How to Operate the Code

Setup

Of course, ensure MATLAB is installed on your system.

1. Running the Script:
   Open MATLAB
   Navigate to the directory containing the script
   Press the "Run" button or type the script name in the command window

2. Functionality:
   The script performs forward kinematics
   Visualizes the robot in 3D space
   Calculates inverse kinematics

3. Outputs:
   DH transformation matrix
   3D visualization of the robot
   Joint angles from inverse kinematics

4. Modifying Parameters:
   Adjust link lengths (l1 to l6) for different robot configurations
   Modify theta, d, a, and alpha arrays to change the DH parameters

5. Troubleshooting:
   Ensure all required functions are in the same directory
   Check for any error messages in the MATLAB command window

Documentation of Major Functions

Forward Kinematics

The forward kinematics function uses the Denavit-Hartenberg (DH) convention to calculate the end-effector position and orientation.

Mathematical Methods used in this project:

- DH Transformation Matrix

Tabulation of Transformation Matrices from the D-H table

$$ {}^{i-1}_{i}T = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_{i-1} \\ \cos(\alpha_{i-1})\sin(\theta_i) & \cos(\alpha_{i-1})\cos(\theta_i) & -\sin(\alpha_{i-1}) & -d_i\sin(\alpha_{i-1}) \\ \sin(\alpha_{i-1})\sin(\theta_i) & \sin(\alpha_{i-1})\cos(\theta_i) & \cos(\alpha_{i-1}) & d_i\cos(\alpha_{i-1}) \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ \alpha_0 = 0, \qquad a_0 = 0, \qquad d_1 = 0 $$

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

- Chain Multiplication

The final transformation is computed by multiplying individual matrices

T06 = T0_1· T1_2· T2_3· T3_4· T4_5· T5_6

<u>Inverse Kinematics</u>

The inverse kinematics function calculates joint angles for a desired end-effector position and orientation.

Using the Geometric Approach:

Joint 1: q1=\atan2(py,px)q1 =\atan2(py ,px)
Joint 2: q2=\atan2(pz−l1,r−l2)+offset2q2 =\atan2(pz −l1 ,r−l2)+offset2
Joint
3: q3=\atan2(l3−pz,l2−r)+offset3q3 =\atan2(l3 −pz ,l2 −r)+offset3

In order to calculate the wrist angles:

Joint 4: q4=\atan2(R2,1,R1,1)q4 =\atan2(R2,1,R1,1)
Joint 5: q5=\atan2(R3,12+R3,22,R3,3)q5 =\atan2(R3,12 +R3,22 ,R3,3)
Joint 6: q6=\atan2(R3,2,−R3,1)q6 =\atan2(R3,2 ,−R3,1)

Where R is the rotation matrix from the transformation matrix T06.

Visualization

The visualization function plots the robot's links and joint frames in 3D space using MATLAB's plotting capabilities.

Notable Matlab Syntax:

- plot3() for drawing links
- quiver3() for representing joint frames
- Color coding for easy identification of links and axes