A GUI Based Kinematic Model Development OF 6 DOF Manipulator Using Matlab

Sundara Mahalingam.V

Assistant Professor, Department Of Mechatronics Engineering, Skcet, Coimbatore, India

Abstract: This Design of a graphical user interface (GUI) is a fine task requiring knowledge of human cognitive behavior, design strategies and programming skills. In this paper, a GUI has been developed for position analysis of 6 DOF using MATALAB software. The kinematic problem is defined as a transformation from the Cartesian space to the joint space. In this paper the Denavit Hardenberg (D-H) model representation was used to model links and joints. Both forward & Inverse kinematic were discussed

Keywords - GUI, Denavit Hardenberg Representation, Forward & Inverse Kinematics.

I. INTRODUCTION

According to oxford English dictionary Robot is a machine capable of carrying out a complex series of actions automatically, especially one programmable by a computer. Robots are very powerful elements of today industry. They are capable of performing many different tasks and operations, are accurate, and do not require common safety and comfort elements humans need. The subject of robotics covers many different areas. Robots alone are hardly ever useful. They are used together with other devices, peripherals, and other manufacturing machines.



II. DENAVIT-HARTENBERG REPRESENTATION

In 1955, Denavit and Hartenberg published a paper in the ASME Journal of Applied Mechanics that was later used to represent and model robots and to derive their equations of motion. This technique has become standard way of representing robots and modeling their motion, and therefore, is essential to learn. The Denavit-Hartenberg (D-H) model of representation is a very simple way of modeling robot links and joints that can be used for any robot configuration, regardless of its sequence or complexity.

Robots may be made of a succession of joints and links in any order. The joints may be either prismatic (linear) or Revolute (Rotational), move in different planes, and have offsets. The links may also be of any length, including zero; may be twisted and bent; and may be in any plane. Therefore, any set of joints and links may create a robot



Fig.2. D-H successive joints & links

2.1 D-H Parameters

There are 2 Links & 2Joint parameters are used in D-H Representation.

2.1.1 Link Parameters

- i. Length of Linki (ai): It is mutual perpendicular distance between Axisi-1 and Axisi.
- ii. Angle of twist of Link_i (α_i): It is defined as the angle between Axis_{i-1} and Axis_i.

2.1.2 Joint Parameters

- **i.** Offset of Link_i (d_i): It is the distance measured from a point where a_{i-1} intersects the Axis_{i-1} to the point where a_i intersects the Axis_{i-1} measured along the said axis.
- **ii.** Joint Angle (Θ i): It is defined as the angle between the extension of a_{i-1} and ai measured about the Axis_{i-1}

2.1.3 The Final Transformation Matrix

The transformation between two successive frames is represented as below.

$${}^{i-1}_{i}T = {}^{i-1}_{A}T {}^{A}_{B}T {}^{B}_{C}T {}^{C}_{i}T$$

= Rot (Z, θ_i) Trans (Z, d_i) Rot (X, α_i) Trans (X, a_i)

(1)

Screwz Screwx

III. KINEMATIC MODEL OF 6 DOF ROBOT

For Kinematic analysis of taken 6 DOF serial link Manipulator, the D-H representation of Forward & Inverse Kinematics are mathematically Obtained first.





The joint & Link parameters of 6 DOF robots are noted below in a table 1.

#	θ	d	а	α
0-1	Θ_1	0	0	90
1-2	Θ_2	0	80	0
2-3	Θ_3	0	70	0
3-4	Θ_4	0	50	-90
4-5	Θ_5	0	0	90
5-6	Θ_6	0	0	0

3.1 Forward Kinematics

If all the robot joint variables are known, using forward kinematic equations, we can calculate where the robot is at any instant.

$$A_{1} = \begin{bmatrix} C_{1} & 0 & S_{1} & 0 \\ S_{1} & 0 & -C_{1} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} A_{2} = \begin{bmatrix} C_{2} & -S_{2} & 0 & C_{2}a_{2} \\ S_{2} & C_{2} & 0 & S_{2}a_{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$A_{3} = \begin{bmatrix} C_{3} & -S_{3} & 0 & C_{3}a_{3} \\ S_{3} & C_{3} & 0 & S_{3}a_{3} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} A_{4} = \begin{bmatrix} C_{4} & 0 & -S_{4} & C_{4}a_{4} \\ S_{4} & 0 & C_{4} & S_{4}a_{4} \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$A_{5} = \begin{bmatrix} C_{5} & 0 & S_{5} & 0 \\ S_{5} & 0 & -C_{5} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} A_{6} = \begin{bmatrix} C_{6} & -S_{6} & 0 & 0 \\ S_{6} & C_{6} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Therefore the total transformation between the base of the robot and the hand will be,

$$^{R}T_{H} = A_{1} A_{2} A_{3} A_{4} A_{5} A_{6}$$

By multiplying all "A" matrix, the final transformation matrix is obtained.

3.2 Inverse Kinematics

This means by placing the hand of the robot at a desired location & Orientation, the joint angle & Link length variables are easily obtained. For calculating the joint & Link parameters, the final forward kinematics equation is used here.

From Equation 2, $^{R}T_{H} = A_{1} A_{2} A_{3} A_{4} A_{5} A_{6}$ Therefore,

$${}_{H}^{R}T = \begin{bmatrix} n_{x} & o_{x} & a_{x} & P_{x} \\ n_{y} & o_{y} & a_{y} & P_{y} \\ n_{z} & o_{z} & a_{z} & P_{z} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(3)

(2)

IJEDR1804080 International Journal of Engineering Development and Research (<u>www.ijedr.org</u>)

The following joint angle values $\Theta_1 \& \Theta_3$ are obtained by multiplying the inverse of "A₁" matrix with whole transformation matrix.

$$A_{1}^{-1} \times \begin{bmatrix} n_{x} & o_{x} & a_{x} & P_{x} \\ n_{y} & o_{y} & a_{y} & P_{y} \\ n_{z} & o_{z} & a_{z} & P_{z} \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_{1}^{-1}[RHS] = A_{2}A_{3}A_{4}A_{5}A_{6}$$
⁽⁴⁾

By solving above equation, Angle 1 & 3 are obtained

$$\theta_1 = \tan^{-1} \left(\frac{P_y}{P_x} \right) \tag{5}$$

$$\theta_3 = tan^{-1}\frac{s_3}{c_3} \tag{6}$$

The following joint angles Θ_2 , Θ_4 , Θ_5 , Θ_6 are Obtained by multiplying the inverse of "A₁, A₂, A₃, A₄," with the whole transformation matrix.

$$A_{1}^{-1}A_{2}^{-1}A_{3}^{-1}A_{4}^{-1} \times \begin{bmatrix} n_{x} & o_{x} & a_{x} & P_{x} \\ n_{y} & o_{y} & a_{y} & P_{y} \\ n_{z} & o_{z} & a_{z} & P_{z} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= A_{1}^{-1}A_{2}^{-1}A_{3}^{-1}A_{4}^{-1}[RHS] = A_{5}A_{6}$$
(7)

By solving the above equation, Angle 2,4,5,6 are Obtained

 \sim

$$\theta_{2} = \tan^{-1} \frac{(C_{3}a_{3} + a_{2})(P_{z} - S_{234}a_{4}) - S_{3}a_{3}(P_{x}C_{1} + P_{y}S_{1} - C_{234}a_{4})}{(C_{3}a_{3} + a_{2})(P_{x}C_{1} + P_{y} - C_{234}a_{4}) + S_{3}a_{3}(P_{z} - S_{234}a_{4})}$$
(8)

$$\theta_4 = \theta_{234} - \theta_2 - \theta_3 \tag{9}$$

$$\theta_5 = tan^{-1} \frac{c_{234}(c_1a_x + s_1a_y) + s_{234}a_z}{s_1a_x - c_1a_y}$$
(10)

$$\theta_6 = tan^{-1} \frac{-S_{234}(C_1 n_x + S_1 n_y) + (C_{234} n_z)}{-S_{234}(C_1 o_x + S_1 o_y) + C_{234} o_z}$$
(11)

IV. GUI DEVELOPMENT FOR 6 DOF ROBOT POSITION ANALYSIS

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.





4.1 Creation of GUI for 6 DOF manipulator:

Fig.5.kinematic analysis GUI - APP for 6 DOF serial link robot

437

4.2 Forward Kinematics Matlab Code

Th_1 = str2double(handles.Theta_1.String)*pi/180; Th_2 = str2double(handles.Theta_2.String)*pi/180; Th_3 = str2double(handles.Theta_3.String)*pi/180; $Th_4 = str2double(handles.Theta_4.String)*pi/180;$ $Th_5 = str2double(handles.Theta_5.String)*pi/180;$ $Th_6 = str2double(handles.Theta_6.String)*pi/180;$ $L_1 = 20;$ $L_2 = 50;$ L 3 = 40; $L_4 = 50;$ $L_5 = 40;$ L 6 = 20; $L(1) = Link([0 L_1 0 pi/2]);$ $L(2) = Link([0 \ 0 \ L_2 \ 0]);$ $L(3) = Link([0 \ 0 \ L_3 \ 0]);$ $L(4) = Link([0 \ 0 \ L_4 \ -pi/2]);$ $L(5) = Link([0 \ 0 \ L_5 \ pi/2]);$ $L(6) = Link([0 L_6 0 0]);$ Robot = SerialLink(L); Robot.name = 'RRR_Robot'; Robot.plot([Th_1 Th_2 Th_3 Th_4 Th_5 Th_6]); $T = Robot.fkine([Th_1 Th_2 Th_3 Th_4 Th_5 Th_6]);$ handles.Pos_X.String = num2str(floor(T.t(1))); handles.Pos_Y.String = num2str(floor(T.t(2))); handles.Pos_Z.String = num2str(floor(T.t(3))); 4.3 Inverse Kinematics Matlab Code: PX = str2double(handles.Pos_X.String); PY = str2double(handles.Pos_Y.String); PZ = str2double(handles.Pos_Z.String); $L_1 = 0;$ L 2 = 80; $L_3 = 70;$ $L_4 = 50;$ L 5 = 0; $L_6 = 0;$ $L(1) = Link([0 L_1 0 pi/2]);$ $L(2) = Link([0 \ 0 \ L_2 \ 0]);$

 $L(3) = Link([0 \ 0 \ L_3 \ 0]);$

 $L(4) = Link([0 0 L_4 -pi/2]);$ $L(5) = Link([0 0 L_5 pi/2]);$ $L(6) = Link([0 L_6 0 0]);$ Robot = SerialLink(L); $Robot.name = 'RRR_Robot';$ T = [1 0 0 PX; 0 1 0 PY; 0 0 1 PZ; 0 0 0 1]; J = Robot.ikine(T,[0 0 0],'mask',[1 1 1 0 0 0])*180/pi;handles.Theta_1.String = num2str(floor(J(1))); handles.Theta_2.String = num2str(floor(J(2))); handles.Theta_3.String = num2str(floor(J(3))); Robot.plot(J*pi/180);

V. SIMULATION RESULTS



Fig.6.Forward Kinematics Results a.) Home Position b.) Positioning at various joint values



Fig.7.Inverse Kinematics Results

VI. CONCLUSION:

Determined Mathematical equations, which describe the kinematic of Taken 6 DOF serial Manipulator & the Graphical user interface of Matlab Peter corke tool box allow to control & simulate the Manipulator. The future scope of the work is by directly interface with the serial manipulator hardware, leads to create a bridge between hardware & software for easy control of robotic system

VII. ACKNOWLEDGMENT

I express my deep sense of gratitude to my Mentor Dr.P.Balamurugan Professor and Head Department of Mechatronics Engineering and our Principal Dr.J.Janet for inspiring guidance and encouragement, which has enabled me to do this work.

REFERENCES

- [1] Claudio Urrea, Juan Cortes, Jose pascal, 2016, Design, Construction.
- [2] Jian Fang, Wei Lei, 2013, four degrees of freedom SCARA robot Kinematics Modeling & Simulation analysis.
- [3] Karan Desai, 2012, Development of a graphical user interface for control of robotic manipulator with sample acquisition capability.
- [4] Malaka Miyuranga Kaluarachi and Fawaz Yahya Anaz, 2012, GUI Teaching Pendent Development for a 6 AXIS Articulated Robot.
- [5] Norbert Piotrowski, Adam Barylski, 2014, modeling a 6-DOF Manipulator using Matlab Software.
- [6] Peter Corke, 2011, Robotics Vision and control Fundamental Algorithms in Matlab.

