1st report of Advanced Topics in Robot Mechanism

Student GU JUN
Student ID 61322300564
Date Nov 18. 2024

A manipulator with 4 DOF

The manipulator consists of 4 degrees of freedom (DOF) with 3 revolute joints and 1 prismatic joint. The manipulator is shown in Fig.1. For this manipulator, the base frame is denoted as R, and the end-effector frame is denoted as E. The joints are labeled as J1, J2, J3, and J4. When the joint variables are set to specific values, the manipulator can be configured in different positions and orientations in 3D space.

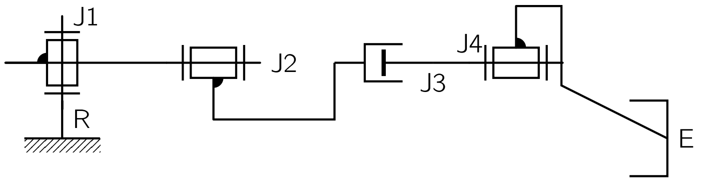

Figure 1: Manipulator with 4 DOF, 3 revolute joints and 1 prismatic joint

DH Notation for Structure Definition

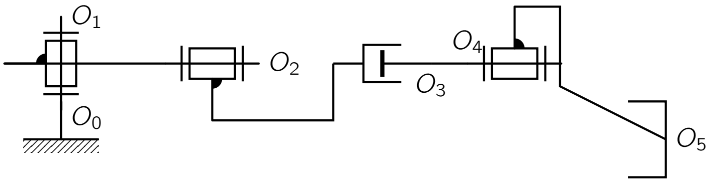

Figure 2: Define the coordinates of the manipulator

After define the coordinates of the manipulator, we can use the DH parameters to define the structure of the manipulator. All lengths and angles are given with a set value. The DH parameters for the manipulator are shown in Table.1.

Joint	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	1	θ_1
2	1	$\frac{\pi}{2}$	0	θ_2
3	1	0	d_3	0
4	0	0	0	θ_4
5	0	$-\pi$	0.1	0

Table 1: DH Parameters for a 3 DOF Robot Arm

Kinematic Equation Derivation

The transformation matrix for each joint using DH parameters is given by:

$$
{}^{n-1}T_n = \left[\begin{array}{ccc|c}
\cos\theta_n & -\sin\theta_n & 0 & a_{n-1} \\
\sin\theta_n\cos\alpha_{n-1} & \cos\theta_n\cos\alpha_{n-1} & -\sin\alpha_{n-1} & -d_n\sin\alpha_{n-1} \\
\sin\theta_n\sin\alpha_{n-1} & \cos\theta_n\sin\alpha_{n-1} & \cos\alpha_{n-1} & d_n\cos\alpha_{n-1} \\
\hline
0 & 0 & 0 & 1
\end{array}\right] \tag{1}
$$

For the given manipulator, the overall transformation matrix from the base to the end-effector is the product of the individual transformation matrices:

$$T = T_1 \cdot T_2 \cdot T_3 \cdot T_4$$

Where each T_i is calculated using the corresponding DH parameters.

Given the DH parameters, we can calculate the transformation matrices for each joint:

$$T_1 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_2 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 1 \\ 0 & 0 & -1 & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_4 = \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & 0 \\ \sin(\theta_4) & \cos(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_5 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The overall transformation matrix T is:

$$T = T_1 \cdot T_2 \cdot T_3 \cdot T_4 \cdot T_5$$

Calculating the product:

$$T = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 1 \\ 0 & 0 & -1 & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\cdot \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & 0 \\ \sin(\theta_4) & \cos(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} c_1 c_2 c_4 - c_1 s_2 s_4 & c_1 c_2 s_4 + c_1 c_4 s_2 & -s_1 & c_1 - \frac{s_1}{10} + c_1 c_2 + d_3 s_1 \\ c_2 c_4 s_1 - s_1 s_2 s_4 & c_2 s_1 s_4 + c_4 s_1 s_2 & c_1 & \frac{c_1}{10} + s_1 + c_2 s_1 - d_3 c_1 \\ c_2 s_4 + c_4 s_2 & s_2 s_4 - c_2 c_4 & 0 & s_2 + 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The above matrix T represents the overall transformation matrix from the base to the end-effector of the manipulator. Here, c_i and s_i are shorthand notations for $\cos(\theta_i)$ and $\sin(\theta_i)$ respectively. This matrix is derived by multiplying the individual transformation matrices T_1, T_2, T_3, T_4, and T_5 using the given DH parameters.

With this matrix, we can calculate the endpoint position and posture of the manipulator for different joint variables.

Joint Variables and Endpoint Positions Calculation

To verify the correctness of the derived kinematic equation, we calculate the endpoint positions for different joint variables. The following table shows the joint variables and their corresponding endpoint positions:

θ_1	θ_2	d_3	θ_4
0	0	0.5	0
$\frac{\pi}{4}$	$\frac{\pi}{4}$	1	$\frac{\pi}{4}$
$\frac{\pi}{2}$	$\frac{\pi}{2}$	1.5	0

Table 2: Joint Variables for testing

Using these joint variables, we calculate the transformation matrix T and extract the endpoint position from the matrix.

For the joint variables $\theta_1 = 0$, $\theta_2 = 0$, $d_3 = 0.5$, and $\theta_4 = 0$, the transformation matrix T is:

$$T = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & -0.4 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For the joint variables $\theta_1 = \frac{\pi}{4}$, $\theta_2 = \frac{\pi}{4}$, $d_3 = 1$, and $\theta_4 = \frac{\pi}{4}$, the transformation matrix T is:

$$T = \begin{bmatrix} 0 & \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & \frac{19\sqrt{2}}{20} + \frac{1}{2} \\ 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{20} + \frac{1}{2} \\ 1 & 0 & 0 & \frac{\sqrt{2}}{2} + 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For the joint variables $\theta_1 = \frac{\pi}{2}$, $\theta_2 = \frac{\pi}{2}$, $d_3 = 1.5$, and $\theta_4 = 0$, the transformation matrix T is:

$$T = \begin{bmatrix} 0 & 0 & -1 & \frac{7}{5} \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The last column of each matrix represents the endpoint position of the manipulator for the given joint variables. The calculated endpoint positions are shown in the following table(after rounding off to 4 decimal places):

θ_1	θ_2	d_3	θ_4	Endpoint Position (x, y, z)
0	0	0.5	0	$(2, -0.4, 1)$
$\frac{\pi}{4}$	$\frac{\pi}{4}$	1	$\frac{\pi}{4}$	$(1.8435, 0.5707, 1.7071)$
$\frac{\pi}{2}$	$\frac{\pi}{2}$	1.5	0	$(1.4, 1, 2)$

Table 3: Joint Variables and Corresponding Endpoint Positions

Matlab simulation & Verification of the manipulator

I use Matlab rigid body tree to prove my result. The main process is define the rigid body tree, set the joint variables, and get the endpoint position. Then compare the result with the calculated result to verify the correctness of the kinematic equation.

Modeling Using DH Parameters

To verify the calculated results, I modeled the manipulator using the DH parameters in Matlab. The manipulator is modeled as a rigid body tree with 5 joints and 5 links. The DH parameters are used to define the structure of the manipulator. One thing to note is that there are 'dh' and 'mdh' options in the rigid body tree. The 'dh' option uses the Denavit-Hartenberg convention for the joint transformations, while the 'mdh' option uses the modified Denavit-Hartenberg convention. In our case, we need to use the 'mdh' option to define the manipulator structure. With the DH parameters from table.1, we can define the rigid body tree for the manipulator as follows:

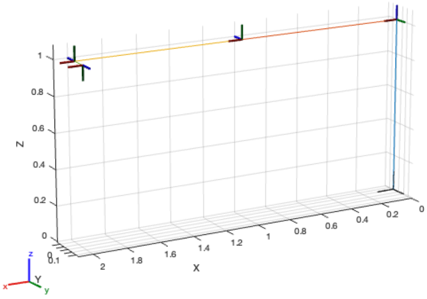

Figure 3: The 4 DOF Robot Arm Model, all joint variables are set to zero

With the joint variables as table.2, we can get the endpoint position of the manipulator. The endpoint positions are calculated using the forward kinematics function of the rigid body tree. And the result is showing in the following figures.

Matlab Result

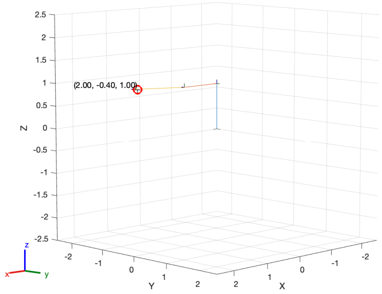

Figure 4: Manipulator configuration for $\theta_1 = 0,\ \theta_2 = 0,\ d_3 = 0.5,\ \theta_4 = 0$

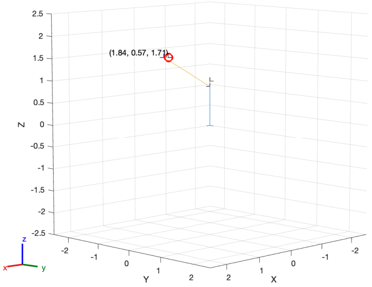

Figure 5: Manipulator configuration for $\theta_1 = \frac{\pi}{4},\ \theta_2 = \frac{\pi}{4},\ d_3 = 1,\ \theta_4 = \frac{\pi}{4}$

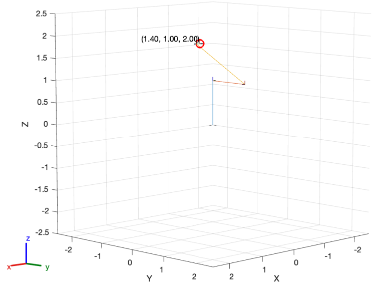

Figure 6: Manipulator configuration for $\theta_1 = \frac{\pi}{2},\ \theta_2 = \frac{\pi}{2},\ d_3 = 1.5,\ \theta_4 = 0$

θ_1	θ_2	d_3	θ_4	Endpoint Position (x, y, z) calculated	Endpoint position in Matlab
0	0	0.5	0	$(2, -0.4, 1)$	$2.0000, -0.4000, 1.0000$
$\frac{\pi}{4}$	$\frac{\pi}{4}$	1	$\frac{\pi}{4}$	$(1.8435, 0.5707, 1.7071)$	$(1.8435, 0.5707, 1.7071)$
$\frac{\pi}{2}$	$\frac{\pi}{2}$	1.5	0	$(1.4, 1, 2)$	$(1.4000, 1.0000, 2.0000)$

Table 4: Comparison of Calculated and Matlab Endpoint Positions

The endpoint positions calculated using the kinematic equation and the endpoint positions obtained from the Matlab simulation are consistent.

Appendix

Matlab Code

```
% Define DH parameters: [a alpha d theta]
theta1 = 0;
theta2 = 0;
theta3 = 0;
dhparams = [0    0    1    0;
  1     pi/2       0    0;
  1     0     -0.2 0;
  0    0    0    0;
  0       -pi  0.1    0;];
% Create a rigid body tree
robot = robotics.RigidBodyTree;
bodies = cell(5,1);
joints = cell(5,1);
for i = 1:5
  bodies{i} = rigidBody([’body’ num2str(i)]);
  if i == 3
    joints{i} = rigidBodyJoint([’jnt’ num2str(i)],"prismatic");
  else
    joints{i} = rigidBodyJoint([’jnt’ num2str(i)],"revolute");
  end
  setFixedTransform(joints{i},dhparams(i,:),"mdh");
  bodies{i}.Joint = joints{i};
  if i == 1 % Add first body to base
    addBody(robot,bodies{i},"base")
  else % Add current body to previous body by name
    addBody(robot,bodies{i},bodies{i-1}.Name)
  end
end
% Define configurations based on the table
configurations = [0, 0, 0.5, 0;
                  pi/4, pi/4, 1, pi/4;
                  pi/2, pi/2, 1.5, 0];
% Loop through each configuration and display the robot
for i = 1:size(configurations, 1)
  config = homeConfiguration(robot);
  config(1).JointPosition = configurations(i, 1);
  config(2).JointPosition = configurations(i, 2);
  config(3).JointPosition = configurations(i, 3);
  config(4).JointPosition = configurations(i, 4);
  % Get the transformation from the base to the end effector
  tform = getTransform(robot, config, ’body5’);
  % Extract the position from the transformation matrix
  position = tform(1:3, 4);
  % Show the robot with the updated configuration
  show(robot, config, ’PreservePlot’, false, ’Visuals’, ’on’);
  % Plot the position of the end effector
  % Display the position of the end effector
  disp([’End effector position for configuration ’ num2str(i) ’:’]);
  disp(position);
  title([’Configuration ’ num2str(i)]);
  drawnow;
  pause(1); % Pause to visualize each configuration
end
```