KINEMATIC ANALYSIS OF SIX DEGREE OF FREEDOM 6R ROBOTIC MANIPULATOR

[1]Eklavya Gupta, [2]Prof. (Dr.) Sunil K. Somani
[1]Research Scholar, [2]VC (Medi-Caps University) & Ph. D. Supervisor
[1]Mechanical Engineering
[1]Mewar University, Chittorgarh, India

Abstract: Robotics is a developing field and enterprises need robots to expand the efficiency. For a specific application, it is wanted to configuration, investigate and simulate the robotic manipulator. The investigation of its movement is one of the imperative viewpoints in planning a robotic manipulator. Kinematics and dynamics are the two critical parameters in the investigation of movement. Kinematics is about the investigation of movement for example of position, speed and acceleration without thinking about the forces. While dynamics manages the investigation of movement by thinking about forces and material properties. Kinematics is classified into forward and inverse kinematics. Forward kinematics manages the investigation of end effector's situation by knowing the joint points and inverse kinematics manages the investigation of the angular positions of the joints for the given end effector's position. In order to get the position of end effector & joint angles, many simulation softwares have been developed by using high level languages. In this work, a Matlab Graphical User Interface Development Environment (GUIDE) is developed using Matlab & is used for forward and inverse kinematics. Using this, on entering the required input parameters, the end positions or the joint angles can be obtained. Also the simulation between initial & final position has been shown.

IndexTerms - Robotic Manipulator; Denavit-Hartenberg (D-H) Parameters; Forward Kinematics; Inverse Kinematics

I. INTRODUCTION

Robotics is an emerging field in every aspects of life. It has been a challenging task to control robotic arm since the inception of robots. Kinematics plays a very vital role in it. The kinematics problem is defined as the transformation from the Cartesian space to the joint space and vice versa. Forward kinematics refers to the use of the kinematic equations of a robot manipulator to compute the position of the end-effector from specified values for the joint parameters. Initially analytic methods were used but now various software based methods are being used by the researchers. The solution for forward kinematics is obtained by multiplying the individual matrices formed by D-H conventions and computing the final transformation matrix which gives the position and orientation elements for inverse kinematic solution. Research works are continuously going on to reduce the complexity. In this work, an attempt is made to get the solution of forward and inverse kinematics with the help of Matlab GUIDE.

II. LITERATURE SURVEY

Analysts (Parthasarathy, Srinivasaragavan, & Santhanakrishnan, 2017) have performed dynamic modeling and simulation of 5-DOF Mitsubishi RM 501 robot and developed algorithm to control comparative sort of robot. SolidWorks software was used for modeling. Adams and Matlab-Simulink were used to perform motion studies, control algorithm simulation, testing and validation. To check the accuracy and precision of the simulation, results of Adams and Matlab-Simulink were compared and verified. Authors (Mariappan & Veerabathiram, 2016) have done modeling and simulation of multi spindle drilling redundant SCARA robot which is used to drill multiple holes in printed circuit board and sheet metal. Modeling is done using SolidWorks software and to study dynamic 3D CAD model of robot is changed in SimMechanics block diagram by exporting into Matlab/SimMechanics, utilizing its motion sensing capability the dynamic parameters viz. velocity and torque of the manipulator are studied and necessary improvement in design of robot is performed. Analysts (Senthilkumar & Parthiban, 2016) have performed modeling, kinematic analysis and simulation of 4-DOF TRLR Polar robot manipulator. Numerical model of kinematic investigation is created by utilizing D-H technique. SolidWorks was used for modeling, forward-inverse kinematics done by D-H transformation matrix, RoboAnalyzer toolbox and Matlab-Simulink toolbox were used for dynamic analysis and mathematical model simulation. Authors (Almusawi, Dulger, & Kapucu, 2016) have performed new inverse kinematic solution for Denso 6-DOF robotic manipulator using artificial neural network (ANN). The modern ANN is a better tool for calculation of inverse kinematics.

Researchers (Filiposka, Djuric, & Elmaraghy, 2015) have studied the kinematic model of a 6-DOF CNC Gantry machine. Forward kinematics is done by using the Maple. And, end effector positions have been plotted by using Matlab. Analysts (Abubakar, Zhongmin, & Ying, 2014) have performed kinematic analysis of 6-DOF Articulated robot by using Matlab, simulation demonstrates that robot is reasonable for taking care of light weight material in a mechanical production system at low speed. Researchers (Jha & Biswal, 2014) have tried to solve inverse kinematics of 4-DOF SCARA robot by using ANN. The ANN model used Multi Layered Perceptron Neural Network (MLPNN) where in gradient descent type of learning rules is applied as MLPNN generates minimum mean square error. ANN can always be update by providing data pair structure and parameter by presenting new training example. Analysts (Fang & Li, 2013) have modeled 4-DOF SCARA robot. Matlab was used to perform simulation and kinematic analysis. Authors (Piltan, Emamzadeh, Hivand, Shahriyari, & Mirzaei, 2012) have focused on robot manipulator analysis and implementation, and design, analyzed and implement nonlinear sliding mode control method. Matlab-Simulink software is used to study the kinematics and dynamics of puma robot. Researchers (Shanmugasundar & Sivaramakrishnana, 2012) have developed VB based software for forward & inverse kinematics for a 7-DOF robotic manipulator. The outcomes are compared with manual computations. Author (Jasim, 2011) have performed calculation of complex inverse kinematics of 4-DOF SCARA robot by using neuro fuzzy network ANFIS. Adaptive neuro fuzzy network enabled ease in operation, and errors are within the acceptable limits and also the simulation is quite fast.

III. Robot Kinematics

Robot Kinematics deals with the movement of multi-degree of freedom kinematic chains that form the structure of robotic systems. It is the study of relationship between the dimensions and connectivity of kinematic chains which gives the position, velocity and acceleration of each link in the robotic system. It is classified into forward kinematics & ineverse kinematics. Figure 1 depicts the relation between the joint angles & end-effector postion via forward & inverse kinematics. Forward kinematics uses the kinematic equations of a robot to compute the position of the end-effector from specified values for the joint parameters. It is the kinematics in which kinematic equations are used to find out /calculate the position of end effector from a particular value for the joint parameters. Process which is used to computes the joint parameters that achieve a specified position of the end-effector is known as inverse kinematics. Inverse kinematics specifies the end-effector location and computes the associated joint angles. The dimensions of the robot and its kinematics equations define the volume of space reachable by the robot, known as its workspace.

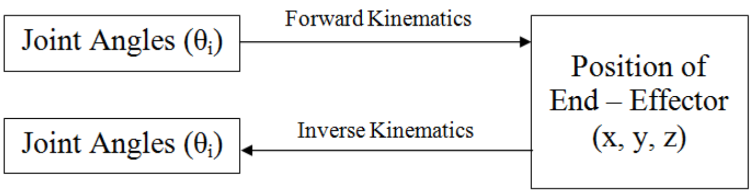

Fig. 1. Relation between Joint angles & end – effector position (Gupta & Somani, 2018)

IV. Denavit-Hartenberg Parameters

In robotic applications, Denavit-Hartenberg, or D-H convention is the most commonly used convention which is used for selecting frames of reference as shown in fig 2. The four parameters a_i, α_i, d_i, and θ_i are named as link length, link twist, joint offset and joint angle, respectively shown in table 1. Each homogeneous transformation T_i is represented as a product of four basic transformations. In 1955 Denavit and Hartenberg proposed a matrix method to describe the relationship of translation or rotation between two consecutive links. It is based on the D-H Coordination system. It is basically to form the attached coordinate system on each link in the joint chains of the robot. The transformations between each two successive joints can be put forth by just substituting the parameters from the parameters table into the T matrix using equation (1).

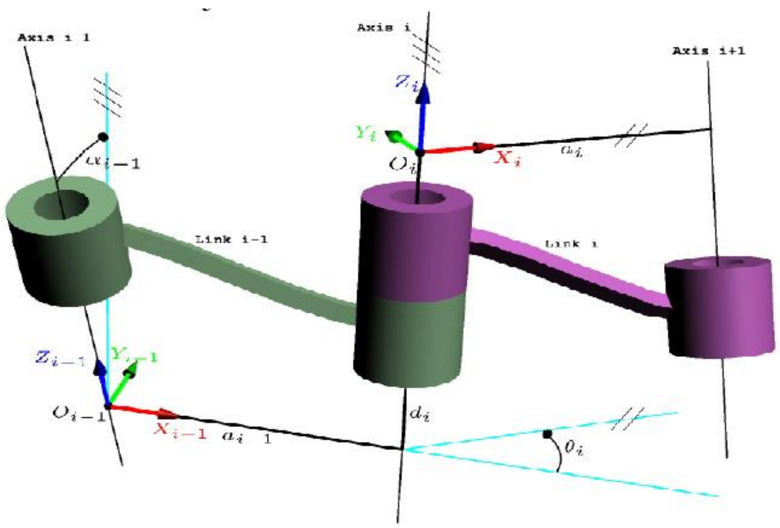

Fig. 2. A representation of D-H Parameters (Rao, Saraf, & Paymal, 2015)

Table 1. The Denavit – Hartenberg Parameter (Generalised) **(Gupta & Somani, 2018)**

Joint No.	Joint Angle (θ°)	Joint Offset (d)	Link Length (a)	Twist Angle (α°)
1	θ_1	d_1	a_1	α_1
2	θ_2	d_2	a_2	α_2
3	θ_3	d_3	a_3	α_3
...	…	…	…	…
…	…	…	…	…
N	θ_n	d_n	a_n	α_n

$$^{i-1}T_i = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Transformations are in consecutive order starting from first joint to nth joint. It will go on to the arm-end of the robot and finally reach to the end effectors. Equation (2) is showing the complete transformation upto end-effector.

$$^0T_n = {}^0T_1 {}^1T_2 \ldots {}^{n-1}T_n$$

$$= \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

For inverse kinematics, the inverse of first transformation matrix is multiplied with the complete transformation matrix obtained in equation (2). On comparing the different elements of matrices getting from equation (3), the values of joint angles are obtained.

$$({}^0T_1)^{-1} \times {}^0T_n = {}^1T_2 \ldots {}^{n-1}T_n \quad (3)$$

where, $({}^0T_1)^{-1} = \frac{Adj({}^0T_1)}{Det({}^0T_1)}$

V. ALGORITHM

In this work, Matlab and robotic toolbox (Corke, 2007) are used as software tools for kinematic analysis. Flowchart of the algorithm used to prepare the Matlab GUIDE for kinematic analysis is shown in fig 3. In this, first of all, user need to insert the D-H parameters i.e. length, link twist, joint offset, then if user wants to go for forward kinematics, the joint angles are entered as input. On clicking the forward kinematics button the output i.e. the end positions will be displayed. And also the simulation of the arm will take place with that particular end position by maintaining the joint angles in the figure window. If user wants to go for the inverse kinematics, he needs to enter the end position of the manipulator, then on clicking the inverse kinematics button, Matlab GUIDE will compute the joint angles and will display them.

VI. MATLAB GUIDE

A Matlab GUIDE has been developed in this research, for both forward kinematic and inverse kinematic analysis. Fig. 4 indicates the screenshot of Matlab GUIDE. Fig. 5 depicts the screenshot for input of D-H Parameters for 6-DOF robotic manipulator. Fig. 6 comprises the screenshot of 6R manipulator at its initial position. Fig. 7 displays the screenshot containing the details of joint angles for forward kinematics & the end position after calculations. Fig. 8 showing the screenshot of final position of 6R manipulator after forward kinematics calculation. Fig. 9 displays the screenshot containing the details of end position for inverse kinematics and joint angles after calculations.

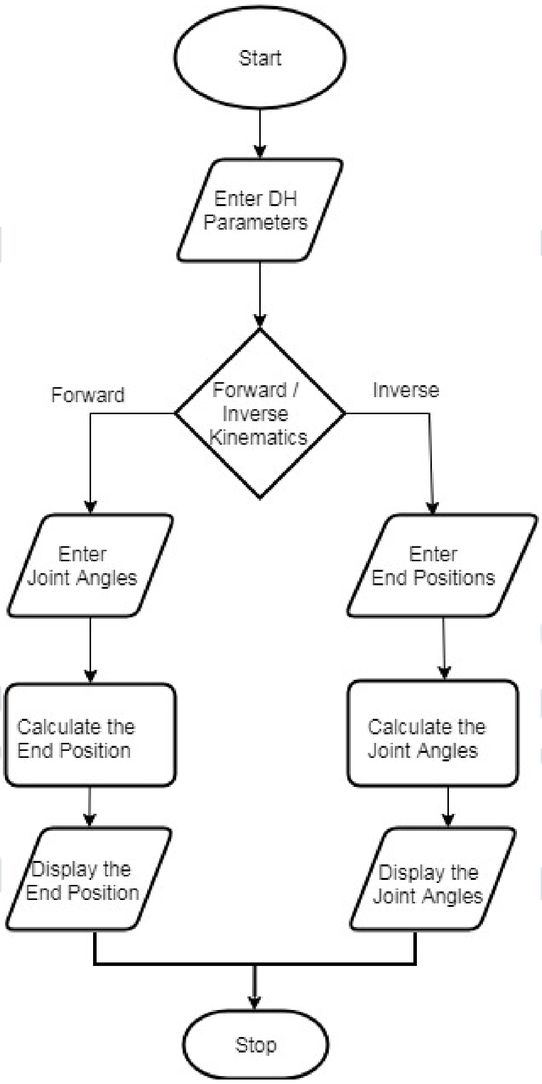

Fig. 3. Algorithm Flowchart

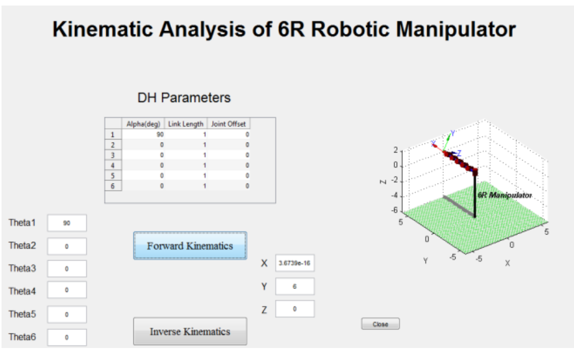

Fig. 4. Screenshot showing Matlab GUIDE

DH Parameters

	Alpha(deg)	Link Length	Joint Offset
1	90	1	0
2	0	1	0
3	0	1	0
4	0	1	0
5	0	1	0
6	0	1	0

Fig. 5. Screenshot showing input of D-H Parameters

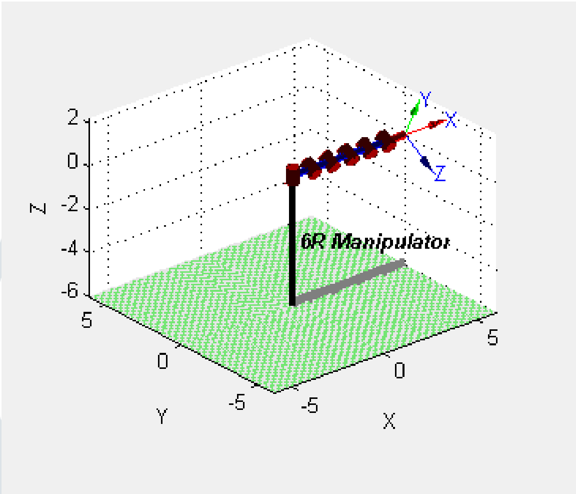

Fig. 6. Screenshot showing initial position of 6R manipulator

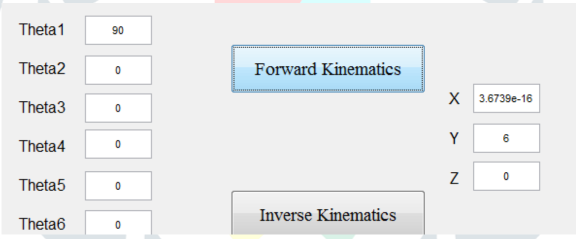

Fig. 7. Screenshot showing forward kinematic results

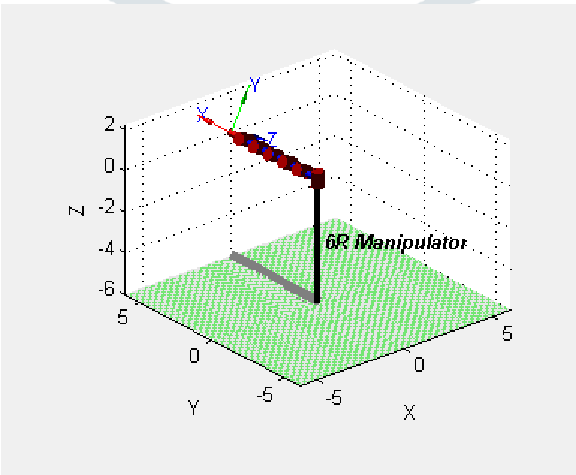

Fig. 8. Screenshot showing final position of 6R manipulator after forward kinematics

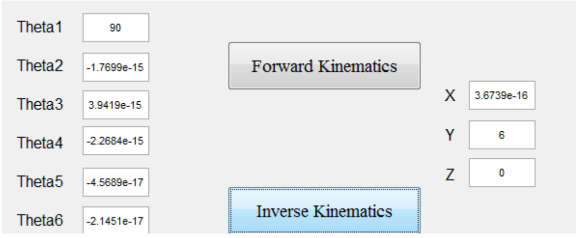

Fig. 9. Screenshot showing inverse kinematic results

VII. Results

For particular DH parameters shown in fig. 5, the forward and inverse kinematics has been computed. Table 2 and 3 are showing the expected and obtained results for forward and inverse kinematics respectively. After getting the obtained results, it can be seen that percentage errors are almost negligible as the difference in the values are more than e-15.

Table 2. Forward Kinematics Results

θ_1	90	Expected	X	0
θ_2	0		Y	6
θ_3	0		Z	0
θ_4	0	Obtained	X	3.6739e-16
θ_5	0		Y	6
θ_6	0		Z	0

Table 3. Inverse Kinematics Results

		Expected		Obtained	
X	3.6739e-16	θ_1	90	θ_1	90
		θ_2	0	θ_2	-1.7699e-15
Y	6	θ_3	0	θ_3	3.9419e-15
		θ_4	0	θ_4	-2.2684e-15
Z	0	θ_5	0	θ_5	-4.5689e-17
		θ_6	0	θ_6	-2.1451e-17

VIII. Conclusions

As robotics world is witnessing excessive complexities in controlling the robotic arm, present research is intended to reduce this complexity and come out with easy method of calculations of forward kinematics and inverse kinematics. It is evident that the Matlab GUIDE can easily be used and at the same time gives valid and authentic results. The results obtained for forward and inverse kinematics through Matlab GUIDE are with negligible errors or say having zero error. With this Matlab GUIDE various end positions can be obtained by entering the different joint angles using forward kinematics. Also various joint angles can be obtained by entering the different end positions using inverse kinematics. For future work, dynamic analysis can be done using Matlab.

References

[1] Abubakar, U., Zhongmin, W., & Ying, G. (2014). Kinematic Analysis and Simulation of a 6 DOF Industrial Manipulator. *International journal of Science and Research , 3* (3), 1-4.

[2] Almusawi, A. R., Dulger, L. C., & Kapucu, S. (2016, July 18). A New Artificial Neural Network Approach in Solving Inverse Kinematics of Robotic Arm. (D. W. kim, Ed.) *Computational Intelligence and Neuroscience* , 1-10.

[3] Corke, P. I. (2007). MATLAB toolboxes: robotics and vision for students and teachers. *IEEE Robotics and Automation Magazine , 14* (4), 16-17.

[4] Fang, J., & Li, W. (2013). Four Degree of Freedom SCARA Robot Kinematics Modeling and Simulation Analysis. *International Journal of Computer Consumer and Control , 2* (4), 20-27.

[5] Filiposka, M., Djuric, A. M., & Elmaraghy, W. (2015). Kinematic Analysis of a 6 DOF Gantry Machine. *SAE International* , 1-7.

[6] Gupta, E., & Somani, S. K. (2018). Forward Kinematics of Robotic Manipulator Using Electronic Spreadsheet. *Innovative Research in Technology, Pharmacy, Management , Science & Humanities*. Indore.

[7] Jasim, W. M. (2011). Solution of Inverse Kinematics for Scara Manipulator using Adaptive Neuro Fuzzy Network. *International Journal on Soft Computing , 2* (4), 59-66.

[8] Jha, P., & Biswal, B. (2014). A Neural Network Approach for Inverse Kinematic of a SCARA Manipulator. *International Journal of Robotics and Automation , 3* (1), 52-61.

[9] Mariappan, S. M., & Veerabathiram, A. (2016, December). Modelling and Simulation of Multi Spindle Drilling Redundant SCARA Robot using Solidworks and MATLAB/ SimMechanics. *Faculty of Engineering, University of Antioquia* , 63-72.

[10] Parthasarathy, T., Srinivasaragavan, V., & Santhanakrishnan, S. (2017). Adams MATLAB co-simulation of a Serial Manipulator. *International Conference on Mechanical and Manufacturing Engineering MATEC Web of Conferences* , 1-6.

[11] Piltan, F., Emamzadeh, S., Hivand, Z., Shahriyari, F., & Mirzaei, M. (2012). Puma 560 Robot Manipulator Position Sliding Mode Control Methods using MATLAB/SIMULINK and their Integration into Graduate/Undergraduate Nonlinear control, Robotics and MATLAB courses. *International Journal of Robotic and Automation , 6* (3), 106-150.

[12] Rao, C. R., Saraf, M. R., & Paymal, R. (2015, May 10). A Review on Latest Simulation Softwares for Kinematic Analysis. *Proceeding of 26th IRF International Conference* , 17-20.

[13] Senthilkumar, K., & Parthiban, A. (2016). Modeling, Simulation and Kinematic Analysis of 4 DOF Polar Robot Manipulator using Robo Analyzer and Matlab Software. *International Journal of Modern Trends in Engineering and Science , 3* (12), 18-22.

[14] Shanmugasundar, G., & Sivaramakrishnana, R. (2012). Software Development for an Inverse Kinematics of Seven degree of Freedom Newly Designed Articulated Inspection Robot. *International Journal of Computer Application , 58* (18), 29-36.