Journal of Science and Technology (JST)
Volume 3, Issue 5, Sep - October 2018, PP 31-42
www.jst.org.in *ISSN: 2456 - 5660*

Algorithmic Approach of Estimating And Selection of Inverse Kinematics For Planar Robotic Arm

Rhenius J.Bright[1], S.Parthasarathi[2]
[1](Mechatronics Engineering Thiagarajar College of EngineeringMadurai-625015, India)
[2](Mechatronics Engineering Thiagarajar College of EngineeringMadurai-625015, India)
[1]rheniusjbright@gmail.com [2]parthasarathi_s@tce.edu

Abstract : *In this project, I researched the kinematic analysis of robot arm. The kinematic analysis is the relationships between the positions, velocities, and accelerations of the links of a manipulator. The kinematics is divided into two types, forward kinematics and inverse kinematics. In forward kinematics, the length of each link and the angle of each joint is given and we have to calculate the position of any point in the work volume of the robot. In inverse kinematics, the length of each link and position of the point in work volume is given and we have to calculate the angle of each joint.The forward kinematic analysis is not difficult to solve. It is solved by using simple homogeneous matrices. On the other hand, the inverse kinematics is so hard to solve and it will be harder if we increase the degrees of freedom. There are different methods to solve the inverse kinematics. The analytic method and Jacobian method are well-known.In the project, I used the analytic method. In the thesis, I designed a prototype robot arm with 3 freedom degrees. User interface application was created in the personal computer and the data was sent to the hardware application board by using serial communication cable. The program that runs over the application board receives the data and operates. So the end-effecter can be moved to the position we want to go.*

Keywords - *Inverse Kinematics, Planar Robotic Arm, Angle Calculation, 3 Degrees of Freedom.*

I. INTRODUCTION

A robotic arm is a type of mechanical arm, usually programmable, with similar functions to a human arm; the arm may be the sum total of the mechanism or may be part of a more complex robot. The links of such a manipulator are connected by joints allowing either rotational motion (such as in an articulated robot) or translational (linear) displacement. The links of the manipulator can be considered to form a kinematic chain. The terminus of the kinematic chain of the manipulator is called the end-effector and it is analogous to the human hand.Manipulators are composed of an assembly of links and joints. Links are defined as the rigid sections that make up the mechanism and joints are defined as the connection between two links. The device attached to the manipulator which interacts with its environment to perform tasks is called the end-effector.The end-effector, or robotic hand, can be designed to perform any desired task such as welding, gripping, spinning etc., depending on the application. For example, robot arms in automotive assembly lines perform a variety of tasks such as welding and parts rotation and placement during assembly. In some circumstances, close emulation of the human hand is desired, as in robots designed to conduct bomb disarmament and disposal.The degrees of freedom of a system is the number of independent motions that are allowed to the body or, in case of a mechanism made of several bodies, number of possible independent relative motions between the pieces of the mechanism.The working envelope or work area is the volume of working or reaching space.Robot kinematics applies geometry to the study of the movement of multi-degree of freedom kinematic chains that form the structure of robotic systems. The emphasis on geometry means that the links of the robot are modelled as rigid bodies and its joints are assumed to provide pure rotation or translation.Robot kinematics studies the relationship between the dimensions and connectivity of kinematic chains and the position, velocity and acceleration of each of the links in the robotic system, in order to plan and control movement and to compute actuator forces and torques. The relationship between mass and inertia properties, motion, and the associated forces and torques is studied as part of robot dynamics.Forward kinematics specifies the joint parameters and computes the configuration of the chain. For serial manipulators this is achieved by direct substitution of the joint parameters into the forward kinematics equations for the serial chain. For parallel manipulators substitution of the joint parameters into the kinematics equations requires solution of a set of polynomial constraints to determine the set of possible end-effector locations. In case of a Stewart platform there are 40 configurations associated with a specific set of joint parameters.Methods for a forward kinematic analysis:

- using straightforward geometry
- using transformation matrices

Inverse kinematics specifies the end-effector location and computes the associated joint angles. For serial manipulators this requires solution of a set of polynomials obtained from the kinematics equations and yields multiple configurations for the chain. The case of a general 6R serial manipulator (a serial chain with six revolute joints) yields sixteen different inverse kinematics solutions, which are solutions of a sixteenth degree polynomial. For parallel manipulators, the specification of the end-effector location simplifies the kinematics equations, which yields formulas for the joint parameters.Inverse kinematics is the process of converting a Cartesian point in space into a set of joint angles to more efficiently move the end-effector of a robot to a desired orientation. Assuming the parameters of a provided robot, a general equation for the end-effector point was calculated and used to plot the region of space that it can reach.Homogeneous transformation is used to solve kinematic problems. This transformation specifies the location (position and orientation) of the hand in space with respect to the base of the robot, but it does not tell us which configuration of the arm is required to achieve this location. It is often possible to achieve the same hand position with many arm configurations.Homogeneous transformation is used to calculate the new coordinate values for a robot part. Transformation matrix must be in square form.

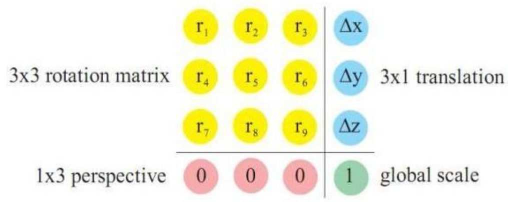


Figure 1 Homogeneous Transformation matrix

3x3 rotation matrix may change with respect to rotation value. 3x1 translation matrix shows the changing value between the coordinate systems. Global scale value is fix and 1. Also 1x3 perspective matrix is fix.In computer vision, image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super-pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image (see edge detection). Each of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic(s). When applied to a stack of images, typical in medical imaging, the resulting contours after image segmentation can be used to create 3D reconstructions with the help of interpolation algorithms like Marching cubes.Digital image processing has dominated over analog image processing with the passage of time due its wider range of applications.The digital image processing deals with developing a digital system that performs operations on an digital image.

II. PROBLEM DESCRIPTION

During Inverse Kinematics, the time taken for predicting movement of link is large.

Other problems in Inverse kinematics:

- There may be multiple solutions,
- For some situations, no solutions,
- Redundancy problem

PROBLEM DEFINITION:

In order to place the end-effector of a redundant manipulator at a desired location (position/orientation) a proper configuration of the arm must be specified, i.e. the suitable values of the joint variables which place the end-effector to the given location must be computed. This is the well known inverse kinematics problem. Mapping from the world coordinates to the joint coordinates for a redundant robot is not one to one, meaning that there may exist an infinite number of joint variable settings which result in a given end-effector position/orientation.

III. OBJECTIVES

To draw the given image with an algorithm for optimizing of inverse kinematic movement of robotic hand.To derive the general equations of inverse kinematics of three link planar robot and to calculate the angles at each joints for particular end-effector points.

IV. FEASIBILITY OF THE PROJECT

During the development of the project, I have been researched the feasibility in the different field, especially software and hardware. The feasibility study is below in details.

A. Software Feasibility

In the software feasibility, I tried to choose the best program that solves my needs. I preferred to use MATLAB. Because it is known that it allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, C#, Java, Fortran and Python. Arduino - Software provides us to load the program onto Arduino UNO.

B . Hardware Feasibility

On the hardware side, we should use Arduino UNO which has serial communication port to send data and usb port to program the chip.

Figure 2 Arduino UNO

In addition to this, we can use servos to rotate robot arm. The servo rotate different angles. Sometimes it is between -90 and 90 degrees. Some of them can rotate about 90 to 180 degrees.

Figure 3 Servo machines

Servo Motor Specifications:

- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 8.5 kgf-cm (4.8 V), 10 kgf-cm (6 V)
- Operating speed: 0.2 s/60^0 (4.8 V), 0.16 s/60^0 (6 V)
- Operating voltage: 4.8 V to 7.2 V
- Dead band width: 5 µs
- Stable and shock proof double ball bearing design
- Temperature range: 0^{0}C – 55^{0}C

Base, Links and End-effector are designed using COMSOL software and fabricated using 3-D printing.

A. ONE ROTATIONAL LINK

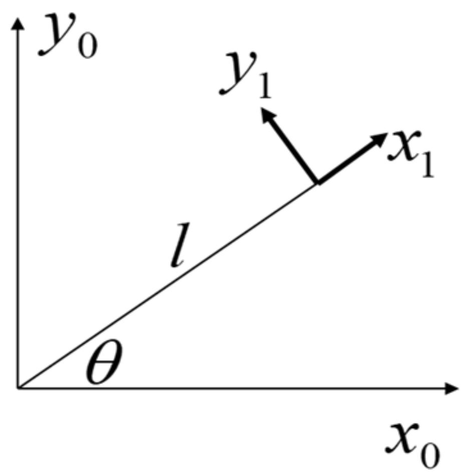


Figure 4 Single link

Forward Kinematics:

$$x_0 = l\cos\theta \qquad ...(1)$$

$$y_0 = l\sin\theta \qquad ...(2)$$

Inverse Kinematics:

Dividing equation(2) by equation(1),

$$y_0 / x_0 = \sin\theta / \cos\theta$$

$$\tan\theta = y_0 / x_0$$

$$\theta = \tan^{-1}(y_0 / x_0) \qquad ...(3)$$

B. TWO ROTATIONAL LINK

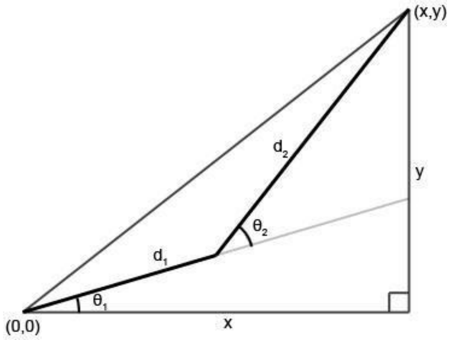


Figure 5 Two link

Forward Kinematics:

$$x_1 = d_1 Cos\theta_1$$

$$y_1 = d_1 Sin\theta_1$$

$$x_2 = d_2 Cos(\theta_1 + \theta_2)$$

$$y_2 = d_2 Sin(\theta_1 + \theta_2)$$

$$x = d_1 Cos\theta_1 + d_2 Cos(\theta_1 + \theta_2) \qquad ...(4)$$

$$y = d_1 Sin\theta_1 + d_2 Sin(\theta_1 + \theta_2) \qquad ...(5)$$

Inverse Kinematics:

Squaring and adding equation (4.4) and equation (4.5),

$$x^2 + y^2 = (d_1 Cos\theta_1 + d_2 Cos(\theta_1 + \theta_2))^2 + (d_1 Sin\theta_1 + d_2 Sin(\theta_1 + \theta_2))^2$$

$$x^2 + y^2 = (d_1{}^2 Cos^2\theta_1 + d_2{}^2 Cos^2(\theta_1 + \theta_2) + 2d_1d_2 Cos\theta_1 Cos(\theta_1 + \theta_2))$$

$$+ (d_1{}^2 Sin^2\theta_1 + d_2{}^2 Sin^2(\theta_1 + \theta_2) + 2d_1d_2 Sin\theta_1 Sin(\theta_1 + \theta_2))$$

$$x^2 + y^2 = d_1{}^2 (Cos^2\theta_1 + Sin^2\theta_1) + d_2{}^2 (Cos^2(\theta_1 + \theta_2) + Sin^2(\theta_1 + \theta_2))$$

$$+ 2d_1d_2 Cos\theta_1 Cos(\theta_1 + \theta_2) + Sin\theta_1 Sin(\theta_1 + \theta_2))$$

$$x^2 + y^2 = d_1{}^2 + d_2{}^2 + 2d_1d_2 (Cos\theta_1 Cos(\theta_1 + \theta_2) + Sin\theta_1 Sin(\theta_1 + \theta_2))$$

Since $Cos(A - B) = CosACosB + SinASinB$,

$$x^2 + y^2 = d_1{}^2 + d_2{}^2 + 2d_1d_2 Cos(\theta_1 - (\theta_1 + \theta_2))$$

$$x^2 + y^2 = d_1{}^2 + d_2{}^2 + 2d_1d_2 Cos(-\theta_2)$$

$$x^2 + y^2 = d_1^2 + d_2^2 + 2d_1 d_2 Cos\theta_2$$

$$(x^2 + y^2 - d_1^2 - d_2^2) / 2d_1 d_2 = Cos\theta_2$$

$$\theta_2 = Cos^{-1}((x^2 + y^2 - d_1^2 - d_2^2) / 2d_1 d_2) \quad ...(6)$$

We know that $Cos^2\theta_1 + Sin^2\theta_1 = 1$,

$$Sin^2\theta_2 = 1 - Cos^2\theta_2$$

$$\theta_2 = \pm Sin^{-1} \sqrt{1 - Cos^2\theta_2} \quad ...(7)$$

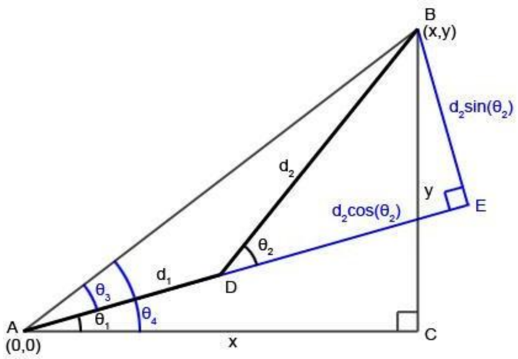


Figure 6 Joining A and B

$$\tan\theta_4 = y / x \quad ...(8)$$

$$\tan\theta_3 = d_2 \sin\theta_2 / (d_1 + d_2 \cos\theta_2) \quad ...(9)$$

$$\theta_4 - \theta_3 = \theta_1 \quad ...(10)$$

Since $\tan(A - B) = (\tan A - \tan B) / (1 + \tan A \tan B)$,

$$\tan(\theta_4 - \theta_3) = (\tan\theta_4 - \tan\theta_3) / (1 + \tan\theta_4 \tan\theta_3) ...(11)$$

Substituting equations (8), (9) and (10) in equation (11)

$$\tan(\theta_1) = \frac{(y/x) - (d_2 \sin\theta_2 / (d_1 + d_2 \cos\theta_2))}{1 + (y/x)(d_2 \sin\theta_2 / (d_1 + d_2 \cos\theta_2))}$$

$$\tan(\theta_1) = \frac{y(d_1 + d_2 \cos\theta_2) - x(d_2 \sin\theta_2) / (x(d_1 + d_2 \cos\theta_2))}{x(d_1 + d_2 \cos\theta_2) + y(d_2 \sin\theta_2) / (x(d_1 + d_2 \cos\theta_2))}$$

$$\tan(\theta_1) = \frac{y(d_1 + d_2 \cos\theta_2) - x(d_2 \sin\theta_2)}{x(d_1 + d_2 \cos\theta_2) + y(d_2 \sin\theta_2)}$$

$$\theta_1 = \tan^{-1}\frac{y(d_1 + d_2\cos\theta_2) - x(d_2\sin\,\theta_2)}{x(d_1 + d_2\cos\theta_2) + y(d_2\,\sin\,\theta_2)} \quad (12)$$

C. THREE ROTATIONAL LINK

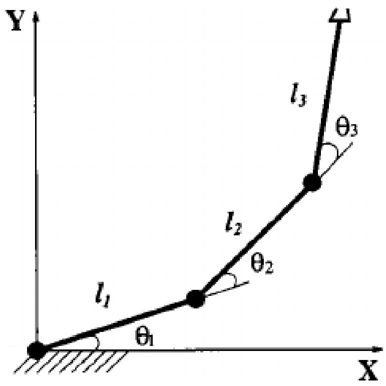


Figure 7 Three link

Forward Kinematics:

$$x_1 = l_1 Cos\theta_1$$

$$y_1 = l_1 Sin\theta_1$$

$$x_2 = l_2 Cos(\theta_1 + \theta_2)$$

$$y_2 = l_2 Sin(\theta_1 + \theta_2)$$

$$x_3 = l_3 Cos(\theta_1 + \theta_2 + \theta_3)$$

$$y_3 = l_3 Sin(\theta_1 + \theta_2 + \theta_3)$$

$$x = l_1 Cos\theta_1 + l_2 Cos(\theta_1 + \theta_2) + l_3 Cos(\theta_1 + \theta_2 + \theta_3) \quad ...(13)$$

$$y = l_1 Sin\theta_1 + l_2 Sin(\theta_1 + \theta_2) + l_3 Sin(\theta_1 + \theta_2 + \theta_3) \quad ...(14)$$

Inverse Kinematics:

Since this robot has three degrees of freedom and in only 2D a third constraint was required. The second and third joint angles are always equal.

$$\theta_2 = \theta_3$$

Let $\phi = \theta_1 + \theta_2 + \theta_3$ which can be obtained by using the formula $\phi = a\tan2(y, x)$.

$$x_2 = x - l_3 Cos(\phi)$$

$$y_2 = y - l_3 Sin(\phi)$$

$$\theta_2 = Cos^{-1}((x_2^2 + y_2^2 - l_1^2 - l_2^2) / 2 l_1 l_2)$$

$$\theta_2 = \pm Sin^{-1} \sqrt{1 - Cos^2\theta_2}$$

$$\theta_1 = \tan^{-1} \frac{y_2 (l_1 + l_2 \cos\theta_2) - x_2 (l_2 \sin\theta_2)}{x_2 (l_1 + l_2 \cos\theta_2) + y_2 (l_2 \sin\theta_2)}$$

All inverse and forward kinematics equations were derived. The inverse kinematics equations are derived using the law of cosines, double angle formulas, trigonometric identities and the quadratic equation. Forward kinematics is the method for determining the orientation and position of the end effector, given the joint angles and link lengths of the robot arm.

I. VI ROBOTIC ARM DESIGN

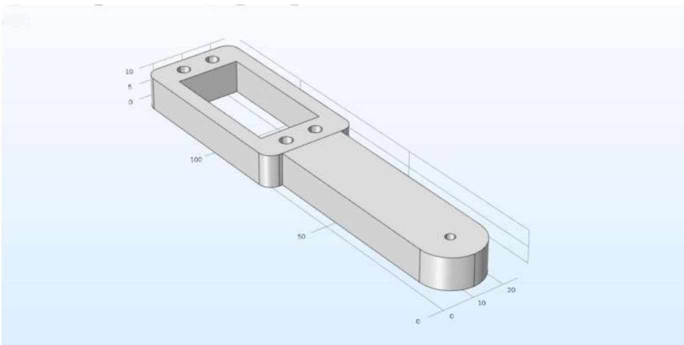


Figure 8 Link design

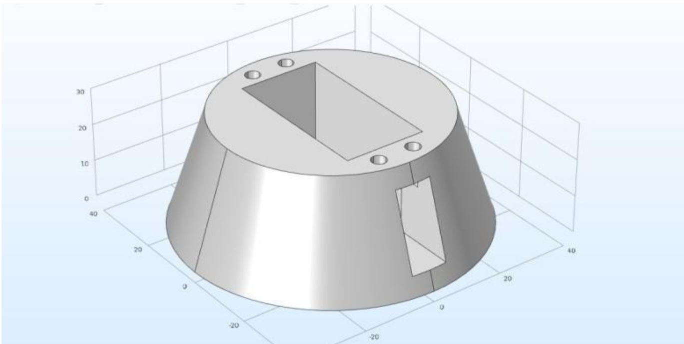


Figure 9 Base design

DESIGN COMPONENTS:

- Pen (End-effector)
- Links (Approx. 10cm each)
- Base
- Servo motors at Joints
- Jumper wires

- Breadboard
- Arduino UNO with cable and its software

Figure 10 Experimental setup

The fabricated base, links, end-effector and the motors are fixed by screws and nuts. Supply and signal are given to the motors from the Arduino UNO board by using jumper wires and breadboard. Developed MATLAB code is connected to the Arduino UNO software and the angles are given to the corresponding motors through Arduino UNO board at each points and the End-effector is actuated and the images are drawn on a paper.

II. VII ALGORITHM

Step 1: Convert HD image to black and white image

Step 2: Convert black and white image to pixel image

Step 3: Convert pixel image to graph

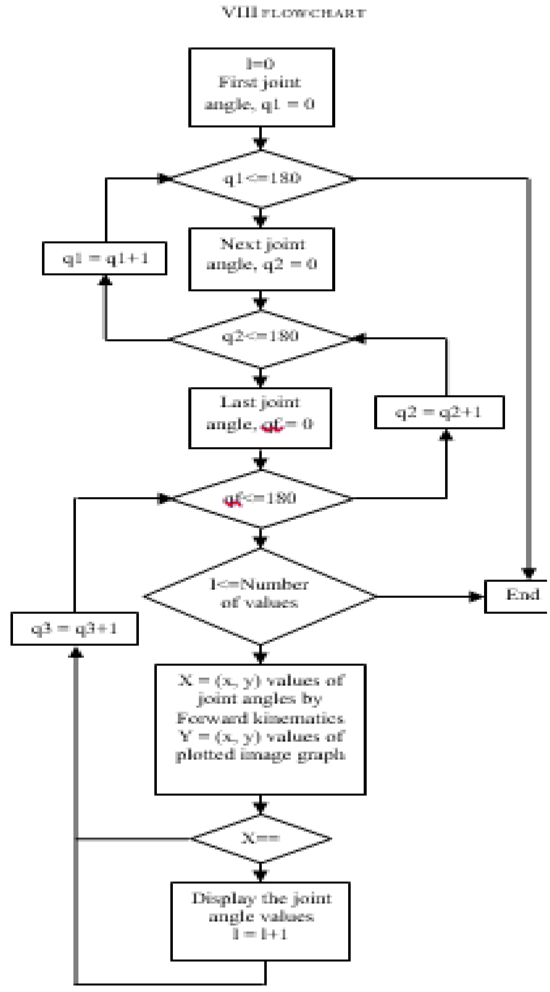
Step 4: Compare x, y values of end-effector by forward kinematics with x, y values plotted in graph

Step 5: Display the joint angles

Step 6: Plot links, joints and end-effector positions and obtain the end-effector position points

Step 7: Compare the end-effector position points with the plotted graph points and calculate the error percentage

III. VIII FLOWCHART



IV. VIII RESULTS

THRESHOLDS:

A threshold is a limit. Thresholds can either be absolute or relative. In the context of gray scale images, an absolute threshold refers to a gray value (e.g. 0-255) and a relative threshold to a gray value difference, i.e. one gray value minus another.A frequent use of thresholds is in binarization of gray scale images, where one absolute threshold divides the histogram into two intervals, below and above the threshold. All pixels below the threshold are made black and all pixels above the threshold are made white.The given original RGB image is converted into black and white image by using binary thresholding to differentiate dark and bright regions. We have to plot points on the dark regions to get the image.

RESOLUTION:

The resolution can be defined in many ways. Such as pixel resolution, spatial resolution, temporal resolution, spectral resolution. Out of which we are going to discuss pixel resolution.In pixel resolution, the term resolution refers to the total number of count of pixels in an digital image. For example, if an image has M rows and N columns, then its resolution can be defined as M X N.If we define resolution as the total number of pixels, then pixel resolution can be defined with set of two numbers. The first number the width of the picture, or the pixels across columns, and the second number is height of the picture, or the pixels across its width.We can say that the higher is the pixel resolution, the higher is the quality of the image. There are many number of pixels available in the image. To reduce the number of pixels to a limited number, the black and white image is converted into 64x64 pixel image which contains a total number of 4096 pixels.This 64x64 pixel image is represented as 64x64 matrix as a_{11}, a_{12}, a_{13},.........which contains 1's and 0's.The positions of 0's represents black region which is converted into x-y graph by plotting n along x-axis and m along y-axis where n and m values are taken from matrix positions a_{mn}.

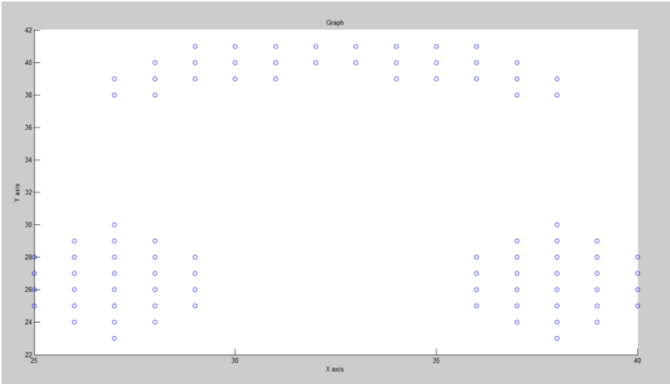


Figure 11 Plotted Graph

By using the equations (4.13) and (4.14) of forward kinematics of three link and comparing it with the plotted points on the graph, the angle at the joints are calculated and the links, joints and end-effector are plotted on a graph and differentiate them by using different colours. Links represented in blue, Joints represented in pink and the position of End-effector in green.

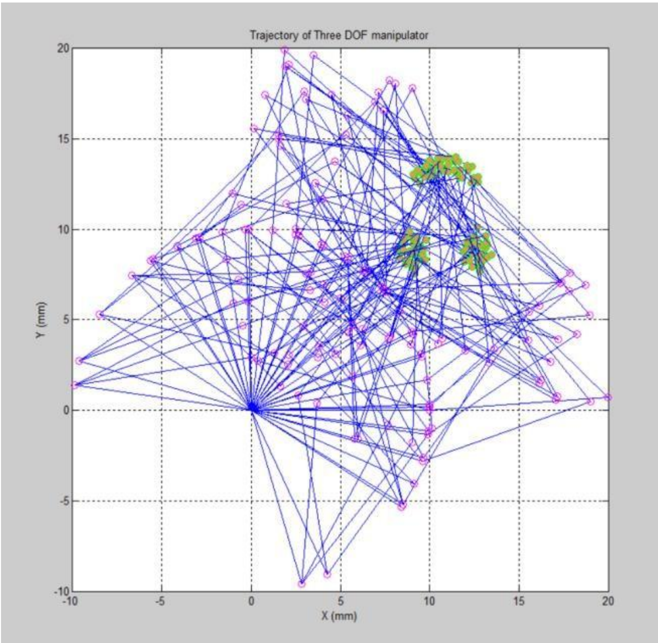


Figure 12 Plotting Links and Joints

To get the image, the plotted end-effector points are again plotted using scatterplot on a different graph which is the resultant output image.

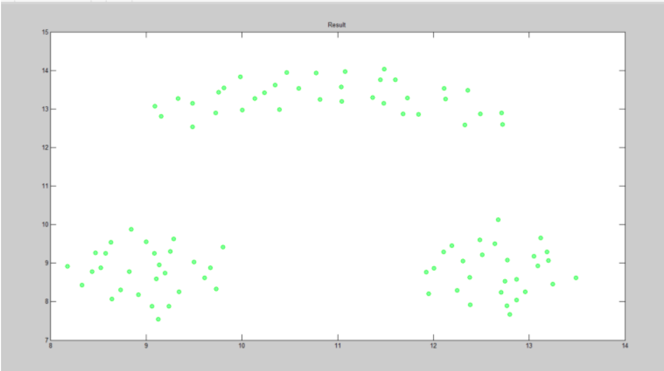


Figure 13 Resultant Plots

V. CONCLUSION

In the robot kinematics, the gripper can be moved where is wanted using rotation of links and joints. For this purpose, links and joints are accepted as a coordinate system individually, as using homogeneous transformations.Robot kinematic is divided in two types: forward kinematic and inverse kinematic. Direct(forward) kinematics involves solving the forward transformation equation to find the location of the hand in terms of the angles and displacements between the links. Inverse kinematics involves solving the inverse transformation equation to find the relationships between the links of the manipulator from the location of the hand in space.

REFERENCES

[1] Denavit, J. & Hartenberg, R. S. (1955). A kinematic notation for lower-pair mechanisms based on matrices. Journal of Applied Mechanics, Vol., 1 (June 1955) pp. 215-221

[2] Funda, J.; Taylor, R. H. & Paul, R.P. (1990). On homogeneous transorms, quaternions, and computational efficiency. IEEE Trans.Robot. Automat., Vol., 6 (June 1990) pp. 382–388

[3] Kucuk, S. & Bingul, Z. (2004). The Inverse Kinematics Solutions of Industrial Robot Manipulators, IEEE Conferance on Mechatronics, pp. 274-279, Turkey, June 2004, Istanbul

[4] Craig, J. J. (1989). Introduction to Robotics Mechanics and Control, USA:AddisionWesley Publishing Company

[5] Hamilton, W. R. (1869). Elements of quaternions, Vol., I & II, Newyork Chelsea

[6] Salamin, E. (1979). Application of quaternions to computation with rotations. Tech., AI Lab, Stanford Univ., 1979

Kotelnikov, A. P. (1895). Screw calculus and some of its applications to geometry and mechanics. Annals of the Imperial University of Kazan.

[8] Pervin, E. & Webb, J. A. (1983). Quaternions for computer vision and robotics, In conference on computer vision and pattern recognition. pp 382-383, Washington, D.C.

[9] Gu, Y.L. & Luh, J. (1987). Dual-number transformation and its application to robotics. IEEE J. Robot. Automat. Vol., 3, pp. 615-623

[10] Kim, J. H. & Kumar, V. R. (1990). Kinematics of robot manipulators via line transformations. J. Robot. Syst., Vol., 7, No., 4, pp. 649–674

[11] Caccavale, F. & Siciliano, B. (2001). Quaternion-based kinematic control of redundant spacecraft/ manipulator systems, In proceedings of the 2001 IEEE international conference on robotics and automation, pp. 435-440

[12] Rueda, M. A. P.; Lara, A. L. ; Marinero, J. C. F.; Urrecho, J. D. & Sanchez, J.L.G. (2002). Manipulator kinematic error model in a calibration process through quaternion-vector pairs, In proceedings of the 2002 IEEE international conference on robotics and automation, pp. 135-140