MMC Project Report

Automatic Pick and Drop Bobotic Arm

Name of Students

Muhammad Abdullah Malik……….(21-ME-103)
Talal Azhar ………………………...(21-ME-179)
Yousaf Raza ………………………(21-ME-163)
Ali Hassan Khalid …………………(21-ME-115)
Rao Asad Jamil ……………………(21-ME-107)
Jamshaid …………………………..(21-ME-47)
Muzammil …………………………(21-ME-83)
Shafay ……………………………..(21-ME-31)
Jawad ………………………………(21-ME-15)
Musa ……………………………….(21-ME-11)
Atiq Ur Rehman ……………………(21-ME-131)
Umais ………………………………(21-ME-175)
Zeeshan Tanveer …………………...(21-ME-43)
Saad Umar …………………………(21-ME-55)
Fiaz ………………………………...(21-ME-19)

Objective

The Pick and Place Automatic Robot (PPAR) is an advanced robotic system designed for automating the process of picking objects from one location and placing them in another. By incorporating technologies such as computer vision, machine learning, and robotic arm manipulation, the PPAR aims to improve efficiency and productivity in industries such as manufacturing, logistics, and assembly. This abstract provides an overview of the PPAR's key components, including its sensors, robotic arm, grippers, and control system. It highlights the benefits of the PPAR, such as increased productivity, improved precision, and reduced human labor requirements. The PPAR represents a significant advancement in automation technology, offering potential forrevolutionizing various industrial applications.

Table of Contents

List of Figures

1 Introduction

The Pick and Place Automatic Robot (PPAR) revolutionizes object handling in industries like manufacturing, logistics, and assembly. It automates the labor-intensive task of picking and placing objects with high efficiency and accuracy using computer vision, machine learning, and a flexible robotic arm. The PPAR's advantages include increased productivity, error reduction, and improved product quality.

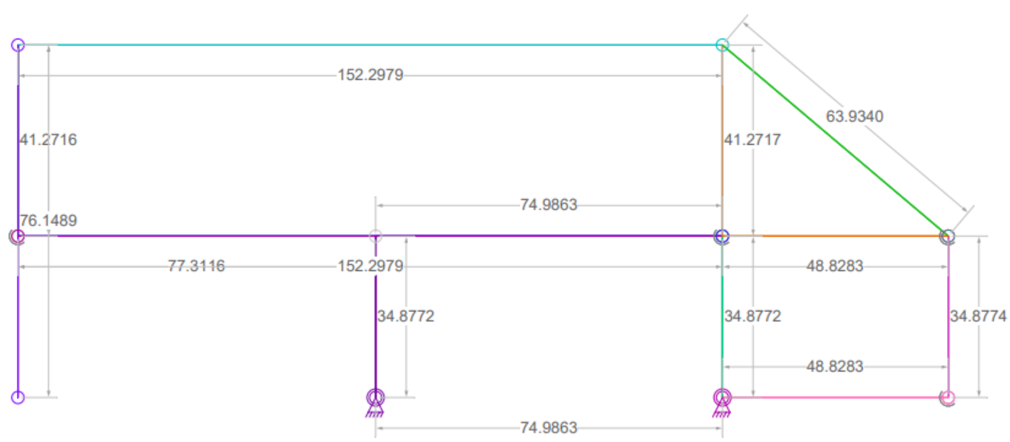
1.1 Aims

- Automated tasks.
- Boost productivity.
- Enhance precision.
- Reduce labor.
- Revolutionize industries.

1.2 Objective

The main objective of the Pick and Place Automatic Robot (PPAR) project is to revolutionize industrial operations by automating the precise picking and placing of objects. By leveraging cutting-edge technologies, such as computer vision and machine learning, the PPAR aims to enhance productivity, optimize processes, and improve overall efficiency in industries such as manufacturing, logistics, and assembly. Ultimately, the project seeks to redefine the way objects are handled, minimizing errors, reducing human intervention, and maximizing operational effectiveness.

2 Schematic Diagram



3 Material Specifications

The main components which are used in this (PPAR) are Servo Motors, Arduino Board (UNO), Acrylic plates, InfraRed (IR) Senor, nuts, and base plate. Their detail is given below with their specs as follows.

3.1 Servo Motor

In the Pick and Place Automatic Robot (PPAR) project, three SG90 servo motors are utilized. The SG90 servo motors are compact and lightweight, making them suitable for small-scale robotic application

Terms	**Values**
Weight	10g
Dimension	22.2x11.8x31 mm(approx.)
Torque	180 N mm
Shaft Diameter	4.5mm
Power	15W
Operating Voltage	4.8V
Rotation Speed	120 rpm

Table 1 SG90 Specifications

3.2 Arduino Board (UNO)

The Arduino Rev3 microcontroller board is used in the PPAR project. It is compact, versatile, and widely employed in robotics and electronics projects. Based on the ATmega328P microcontroller, it provides a reliable platform for controlling PPAR components. The Arduino Rev3 has digital and analog I/O pins for seamless integration with sensors and actuators. Its compatibility with Arduino programming language enables rapid prototyping. The compact size and robust capabilities make it ideal for efficiently powering and controlling the PPAR project.

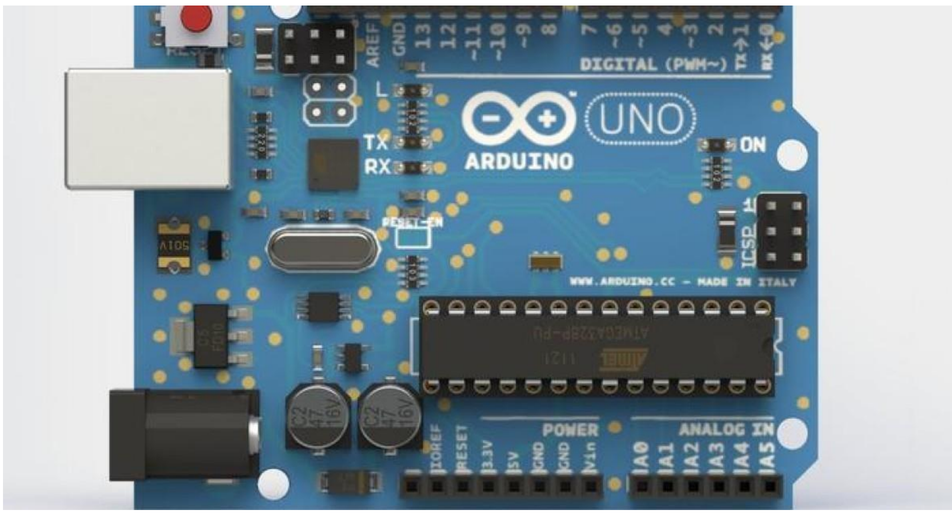


Figure 1: Arduino Board (UNO)

Terms	**Values**
Microcontroller	ATmega328P
PWM Digital I/O Pins	6
Analog input pins	6
Flash Memory	32 KB
Length x Width	68.6mm x 53.4mm
Weight	30 g
Operating Voltage	12 V

3.3 Acrylic Plates

The PPAR project uses 3mm acrylic plates for constructing the robot's arms and links. Acrylic is a lightweight and durable material known for transparency and impact resistance. The 3mm thickness provides strength while keeping weight manageable. Acrylic plates are cut and shaped according to design requirements, allowing easy assembly and structural stability for efficient object handling.

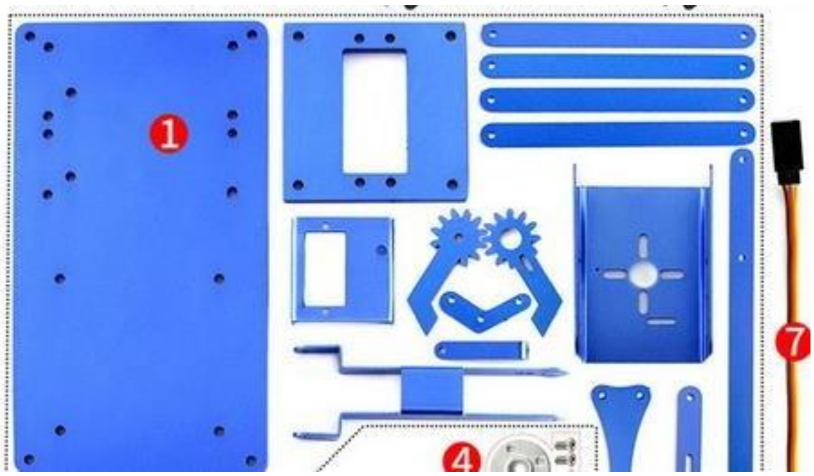


Figure 2 Acrylic Kit

Terms	**Values**
Max Tensile Stress	10,000 psi
Tensile Strength	69 N/mm^2
Weight	96 g
Elastic Strength	400,000 psi OR 2760 N/mm^2

3.4 Bluetooth Module

The IR sensor is vital in the PPAR project. It detects and measures infrared radiation emitted by nearby objects. By emitting an infrared beam and measuring its reflection or intensity, it enables proximity detection. This allows the robot to sense object presence or absence, aiding in accurate navigation during the picking and placing process. The IR sensor offers a reliable non-contact method of object detection, ensuring efficiency and precision in the PPAR system.

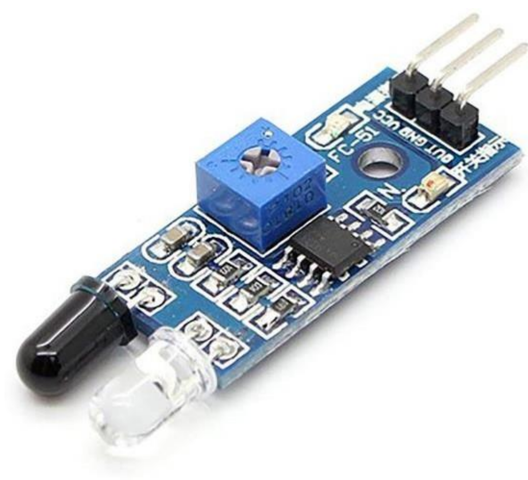


Figure 3: Infrared Senso

Terms	**Values**
Operating Voltage	12 V
Board size	120mm Length x 30mm Width
Detection Range	180 degrees
Active Output level	“O” (low) when an obstacle is detected

4 Steps taken in designing

Designing a robotic arm in SolidWorks involves several steps to ensure that the final model is accurate, functional, and manufacturable. Here's a step-by-step guide to help you through the process:

4.1 Define Requirements and Constraints:

- Determine the specific tasks and operations the robotic arm will perform.
- Consider payload capacity, reach, degrees of freedom, and end effector requirements.
- Identify any space or weight constraints.

4.2 Conceptual Design:

- Sketch different concepts for the robotic arm based on the defined requirements.
- Explore various configurations, joint types, and linkages.
- Consider ergonomics, efficiency, and ease of maintenance.

4.3 Create a SolidWorks Assembly:

- Start a new assembly file in SolidWorks.
- Import or create individual parts for each component of the robotic arm, such as links, joints, motors, and end effectors.
- Assemble the parts together according to the conceptual design.

4.4 Design Individual Components:

- Create detailed part models for each component of the robotic arm.
- Use features like extrusions, revolves, sweeps, and lofts to create the desired shapes.
- Pay attention to dimensions, clearances, and interface points between components.

4.5 Define Joints and Constraints:

- Define the joints and motion constraints between the different parts of the robotic arm.
- Use SolidWorks mates to simulate joints such as revolute, prismatic, and spherical.
- Ensure that the joints allow the desired range of motion for each link.

4.6 Test and Validate:

- Perform motion studies and simulations to test the functionality of the robotic arm.
- Verify that the arm can reach desired positions, carry expected loads, and operate within defined constraints.
- Iterate on the design as needed based on simulation results and feedback.

4.7 Finalize Design and Release:

- Complete any remaining revisions or refinements based on testing and feedback.
- Obtain final approvals from stakeholders.
- Prepare files for manufacturing, including part files, assembly files, and drawings.

5 SolidWorks Model

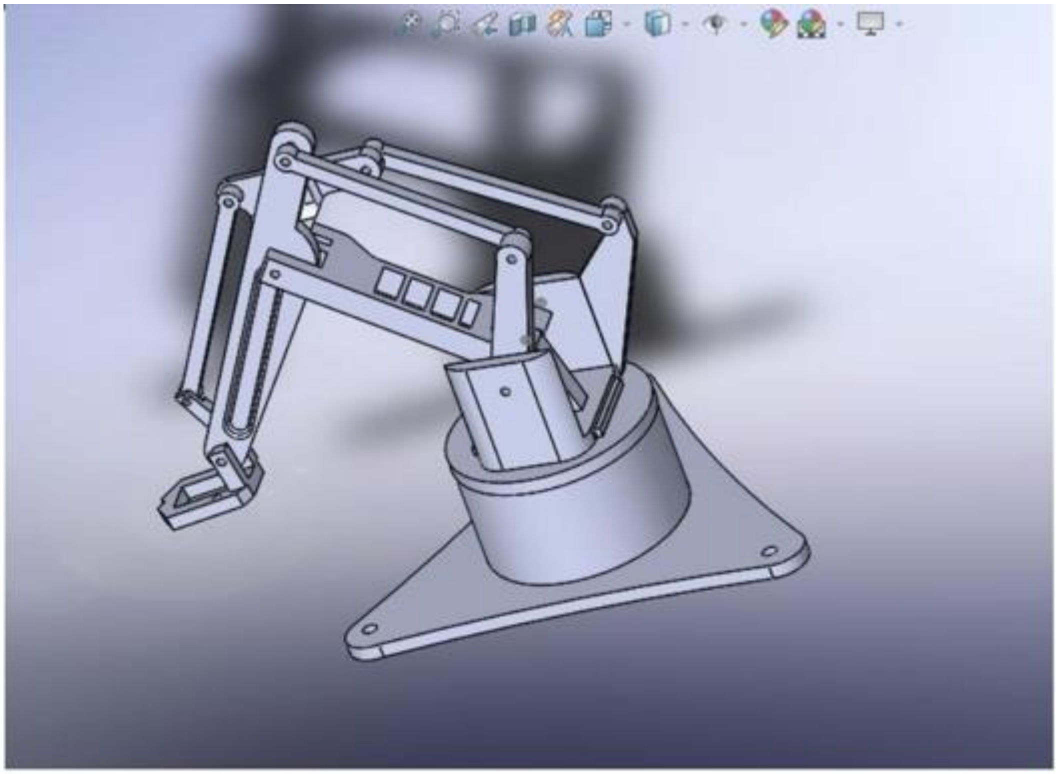


Figure 4: SolidWorks Mode

Data Collection

Total Mass of Acrylic = 100 g (4 plates)

Mass of each Plate = 25 g

Mass of screws $= \frac{382\ \text{g}}{1000\ \text{pieces}}$ (Avg of m_2 , m_3)

Mass of One screw $= 0.3829$

Motor	Denotations	Rotation (axis)
Base motor	M_1	z
Claw Motor	M_3	x
Shoulder motor	M	y

$W_{M1} = M_1$ mass

$W_{M2} = M_2$ mass

$W_{M3} = M_3$ mass

Weight of servo = 10 g

$M_1 \rightarrow$ Bears weight of 3 sheets of acrylic parts + 2 Servos + 50 Screw Pieces (Nuts/Bolts)

$$W_{M1} = 3(25) + 2(10) + 50(0.382)$$
$$\boldsymbol{W_{M1} = 114.1\ \mathrm{g}}$$

$M_2 \rightarrow$ Bears weight of Links and claws + 30 screws + 1 servo

$$W_{M2} = 1.75(25) + 30(0.382) + 10$$
$$\boldsymbol{W_{M2} = 65.21\ \mathrm{g}}$$

M_3 → Bears weight of Claw + 5 screws

$$W_{M3} = 0.5(25) + 5(.382)$$
$$W_{M3} = 14.41 \text{ g}$$

Acrylic material

$$\sigma_{tmax} = 10{,}000 \text{ psi or (Tensile strength)} = 69 \text{ N/mm}^2$$
$$E = 400{,}000 \text{ psi} = 2760 \text{ N/mm}^2$$

Servo SG90 Torque

$$\text{Torque} = 1.8 \text{ kgf cm} = 180 \text{ N mm}$$
$$\text{Power} = 15 \text{ W}, \text{ Shaft Dia} = 4.5 \text{ mm, weight} = 10 \text{ g}$$

Calculations for motor M_2

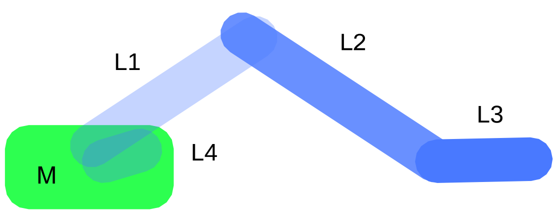


Figure 12: Motor 2 links

$$L_1 = 7.5 \text{ cm}, L_2 = 8.5 \text{ cm}, L_3 = 4 \text{ cm}, L_4 = 2 \text{ cm}$$
$$Total\ L = 220 \text{ mm}$$

Assumption

$$Max\ stretchable\ Length = Half\ of\ Total\ Length$$

$$\text{Length} = 110 \text{ mm}$$

Torque:

$$T_{max} = F * L * S \cdot F$$
$$F = \frac{T}{L * S \cdot F}$$
$$F = \frac{180}{110 * 1.6}$$
$$F = 104 \text{ N} = W$$

S.F - Safety Factor is 1.6

But $w = mg$

$$M_{max} = 104 \text{ g}$$

Maximum weight M_2 can lift = 104 g

Weight it is already lifting $W_{M2} = 65.21$ g

$$\text{Weight Lift} = M_{\max} - W_{M2}$$
$$\boldsymbol{m = 38.8 \text{ g}}$$

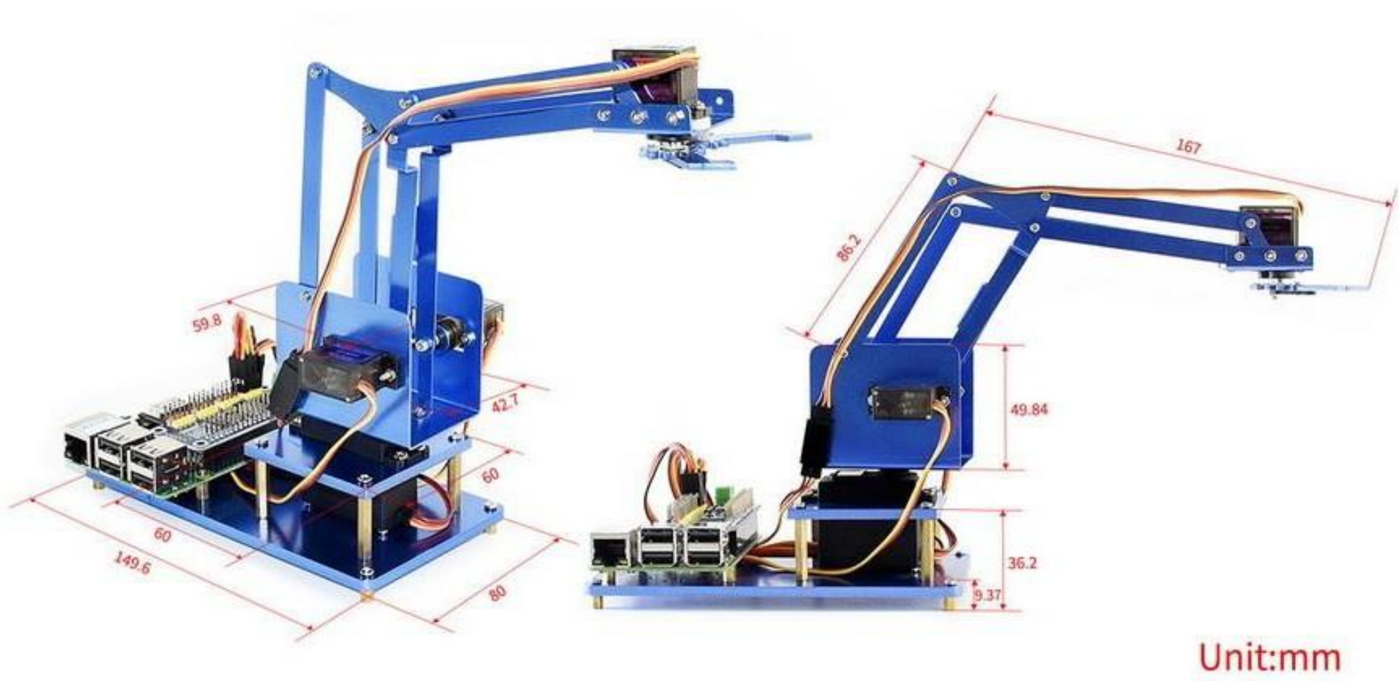


Figure 13: All Dimension

6 Kinematics Analysis

We considered the linkage shown below where we perform Kinematics Analysis on open-source software named *Linkage.*

Links	Length	Type
L1	153 mm	Ground
L2	86.2mm	Rocker
L3	127 mm	Coupler
L4	40 mm	Slider

Analysis:

Crank Slider Mechanism

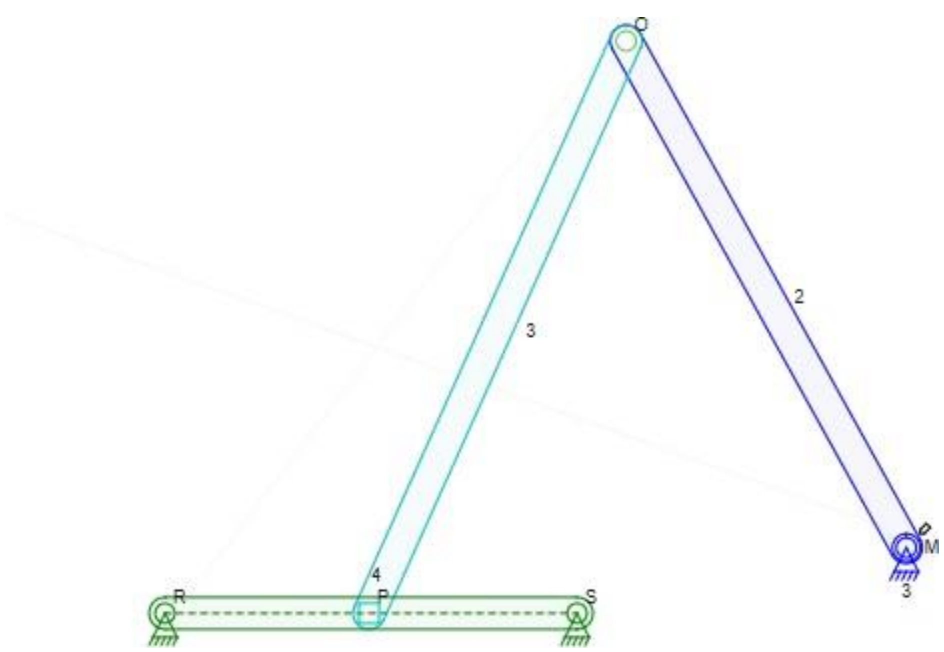


Figure 14 Crank Slider Mechanism

Grashoff Condition

Predicts rotation behavior or rotatability of 4-bar linkage's inversions, based only on link lengths.

Let:

S = length of shortest link = 40 mm

L= length of longest link = 153 mm

P= length of one remaining link = 86.2 mm

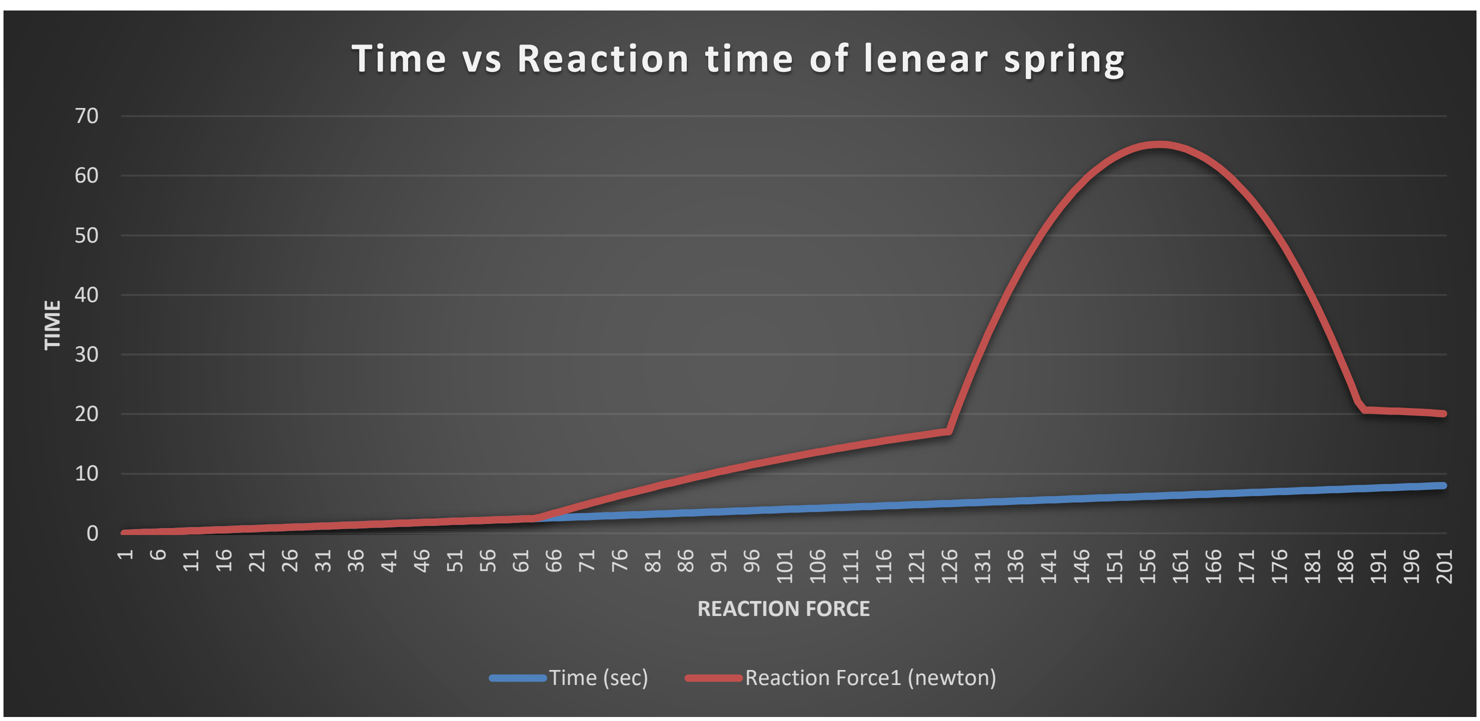
Q= length of another remaining link = 127 mm

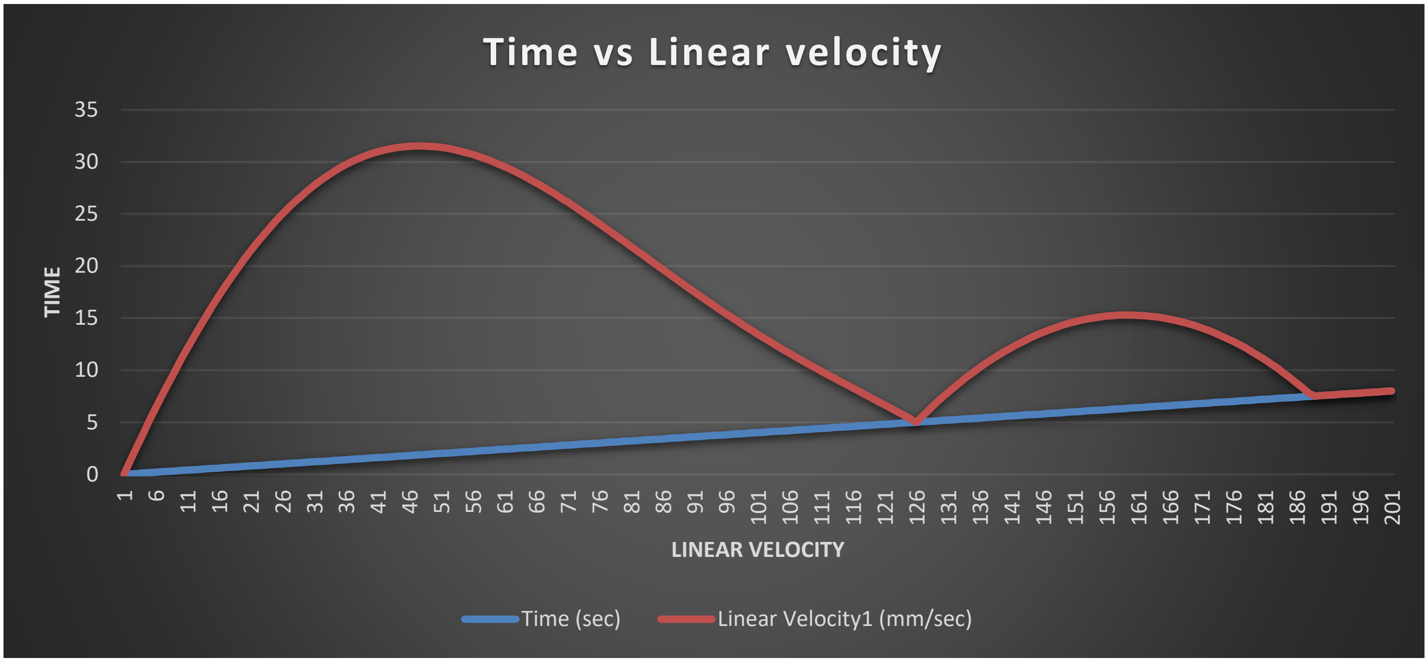
$$S + L \le P + Q$$

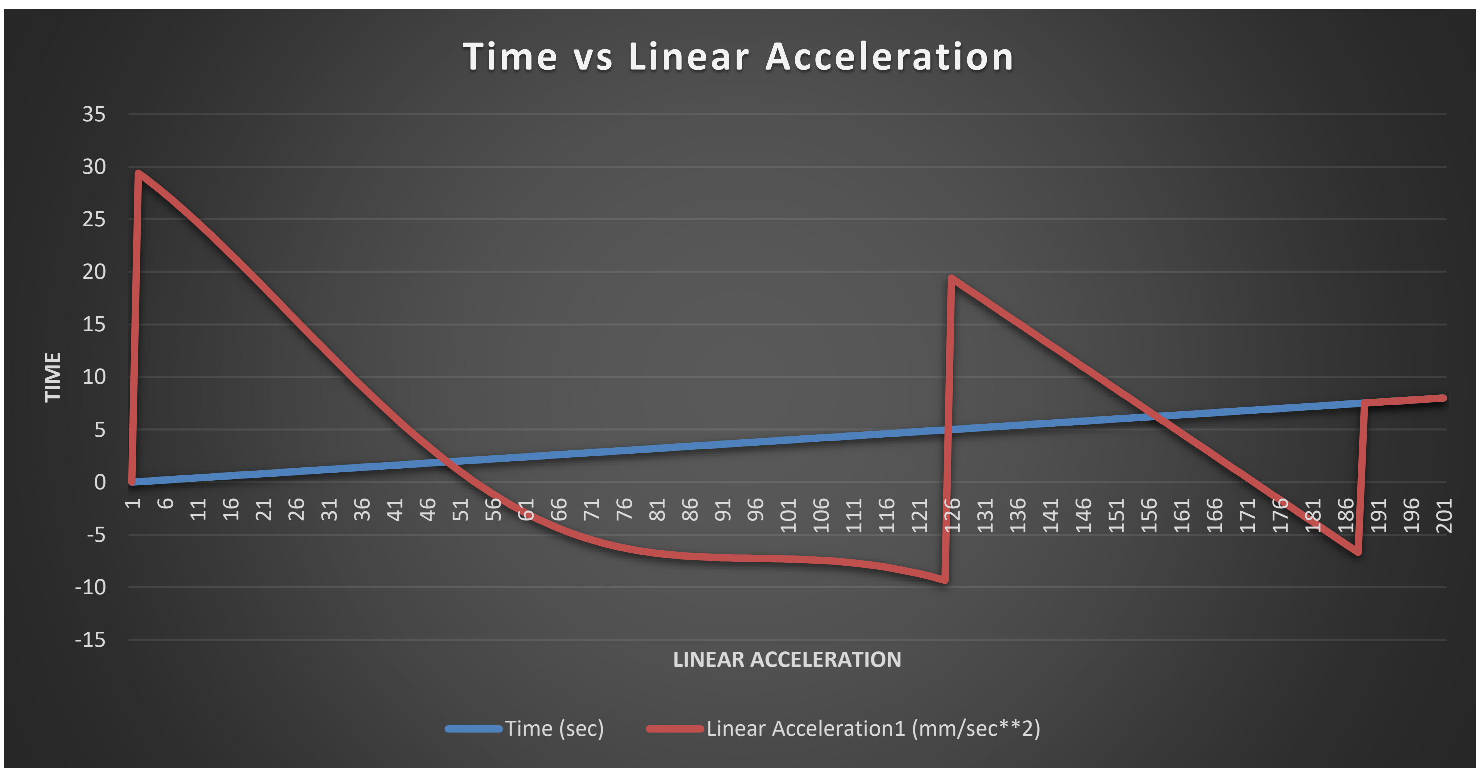
$$193 \le 213.2$$

Grashoff condition is satisfied.

7 Results




Time vs Linear velocity
TIME
35
30
25
20
15
10
5
0
1 6 11 16 21 26 31 36 41 46 51 56 61 66 71 76 81 86 91 96 101 106 111 116 121 126 131 136 141 146 151 156 161 166 171 176 181 186 191 196 201
LINEAR VELOCITY
Time (sec)
Linear Velocity1 (mm/sec)


Time vs Linear Acceleration
TIME
35
30
25
20
15
10
5
0
-5
-10
-15
1 6 11 16 21 26 31 36 41 46 51 56 61 66 71 76 81 86 91 96 101 106 111 116 121 126 131 136 141 146 151 156 161 166 171 176 181 186 191 196 201
LINEAR ACCELERATION
Time (sec)
Linear Acceleration1 (mm/sec**2)

8 Suggestions

8.1 Increase in DOF

We can Increase the DOF of project for better control and more mass lift. It will also provide smoother and more precise motion. This can be achieved by adding more input motors via the help of binary and tertiary links.

8.2 Load Capacity

The load capacity of the robot can be increased by replacing the normal SG90 Motors with high torque servo motors, especially ones with metallic gears.

8.3 High Strength

The overall strength of the project can be increased by replacing the acrylic material with a stronger version having better ultimate tensile strength and more effective properties.

8.4 Control

The autonomous control of robotic arm can be perfected by continuously tuning the IR sensor by both experimental and modelling means. In that way, it will sense an object more quickly at a specified range.

Reference

9 Extreme Position Pictures

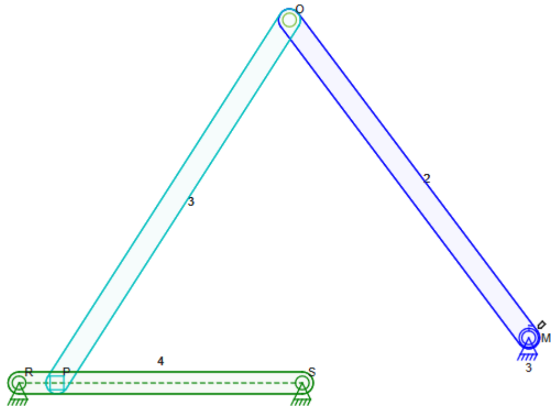


Figure 6: Extreme Left Position

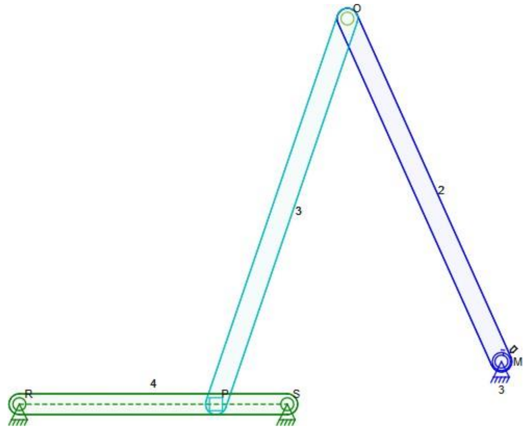


Figure 7: Extreme Right Position

10 DOF:

Link (L)	Joint (J)
5	4

$M = 3 (L - 1) - 2 J$

$M = 3 (5 – 1) – 2 (4)$

$M = 4 = DOF$

10.1 Claw Motor M_2

Top View

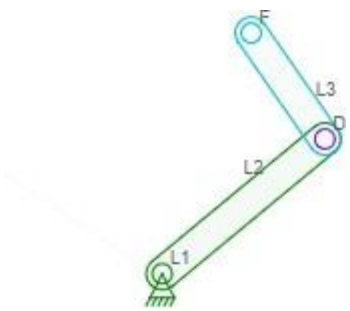
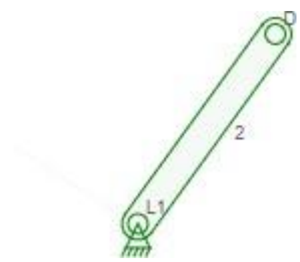


Figure 9: Claw Motor Top View

Since L3 is connected to L2 without any joint (see in real model) so it is part of the L2. So final diagram is



: Claw Motor Final Top View

10.2 Arm Motor M2

Crank-Slider Mechanism

Side View

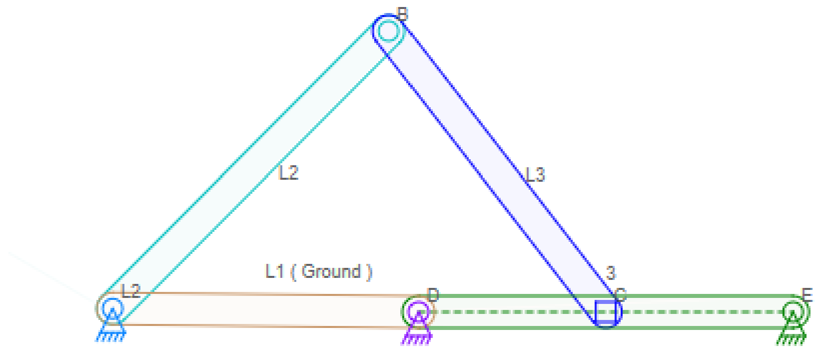


Figure 11: Arm Motor Side view

11 Arduino Code:

```
#include <SoftwareSerial.h>
#include <Servo.h>


#define SERVO_PIN_1 3   // waist
#define SERVO_PIN_2 6   // shoulder
#define SERVO_PIN_3 9   // ankle wrist
#define SERVO_PIN_4 10  // gripper  claw


Servo myservo1;
Servo myservo2;
Servo myservo3;
Servo myservo4;

char val = 0;

void setup() {
```

```
  Serial.begin(9600);
  myservo1.attach(SERVO_PIN_1, 500, 2500);
  myservo2.attach(SERVO_PIN_2, 500, 2500);
  myservo3.attach(SERVO_PIN_3, 500, 2500);
  myservo4.attach(SERVO_PIN_4, 500, 2500);
}

void loop() {
  if (Serial.available()) {
    val = Serial.read();

    if (val == 'a') {

      for (int i = 0; i < 50; i++) {
        myservo1.write(myservo1.read() - 1);
        delay(20);
        if (Serial.available() && Serial.read() == 's') {
          stopMotors();
          return;
        }
        val = 's';
      }
    } else if (val == 'b') {

      for (int i = 0; i < 50; i++) {
        myservo1.write(myservo1.read() + 1);
        delay(20);
        if (Serial.available() && Serial.read() == 's') {
          stopMotors();
          return;
        }
      }
    } else if (val == 'c') {

      for (int i = 0; i < 50; i++) {
        myservo2.write(myservo2.read() - 1);
        delay(20);
        if (Serial.available() && Serial.read() == 's') {
          stopMotors();
          return;
        }
      }
    } else if (val == 'd') {

      for (int i = 0; i < 50; i++) {
        myservo2.write(myservo2.read() + 1);
        delay(20);
        if (Serial.available() && Serial.read() == 's') {
```

```
      stopMotors();
      return;
    }
  }
} else if (val == 'e') {

  for (int i = 0; i < 50; i++) {
    myservo3.write(myservo3.read() - 1);
    delay(20);
    if (Serial.available() && Serial.read() == 's') {
      stopMotors();
      return;
    }
  }
} else if (val == 'f') {

  for (int i = 0; i < 50; i++) {
    myservo3.write(myservo3.read() + 1);
    delay(20);
    if (Serial.available() && Serial.read() == 's') {
      stopMotors();
      return;
    }
  }
} else if (val == 'g') {

  for (int i = 0; i < 50; i++) {
    myservo4.write(myservo4.read() - 1);
    delay(20);
    if (Serial.available() && Serial.read() == 's') {
      stopMotors();
      return;
    }
  }
} else if (val == 'h') {

  for (int i = 0; i < 50; i++) {
    myservo4.write(myservo4.read() + 1);
    delay(20);
    if (Serial.available() && Serial.read() == 's') {
      stopMotors();
      return;
    }
  }
} else if (val == 's') {
  stopMotors();
}
```

```
    Serial.println(val);
  }
}
void stopMotors() {
  myservo1.write(myservo1.read());
  myservo2.write(myservo2.read());
  myservo3.write(myservo3.read());
  myservo4.write(myservo4.read());
```

11.1 MATLAB:

MATLAB Code for graph

```
% Define linkage geometry (link lengths)
L1 = 153; % length of link 1
L2 = 86.2; % length of link 2
L3 = 127; % length of link 3
L4 = 40; % length of link 4

% Define joint angles
theta2 = linspace(0, pi/12, 100); % range of theta2 values

% Solve kinematic equations
x = L2*cos(theta2) + sqrt(L1^2 - L2^2*sin(theta2).^2);
y = L2*sin(theta2) + L3 - sqrt(L1^2 - L2^2*sin(theta2).^2);

% Plot linkage positions
figure;
plot(x, y);
xlabel('x');
ylabel('y');
title('Four-Bar Linkage Kinematic Analysis');
```

11.2 Graph

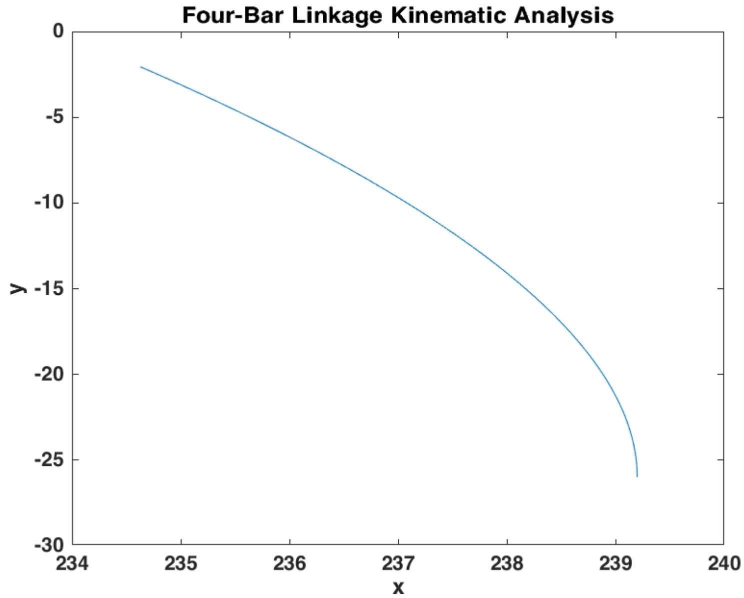


Figure 15: 4 bar linkage Kinematic Analysis Graph

12 Cost Analysis

Items	Quantity	Price
Arduino	01	2100
Charger	01	350
Electronics	-	630
Acrylic Kit	01	3000
Cardboard	-	150
Total Cost of Project		**6130**

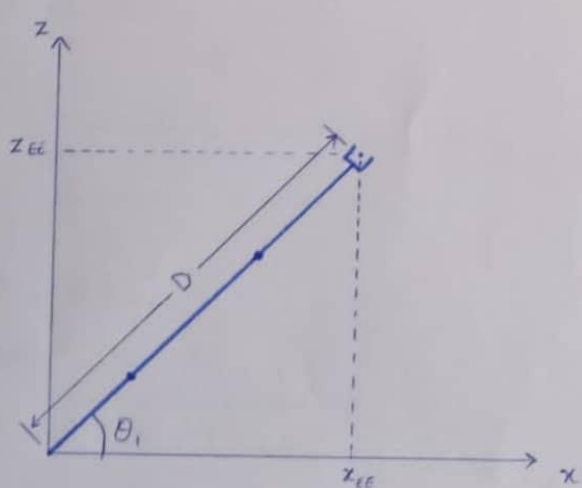
Inverse Kinematics:

Inverse Kinematics :-

We determined the path or trajectory of robot end effector where to move, then calculated how many angles values each joint should rotate, because when programming a robot we need to send 'signal' from controller board to actuator, the signal contain a value of information how much angle, servo motor rotate that drive the robot joints, hence why we need Inverse Kinematics

We used geometric method to determine the joint angles of robot arm to reach desired end effector position.

lets assume we look at robotic arm from above,



$$D = \sqrt{(x_{EE})^2 + (z_{EE})^2}$$

$$\frac{z_{EE}}{x_{EE}} = \tan \theta_1$$

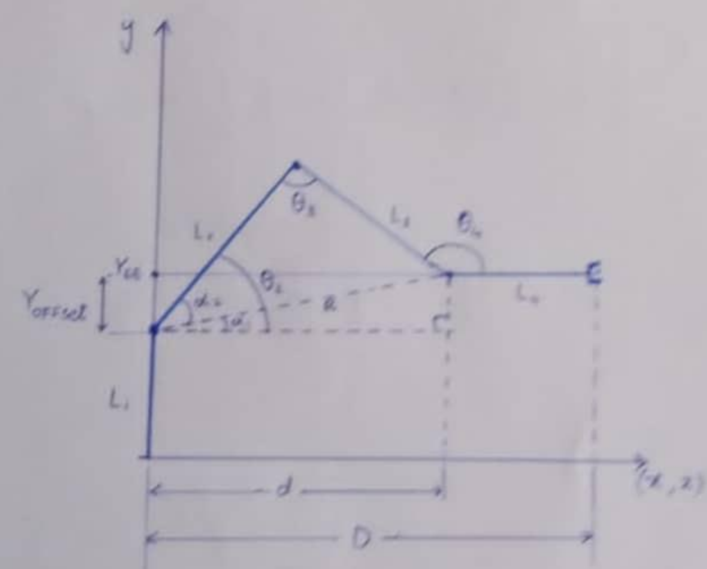
$$\theta_1 = \tan^{-1}\left(\frac{z_{EE}}{x_{EE}}\right)$$

Now to find other three angles & joint rotation,

We will study 2 cases.

Case 1 :

$Y_{EE} \geq L_1$



$$d = D - L_4$$

$$Y_{offset} = Y_{EE} - L_1$$

$$R = \sqrt{(d)^2 + (Y_{offset})^2}$$

$$\frac{d}{R} = \cos \alpha_1$$

$$\alpha_1 = \cos^{-1} \frac{d}{R}$$

Apply cosine law

$$(L_3)^2 = (L_2)^2 + R^2 - 2 \times L_2 \times R \cos\alpha_2$$

$$\alpha_2 = \cos^{-1}\left(\frac{L_2^2 + R^2 - L_3^2}{2 L_2 R}\right)$$

$$\theta_2 = \alpha_1 + \alpha_2$$

Similarly for θ_3

$$R^2 = L_2^2 + L_3^2 - 2 \times L_2 L_3 \times \cos\theta_3$$

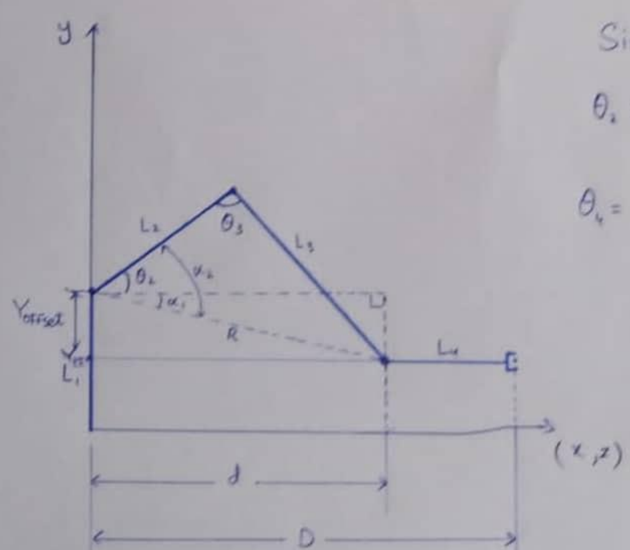
$$\theta_3 = \cos^{-1}\left(\frac{L_2^2 + L_3^2 - R^2}{2 L_2 L_3}\right)$$

for θ_4

$$\theta_4 = 180^\circ - \left\{ \left[180^\circ - (\alpha_2 + \theta_3) \right] - \alpha_1 \right\}$$

Case 2 :-

$$Y_{EE} \leq L_1$$



Similarly

$$\theta_2 = \alpha_2 - \alpha_1$$

and

$$\theta_4 = 180^\circ - \left[\left\{ 180^\circ - (\alpha_2 + \theta_3) \right\} + \alpha_1 \right]$$

Mechanical Advantage:

link $= l = 40$ mm.
Connecting rod $= 90$ mm
M.A $= 2.5$

Weight lifted = M.A x Force input.
$= 2.5 \times 0.38418$
$= 0.9604$ N-m.

$W = mg$.
$m = \frac{W}{g} = \frac{0.9604}{9.8} = 0.098 = 98g$.

For Elbow:

Rocker length = 120 mm
Crank = 45 mm.
Connecting rod = 50 mm

$M.A = \frac{45 + 120}{50}$

$= 3.3$

Weight lifted $= 3.3 \times 0.9604$
$= 1.267$

$m = W/g = \frac{1.267}{9.8} = 0.129 = 129\,g$

Hence it can lift weight up to 150g.