Collage of Engineering
Salahaddin University-Erbil

Design and Implementation of 3D Robotic Arm

Research Project

Submitted to the College of Engineering In Partial Fulfillment of the Requirements for the degree of BSc in Computer and Control Engineering in the Department of Electrical Engineering
October 30, 2019

By:

Ali Ahmed Salih
Ahmed Ali Ahmed
Asher Ghazi Ashad
Ayub Dlshad Jawad
Abdullah Muhammed Abdullah

SUPERVISOR'S CERTIFICATION

I certify that this Project, titled 'Design and implementation of 3D of robotic arm' and presented by (Ahmad Ali , Asher Ghazi , Abdullah Muhammed , Ali Ahmed and Ayub Dlshad) was prepared under my supervision at the University of Salahaddin – Erbil as a partial requirement for the degree of Bachelors of Science in Electrical Engineering.

Signature:

Name: Mr. Ahmed Kh. Ahmed

Date: 9/6/2020

Acknowledgment:

Special thanks to our supervisor Mr. Ahmed Khalid for everything he did for us. And we would like also to thank our families and colleagues for all the supports to create this project.

ABSTRACT

The purpose of our research is to model design, implement and control a three DOF Robot arm with using of Servomotors. The Three-Link Planar arm is controlled by a Arduino Mega Microcontroller and its main function is to sketch an image by processing it through Matlab and the Algorithm of Image Processing. The forward and inverse kinematics analysis is presented in our paper and explained, for angular and position configurations. Although our research is presented for Academic purposes, yet it can be suitable for many industrial applications.

Table of Contents

LIST OF FIGURES

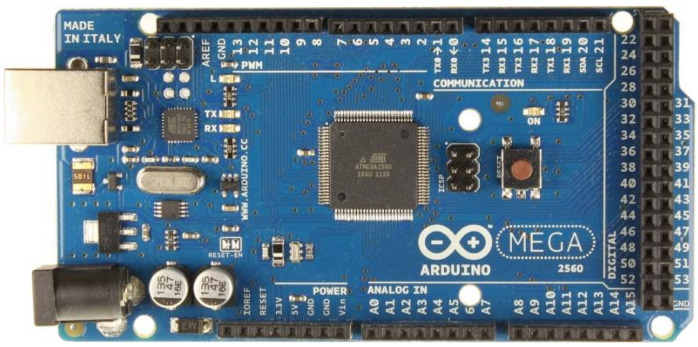
LIST OF TABLES

DEFINITIONS

Servomotor: is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration.

Fig(1). Servomotor.

Arduino: is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices.

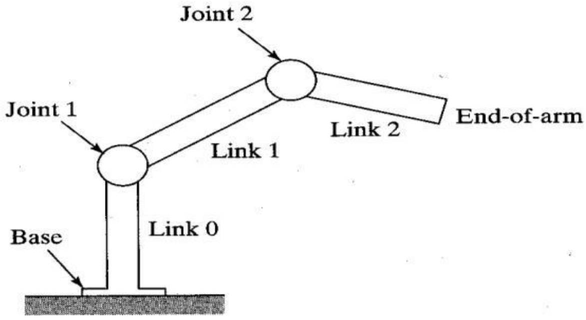


Fig(2) Arduino MEGA.

PC (Personal Computer): is a multi-purpose computer whose size, capabilities, and price make it feasible for individual use.

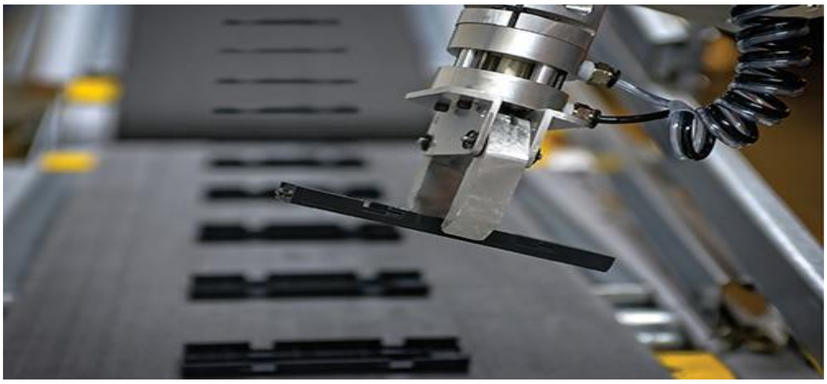
Fig(3). Personal Computer.

Links and Joints: The links of such a manipulator are connected by joints allowing either rotational motion translational (linear) displacement. The links of the manipulator can be considered to form a kinematic chain.



Fig(4). Links and Joints.

End-Effectors: In robotics, an end effector is the device at the end of a robotic arm, designed to interact with the environment. The exact nature of this device depends on the application of the robot. In the strict definition, which originates from serial robotic manipulators, the end effector means the last link of the robot.



Fig(5). End-Effectors.

ACRONYMS

- DOF = Degree of Freedom.
- D-H Parameters = Denavit and Hartenberg Parameters.
- PC = Personal Computer.
- ISO = International Organization for Standardization.
- CNC = Computer Numerical Control.
- PID = Proportional Integral Derivative.
- PWM = Pulse with Modulations.

Chapter1

1.1 Definition of Manipulator and planar

a robotic manipulator is a devise working under human control to handle materials with or without direct human contact [6], More comprehensively, the International Organization for Standardization (ISO) defines a manipulator to be “a machine, the mechanism of which usually consists of a series of segments, jointed or sliding relative to one another, for the purpose of grasping and/or moving objects (pieces or tools) usually in several degrees of freedom [1]. Robot manipulators can also be classified according to their nature of motion. A rigid body is said to perform a planar motion if all particles in the body describe plane curves that lie in parallel planes [5] .

1.2 Literature Review

1- Al-mamon university college ,computer engineering techniques department, Iraq ,Baghdad.
Correspondence Auther : Ahmed A.radhi
Accepted date :28 july 2018
Design and implementation of wireless mobile robotic arm controller based smart phone and embedded system via Bluetooth.
Abstract: The purpose of this research is to design and implement a pick and place objects mobile robotic arm based android controlling via Bluetooth by using Arduino microcontroller.
2- department of electrical and electronics engineerin university of Maiduguri,Maiduguri, Nigeria
Accepted date: march 2019
By : A.B.Buji,Y,P. Mshelia,A.G.Ibrahim and M.A.Sarki.
Model design ,simulation and control of a robotic arm using PIC 16F877A MICROCONTROLLER.
Abstract: This paper focuses on model design, simulation and control of a five degree of freedom (DoF) robotic arm using servo motors. The robotic arm is controlled by a PIC 16F877A microcontroller and its main function is to generate pulse width modulation (PWM) signs which are to the servo motors for achieving the desired rotation angle.

3- Y.D.Patel (1),P.M.George (2)
(1) associate professor ,department of mechanical engineering ,A.D.patel institute of technology,new vallabh vidyanagar, gujarta ,india.
(2) professor,department of mechanical engineering,birla vishvakarma mahavidyalaya engineering collage , vallabh vidyanagar,india.
Accepted date:sep-2013.
PERFORMANCE MEASUREMENT AND DYNAMIC ANALYSIS OF TWO DOF ROBOTIC ARM MANIPULATOR.
Abstract: Forward and inverse kinematic analysis of 2DOF robot is presented to predict singular configurations. Cosine function is used for servo motor simulation of kinematics and dynamics using Pro/Engineer. The significance of joint-2 for reducing internal singularities is highlighted. Performance analysis in terms of condition number, local conditioning index and mobility index is carried out for the manipulator.
Dynamic analysis using Lagrangian's and Newton's Euler approach is worked out analytically using MATLAB and results are ploted for their comparison.
4- Mashad Uddin Saleh (1), Gazi Mahamud Hasan (2), Mohammad Abdullah Al Shohel (3), Md. Abul Hasnat Ferdous (4)", Biswajit Biswas Dipan (5)
Alumni, Dept. of EEE, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh (1,5).
MSc Student, Dept. of EEE, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh (2,3).
BSc Student, Dept. of EEE, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh (4).
Accepted date:10,October 2013.
DESIGN AND IMPLEMENTATION OF A SIMPLE, LOW-COST ROBOTIC ARM.
Abstract: This paper describes the design of a 3-DOF robotic arm very much suitable for industrial purposes. The three DOF can perform lateral, vertical and grabbing movement profoundly.
5- department of mechatronic engineering technological university thanlyin, Myanmar
By : war war naing , kyi zar aung , aung thike.
Accepted date : 2018
Abstract: The purpose of this project is to eliminate the manual control for pick and place system. The robot arm is designed with three joints, two links and three Dc motors. Arduino microcontroller is used to generate the required angular position of the robot joints.

Chapter 2

2.1 Manipulator Kinematic Model

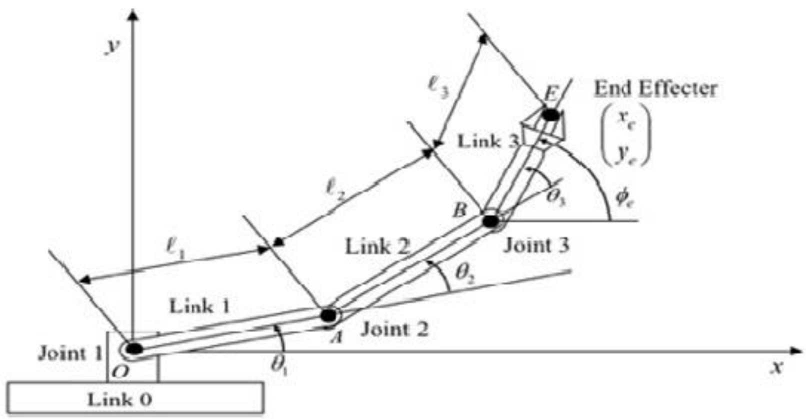
Kinematics is the science of motion that treats motion without regard to the forces which cause it. Within the science of kinematics, one studies position, velocity, acceleration, and all higher order derivatives of the position variables (with respect to time or any other variable(s)). It describes the relationship between robot end-effector position and orientation in space and manipulator joint angles. Manipulators consist of nearly rigid links, which are connected by joints that allow relative motion of neighboring links. These joints are usually instrumented with position sensors, which allow the relative position of neighboring links to be measured. In the case of rotary or revolute joints, these displacements are called joint angles. The number of degrees of freedom that a manipulator possesses is the number of independent position variables that would have to be specified in order to locate all parts of the mechanism [4].

2.2 Planar Kinematics

To begin with, we will restrict ourselves to a class of robot mechanism that work within a plane, i.e. Planar Kinematics. Planar kinematics is much more tractable mathematically, compared to general three-dimensional kinematics [2].

2.2.1 Planar Kinematics of Serial Link Mechanism [2]

The three DOF planar robot arm shown in Fig(6) The arm consists of one fixed link and three movable links that move within the plane



Fig(6). Three DOF Planar Robot with three revolute joints.

To describe this robot arm, a few geometric parameters are needed. First, the length of each link is defined to be the distance between adjacent joint axes. Let points *O, A*, and *B* be the locations of the three joint axes, respectively, and point E be a point fixed to the end-effecter. Then the link lengths are l1=OA , l2=AB , l3=BE. Let us assume that Actuator 1 driving link 1 is fixed to the base link (link 0), generating angle θ1 , while Actuator 2 driving link 2 is fixed to the tip of Link 1, creating angle θ2 between the two links, and Actuator 3 driving Link 3 is fixed to the tip of Link 2, creating angle θ3 , as shown in the figure. Since this robot arm performs tasks by moving its end-effecter at point E, we are concerned with the location of the end-effecter. To describe its location, we use a coordinate system, O-*xy*, fixed to the base link with the origin at the first joint, and describe the end-effecter position with coordinates *Xe* and *Ye* . We can relate the end-effecter coordinates to the joint angles determined by the three actuators by using the link lengths and joint angles defined above:

$$X_e = l_1 \cos\theta_1 + l_2 \cos(\theta_1+\theta_2) + l_3 \cos(\theta_1+\theta_2+\theta_3).$$

$$Y_e = l_1 \sin\theta_1 + l_2 \sin(\theta_1+\theta_2) + l_3 \sin(\theta_1+\theta_2+\theta_3).$$

This three DOF robot arm can locate its end-effecter at a desired orientation as well as at a desired position. The orientation of the end-effecter can be described as the angle the centerline of the end-effecter measured from the positive x coordinate axis. This end-effecter orientation φe is related to the actuator displacements as $\varphi_e=\theta_1+\theta_2+\theta_3$. The above three equations describe the position and orientation of the robot end-effecter viewed from the fixed coordinate system in relation to the actuator displacements. In general, a set of algebraic equations relating the position and orientation of a robot end-effecter, or any significant part of the robot, to actuator or active joint displacements, is called Kinematic Equations, or more specifically, Forward Kinematic Equations in the robotics literature.

2.2.2 Inverse Kinematics of Planar Mechanism

The vector kinematic equation derived in the previous section provides the functional relationship between the joint displacements and the resultant end-effecter position and orientation. By substituting values of joint displacements into the right-hand side of the kinematic equation, one can immediately find the corresponding end-effecter position and orientation. The problem of finding the end-effecter position and orientation for a given set of joint displacements is referred to as the direct kinematics problem. This is simply to evaluate the right-hand side of the kinematic equation for known joint displacements. In this section, we discuss the problem of moving the end-effecter of a manipulator arm to a specified position and orientation. We need to find the joint displacements that lead the end-effecter to the specified position and orientation. This is the inverse of the previous problem.

2.3 Applications of planar arm robot

1- Robot for writing or sketching.
2-Robot for laser graving or drill graving like CNC machines.
3- 3D printer.
4- Epson T3 Scara Robot.

Chapter 3

3.1 Robot analysis

The analysis of a robot consists of determining D-H parameters, calculating robot kinematics and dynamics, and controlling the robot via a control scheme. The control scheme may be PID, Fuzzy, Microcontroller or Visual Control.

3.2 Forward kinematics

Calculating the position and orientation of the end-effector in terms of the joint variables is called as forward kinematics. In order to have forward kinematics for a robot mechanism in a systematic manner, one should use a suitable kinematics model. Denavit-Hartenberg method that uses four parameters is the most common method for describing the robot kinematics[3].

we obtain the general form:

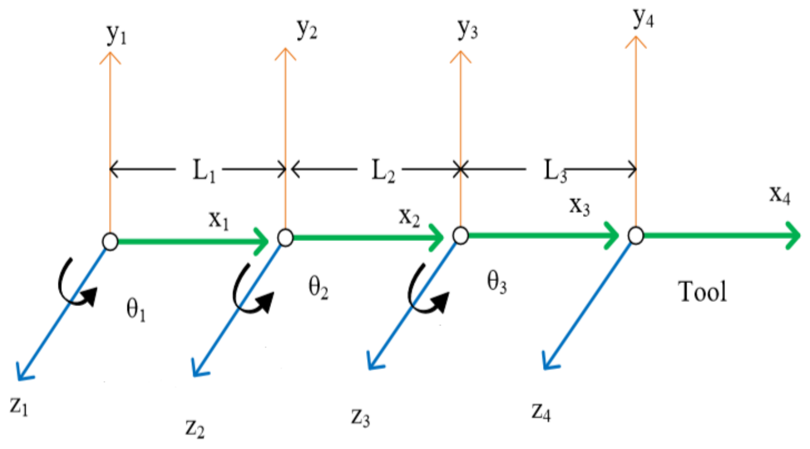
$$ {}^{i-1}_{\ \ i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Where the four quantities, αi, ai, di, and θi are the parameters of link i and joint i. The various parameters in previous equation are given the following names:

- ai (Length) is the distance from zi to zi+1, measured along zi.
- αi (Twist), is the angle between z i and z i+1 measured about xi.
- di (Offset), is the distance from xi to xi+1 measured along zi.
- θi (Angle), is the angle between xi and xi+1 measured about zi.

3.2.1 Determining D-H Parameters

D-H parameters table is a notation developed by Denavit and Hartenberg, is intended for the allocation of orthogonal coordinates for a pair of adjacent links in an open kinematic system. It is used in robotics, where a robot can be modeled as a number of related solids (segments) where the D-H parameters are used to define the relationship between the two adjacent segments. The first step in determining the D-H parameters is locating links and determines the type of movement (rotation or translation) for each link[4].
Using D-H parameters of three-link planar manipulator defined in the previous steps in Table 1, and coordinate robot frame assignment of Robot Arm shown in Fig(7). Robot model in addition to previously determined D-H parameters contains physical parameters which is using in the calculation of the dynamics movement[4].



Fig(7). Coordinate Frame Assignment of Robot Arm.

i	αi-1	$\boldsymbol{ai-1}$	*di*	Θ
1	*0*	*0*	*0*	$\theta 1$
2	*0*	*L1*	*0*	$\theta 2$
3	*0*	*L2*	*0*	$\theta 3$

Table 1. D-H PARAMETERS OF THREE-LINK PLANAR MANIPULATOR.

- Because the arm lies in a plane with all Z axes parallel, there are no link offsets—all d_i are zero.
- All joints are rotational, so when they are at zero degrees, all X axes must align.
- Note that, because the joint axes are all parallel and all Z axes are taken as out of the paper, all α_i are zero.

$$ {}^{0}_{1}T=\begin{bmatrix} c1 & -s1 & 0 & 0 \\ s1 & c1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^{1}_{2}T=\begin{bmatrix} c2 & -s2 & 0 & L1 \\ s2 & c2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{2}_{3}T=\begin{bmatrix} c3 & -s3 & 0 & L2 \\ s3 & c3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^{3}_{tool}T=\begin{bmatrix} 1 & 0 & 0 & L3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{0}_{3}T={}^{0}_{1}T * {}^{1}_{2}T * {}^{2}_{3}T $$

$$ {}^{0}_{3}T=\begin{bmatrix} c123 & -s123 & 0 & L1c1+L2c12 \\ s123 & c123 & 0 & L1s1+L2s12 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Finally, to obtain ${}^{0}_{tool}T$, multiply ${}^{0}_{3}T$ and ${}^{3}_{tool}T$ then get:

$$ {}^{0}_{tool}T = \begin{bmatrix} c123 & -s123 & 0 & L1c1 + L2c12 + L3c123 \\ s123 & c123 & 0 & L1s1 + L2s12 + L3s123 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Where,

$$ c1 = cos(\theta 1) $$

$$ s1 = sin(\theta 1) $$

$$ c12 = cos(\theta 1 + \theta 2) $$

$$ s12 = sin(\theta 1 + \theta 2) $$

$$ c123 = cos(\theta 1 + \theta 2 + \theta 3) $$

$$ s123 = sin(\theta 1 + \theta 2 + \theta 3) $$

3.3 Inverse kinematics

Inverse kinematics can be computed from given Cartesian position and orientation of end effector and reverse of this would yield forward kinematics. Which is nothing but finding out end effector coordinates and angles from given joint angles. Forward kinematics of serial manipulators gives exact solution while inverse kinematics yields number of solutions. The complexity of inverse kinematic solution arises with the increment of degrees of freedom. Therefor e it would be desired to adopt optimization techniques. Although the optimization techniques gives number of solution for inverse kinematics problem but it converses the best solution for the minimum function value.

Inverse kinematics solution of robot manipulators has been considered and developed different solution scheme in last recent years because of their multiple, nonlinear and uncertain solutions. There are different methodologies for solving inverse kinematics for example iterative, algebraic and geometric (etc..) solutions[7].

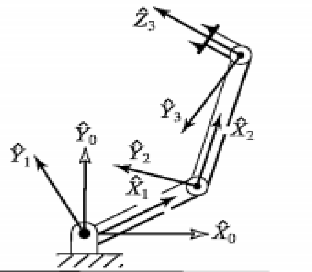
3.3.1 Algebraic

In this thesis for solving kinematic equations, we will consider algebraic method to the solution of a simple planar three-link manipulator[4].

3.3.1.1 Algebraic Solutions

Consider the three-link planar manipulator introduced in our thesis before. It is shown with its link parameters in Fig(8). Following the method of Algebraic solutions, we can use the link parameters easily to find the Inverse kinematic equations of this arm:

$$
{}^{0}_{3}T=\begin{bmatrix} c123 & -s123 & 0 & L1c1+L2c12 \\ s123 & c123 & 0 & L1s1+L2s12 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$



i	$\alpha_i - 1$	$a_i - 1$	d_i	θ_i
1	0	0	0	θ_1
2	0	L_1	0	θ_2
3	0	L_2	0	θ_3

Fig(8). Three-link planar manipulator and its link parameters.

To focus our discussion on inverse kinematics, we will assume that the necessary transformations have been performed so that the goal point is a specification of the wrist frame relative to the base frame, that is ${}^{B}_{W}T$, Because we are working with a planar manipulator. We will assume a transformation with the structure.

$$ {}^{B}_{W}T = \begin{bmatrix} c\phi & -s\phi & 0 & x \\ s\phi & c\phi & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

All attainable goals must lie in the subspace implied by the structure of equation above,

We arrive at a set of four nonlinear equations that must be solved for θ_1, θ_2 and θ_3:

$$ c_\phi = c123 $$

$$ s_\phi = s123 $$

$$ x = L1c1 + L2c12 $$

$$ y = L1s1 + L2s12 $$

We now begin our algebraic solution of equations above, square both and add them, we obtain

$$ x^2 + y^2 = L1^2 + L2^2 + 2 * L1L2c2 $$

where we have made use of

$$ c12 = c1c2 - s1s2 $$

$$ s12 = c1s2 + s1c2 $$

We obtain

$$ c2 = \frac{x^2 + y^2 - L1^2 - L2^2}{2L1L2} $$

In order for a solution to exist, the right-hand side of equation above must have a value between —1 and 1. In the solution algorithm, this constraint would be checked at this time to find out whether a solution exists. Physically, if this constraint is not satisfied, then the goal point is too far away for the manipulator to reach.

Assuming the goal is in the workspace, we write an expression for as

$$ s2 = \pm\sqrt{1 - c^2} $$

Finally, we compute θ_2 using the two-argument arctangent routine

$$ \theta_2 = Atan2(s2, c2) $$

The choice of signs in S_2 equation corresponds to the multiple solution in which we can choose the "elbow-up" or the "elbow-down" solution.

In determining θ_2 we have used one of the recurring methods for solving the type of kinematic relationships that often arise, namely, to determine both the sine and cosine of the desired joint angle and then apply the two-argument arctangent.

Having found we can solve for θ_1

$$X=k1c1-k2s2$$

$$Y=k1s1+k2c1$$

$$K1=L1+L2c2$$

$$K2=L2s2$$

In order to solve an equation of this form, we perform a change of variables. Actually, we are changing the way in which we write the constants k1 and k2.

if

$$r=+\sqrt{k1^2+k2^2}$$

and

$$y=Atan(k2,k1)$$

then

$$k1=r\ cosy$$

$$k2=r\ siny$$

Equations $X=k1c1-k2s2$ and $Y=k1s1+k2c1$ can now be written as

$$\frac{x}{r}=cosy\ cos\theta1-siny\ sin\theta1$$

$$\frac{y}{r}=cosy\ sin\theta1+siny\ cos\theta1$$

so

$$cos\ (y+\theta1)=\frac{x}{r}$$

$$sin\ (y+\theta1)=\frac{y}{r}$$

Using the two-argument arctangent, we get

$$Y + \theta 1 = Atan2\left(\frac{y}{r}, \frac{x}{r}\right) = Atan2(y, x)$$

And so

$$\theta 1 = Atan2(y, x) - Atan2(k2, k1)$$

$$\theta 1 + \theta 2 + \theta 3 = Atan2\left(s\phi, c\phi\right) = \phi$$

From this, we can solve for 03, because we know the first two angles. It is typical with manipulators that have two or more links moving in a plane that, in the course of solution, expressions for sums of joint angles arise. In summary, an algebraic approach to solving kinematic equations is basically one of manipulating the given equations into a form for which a solution is known.

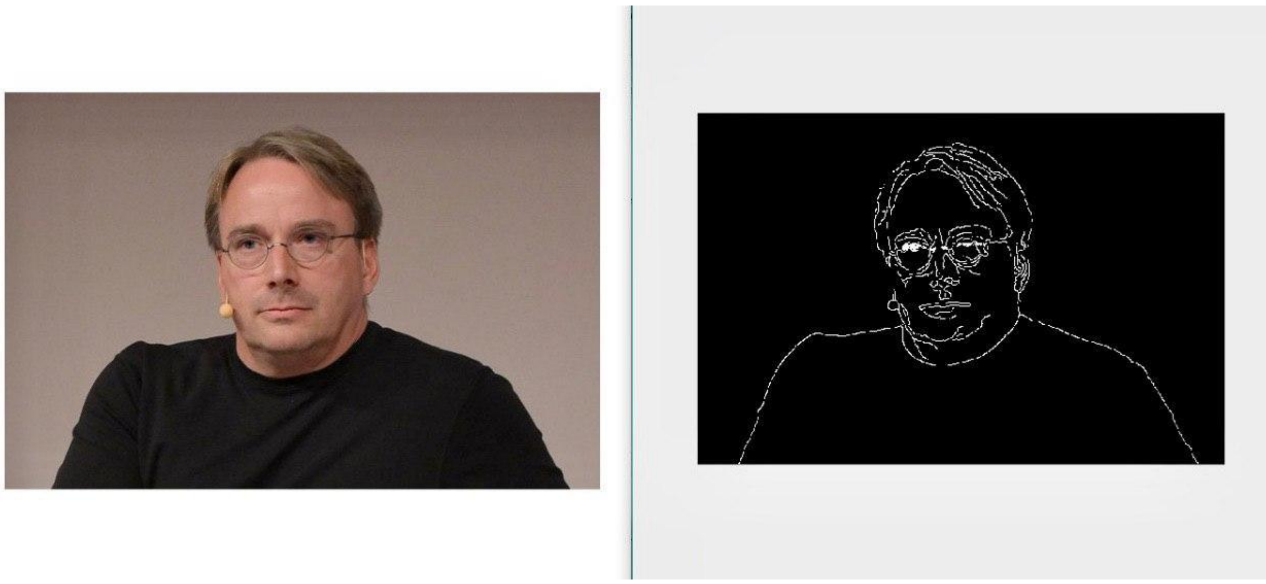
Chapter 4

4.1 Software Packages

With Matlab Support Package for Arduino Hardware, we can use Matlab to interactively communicate with an Arduino board. This package allows using an Arduino board that we connected to the computer to perform Digital Inputs and Outputs, (and command motors) from Matlab, so it could enable us to control our servomotors that are connected with the Arduino board, which are attached to the robot links with digital and PWM outputs. This package connects Arduino board to the Matlab with a serial I/O port, which is the same port that used before to connect the Arduino board with the computer[8].

4.2 Image Processing

Whenever we draw an image, we try to find out the important edges or features that needs to be drawn, for that purpose, we use Canny edge detection Algorithm. If we want to draw all formats of images, we must convert images in other formats to gray scale images. The Matlab 2017 software has built in commands to convert images from one format to another. The edge detection is a famous image processing technique and there are several edge detection Algorithms available, we tried with various Algorithms and found that Canny edge detection Algorithm was best suited for our task and project. Because the Canny edge detection Algorithm finds out real and localized edges without being affected by noise[9].



Fig(9). An example of Image Processing.

4.3 Source Code

```
%the canny edge detection algorithm to process the image
u=imread('imge.png');
i=rgb2gray(u);
w=edge(i,'canny',[.05,.20]);

EyeDetect = vision.CascadeObjectDetector('EyePairBig');
if step(EyeDetect,u)
BB=step(EyeDetect,u);
l=round(BB(1,4)/3);
for s=(BB(1,2)+l):(BB(1,2)+2*l)
    for t=BB(1,1):(BB(1,1)+BB(1,3))
        if i(s,t)<72
           w(s,t)=1;
        end
    end
end
end

I=imrotate(w,180);
imshow(w);
impixelinfo; % creating matrix of pixels of the image


a=arduino('COM3'); %creating an object for the arduino and
% spesifing the port that arduino connected through it to the computer
servoAttach(a,7); % attaching the tool tip servo motor
servoAttach(a,8);% attaching the first servo motor
servoAttach(a,9);% attaching the second servo motor
servoAttach(a,10);% attaching the third servo motor
set(0,'RecursionLimit',2000);
servoWrite(a,7,40); % pull up the pen
pause(0.02); % delay
global F S T;
F=servoRead(a,9); % reading the position of the first servo in degrees
S=servoRead(a,8); % reading the position of the second servo in degrees
T=servoRead(a,10); % reading the position of the third servo in degrees
pause(0.1); % delay
e=size(I);
%the loop checks all pixels of the image if they are 1 or 0
%if there is a 1 the function 'draw' will be called to draw that pixel.
for p=2:1:e(1,1)
    for t=2:1:e(1,2)
       if I(p,t)==1
           I=draw(a,I,p,t);
       end
    end
end
%the function draw checks 8 neighboring pixels if any one of
%them is also 1 then the pen reaches that pixel without lifting the pen up
%and deletes the previous pixel to avoid repetition
%The function repeats itself recursively and creates smooth lines.
function I=draw(a,I,m,n)
if                    I(m-1,n-1)==1||I(m,n-1)==1||I(m-1,n)==1||I(m+1,n-1)==1||I(m-
1,n+1)==1||I(m,n+1)==1||I(m+1,n)==1||I(m+1,n+1)==1
   reach(a,m,n,size(I));
   servoWrite(a,7,100); %push down the pen to write
    pause(0.01); %delay
   r=size(I);
   I(m,n)=0; %deletes the wrote pixel
```

```
  if m-1>0&&n-1>0&&m<r(1,1)&&n<r(1,2)
      for i=m-1:m+1
          for j=n-1:n+1
              if I(i,j)==1
                 I=draw(a,I,i,j);
              end
          end
      end
   end
end
  servoWrite(a,7,40);
  pause(0.01);
end
%the function 'reach' is sets the angles to the servos
function reach(a,m,n,s)
global F S T;
[th1,th2,th3]=calct(m,n,s(1,1),s(1,2));
servoAngle(a,8,th1);
servoAngle(a,9,th2);
servoAngle(a,10,th3);
F=th1;
S=th2;
T=th3;
end
%the function 'calct' calculates the angles 'theta 1, theta 2, and theta 3'
function [th1,th2,th3]=calct(pt_y,pt_x,n_y,n_x)
y=(pt_y/n_y)*30 + 10;  %calculating y with scaling and shifting
x=(pt_x/n_x)*20 + 10;  %calculating X with scaling and shifting
cos_th2=(((x^2)+(y^2))-338)/338; %calculating Cosine(th 2)
sin_th2=sqrt(1-(cos_th2)^2); %calculating sine of th2
th2=acosd(cos_th2); % calculating th2 in degrees
th2=round(th2);    %rounding value of th2
k1=13+(13*cos_th2); % K1
k2=13*sin_th2;      %K2
lamda=atan2d(k2,k1); %calculating lamda
th1=(atan2d(y,x))-lamda; %calculating th1 in degrees
th1=round(th1); %rounding value of th1
th3=180-th1-th2; %calculating th3 in degrees
end
%function "servoangle" sets the angles of the servos step by step with an %each
step (0.1 degree) to move the pen from the required pixel to the next %pixel
with an ease to draw smooth lines.

function servoAngle(a,p,n)
global F S T;
if p==8
    if abs(F-n)>1
        if n>E
            for i=F:0.1:n
                servoWrite(a,p,i);
            end
            pause(0.5);
        else
             for i=n:0.1:E
                 servoWrite(a,p,F-i+n);
             end
             pause(0.5);
        end
    else
        servoWrite(a,p,n);
    end
elseif p==9
    if abs(S-n)>1
```

```
        if n>S
            for i=S:0.1:n
                servoWrite(a,p,i);
            end
            pause(0.5);
        else
             for i=n:0.1:S
                 servoWrite(a,p,S-i+n);
             end
             pause(0.5);
        end
    else
        servoWrite(a,p,n);
    end
end
else
    if abs(T-n)>1
        if n>T
            for i=T:0.1:n
                servoWrite(a,p,i);

            end
            pause(0.5);
        else
             for i=n:0.1:T
                 servoWrite(a,p,T-i+n);
             end
             pause(0.5);
        end
    else
        servoWrite(a,p,n);
    end
end
end
```

4.4 Result and Conclusion

The (canny edge detection) converts the required image that need to be drawn as in the Fig(9). We get the matrix of pixels of the image, and then the program starts checking the matrix and finding the 1s that they represent the white pixels and need to be drawn, with the help of inverse kinematics equations we can easily find the angles of joints and set the angles to the servos to reach the required pixels and puts the pen down then it checks neighboring 8 pixels, if it finds a 1 it reaches that point without lifting the pen up and deletes the previous pixel to avoid repetition and creating smooth lines. Then it completes other parts of image until it creates the whole image.

To get the aim of our project there will always be a set of objectives, we have designed and implemented the planar arm robot in aims of academic purposes, which we can take it to the next upper levels of industrial manufactories, in order to make vast and critical changes in the industrial world.

References

- Manipulating Industrial Robots—Vocabulary, ISO 8373, 1994 [1].
- Introduction to Robotics, H. Harry Asada [2].
- Introduction to Robotics Mechanics and Control Third Edition John J. Craig [4].
- Robot analysis by LUNG-WEN TSAI [5].
- irtual Manufacturing By Wasim Ahmed Khan, Abdul Raouf, Kai Cheng [6].
- J J. DENAVIT, R.S. HARTENBERG, "A Kinematics Notion For Lower-Pair Mechanisms Based On Atrices"[3].
- Funda J, Taylor R, Paul R. On homogeneous transforms, quaternions, and computational efficiency. IEEE Transactions on Robotics and Automation[7].
- https://www.mathworks.com/matlabcentral/fileexchange/47522-matlab-support-package-for-arduino-hardware[8].
- Ranita, Biswas. and Jaya, Sil., 'An Improved Canny Edge Detection,Algorithm Based on Type-2 Fuzzy Sets', Proceedings of Procedia Technology, 2012, Vol. 4, pp.820 – 824[9].