International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 2, Issue 4, pp 6-10, June-2018

Simulation and Modeling of 6-DOF Robot Manipulator Using Matlab Software

[1]Thavamani.P, [2]Ramesh.K, [3]Sundari.B

[1]M.E Scholar, Applied Electronics, JCET, Dharmapuri, Tamilnadu, India
[2]Associate Professor, Dept. of ECE, JCET, Dharmapuri, Tamilnadu, India
[3]Assistant Professor, Dept. of ECE, JCET, Dharmapuri, Tamilnadu, India

***Abstract* - Mechanical robotic arms have been developing in a substantially complete manner in recent years. Robotic arms can support people in remote areas or in various regions where access is limited to move objects easily and perform things with high speed and precision. A robotic arm is basically a mechanical arm which is programmable with performance same as a human arm. Robotic arms can be a part or total mechanism of a complex robot. Robotic hands are capable of doing any work that is required in designs such as painting, welding an object and so on. For example, robotic arms are doing a variety of tasks such as drilling, nut tightening of rotating parts and spare parts in the assembly lines of car factories. The purpose of this paper is design of a robotic arm with 6 DOF and motion simulation using a MATLAB software code and Simulink modeling. The equations of position, angles and paths are being controlled using a GUI interface. By the slider controls in the coding position angle and path of motion range will be displayed and path traces are drawn by tracking the arm movements in 3D-space using a system block of individual joints in the diagram of Simulink environment.**

Keywords: 6 DOF, Robotic ARM, Manipulator, Simulink, MATLAB, Mechanical ARM.

I. INTRODUCTION

a) Robot Description

The robot is a 6 degree of freedom robot arm manipulator as shown in the figure. It has one prismatic joint and five revolute joints giving a configuration of RRPRRR. The last three revolute joints constitute a wrist and the first three, an arm. It has total 7 links including the base link [1]. The arm accomplishes the task of reaching the desired position and the wrist helps in the orientation of the tool. All the joints are actuated by DC motors and have only one degree of freedom.

b) Forward Kinematics

Forward kinematics gives the position and orientation of the end-effector matrix for an n-degree of freedom manipulator with respect to a base frame. Here the joint variables are known and the manipulator under consideration is a 6-degree of freedom Stanford manipulator. In revolute joints, the angles between the links are the joint variables [2]-[3]. In prismatic joints, the link extension and contraction are the joint variables. The overall transformation matrix describes the position of end effector or tool-frame with respect to the fixed base frame. For this purpose, frames are assigned to each link starting from the base frame. Thus a manipulator having n links will have n+ 1 frame with the reference base frame named as {0}. A transformation matrix relating two frames attached to two adjacent links is obtained and an overall description of the tool-frame is obtained by combining the individual transformation matrices [4]. DH notation describes the assignment of frames to each link in a kinematic chain and defines a manipulator in terms of four joint link parameters. The four parameters are link length, link twist, joint distance and joint angle.

c) Links and Joints Convention

A robot manipulator having n DOF is modeled as a manipulator having n joints, as one joint has only one degree of freedom unlike a ball socket joint that has two degrees of freedom [5]. It is an open kinematic chain of rigid links connected to each other by revolute or

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 2, Issue 4, pp 6-10, June-2018

prismatic joints to allow relative motion between the two links. It can be concluded that an n DOF manipulator has n joints and n+1 links. The links are numbered from 0 to n and the joints from 1 to n. Thus joint i connects link i-1 and link i. Thus to move link i, we need to actuate joint i. The Joint Parameters of Stanford Manipulator are listed in table below

Sr. No	Joints	a (mm)	α (deg)	d (mm)	Initial Value(JV)	Final Value(JV)
1	Joint 1	0	-90	0.763	0	80
2	Joint 2	0	-90	0.393412	-90	[illegible]
3	Joint 3	0	0	Variable	[illegible]	[illegible]
4	Joint 4	0	-90	0.2268	0	60
5	Joint 5	0	-90	0	180	150
6	Joint 6	0	0	0.4318	180	150

Table-1: D-H Parameters of a Manipulator

Since joint angle and joint variable are the only variables in revolute and prismatic joint respectively we will be taking 4 sets of values for these D-H parameters and finding the end effectors for these four sets using forward kinematics [6]-[8]. The step-by-step frame assignment for each joint-link of the manipulator is shown below. Frames are first assigned to the intermediate links, links 1 to 5 and then to link 0 and link 6.

II. PROPOSED SYSTEM

A six DOF robotic arm which consists of six links is developed using MATLAB simulation tool and then the forward and inverse kinematics analysis is performed in this project. The six DOF manipulator will have six joints from joint 1 to joint 6 and six links from link 0 to link 5. In this project, forward kinematics is calculated using the transformation matrix method. A model in matlab using robotics tool bar is created to determine the scope and motion simulation and robot arm design.

III. MATLAB MODELLING

Matlab Robot Creation Flow Chart

In this chapter, a detailed discussion on the implementation, creation and the forward and inverse kinematics analysis of the robotic arm using 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 project, we used MATLAB R2016B for robot creation and simulation. In the MATLAB tool, RVC feature which consists of the robotic 3D capability is initialized. With the RVC feature, the robot arm can be developed, controlled and manipulated. I have a file, startup_rvm.m which calls several robot functions by calling the respective .m files of each function [9]-[10]. 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. In addition, 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 connecting all links together in a serial fashion. The MATLAB file, SerialLink.m does connect the vector of link objects and forms a serial-link robot object.

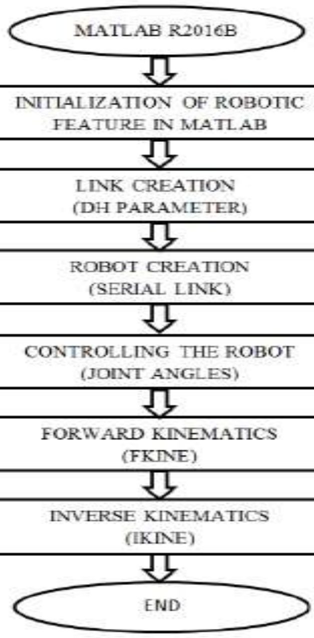

Figure-1: Matlab Program Flowchart

The MATLAB coding was used to do mathematical iterations of the serial link manipulator. The variables $\theta 1$, $\theta 2$, $\theta 3$, and $\theta 4$ respectively represent the joint

ISSN (online): 2581-3048

Volume 2, Issue 4, pp 6-10, June-2018

axes 1 through 4. Kinematics equations from the overall transformation matrix were developed using the MATLAB R 14.0b. A code has been developed to generate the forward kinematics equations and calculate the robot Manipulator position and orientation in terms of joint angles and its output is compared with Robot software (which is the simulate program supplied with the robot system) for four sets of joint parameters. The result of end-effecter's position from MATLAB iteration was then compared with experimental result generated from inbuilt software tool box [11]. For different keyboard values entered on the software, the corresponding joint angles, simulation and experimental positions for the end-effecter are presented.

The output of SerialLink.m gives all the links information and robot dimensions. Figure 7 shows the output of SerialLink.m (Serial-Link robot). The DH parameters of each link and also shown in figure. With the help of the MATLAB code for a 6-DOF STANFORD Manipulator, the end effector position was obtained for four different sets of joint variables (joint angles). The input values for the prismatic joint were also varied and the results were obtained.

IV. RESULT SUMMARY

a) Robot link output

LINK	Joint angle	Joint offset (d_i)	Link length (a_i)	Twist angle (α_i)	Prismatic or Revolute
Link 1	Q1	0	0	-pi/6	Revolute
Link 2	Q2	0.5	0.699	0	Revolute
Link 3	Q3	0.0948	0.0948	pi/2	Revolute
Link 4	Q4	0.68	0	-pi/2	Revolute
Link 5	Q5	0	0	pi/2	Revolute
Link 6	Q6	0.853	0	0	Revolute

Table-2: Robot link output

Table 2 shows the parameters of each link of the six-link robotic arm. The parameters are joint offset, link length, twist angle and joint angle. In addition, it also shows the configuration of each link (prismatic or revolute). All these parameters together form the DH parameters for the six link robotic arm used in this project.

b) Input joint angles

Table 3 shows the six joint angles for each input (example). The robot position for each input (example) is shown in the previous chapter. It is also proven that changing the angle of any joint would result in a different end-effector position.

Joint angle	Input 1	Input 2	Input 3	Input 4	Input 5	Input 6	Input 7
θ_1	11	9	13	6	50	9	7
θ_2	16	18	18	11	48	16	14
θ_3	21	19	25	21	42	25	20
θ_4	29	17	37	30	31	35	23
θ_5	41	27	44	39	23	43	34
θ_6	51	36	50	46	15	51	49

Table-3: Input joint angle

c) Forward kinematics result

Position vector	Output 1	Output 2	Output 3	Output 4	Output 5	Output 6	Output 7
Nx	-0.6339	0.5071	0.6414	0.9870	-0.7088	-0.0487	0.4069
Ny	-0.6871	-0.8599	-0.2589	-0.0348	0.6055	-0.2410	0.2719
Nz	0.3515	-0.0583	-0.4785	0.1521	0.2346	-0.8880	0.8721
Ox	-0.4250	-0.8597	0.5456	-0.1251	-0.6426	0.8466	-0.7758
Oy	-0.0604	-0.4999	-0.0833	-0.7394	-0.4732	0.5295	-0.4028
Oz	-0.9026	-0.1052	0.8139	0.6303	-0.6042	-0.1158	0.4871
Ax	0.6442	0.0614	-0.5384	0.0932	-0.2916	0.5390	0.4835
Ay	-0.7234	0.1033	-0.7865	-0.6497	-0.5788	-0.8428	-0.0741
Az	-0.2485	-0.9928	0.2733	-0.7545	0.7615	0.0940	0.0468
Px	0.1644	0.2368	-0.2747	0.3070	-0.0804	0.9605	-0.2513
Py	0.4157	0.7299	-1.2823	-1.4058	-0.3631	-1.7079	0.5865
Pz	0.0945	0.2215	1.2388	0.4706	1.2304	0.3677	-0.1045

Table-4: Forward kinematics result

Table shows the forward kinematics result for the input combinations (with six different joint angles) mentioned in table 3. Vectors N, 0 and A represent the

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 2, Issue 4, pp 6-10, June-2018

orientation of the end-effector of the robot. Vector P represents the position of the end-effector of the robot.

d) Inverse kinematics result

Joint angle	Input 1	Input 2	Input 3	Input 4	Input 5	Input 6	Input 7
θ_1	11	9	13	6	50	9	7
θ_2	16	18	10	11	48	16	14
θ_3	33	19	25	31	42	23	20
θ_4	28	17	17	30	31	33	23
θ_5	41	27	44	39	23	43	34
θ_6	31	36	30	40	13	32	49

Table-5: Inverse kinematics result

Table 5 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 those forward and inverse kinematics are inverse function of each other.

e) 6DOF Robot Manipulator

With the cad2matdemo Matlab program here a 6 link 3D Robot Manipulator Arm is simulated. The robot arm is moved with the kinematic control through direct input of joint angles and also by slider bars in the GUI. The inverse kinematics options are also been developed built in but the feature has been not yet enabled for automation.

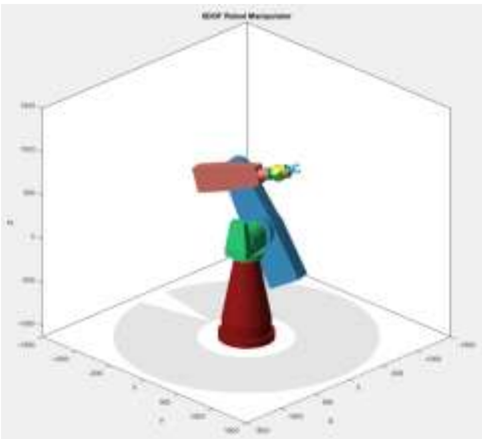

Figure-2: Simulink Model of Manipulator Arm

A random input or particular co-ordinates can also be provided in the interface window and thus the robot animates between those set locations. Here two files are used which are the main program and the other one is a six 3D-links data file that will be read during the start up. This Simulation model of a 6-DOF robot manipulator arm is capable of solving both the Forward and Inverse Kinematics based problems. The GUI designed here has been provided with the feature of slider based angular motion control.

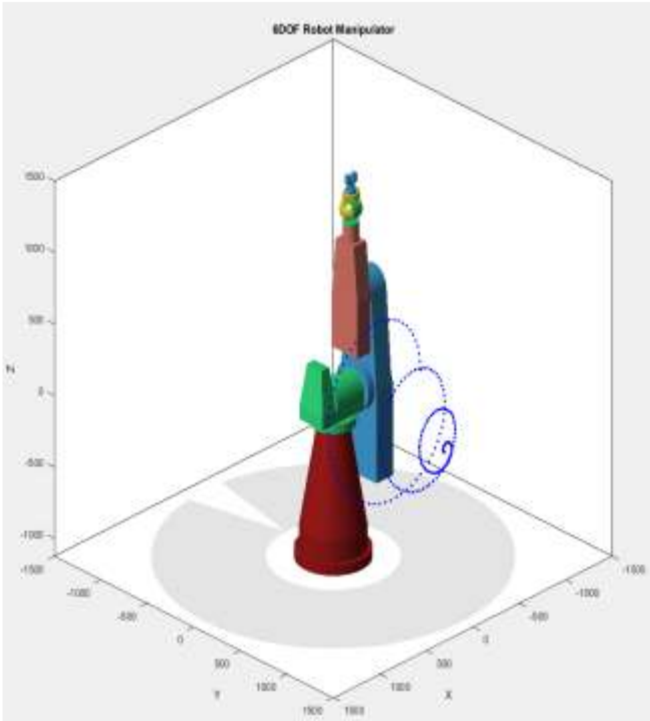

Figure-3: Drawing a Fibonacci Pattern

V. CONCLUSION

In this paper a six DOF robot mechanism has been simulated in a 3D space using MATLAB simulation tool. The motion of the robot through various angles and co-ordinates can be controlled in various directions along with various joint angle combinations. The forward and inverse kinematics concepts were also used to find the position end-effectors for various joint angles, and joint angles for the various end-effectors. By this development and through implementation of kinematics methodology is used in this modeling. It is clear that the inverse kinematics is broadly utilized in various practical applications of robotics and the result of forward kinematics also been developed for every joint angle by

ISSN (online): 2581-3048
Volume 2, Issue 4, pp 6-10, June-2018

combining differences in all configurations in position vector. The results were verified for the forward kinematics and inverse function in the inverse kinematics.

REFERENCES

[1] S.G. Anavatti, S.A. Salman and J.Y. Choi, "Fuzzy + PID Controller for Robot Manipulator," *International Conference on Computational Intelligence for Modelling Control and Automation.* pp. 75, Dec. 2006.

[2] Shiuh-Jer, H. and L. Ji-Shin."A stable self-organizing fuzzy controller for robotic motion control." *Industrial Electronics, IEEE Transactions on 47(2):* 421-428, 2000.

[3] Kiam Heong, A. "PID control system analysis, design, and technology." Control Systems Technology, *IEEE Transactions on 13(4):* 559-576, 2005.

[4] Visioli, "Tuning of PID Controllers with Fuzzy Logic," *IEEE Proceedings Control Theory and Applications,* Vol. 148, No. 1, pp. 1-8, 2001.

[5] Perry, T. S. "Lotfi A. Zadeh [fuzzy logic inventor biography]." *Spectrum, IEEE 32*(6): 32-35, 1995.

[6] E.H. Mamdani, "Applications of fuzzy logic to approximate reasoning using linguistic synthesis," *IEEE Transactions on Computers,* Vol. 26, No. 12, pp. 1182–1191, Dec. 1977.

[7] Z.-Y. Zhao, M. Tomizuka and S. Isaka, "Fuzzy Gain Scheduling PID Controller," *IEEE Transactions on Systems, Man., Cybernetics Society,* Vol. 23, No. 5, pp.1392–1398, Sep/Oct. 1993.

[8] R.P. Copeland and K.S. Rattan, "A Fuzzy Logic Supervisor for PID Control of Unknown Systems," *IEEE International Symposium on Intelligent Control.,* pp.22–26, Aug.1994.

[9] W. Li, "Design of a Hybrid Fuzzy Logic Proportional plus Conventional Integral Derivative Controller", *IEEE Transaction on Fuzzy Systems,* Vo1. 6, No. 4, pp.449–463, Nov 1998.

[10] SerdarKucuk and ZaferBingul (2006). Robot Kinematics: Forward and Inverse Kinematics, Industrial Robotics: *Theory, Modelling and Control, Sam Cubero (Ed.),* ISBN: 3-86611-285-8, InTech,

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

How to cite this article:

Thavamani.P, Ramesh.K, Sundari.B, "Simulation and Modeling of 6-DOF Robot Manipulator Using Matlab Software", in *International Research Journal of Innovations in Engineering and Technology (IRJIET),* Volume 2, Issue 4, pp 6-10, June 2018.

****

