JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

FORWARD AND INVERSE KINEMATIC ANALYSIS OF 4-DOF TRRR ROBOTIC ARM USING MATLAB

[1]Arumalla Johnson, [2]M. Venkatesh,

[1]Senior Assistant Professor, [2] Assistant Professor

[1&2]Department of Mechanical Engineering

[1]Kamala Institute of Technology and Science, Singapur, Huzurabad, Telangana, India

Abstract: This paper aims to perform the kinematic analysis of Four degrees of freedom robotic arm. The kinematic analysis is the relationship between the positions, velocities, and accelerations of the links of a manipulator. The kinematics separate into two types, direct 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 joint arrangement of every robot can be described by the Denavit-Hartenberg parameters. These parameters are enough to obtain a working of the robot described and presented in a MATLAB program. The main aim of this work is to develop forward and inverse kinematic models of three degrees of freedom articulated arm using MATLAB GUI to optimize the manipulative task execution. Forward kinematics analysis is done for the flexible twist angle, link lengths, and link offsets of each joint by varying joint angles to specify the position and orientation of the end effectors. Forward analysis can be used to provide the position of some point on the end effectors together with the orientation of the end effectors measured relative to a coordinate system fixed to the ground for a specified set of joint variables. This simulation allows the user to get forward kinematics and inverse kinematics of articulated robots in various link length parameters and joint angles and corresponding end effector's position and orientation.

***Index Terms* – Kinematic Analysis, Forward Kinematics, Inverse Kinematics, MATLAB.**

I. *INTRODUCTION*

A robot is a machine that collects information about the environment using some sensors and makes a decision automatically. People prefer it to use different fields, such as industry, some dangerous jobs including radioactive effects. In this point, robots are regarded as a server. They can be managed easily and provide many advantages. Robot kinematics is the study of the motion(kinematics) of robots. In a kinematic analysis, the position, velocity, and acceleration of all the links are calculated without considering the forces that cause this motion. The relationship between motion and the associated forces and torques is studied in robot dynamics [1]. Robot kinematics deals with aspects of redundancy, collision avoidance, and singularity avoidance. While dealing with the kinematics used in the robots, we deal with each part of the robot by assigning a frame of reference to it, and hence a robot with many parts may have many individual frames assigned to each movable part. For simplicity, we deal with the single manipulator arm of the robot. Each frame is named systematically with numbers, for example, the immovable base part of the manipulator is numbered 0, and the first link joined to the base is numbered 1, and the next link 2 and similarly till n for the last nth link [1]. In the kinematic analysis of manipulator position, there are two separate problems to solve: direct kinematics, and inverse kinematics. Direct 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. In the next chapters, inverse and forward kinematic will be represented in detail [2]. A robot arm is a known manipulator. It is composed of a set of joints separated in space by the arm links. The joints are where the motion in the arm occurs. In basic, a robot arm consists of the parts: base, joints, links, and a gripper. The base is the basic part over the arm, it may be fixed or active. The joint is flexible and joins two separated links. The link is fixed and supports the gripper. The last part is a gripper. The gripper is used to hold [2].

In the paper, the 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.

The Denavit-Hartenberg illustration of forward kinematic equations of robots has grown to be the standard procedure for model robots and their motions. The technique summarizes the relationship between two joints in a concise set of four parameters. Any robot can be modeled using the D-H representation. A computer code has been formed in MATLAB to implement the modeling of any robot with only the DH parameters as input. The purpose of the simulator is to create an accurate Forward kinematic and inverse kinematics representation of any type of robot and its motions. The simulator also allows for the independent manipulation of each joint of the modeled robot. Presented in this paper are the details of this TRRR Robot as well as background on the D-H representation and some analysis on how effectively the TRRR model is in MATLAB.

II.KINEMATICS OF TRRR ROBOT

2.1 Forward Kinematics

Four frames were assigned to the robotic arm as shown in Figure 2.1.1. The frame assignment procedure followed can be summarized in three steps

Assign the z_i axis pointing along the i-th joint axis (either one of two choices). It is better to assign z_0 z_{n-1} and parallel to z_1 z_n and, respectively.

Assign the x_i axis pointing along the common perpendicular from z_i to z_{i-1}.

Assign the y_i axis according to the right-hand rule.

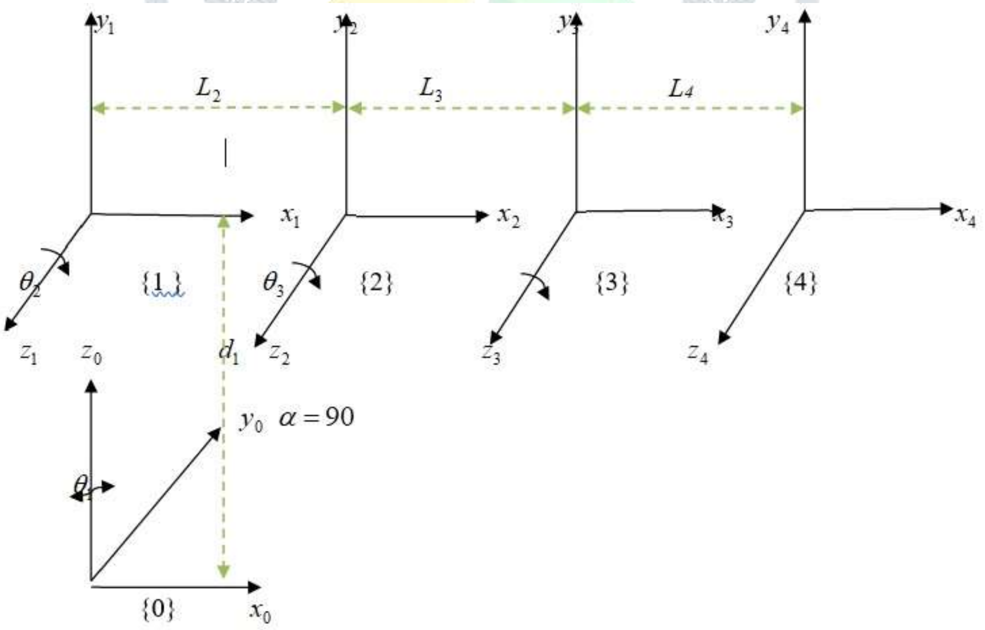


Fig 2.1.1 Assignment of frames for TRRR robot configuration.

The non-rotating reference Frame {0} was attached to a fixed point at the center of the rotary base. Frame {1} corresponding to Link (1) was attached at its upper end. Note that the origin of this frame could have been appointed to any location along with Link (1) since Z0 and Z1 are parallel to each other. Frame {2} rotates with Link (2) and is located at the intersection between Link (1) and Link (2), thus having X2 along the common perpendicular pointing from Z2 to Z3. Frame {3} rotates with Link (3) and is attached at its intersection with Link (2). Finally, Frame {5} corresponding to the end-effector is located midway between the two grippers. Based on the defined frames, the Denavit-Hartenberg (D-H) parameters were determined as shown in Table 2.1.

Table 2.1.D-H parameters of a 4-DOF TRRR Robotic Arm

Link (i)	θ_i	d_i	α_i	a_i	q_i	$C\theta_i$	$S\theta_i$	$C\alpha_i$	$S\alpha_i$
1	θ_1	d_1	90	0	θ_1	$C\theta_1$	$S\theta_1$	0	1
2	θ_2	0	0	L_2	θ_2	$C\theta_2$	$S\theta_2$	1	0
3	θ_3	0	0	L_3	θ_3	$C\theta_3$	$S\theta_3$	1	0
4	θ_4	0	0	L_4	θ_4	$C\theta_4$	$S\theta_4$	1	0

The definition of each of the D-H parameters is as follows:

α_i : Link twist is the angle about x_i between (z_i , z_{i+1})

a_i : Link length is the distance along x_i between (z_i , z_{i+1})

d_i : Link offset is the distance along z_i between (x_{i-1} , x_i)

θ_i : Joint angle is the angle about z_i between (x_{i-1} , x_i)

The D-H parameters were then used to compute the homogenous transform matrices (T_1^0 , T_2^1 a T_3^2 and T_4^3) by substituting in the general formula T_i^{i-1}.

$$T_i^{i-1} = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & C\theta_i a_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & S\theta_i a_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad T_1^0 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^1 = \begin{bmatrix} C_2 & -S_2 & 0 & C_2 L_2 \\ S_2 & C_2 & 0 & S_2 L_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_3^2 = \begin{bmatrix} C_3 & -S_3 & 0 & C_3 L_3 \\ S_3 & C_3 & 0 & S_3 L_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_4^3 = \begin{bmatrix} C_4 & -S_4 & 0 & C_4 L_4 \\ S_4 & C_4 & 0 & S_4 L_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where C_i is the cosine of the angle θ_i , and S_i represents the sine.

By multiplying the homogeneous transformation matrices in the right order, T_4^0 describing the end-effector frame with the reference frame may be obtained.

$$T_4^0 = T_1^0 T_2^1 T_3^2 T_4^3$$

$$T_4^0 = \begin{bmatrix} R(\theta) & P(\theta) \\ 0 & 1 \end{bmatrix}$$

$$R(\theta)=\begin{bmatrix} C_1C_{234} & -C_1S_{234} & S_1 \\ S_1C_{234} & -S_1S_{234} & -C_1 \\ S_{234} & C_{234} & 0 \end{bmatrix}$$ eq (1)

$$P(\theta)=\begin{bmatrix} dx \\ dy \\ dz \end{bmatrix}=\begin{bmatrix} C_1(L_3C_{23}+L_2C_2+L_4C_{234}) \\ S_1(L_3C_{23}+L_2C_2+L_4C_{234}) \\ d_1+L_3S_{23}+L_2S_2+L_4S_{234} \end{bmatrix}$$ Eq (2)

Notice that C_{234} is the cosine of the sum of $\theta_2\ \theta_3\ \theta_4$ C_{23} and is the cosine of the sum of θ_2 and θ_3 . Similarly sine angles.

Here eq(1) represents the orientation of the end effector and eq(2) represents the position of an end effector.

The end effector orientation is:

$$\phi=\theta_2+\theta_3+\theta_4$$ Eq (3)

The DH parameters are akin to the configuration of the robot. For different manipulator structures, the kinematics equations are not unique. Moreover, kinematics equations of the manipulator based on the DH convention provide some singularity making the equations difficult to solve or unsolvable in some cases. In addition, in the DH convention, the common normal is not defined properly when axes of two joints are parallel. In this case, the DH method has a singularity, where a little change in the spatial coordinates of the parallel joint axes can create a huge misconfiguration in the representation of the DH coordinates of their relative position.

2.2 Inverse Kinematics

Here we got a set of nonlinear equations with four unknowns. Solving these equations algebraically, known as inverse kinematics, requires that we need to know the joint variables θ_1, θ_2, θ_3 θ_4 and for a given end-effector position and orientation. We get from equation (2), by dividing, squaring, adding, and using some trigonometric formulas.

$$\theta_1=\tan^{-1}\left(\frac{dy}{dx}\right)$$ Eq (4)

$$\theta_2=\tan^{-1}(c\pm\sqrt{r^2-c^2})-\tan^{-1}(a,b)$$ Eq (5)

$$\theta_3=\cos^{-1}\left(\frac{A^2+B^2+C^2-l_2^2-l_3^2}{2l_2\ l_3}\right)$$ Eq (6)

Where

$a=l_3\sin\theta_3,$ $b=l_2+l_3\cos\theta_3,$ $c=dz-d_1-l_4\sin\phi,$ and $r=\sqrt{a^2+b^2}$

$A=(dx-l_4C_1C\phi),$ $B=(dy-l_4S_1C\phi),$ and $C=(dz-d_1-l_4S\phi),$

Having determined θ_1, θ_2 and θ_3 , we can then find θ_4 from the end effector orientation ϕ as follows:

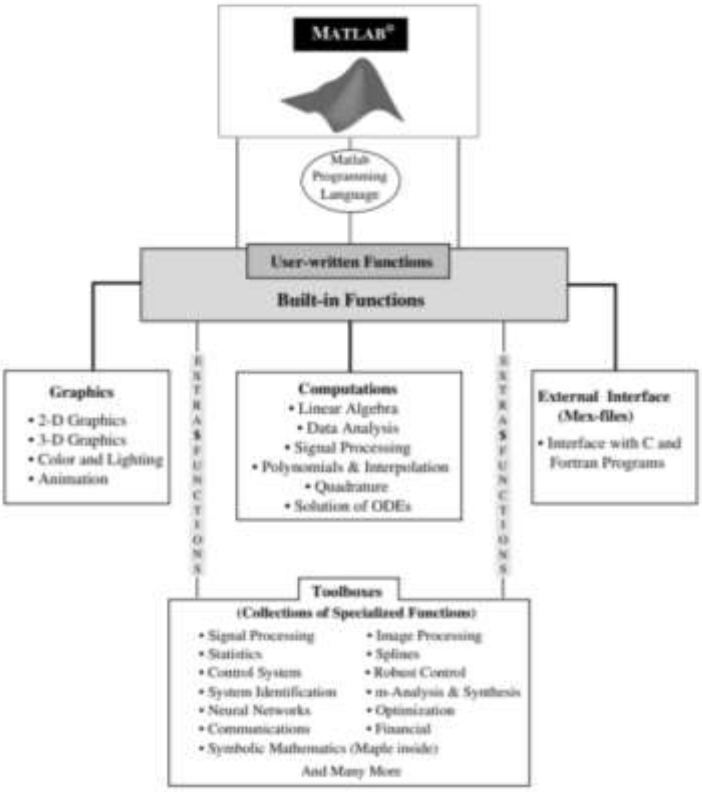
$$\theta_4=\phi-\theta_2-\theta_3$$

III.SIMULATION OF THE ROBOTIC ARM

Since robotics is a multidisciplinary subject, its study requires skills from different fields of knowledge, not easily available. Thus, simulation has been recognized as a suitable tool that combines all these features enabling the user of direct visualization of different kinds of motion that a robot may perform, making the role of simulation very important in robotics. Using the robotics toolbox together with the MATLAB software, the kinematics of a robotic arm can be simulated and analyzed based on the DH convention described before. The toolbox takes a conventional approach to represent the kinematics and dynamics of serial-link robotic arms. Besides, it provides several functions and routines, which are handy for the simulation and scrutiny of robotic manipulators, like kinematics, dynamics, and trajectory creation.

MATLAB is a software package for high-performance numerical computation and visualization. It provides an interactive environment with hundreds of built-in functions for technical computation, graphics, and animation. Best of all, it also provides easy extensibility with its high-level programming language. The name MATLAB stands for Matrix Laboratory.

The diagram in Fig 3.1 below shows the main features and capabilities of MATLAB. MATLAB's built-in functions provide excellent tools for linear algebra computations, data analysis, signal processing, optimization, numerical solution of ordinary differential equations (ODEs), quadrature, and many other types of scientific computations. Most of these functions use state-of-the-art algorithms. There are numerous functions for 2-D and 3-D graphics as well as for animation. Also, for those who cannot do without their Fortran or C codes, MATLAB even provides an external interface to run those programs from within MATLAB. The user, however, is not limited to the built-in functions; he can write his functions in the MATLAB language. Once written, these functions behave just like built-in functions. MATLAB's language is very easy to learn and to use.



. Fig.3.1 A schematic diagram of MATLAB features.

3.1 MATLAB Robot Creation Flow Chart

In this section, a detailed discussion on the implementation, creation, and forward and inverse kinematics analysis of the robotic arm using the MATLAB tool is provided. The flow chart below explains the process starting with creating the robot, controlling the robot with input joint angles, forward kinematics, and inverse kinematics functions used for the implementation.

In this MATLAB R2019a was used for robot creation and simulation. In the MATLAB tool, the Robotic toolbox with the rvc feature which consists of the robotic 3D capability is initialized. With the RVC feature, the robot arm can be developed, controlled, and manipulated. we have a file, startup_rvm.m which calls several robot functions by calling the respective .m files of each function.

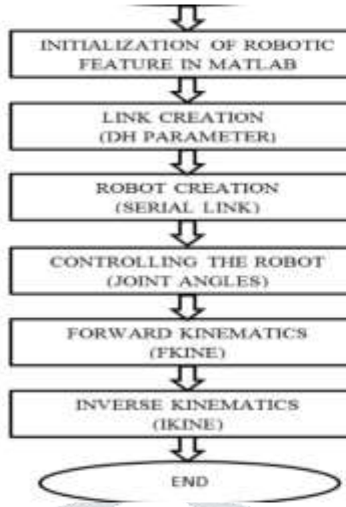


Fig.3.1.1 Flow chart.

A MATLAB file, LINK.m performs the process of link creation. This file holds all information related to a robot link such as the kinematics parameters, rigid body inertial parameters, motor, and transmission parameters. Also, there are classes and functions to receive the four parameters of the DH convention and construct the link. The fifth parameter of a link determines its type, revolute (0), or prismatic (1). The robot is created by serially connecting all links. The MATLAB file, SerialLink.m does connect the vector of link objects and forms a serial-link robot object. The output of SerialLink.m gives all the links information and robot dimensions. Figure 1.4 shows the output of SerialLink.m (Serial-Link robot).

In this, a robot with four links in the revolute configuration is considered. The D-H parameters of these four DOF robots are obtained from the result as represented in the following figures

Fig3.1.2 Four-link revolute robot (TRRR Arm)

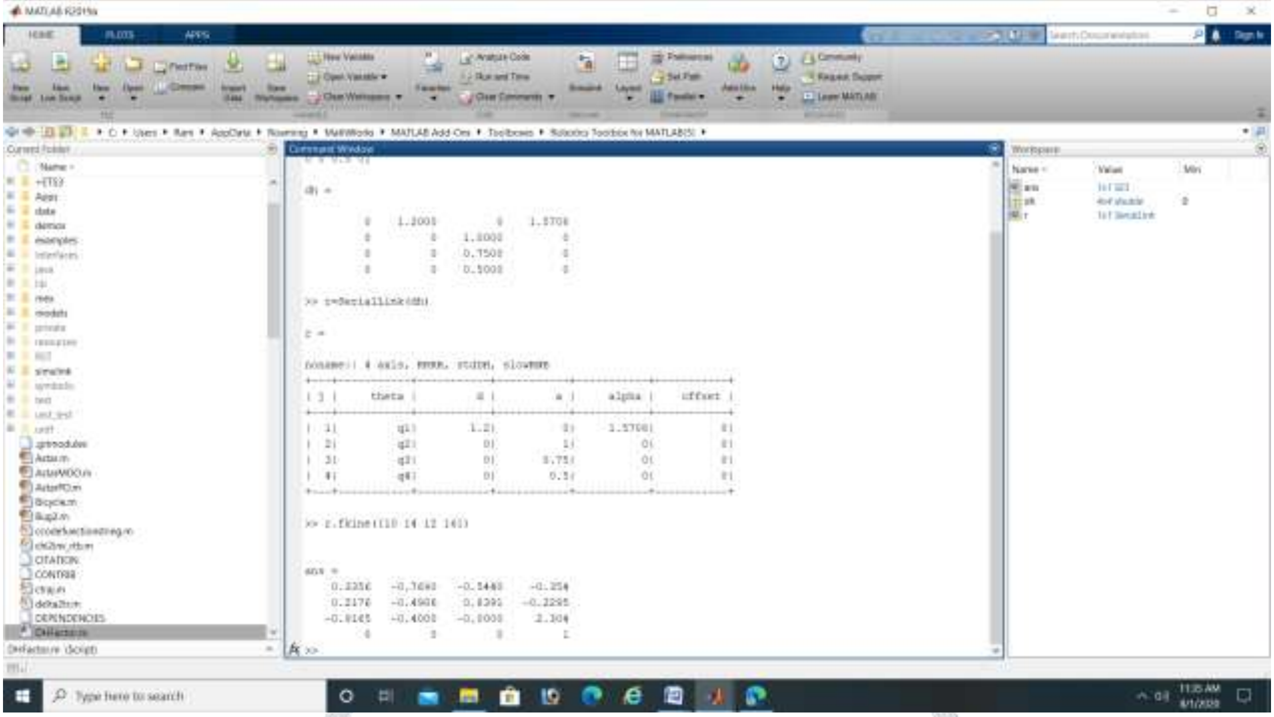


Fig3.1.3 Forward kinematics with joint angles (10,14,12,16)

Forward kinematics with joint angles (10,14,12,16) are obtained in a 4x4 matrix form from which the values of the position vectors are obtained.

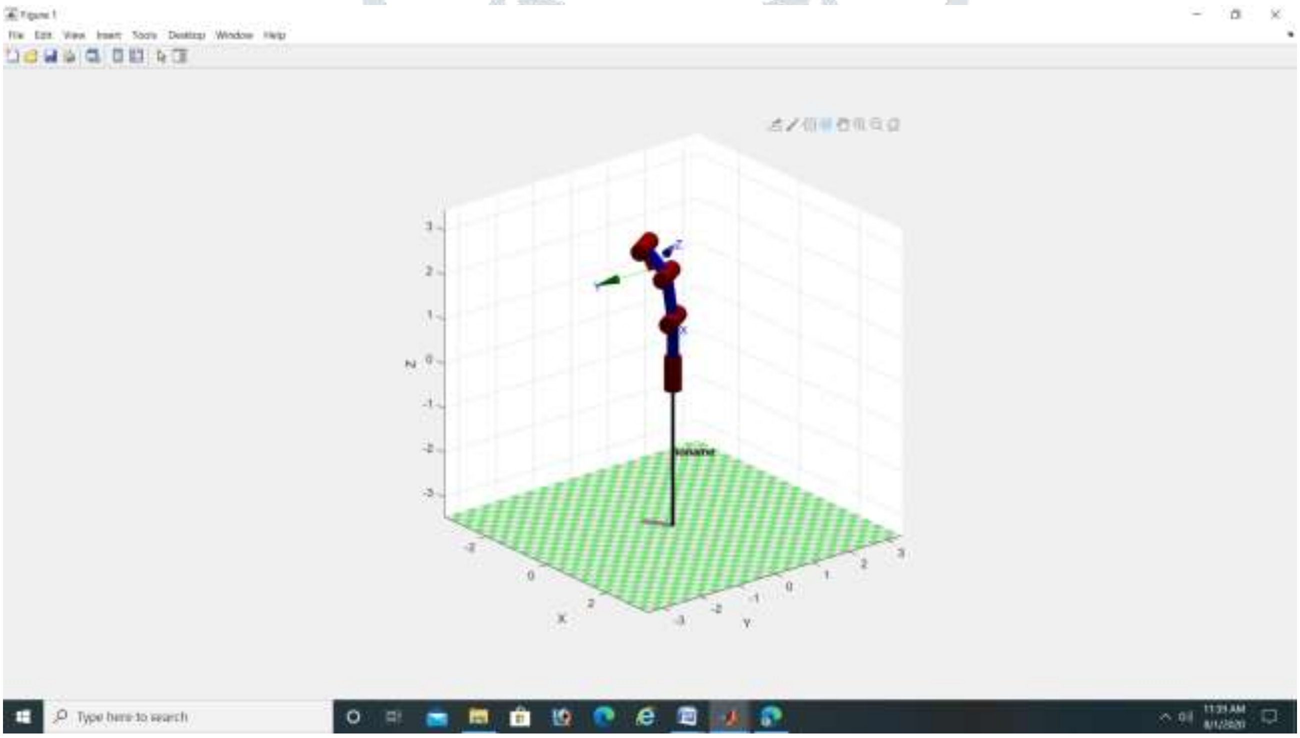


Fig3.1.4 Robot position with joint angles (10,14,12,16)

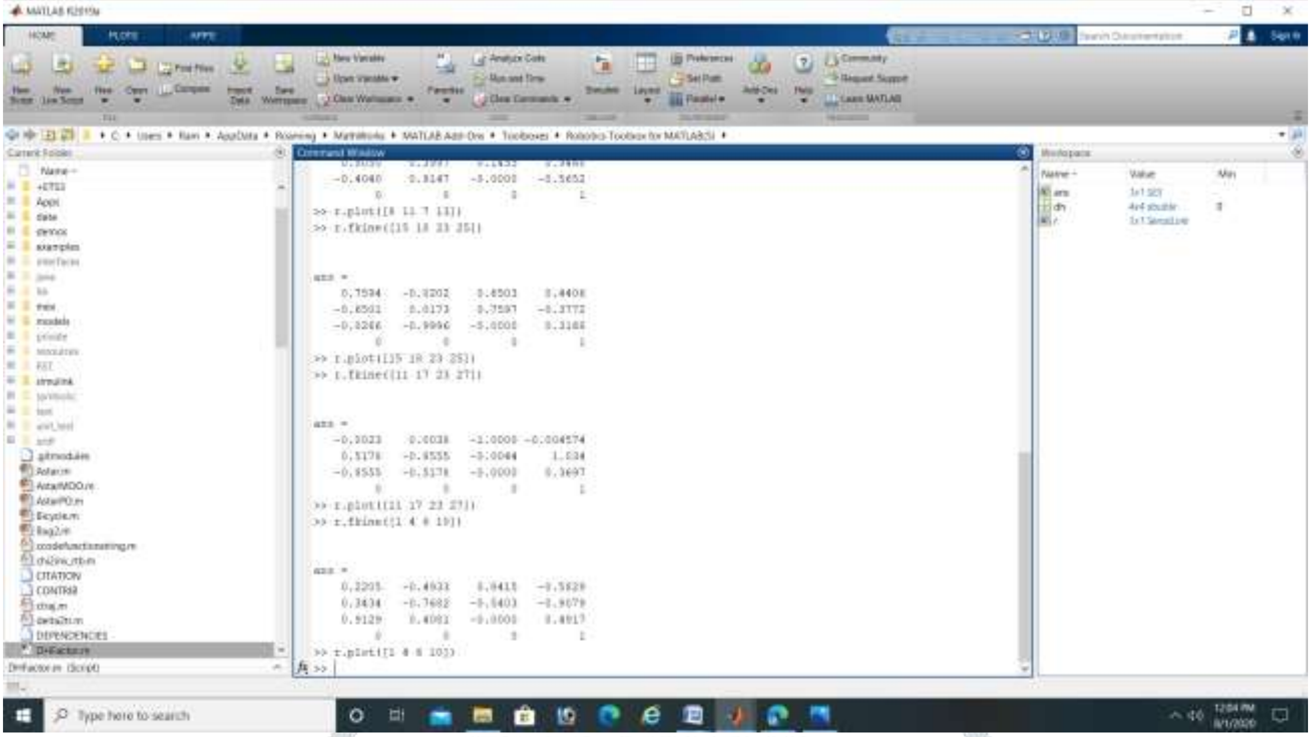


Fig.3.1.5 Forward kinematics with joint angles (1,4,6,10)

Forward kinematics with joint angles (1,4,6,10) obtained in 4x4 matrix form from which the values of the position vectors are obtained

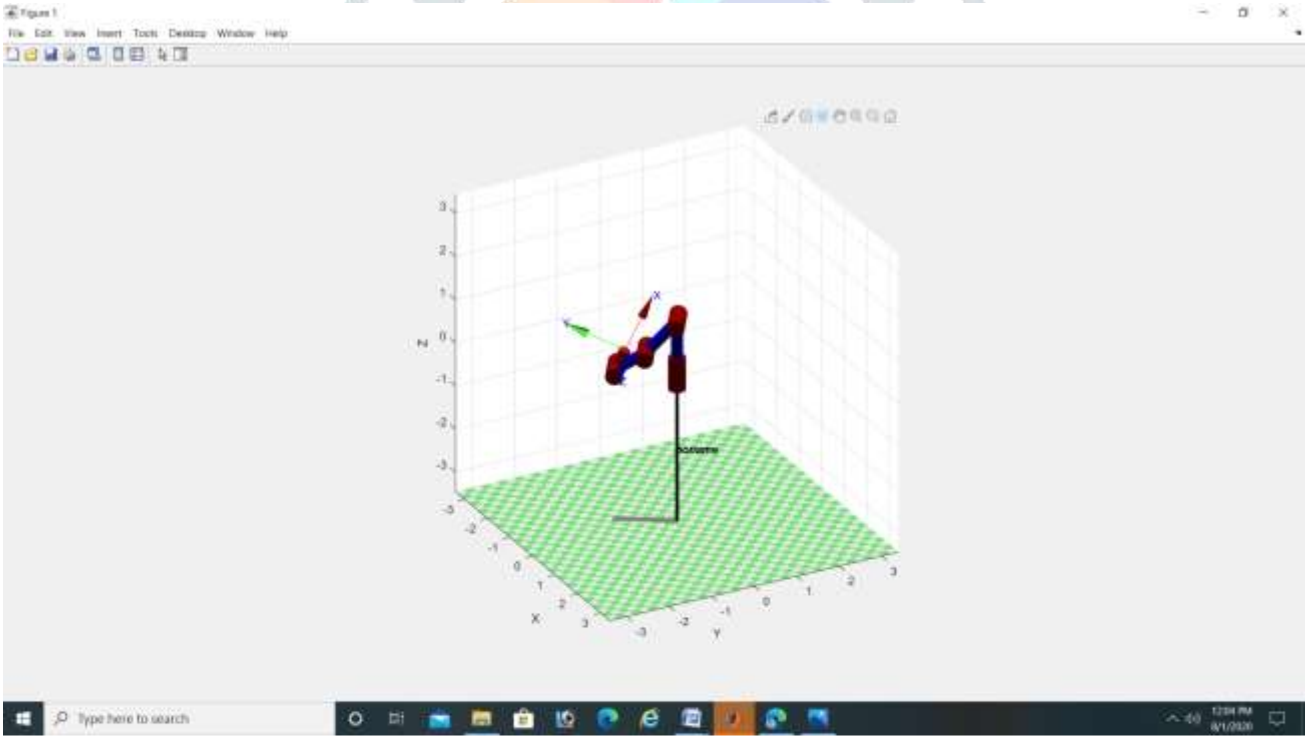


Fig 3.1.6 Robot position with joint angles (1,4,6,10)

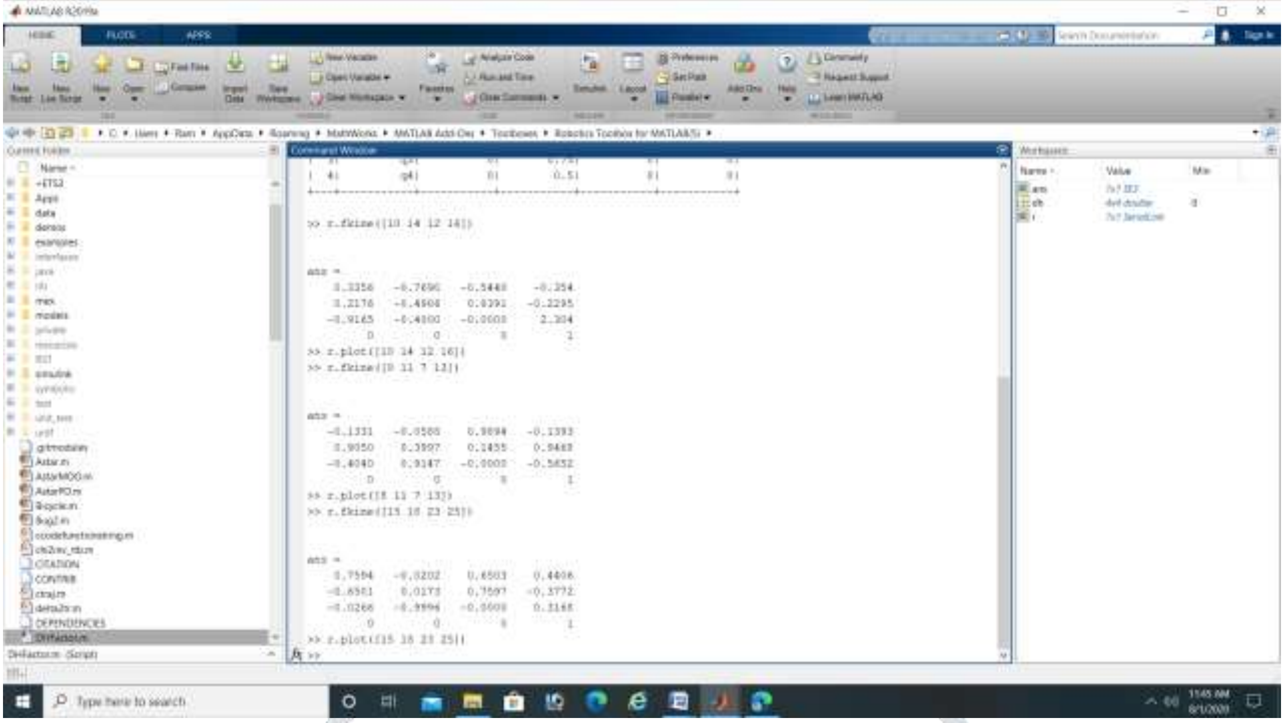
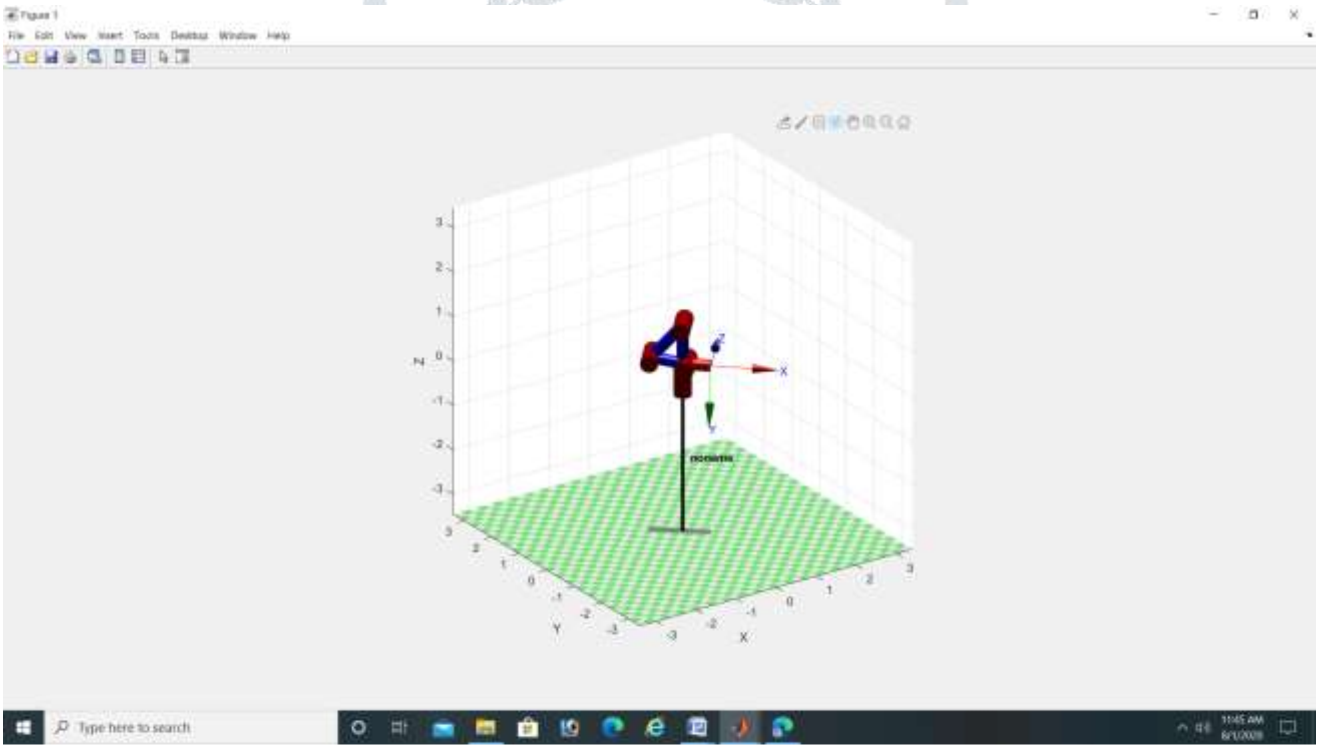


Fig.3.1.7 Forward kinematics with joint angles (15,18,23,25)

Forward kinematics with joint angles (15,18,23,25) are obtained in a 4x4 matrix form from which the values of the position vectors are obtained.



To view the 3D representation of the robot created, the Plot.m function is executed on the robot links to display the graphical animation based on the kinematic model. The plot function in the file joins the origins of the link co-ordinate frames and displays the 3D-View. The robot's movement can be controlled by changing the joint angles.

The file, fkine.m has a function that outputs the robot-end effector position as a homogenous transformation for the joint configuration. The result is provided in a matrix format as described in the forward kinematic chapter. The fkine.m file uses the transformation matrix method to provide the result. Similarly, the file ikine.m outputs the joint coordinates corresponding to the robot end-effector position.

IV.RESULTS AND SUMMARY

The following table shows the parameters of each link of the four-link robotic arm. The parameters are joint offset, link length, twist angle, and joint angle. Also, it shows the configuration of each link (prismatic or revolute). All these parameters together form the DH parameters for the four-link robotic arm used in this project.

Table 4.1: Robot link output

LINK	**Joint angle**	**Joint offset** (d_i)	**Link length** (a_i)	**Twist angle** (α_i)	**Prismatic or Revolute**
Link 1	q_1	1.2	0	$\frac{\pi}{2}$	Revolute (twist)
Link 2	q_2	0	1	0	Revolute
Link 3	q_3	0	0.75	0	Revolute
Link 4	q_4	0	0.5	0	Revolute

Table 4.1 shows the four joint-link Parameters for each input (example). The robot position for each input (example) is shown in the previous section. It is also proven that changing the angle of any joint would result in a different end-effector position.

. Table 4.2: Input Joint Angles

Joint Variable	**Joint angle 1**	**Joint angle 2**	**Joint angles 3**
θ_1	10	1	15
θ_2	14	4	18
θ_3	12	6	23
θ_4	16	10	25

Table 4.2 shows the input joint angles at three joints excluding the base joint in terms of four variables

Table 4.3: Forward Kinematics Result

Position vector	Joint angle 1	Joint angle 2	Joint angle 3
r_1	0.3356	0.2205	0.7594
r_2	-0.7690	-0.4933	-0.0202
r_3	-0.5440	0.8415	0.6503
r_4	0.2176	0.3434	-0.6501
r_5	-0.4986	-0.7682	0.0173
r_6	0.8391	-0.5403	0.7597
r_7	-0.9165	0.9129	-0.0266
r_8	-0.4000	0.4081	-0.9996
r_9	-0.0000	-0.0000	-0.0000
p_x	-0.354	-0.5829	0.4406
p_y	-0.2295	-0.9079	-0.3772
p_z	2.304	0.4917	0.3168

Table 4.3 shows the forward kinematics result for the input combinations (with three different joint angles) mentioned in table 3. r_1 - r_9 Vectors represent the orientation of the end-effector of the robot. Vector P represents the position of the end-effector of the robot

Table 4.4: Inverse Kinematics Result

Joint Variable	Joint angle 1	Joint angle 2	Joint angle 3
θ_1	10	1	15
θ_2	14	4	18
θ_3	12	6	23
θ_4	16	10	25

Table 4.4 shows the inverse kinematics result of different input combinations of the position vectors obtained in the forward kinematics result mentioned in table 4. With the orientation and position vectors as input, the joint angles are obtained as output. It proves that forward and inverse kinematics is an inverse function of each other.

V. CONCLUSION

In this Paper, Forward and Inverse Kinematic Analysis of a 4- DOF TRRR articulated arm was studied. Then simulation of the robotic arm was done using the MATLAB simulation tool. The simulation has detected the movement of each joint of the robot arm the motion of the robot was controlled in different directions using various joint angle combinations. The concept of forward and inverse kinematics was used to determine the end-effector position for fixed joint angles and the joint angles for a fixed end-effecter. Through research and implementation of the kinematics concept in the work, it became clear that inverse kinematics is widely used in practical applications. The result of the forward kinematics was also discussed for each joint angle combination emphasizing the difference in the configuration of the position vector. The results confirmed that the forward kinematics is the inverse function of the inverse kinematics, i.e. the forward kinematic problem uses the kinematics equations to determine the position given the joint angles. The inverse kinematics problem computes the joint angles for a desired position of the end effector.

.

REFERENCES

[1] Robot Kinematics, Wikipedia Web Site. http://www.wikipedia.com

[2] Kay, J. Introduction to Homogeneous Transformations & Robot Kinematics, Rowan University Computer Science Department

[3] Mittal, RK; Nagrath, J; Robotics and Control, Tata McGraw-Hill. 2005

[4] Denavit, J. and Hartenberg, R.S. 1955. A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices. Journal of Applied Mechanics 22 (June 1955), 215 – 221

[5] Shah, J., Rattan, S. S., and Nakra, B.C. 2013. EndEffector Position Analysis Using Forward Kinematics for 5 DOF Pravak Robot Arm. International Journal of Robotics and Automation 2(3) (Sept. 2013), 112 - 116.

[6] Desh ande VA, and P.M. George, "Analytical solution for inverse kinematics of Scorbot ER Vu Plus, International Journal of Emerging Technology Advanced. Engineering, Vol. 2, pp. 478-481, 2012.

[7] Niku, SB; Introduction to Robotics: Analysis, Systems, Applications. Prentice-Hall. 2001