KINEMATICS ANALYSIS AND MODELING IN MATLAB OF THE 6 DOF ROBOTIC ARM

Lecturer PhD, Gheorghe GÎLCĂ, “Constantin Brâncuși” University from Tîrgu-Jiu, ROMANIA, gheorghe.gilca@yahoo.com

ABSTRACT: The aim of this study is to analyze the robot arm kinematics which is very important for the movement of all robotic joints. Also they are very important to obtain the indication for controlling or moving of the robot arm in the workspace. In this study the forwards and inverse kinematics of 6 degree of freedom robot arm will be accomplished by using Matlab. Finding the parameters of Denavit-Hartenberg representation, the kinematic equations of motion can be derived which solve the problems of automatic control of the 6 revolute joints manipulator.

KEYWORDS: GUI Matlab, Denavit-Hardenb erg representation, forward & inverse kinematics

1. INTRODUCTION

The kinematics problem is related to finding the transformation from the Cartesian space to the joint space and vice versa. The solutions of the kinematics problem of any robot manipulator have two types; the forward kinematic and inverse kinematics. When all joints are known the forward kinematic will determine the Cartesian space, or where the manipulator arm will be. In the inverse kinematic the calculations of all joints is done if the desired position and orientation of the end-effectors is determined, that means by the inverse kinematic the robotic arm joint space angles will be calculated as referred to [1]. The kinematic analysis of industrial robots was discussed in many literatures [1], [2]. Koyuncu and Guzel [3] suggested a method for solving the kinematics of the Lynx 6d of Robot and propose a software package named MSG that used to test the behavior of robot motion. Qassem et al. [4] proposed a software package to solve the kinematics of the AL5B Robot arm. More analysis have been achieved for modeling a 6 dof robotic manipulators using the MATLAB software for their simulation by Iqbal et al. [5], Kumar et al. [6] and Singh et al. [7].

2. DENAVIT-HARTENBERG REPRESENTATION

A commonly used convention for selecting frames of reference in robotics applications is the Denavit and Hartenberg (D–H) convention which was introduced by Jacques Denavit and Richard S. Hartenberg [8],[9]. In this convention, coordinate frames are attached to the joints between two links such that one transformation is associated with the joint, [Z], and the second is associated with the link [X]. The coordinate transformations along a serial robot consisting of n links form the kinematics equations of the robot,

\[ [T]=[Z_1][X_1][Z_2][X_2].........[X_{n-1}][Z_n][X_n] \]  

, where \([T]\) is the transformation locating the end-link.

In order to determine the coordinate transformations \([Z]\) and \([X]\), the joints connecting the links are modeled as either hinged or sliding joints, each of which have a unique line \(S\) in space that forms the joint axis and define the relative movement of the two links. A typical
serial robot is characterized by a sequence of six lines \( S_i \), \( i = 1, \ldots, 6 \), one for each joint in the robot. For each sequence of lines \( S_i \) and \( S_{i+1} \), there is a common normal line \( A_{i,i+1} \). The system of six joint axes \( S_i \) and five common normal lines \( A_{i,i+1} \) form the kinematic skeleton of the typical six degree of freedom serial robot. Denavit and Hartenberg introduced the convention that Z coordinate axes are assigned to the joint axes \( S_i \) and X coordinate axes are assigned to the common normals \( A_{i,i+1} \). This convention allows the definition of the movement of links around a common joint axis \( S_i \) by the screw displacement,

\[
\begin{bmatrix}
\cos \theta_i & -\sin \theta_i & 0 & 0 \\
\sin \theta_i & \cos \theta_i & 0 & 0 \\
0 & 0 & 1 & d_i \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

where \( \theta_i \) is the rotation around and \( d_i \) is the slide along the Z axis—either of the parameters can be constants depending on the structure of the robot. Under this convention the dimensions of each link in the serial chain are defined by the screw displacement around the common normal \( A_{i,i+1} \) from the joint \( S_i \) to \( S_{i+1} \), which is given by

\[
\begin{bmatrix}
1 & 0 & 0 & r_{i,i+1} \\
0 & \cos \alpha_{i,i+1} & -\sin \alpha_{i,i+1} & 0 \\
0 & \sin \alpha_{i,i+1} & \cos \alpha_{i,i+1} & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

where \( \alpha_{i,i+1} \) and \( r_{i,i+1} \) define the physical dimensions of the link in terms of the angle measured around and distance measured along the X axis.

The reference frames are laid out as follows:
1. the z-axis is in the direction of the joint axis
2. the x-axis is parallel to the common normal: \( x_n = z_n \times z_{n-1} \) (or away from \( z_{n-1} \))
3. the y-axis follows from the x- and z-axis by choosing it to be a right-handed coordinate system.

The following four transformation parameters are known as D–H parameters [10]:
- d – offset along previous z to the common normal
- \( \theta_i \) - angle about previous z, from old x to new x
• $a_i$-length of the common normal ($\alpha$, but if using this notation, do not confuse with $\alpha$). Assuming a revolute joint, this is the radius about previous $z$.
• $\alpha$ - angle about common normal, from old $z$ axis to new $z$ axis

3. THE KINEMATIC MODEL OF A ROBOTIC ARM WITH 6 DOF

For Kinematic analysis of taken 6 DOF serial link manipulator, the D-H representation of Forward & Inverse Kinematics are mathematically obtained first. In figure 2 is shown a simple 6 DOF robotic arm.

![6 DOF robot manipulator](image)

Figure 2. 6 DOF robot manipulator

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

<table>
<thead>
<tr>
<th>#</th>
<th>$\theta$</th>
<th>d</th>
<th>a</th>
<th>$\alpha$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1</td>
<td>$\theta_1$</td>
<td>0</td>
<td>0</td>
<td>90</td>
</tr>
<tr>
<td>1-2</td>
<td>$\theta_2$</td>
<td>0</td>
<td>80</td>
<td>0</td>
</tr>
<tr>
<td>2-3</td>
<td>$\theta_3$</td>
<td>0</td>
<td>70</td>
<td>0</td>
</tr>
<tr>
<td>3-4</td>
<td>$\theta_4$</td>
<td>0</td>
<td>50</td>
<td>-90</td>
</tr>
<tr>
<td>4-5</td>
<td>$\theta_5$</td>
<td>0</td>
<td>0</td>
<td>90</td>
</tr>
<tr>
<td>5-6</td>
<td>$\theta_6$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1. D-H Parameter for taken 6 DOF robot

The joint variables of the robot are given to determine the position and orientation effector. Each joint for each frame has a single degree of freedom and can be represented by a single number, which is the angle of rotation in the case of a revolute joint i.e., ($\theta_1$, $\theta_2$,...,..., $\theta_n$).

3.1 Forward kinematics

Forward kinematics analysis is the process of calculating the position and orientation of the end-angles so by substituting these parameters in the homogenous transformation matrix from joint $i$ to joint $i+1$ [1]:

$$A_i = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha & s\theta_i s\alpha & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha & -c\theta_i s\alpha & a_i s\theta_i \\ 0 & s\alpha & c\alpha & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ (4)

The transformation matrices joints can be obtained as shown:
where, the matrix $A_1$ for example shows the transformation between frames 0 and 1, $C = \cos \theta$ and $S = \sin \theta$.

\[
A_1^1 = A_1 = \begin{bmatrix}
C_1 & -S_1 & 0 & 0 \\
S_1 & C_1 & 0 & 0 \\
0 & 0 & 1 & d_1 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

Performing the composition from the $n$-th frame to the base frame we multiply the six matrices from 1 to 6:

\[
A_6^n = A_1^0 * A_2^1 * A_3^2 * A_4^3 * A_5^4 * A_6^5
\]

where, $p_x$, $p_y$, $p_z$ represent the position and $\{(n_1, n_2, n_3), (o_1, o_2, o_3), (a_1, a_2, a_3)\}$, represent the orientation of the end-effector, they can be calculated in terms of joint angles.
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 11, the following joint angle values $\theta_1$ & $\theta_3$ are obtained by multiplying the inverse of “$A_1$” matrix with whole transformation matrix:

$$A_1^{-1} \times \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_1^{-1}[\text{RHS}] = A_2A_3A_4A_5A_6$$

By solving above equation, angle 1 & 3 are obtained:

$$\theta_1 = \tan^{-1} \left( \frac{P_y}{P_x} \right) \quad \theta_3 = \tan^{-1} \frac{S_3}{C_3}$$

The following joint angles $\theta_2$, $\theta_4$, $\theta_5$, $\theta_6$, are obtained by multiplying the inverse of “$A_1$, $A_2$, $A_3$, $A_4$,” with the whole transformation matrix:

$$A_1^{-1}A_2^{-1}A_3^{-1}A_4^{-1} \times \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_1^{-1}A_2^{-1}A_3^{-1}A_4^{-1}[\text{RHS}] = A_5A_6$$

By solving the above equation, angle 2,4,5,6 are obtained:

$$\theta_2 = \tan^{-1} \frac{(C_3a_3 + a_2)(P_x - S_{234}a_4) - S_3a_3(P_xC_1 + P_yS_1 - C_{234}a_4)}{(C_3a_3 + a_2)(P_xC_1 + P_y - C_{234}a_4) + S_3a_3(P_x - S_{234}a_4)}$$

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

$$\theta_5 = \tan^{-1} \frac{C_{234}(C_1a_x+S_1a_y)+S_{234}a_z}{S_1a_x-C_1a_y}$$

$$\theta_6 = \tan^{-1} \frac{-S_{234}(C_1n_x+S_1n_y)+(C_{234}n_z)}{-S_{234}(C_1a_x+S_1a_y)+C_{234}a_z}$$

4. GUI CREATION 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.

4.1 Creation of GUI for 6 DOF manipulator

Figure 3 shows the interface creator in matlab, here the interface components are introduced, and figure 4 shows the interface created to solve the kinematic and inverse model of the robotic arm with 6DOF.
Figure 3. Kinematic analysis GUI layout creator

Figure 4. Kinematic analysis GUI – APP for 6 DOF serial link robot

4.2 Forward Kinematics Matlab Code

\[
\begin{align*}
\text{Th}_1 &= \text{str2double}(	ext{handles.\text{Theta}_1.String}) \times \pi / 180; \\
\text{Th}_2 &= \text{str2double}(	ext{handles.\text{Theta}_2.String}) \times \pi / 180; \\
\text{Th}_3 &= \text{str2double}(	ext{handles.\text{Theta}_3.String}) \times \pi / 180; \\
\text{Th}_4 &= \text{str2double}(	ext{handles.\text{Theta}_4.String}) \times \pi / 180; \\
\text{Th}_5 &= \text{str2double}(	ext{handles.\text{Theta}_5.String}) \times \pi / 180; \\
\text{Th}_6 &= \text{str2double}(	ext{handles.\text{Theta}_6.String}) \times \pi / 180; \\
L_1 &= 20; \\
L_2 &= 50;
\end{align*}
\]
4.3 Inverse Kinematics Matlab Code:

```matlab
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';
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)));
```

```matlab
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);
```
5. CONCLUSIONS
Matlab/Simulink is especially useful for generating the approximate solutions of mathematical models that may be prohibitively difficult to solve "by hand." 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.

REFERENCES