


Fig. 4. Assigning Coordinate frame

TABLE I. LINK LENGTHS FOR THE DEVELOPED ROBOTIC ARM

| Joint | Waist | Shoulder | Elbow | Wrist |
|---|---|---|---|---|
| Symbol | $L_1$ | $L_2$ | $L_3$ | $L_4$ |
| Link Length (mm) | 350 | 250 | 220 | 180 |

A general method to calculate the matrices are as follows:

$$
{}_iT^{i-1} = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S_i & C_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
S_i = Sin_i, C_i = Cos_i, S_i = Sin_i, C_i = Cos_i \tag{1}
$$

$$
{}_nT^0 = {}_1T^0(q_1){}_2T^0(q_2)......{}_nT^{n-1}(q_n) \tag{2}
$$

The overall transformation matrix for the robotic arm is derived from the basic translation and rotation of the matrix as,

$$
T = \begin{bmatrix} C_1 S_a C_5 & -C_1 S_a S_5 + S_1 C_5 & C_1 C_a & C_1 c \\ S_1 C_b C_5 - C_1 S_5 & -S_1 S_b S_5 - C_1 C_5 & S_1 C_b & S_1 c \\ -C_b C_5 & C_b S_5 & -S_b & d \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

where, $C_{23} = S_{23} = a$
$C_{234} = S_{234} = b$
$L_2 C + L_3 C_a + L_5 C_{234} = c$
$L_1 - L_2 S_2 - L_3 S_a - L_5 S_b = d$

### B. Inverse kinematics

Inverse Kinematics is the procedure in which the joints are controlled in order to achieve the end position, given the position and orientation of the robotic arm. Solving Inverse kinematics is important compared to forward kinematics, as it can move the gripper to the target position, which would be helpful in grasping any object at the target location. There are two different methods in solving inverse kinematics [8]: geometrical and algebraic methods. As algebraic methods results in complicated solution as the Degree of Freedom (DOF) increases, geometrical methods are discussed in this paper. The target location is manually entered, algorithm checks is the target is within the workspace, if the target object is within the location Jacobian matrix is calculated and pseudo inverse is also calculated. This results in nonlinear solutions and is solved finding the minimum constraints, the end-effector frame reaches the object, if the solution is equal to that of the entered target location.

## V. RESULTS AND DISCUSSION

To test the functionality of the developed algorithm in Arduino IDE and processing, Proteus is used. Serial communication takes place between Proteus, Arduino IDE and Processing using COMPIM. Fig 5. shows the complete software setup in Proteus where serial communication takes place and the motors starts to rotate depending on the button pressed on the GUI Forward kinematics is implemented in Matlab and the results

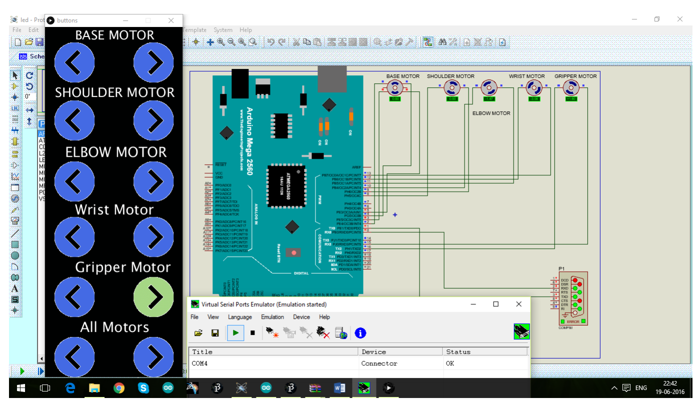


Fig. 5. Complete software setup for the system

for the same is shown in the Fig 6. The frames are assigned for each link, the joint angles are assigned for each joint for finding the position and orientation of the gripper. The animation for the forward kinematics shows the movement of frames from home position to the position defined. Using the transformation matrix calculated in the forward kinematic modelling, inverse kinematics is derived using Jacobian matrix and pseudo inverse to find the joint angles for each joint, given the position of the target object.

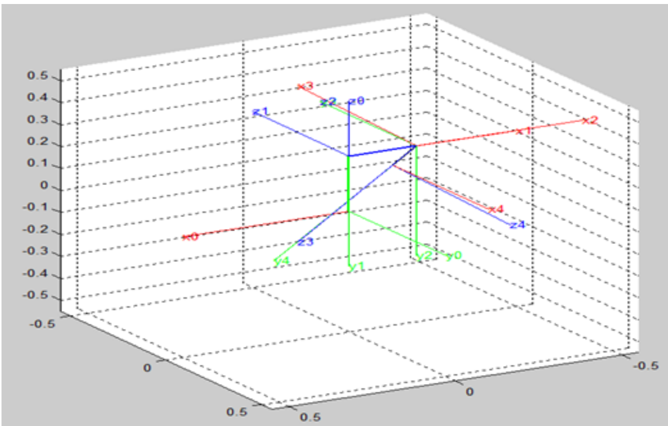


Fig. 6. Plot for joint angle configuration $[0^0 - 90^0 90^0 90^0]$

Fig 7 shows the animation of the stick diagram in Matlab, the frame of the end effector reaches the target location. $q_i$ is the actual joint angles for the robotic arm to move to the target location, as shown in the Table. II. $q_0$ is actual joint angles which is calculated by the robotic arm automatically using Inverse kinematics. Table. II shows the values of $q_i$ and $q_0$ which approximately equal to the desired location of the target.

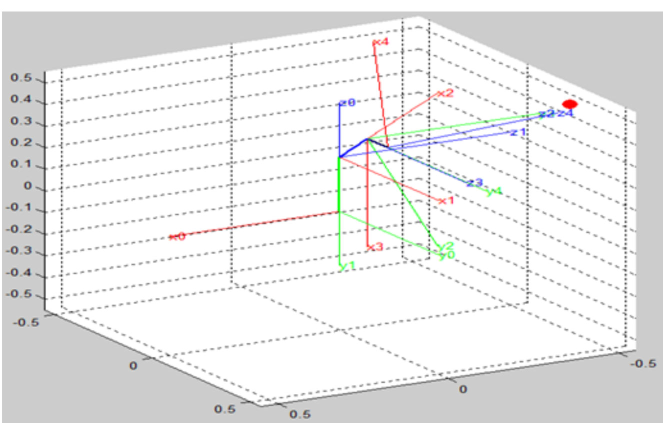


Fig. 7. Plot for Inverse kinematics to reach the target]

TABLE II. Link Lengths for the developed robotic arm

| Joint | Waist | Shoulder | Elbow | Wrist |
|---|---|---|---|---|
| $q_1$ | 1.570796 | -0.7854 | 0.785398 | 0.392699082 |
| $q_0$ | 1.570796 | -0.7854 | 0.785398 | 0.344830452 |

Fig 8. shows the complete setup of the robotic arm which includes GUI, Potentiometers at the elbow joint for detecting the position of the motor shaft, force sensor at the gripper.

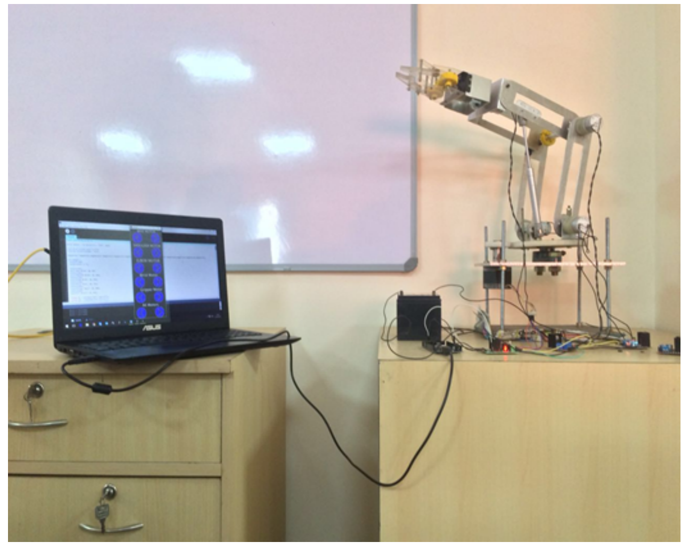

Fig. 8. Plot for Inverse kinematics to reach the target]

When the buttons on the GUI are pressed the corresponding motor rotates until an error signal is sent by the sensors. Hardware setup of the robotic arm can be automated using Inverse kinematics to move the arm to the target location.

## References


[1] Kurbanhusen Mustafa, S., Yang, G., Huat Yeo, S., Lin, W. and Chen, M., 2008. Self-calibration of a biologically inspired 7 DOF cable-driven robotic arm. Mechatronics, IEEE/ASME Transactions on, 13(1), pp.66-75.

[2] Kim, H.S., Min, J.K. and Song, J.B., 2016. Multiple-Degree-of-Freedom Counterbalance Robot Arm Based on Slider-Crank Mechanism and Bevel Gear Units. IEEE Transactions on Robotics, 32(1), pp.230-235.

[3] Kim, H.J., Tanaka, Y., Kawamura, A., Kawamura, S. and Nishioka, Y., 2015, August. Development of an inflatable robotic arm system controlled by a joystick. In Robot and Human Interactive Communication (RO-MAN), 2015 24th IEEE International Symposium on (pp. 664-669). IEEE.

[4] Mohammed, A.A. and Sunar, M., 2015, May. Kinematics modeling of a 4-DOF robotic arm. In Control, Automation and Robotics (ICCAR), 2015 International Conference on (pp. 87-91). IEEE.

[5] Siciliano, B., 2009. Robotics. London: Springer

[6] Zhang, Z., 2015. Wearable sensor technologies applied for post-stroke rehabilitation (Doctoral dissertation, RMIT University).

[7] Qassem, M.A., Abuhadrous, I. and Elaydi, H., 2010, March. Modeling and Simulation of 5 DOF educational robot arm. In Advanced Computer Control (ICACC), 2010 2nd International Conference on (Vol. 5, pp. 569-574). IEEE.

[8] Bhuyan, A.I. and Mallick, T.C., 2014, October. Gyro-accelerometer based control of a robotic Arm using AVR microcontroller. In Strategic Technology (IFOST), 2014 9th International Forum on (pp. 409-413). IEEE.

[9] Anon, 2016. The Ninth International Symposium. [online] Robotics Research. Available at: http://Robotics Research: The Ninth International Symposium [Accessed 4 Apr. 2016]

[10] Condit, R. and Jones, D.W., 2004. Stepping motors fundamentals. Microchip Application Note: AN907,[Online]. Available: www. microchip. Com