4R PLANAR MANIPULATOR

A MAJOR PROJECT REPORT

Submitted by

CH.EN.U4AIE20001	ANBAZHAGAN E
CH.EN.U4AIE20053	RAMYA POLAKI
CH.EN.U4AIE20069	TRINAYA KODAVATI
CH.EN.U4AIE20031	SMITHIN REDDY K
CH.EN.U4AIE20035	ABHIRAM KUNCHAPU

Under the Guidance of

Dr. Manoj Prabhakaran

In partial fulfilment for the award of the degree

Of

BACHELORS OF TECHNOLOGY

In

COMPUTER SCIENCE ENGINEERING

AMRITA SCHOOL OF ENGINEERING, CHENNAI

AMRITA VISHWA VIDYAPEETHAM

CHENNAI – 601103, TAMIL NADU

DECEMBER -2021

ACKNOWLEDGMENT:

We offer our sincere pranams at the lotus feet of Universal guru, MATA AMRITANANDAMAYI DEVI who blessed us with her grace to make this a successful major project.

We express our deep sense of gratitude to Dr. P Shankar, Principal, Amrita School of Engineering, Chennai for his kind support. We would like to extend my gratitude and heartfelt thanks to Dr. A Manikandan, Chairperson, Department of Electrical and Electronics Engineering, for his constant help, suggestions, and inspiring guidance. I am grateful to my guide Dr. Manoj Prabhakaran, for his invaluable support and guidance during the major project work. I am also indebted to Dr. Prasanna Kumar, Chairperson of the Department of AI, ASE, Chennai for his guidance.

We are also thankful to all other staff members of the Department of Computer Science Engineering for their valuable help, all my classmates who have always been a source of strength, for always being there and extending their valuable helps to the successful completion of this work

Table of contents: **Pg No:**

Abstract: ---3

Chapter 1:

Introduction---4

Problem Statement --4

Chapter 2:

Implementation---5

Chapter 3:

Discussion &Results--12

Conclusion---15

Future Scope-- 15

References--15

ABSTRACT:

For a 4R planar manipulator, the Kinematic Model of Manipulator is developed using the *Denavit-Hartenberg* (D-H) approach. Analysis of forward and inverse kinematics of manipulator is done using the method that uses algebraic and geometric methods to solve forward and inverse kinematics. The 4R robot arm has four joints to imitate a human upper arm namely joints 1, 2, 3, and 4 that rotate around x, y, and z axes respectively. MATLAB code is used to simulate forward kinematics whereas Robotics Toolbox is used to simulate animation of forward kinematics , and inverse kinematics is applied using geometric approach in order to understand the actual formulation, providing the theoretical foundation for manipulator analysis and study. Finally, simulation results for the kinematics model using the MATLAB program based on the DH convention are presented in our project.

CHAPTER 1

INTRODUCTION:

Robotics is becoming an increasingly important aspect of the global economy. Robotic design and application innovations demonstrate how robot technology has the potential to improve people's lives. Robotics have been widely used in the manufacturing business to boost productivity and reduce labour expenses. These robots are constructed with various structures to perform specialized jobs that are controlled by computer programming to generate various robot movements or trajectories when manipulating products on the assembly line.

Nowadays, robotics is a rich area of research, in terms of its kinematics, dynamics, and control. Kinematics, in particular, plays a significant role in robotics and especially in the study of industrial manipulators' behaviour. Therefore, a decisive step in any robotics system is the analysis and modelling of the manipulator kinematics. The purpose of manipulator kinematics is to establish a relationship between the position and attitude of the manipulator space and the movement of each joint, which gives a way for analysing manipulator motion control. Forward kinematics and inverse kinematics are the two main areas of study. Forward kinematics is the process of determining the position and attitude of the end effector relative to the coordinate system using known structural parameters and motion parameters of each manipulator joint; inverse kinematics is the process of determining the motion parameters of each manipulator joint using known structural parameters and the position and attitude of the end effector, which is a nodus of manipulator kinematics.

PROBLEM STATEMENT

To find forward and inverse kinematics of the 4R planar manipulator *(fig1).*

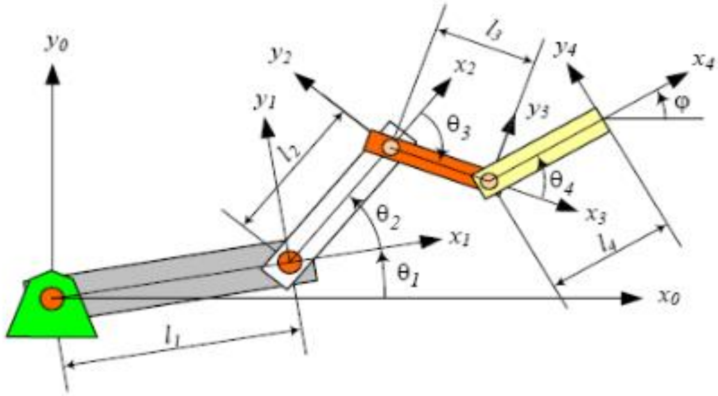


Fig1 4R planar manipulator

CHAPTER 2

IMPLEMENTATION:

DH convention:

In the kinematics study of the robotic manipulator, the Denavit-Hartenberg (DH) convention is often used. It works by attaching a coordinate frame to each joint and specifying four DH parameters for each connection, then constructing a DH table using these parameters. Finally, a transformation matrix is created between distinct coordinate frames. The main goal is to keep track of the EE or gripper's position and orientation in its work area. Using the DH technique, we will first determine the link between joint variables and gripper position and orientation. DH parameters for *fig-1* are shown below.

link	a_i	$\propto_i$	d_i	θ_i
1	l_1	0	0	θ_1
2	l_2	0	0	θ_2
3	l_3	0	0	$-\theta_3$
4	l_4	0	0	θ_4

DH parameters table

By applying the Denavit-Hartenberg (DH) notations for the joint coordinates, the DH-table can be constructed as listed above in Table.

Where,

a_i = distance from z_{i-1} to z_i measured along x_i

d_i = distance from x_{i-1} to intersection of x_i measured along z_{i-1}

θ_i = angle between x_{i-1} and x_i measured along z_{i-1}

α_i = angle between x_{i-1} and z_i measured along x_i

Transformation matrix :

The matrices associated with these operations are:

$$Transz_i(d_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad Rotz_i(\theta_i) = \begin{bmatrix} cos\theta_i & -sin\theta_i & 0 & 0 \\ sin\theta_i & cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Similarly,

$$Transx_i(a_i) = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad Rotx_i(\alpha_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\alpha_i & -sin\alpha_i & 0 \\ 0 & sin\alpha_i & cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The use of the Denavit-Hartenberg convention yields the link transformation matrix, [i-1Ti] as

$$T_{i-1}^{i} = \begin{pmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & -\sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

known as the Denavit-Hartenberg matrix or Transformation matrix.

Forward Kinematics:

Forward kinematics is the application of a robot's kinematic equations to calculate the position of the end-effector given a set of joint parameters.

Once the DH table is ready, the transformation matrices are easy to find. Generally, the matrix of transformation from the frame B_i to the frame B_{i-1} for the standard DH method is given by

Transformation matrix

$$T_{i-1}^{i} = \begin{pmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & -\sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Individual transformation matrices for i =1, 2, 3, and 4

Substituting DH values of link 1 from DH parameters table

i=1

$$T_{i-1}^{i} = \begin{pmatrix} \cos\theta_1 & -\cos 0\sin\theta_1 & -\sin 0\sin\theta_1 & a_1\cos\theta_1 \\ \sin\theta_1 & \cos 0\cos\theta_1 & -\sin 0\cos\theta_1 & a_1\sin\theta_1 \\ 0 & \sin 0 & \cos 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_0^1 = \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & a_1\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & a_1\sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Similarly for link 2

i=2

$$T_1^2 = \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & a_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & a_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Similarly for link 3

i=3

$$T_2^3 = \begin{pmatrix} \cos(-\theta_3) & -\sin(-\theta_3) & 0 & a_3\cos(-\theta_3) \\ \sin(-\theta_3) & \cos(-\theta_3) & 0 & a_3\sin(-\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_2^3 = \begin{pmatrix} \cos\theta_3 & \sin\theta_3 & 0 & a_3\cos\theta_3 \\ -\sin\theta_3 & \cos\theta_3 & 0 & -a_3\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Similarly for link 4

i=4

$$T_3^4 = \begin{pmatrix} \cos\theta_4 & -\sin\theta_4 & 0 & a_4\cos\theta_4 \\ \sin\theta_4 & \cos\theta_4 & 0 & a_4\sin\theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$\mathrm{T}_0^1, \mathrm{T}_1^2, \mathrm{T}_2^3, \mathrm{T}_3^4$ are individual transform matrices

$$\cos(\theta_1+\theta_2) = C_1C_2 - S_1S_2;\ \cos(\theta_1-\theta_2) = C_1C_2 + S_1S_2$$

$$\sin(\theta_1+\theta_2) = S_1C_2 + C_1S_2;\ \sin(\theta_1-\theta_2) = S_1C_2 + C_1S_2$$

$$C_{ij} = \cos(\theta_i + \theta_j)$$

$$C_i - \cos\theta_i \quad S_i - \sin\theta_i$$

$$T_0^4 = T_0^1.T_1^2.T_2^3.T_3^4$$

$$T_0^4 = \begin{pmatrix} C_1 & -S_1 & 0 & a_1C_1 \\ S_1 & S_1 & 0 & a_1S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} C_2 & -S_2 & 0 & a_2C_2 \\ S_2 & C_1 & 0 & a_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} * T_2^3 * T_3^4$$

$$T_0^4 = \begin{pmatrix} C_1C_2 - S_1S_2 & -C_1S_2 - S_1C_2 & 0 & a_1C_1C_2 - a_2S_1S_2 + a_1C_1 \\ S_1C_2 + C_1S_2 & -S_1S_2 + C_1C_2 & 0 & a_2S_1C_2 + a_2C_1S_2 + a_1S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} * T_2^3 * T_3^4$$

$$T_0^4 = \begin{pmatrix} C_{12} & -S_{12} & 0 & a_1C_{12} + a_1C_1 \\ S_{12} & C_{12} & 0 & a_1S_{12} + a_1S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} C_3 & S_3 & 0 & a_3C_3 \\ -S_3 & C_3 & 0 & -a_3S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} * T_3^4$$

$$T_0^4 = \begin{pmatrix} C_{12}C_3 + S_{12}S_3 & C_{12}S_3 - S_{12}C_3 & 0 & a_3C_{12}C_3 + a_3S_{12}S_3 + a_2C_{12} + a_1C_1 \\ S_{12}C_3 - C_{12}S_3 & S_{12}S_3 + C_{12}C_3 & 0 & a_3S_{12}C_3 - a_3C_{12}S_3 + a_2S_{12} + a_1S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} * T_3^4$$

$$T_0^4 = \begin{pmatrix} \cos(\theta_1+\theta_2-\theta_3) & -\sin(\theta_1+\theta_2-\theta_3) & 0 & a_3\cos(\theta_1+\theta_2-\theta_3) + a_2C_{12} + a_1C_1 \\ \sin(\theta_1+\theta_2-\theta_3) & \cos(\theta_1+\theta_2-\theta_3) & 0 & a_3\sin(\theta_1+\theta_2-\theta_3) + a_2S_{12} + a_1S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} * T_3^4$$

$$T_0^4 = \begin{pmatrix} C_{12-3} & -S_{12-3} & 0 & a_3C_{12-3} + a_2C_{12} + a_1C_1 \\ S_{12-3} & S_{12-3} & 0 & a_3S_{12-3} + a_2S_{12} + a_1S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} C_4 & -S_4 & 0 & a_4C_4 \\ S_4 & C_4 & 0 & a_4S_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Here\ C_{12-3} = \cos(\theta_1+\theta_2-\theta_3), S_{12-3} = \sin(\theta_1+\theta_2-\theta_3)$$

$T_0^4 =$

$$\begin{pmatrix} C_{12-3}C_4 - S_{12-3}S_4 & -S_{12-3}C_4 - C_{12-3}S_4 & 0 & a_4(C_{12-3}C_4 - S_{12-3}S_4) + a_3C_{12-3} + a_2C_{12} + a_1C_1 \\ S_{12-3}C_4 - C_{12-3}S_4 & S_{12-3}S_4 - C_{12-3}C_4 & 0 & a_4(C_{12-3}C_4 - S_{12-3}S_4) + a_3S_{12-3} + a_2S_{12} + a_1S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Final column (4th column of T_0^4) matrix gives general coordinates of end effector

$$[x \quad y \quad z] = [a_4(C_{12-34}) + a_3C_{12-3} + a_2C_{12} + a_1C_1 \quad a_4(S_{12-34}) + a_3S_{12-3} + a_2S_{12} + a_1S_1 \quad 0]$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_4cos(\theta_1 + \theta_2 - \theta_3 + \theta_4) + a_3\cos(\theta_1 + \theta_2 - \theta_3) + a_2\cos(\theta_1 + \theta_2) + a_1cos\theta_1 \\ a_4sin(\theta_1 + \theta_2 - \theta_3 + \theta_4) + a_3\sin(\theta_1 + \theta_2 - \theta_3) + a_2\sin(\theta_1 + \theta_2) + a_1sin\theta_1 \\ 0 \end{bmatrix}$$

Therefore,

X = $a_4cos(\theta_1 + \theta_2 - \theta_3 + \theta_4) + a_3\cos(\theta_1 + \theta_2 - \theta_3) + a_2\cos(\theta_1 + \theta_2) + a_1cos\theta_1$

Y = $a_4sin(\theta_1 + \theta_2 - \theta_3 + \theta_4) + a_3\sin(\theta_1 + \theta_2 - \theta_3) + a_2\sin(\theta_1 + \theta_2) + a_1sin\theta_1$

$angle\ \emptyset\ given\ in\ the\ manipulator\ is\ equal\ to\ (\theta_1 + \theta_2 - \theta_3 + \theta_4)$

Inverse Kinematics:

The application of kinematic equations to determine the motion of a robot to attain a desired position is known as inverse kinematics. A robotic arm in a production line, for example, needs accurate mobility from an initial position to a target position between bins and manufacturing machines to execute automated bin selection. The end-effector refers to the grasping end of a robot arm. The robot configuration is a list of joint positions that are both inside the robot model's position restrictions and do not violate any of the robot's constraints.

We end up with a set of four nonlinear equations with four unknowns. Solving these equations algebraically, known as the inverse kinematics, requires that we need to know the joint variables $\theta_1, \theta_2, \theta_3, and\ \theta_4$ for a given EE position $[d_x \quad d_y \quad d_z]$ and orientation ϕ.

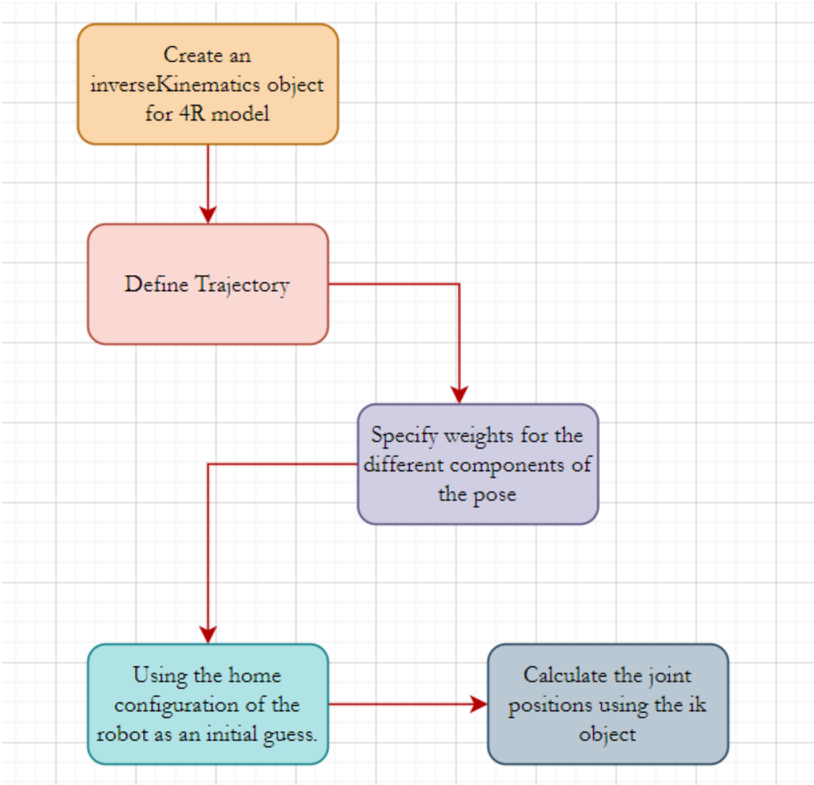


Fig 2 Flow chart

Procedure for solving inverse kinematics

Trajectory:

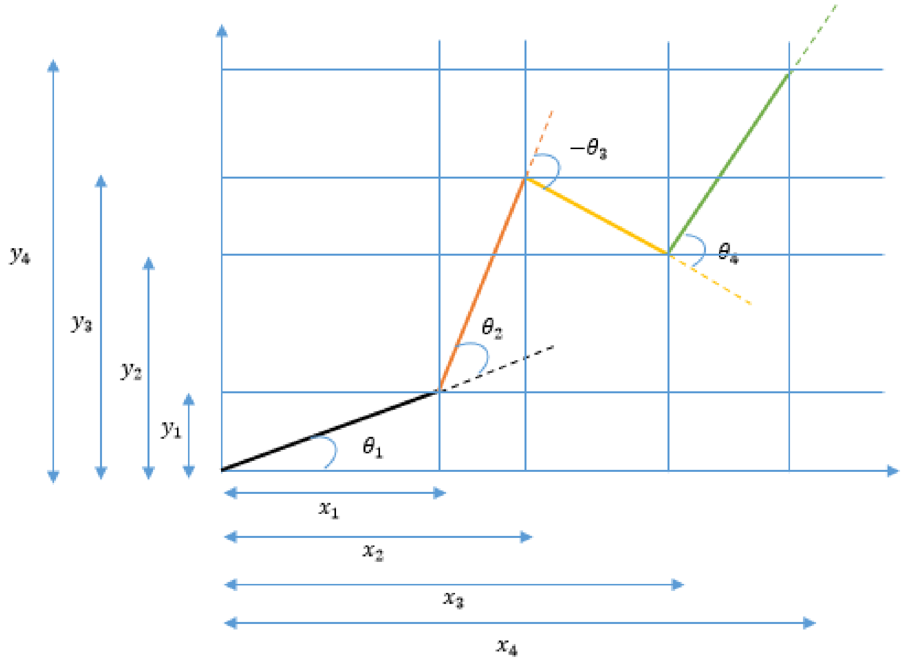
The Trajectories feature displays the path an object travels over time. This is useful for seeing at a glance how the object moves with respect to other objects in the scene during an animation without actually having to play the animation

IK object:

Based on a rigid body tree model, the inverse kinematics System object builds an inverse kinematic (IK) solver to determine joint configurations for a particular end-effector pose. Using the rigid body tree class, created a rigid body tree model for our robot. This model specifies all of the joint constraints enforced by the solver. The joint limits provided in the robot model are obeyed if a solution is possible.

Consider employing generalised inverse kinematics to express additional restrictions outside the end-effector pose, such as aiming constraints, position bounds, or orientation targets. This object allows you to compute IK solutions with multiple constraints.

Finding Orientations – Geometrical Approach :



By Pythagoras theorem,

$x_1 \; = \; l_1 * sin\theta_1$

$y_1 \; = l_1 * sin\theta_1$

$x_2 \; = \; x_1 + \; (cos(\theta_1 \, + \, \theta_2)) * l_2$

$y_2 \; = \; y_1 \, + \; (sin(\theta_1 \, + \, \theta_2)) * l_2$

$x_3 \; = \; x_2 + \; (cos(\theta_1 \, + \, \theta_2 \, + \, \theta_3)) * l_3$

$y_3 = \; y_2 + \; (sin(\theta_1 \, + \theta_2 \, + \, \theta_3)) * l_3$

$x_4 \; = \; x_3 + \; (cos(\theta_1 \, + \, \theta_2 \, + \, \theta_3 \, + \, \theta_4)) * l_4$

$y_4 \; = \; y_3 \, + \; (sin(\theta_1 \, + \theta_2 \, + \, \theta_3 \, + \, \theta_4)) * l_4$

$sin^{-1}\frac{y_1}{l_1} \; = \; \theta_1$

$sin^{-1}\frac{y_2 - y_1}{l_2}$ – $\theta_1 \; = \; \theta_2$

$sin^{-1}\frac{y_3 - y_2}{l_3}$ – θ_2 – $\theta_1 \; = \; \theta_3$

$sin^{-1}\frac{y_4 - y_2}{l_4}$ – θ_1 – θ_2– $\theta_3 \; = \; \theta_4$

CHAPTER 3

DISCUSSION & RESULTS:

In our project , we are focussing on building a Matlab GUI with inputs , which in result will give corresponding outputs. In Fig.3.1 , we can see the main frame which will allow us to select the Kinematics.

Here we implemented Forward and Inverse Kinematics with unique solutions. Former with Link lengths and DH parameters as input , latter with positions of links as in Fig 3.1 and Fig 3.3.

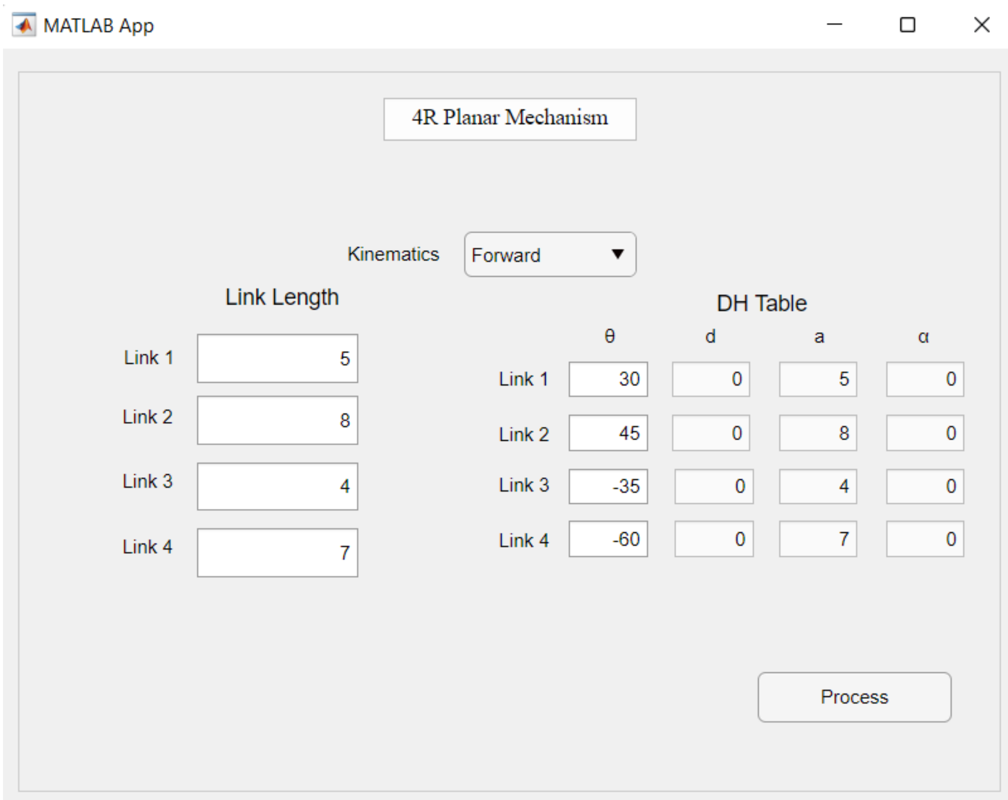


Fig 3.1

And in the Forward Frame , we plotted the Mechanism with the angles and link lengths provided and position of end effector is been calculated and show in

Fig 3.2

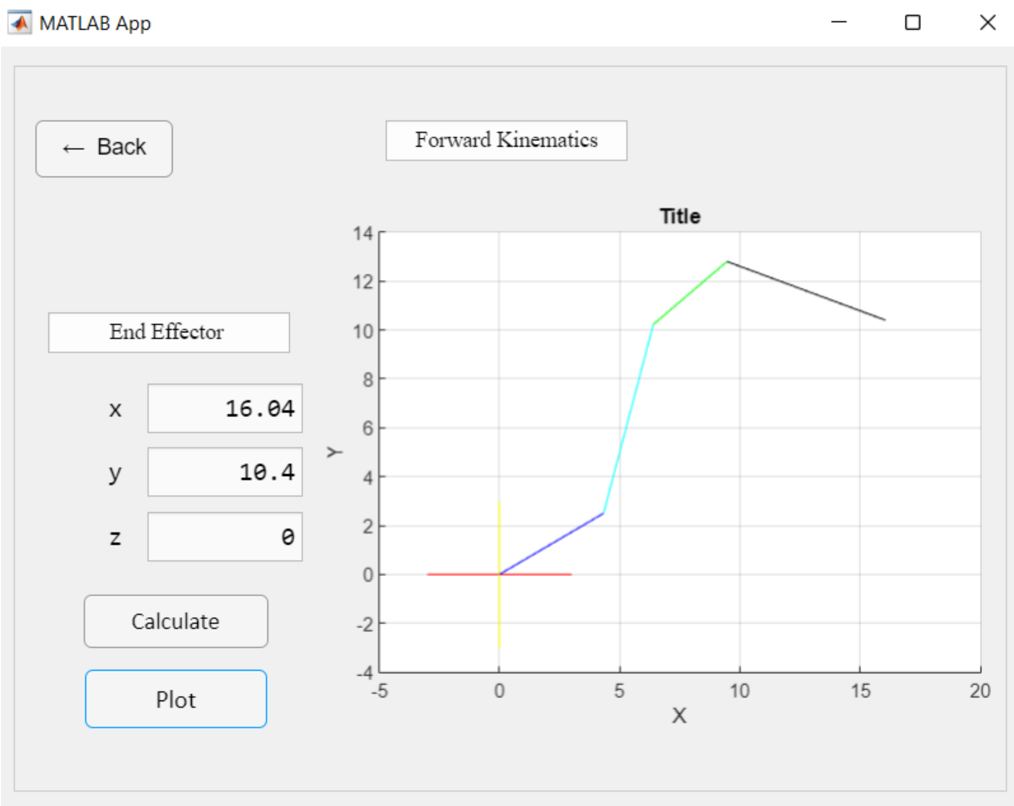

MATLAB App
← Back
Forward Kinematics
End Effector
x 16.04
y 10.4
z 0
Calculate
Plot
Title
Y
X

Fig 3.2

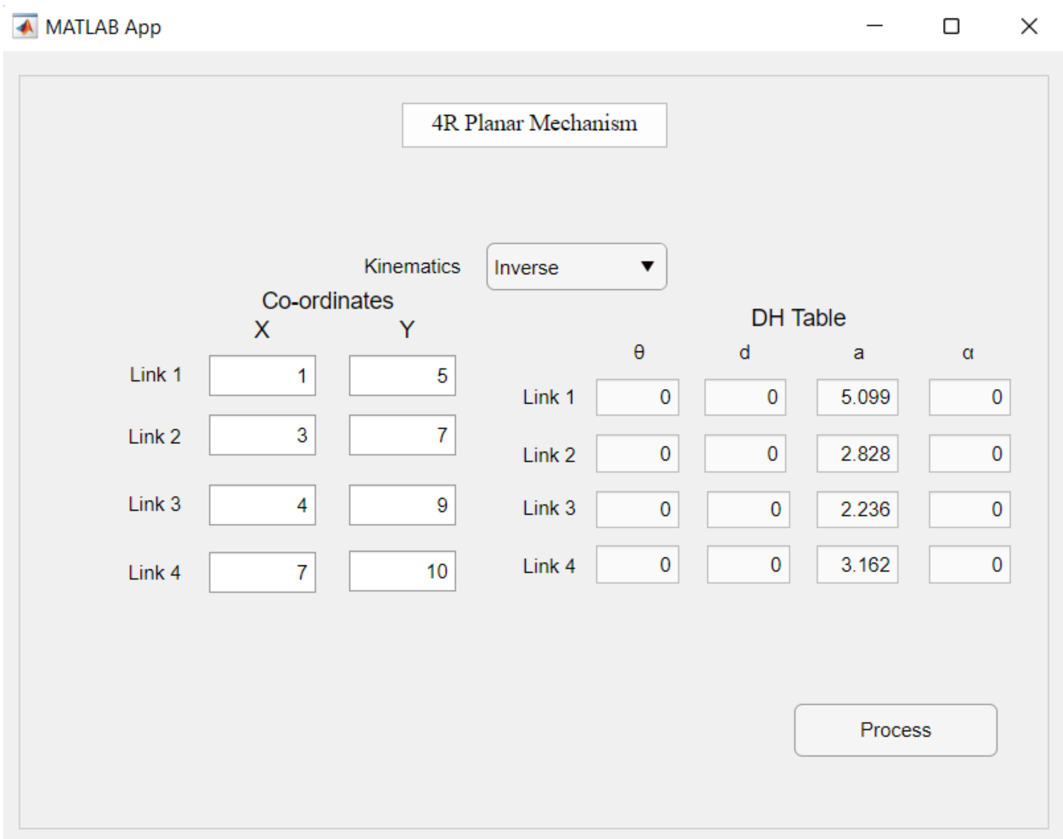

MATLAB App
4R Planar Mechanism
Kinematics
Inverse
Co-ordinates
X Y
Link 1 1 5
Link 2 3 7
Link 3 4 9
Link 4 7 10
DH Table
θ d a α
Link 1 0 0 5.099 0
Link 2 0 0 2.828 0
Link 3 0 0 2.236 0
Link 4 0 0 3.162 0
Process

Fig 3.3

As in Forward frame , in inverse as well, we calculate joint angles and print in the output boxes and plot the mechanism using positions of links as in Fig 3.4

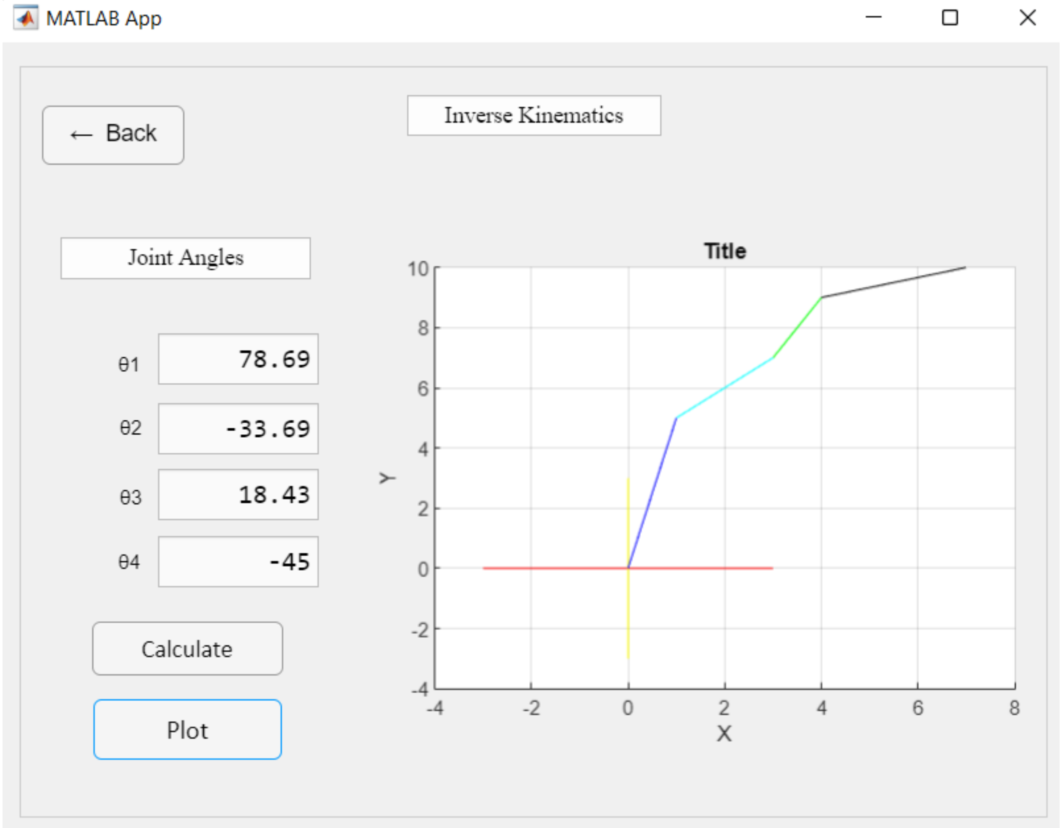
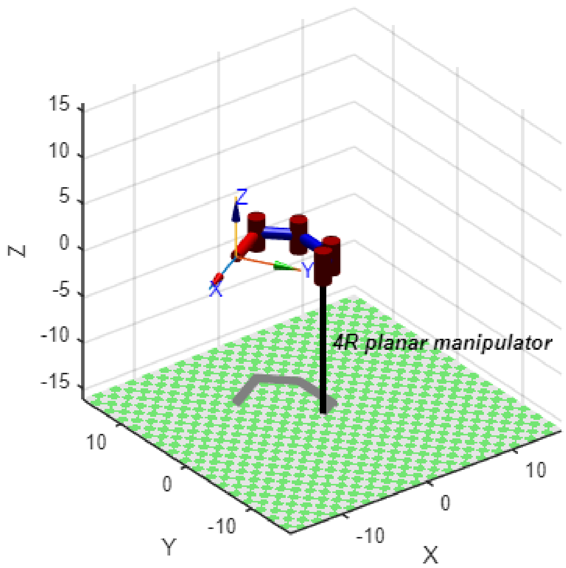


Fig 3.4

ANIMATION :



CONCLUSION:

In this project, we have depicted a part of Formulations and Equations of Kinematics of 4R Planar Mechanism. And also it is important to understand the concepts of DH parameter and Kinematics in order to work along the real time projects based on robotics and drones. So we implemented as Matlab GUI for demonstrating the kinematics of 4R Planar Mechanism.

FUTURESCOPE:

We would like to improve our project in Inverse kinematics part as input with only end effector position and plotting the trajectory in a range of theta with multiple solutions of the mechanism.

REFERENCES:

1. https://www.researchgate.net/publication/279201859_Kinematics_Modeling_of_a_4-DOF_Robotic_Arm
2. https://www.atlantis-press.com/article/25877710.pdf
3. https://link.springer.com/chapter/10.1007/978-3-319-62533-1_1
4. https://www.mathworks.com/help/robotics/ref/inversekinematics-system-object.html
5. https://in.mathworks.com/help/fuzzy/modeling-inverse-kinematics-in-a-robotic-arm.html
6. An Introduction to Robot Kinematics (cmu.edu)
7. (book.pdf) (duke.edu)
8. Microsoft PowerPoint - Lect5_Forward-InverseKinematicsII_Short.ppt [Compatibility Mode] (usc.edu)