Misr University of Science and Technology

College of Engineering and Technology

Department of Mechatronics Engineering

MTE599A - Senior MTE project 1

Artistic Robot

By *Group 9:*

NAME	ID
Mohamed Hazem	75075
Ola Khaled	75205
Salwa Noaman	75201
Kareem Sameh	75293
Yousef Emad	75041

Supervised By:

Prof. Dr. Eng. Mohamed Hamdy

Prof. Dr. Eng. Mohamed Ibrahim

Prof. Dr. Eng. Ramadan El Soeudy

Prof. Dr. Eng. Bahaa M. Nasser

TABLE OF CONTENTS

Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Explanation
DOF	Degree of freedom
SFD	Shear force diagram
BDM	Bending moment diagram
FBD	Free body diagram
FOS	Factor of safety
3D	Three Dimensions
FDM	Fused deposition modelling
PLA	Polylactide
D-H	Denavit-Hartenberg

DECLARATION

I hereby certify that this material, which I now submit for assessment on the program of study leading to the award of Bachelor of Science in Mechatronics Engineering is entirely our own work, that we have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others and to the extent that such work, if any, has been cited and acknowledged within the text of my work.

Signed: ______________ *Mohamed Hazem Abdulkader* _______________

Date: *30, December 2021*

Signed: ______________ *Ola Khaled Elgibaly* ______________________

Date: *30, December 2021*

Signed: ______________ *Salwa Nouman AlBakri* ____________________

Date: *30, December 2021*

Signed: ______________ *Kareem Sameh Galal* ______________________

Date: *30, December 2021*

Signed: ______________ *Youssef Emad Eldien Dirar* _________________

Date: *30, December 2021*

ABSTRACT

In this paper we present an innovative robotic painting system for rendering digital images into watercolor artworks. The installation is composed of a 4-degrees of freedom articulated robot and a series of image processing and path planning algorithms. The Pointillism technique elaborates a digital image into a sequence of points that the robot reproduces on a canvas.

INTRODUCTION

Robotic painting is a challenging task that is motivated by an inner wish to discover novel forms of art and to experiment the technological advances to create something that can be aesthetically appreciated. Developing an automatic robotic painting system is hard, since the process comprises several different fields, including robotics, automation, image processing and art.

Busker Robot, which name refers to street artists, has been previously presented **[1,2]** and has been showcased for the first time in 2016 at the exhibition “Art, Science, Technology, Robotics” in Trieste, Italy. Then, it was shown at the SPS IPC Drives Italy 2017, at the “Algorithmic Arts and Robotics” exhibition during the international event Trieste Next 2017, at “Piccolo Teatro” in Milan, 2017, and, more recently, at the international festival “Robotics” in Trieste, 2018. **[3]** Furthermore, it took part to the 2018 International Robotic Art Competition (Robot Art) **[4]**, a context where 19 teams from all over the world competed and more than 100 artworks were created by painting robots; Busker Robot won an Honorable Mention.

PROJECT PURPOSE AND CONSTRAINTS:

A 4 degree of freedom (DOF) articulated arm that takes a photo and then paints the image on the canvas, making an industrial-styled machine that resembles the more organic form of a traditional portrait painter using a canvas.

PROJECT CONCEPT:

- Can fit on a desk.
- The arm has 4 (DOF) which is the minimum requirement for painting.
- 40x40 mm paint canvas (Drawing area) and it is removable.
- The end effector of the arm accepts an ordinary paint brush (That is approximately 150 grams).
- paints a photo.
- retrieves paint from a palette.
- cleans itself off in a water cup.

AIM AND OBJECTIVES:

The aim of this project is to design a prototype of a Canvas paining robot arm; that acts like a human artist and paints using path and motion planning with no obstruction using a painting brush.

And the main objectives are to:

- Design a robotic system that is composed of a 4DOF robotic arm and a series of image processing and path planning algorithms that are capable of interpreting, an input digital image into a real artwork.

- Not to faithfully reproduce an image, as typical printers or plotters do, but to introduce an original contribution.

- Image processing and path planning algorithms that are capable of interpreting, an input digital image into a real artwork.

CHAPTER 1: DESIGN AND DRAWINGS

Robotic arm system often consists of links, joints, actuators, sensors, and controllers. The links are connected by joints to form an open kinematic chain. One end of the chain is attached to the robot base, and another is equipped with a tool (A painting brush in our case), which is used to paint on a canvas.

The links of the manipulator are connected by rotary joints allowing rotational motion and is forming a kinematic chain.

A robotic arm with only four degrees of freedom is designed because it is adequate for most of the necessary movement. At the same time, it is competitive by its complexity and cost-saving as number of actuators in the robotic arm increase with degrees of freedom.

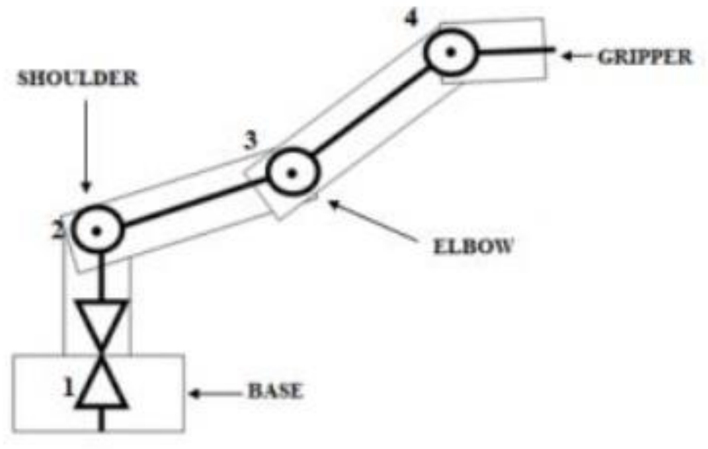


Figure 1 Free body diagram of the robotic arm

• 3D MODEL USING SOLIDWORKS:

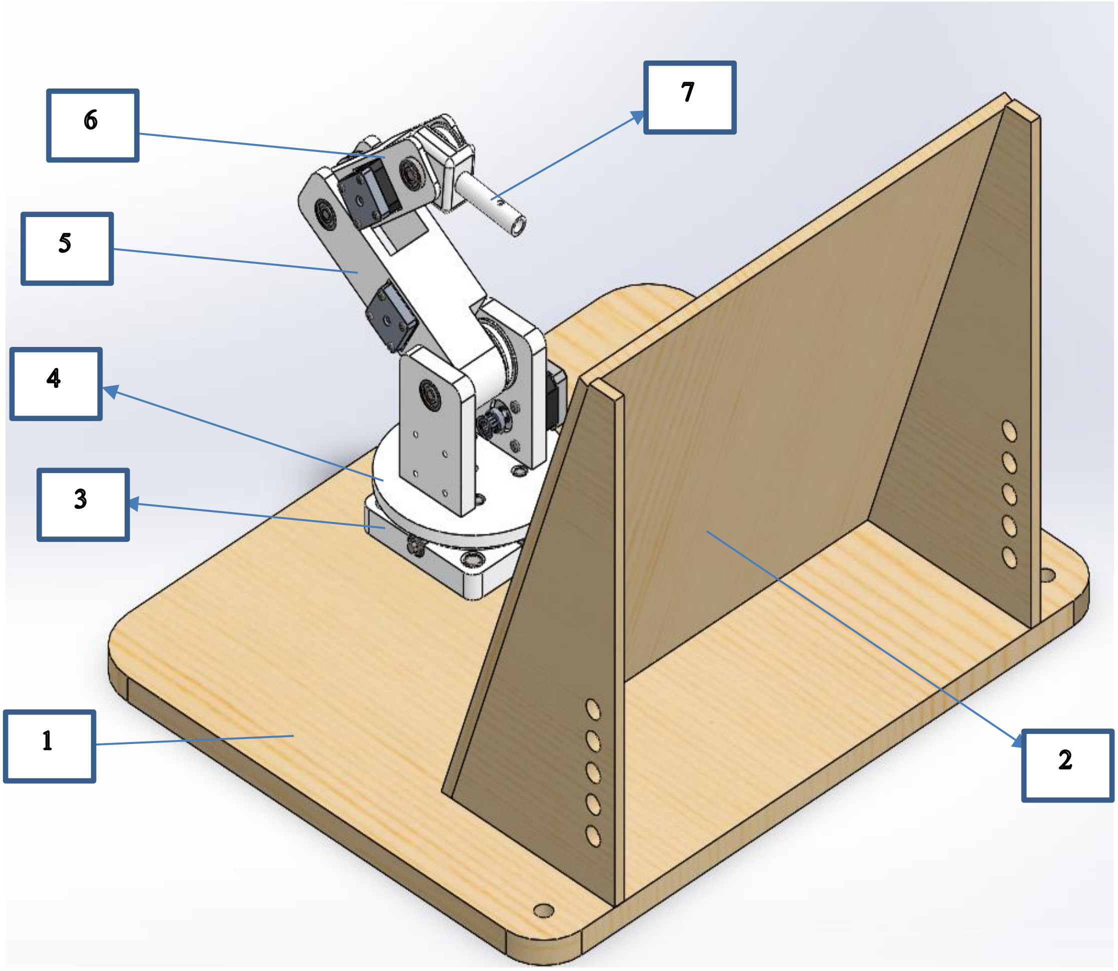


Figure 2 Solid works design

1	table
2	Canvas holder
3	base
4	Link 1 holder
5	Link 1
6	Link 2
7	End effector

Table 1 Parts

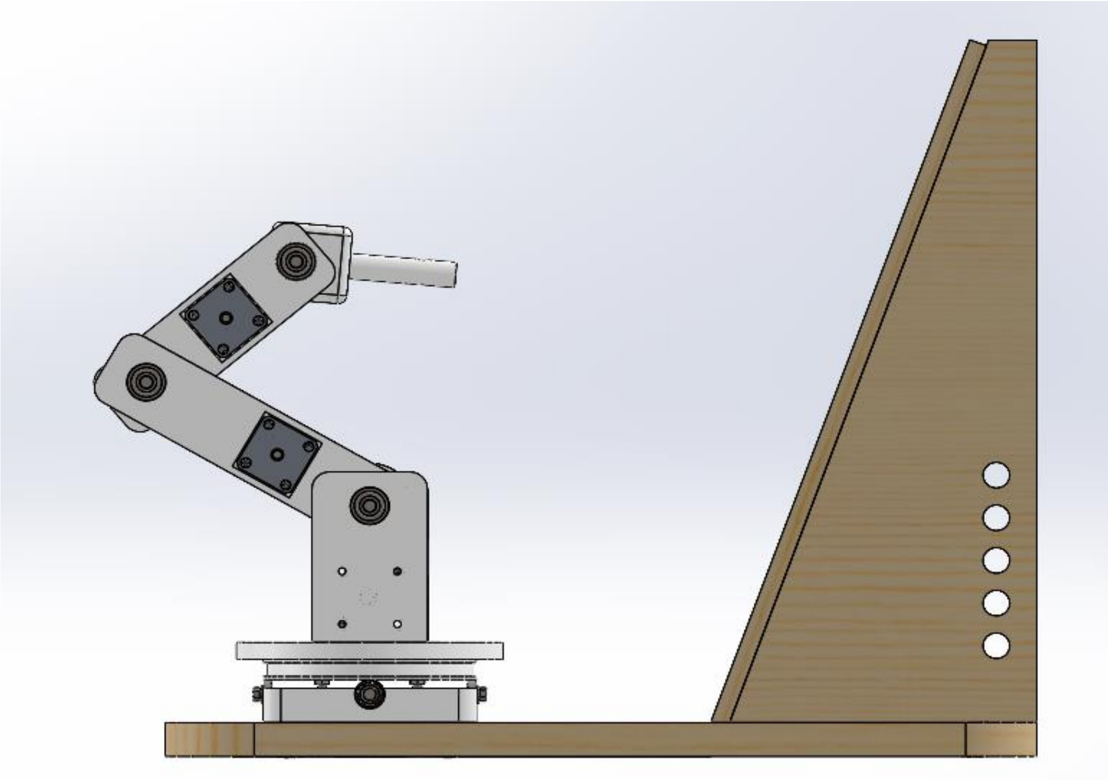


Figure 3 Side view

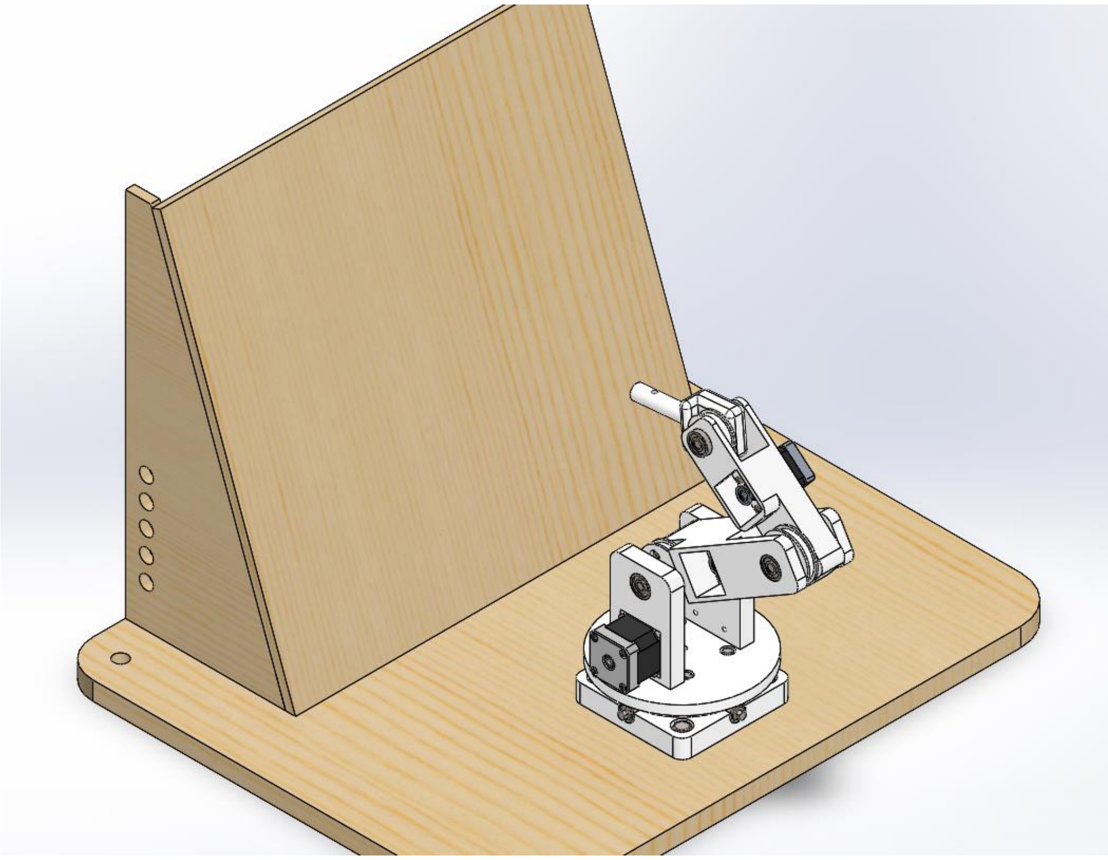


Figure 4 Isometric

2D DRAWINGS OF THE DESIGN :

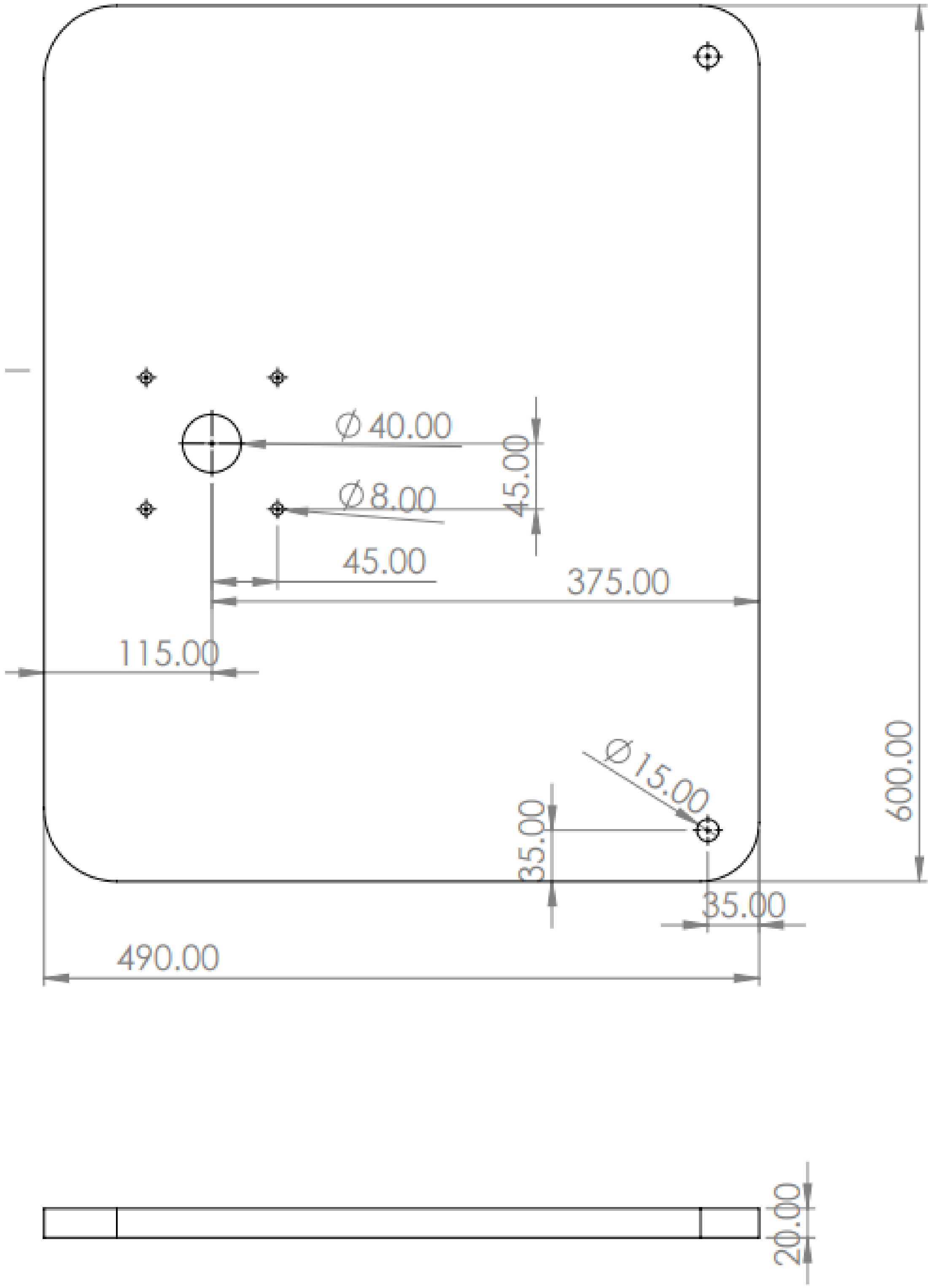


Figure 5 Mounting Table

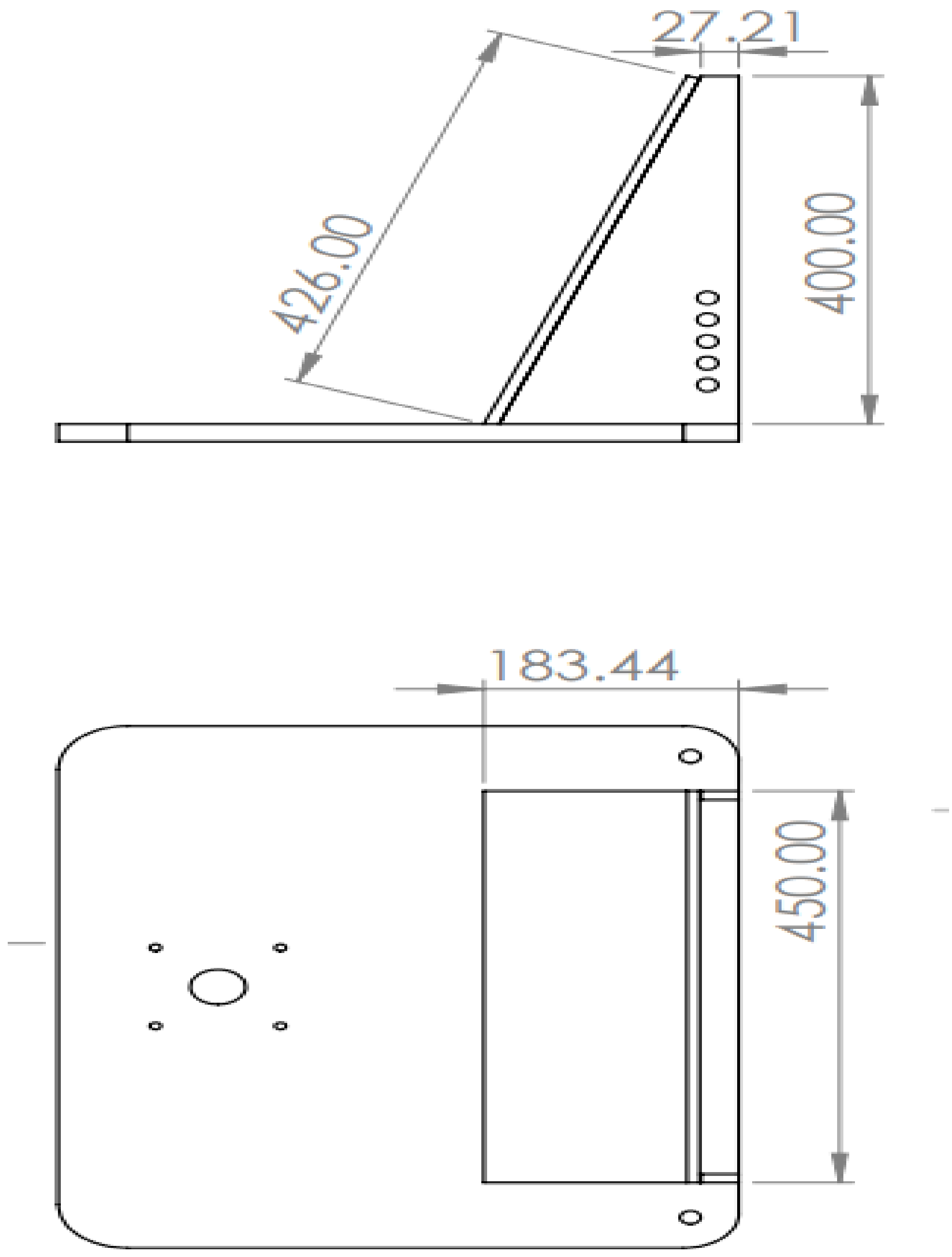


Figure 6 canvas holder and table

Part	table
Part material	PLA filament
Dimension	(mm)
Tolerance	± 0.01

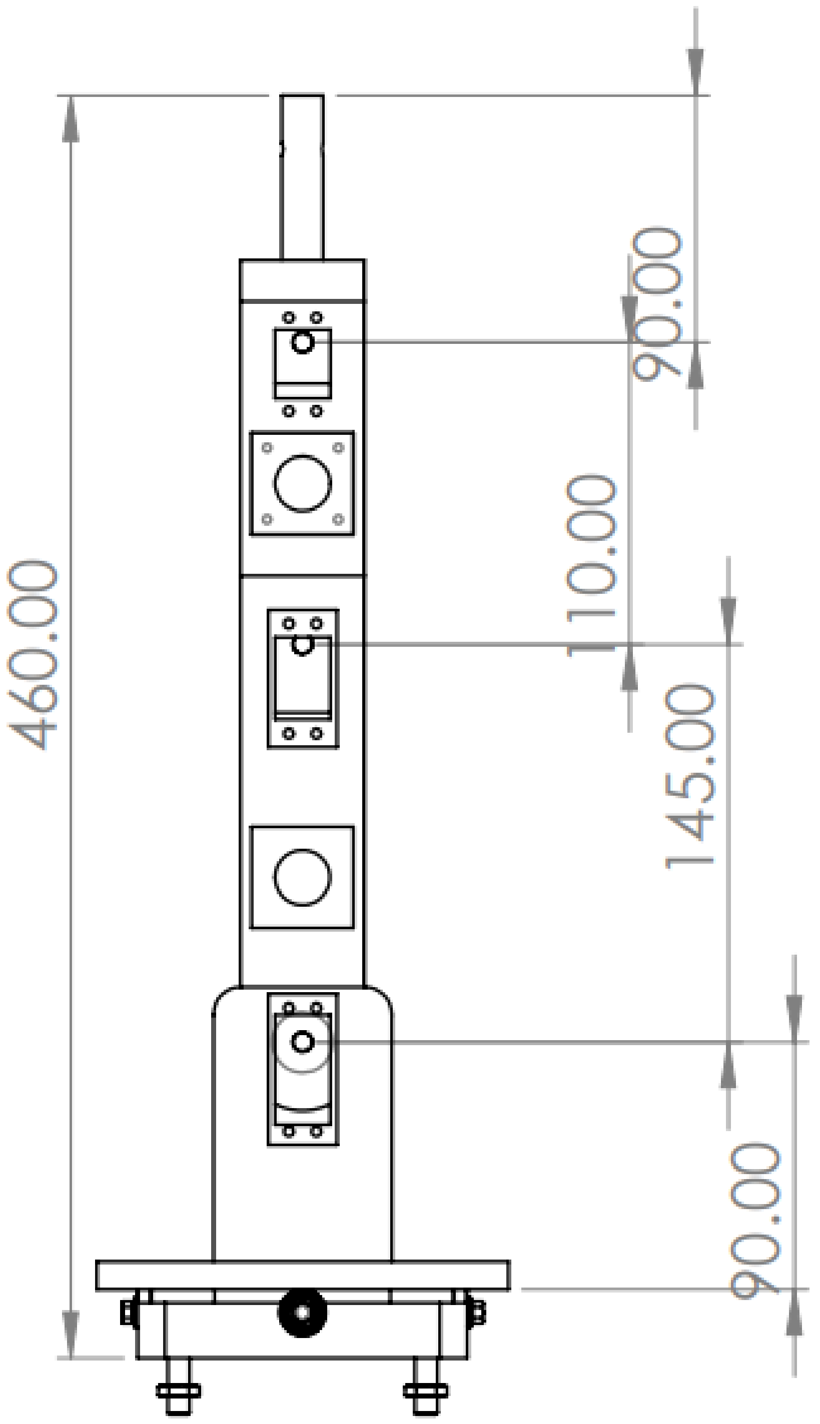


Figure 7 4DOF robot arm with dimensions of links

Base:

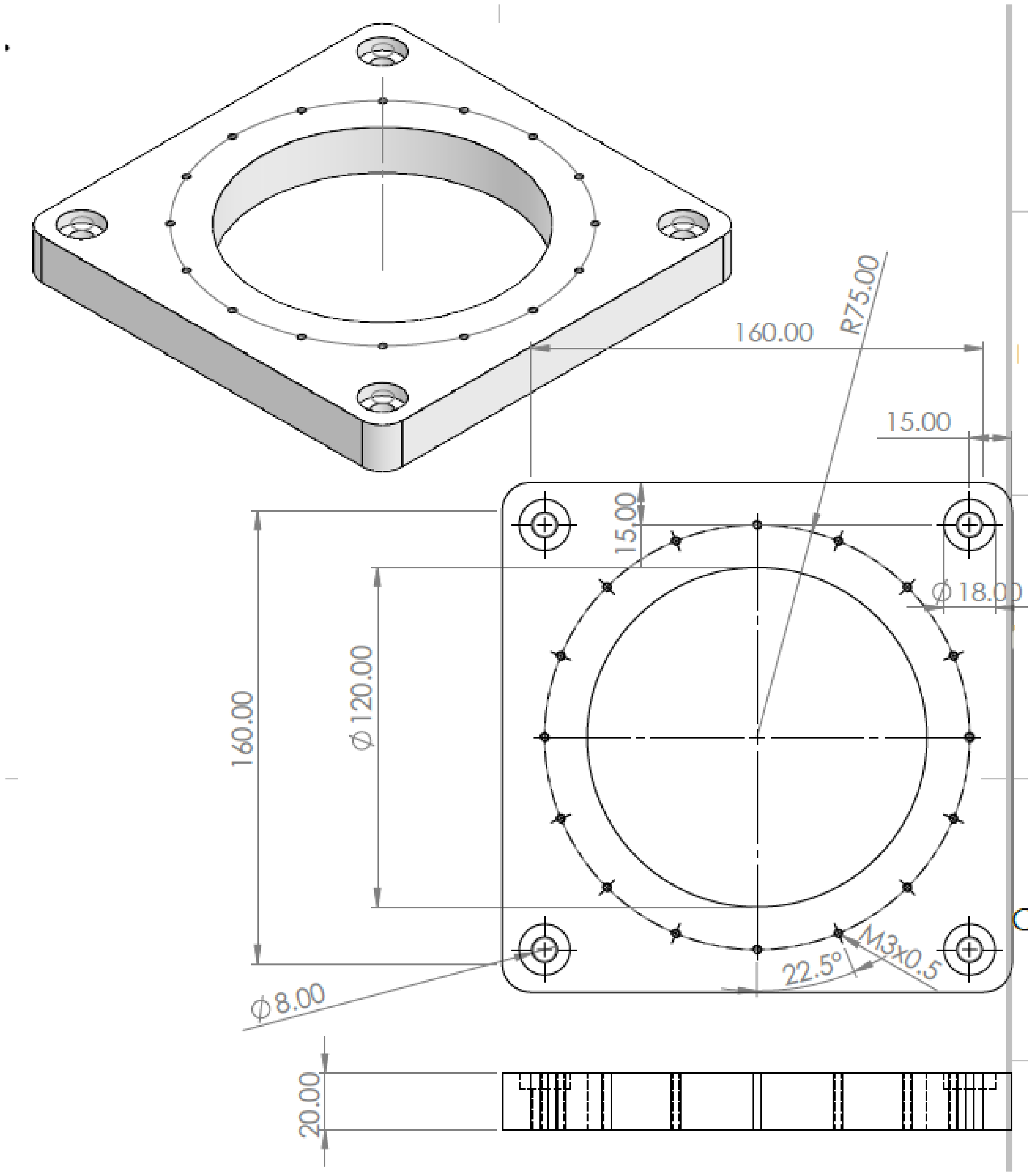


Figure 8 Base

Part	Base
Part material	PLA filament
Dimension	(mm)
Tolerance	± 0.01

Holder:

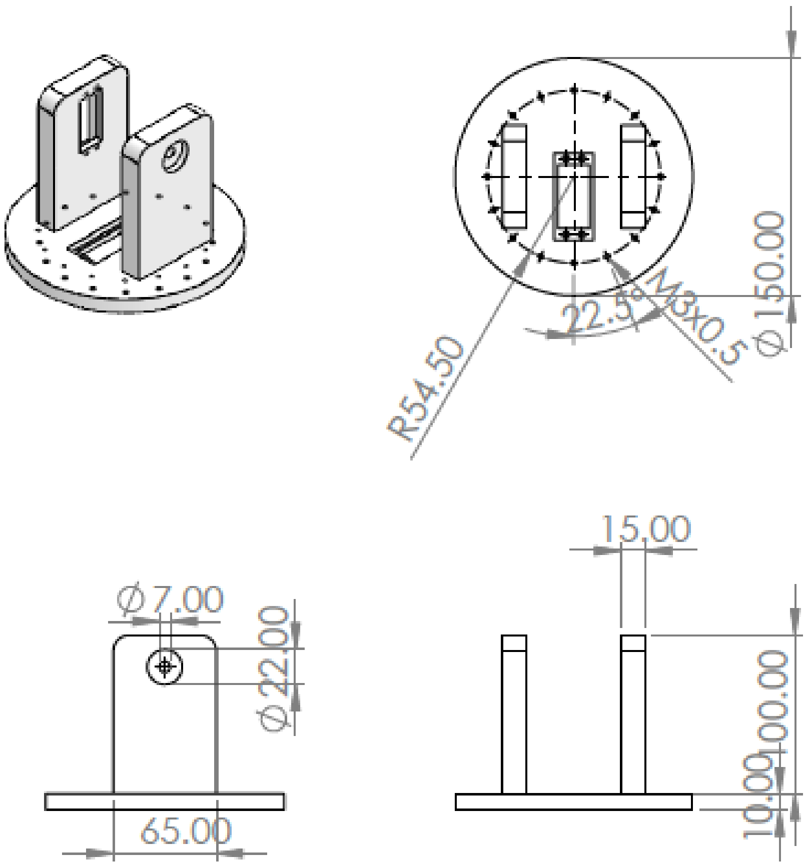


Figure 9 Rotary holder

Part	Rotary holder
Part material	PLA filament
Dimension	(mm)
Tolerance	± 0.01

Link1:

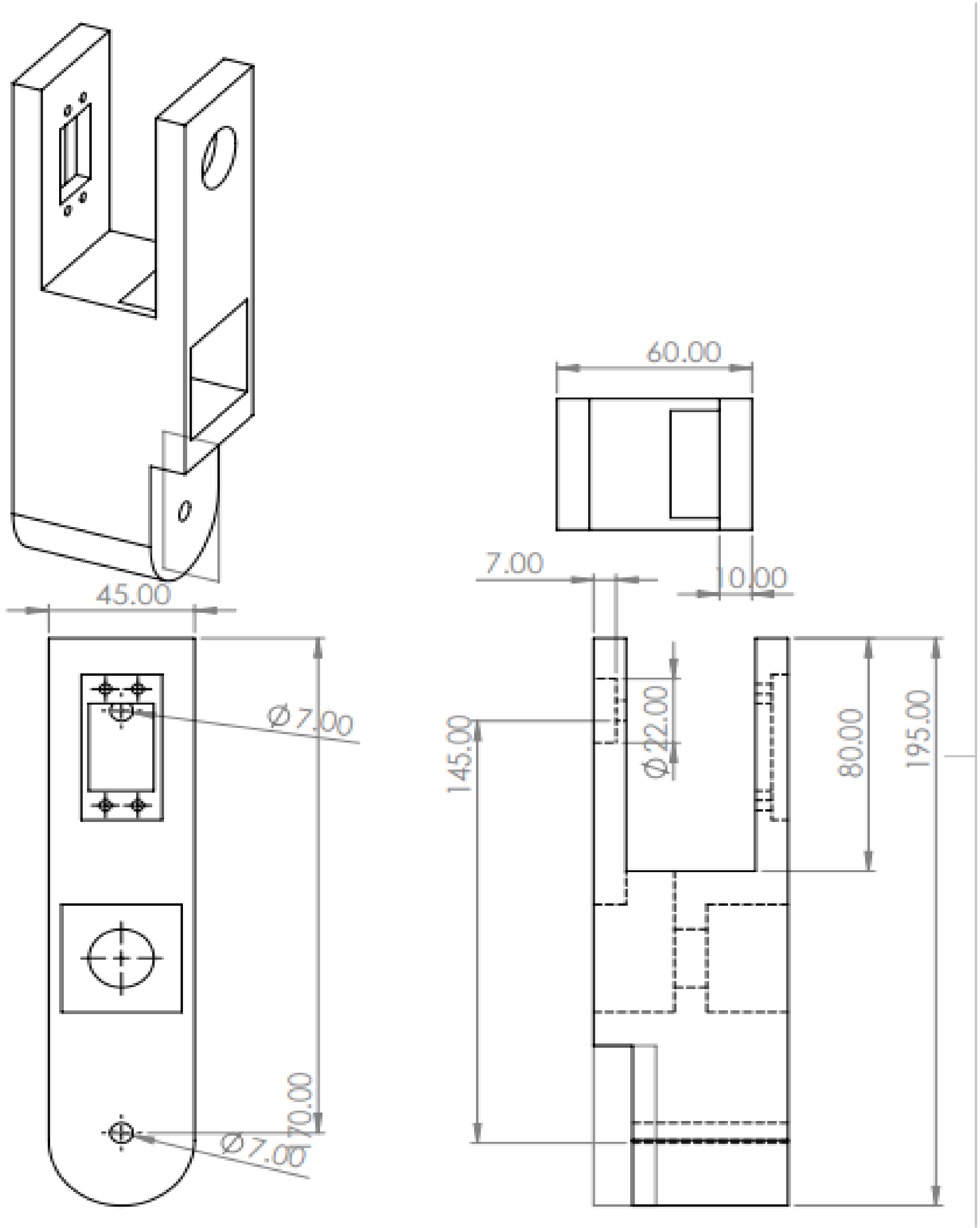


Figure 10 Link 1

Part	Link 1
Part material	PLA filament
Dimension	(mm)
Tolerance	± 0.01

Link2:

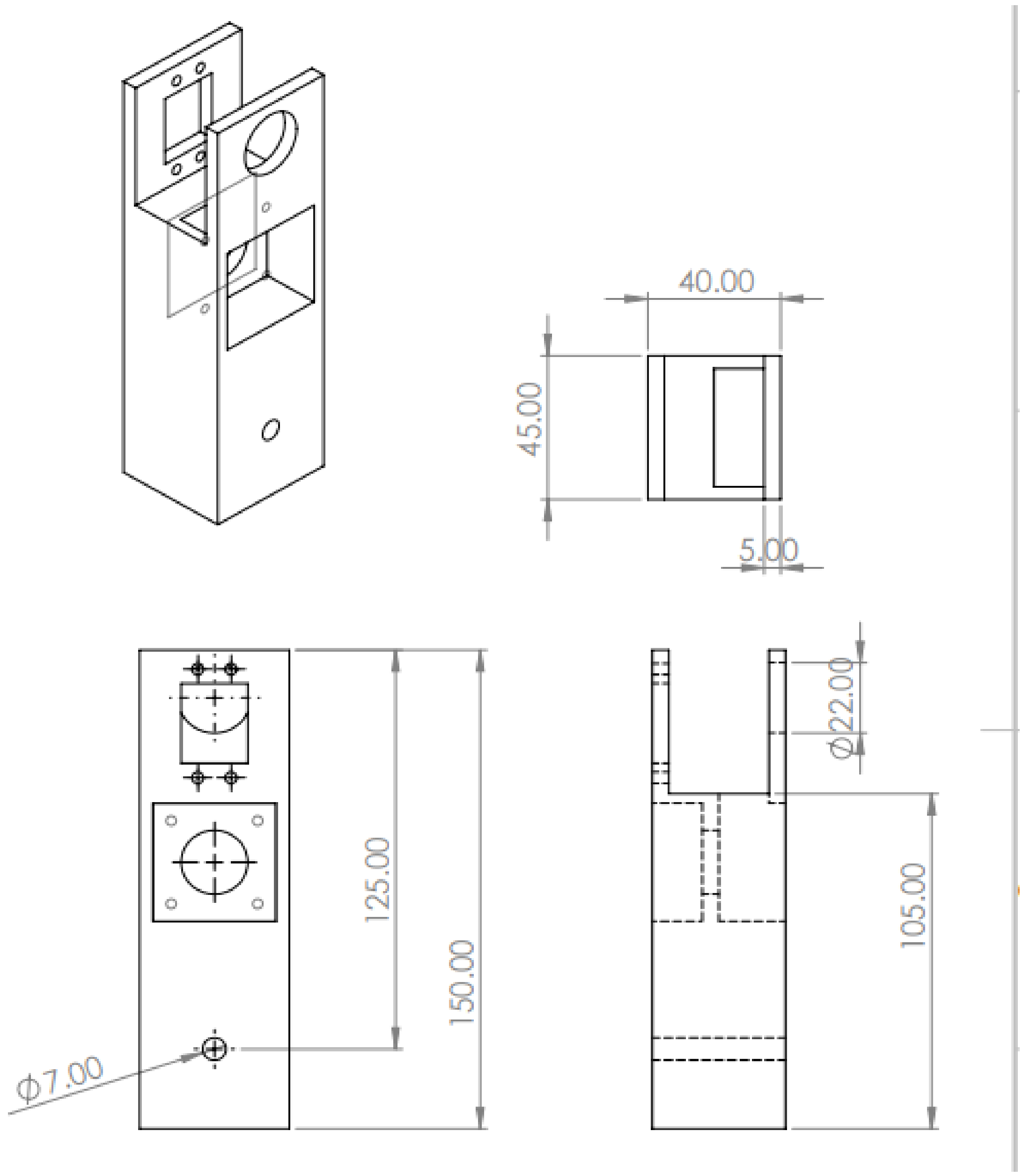


Figure 11 Link 2

Part	Link 2
Part material	PLA filament
Dimension	(mm)
Tolerance	± 0.01

End Effector:

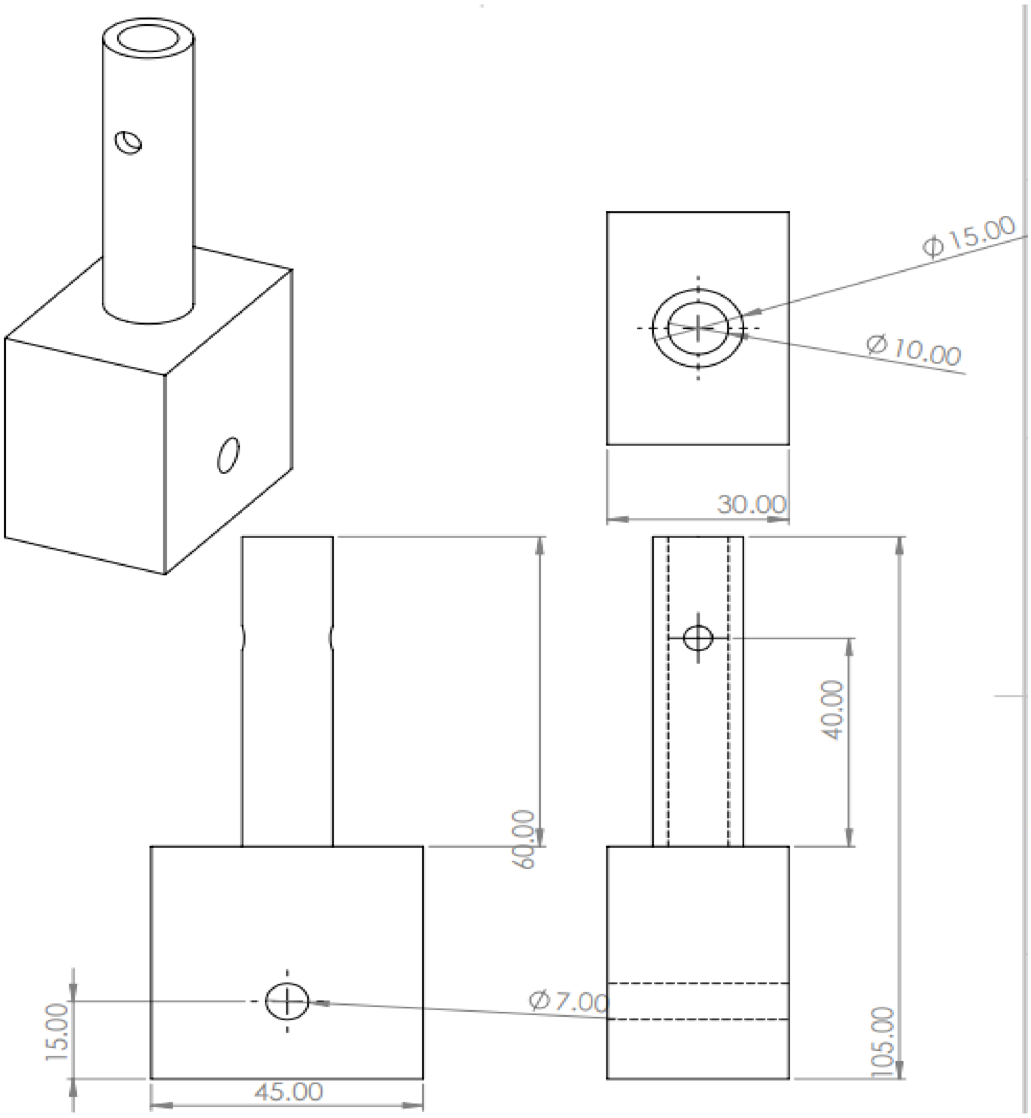


Figure 12 End effector

Part	End effector
Part material	PLA filament
Dimension	(mm)
Tolerance	± 0.01

slewing bearing:

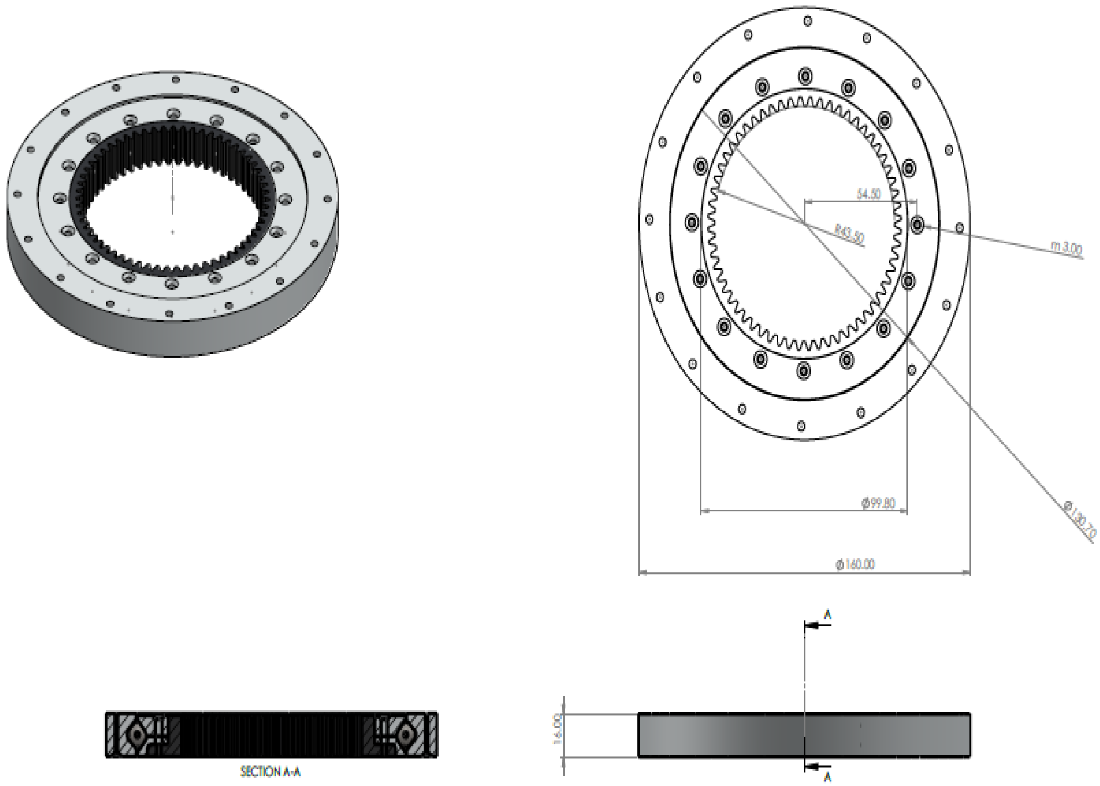


Figure 13 Slewing bearing

Part	Slewing bearing
Part material	PLA filament
Dimension	(mm)
Tolerance	± 0.01

CHAPTER 2: FORCE ANALYSIS & CALCULATIONS

Free body diagram:

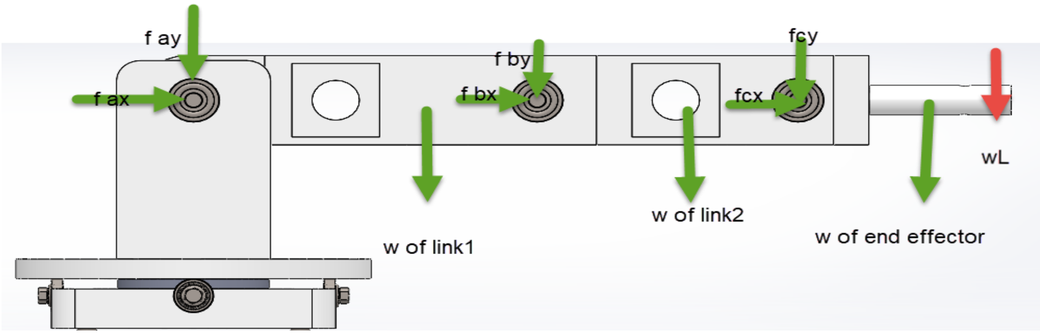
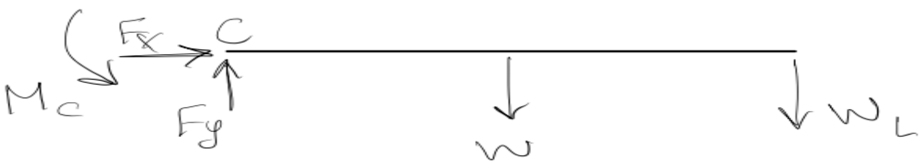
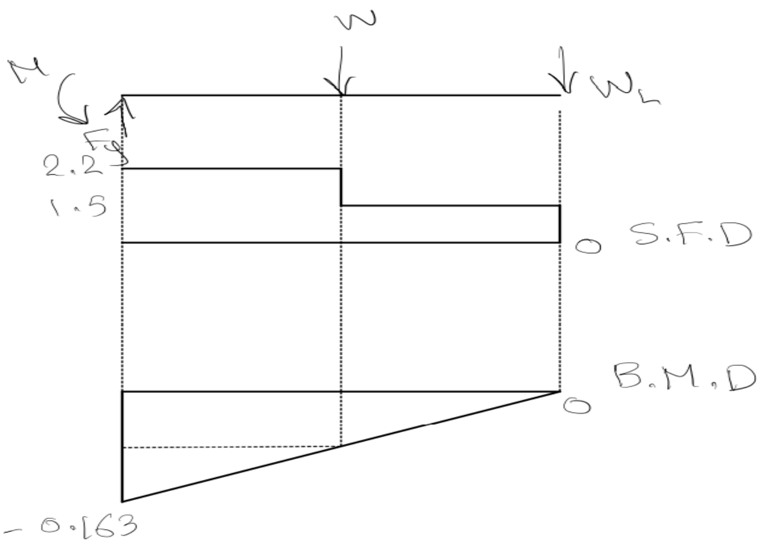


Figure 14 FBD of the whole arm

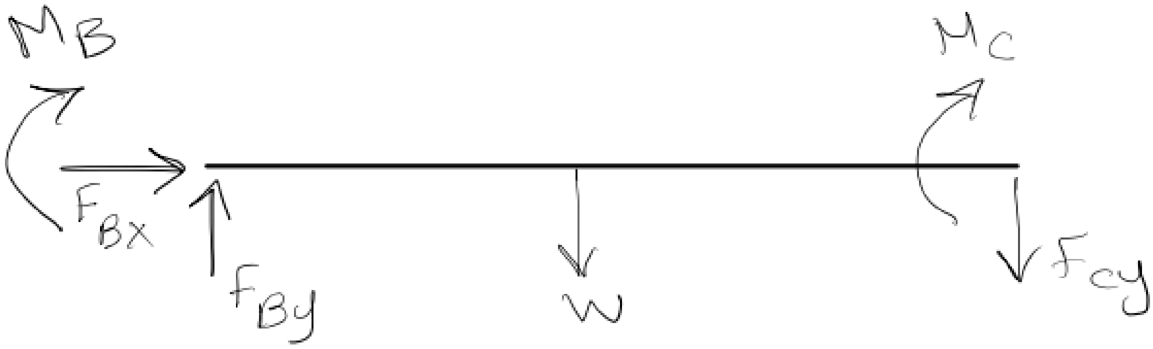
1. End effector:



Given that $W_l = 0.15$ kg , $w = 0.07$ kg L = 90 MM
$\Sigma F_x = 0 , F_x = 0$
$\sum F_y = 0 , F = (w + w_L) \times g =$ (0.15 +0.07)x9.81=2.15 KN



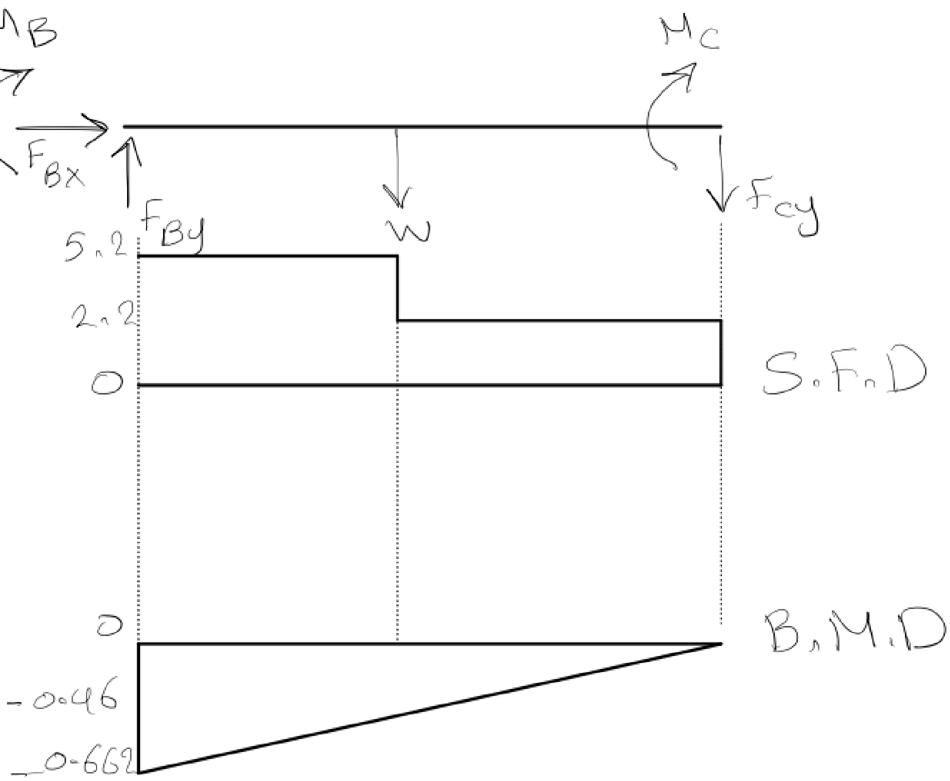
LINK 2:



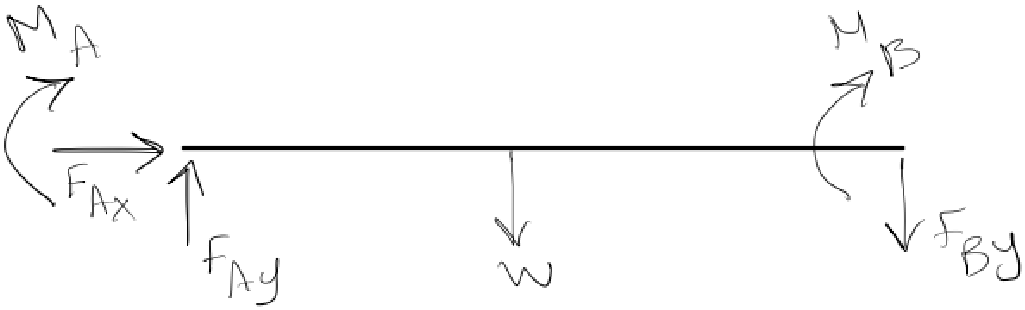
Given that $F_{Cy} = 2.2\ N$, $m_c = 0.163\ N \cdot m$, $w = 0 \cdot 3\ kg$ L= 135mm

$\Sigma F_x = 0$, $F_x = 0$

$\sum F_y = 0$, $F_{By} = (w \times g) + F_{Cy}$ = 3 + 2.2= 5.2 N

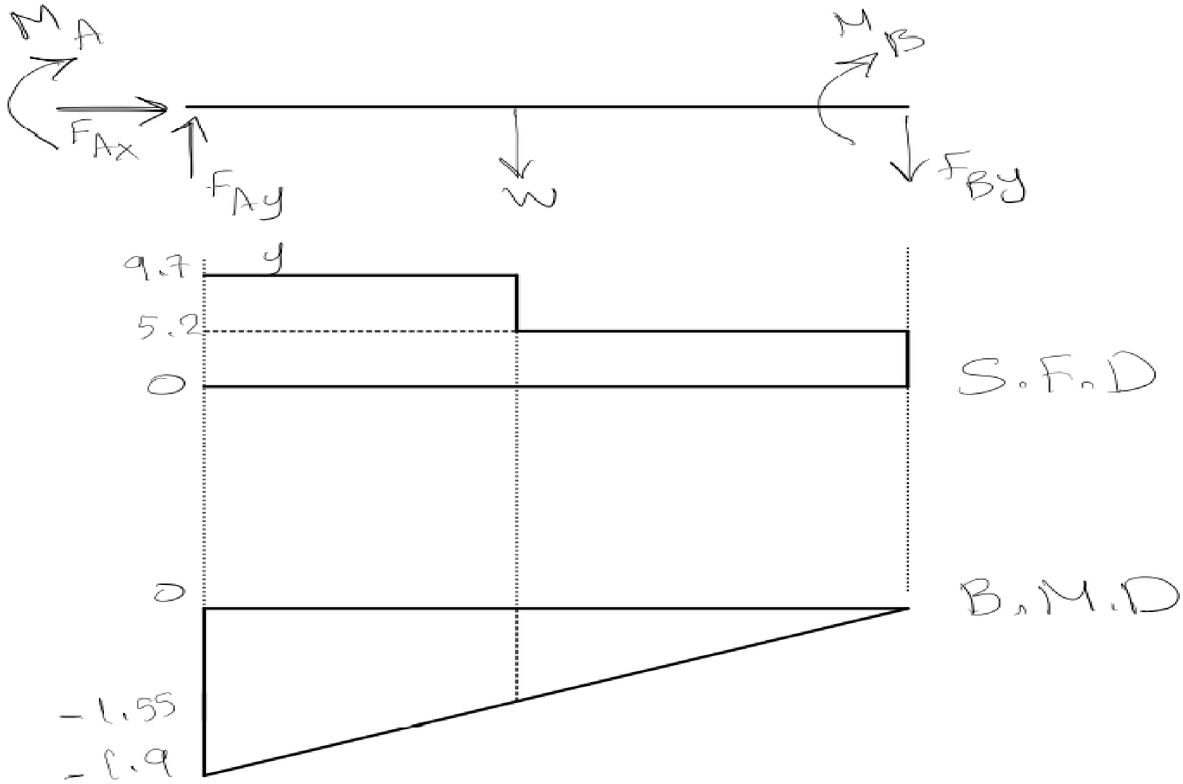


2. LINK 1:



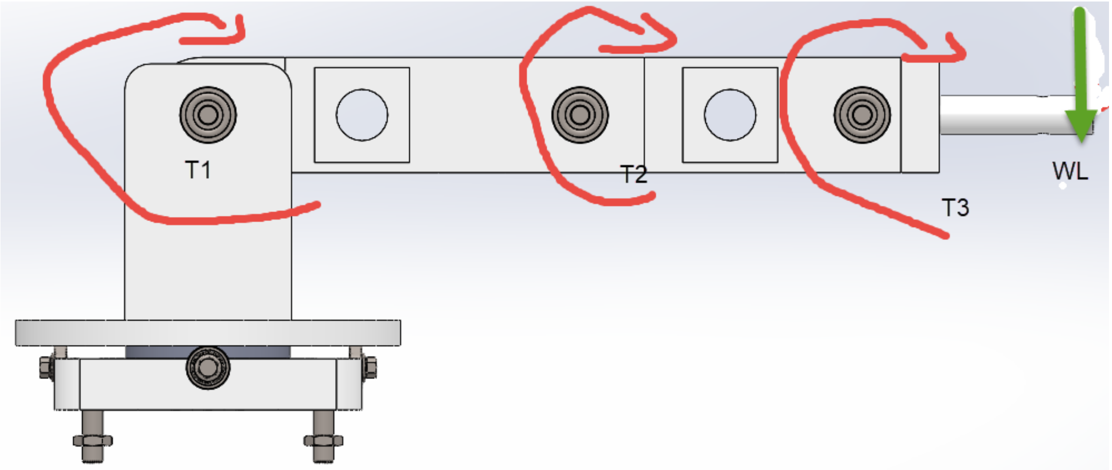
$\Sigma F_x = 0$, $F_{Ax} = 0$

$\sum F_y = 0$, $F_{Ay} = (w \times g) + F_{By}$ = 4.5 + 5.2= 9.7 N L= 0.17



Torque calculations:

FoS=1.5



For end effector:

T_3= (w_L*g*l_3) + (w*g*l_3/2)

T_3= (1.5*0.09) + (0.7*0.045)

T_3= 0.1633 N.m

T_3=0.1633*fos= 0.2445 N.m

For link2:

T_2= (w_L*g*l_3+l_2) +(w*g*l_3/2+l_2) +(w_{m1}*g*l_2) +(w*g*l_2/2)

T_2= (1.5*0.225) + (0.7*0.18) +(0.4*0.135)+(1.6*0.0675)

T_2= 0.6255N.m

T_2=0.6255*fos= 0.93825N.m

For link1:

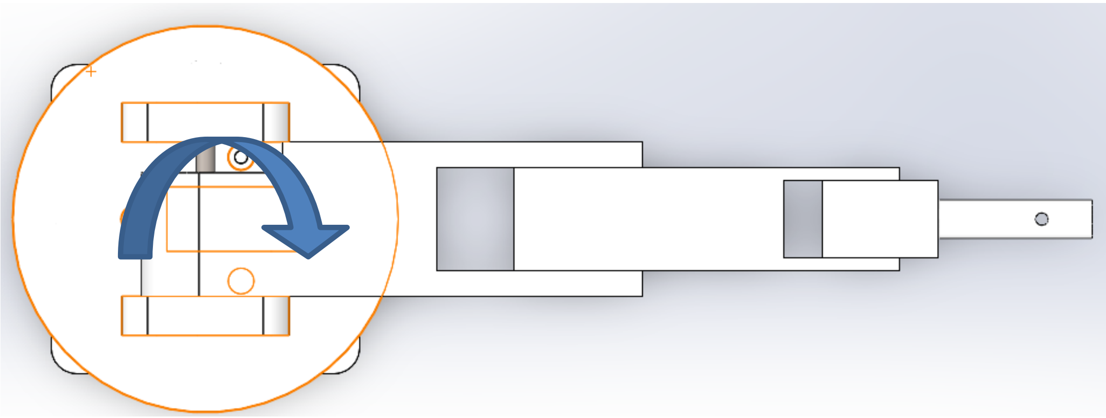
T_1=(w_L*g*l_3+l_2+l_1) +(w*g*l_3/2+l_2+l_1) +(w_{m1}*g*l_2+l_1) +(w*g*l_2/2+l_1)+(w_{m2}*l_1)+(w*g*l_1/2)

T_1=(1.5*0.395) + (0.7*0.35) +(0.4*0.305)+(1.6*0.2375)+(0.7*0.17)+(0.3*0.085)

T_1= 1.5N.m

T_1= 1.5*fos= 2.25N.m

For base:



T_0=I*a

I= 1/3*w*l^2 a=2π

I= 1/3*2*$(0.5)^2$ =0.1667

T_0=I*a= 0.1677*2 π = 1.04N.m

T_0= 1.04*fos= 1.56N.m

Using simulation:

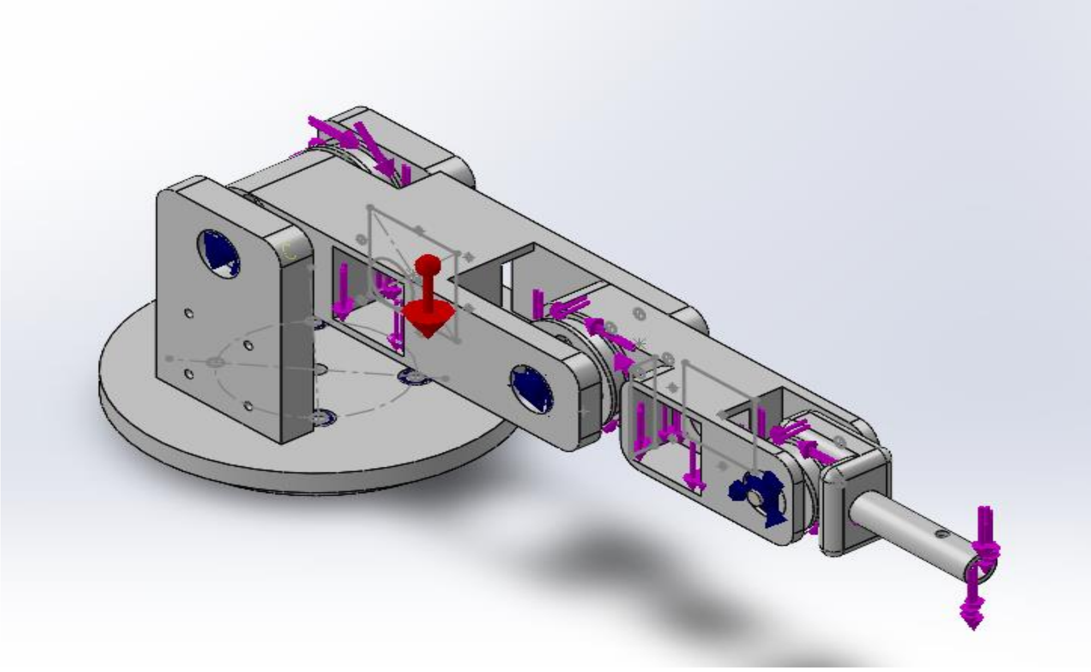


Figure 15 Before simulation

The load:

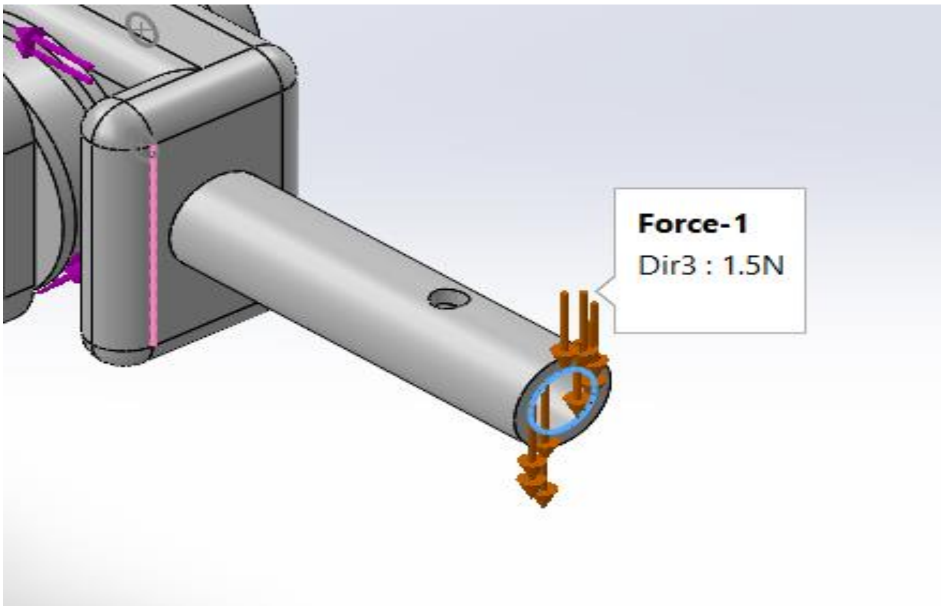


Figure 16 The load applied

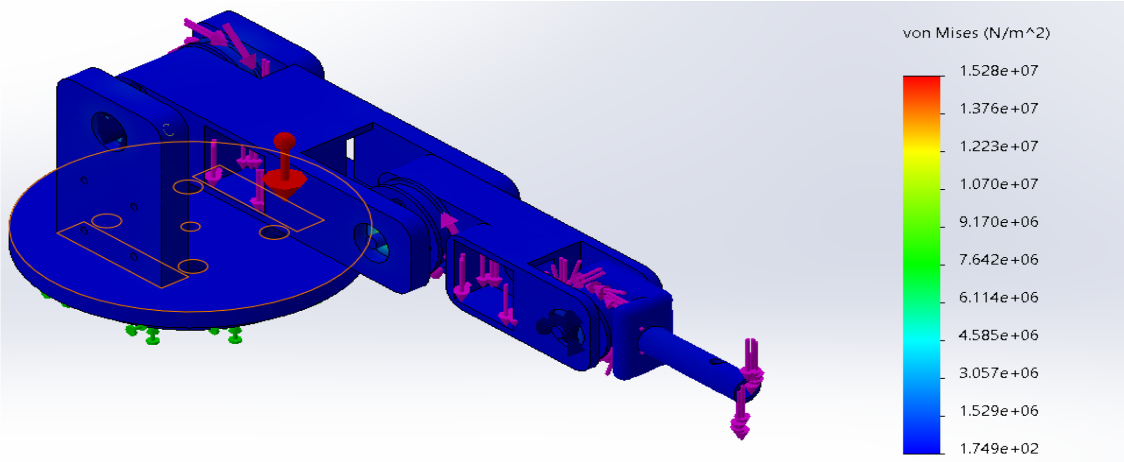


Figure 17 Stress analysis using simulation

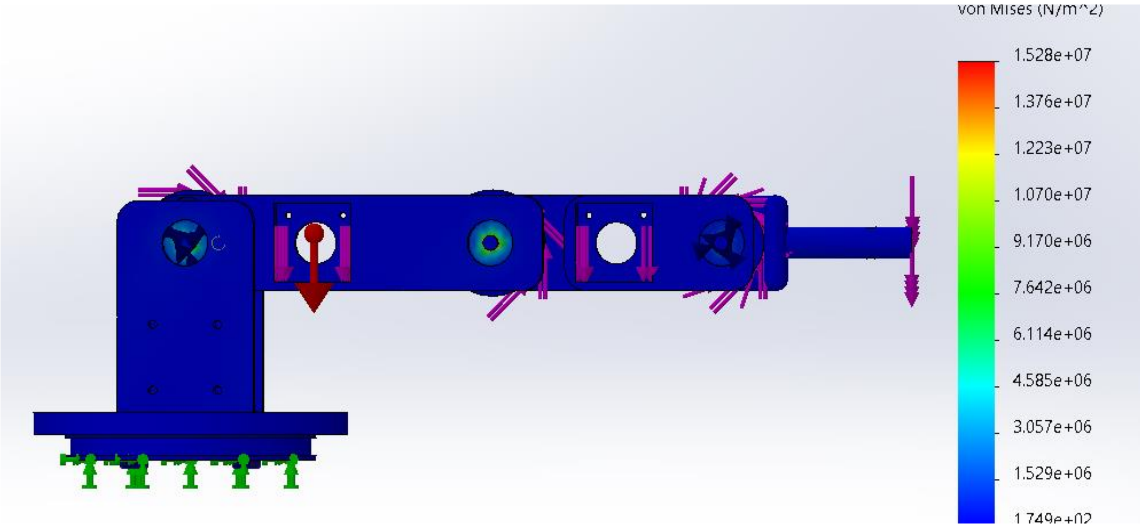


Figure 18 Stress analysis

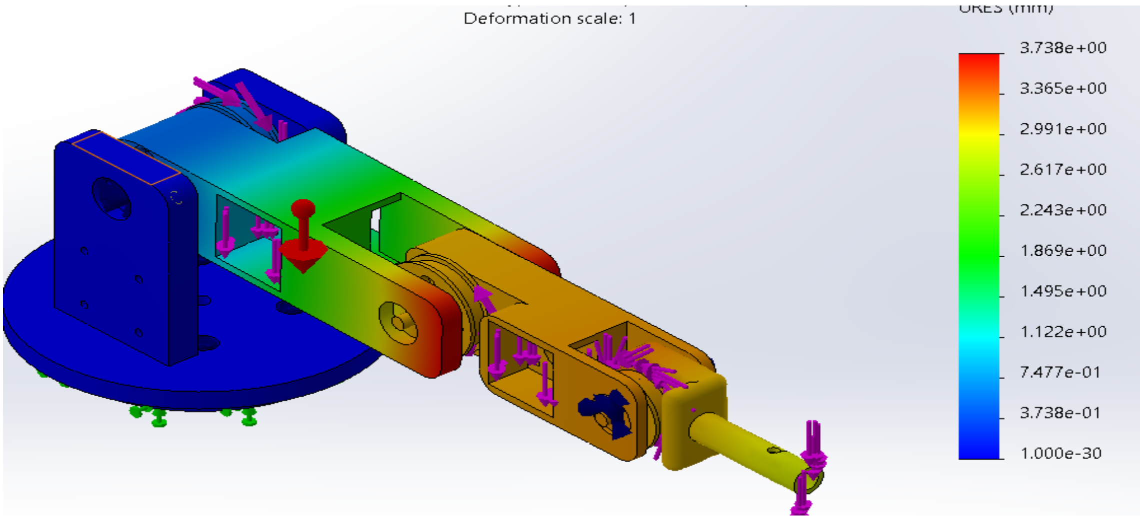


Figure 19 Displacement

CHAPTER 3: 3D PRINTING

3D Printing Technology:

3D printing, also known as additive manufacturing, is a method of creating a three dimensional object layer-by-layer using a computer created design. 3D printing is an additive process whereby layers of material are built up to create a 3D part. This is the opposite of subtractive manufacturing processes, where a final design is cut from a larger block of material. As a result, 3D printing creates less material wastage. **[5]**

What Materials can be used in 3D Printing?

There are a variety of 3D printing materials, including thermoplastics such as acrylonitrile butadiene styrene (ABS), Polylactic acid (PLA), metals (including powders), resins and ceramics.

When was 3D Printing Invented?

The earliest 3D printing manufacturing equipment was developed by Hideo Kodama of the Nagoya Municipal Industrial Research Institute when he invented two additive methods for fabricating 3D models. **[6]**

Building on Ralf Baker's work in the 1920s for making decorative articles (patent US423647A), Hideo Kodama's early work in laser cured resin rapid prototyping was completed in 1981. His invention was expanded upon over the next three decades, with the introduction of stereolithography in 1984. Chuck Hull of 3D Systems invented the first 3D printer in 1987, which used the stereolithography process. This was followed by developments such as selective laser sintering and selective laser melting, among others. Other expensive 3D printing systems were developed in the 1990s-2000s, although the cost of these dropped dramatically when the patents expired in 2009, opening up the technology for more users.

- **material used and assumptions:**

Polylactic Acid (PLA):

Is different than most thermoplastic polymers in that it is derived from renewable resources like corn starch or sugar cane. Most plastics, by contrast, are derived from the distillation and polymerization of nonrenewable petroleum reserves. Plastics that are derived from biomass (PLA) are known as “bioplastics.” **[6]**

Polylactic Acid is biodegradable and has characteristics similar to polypropylene (PP), polyethylene (PE), or polystyrene (PS). It can be produced from already existing manufacturing equipment (those designed and originally used for petrochemical industry plastics). This makes it relatively cost efficient to produce. Accordingly, PLA has the second largest production volume of any bioplastic (the most common typically cited as thermoplastic starch).

PLA is classified as a “thermoplastic” polyester (as opposed to “thermoset”), and the name has to do with the way the plastic responds to heat. Thermoplastic materials become liquid at their melting point (150-160 degrees Celsius in the case of PLA). A major useful attribute about thermoplastics is that they can be heated to their melting point, cooled, and reheated again without significant degradation.

Applications of Polylactic Acid:

Some of the most common uses include plastic films, bottles, and biodegradable medical devices (screws, pins, rods, and plates that are expected to biodegrade within 6-12 months).
PLA constricts under heat and is thereby suitable for use as a shrink wrap material. Additionally, the ease with which Polylactic Acid melts allows for some interesting applications in 3D printing (namely “lost PLA casting”). On the other hand, its low glass transition temperature makes many types of PLA (for example, plastic cups) unsuitable to hold hot liquid.

PLA is one of two common plastics used on FDM machines (Fused deposition modeling) As known as (3D printing) and is commonly available as a 3D printable filament. PLA filament for 3D printing is typically available in a myriad of colors. Polylactic Acid could be CNC machined but it is typically not available in sheet stock or rod form. It is, however, typically available as a thin film for thermoforming or in the form of plastic pellets for injection molding. To adjust material properties, plastic injection mold pellets are typically produced and/or blended together.

One of the interesting things you can do with PLA on a 3D printer is called “lost PLA casting.” This is a process where PLA is printed in the shape of an interior cavity and then encased with plaster-like materials. The PLA is later burned out as it has a lower melting temperature than the surrounding material. The end result is a void that can be filled (often with molten metal).

PLA:

Key properties:

- High rigidity
- Good impact resistance, even at low temperatures
- Good insulating properties
- Good weldability
- Good abrasion and strain resistance
- High dimensional stability (Mechanically strong and stable over time)
- High surface brightness and excellent surface aspect

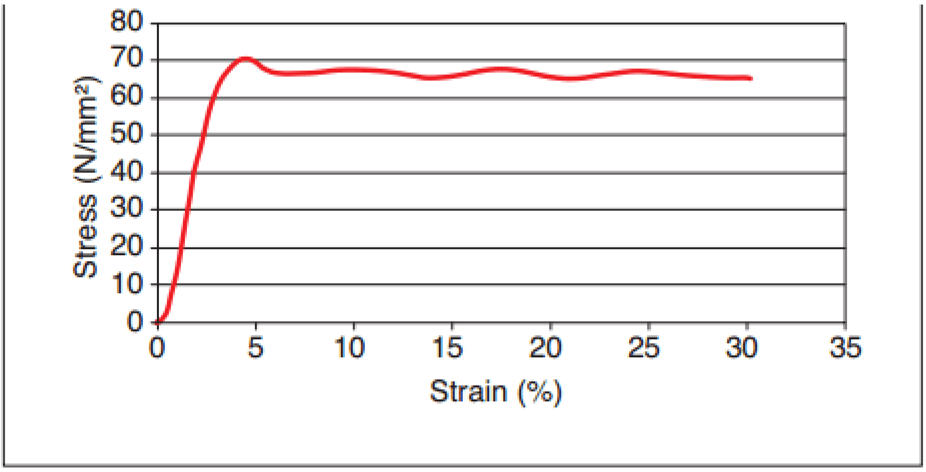
Chemical Properties of PLA:

- Very good resistance to diluted acid and alkalis
- Moderate resistance to aliphatic hydrocarbons
- Poor resistance to aromatic hydrocarbons, halogenated hydrocarbons and alcohols

Mechanical Properties of PLA:

- PLA shows excellent mechanical properties, it is hard and tough in nature and thus delivers good impact strength. PLA offers a high degree of surface quality.

- **Stress strain curve:**



Elongation at Break	10 - 50 %
Elongation at Yield	1.7 - 6 %
Hardness Shore D	100
Strength at Yield (Tensile)	29.6 - 70 MPa
Toughness (Notched Izod Impact at Room Temperature)	200 - 215 J/m
Toughness at Low Temperature (Notched Izod Impact at Low Temperature)	20 - 160 J/m

Table 2 mechanical properties of PLA

Electrical Properties of PLA:

- PLA exhibits good electrical insulating properties.

Arc Resistance	60 - 120 sec
Dielectric Constant	2.7 - 3.2
Dielectric Strength	15.7 - 34 kV/mm
Dissipation Factor	50 - 190 x 10^{-4}
Volume Resistivity	14 - 16 x 10^{15} Ohm.cm

Table 3 Electrical properties of PLA

PLA is readily modified both by the addition of additives and by variation of the ratio of the three monomers Acrylonitrile, Butadiene and Styrene. Heat stabilizers, hydrolysis stabilizers, lubricants, UV stabilizers etc. are being used in non-reinforced and reinforced grades to increase specific material properties.

Hence, grades available include:

- High and medium impact
- High heat resistance, and
- Electroplatable

Limitations of PLA:

- Poor weathering resistance
- Ordinary grades burn easily and continue to burn once the flame is removed
- Scratches easily
- Poor solvent resistance, particularly aromatic, ketones and esters
- Can suffer from stress cracking in the presence of some greases
- Low dielectric strength

CHAPTER 4: ELECTRICAL AND CONTROL

Main Components:

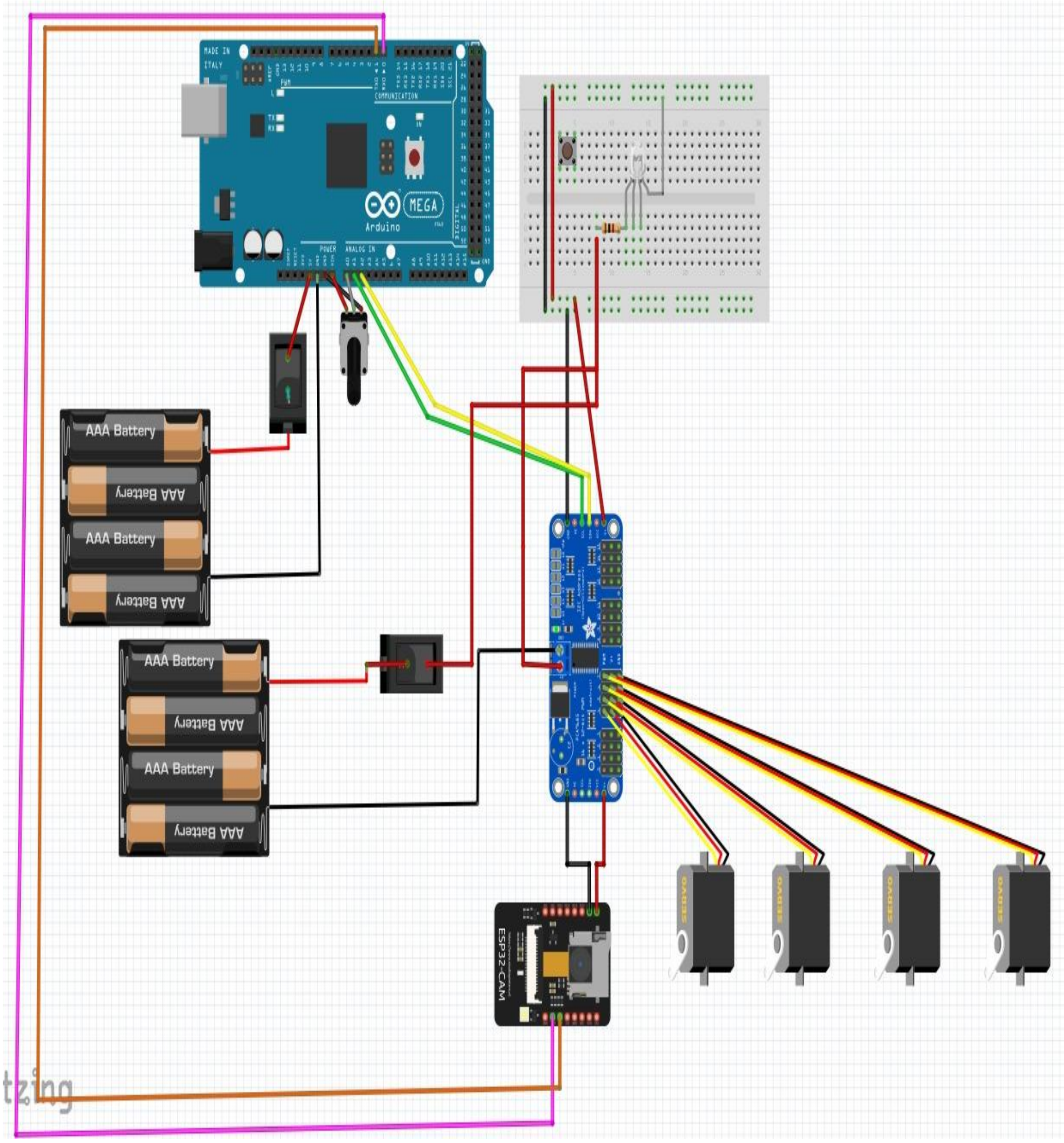
- Arduino Mega 2560
- (4x) Servo motors
- (2x) 5v Power Supply
- Push buttons
- uCam-II (raw/JPEG TTL serial camera module)
- (2x) drivers or Multi Channel PWM driver
- (2x) SPST switches
- Momentary push-button (panel-mount)
- 10K ohm linear potentiometer (panel-mount)
- (11x) 10mm diffused white LED

Description:

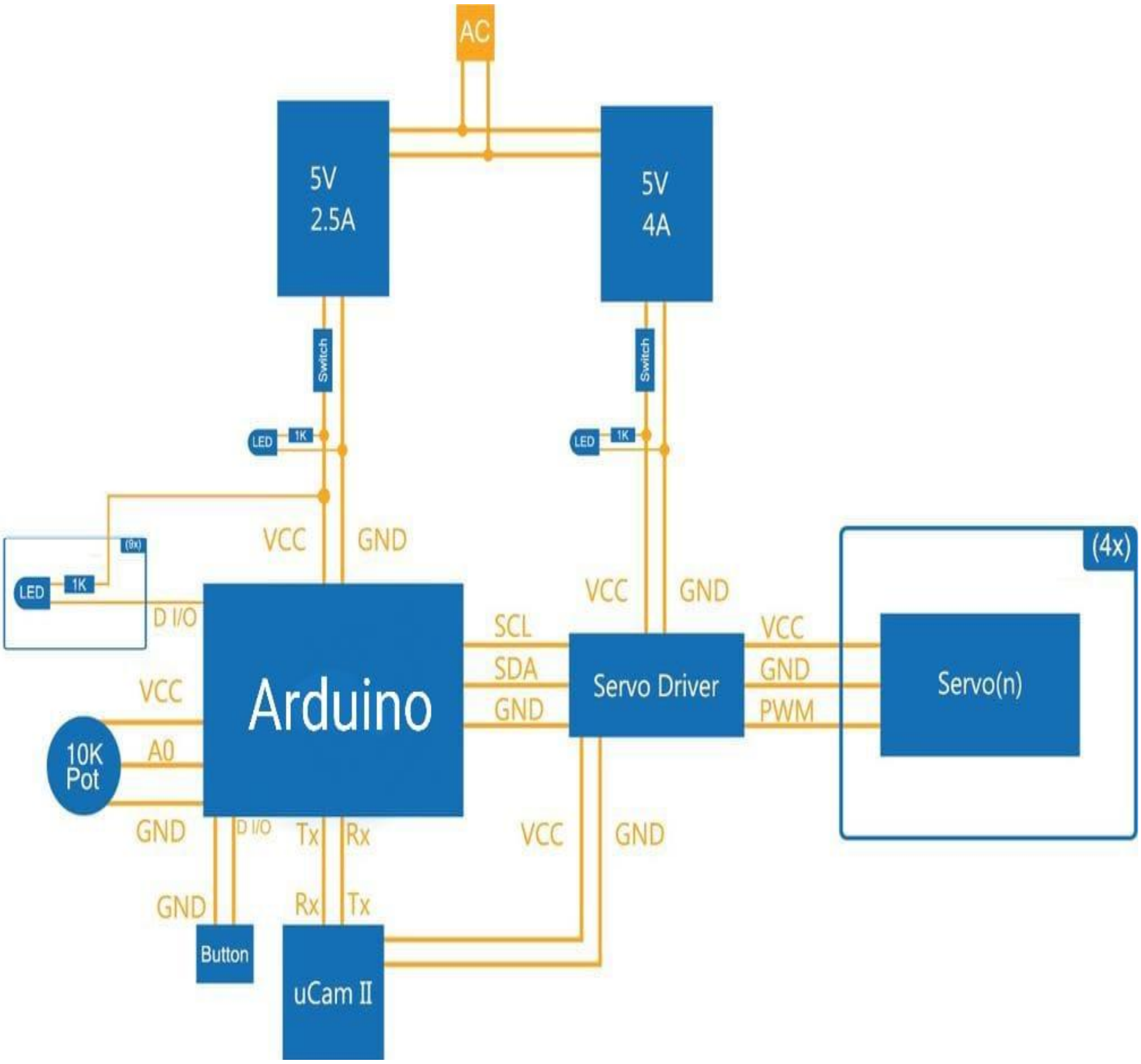
- The camera is fairly low power, and each LED has a 1K ohm current limiting resistor, so the logic power draw should be fine from the 2.5A unit supplied with the Arduino Mega.
- The system runs off of two 5-volt power supplies: one for logic and one for the motor supply.
- The main "user interface" only consists of a potentiometer and a momentary pushbutton. The potentiometer is useful for calibrating the motor positions before the final program.

CIRCUIT DIAGRAM:

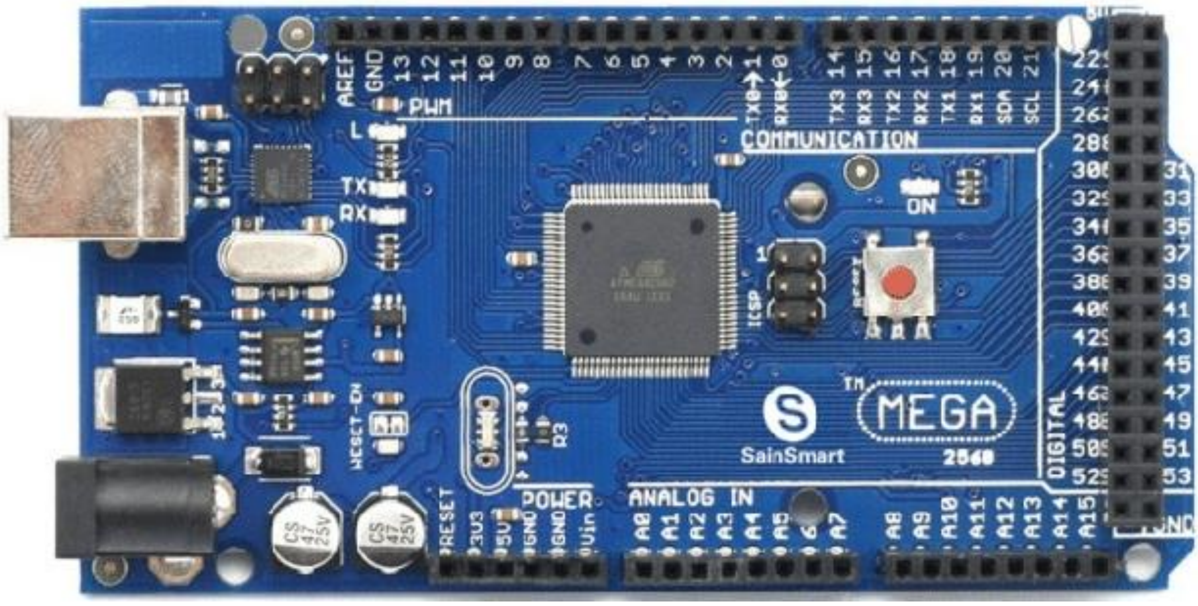
By using Fritzing:



Block Diagram:

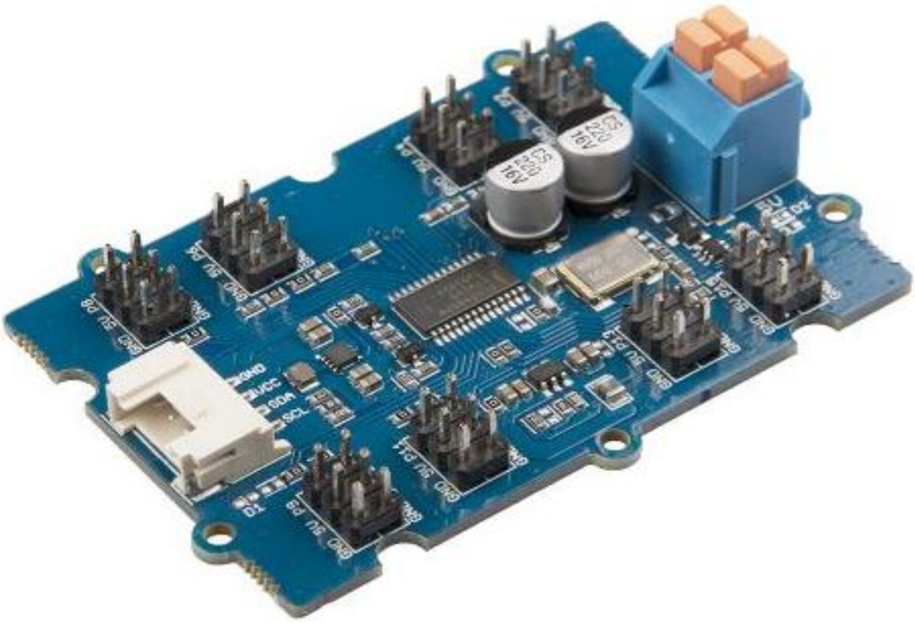


Arduino Mega 2560:



Arduino Mega is the heart of this system, The **Arduino Mega 2560** is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega 2560 board is compatible with most shields designed for the Uno and the former boards Duemilanove or Diecimila.

- **<u>Multi-channel PWM driver (Pulse Width Modulation)</u>**



is an efficient way to vary the speed and power of motors.... These drivers can for example be used to vary the speed of small electric vehicles. It generates analogue signals by using a digital source.
A PWM signal is basically a square wave which is switched between on and off state.\The duty cycle and frequency of a PWM signal determine its behavior.

<u>Applications:</u>

- Drive buzzer with different loudness.
- Control speed of the <u>motor.</u>
- Control the direction of a <u>servo.</u>
- Provide an analog output.
- Generate audio signal.
- Telecommunication: Encode message.

Servo Motors:

servomotor is **a rotary actuator or linear actuator** that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback

SPST switch:

A Single Pole Single Throw (SPST) switch is a switch that only has a single input and can connect only to one output.

uCam-II (raw/JPEG TTL serial camera module):

The uCAM-II (microCAM-II) is a highly integrated serial camera module which can be attached to any host system that requires a video camera or a JPEG compressed still camera for embedded imaging applications. The module uses a CMOS VGA colour sensor along with a JPEG compression chip that provides a low cost and low powered camera system. The module has an on-board serial interface (TTL) that is suitable for a direct connection to any host micro-controller UART or a PC system COM port.

The uCAM-II is capable of outputting both RAW format and JPEG format images. User commands are sent using a simple serial protocol that can instruct the camera to send low resolution (80x60 to 160x120) single frame raw images for viewing or high resolution (160x128 to 640x480) JPEG images for storage or viewing.

CHAPTER 5: ARM ROBOT KINEMATICS

Introduction:

Our robot-arm has 4 degrees of freedom (DOF). A painting brush is mounted on the end effector. A serial manipulator consists of a fixed base, a series of links connected by joints, and ending at a free end carrying the tool or the end-effector. In contrast to parallel manipulators, there are no closed loops. By actuating the joints, one can position and orient the end-effector in a plane or in three-dimensional (3D) space to perform desired tasks with the painting brush.

This chapter deals with kinematics of serial manipulators where the motion of links is studied without considering the external forces and torques which cause these motions. The serial manipulator geometries are described using the well-known Denavit-Hartenberg (D-H) parameters. Two well-known problems, namely the forward and inverse kinematics problems, are posed, their solution procedures discussed in detail and illustrated with examples of planar and spatial serial manipulators. **[8]**

There is a 6 Degree of Freedom 'law' of rigid body movement in three-dimensional space, there are translational movements along axis x, y, z (3 DoF), and rotational movements along axis x, y, z (3 DoF), so the total is six. This robot end effector is capable of all three-axis translational freedoms, and one rotational freedom, so hence why it is 4 DoF robot.

System Planing:

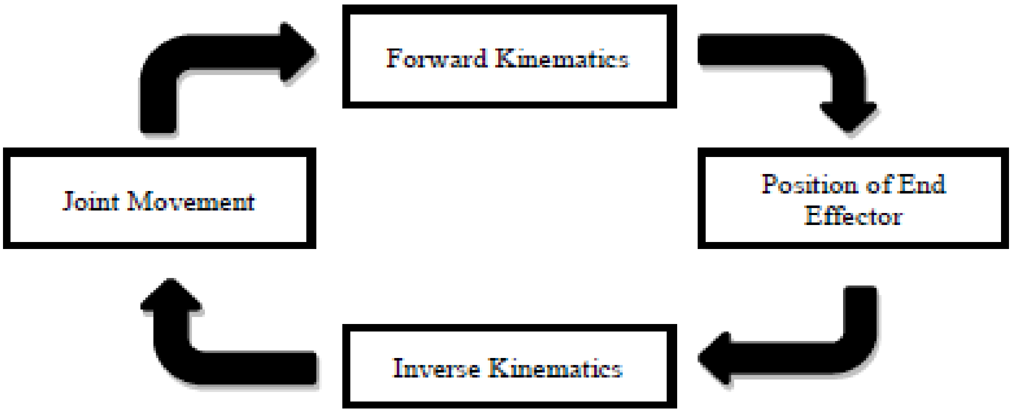


Figure 20The block diagram explains how the kinematics system of arm robot evaluated. There are 2 variable of robot kinematics forward and inverse kinematics.

Forward Kinematics:

We move the arm by sending pulses (that correspond to angles) to the motors, to choose the angles and to know where the arm and brush are we use kinematics.

- Kinematics is the branch of classical mechanics which describes the motion of points, bodies (objects) and systems of bodies (groups of objects) without consideration of the causes of motion.
- Forward kinematics allows us to determine the position of points of the body in space, given to position of the individual joints. Since we can define the angular position of the motors, we can determine where in 3D space the tip of the brush is.
- We'll want to work in Cartesian coordinate space since the canvas is ultimately a set number of points on a plane.

2D FORWARD KINEMATICS:

Before going into full 3D space, we will take a look at determining the position of a single point, given a single angular input.
To begin, let's draw a point A at the XY origin (0, 0). Point A represents the axis of rotation for motor A (the shoulder).
The strut that extends from motor A to the the shaft of motor B is a fixed length, we'll call this line segment L1.

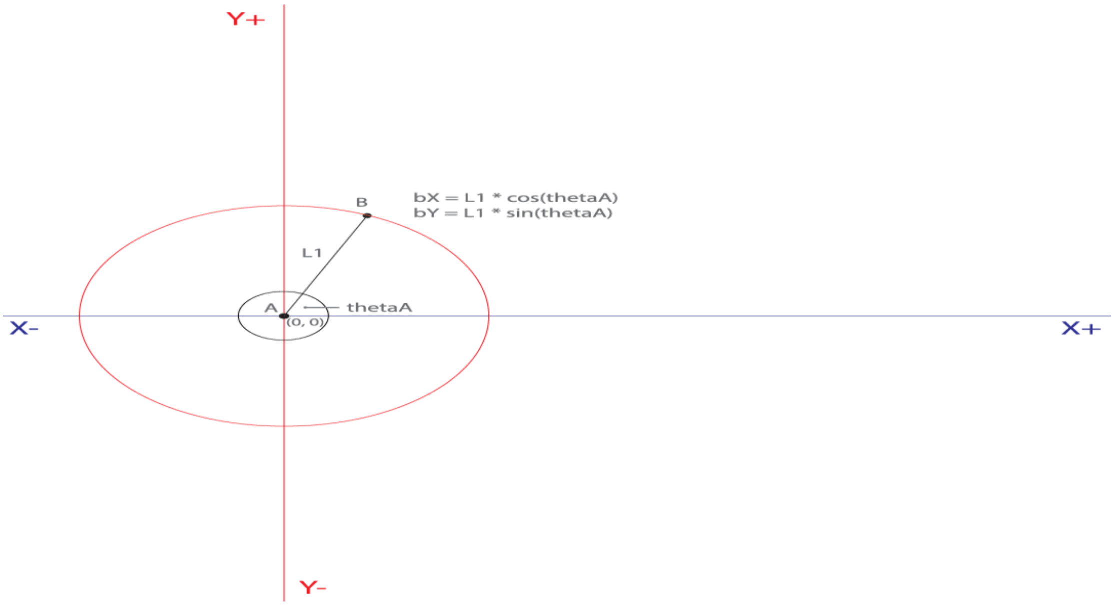


Figure 21 Graph1 working in 2d space

So how do we find the location of point B, given the angle thetaA?

Using L1 as the radius of a circle about point A, we can find the Cartesian coordinates of point B with (X equal to L1 x cos(thetaA) and Y equal to L1 x sin(thetaY).

We'll want to work in Cartesian coordinate space since the canvas is ultimately a set number of points on a plane.

Given point B and thetaB, we can now find point C.

For the X value of point C, we'll multiply L2 by the cosine of (thetaA + thetaB) and then add this value to the X value of point B.

For the Y value of point C, we'll multiply L2 by the sine of (thetaA + thetaB) and then add this to the Y value of point B.

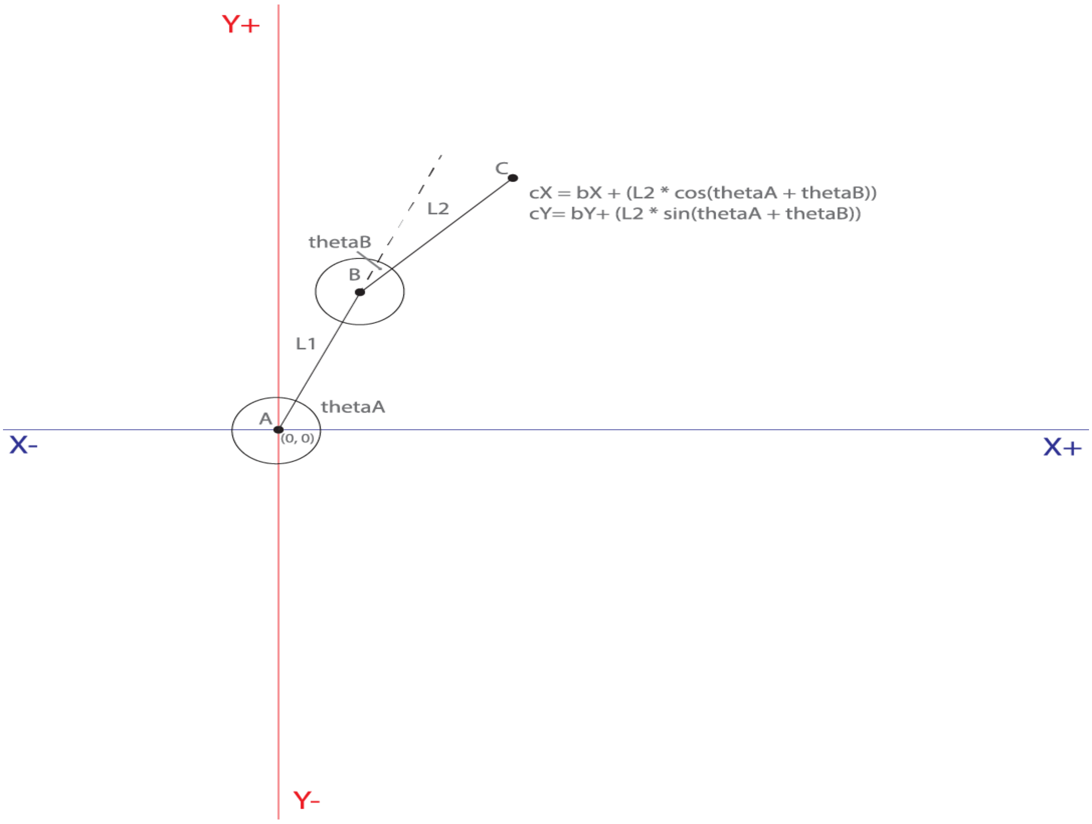


Figure 22 graph2 working in 2d space

This pattern extends to find point D (the tip of the brush), the equation of finding which is shown in the third graph.

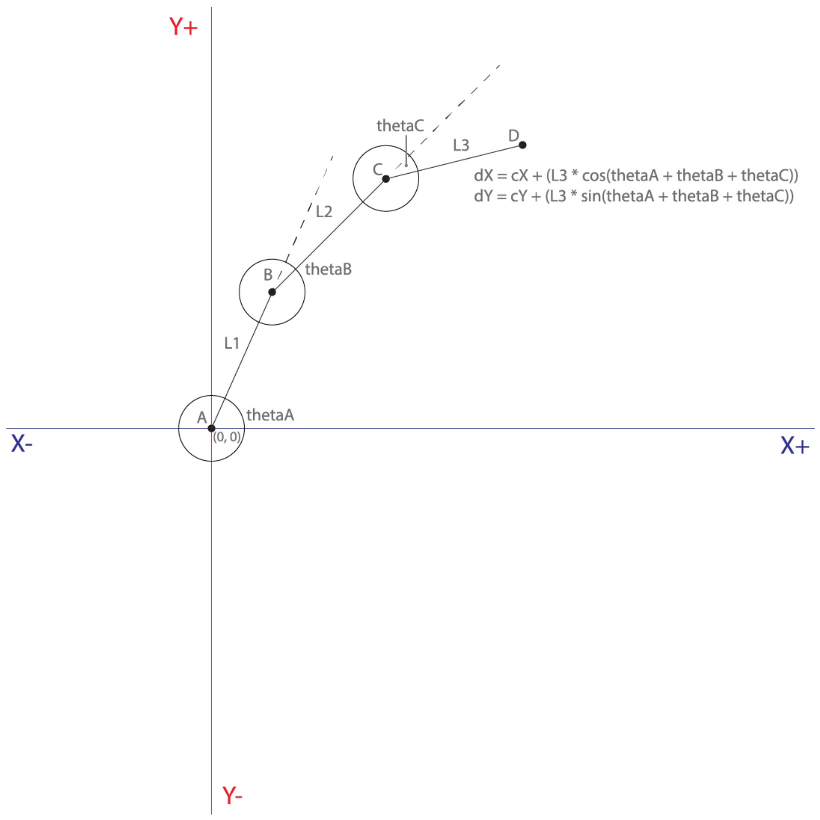


Figure 23 Final graph working in 2d space

For the first link:

$$bx = L1\cos\theta_A$$

$$by = L1\sin\theta_A$$

For the second link:

$$cx = bx + L2\cos(\theta_A + \theta_B)$$

$$cy = by + L2\sin(\theta_A + \theta_B)$$

For the third link: (Position of end effector)

$$dx = cx + L3\cos(\theta_A + \theta_B + \theta_C)$$

$$dy = cy + L3\sin(\theta_A + \theta_B + \theta_C)$$

3D FORWARD KINEMATICS:

While the upper arm only moves along a 2D plane, the base servo rotates all those points into 3D space.

Using DH parameters:

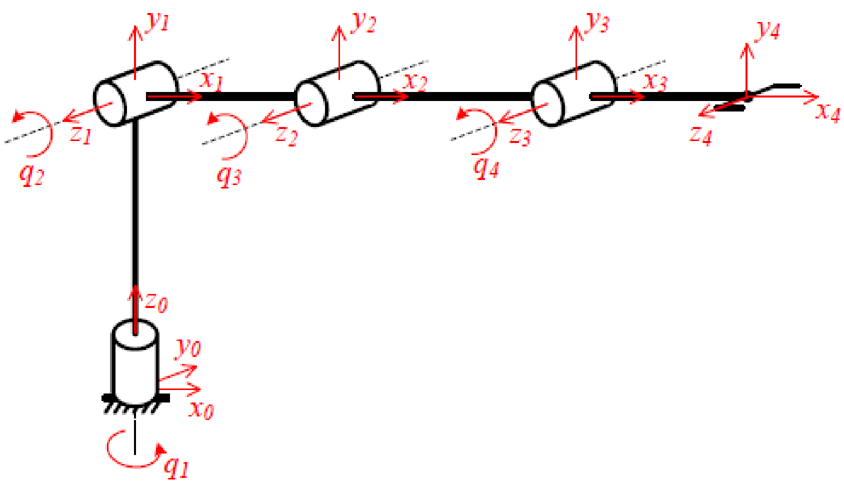


Figure 24 4DOF robot arm kinematic structure and table with DH parameters.

Using Denavit-Hartenberg (D-H) parameters:

1. Rotation of θ about current Z axis.
2. Translation of d along current Z axis.
3. Translation of a along current X axis (Offset).
4. Rotation of α about current X axis.

Table 4 DH parameters

n	θi	Di	ai	αi
1	q1	L1	0	$\pi/2$
2	q2	0	L2	0
3	q3	0	L3	0
4	q4	0	L4	0

Using the equation:

$$A_i = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A1 = \begin{vmatrix} C\theta 1 & -S\theta 1 C\pi/2 & S\theta S\pi/2 & 0 \\ S\theta 1 & C\theta 1 C\pi/2 & -C\theta 1 S\pi/2 & 0 \\ 0 & S\pi/2 & C\pi/2 & L1 \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} C\theta 1 & 0 & S\theta 1 & 0 \\ S\theta 1 & 0 & -C\theta 1 & 0 \\ 0 & 1 & 0 & L1 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$A2 = \begin{vmatrix} C\theta 2 & -S\theta 2 C0 & S\theta 2 S0 & L2C\theta 2 \\ S\theta 2 & C\theta 2 C0 & -C\theta 2 S0 & L2S\theta 2 \\ 0 & S0 & C0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} C\theta 2 & -S\theta 2 & 0 & L2C\theta 2 \\ S\theta 2 & C\theta 2 & 0 & L2S\theta 2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$A3 = \begin{vmatrix} C\theta 3 & -S\theta 3 C0 & S\theta 3 S0 & L3C\theta 3 \\ S\theta 3 & C\theta 3 C0 & -C\theta 3 S0 & L3S\theta 3 \\ 0 & S0 & C0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} C\theta 3 & -S\theta 3 & 0 & L3C\theta 3 \\ S\theta 3 & C\theta 3 & 0 & L3S\theta 3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$A4 = \begin{vmatrix} C\theta 4 & -S\theta 4 C0 & S\theta 4 S0 & L4C\theta 4 \\ S\theta 4 & C\theta 4 C0 & -C\theta 4 S0 & L4S\theta 4 \\ 0 & S0 & C0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} C\theta 4 & -S\theta 4 & 0 & L4C\theta 4 \\ S\theta 4 & C\theta 4 & 0 & L4S\theta 4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$${}^{0}_{4}T = A1A2A3A4$$

Another method to find 3D coordinate of point D (For illustration):

While the upper arm only moves along a 2D plane, the base servo rotates all those points into 3D space. To calculate the new coordinates of our joint points, we'll need to flip our perspective.

Looking from the top down, the arm extends positively along the X axis. We've now got a new polar space where the radius of the circles for each point is equal to the X coordinate of each point on the XY plane. given angles thetaA, thetaB, and thetaC we can now find the 3D coordinate of point D.

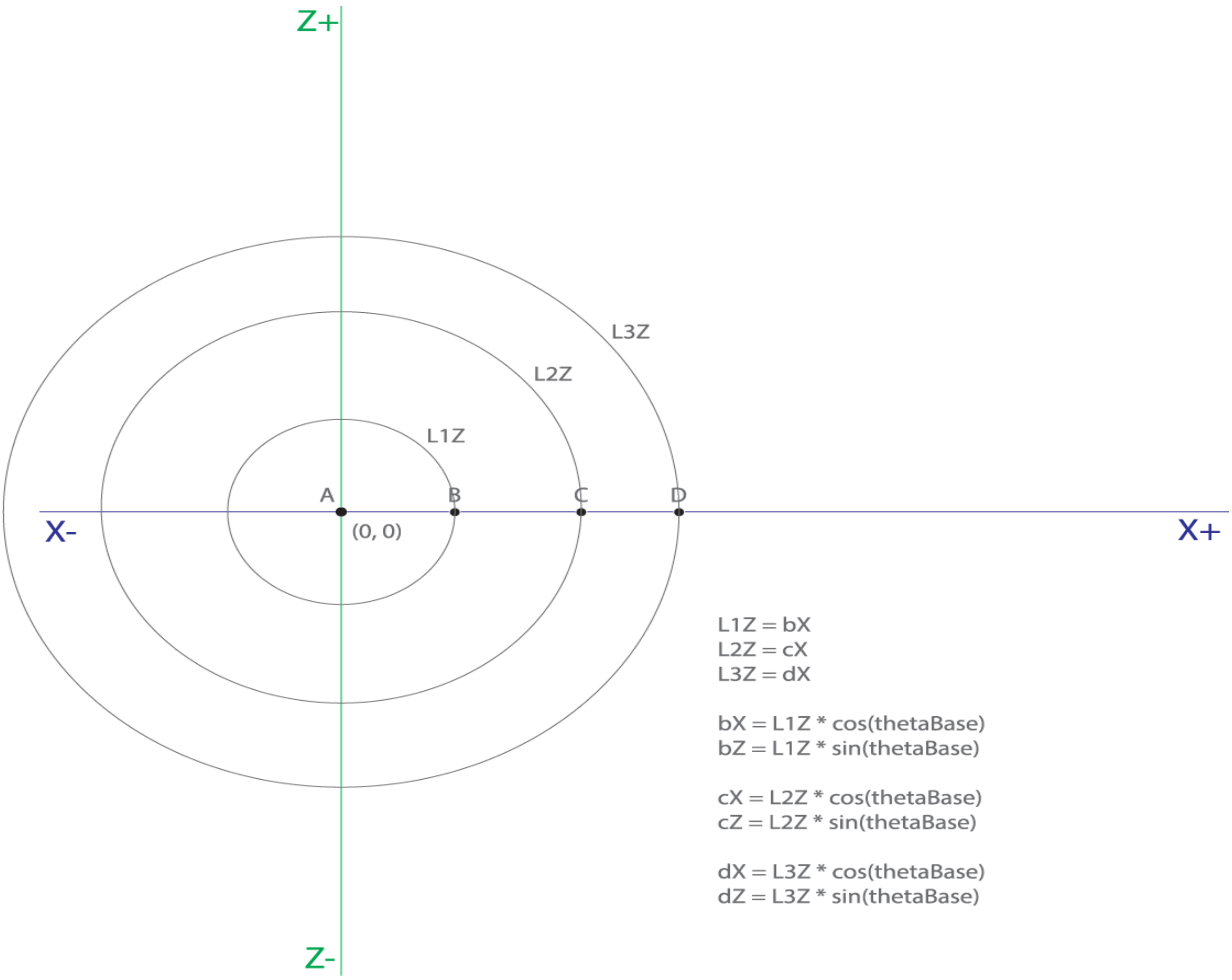


Figure 25 Graph 1 of working in 3d space

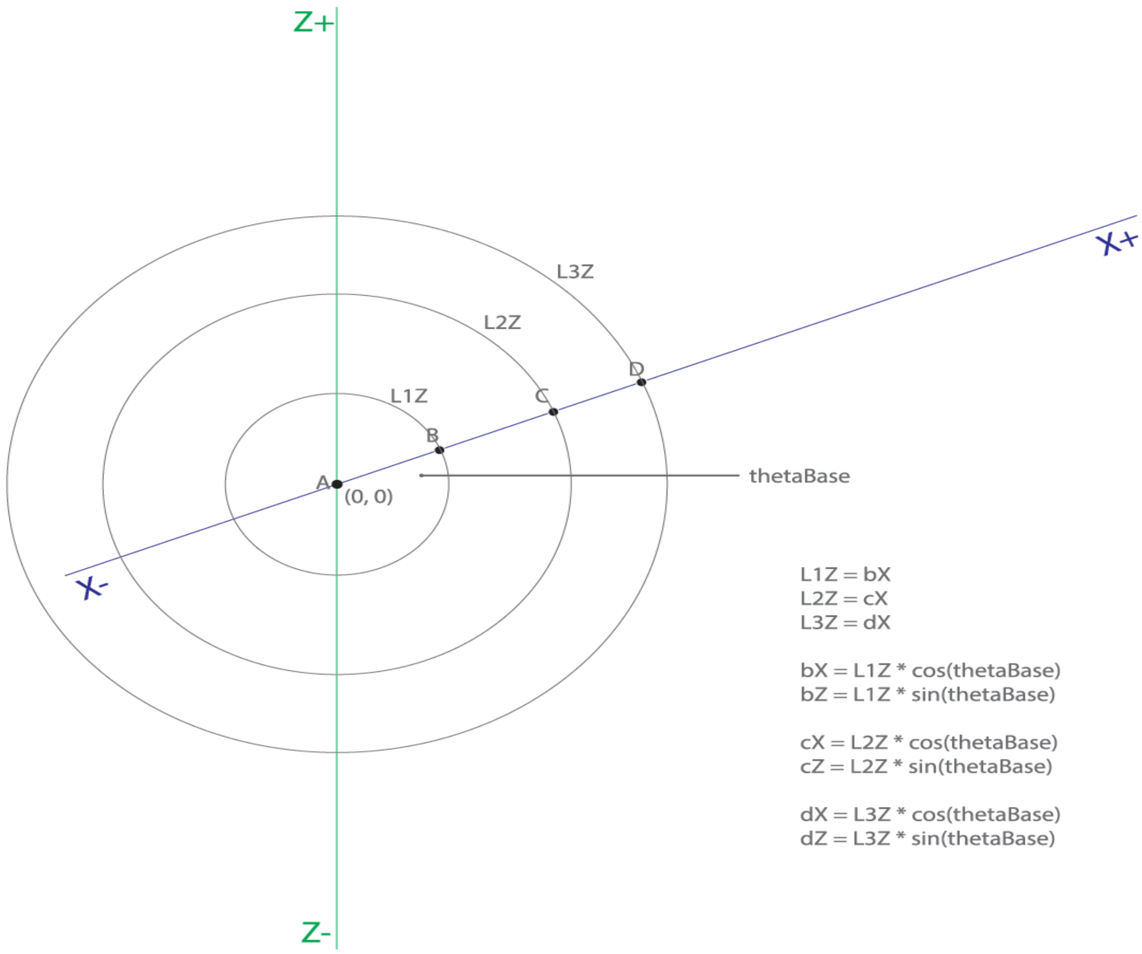


Figure 26 Graph 2 of working in 3D space

INVERSE KINEMATICS:

The inverse kinematics problem of the serial manipulators has been studied for many decades. It is needed in the control of manipulators. Solving the inverse kinematics is computationally expansive and generally takes a very long time in the real time control of manipulators. Tasks to be performed by a manipulator are in the Cartesian space, whereas actuators work in joint space. Cartesian space includes orientation matrix and position vector. However, joint space is represented by joint angles. The conversion of the position and orientation of a manipulator end-effector from Cartesian space to joint space is called as inverse kinematics problem. There are two solutions approaches namely, geometric and algebraic used for deriving the inverse kinematics solution, analytically. Let’s start with geometric approach.

For the brush to paint a pictures it needs to be given specified positions, those positions are then defined by value of each x, y, z axis, then how much the angle of each joint should rotate in order for the end effector of the robot to reach that point are solved by implementing some mathematical equation (geometric formula), this is what the Inverse Kinematic method is, a method that converts cartesian coordinate of a robot end effector position to polar coordinate (angles) of each joint.

Because each joint of this robot arm driven by electric motor (servo motor) then the value of angle each joint calculated from geometric equation is the value of how much each these servo motor on the system should rotate in order to make the gripper reach the position or move according to trajectory movement of the end effector (gripper) as we want.

In this project, for the inverse kinematic equation we will use a geometric method to determine the joint angles of the robot arm to reach the desired end effector position. For reducing the complexity, we reduced the problem from 3D to 2D point of view.

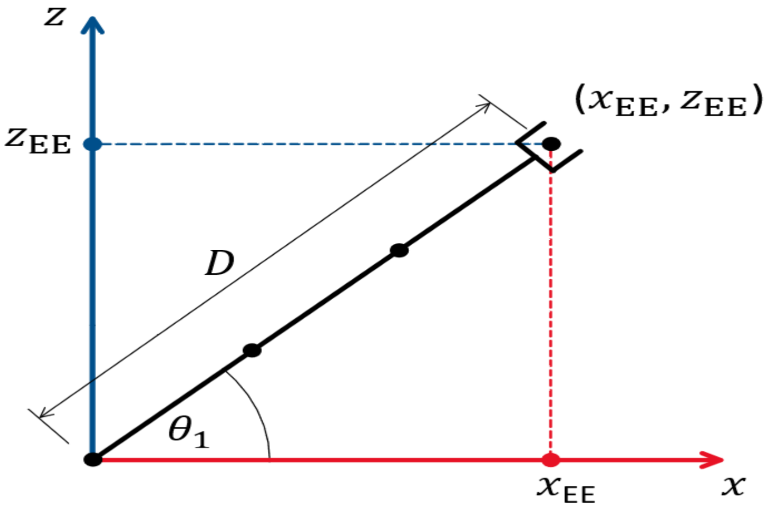


Figure 27 Plan View of the arm

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

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

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

Assuming that we are looking at the robot arm from above, which represent plane (x, z). To make the gripper (end effector) reach desired coordinate position the first joint should rotate at variable declared as "Theta 1", beside the first joint rotation, the arm needs to span to make end effector reach desired coordinate position. Let's symbolize variable how far the arm needs to span as "D". Calculation for Theta 1 and D is shown on picture above.

In the previous section we have implemented formula from to calculate the rotation angle of joint 1 (Theta 1), now we are analyzing another 2D viewpoint to determine the equation to find the other three joint angles (Theta 2, Theta 3, and Theta 4). In short, how much angle joint 2, 3, and 4 needed to rotate to make the arm form span from length that calculated before (D).

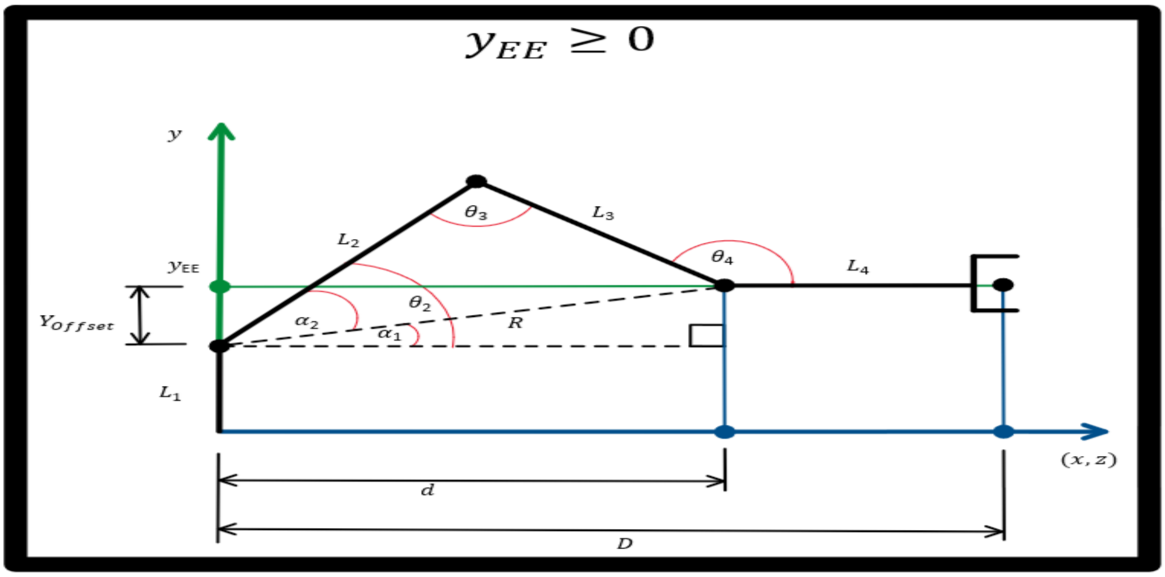


Figure 28 Elevation view

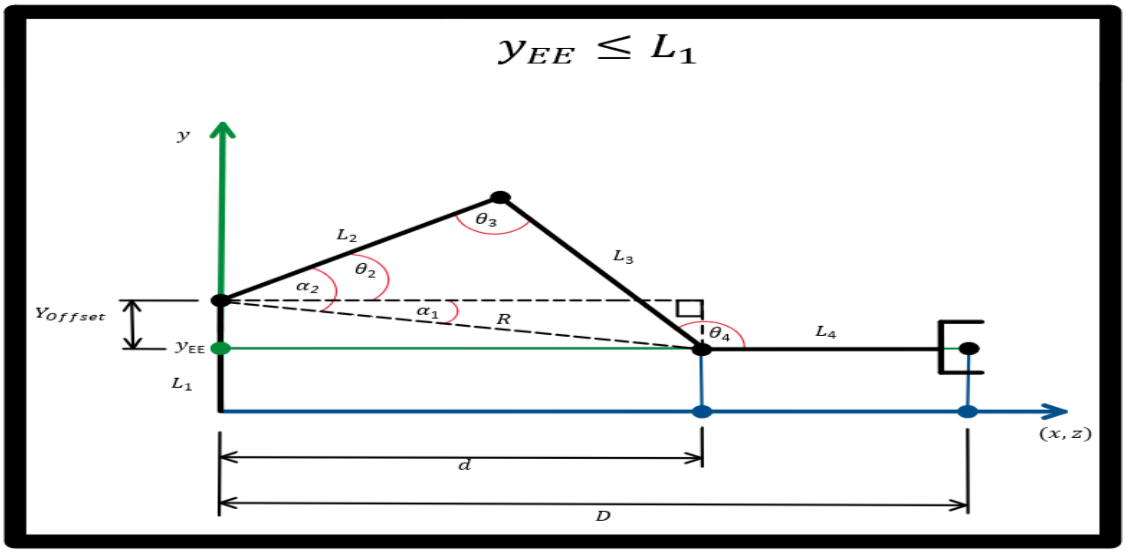


Figure 29 Elevation view

Calculation to find Theta 2 (joint 2 rotation angle) and Theta 4 (joint 4 rotation angle) when position of gripper (end effector) lower than joint 2 position in y axis direction a bit requires different formula than if end effector position higher than joint 2 position.

$$\boldsymbol{y_{EE} \leq L_1}$$

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

$$\theta_4 = 180° - \{[180° - (\alpha_2 + \theta_3)] + \alpha_1\}$$

$$\boldsymbol{y_{EE} \geq L_1}$$

$$d = D - L_4$$

$$Y_{Offset} = y_{EE} - L_1$$

$$R = \sqrt[2]{(d)^2 + (Y_{Offset})^2}$$

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

$$\alpha_1 = cos^{-1}\left(\frac{d}{R}\right)$$

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

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

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

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

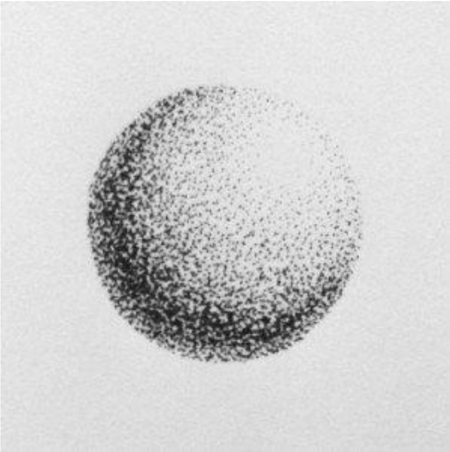
$$\theta_3 = cos^{-1}\left(\frac{{L_2}^2 + {L_3}^2 - R^2}{2 \times L_2 \times L_3}\right)$$

$$\theta_4 = 180° - \{[180° - (\alpha_2 + \theta_3)] - \alpha_1\}$$

This inverse kinematic formula is used only to make link 4 (L4) parallel to the ground/surface, but since this robot is a 4 Degree of Freedom (4DoF) robot arm, there is flexibility to not only determined position where its end effector should move/reach but also the orientation of the end effector, but it requires a different formula.

CHAPTER 6: IMAGE PROCESSING:

Pointillism technique:



Pointillism is a painting technique in which the painter places dots of paint on the canvas in such a way that they blend together into desired forms when viewed from a distance. Pointillism was invented by the French Neo-Impressionist Georges Seurat (1859-1891).

In this brief note, we describe how to use linear programming to construct a pointillist portrait. **[9, 10, 11]**

- **The Photograph**

  We begin the construction of our portrait by converting a $r_{photo} \times C_{photo}$ digital photograph of the subject of our portrait into PGM (portable graymap) format. By doing this, we are converting the photograph into black-and-white, treating each of its $r_{photo}\ C_{photo}$ pixels as an integer between 0 and 255. These integers are grayscale values. A grayscale value of 0 tells us that the pixel is completely black, while a grayscale value of 255 tells us that the pixel is completely white. A grayscale value strictly between 0 and 255 indicates that the pixel is some shade of gray other than completely black or completely white. The larger the grayscale value, the lighter the shade of gray.

Note that at this point, our photograph can be thought of as a list of $r_{photo}C_{photo}$ integers, each one between 0 and 255.

Next, we compress our photograph. (We reduce both the number of integers in our list and the largest integer in our list.)

First, we find positive integers $r_{portrait}$and $C_{portrait}$and a small positive integer k such that

r_{photo}= $K_{portrait}$

C_{photo} = $K_{Cportrait}$.

(If necessary, we crop the photograph.)

Next, we divide the canvas into $r_{portrait}$rows and $C_{portrait}$columns of unit squares, and we partition the pixels of our photograph into $r_{portrait}$ rows and $C_{portrait}$ columns of k × k squares of pixels.

Then, for each row 1 ≤ i ≤ $r_{portrait}$ and column 1 ≤ j ≤ $C_{portrait}$ of our photograph, we compute the mean grayscale value $\mu_{i,j}$of the pixels in square (i, j) and set

$$g_{i,j} = \gamma - [\gamma\mu_{i,j}/256]$$

where γ is a positive integer. By doing this, we are defining $g_{i,j}$ to be the average darkness of square (i, j) of our photograph on a 0 (completely white) to γ (completely black) gray scale.

Note that at this point, our photograph can be thought of as a list of $r_{portrait}$ $C_{portrait}$ integers, each one between 0 and γ.

CHAPTER7: SOFTWARE DESIGN OVERVIEW

Design Overview: The flowchart below lists the ultimate process of interaction between the user and getting a photo. In order to reach this point, we'll need to run three programs:

- Motor Control Capture
- Target Point Pre-Processing
- Main runtime code.

SYSTEM FLOW CHART:

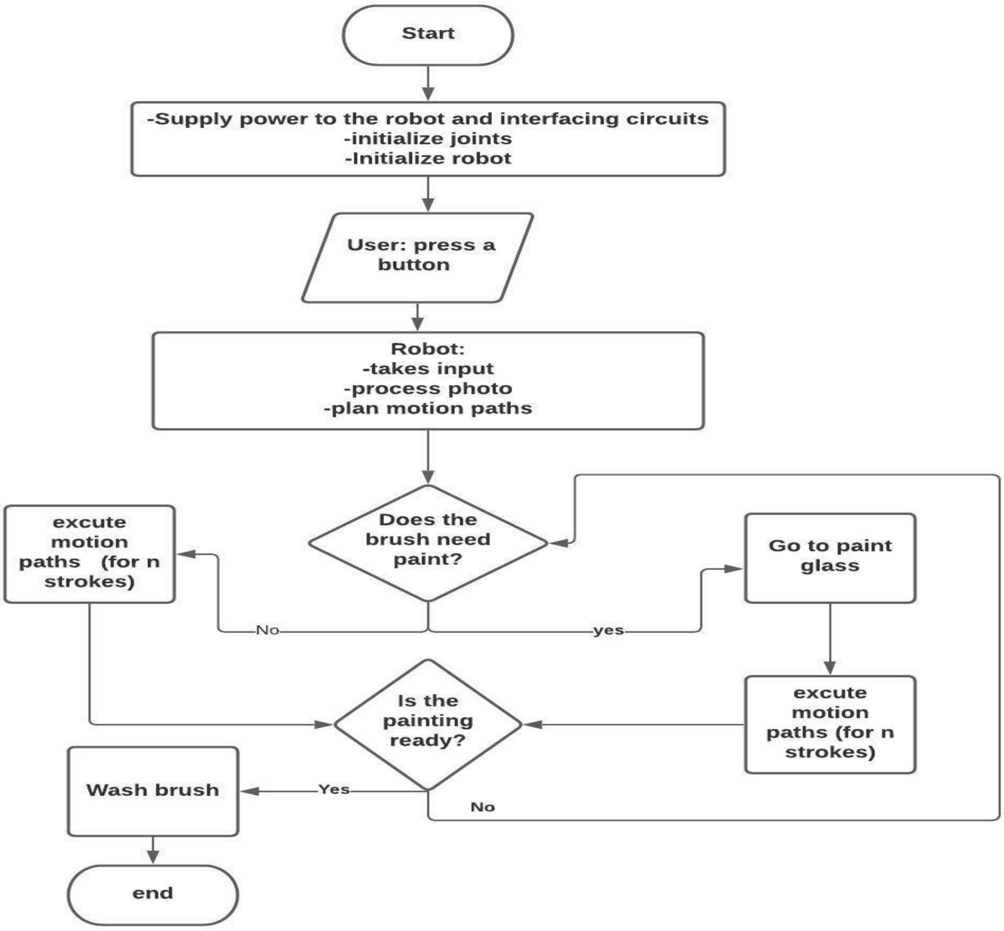


Figure 30 System flowchart

SYSTEM ALGORITHM:

-Start

-Power up robot

-User Input= Press a button (to take a picture)

-Processing photo

-loading a shape sequence (of 4 angles) to position the arm upright

-load the path for reaching the palette

-load path of returning to the upright position

-read the array of the image

-Determine if a pixel is light or dark

-if a pixel is dark then load the matching shape (the output array from the Processing sketch) into the queue

-load a shape that equals the angles

- if thirty shapes have been loaded into the queue

- load the sequence for the path for adding paint to the brush and return

-when the cycle is completed, load path sequence "going to sleep" in the cup of water

-end

MANUFACTURABILITY:

Industrial paint robots have been used for decades
in automotive paint applications. Early paint robots
were hydraulic versions - which are still in use today but are of inferior quality and safety - to the latest electronic offerings.

Originally industrial paint robots were large and expensive, but robot prices have come down to the point that general industry can now afford the same level of automation used by the large automotive manufacturers.

The selection of modern paint robot varies much more in size and payload to allow many configurations for painting items of all sizes.

Painting robots generally have five or six axis motion, three for the base motions and up to three for applicator orientation.

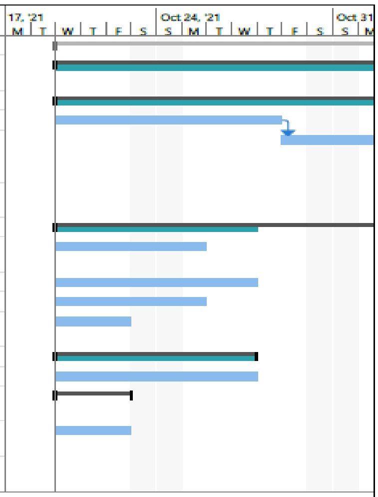
DESIGN ALTERNATIVES AND JUSTIFICATION:

- 5 or 6 DOF arm instead of 4 DOF to allow even better access.
- Using a pencil or a normal pen instead of a painting brush.
- Using multiple colours instead of one colour.

PROJECT EXECUTION

Project Tasks and Gantt chart for semester 1:

ID	Duration	Start	Finish	Predecessors	Notes	Resource Names
0	**72 days**	**Wed 10/20/**	**Thu 1/27/2**			
1	**30 days**	**Wed 10/20/21**	**Tue 11/30/21**			
2	**14 days**	**Wed 10/20/2**	**Mon 11/8/21**			
3	7 days	Wed 10/20/2	Thu 10/28/2		all team	
4	4 days	Fri 10/29/21	Wed 11/3/21	3	Hazem & Salwa	
5	5 days	Thu 11/4/21	Wed 11/10/21	4	All team	
6	**16 days**	**Wed 10/20/2**	**Wed 11/10/2**			
7	4 days	Wed 10/20/21	Mon 10/25/21		Hazem	
8	6 days	Wed 10/20/2	Wed 10/27/2		Salwa	
9	4 days	Wed 10/20/2	Mon 10/25/2		Hazem	
10	3 days	Wed 10/20/21	Fri 10/22/21		Hazem & Salwa	
11	**6 days**	**Wed 10/20/2**	**Wed 10/27/2**			
12	6 days	Wed 10/20/2	Wed 10/27/2		Hazem & Salwa	
13	**3 days**	**Wed 10/20/21**	**Fri 10/22/21**		**All team**	
14	3 days	Wed 10/20/21	Fri 10/22/21		All team	
15	**22 days**	**Mon 11/22/21**	**Tue 12/21/21**			

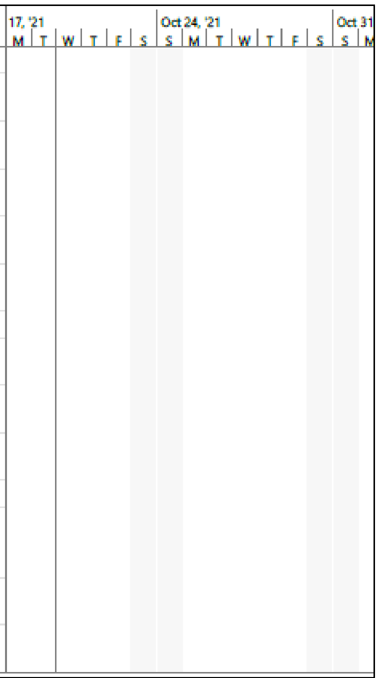


Project: Graduation Project
Date: Wed 1/26/22

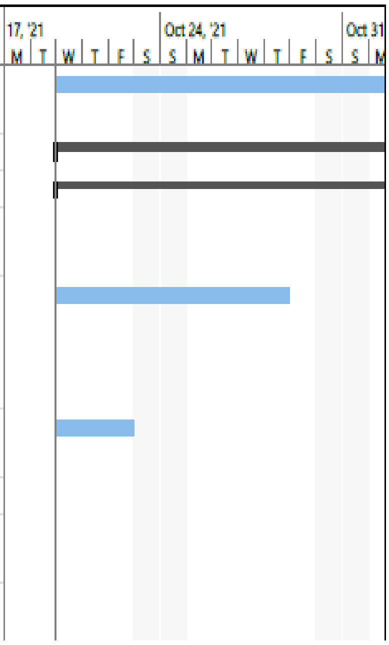
Task
Split
Milestone
Summary
Project Summary
Inactive Task
Inactive Milestone
Inactive Summary
Manual Task
Duration-only
Manual Summary Rollup
Manual Summary
Start-only
Finish-only
External Tasks
External Milestone
Deadline
Progress
Manual Progress

Page 1

ID	Duration	Start	Finish	Predecessors	Notes	Resource Names
16	**21 days**	**Mon 11/22/2**	**Mon 12/20/2**			
17	2 days	Fri 12/17/21	Mon 12/20/21	21	All team	
18	3 days	Mon 11/22/21	Wed 11/24/21		Karim	
19	6 days	Thu 11/25/21	Thu 12/2/21	18	karim & ola & yousef	
20	2 days	Fri 12/3/21	Mon 12/6/21	19	all team	
21	8 days	Tue 12/7/21	Thu 12/16/21	20		
22	**27 days**	**Mon 11/22/2**	**Tue 12/28/2**			
23	4 days	Tue 12/21/21	Fri 12/24/21	18SS,17	all team	
24	10 days	Mon 12/27/21	Fri 1/7/22	23	Yousef	
25	4 days	Mon 1/10/22	Thu 1/13/22	24	yousef & ola	
26	6 days	Fri 1/14/22	Fri 1/21/22	25	ola	
27	4 days	Mon 1/24/22	Thu 1/27/22	26		
28	**27 days**	**Mon 11/22/21**	**Tue 12/28/21**		**yousef & ola**	
29	27 days	Mon 11/22/21	Tue 12/28/21			



ID	Duration	Start	Finish	Predecessors	Notes	Resource Names
30	21 days	Wed 10/20/21	Wed 11/17/21			
31	**45 days**	**Wed 10/20/2**	**Tue 12/21/2**			
32	**45 days**	**Wed 10/20/2**	**Tue 12/21/2**			
33	4 days	Mon 11/22/21	Thu 11/25/21		Hazem & Salwa	
34	7 days	Wed 10/20/21	Thu 10/28/21		Hazem & Salwa	
35	3 days	Wed 10/20/21	Fri 10/22/21		all team	
36	**7 days**	**Mon 12/13/2**	**Tue 12/21/2**			
37	3 days	Mon 12/13/21	Wed 12/15/21		yousef & ola	
38	4 days	Thu 12/16/21	Tue 12/21/21	37	yousef & ola	



CONCLUSION

We are going to present a non-photorealistic rendering techniques for artistic robotic painting, that will be implemented for the processing of digital images into real artworks, our robotic painting system. The system is going to be capable of reproducing an image that has been previously processed through the application of artistic rendering algorithm.

ACKNOWLEDGMENT

Most importantly every member of our group would like to express his/her deepest gratitude for our professors and mentors, for the support and guidance throughout this semester and we would be very grateful to continue under this guidance for the next semester as well. We hope we had met our professor's standards and we wish them health and good fortune.

Thank You

REFERENCES

1. Scalera, L.; Seriani, S.; Gasparetto, A.; Gallina, P. Watercolour Robotic Painting: A Novel Automatic System for Artistic Rendering. *J. Intell. Rob. Syst.* **2018**.

2. Scalera, L.; Seriani, S.; Gasparetto, A.; Gallina, P. Busker robot: A robotic painting system for rendering images into watercolour artworks. *Mech. Mach. Sci.* **2019**, *66*, 1–8.

3. Robotics, Festival of Art and Robotics, Trieste 2018. **https://robotics.gruppo78.it/?lang=en**

4. International RoboticArt Competition (RobotArt). **http://robotart.org/archives/2018/artworks/**

5. "What Is 3D Printing? - Technology Definition and Types." *TWI*, https://www.twi-global.com/technical-knowledge/faqs/what-is-3d-printing.

6. "When Was 3D Printing Invented? the History of 3D Printing -." *BCN3D Technologies*, 10 Dec. 2021, https://www.bcn3d.com/the-history-of-3d-printing-when-was-3d-printing-invented/.

7. "What Is Pla? (Everything You Need to Know)." *TWI*, https://www.twi-global.com/technical-knowledge/faqs/what-is-pla.

8. Deb, Satyaranjan, and Sankha Deb. *Robotics Technology and Flexible Automation*. Tata McGraw-Hill Education, 2010.

9. Bosch, R. 2003. Dominizing Venus. OR/MS Today. April 20-21.

10. Bosch, R., A. Herman. 2004. Continuous line drawings via the traveling salesman problem. Oper. Res. Lett. 32 302-303.

11. Chv´atal, V. 1983. Linear Programming. WH Freeman, New York

APPENDIX

Datasheets:

Data sheet for Arduino:

Tech Specs	Arduino Mega 2560
Board Size	101.52 × 53.3mm
Microcontroller/ Clock Speed	ATmega2560/ 16MHz
SRAM (Main Memory)	8kB
Flash Memory	256kB
EEPROM	4kB
Operating Voltage	+5V
Input Voltage (recommended)	+7~+12V
Output Voltage	+5V, +3.3V
Digital I/O Pins	54
PWM Digital I/O Pins	15
Analog Input Pins	16
Analog Output Pins (DAC)	–
Rated Current per Pin	20mA/Pin
Program Writing Pins	USB Type-B ICSP
Interface	UART I2C SPI

Codes:

```
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
#include "armPositionsAdj.h"
#include "cameraDefinitions.h"

#define OFF 1
#define ON 0
#define TURN_ON -1
#define TURN_OFF 0
#define NUM_SERVOS 5
#define SERVO_BASE 0
#define SERVO_A 1
#define SERVO_B 2
#define SERVO_C 3

#define buttonPin 3
#define led0Pin 4 // 0
#define led1Pin 5 // 1
#define led2Pin 6 // 2
#define led3Pin 7 // 3
#define ledPaintingPin 8 // painting
#define ledRecordingPin 9 // recording
#define ledProcessingPin 10 // processing
#define ledSystemReadyPin 11 // system ready
#define ledTakingPhotoPin 12 // taking photo
#define numLEDs 9
```

#define ledOffset 4

// the center degree pulse length for each motor varies, since the horns cannot be perfectly aligned, this will act as a frame of reference for the motor shaft's position
float SERVO_BASE_FORWARD_DEGREES = 0.0; // 365 // rotating Clockwise = larger pulses
float SERVO_A_UP_DEGREES = 90.0; // 357 // moving right = larger pulses
float SERVO_B_UP_DEGREES = 0.0; // 360 // moving right smaller pulses
float SERVO_C_UP_DEGREES = 0.0; // 363 // moving right = smaller pulses
int SERVO_BASE_FORWARD_PULSE = 365; // 0 degrees // rotating Clockwise = larger pulses
int SERVO_A_UP_PULSE = 357; // 90 degrees // moving right = larger pulses
int SERVO_B_UP_PULSE = 360; // 0 degrees // moving right smaller pulses
int SERVO_C_UP_PULSE = 363; // 0 degrees // moving right = smaller pulses

#define ANALOG_PIN A0