

# Design and Control of a Robotic Manipulator for Pick and Place Applications

Adham Mohamed, Ahmed Abd-Elmonen, Ahmed Asem, Ahmed Atef, Ahmed Kamal Mohamed, El-Hussein Abd El Ghani, Mohamed Yasser, Mostafa Ashraf, and Omar Mohamed

*Egyptian Academy for Engineering and Advanced Technology, Egypt, adhamgoda252@gmail.com,*

*ahmedelkholy382@gmail.com, ahmedasem2691999@gmail.com, ahmedatef026@gmail.com, ahmd.kamal150@gmail.com,*

*elhusseinabdelghani@gmail.com, m01202698139@gmail.com, mostafa.elazab19@gmail.com, omahamad825@gmail.com*

Supervisor: Mahmoud Ahmed, Dr.

*Egyptian Academy for Engineering and Advanced Technology, Egypt, m.abohatab@eaeat-academy.edu.eg*

**Abstract**— *In this paper, the design and control of a robotic manipulator is introduced. The objective of this robotic manipulator is to be used for pick and place applications. The robotic arm's movement may be monitored and managed remotely. The controller for the robotic arm in this project is an Arduino Mega board. Three stepper motors and two servo motors are used in the robotic arm to move three links and a gripper, respectively. Both the forward and backward kinematics are obtained, such that the relation between the position of end-effector in Cartesian coordinates and the joint angles are determined. The determined joint angles are used to form the trajectory to be tracked by the controller. The parts of the robot manipulator are constructed using SolidWorks, a stress analysis was carried out on each part. On the Arduino mega board, the controller system is built using the Arduino IDE programming language. It also enables choosing and positioning activities to those found in the manufacturing processes, as well as monitoring and tracking the robotic arm's path.*

**Keywords**— *Robotics, computer aided design, system modeling, control.*

## I. INTRODUCTION

One of the automation methods that is frequently used in a number of industries, including heavy industry, the health industry, and smart homes, is robotics. Robotics is an interdisciplinary field of technology that integrates mechanical, electronic, and electrical engineering with artificial intelligence, computer vision, and other computer sciences to produce robots that can do jobs like people and even take their place [1]. One of the most common robots is a pick and place robot, which is used to expedite the process of picking up objects, transferring them, and placing them in new locations [2].

The industrial pick and place robot is one of the most essential indicators. The major goals of this project are to develop and build a four-degree-of-freedom pick-and-place robotic arm. This project will be capable of controlling and executing simple actions like as grabbing, lifting, placing, and releasing. Using an ESP 32 controller, this project will design and create a "Robotic Arm for Pick and Place Application." This project combines electronic and electrical knowledge [3],[4].

The research looks at how robots have been introduced into the workplace to replace humans, particularly in repetitive

tasks. Mechanical adjustments were made to the robot's body, and electrical parts that might function as a robotic arm were selected. The Arduino Uno, which serves as the robot's brain, is controlled by the Arduino Ethernet Shield, which serves as the Arduino Uno's internet interface. Inputting the appropriate degree of robotic arm movement controls the robot's movement, and the robotic arm will then move to the desired movement that has been inputted. The robotic arm may also make a preprogrammed movement at the push of a button. The user interface that will be provided when the operator uses the internet to operate the robotic arm is created using an HTML [5], [6], [7]. Based webserver in this project. Acrylic was chosen as the robot's foundation because it is easy to shape, affordable, and strong enough to sustain the weight and movement of the motor. The robot gripper is also made of aluminum, following the same logic as the main robot arm structure. The main components of this system are the robotic arm and the computer system. In this project, the Arduino Uno acts as the system's controller [8],[9],[10]. The Arduino Uno will connect to the internet through a LAN connection with the help of an Arduino Ethernet shield. Any computer with an internet connection may then access and operate the robotic arm. To link the robotic arm to the internet, a relay infrastructure is used. Relay technology offers safe Web connectivity to embedded systems behind a firewall or NAT. Yaler.net is a company that offers a free relay infrastructure [11],[12],[13]. The major purpose of this specification is to clarify some of the project's essential components and ensure that it is both feasible and fit for use in the marketplace. The inspection robot's hardware is enhanced by the robot software, which provides basic low [14],[15]. Level hardware control, such as receiving input data from the internet and activating servo motors.

The Internet of Things (IoT) is a new age that has emerged as a result of technological advancements (IoT). The Internet of Things (IoT) is the networked linking of physical objects, such as automobiles, buildings, and other items, in order to collect and share data. It also allows things to be detected or controlled remotely through existing network infrastructure, resulting in better performance, accuracy, and cost savings. Robots are currently widely utilised in most

sectors owing to additional benefits like as reliability, high precision, and accuracy, which may compensate for a human's incapacity to function in a dangerous environment [16]. As a result, it's reasonable to claim that robots are being used to replace people. A robot is a machine that uses user. Created programming to do a specific activity. It also allows for simultaneous multitasking, which is quite useful. The most prevalent industrial robot is the robotic arm. A robotic arm is a mechanical arm that may be taught to do tasks similar to those performed by a human arm. The robotic arm's main purpose is to move an end effector from one position to another in order to pick up and carry various things.

## II. MECHANICAL DESIGN

In many sectors that manufacture goods, robots are essential. The rationale is that operating a robot cost far less per hour than using human labor to carry out the same task. Furthermore, once trained, robots consistently carry out tasks with a high level of precision that exceeds that of the most skilled human operator. However, human operators are far more adaptable. People can simply move between employment duties. Job-specific robots are created and programmed [17].

To do pick-and-place tasks, we need to design a robotic arm that can be employed in a variety of settings, namely risky close quarters where exposure to chemicals might be harmful to workers' health. Using a 4-DOF robotic arm with stepper motors acting on the joints, our research will attempt to emulate the movements of a human arm. Aiming at a compact arm to be easily placed and used in small working areas and to be easy to operate and handle in a safe manner. Links should be of a small thickness not a thick one to achieve lightweight and low cost by using suitable motors for the small loads acting on each joint and avoiding more material usage.

### A. Static Torque Calculations

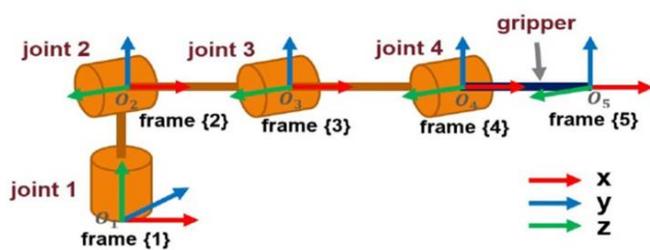


Fig. 1 Kinematic Diagram

#### Joint 1

The torque applied on joint 1 is,

$$T_1 = (M_l + M_g) \times g \times R_1, \quad (1)$$

where,

$M_l$ : mass of load

$M_g$ : mass of gripper equals 340.25 g

$R_1$ : distance between load and motor 1

$g$ : acceleration due to gravity that equals  $9.81 \text{ m/s}^2$

#### Joint 2

The torque applied on joint 2 is,

$$T_2 = (M_l + M_g + M_1 + M_{l1}) \times g \times R_{12}, \quad (2)$$

where,

$M_1$ : mass of motor 1

$R_{12}$ : the distance between load and motor 2 equals  $r_1 + r_2$

$M_{l1}$ : mass of link 1

#### Joint 3

The torque applied on joint 3 is,

$$T_3 = (M_l + M_g + M_1 + M_{l1} + M_{l2} + M_2) \times g \times R_{13}, \quad (3)$$

where,

$M_2$ : mass of motor 2

$M_{l1}$ : mass of link 1

$M_{l2}$ : mass of link 2

$R_{13}$ : distance between load and motor 3

#### Joint 4

The torque applied on joint 4 is,

$$T_4 = (M_l + M_g + M_1 + M_{l1} + M_{l2} + M_2 + M_b + M_3) \times g \times R_{14}, \quad (4)$$

where,

$M_3$ : mass of motor 3

$M_{l1}$ : mass of link 1

$M_{l2}$ : mass of link 2

$M_b$ : mass of base

$R_{14}$ : distance between load and motor 4,

where,

$$R_{14} = (r_1 + r_2 + r_3 + r_4)$$

The following parameters in Table I are considered for static torque calculations,

TABLE I  
STATIC TORQUE CALCULATIONS

Joint	Link length (cm)	Link mass (kg)	Motor mass (kg)	Torque (kg.cm)	Torque (N.m)
1	10	0.26	0.04	1.7	0.167
2	40	0.59	0.02	26.3	2.579
3	72	1.09	1.15	213.86	20.972
4	10	1.5	1.1	263.86	25.876

### B. Selection of Bearings

Factors affecting selection of bearings are:

1. Loads: Both radial and axial loads are present in our project
2. Rotational Speed: Low speed rotations are performed Based on the previous factors the taper roller bearing was chosen.

### C. Selection of Material

We considered the materials and components we could use in the manufacture of the pick and place robot arm, depending on specific objectives, requirements, and intended uses of an all

pieces of machine, with consideration for budget, capacity, and the construction process of the pick and place robot arm.

**- Robot Base Material**

MDF, or medium-density fiberboard, medium density fiberboard is one type of composite wood product. Wood scraps are used to make MDF board. Natural wood is more expensive than MDF, which is a more affordable option. It is therefore a less costly option than bare wood. We utilized 14 mm thick MDF for the project basis and 18 mm thick plywood to construct it.



Fig. 2 MDF material

**- Links**

When weight to strength ratio is crucial, aluminum is the best material to use. Cast iron, another typical metal used in robotics, has a lower tensile strength than steel. The most accessible of the three, aluminum offers an affordable product.

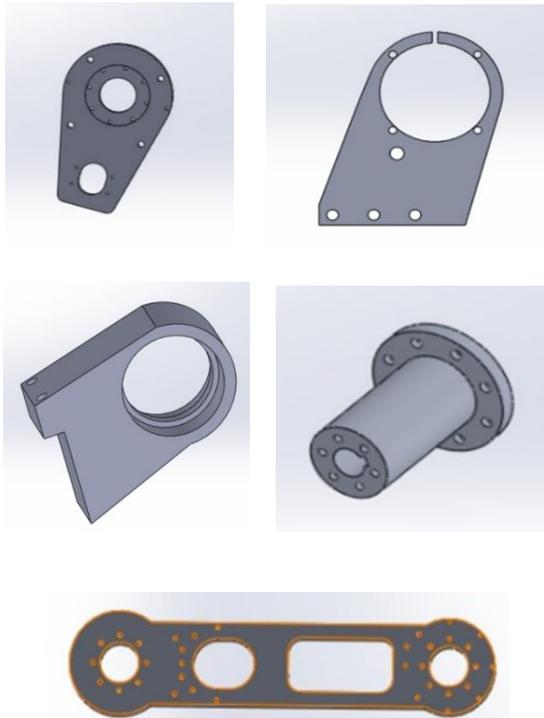


Fig. 3 Turret housing for joint 2, spindle, and Links on SolidWorks

**- Gripper**

The best option for us was 3D printing material, depending on the intricacy of the gripper design and our goal

load. The additive manufacturing process of 3D printing is used to create parts. It often produces more complex geometries quickly and at low fixed setup costs. It is frequently used in engineering, particularly for creating and prototyping lightweight geometries.

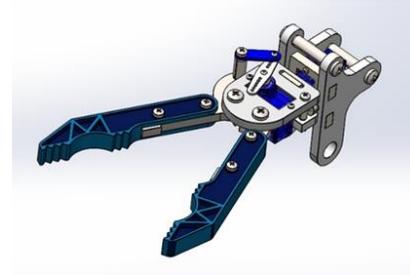


Fig. 4 Gripper

**- Robotic Arm Assembly**



Fig. 5 Robotic arm assembly

**- Stress-Strain Analysis**

Because most engineering components fail due to stress, stress analysis is an essential aspect of engineering science. The stress analysis gives an indicator of device structural reliability. The study of stress in solid things is known as stress analysis. Given the external pressures operating on the system, the primary task in stress analysis is to identify the distribution of internal stresses across the system. In theory, this entails deciding, whether implicitly or explicitly. The final goal of any study is to compare the produced stresses, strains, and deflections to the design criteria. To avoid failure, all structures and its components must be constructed to have a capacity larger than what is predicted to develop throughout their usage.

The computed stress in a member is compared to the strength of the material from which it is made, and the ratio of the calculated stress to the material's strength is determined. The ratio must be higher than 1.0 in order for the member to not fail. The ratio of permissible stress to produced stress, on the other hand, must be larger than 1.0 as a factor of safety (design factor) in the structure's design requirement. The design factor (a number larger than 1.0) indicates the degree of uncertainty in the load values, material strength, and failure implications. The working, design, or limit stress is the stress that the structure is intended to undergo.[4]

$$\text{stress} = (\text{elastic modulus}) \times \text{strain}, \tag{5}$$

where,

$$\text{stress} = \frac{F}{A}, \tag{6}$$

$$\text{strain} = \frac{\Delta L}{L_0}. \tag{7}$$

$F$  is the applied force,  $A$  is the area,  $\Delta L$  is the change in length due to applied force, and  $L_0$  is the original length.

Fig. 6 is an example of stress. Strain analysis applied in SolidWorks on a link in our project in which a force is applied on the part in a direction similar to the direction of forces the part will be exposed to in our application. The grading of color column on the side of the picture stand for the deformation and hence for the stress with blue is the least stress and deformation and red is the highest values.

Areas with blue color are under minimum stress, green areas are under medium stress while red areas are the parts of the design which has a great amount of stress which is absent in all of our linkage designs.

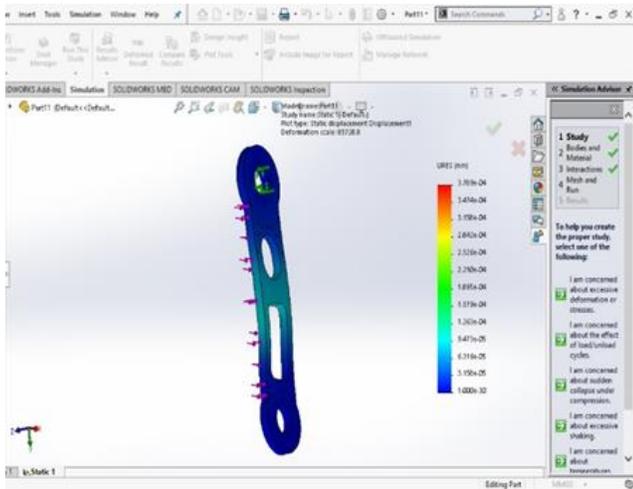


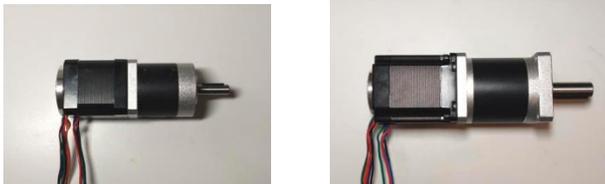
Fig. 6 Stress strain analysis

### III. ELECTRICAL HARDWARE

#### A. Electrical Components

##### - Stepper Motors

Four stepper motors have been selected according to the desired torques obtained from Table I. These motors are shown in Fig. The *NEMA 17 SKU: 17HS15-1684D-HG10* is used for joint 1 and the *NEMA 23 SKU: 23HS22-2804D-HG50* is used for joint 2.



Joint 1 NEMA 17 Stepper Joint 2 Motor NEMA 23 Steeper  
Fig. 7 The used stepper motors

##### - Servo Motor

As the gripper doesn't require high braking torque for fixing and holding the object, a servo motor is used. The *Servo Motor SG90* shown in Fig. was selected to be used for the gripping task.

We must power the motor with +5V via the Red and Brown wires, then transmit PWM signals to the orange color wire to make it rotate. To make this motor function, we'll need

something that can create PWM signals, which Arduino. the PWM signal generated should have a frequency of 50Hz and a PWM period of 20ms.



Fig. 8 The used stepper motors

##### - Power Supply

A power supply is used to convert from 220 Volt AC to 24 Volt DC with output current of 8.3 A



Fig. 9 MIWE Power Supply

##### - Arduino Mega

Arduino Mega 2560 It has a USB port, a power connector, an ICSP header, 16 analogue inputs, a 16 MHz crystal oscillator, and reset buttons. It also has 54 digital pins for input and output (pin 15 may be used for PWM outputs). To begin, just use a USB cord to connect it to a computer, or power it with an AC-to-DC converter or battery.



Fig. 10 Arduino Mega 2560 and Stepper Driver TP 6600

- Stepper Motors Driver

The stepper motor driver allows for both direction and speed control. With six DIP switches, you may adjust the output current and micro-step. In all, there are 8 different types of current control (0.5A, 1A, 1.5A, 2A, 2.5A, 2.8A, 3.0A, 3.5A) and 7 different types of micro-steps (1, 2 / A, 2 / B, 4, 8, 16, 32). Additionally, all signal terminals use high-speed opt coupler isolation, which improves the device's capacity to block high-frequency interference.

B. Wiring Diagram

- Stepper Circuit Diagram

The connection between the joint Motors (Stepper Motors) is shown in Fig. 11 as it illustrates how the wires are connected between the Arduino Mega pins and Stepper Motors Ports to control the operation of the joints.

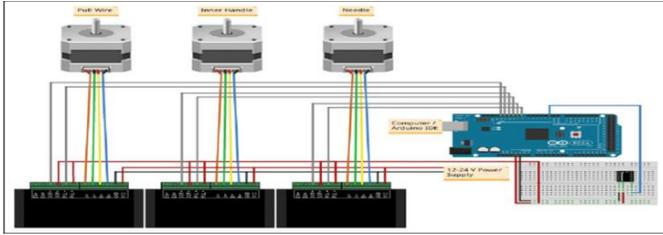


Fig. 11 Circuit diagram which illustrates connection between Arduino Mega and stepper motors

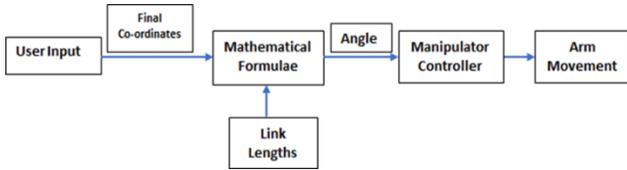


Fig. 12 Flowchart for forward kinematics

The user enters the object's coordinates as an input into the processing program while using the inverse kinematics mode. Regarding the base coordinate system, whose origin is set at the base of the robotic arm, these coordinates are defined. The angles that each motor must move by are calculated using the geometric method described in the section before this one. Once the microcontroller receives these angles as input, the arm is moved. In the form of a flowchart, and the Fig shows the full process.

IV. System Modelling

An arm robot manipulator is a set of links connected by joints, the lowest part is called the base, and the end part is called the end-effector. The number of joints and how it moves define the degree of freedom (DOF). The joints are characterized as revolute or prismatic joints. In this paper, the designed and constructed robot consists of all revolute joints with four degrees of freedom [18].

For the sake of simplicity, the kinematics and dynamics are derived for 2 DOF robot arm as shown in Fig. 13,

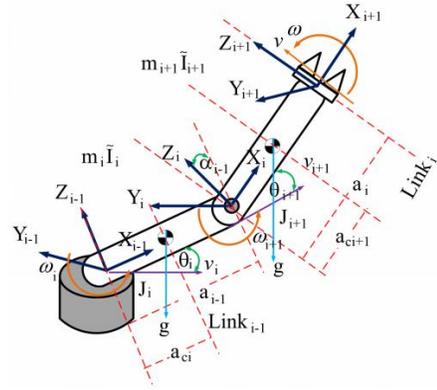


Fig. 13 DOF robotic manipulator with revolute joints

$X_{i-1}, X_i, X_{i+1}, Y_{i-1}, Y_i, Y_{i+1}, Z_{i-1}, Z_i, Z_{i+1}$  are manipulator coordinate frames.

$a_{i-1}, a_i, a_{i+1}$ , are the link length,  $\alpha_{i-1}, \alpha_i, \alpha_{i+1}$ , are the link twist,  $d_i$  is the link offset,  $\theta_i$  and  $\theta_{i+1}$  are the joint angles.

$J$  is robot's Jacobian matrix;  $J_i$  is link's Jacobian matrix.

$v_i$  is link's translational velocity and  $\omega_i$  is link's rotational velocity.

$m_i$  is link's mass.

$\tilde{I}_i$  is link's inertia

$a_{ci}$  is the length from link's center of gravity to link's end.

$g$  is gravitational force.

A. Forward Kinematics

The most applied method for obtaining the forward kinematics is the algebraic approach by using the Denavit-Hartenberg (DH) parameters. The distance from  $Z_{i-1}$  to  $Z_i$  measured along  $X_{i-1}$  is  $a_{i-1}$ , the angle between  $Z_{i-1}$  and  $Z_i$  measured along  $X_i$  is  $\alpha_{i-1}$ , the distance from  $X_{i-1}$  to  $X_i$  measured about  $Z_i$  is  $\theta_i$ . DH representation of Fig. is given in Table II.

One of the most commonly used methods is the Denavit-Hartenberg analysis, in which the direct kinematics are governed by certain parameters that must be set for each mechanism. The homogeneous transformation matrix, on the other hand, was chosen. Once one coordinate transformation between two frames is simply established, where the position and orientation are fixed one with respect to the other, it is possible to work with elementary homogeneous transformation procedures. D-H parameters are defined in Table for the allocated frames.

DH Parameters of the Arm

i	$\alpha_i$	$a_i$	$d_i$	$\theta_i$
1	90	0	L1	$\theta_1$
2	0	L2	0	$\theta_2$
3	0	L3	0	$\theta_3$
4	90	0	0	$\theta_4$

$${}_{i+1}^i T = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (8)$$

Therefore, the forward kinematics of n-DOF with n-joints is calculated from the base (0) of the robot to the end-effector (n) and given by the matrix [19],[20],

$${}^0_n T = {}^0_1 T \times {}^1_2 T \times \dots \times {}^{n-1}_n T. \quad (9)$$

Based on (2), the transformation matrix of 2 DOFs manipulator shown in Fig. can be expressed as,

$${}^0_1 T = \begin{bmatrix} c_1 & -s_1 & 0 & a_0 c_1 \\ s_1 & c_1 & 0 & a_0 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^1_2 T = \begin{bmatrix} c_2 & -s_2 & 0 & a_1 c_2 \\ s_2 & c_2 & 0 & a_1 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^0_2 T = \begin{bmatrix} c_{12} & -s_{12} & 0 & a_0 c_1 + a_1 c_{12} \\ s_{12} & c_{12} & 0 & a_0 s_1 + a_1 s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (10)$$

where,  $r_{k,j}$  is the rotational elements and  $k$  and  $j = 1,2,3$ .  $r_{k,j}$  is calculated using inverse kinematics. The maximum considered DOF is 6 DOF, otherwise, it is a redundant robot which kinematics will be different.

From (4), robot's position is

$$\begin{aligned} p_x &= a_0 c_1 + a_1 c_{12}, \\ p_y &= a_0 s_1 + a_1 s_{12}, \\ p_z &= 0. \end{aligned} \quad (11)$$

If all of the coordinate frames in fig. 1 are removed and only the base is considered, robot zero position is  $p_x = a_0 + a_1$ .

### B. Inverse Kinematics

There are two methods to transform a manipulator in Cartesian space into Joints space where the actuators work, namely, geometric and algebraic [21],[22]. For manipulator given in Fig. 1, the geometric solution is the easiest way to find inverse kinematics. In this approach, the manipulator is considered in 2D, therefore the considered positions are only  $p_x$  and  $p_y$ , therefore from (4), robot's position is

$$\begin{aligned} p_x &= a_1 c_1 + a_2 c_{12}, \\ p_y &= a_1 s_1 + a_2 s_{12}, \end{aligned} \quad (12)$$

By summing the square of  $p_x$  and  $p_y$ ,  $\theta_2$  is obtained as follow

$$p_x^2 + p_y^2 = a_0^2(c_1^2 + s_1^2) + a_1^2(c_{12}^2 + s_{12}^2) + 2a_0 a_1(c_1 c_{12} + s_1 s_{12}), \quad (13)$$

where based on trigonometric law,

$$c_{12} = c_1 c_2 - s_1 s_2, \quad s_{12} = s_1 c_2 + c_1 s_2, \quad \text{and} \quad c_1^2 + s_1^2 = 1.$$

Hence, (7) can be written as

$$p_x^2 + p_y^2 = a_0^2 + a_1^2 + 2a_0 a_1 c_2, \quad (14)$$

And from (14)

$$c_2 = \frac{p_x^2 + p_y^2 - a_0^2 - a_1^2}{2a_0 a_1}. \quad (15)$$

Since  $c_1^2 + s_1^2 = 1$ , then,

$$s_2 = \sqrt{1 - \left(\frac{p_x^2 + p_y^2 - a_0^2 - a_1^2}{2a_0 a_1}\right)^2}. \quad (16)$$

From (15) and (16), there are two possible solutions for  $\theta_2$ , that are

$$\theta_2 = \text{Atan2}\left(\pm \sqrt{1 - \left(\frac{p_x^2 + p_y^2 - a_0^2 - a_1^2}{2a_0 a_1}\right)^2}, \frac{p_x^2 + p_y^2 - a_0^2 - a_1^2}{2a_0 a_1}\right).$$

(17)

By revisiting and multiplying (13) with  $c_1$  and  $s_1$ ,

$$\begin{aligned} c_1 p_x + s_1 p_y &= a_0(c_1^2 + s_1^2) + a_1 c_2(c_1^2 + s_1^2), \\ -s_1 p_x + c_1 p_y &= a_1 s_2(c_1^2 + s_1^2), \\ c_1 p_x + s_1 p_y &= a_0 + a_1 c_2, \\ -s_1 p_x + c_1 p_y &= a_1 s_2. \end{aligned} \quad (18)$$

By combining (12),

$$c_1(p_x^2 + p_y^2) = p_x(a_0 + a_1 c_2) + p_y a_1 s_2.$$

Therefore,

$$c_1 = \frac{p_x(a_0 + a_1 c_2) + p_y a_1 s_2}{p_x^2 + p_y^2}. \quad (19)$$

$$s_1 = \sqrt{1 - \left(\frac{p_x(a_0 + a_1 c_2) + p_y a_1 s_2}{p_x^2 + p_y^2}\right)^2}. \quad (20)$$

Finally, two possible solutions for  $\theta_1$  are given by

$$\theta_1 = \text{Atan2}\left(\pm \sqrt{1 - \left(\frac{p_x(a_0 + a_1 c_2) + p_y a_1 s_2}{p_x^2 + p_y^2}\right)^2}, \frac{p_x(a_0 + a_1 c_2) + p_y a_1 s_2}{p_x^2 + p_y^2}\right).$$

(21)

Calculating inverse kinematics with the geometric method is very cumbersome, and more DOFs will result in a more cumbersome calculation. Another method is algebraic, by expanding the DH convention, however, for 2DOFs, it is easier with the geometric method.

The position of the end-effector shown in fig. 1 is given by eq. 5 and the relationship between joints and end-effector is given by

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \dot{q}. \quad (22)$$

where,  $v$  and  $\omega$  are the translational and rotational velocity of the end-effector,  $q$  is robot positions.

The objectives of Jacobian implementation is to define joint and workspace velocities, the applied forces and torques, and manipulability properties, and understand the singular configurations.

The first step is to define the orientation of rigid body which consists of the Euler angles  $(\phi, \theta, \psi)$  associated with  $z_0, y_1$  and  $z_2$ . If  $z$  axis is at the base frame considered parallel with the end-effector, and the end-effector position is given in (12).

The Euler rotation is,

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \theta_1 + \theta_2 \\ 0 \\ 0 \end{bmatrix}. \quad (23)$$

The Jacobian of a 2DOFs manipulator shown in Fig. 1 is

$$J = \begin{bmatrix} z_0 \times (p_2 - p_0) & z_1 \times (p_2 - p_1) \\ z_0 & z_1 \end{bmatrix}, \quad (24)$$

Where the frame origins are

$$p_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad p_1 = \begin{bmatrix} a_0 c_1 \\ a_0 s_1 \\ 0 \end{bmatrix}, \quad p_2 = \begin{bmatrix} a_0 c_1 + a_1 c_{12} \\ a_0 s_1 + a_1 s_{12} \\ 0 \end{bmatrix}, \quad (25)$$

And the rotational axes given by

$$z_0 = z_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \quad (26)$$

Therefore, the components of (19) are,

$$z_0 \times (p_2 - p_0) = \begin{bmatrix} 0 & 0 & a_0 s_1 + a_1 s_{12} \\ 0 & 0 & -a_0 c_1 - a_1 c_{12} \\ -a_0 s_1 - a_1 s_{12} & -a_0 c_1 + a_1 c_{12} & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

$$z_0 \times (p_2 - p_0) = \begin{bmatrix} -a_0 s_1 - a_1 s_{12} \\ a_0 c_1 + a_1 c_{12} \\ 0 \end{bmatrix},$$

$$z_1 \times (p_2 - p_1) = \begin{bmatrix} 0 & 0 & a_1 s_{12} \\ 0 & 0 & -a_1 c_{12} \\ a_1 s_{12} & a_1 c_{12} & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

$$z_1 \times (p_2 - p_1) = \begin{bmatrix} -a_1 s_{12} \\ a_1 c_{12} \\ 0 \end{bmatrix}. \quad (27)$$

Hence, the Jacobian matrix is

$$J(q)\dot{q} = \begin{bmatrix} -a_0 s_1 - a_1 s_{12} & -a_1 s_{12} \\ a_0 c_1 + a_1 c_{12} & a_1 c_{12} \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (28)$$

The translational and rotational velocities of end-effector in Fig. are

$$\begin{aligned} v &= J_{v1}\dot{q}_1 + J_{v2}\dot{q}_2, \\ \omega &= J_{\omega1}\dot{q}_1 + J_{\omega2}\dot{q}_2. \end{aligned} \quad (29)$$

where  $J_{v1} = -a_0 s_1 - a_1 s_{12}$ ,  $J_{v2} = -a_1 s_{12}$ ,  $J_{\omega1} = a_0 c_1 + a_1 c_{12}$ , and  $J_{\omega2} = a_1 c_{12}$ . The column in Jacobian matrix defines the effect of i-th joint on the end-effector velocities.

### C. Dynamics of the Robotic Manipulator

Dynamics represents the mathematical model that describes the motion of the robotic manipulator, which are the motions resulted from the torques applied by the actuators or other external forces applied to the manipulator. The analysis starts from a position and accordingly the vector of joint torques is calculated. The dynamics also presents the energy needed to move the system.

The generic robot dynamics is represented by,

$$\begin{aligned} M(q)\ddot{q} + C(q, \dot{q})\dot{q} + D\dot{q} + g(q) &= \tau, \\ I\ddot{\theta} + d\dot{\theta} + mgL \sin(\theta) &= \tau. \end{aligned} \quad (30)$$

Where  $M(q)$  is a  $n \times n$ , symmetric and positive definite mass matrix of arm robot manipulator,  $C(q, \dot{q})$  is a  $n \times 1$  vector  $v$ , a quadratic functions of the joint velocities,  $D\dot{q}$  is the friction,  $g(q)$  is the gravity term,  $I$  is the inertia matrix, and  $q$  is robot position.

The derivation of the dynamic is coming from kinematic, and from Jacobian in (26), the Jacobian of each robot's link in Fig. 1 are

$$J_v^1 = \begin{bmatrix} -a_1 s_{12} & 0 \\ a_1 c_{12} & 0 \\ 0 & 0 \end{bmatrix}, \quad J_v^2 = \begin{bmatrix} -a_0 s_1 - a_1 s_{12} & -a_1 s_{12} \\ a_0 c_1 + a_1 c_{12} & a_1 c_{12} \\ 0 & 0 \end{bmatrix}$$

$$J_\omega^1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad J_\omega^2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}, \quad (31)$$

Since the z-axes in the frame of link  $i$  and link  $i + 1$  are parallel to the axis of  $F_0$ , therefore  $\omega$  is considered the same with  $\omega_z$ .

Kinetic energy  $K$  of the robot in fig. 1 is

$$K = \frac{1}{2} \dot{q}^T [m_1 J_v^{1T} J_v^1 + m_2 J_v^{2T} J_v^2 + J_\omega^{1T} \tilde{I}_1 J_\omega^1 + J_\omega^{2T} \tilde{I}_2 J_\omega^2] \dot{q}. \quad (32)$$

where,

$$J_\omega^{1T} \tilde{I}_1 J_\omega^1 + J_\omega^{2T} \tilde{I}_2 J_\omega^2 = \begin{bmatrix} \tilde{I}_1 + \tilde{I}_2 & \tilde{I}_2 \\ \tilde{I}_2 & \tilde{I}_2 \end{bmatrix}. \quad (33)$$

By neglecting friction, the mass matrix  $M(q)$  from equation (30) is

$$M(q) = \begin{bmatrix} M_{11} & M_{21} \\ M_{12} & M_{22} \end{bmatrix}, \quad (34)$$

where,

$$\begin{aligned} M_{11} &= m_1 a_{c1}^2 + m_2 (a_0^2 + a_{c2}^2 + 2a_0 a_{c2} c_2) + \tilde{I}_1 + \tilde{I}_2, \\ M_{12} &= m_2 (a_{c2}^2 + a_0 a_{c2} c_2) + \tilde{I}_2, \\ M_{21} &= m_2 (a_{c2}^2 + a_0 a_{c2} c_2) + \tilde{I}_2, \\ M_{22} &= m_2 a_{c2}^2 + \tilde{I}_2. \end{aligned}$$

The quadratic functions of the joint velocities  $C(q, \dot{q})$  from (26) is,

$$C(q, \dot{q}) = \begin{bmatrix} h\dot{\theta}_1 & h(\dot{\theta}_1 + \dot{\theta}_2) \\ -h\dot{\theta}_1 & 0 \end{bmatrix}, \quad (35)$$

where,  $h$  is calculated from the Christoffel symbols  $c_{ijk} = \frac{1}{2} [\frac{\partial M_{kj}}{\partial q_i} + \frac{\partial M_{ki}}{\partial q_j} - \frac{\partial M_{ij}}{\partial q_k}]$  as follows,

$$\begin{aligned} c_{111} &= \frac{1}{2} \frac{\partial M_{11}}{\partial q_1} = 0, \\ c_{121} = c_{211} &= \frac{1}{2} \frac{\partial M_{11}}{\partial q_2} = -m_2 a_0 a_{c2} s_2 = h, \\ c_{221} &= \frac{\partial M_{12}}{\partial q_2} - \frac{1}{2} \frac{\partial M_{22}}{\partial q_1} = h, \\ c_{112} &= \frac{\partial M_{21}}{\partial q_1} - \frac{1}{2} \frac{\partial M_{11}}{\partial q_2} = -h, \\ c_{122} = c_{212} &= \frac{\partial M_{22}}{\partial q_1} = 0, \\ c_{222} &= \frac{\partial M_{22}}{\partial q_2} = 0. \end{aligned} \quad (36)$$

The gravitational forces working on the robot in Fig. is given by,

$$N(q) = \dot{M}(q) - 2N(q, \dot{q}),$$

$$\begin{aligned}
&= \begin{bmatrix} 2h\dot{\theta}_1 & h\dot{\theta}_2 \\ -\dot{\theta}_2 & 0 \end{bmatrix} - 2 \begin{bmatrix} h\dot{\theta}_1 & h(\dot{\theta}_1 + \dot{\theta}_2) \\ -h\dot{\theta}_1 & 0 \end{bmatrix}, \\
&= \begin{bmatrix} 0 & -2h\dot{\theta}_1 + h\dot{\theta}_2 \\ 2h\dot{\theta}_1 + h\dot{\theta}_2 & 0 \end{bmatrix},
\end{aligned} \tag{37}$$

which is a skew-symmetric.

Therefore, robot dynamic in (30) can be written as,

$$\begin{aligned}
M_{11}\ddot{\theta}_1 + M_{12}\ddot{\theta}_2 + c_{121}\dot{\theta}_1\dot{\theta}_2 + c_{211}\dot{\theta}_2\dot{\theta}_1 + c_{221}\dot{\theta}_2^2 + g_1 &= \tau_1, \\
M_{21}\ddot{\theta}_1 + M_{22}\ddot{\theta}_2 + c_{111}\dot{\theta}_1^2 + g_2 &= \tau_2
\end{aligned} \tag{38}$$

The potential energy of each link in Fig.,  $P_i$ , are

$$\begin{aligned}
P_1 &= m_1 g a_{c1} s_1, \\
\text{and} \\
P_2 &= m_2 g (a_0 s_1 + a_{c2} s_{12}).
\end{aligned} \tag{39}$$

Therefore, the potential energy P of the arm manipulator is

$$P = P_1 + P_2 = (m_1 a_{c1} + m_2 a_0) g s_1 + m_2 g a_{c2} c_{12} \tag{40}$$

where,

$$\begin{aligned}
g_1 &= \frac{\partial P}{\partial \theta_1} = (m_1 a_{c1} + m_2 a_0) g c_1 + m_2 g a_{c2} s_{12}, \\
g_2 &= \frac{\partial P}{\partial \theta_2} = m_2 g a_{c2} s_{12}.
\end{aligned}$$

The kinetic energy for each link,  $K_i$  are given by

$$\begin{aligned}
K_1 &= \frac{1}{2} m_1 a_{c1}^2 \dot{\theta}_1^2 + \frac{1}{2} \tilde{I}_1 \dot{\theta}_1^2, \\
K_2 &= \frac{1}{2} m_2 \dot{p}_{c2}^T \dot{p}_{c2} + \frac{1}{2} (\tilde{I}_1 + \tilde{I}_2)^2,
\end{aligned} \tag{41}$$

where,

$$\dot{p}_{c2}^T \dot{p}_{c2} = a_0^2 \dot{\theta}_1^2 + a_{c1}^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2a_0 a_{c2} c_2 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2).$$

## V. CONTROL ALGORITHM

The signal line is used to operate a servo motor by delivering a sequence of pulses. The control signal should have a frequency of 50Hz, or a pulse every 20ms. The angular location of the servo is determined by the width of the pulse, and these types of servos can normally spin 180 degrees (they have a physical limits of travel) [23].

Pulses of 1 millisecond length equate to 0 degrees, 1.5 milliseconds to 90 degrees, and 2 milliseconds to 180 degrees. Though the lowest and maximum durations of the pulses might vary between brands, they can be as little as 0.5ms for 0 degrees and as long as 2.5ms for 180 degrees. as shown in Fig. 14.

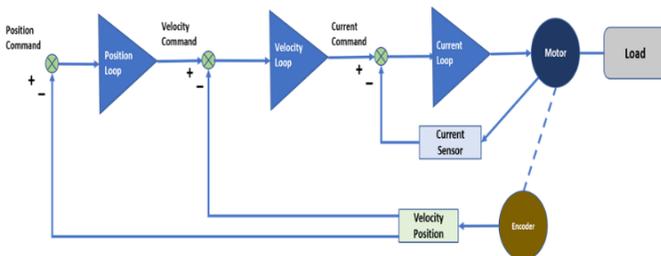


Fig. 14 Servo motor control loop diagram

An in-depth study into hobby servo motors will demonstrate how they function and how to control them with Arduino. A DC motor, a gearbox, a potentiometer, and a control circuit are the four major components of a hobby servo. The DC motor has a low torque and a high speed, but the gearbox decreases the speed to roughly 60 RPM while increasing the torque.



Fig. 15 Servo motor control loop structure

The potentiometer is coupled to the final gear or output shaft, so that when the motor spins, the potentiometer rotates as well, providing a voltage that is proportional to the output shaft's absolute angle. This potentiometer voltage is compared to the voltage coming from the signal line in the control circuit. The controller activates an inbuilt H-Bridge if necessary, allowing the motor to rotate in either direction until the two signals reach a zero difference, as well as connecting the Arduino ground to the servo ground.

*Pick and Place Control Algorithm*

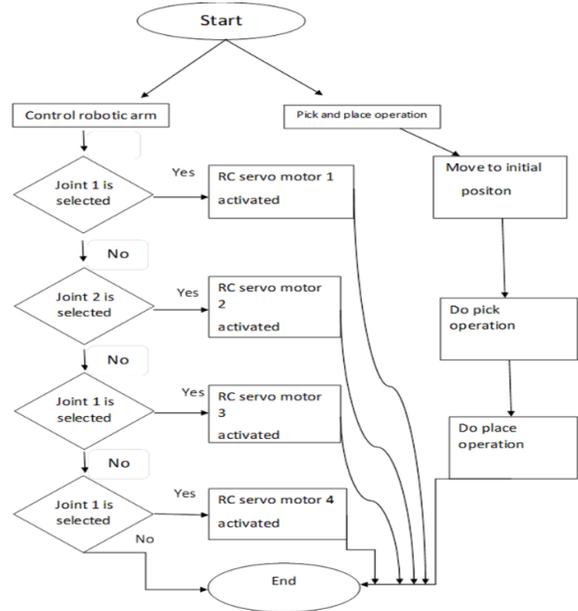


Fig. 16 Servo motor control loop diagram

Fig. 16 is a flowchart that explains the consequences of operation of the arm control system and the arm mechanism as a result of the orders set by the internet GUI site/page showing the selection of each joint followed by activation the motors and at last the pick and place operations [24], [25].

## VI. SIMULATION OF THE ROBOTIC ARM

A simulation of the movements of the arm with given goal point and positions is shown in Fig. 17 and Fig. 18, showing both the home position and the left-down position.

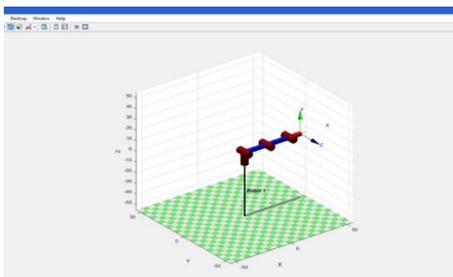


Fig. 17 Homing position

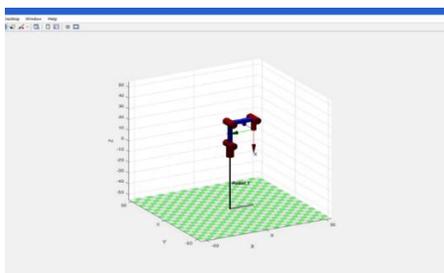


Fig. 18 Left-down position

## VI. MANUFACTURING AND ASSEMBLY

### A. Mechanical Assembly Steps

All the parts designed and constructed on SolidWorks are manufactured and then assembled together with the selected bearings and actuating motors. The assembled real robot can be seen in Fig. (). The motors are then connected with the drivers and the terminals of the Arduino Mega control board to get the control signals. All these hardware components is fed by the power supply as shown in Fig.



Fig. 19 The assembled robot

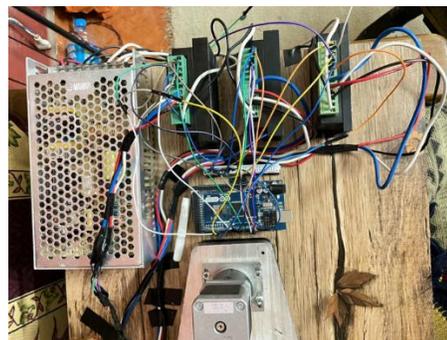


Fig. 20 The wiring diagram

## VII. CONCLUSION

Our goal was to design and control a pick and place robotic arm to move objects from place to another place. The design goal was to build small and strong robot which consists of 4 mechanical parts: link 1, link 2, link 3 and gripper.

It works with 3 stepper motors which provides fast movement with enough power and high torque that used for moving the joints in the working-space to our desired coordinates (location), also we used 2 servo motors for gripper one for open and close gripper and the other servo motor for gripper rotation.

Motors are connected to motors drivers which is powered by a (220V-24V) power supply and our microcontroller which is an Arduino Mega gives output signals to the drivers which in return rotates motors with step and direction to serve our interests.

We build an application to control the movement of our robot by reading the data we enter on our application and send it to microcontroller to execute our instructions and give the desired movement.

In the end a pick and place robotic has become one of the top priorities in the robotics future to increase accuracy of work, to reduce main effort and to reduce time consumption and with our main purpose safe people's health and integrity of body.

## ACKNOWLEDGMENT

We would like to express our gratitude to our supervisor Dr. Mahmoud Ahmed for his help and support. We appreciate the facilities provided by the Egyptian Academy for Engineering and Advanced Technology as the cooperation with the factories of military production that supported us by the manufacturing of the main components of the project.

## REFERENCES

- [1] K. E. Clothier and Y. Shang, "A Geometric Approach for Robotic Arm Kinematics with Hardware Design, Electrical Design, and Implementation," *J. Robot.*, 2010.
- [2] R. N. Jazar, *Theory of Applied Robotics*. Boston, MA: Springer US, 2010.
- [3] L. Barinka and R. Berka, "Inverse kinematics-basic methods," <http://www.cescg.org/CESCG 2002/LBarinka/paper.pdf>, 2002.
- [4] D. W. Hart, "Power Electronics-DC-DC Converter," Valparaiso University, Valparaiso, Indiana 2, pp.198-205, 2011.
- [5] N. Mohan, T. M. Undeland, and W. Robbins. *Power Electronics*, second edition, John Wiley and Sons, Inc., NY, USA, 1995.

- [6] M. Scarpino, *Motors for Makers: A Guide to Steppers, Servos, and Other Electrical Machines*. Pearson Education, 2016.
- [7] M. Ruzzenente, M. Koo, K. Nielsen, L. Grespan, and P. Fiorini, "A review of robotics kits for tertiary education," in *Proceedings of International Workshop Teaching Robotics Teaching with Robotics: Integrating Robotics in School Curriculum*, 2012, pp. 153-162.
- [8] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC press, 1994.
- [9] L. Žlajpah, "Simulation in robotics," *Math. Comput. Simul.*, vol. 79, no. 4, pp. 879-897, Dec. 2008.
- [10] P. I. Corke et al., "A computer tool for simulation and analysis: the Robotics Toolbox for MATLAB," in *Proc. National Conf. Australian Robot Association*, 1995, pp. 319-330.
- [11] P. I. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Berlin: Springer, 2011.
- [12] A. Petrillo, F.o De Felice, R. Cioffi and Federico Zomparelli, "Fourth Industrial Revolution: Current Practices, Challenges, and Opportunities", *Digital Transformation in Smart Manufacturing*, Dr. Antonella Petrillo (Ed.), InTech, 2018. DOI: 10.5772/intechopen.72304.
- [13] J. E. Agolla, "Human Capital in the Smart Manufacturing and Industry 4.0 Revolution", *Digital Transformation in Smart Manufacturing*, Dr. Antonella Petrillo (Ed.), InTech, 2018. DOI: 10.5772/intechopen.73575.
- [14] C. Chen and C. Peng (2010). "Coordinate Transformation Based Contour Following Control for Robotic Systems", *Advances in Robot Manipulators*, Ernest Hall (Ed.), InTech, pp. 183-204, 2010. DOI: 10.5772/9541.
- [15] M. Ceccarelli and E. Ottaviano, "Kinematic Design of Manipulators", *Robot Manipulators*, Marco Ceccarelli (Ed.), InTechopen, pp. 49-72, 2008. ISBN: 978-953-7619-06-0.
- [16] H. Lin, Y. Liu, and Y. Lin, "Intuitive Kinematic Control of a Robot Arm via Human Motion," in *Proceeding of 37th National Conference on Theoretical and Applied Mechanics (37th NCTAM 2013) & The 1st International Conference on Mechanics (1st ICM)*, *Procedia Engineering*, vol. 79, pp. 411-416, 2014.
- [17] A. B. Rehiara, "Kinematics of Adept Three Robot Arm", *Robot Arms*, Prof. Satoru Goto (Ed.), pp. 21-38, InTechopen, 2011, ISBN: 978-953-307-160-2.
- [18] P. Donelan, "Kinematic Singularities of Robot Manipulators", *Advances in Robot Manipulators*, Ernest Hall (Ed.), InTechopen, pp. 401-416, 2010. ISBN: 978-953-307-070-4.
- [19] W. Song, and G. Hu, "A Fast Inverse Kinematics Algorithm for Joint Animation", in *Proceeding of 2011 International Conference on Advances in Engineering*, *Procedia Engineering*, vol. 24, pp. 350-354, 2011.
- [20] T. Balkan, M. K. Ozgoren, and M. A. S. Arıkan, "Structure Based Classification and Kinematic Analysis of Six-Joint Industrial Robotic Manipulators", *Industrial Robotics: Theory, Modelling and Control*, Sam Cubero (Ed.), InTechOpen, pp. 149-184, 2006. ISBN: 3-86611-285-8
- [21] S. Zodey, and S. K. Pradhan, "Matlab Toolbox for Kinematic Analysis and Simulation of Dexterous Robotic Grippers", *Procedia Engineering*, Vol. 97, pp. 1886-1895, 2014. <https://doi.org/10.1016/j.proeng.2014.12.342>
- [22] S. Kucuk and Z. Bingul, "Robot Kinematics: Forward and Inverse Kinematics", *Industrial Robotics: Theory, Modelling, and Control*, Sam Cubero (Ed.), InTechopen, pp. 117-148, ISBN: 3-86611-285-8.
- [23] I. A. Sultan "Inverse Position Procedure for Manipulators with Rotary Joints," *Industrial Robotics: Theory, Modelling and Control*, Sam Cubero (Ed.), InTechopen, pp. 185-210, 2006. ISBN: 3-86611-285-8.
- [24] K. E. Clothier, and Y. Shang, "A Geometric Approach for Robotic Arm Kinematics with Hardware Design, Electrical Design, and Implementation", *Journal of Robotics*, vol. 2010, Article ID 984823, 10 pages, 2010. doi:10.1155/2010/984823.
- [25] A. Mohammed, B. Schmidt, L. Wang, and L. Gao, "Minimizing Energy Consumption for Robot Arm Movement", *Procedia CIRP*, Vol. 25, pp. 400-405, 2014. <https://doi.org/10.1016/j.procir.2014.10.055>.