KINEMATIC MODELING OF 6 DOF ROBOTIC ARM WITH MATLAB

Lecturer PhD, Gheorghe GÎLCĂ, "Constantin Brâncuşi" 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 fredom 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: Denavit-Hardenberg representation, forward & inverse kinematics

1. Introduction

The problem of kinematic modeling is usually categorized into two sub-problems. First is the forward or direct kinematics, which is the problem of solving the Cartesian position and orientation of a mechanism, given the knowledge of the kinematic structure and the joint coordinates. The second sub-problem is Inverse Kinematics (IK), which computes the joint variables using the given information of a robot's end-effector position and orientation. In case of serial robotic arms, IK problem is more complex than direct kinematic problem [1].

Scientific community reports various robot modeling and analysis techniques. Clothier et al. [2] proposed a geometric model for a solve the unknown joint angles required for positioning a mobile robot. Popovic et al. [3] developed a upper arm motion analysis strategy, while the full body kinematics of a radially symmetric six-legged robot was presented by Wang et al. [4]. A mathematical approach to analyze the kinematics of a humanoid robot was reported in [5]. Cubero proposed an IK model for a solve for all the common variables of a robotic arm manipulator [6]. This model was based on the direct kinematic solution.

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 [7],[8]. 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,

$$[T]=[Z_1][X_1][Z_2][X_2]\ldots\ldots\ldots\ldots [X_{n-1}][Z_n][X_n] \quad (1)$$

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, $i = 1,...,6$, 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,

$$[Z_i] = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

where: θ_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:

$$[X_i] = \begin{bmatrix} 1 & 0 & 0 & r_{i,i+1} \\ 0 & \cos\alpha_{i,i+1} & -\sin\alpha_{i,i+1} & 0 \\ 0 & \sin\alpha_{i,i+1} & \cos\alpha_{i,i+1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

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.

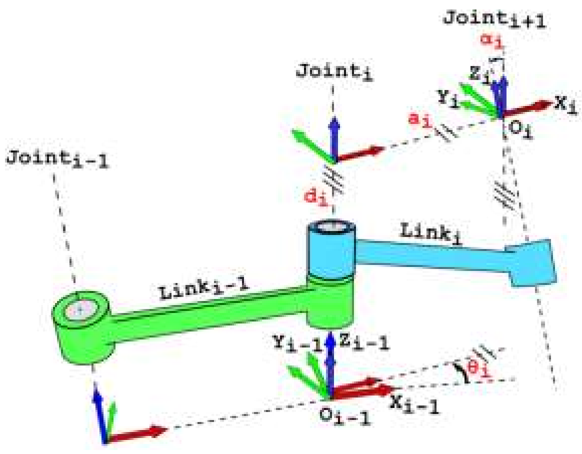

Figure 1. D-H successive joints & links

The following four transformation parameters are known as D–H parameters [9]:

- d – offset along previous z to the common normal
- θ - angle about previous z, from old x to new x

• a_i-length of the common normal (, but if using this notation, do not confuse with α). Assuming a revolute joint, this is the radius about previous z.

• α - angle about common normal, from old z axis to new z axis

3. The kinematic model of a 6 dof robotic arm

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.

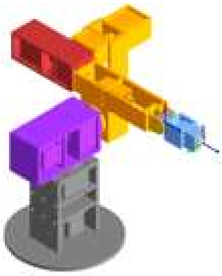

Figure 2. 6 DOF robotic arm manipulator

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

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

	Θ	**d**	**a**	**α**
-1	θ_1	0.305	0	90
-2	θ_2	0	0.22	0
-3	θ_3	0	0	90
-4	θ_4	0.352	0	90
-5	θ_5	0	0	-90
-6	θ_6	0.1315	0	0

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., (θ_1, θ_2,......, θ_{n-1},θ_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:

$$Ai = \begin{bmatrix} C\theta i & -S\theta iC\alpha i & S\theta iS\alpha i & aiC\theta i \\ S\theta i & C\theta iC\alpha i & -C\theta iS\alpha i & aiS\theta i \\ 0 & S\alpha i & C\alpha i & di \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

The transformation matrices joints can be obtained as shown:

$$A_1^0 = A_1 = \begin{bmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

where, the matrix A_1 for example shows the transformation between frames 0 and 1, C = cosθ and S = sinθ.

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

$$A_6^0 = A_1^0 * A_2^1 * A_3^2 * A_4^3 * A_5^4 * A_6^5 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

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 6, the following joint angle values θ1 & θ3 are obtained by multiplying the inverse of "A_1" matrix with whole transformation matrix:

$$A_1^{-1} \times \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_1^{-1}[RHS] = A_2A_3A_4A_5A_6 \quad (7)$$

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

$$\theta_1 = tan^{-1}\left(\frac{P_y}{P_x}\right) \qquad \theta_3 = tan^{-1}\frac{S_3}{C_3} \quad (8)$$

The following joint angles θ_2, θ_4, θ_5, θ_6, are obtained by multiplying the inverse of "A_1, A_2, A_3, A_4," with the whole transformation matrix:

$$A_1^{-1}A_2^{-1}A_3^{-1}A_4^{-1} \times \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_1^{-1}A_2^{-1}A_3^{-1}A_4^{-1}[RHS] = A_5A_6 \quad (9)$$

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

$$\theta_2 = tan^{-1}\frac{(C_3a_3 + a_2)(P_z - S_{234}a_4) - S_3a_3(P_xC_1 + P_yS_1 - C_{234}a_4)}{(C_3a_3 + a_2)(P_xC_1 + P_y - C_{234}a_4) + S_3a_3(P_z - S_{234}a_4)} \quad (10)$$

$$\theta_4 = \theta_{234} - \theta_2 - \theta_3 \quad (11)$$

$$\theta_5 = tan^{-1}\frac{C_{234}(C_1a_x + S_1a_y) + S_{234}a_z}{S_1a_x - C_1a_y} \quad (12)$$

$$\theta_6 = tan^{-1}\frac{-S_{234}(C_1n_x + S_1n_y) + (C_{234}n_z)}{-S_{234}(C_1o_x + S_1o_y) + C_{234}O_z} \quad (13)$$

4. Build a 6 dof robotic arm in matlab

The DH parameters define the geometry of the robot with relation to how each rigid body is attached to its parent. For convenience, setup the parameters for the robotic arm in a table 1. The robotic arm is a serial chain manipulator. The DH parameters are relative to the

previous row in the matrix, corresponding to the previous joint attachment.
To create a model of the robotic arm in Matlab, we must go through the following steps [10]:

a) we create the rigid structure of the tree-type robot (from the base to the top)
b) we create the first rigid body and add it to the robot. To add a rigid body we will execute as follows:
   - we create a rigidBody object and give it a unique name.
   - we create a rigidBodyJoint object and give it a unique name.
   - we use setFixedTransform to specify body-to-body transformation using DH parameters. The last element of the DH parameters, theta, is ignored because the angle is dependent on the joint position.
   - we call addBody to attach the first body joint to the base frame of the robot.

With the following lines of code, the 3D positioning of the joints of the robotic arm with 6 dof is created in matlab:

```
robot = rigidBodyTree;
body1 = rigidBody('body1');
jnt1 = rigidBodyJoint('jnt1','revolute');
setFixedTransform(jnt1,dhparams(1,:),'dh');
body1.Joint = jnt1;
addBody(robot,body1,'base')
body2 = rigidBody('body2');
jnt2 = rigidBodyJoint('jnt2','revolute');
body3 = rigidBody('body3');
jnt3 = rigidBodyJoint('jnt3','revolute');
body4 = rigidBody('body4');
jnt4 = rigidBodyJoint('jnt4','revolute');
body5 = rigidBody('body5');
jnt5 = rigidBodyJoint('jnt5','revolute');
body6 = rigidBody('body6');
jnt6 = rigidBodyJoint('jnt6','revolute');
setFixedTransform(jnt2,dhparams(2,:),'dh');
setFixedTransform(jnt3,dhparams(3,:),'dh');
setFixedTransform(jnt4,dhparams(4,:),'dh');
setFixedTransform(jnt5,dhparams(5,:),'dh');
setFixedTransform(jnt6,dhparams(6,:),'dh');
body2.Joint = jnt2;
body3.Joint = jnt3;
body4.Joint = jnt4;
body5.Joint = jnt5;
body6.Joint = jnt6;
addBody(robot,body2,'body1')
addBody(robot,body3,'body2')
addBody(robot,body4,'body3')
addBody(robot,body5,'body4')
addBody(robot,body6,'body5')
show(robot);
axis([-0.5,0.5,-0.5,0.5,-0.5,0.5])
axis off
```

Figure 3 shows the position of the joints and the 3D simulation of the robotic arm.

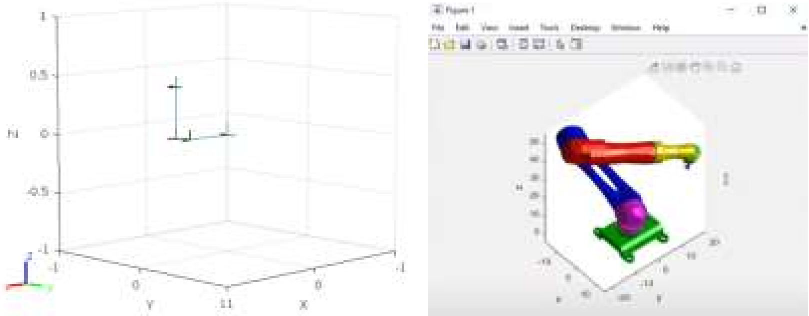

Figure 3. a) The position of the joints b) The simulation of the robotic arm in matlab

4. 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

[1]. **Mark S., Seth H. and Vidyasagar M.,** *Robot modeling and control,* John Wiley & Sons, 2006.

[2]. **Clothier K.E. and Shang Y,** *A geometric approach for robotic arm kinematics with hardware design, electrical design and implementation*, Journal of Robotics, Article ID 984823, Vol. 2010.

[3]. **Popovic N., Williams S, Schmitz-Rode T., Rau G., Disselhorst-Klug C.,** *Robot-based methodology for a kinematic and kinetic analysis of unconstrained, but reproducible upper extremity movement*, Journal of Biomechanics, Vol. 42 No. 10, 2009

[4]. **Wang Z., Xilun D., Alberto R. and Alessandro G.,** *Mobility analysis of the typical gait of a radial symmetrical six-legged robot***,** Mechatronics, Vol.21, No.7, pp1133-1146, 2011

[5]. **Man C., Fan X., Li C. and Zhao Z.,** *Kinematics analysis based on screw theory of a humanoid robot***,** Journal of China University of Mining & Technology, Vol.17 No.1, pp. 49–52, 2007.

[6]. **Cubero S**., *An inverse kinematics method for controlling all types of serial-link robot arms*, Mechatronics and Machine Vision in Practice, Springer-Verlag, Berlin, pp.217-232.

[7]. **T. Kanda, T. Hirano, D. Eaton, and H. Ishiguro**, *Interactive robots as social partners and peer tutors for children: a field trial*, Human–Computer Interaction,19(1), 2004, 61–84.

[8]. **J. Han and D. Kim**, *R-Learning services for elementary school students with a teaching assistant robot*, Proc. HRI, 2009, 255–256.

[9]. **S.Y. Okita, V. Ng-Thow-Hing, and R. Sarvadevabhatla,** *Learning together: asimo developing an interactive learning partnership with children*, Proc. ROMAN, 2009, 1125–1130.

[10]. https://www.mathworks.com/