Momentum observer for collision detection of UR10e 6DOF Robot

Tran Van Minh Hieu

February 2026

1 Configuration of UR10e robot

These configuration are taken from the follwing document Datasheet UR10e

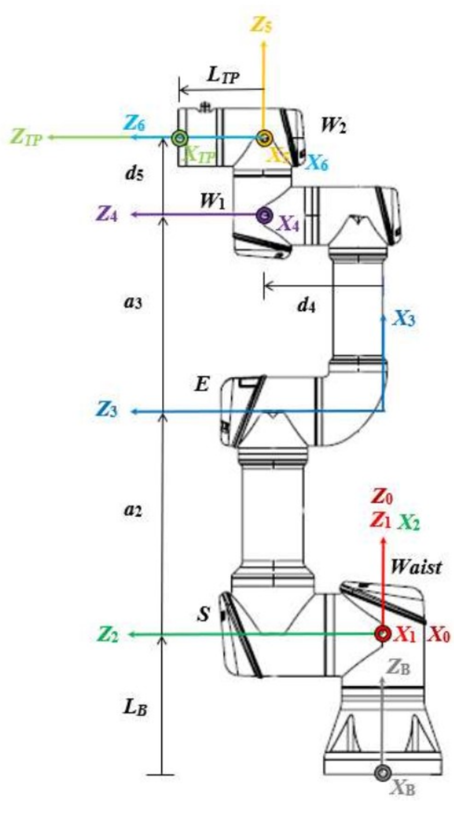

Figure 1: UR10e robot coordinate configuration

Table 1: General Modified DH Parameters for UR10e

Joint (i)	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	d_1	q_1
2	$\pi/2$	0	0	q_2
3	0	a_2	0	q_3
4	0	a_3	d_4	q_4
5	$-\pi/2$	0	d_5	q_5
6	$\pi/2$	0	d_6	q_6

Table 2: Geometric Parameter Values of UR10e

Parameter	**Value (m)**
d_1	0.181
a_2	-0.6127
a_3	-0.5716
d_4	0.1639
d_5	0.1157
d_6	0.0922

Table 3: Dynamic Parameters of UR10e Robot Links in custom local coordinate in DH table 1

Link	**Mass (kg)**	**Center of Mass (m)** $[x, y, z]$
Link 1	7.37	$[0.021, 0.0, 0.027]$
Link 2	13.05	$[0.38, 0.0, 0.158]$
Link 3	3.99	$[0.24, 0.0, 0.068]$
Link 4	2.10	$[0.0, 0.007, 0.018]$
Link 5	1.98	$[0.0, 0.007, 0.018]$
Link 6	0.62	$[0.0, 0.0, -0.026]$

Table 4: Geometric Parameters of UR10e Links

Link	**Length** L **(m)**	**Radius** r **(m)**
Link 1 (Shoulder)	0.181 (d_1)	0.075
Link 2 (Upper Arm)	0.613 (a_2)	0.054
Link 3 (Forearm)	0.572 (a_3)	0.049
Link 4 (Wrist 1)	0.174 (d_4)	0.045
Link 5 (Wrist 2)	0.120 (d_5)	0.045
Link 6 (Wrist 3)	0.117	0.045

2 Kinematics calculation

The general homogeneous transformation matrix based on Modified DH convention in 1 is defined as:

$$
{}^{i-1}_{i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -d_i s\alpha_{i-1} \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}
$$

2.1 Link 1 Analysis

The transformation matrix from the base frame to frame 1 is:

$$
{}^{0}_{1}T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}
$$

2.1.1 Jacobian translational matrix

The Center of Mass (COM) of Link 1 in its local coordinate system is defined as:

$$
{}^{1}P_{COM1} = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} \tag{3}
$$

The global position of the COM for Link 1 relative to the Base Frame (Frame 0) is:

$$
{}^{0}P_{COM1} = {}^{0}_{1}T \cdot {}^{1}P_{COM1} \tag{4}
$$

Performing the matrix multiplication:

$$
{}^{0}P_{COM1} = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} = \begin{bmatrix} x_1 c_1 - y_1 s_1 \\ x_1 s_1 + y_1 c_1 \\ z_1 + d_1 \\ 1 \end{bmatrix} \tag{5}
$$

The Jacobian translation matrix J_{v1} for the COM of Link 1 is the partial derivative of the position vector with respect to q:

$$
\begin{aligned}
\frac{\partial}{\partial q_1}(x_1 c_1 - y_1 s_1) &= -x_1 s_1 - y_1 c_1 \\
\frac{\partial}{\partial q_1}(x_1 s_1 + y_1 c_1) &= x_1 c_1 - y_1 s_1 \\
\frac{\partial}{\partial q_1}(z_1 + d_1) &= 0
\end{aligned} \tag{6}
$$

Thus, the 3×6 Translation Jacobian matrix for Link 1 is:

$$J_{v1} = \begin{bmatrix} -x_1 s_1 - y_1 c_1 & 0 & 0 & 0 & 0 & 0 \\ x_1 c_1 - y_1 s_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{7}$$

2.1.2 Jacobian rotational matrix

The rotational part of the transformation matrix ${}^0_1T$ is:

$${}^0_1R = \begin{bmatrix} c_1 & -s_1 & 0 \\ s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

The skew-symmetric matrix $\hat{\omega}_1$ is calculated as:

$$\hat{\omega}_1 = {}^0\dot{R}_1 {}^0_1R^T = \dot{q}_1 \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{9}$$

The rotational Jacobian $J_{\omega 1}$ is extracted as:

$$J_{\omega 1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{10}$$

2.2 Link 2 Analysis

The transformation matrix from frame 1 to frame 2 is:

$${}^1_2T = \begin{bmatrix} -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

The transformation matrix from the base frame to frame 2 is:

$${}^0_2T = {}^0_1T . {}^1_2T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -c_1 s_2 & -c_1 c_2 & s_1 & 0 \\ -s_1 s_2 & -s_1 c_2 & -c_1 & 0 \\ c_2 & -s_2 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

2.2.1 Jacobian translational matrix

The Center of Mass (COM) of Link 2 in its local coordinate system is defined as:

$${}^2P_{COM2} = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{bmatrix} \tag{13}$$

The global position of the COM for Link 2 relative to the Base Frame (Frame 0) is:

$$
{}^{0}P_{COM2} = {}^{0}_{2}T \cdot {}^{2}P_{COM2} = \begin{bmatrix} -x_2c_1s_2 - y_2c_1c_2 + z_2s_1 \\ -x_2s_1s_2 - y_2s_1c_2 - z_2c_1 \\ x_2c_2 - y_2s_2 + d_1 \\ 1 \end{bmatrix} \tag{14}
$$

The Jacobian translation matrix J_{v2} involves partial derivatives with respect to q_1 and q_2 (columns 3 to 6 are zero).

The Translational Jacobian J_{v2} is obtained by taking partial derivatives of the position vector ${}^{0}P_{COM2}$ with respect to the active joints q_1 and q_2 (columns 3 to 6 are zero).

1. Column 1 (Derivatives w.r.t q_1):

$$
\begin{aligned}
J_{v2}^{(1x)} &= x_2s_1s_2 + y_2s_1c_2 + z_2c_1 \\
J_{v2}^{(1y)} &= -x_2c_1s_2 - y_2c_1c_2 + z_2s_1 \\
J_{v2}^{(1z)} &= 0
\end{aligned} \tag{15}
$$

2. Column 2 (Derivatives w.r.t q_2):

$$
\begin{aligned}
J_{v2}^{(2x)} &= -c_1(x_2c_2 - y_2s_2) \\
J_{v2}^{(2y)} &= -s_1(x_2c_2 - y_2s_2) \\
J_{v2}^{(2z)} &= -x_2s_2 - y_2c_2
\end{aligned} \tag{16}
$$

Thus, the 3×6 Translation Jacobian matrix for Link 2 is:

$$
J_{v2} = \begin{bmatrix} (x_2s_1s_2 + y_2s_1c_2 + z_2c_1) & -c_1(x_2c_2 - y_2s_2) & 0 & 0 & 0 & 0 \\ (-x_2c_1s_2 - y_2c_1c_2 + z_2s_1) & -s_1(x_2c_2 - y_2s_2) & 0 & 0 & 0 & 0 \\ 0 & (-x_2s_2 - y_2c_2) & 0 & 0 & 0 & 0 \end{bmatrix} \tag{17}
$$

2.2.2 Jacobian rotational matrix

The rotational part of the transformation matrix ${}^{0}_{2}T$ is extracted as:

$$
{}^{0}_{2}R = \begin{bmatrix} -c_1s_2 & -c_1c_2 & s_1 \\ -s_1s_2 & -s_1c_2 & -c_1 \\ c_2 & -s_2 & 0 \end{bmatrix} \tag{18}
$$

To derive the rotational Jacobian, we utilize the relationship between the time derivative of the rotation matrix and the angular velocity vector. The skew-symmetric matrix $\hat{\omega}_2$ is defined as:

$$
\hat{\omega}_2 = {}^{0}_{2}\dot{R}\,{}^{0}_{2}R^{T} \tag{19}
$$

Expanding equation (19) and grouping terms by $\dot{q}_1$ and $\dot{q}_2$, we identify the columns of the Jacobian matrix corresponding to the rotation axes:

- The coefficient vector for $\dot{q}_1$ (Column 1) corresponds to the base rotation:

$$J_{\omega 2}^{(1)} = [0, 0, 1]^T$$

- The coefficient vector for $\dot{q}_2$ (Column 2) corresponds to the shoulder rotation axis transformed to the base frame:

$$J_{\omega 2}^{(2)} = [s_1, -c_1, 0]^T$$

Thus, the resulting 3×6 Rotational Jacobian matrix for Link 2 is:

$$J_{\omega 2} = \begin{bmatrix} 0 & s_1 & 0 & 0 & 0 & 0 \\ 0 & -c_1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{20}$$

2.3 Link 3 Analysis

The transformation matrix from Frame 2 to Frame 3 is:

$$^2_3T = \begin{bmatrix} c_3 & -s_3 & 0 & a_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{21}$$

The transformation matrix from the Base Frame to Frame 3 is computed as:

$$^0_3T = {^0_2T} \cdot {^2_3T} = \begin{bmatrix} -c_1 s_{23} & -c_1 c_{23} & s_1 & -a_2 c_1 s_2 \\ -s_1 s_{23} & -s_1 c_{23} & -c_1 & -a_2 s_1 s_2 \\ c_{23} & -s_{23} & 0 & d_1 + a_2 c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{22}$$

Note:

- $c_{ij} = \cos(q_i + q_j)$ and $s_{ij} = \sin(q_i + q_j)$.

2.3.1 Jacobian Translational Matrix (Link 3)

The Center of Mass (COM) of Link 3 in its local coordinate system is:

$$^3P_{COM3} = \begin{bmatrix} x_3 \\ y_3 \\ z_3 \\ 1 \end{bmatrix} \tag{23}$$

The global position $^0P_{COM3}$ is derived by multiplying $^0_3T \cdot {^3P_{COM3}}$. To explicitly show the structure, we group the terms projecting onto the horizontal plane into a term L_{proj}:

$$L_{proj} = x_3 s_{23} + y_3 c_{23} + a_2 s_2 \tag{24}$$

The global coordinates of the COM are then:

$$ {}^{0}P_{COM3} = \begin{bmatrix} -c_1 L_{proj} + z_3 s_1 \\ -s_1 L_{proj} - z_3 c_1 \\ x_3 c_{23} - y_3 s_{23} + a_2 c_2 + d_1 \\ 1 \end{bmatrix} \tag{25} $$

The Translational Jacobian J_{v3} is obtained by taking partial derivatives of the position vector with respect to q_1, q_2, q_3.

1. Column 1 (Derivatives w.r.t q_1):

$$ \begin{aligned} J_{v3}^{(1x)} &= s_1 L_{proj} + z_3 c_1 \\ J_{v3}^{(1y)} &= -c_1 L_{proj} + z_3 s_1 \\ J_{v3}^{(1z)} &= 0 \end{aligned} \tag{26} $$

2. Column 2 (Derivatives w.r.t q_2):

$$ \begin{aligned} J_{v3}^{(2x)} &= -c_1(x_3 c_{23} - y_3 s_{23} + a_2 c_2) \\ J_{v3}^{(2y)} &= -s_1(x_3 c_{23} - y_3 s_{23} + a_2 c_2) \\ J_{v3}^{(2z)} &= -x_3 s_{23} - y_3 c_{23} - a_2 s_2 \end{aligned} \tag{27} $$

3. Column 3 (Derivatives w.r.t q_3):

$$ \begin{aligned} J_{v3}^{(3x)} &= -c_1(x_3 c_{23} - y_3 s_{23}) \\ J_{v3}^{(3y)} &= -s_1(x_3 c_{23} - y_3 s_{23}) \\ J_{v3}^{(3z)} &= -x_3 s_{23} - y_3 c_{23} \end{aligned} \tag{28} $$

Thus, the Jacobian matrix J_{v3} is:

$$ J_{v3} = \begin{bmatrix} J_{v3}^{(1x)} & J_{v3}^{(2x)} & J_{v3}^{(3x)} & 0 & 0 & 0 \\ J_{v3}^{(1y)} & J_{v3}^{(2y)} & J_{v3}^{(3y)} & 0 & 0 & 0 \\ 0 & J_{v3}^{(2z)} & J_{v3}^{(3z)} & 0 & 0 & 0 \end{bmatrix} \tag{29} $$

2.3.2 Jacobian Rotational Matrix (Link 3)

The rotational part of the transformation matrix ${}^{0}_{3}T$ is extracted as:

$$ {}^{0}_{3}R = \begin{bmatrix} -c_1 s_{23} & -c_1 c_{23} & s_1 \\ -s_1 s_{23} & -s_1 c_{23} & -c_1 \\ c_{23} & -s_{23} & 0 \end{bmatrix} \tag{30} $$

Using the relationship $\hat{\omega}_3 = {}^{0}\dot{R}_3 {}^{0}{R_3}^{T}$, we identify the rotation axes corresponding to each joint:

- **Joint 1 (Base):** Rotates around the Z-axis of Frame 0.

$$J_{\omega 3}^{(1)} = [0, 0, 1]^T$$

- **Joint 2 (Shoulder):** Rotates around the axis determined by Link 1 configuration (same as in Link 2 analysis).

$$J_{\omega 3}^{(2)} = [s_1, -c_1, 0]^T$$

- **Joint 3 (Elbow):** Since the axes of Joint 2 and Joint 3 are parallel in the UR10e structure, their rotation vectors in the base frame are identical.

$$J_{\omega 3}^{(3)} = [s_1, -c_1, 0]^T$$

Constructing the Jacobian by stacking these vectors:

$$J_{\omega 3} = \begin{bmatrix} 0 & s_1 & s_1 & 0 & 0 & 0 \\ 0 & -c_1 & -c_1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{31}$$

2.4 Link 4 Analysis

$${}^{3}_{4}T = \begin{bmatrix} s_4 & c_4 & 0 & a_3 \\ -c_4 & s_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{32}$$

The transformation matrix from the Base Frame to Frame 4 is computed as:

$${}^{0}_{4}T = {}^{0}_{3}T \cdot {}^{3}_{4}T = \begin{bmatrix} c_1 c_{234} & -c_1 s_{234} & s_1 & d_4 s_1 - c_1(a_2 s_2 + a_3 s_{23}) \\ s_1 c_{234} & -s_1 s_{234} & -c_1 & -d_4 c_1 - s_1(a_2 s_2 + a_3 s_{23}) \\ s_{234} & c_{234} & 0 & d_1 + a_2 c_2 + a_3 c_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{33}$$

Note:

- $c_{234} = \cos(q_2 + q_3 + q_4)$, $s_{234} = \sin(q_2 + q_3 + q_4)$.
- $s_{23} = \sin(q_2 + q_3)$, $c_{23} = \cos(q_2 + q_3)$.

2.4.1 Jacobian Translational Matrix (Link 4)

The Center of Mass (COM) of Link 4 in its local coordinate system is defined as:

$${}^{4}P_{COM4} = \begin{bmatrix} x_4 \\ y_4 \\ z_4 \\ 1 \end{bmatrix} \tag{34}$$

The global position ${}^0P_{COM4}$ is derived by multiplying the transformation matrix ${}^0_4T$ with the local COM vector. To simplify the derivatives, a projection term L_{proj4} is defined:

$$L_{proj4} = -x_4c_{234} + y_4s_{234} + a_2s_2 + a_3s_{23} \tag{35}$$

Additionally, the effective vertical offset from the link is defined as:

$$H_{eff4} = x_4s_{234} + y_4c_{234} + a_2c_2 + a_3c_{23} \tag{36}$$

The global coordinates of the COM are then expressed as:

$${}^0P_{COM4} = \begin{bmatrix} -c_1L_{proj4} + (z_4 + d_4)s_1 \\ -s_1L_{proj4} - (z_4 + d_4)c_1 \\ H_{eff4} + d_1 \\ 1 \end{bmatrix} \tag{37}$$

The Translational Jacobian J_{v4} is obtained by taking partial derivatives of the position vector with respect to q_1, q_2, q_3, q_4 (columns 5 and 6 are zero).

1. Column 1 (Derivatives w.r.t q_1):

$$\begin{aligned} J_{v4}^{(1x)} &= s_1L_{proj4} + (z_4 + d_4)c_1 \\ J_{v4}^{(1y)} &= -c_1L_{proj4} + (z_4 + d_4)s_1 \\ J_{v4}^{(1z)} &= 0 \end{aligned} \tag{38}$$

2. Column 2 (Derivatives w.r.t q_2): The derivative of the horizontal projection is $\frac{\partial L_{proj4}}{\partial q_2} = H_{eff4}$. The derivative of the vertical height is $\frac{\partial H_{eff4}}{\partial q_2} = -L_{proj4}$.

$$\begin{aligned} J_{v4}^{(2x)} &= -c_1(H_{eff4}) \\ J_{v4}^{(2y)} &= -s_1(H_{eff4}) \\ J_{v4}^{(2z)} &= -L_{proj4} \end{aligned} \tag{39}$$

3. Column 3 (Derivatives w.r.t q_3): Terms involving a_2 are constant with respect to q_3. Let $H_{34} = x_4s_{234} + y_4c_{234} + a_3c_{23}$ and $L_{34} = -x_4c_{234} + y_4s_{234} + a_3s_{23}$.

$$\begin{aligned} J_{v4}^{(3x)} &= -c_1(H_{34}) \\ J_{v4}^{(3y)} &= -s_1(H_{34}) \\ J_{v4}^{(3z)} &= -L_{34} \end{aligned} \tag{40}$$

4. Column 4 (Derivatives w.r.t q_4): Only terms involving q_4 (attached to x_4, y_4) are differentiated.

$$\begin{aligned} J_{v4}^{(4x)} &= -c_1(x_4s_{234} + y_4c_{234}) \\ J_{v4}^{(4y)} &= -s_1(x_4s_{234} + y_4c_{234}) \\ J_{v4}^{(4z)} &= -(-x_4c_{234} + y_4s_{234}) = x_4c_{234} - y_4s_{234} \end{aligned} \tag{41}$$

Thus, the Jacobian matrix J_{v4} is constructed as:

$$J_{v4} = \begin{bmatrix} J_{v4}^{(1x)} & J_{v4}^{(2x)} & J_{v4}^{(3x)} & J_{v4}^{(4x)} & 0 & 0 \\ J_{v4}^{(1y)} & J_{v4}^{(2y)} & J_{v4}^{(3y)} & J_{v4}^{(4y)} & 0 & 0 \\ 0 & J_{v4}^{(2z)} & J_{v4}^{(3z)} & J_{v4}^{(4z)} & 0 & 0 \end{bmatrix} \tag{42}$$

2.4.2 Jacobian Rotational Matrix (Link 4)

The rotational part of the transformation matrix ${}^0_4T$ is extracted as:

$${}^0_4R = \begin{bmatrix} c_1c_{234} & -c_1s_{234} & s_1 \\ s_1c_{234} & -s_1s_{234} & -c_1 \\ s_{234} & c_{234} & 0 \end{bmatrix} \tag{43}$$

Using the relationship $\hat{\omega}_4 = {}^0\dot{R}_4 {}^0R_4{}^T$, we identify the rotation axes corresponding to each joint. Since Link 4 motion depends on joints q_1, q_2, q_3, q_4: After calculation, construct the Jacobian by stacking these vectors:

$$J_{\omega 4} = \begin{bmatrix} 0 & s_1 & s_1 & s_1 & 0 & 0 \\ 0 & -c_1 & -c_1 & -c_1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{44}$$

2.5 Link 5 Analysis

The transformation matrix from Frame 4 to Frame 5 is:

$${}^4_5T = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & -d_5 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{45}$$

The transformation matrix from the Base Frame to Frame 5 is computed as:

$${}^0_5T = {}^0_4T \cdot {}^4_5T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{46}$$

Where the orientation components are:

$$r_{11} = c_1c_{234}c_5 - s_1s_5 \qquad r_{12} = -c_1c_{234}s_5 - s_1c_5 \qquad r_{13} = -c_1s_{234}$$
$$r_{21} = s_1c_{234}c_5 + c_1s_5 \qquad r_{22} = -s_1c_{234}s_5 + c_1c_5 \qquad r_{23} = -s_1s_{234}$$
$$r_{31} = s_{234}c_5 \qquad r_{32} = -s_{234}s_5 \qquad r_{33} = c_{234}$$

And the position components:

$$p_x = d_4s_1 - c_1(a_2s_2 + a_3s_{23} + d_5s_{234})$$
$$p_y = -d_4c_1 - s_1(a_2s_2 + a_3s_{23} + d_5s_{234})$$
$$p_z = d_1 + a_2c_2 + a_3c_{23} + d_5c_{234}$$

2.5.1 Jacobian Translational Matrix (Link 5)

The global position of the Center of Mass (COM) for Link 5, denoted as ${}^{0}P_{COM5}$, depends on the configuration of the first five joints. It is derived from the transformation chain:

$$ {}^{0}P_{COM5} = {}^{0}_{5}T(q_1, q_2, q_3, q_4, q_5) \cdot {}^{5}P_{COM5} \tag{47} $$

Due to the complexity of the symbolic expressions resulting from the accumulation of multiple rotation matrices, the Translational Jacobian $J_{v5} \in \mathbb{R}^{3\times 6}$ is computed analytically by taking the partial derivatives of the position vector ${}^{0}P_{COM5}$ with respect to each joint variable.

The structure of the Jacobian matrix is as follows:

$$ J_{v5} = \frac{\partial^{0} P_{COM5}}{\partial q} = \begin{bmatrix} \frac{\partial P_x}{\partial q_1} & \frac{\partial P_x}{\partial q_2} & \frac{\partial P_x}{\partial q_3} & \frac{\partial P_x}{\partial q_4} & \frac{\partial P_x}{\partial q_5} & 0 \\ \frac{\partial P_y}{\partial q_1} & \frac{\partial P_y}{\partial q_2} & \frac{\partial P_y}{\partial q_3} & \frac{\partial P_y}{\partial q_4} & \frac{\partial P_y}{\partial q_5} & 0 \\ \frac{\partial P_z}{\partial q_1} & \frac{\partial P_z}{\partial q_2} & \frac{\partial P_z}{\partial q_3} & \frac{\partial P_z}{\partial q_4} & \frac{\partial P_z}{\partial q_5} & 0 \end{bmatrix} \tag{48} $$

Remarks:

- Columns 1 through 5 contain non-zero terms reflecting the contribution of joints q_1 to q_5 to the linear velocity of Link 5.
- Column 6 is a zero vector since the motion of Joint 6 does not affect the position of the preceding link (Link 5).
- The explicit symbolic terms are omitted here for brevity but are fully calculated in the simulation software (MATLAB).

2.5.2 Jacobian Rotational Matrix (Link 5)

The rotation matrix ${}^{0}_{5}R$ derived from ${}^{0}_{5}T$ is:

$$ {}^{0}_{5}R = \begin{bmatrix} c_1c_{234}c_5 - s_1s_5 & -c_1c_{234}s_5 - s_1c_5 & -c_1s_{234} \\ s_1c_{234}c_5 + c_1s_5 & -s_1c_{234}s_5 + c_1c_5 & -s_1s_{234} \\ s_{234}c_5 & -s_{234}s_5 & c_{234} \end{bmatrix} \tag{49} $$

Using the relationship $\hat{\omega}_5 = {}^{0}_{5}\dot{R}{}^{0}_{5}R^{T}$, we identify the rotation axes corresponding to each joint. Since Link 5 motion depends on joints q_1, q_2, q_3, q_4, q_5: After calculation, extract the skew elements in the $\hat{\omega}_5$ variable and constructs the rotation Jacobian:

$$ J_{\omega 5} = \begin{bmatrix} 0 & s_1 & s_1 & s_1 & -s_{234}c_1 & 0 \\ 0 & -c_1 & -c_1 & -c_1 & -c_{234}s_1 & 0 \\ 1 & 0 & 0 & 0 & c_{234} & 0 \end{bmatrix} \tag{50} $$

2.6 Link 6 Analysis (End-Effector)

The transformation matrix from Frame 5 to Frame 6 is:

$$
{}^{5}_{6}T = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{51}
$$

The final Forward Kinematics matrix, representing the pose of the End-Effector relative to the Base Frame, is computed by the chain multiplication:

$$
{}^{0}_{6}T = {}^{0}_{5}T \cdot {}^{5}_{6}T \tag{52}
$$

2.6.1 Jacobian Translational Matrix (Link 6)

The linear velocity of the End-Effector depends on all six joints (q_1 to q_6).

$$
{}^{0}P_{COM6} = {}^{0}_{6}T \cdot {}^{6}P_{COM6} \tag{53}
$$

The Translational Jacobian J_{v6} is expressed analytically as the gradient of the COM position vector of link 6:

$$
J_{v6} = \begin{bmatrix} \frac{\partial p_x}{\partial q_1} & \frac{\partial p_x}{\partial q_2} & \cdots & \frac{\partial p_x}{\partial q_6} \\ \frac{\partial p_y}{\partial q_1} & \frac{\partial p_y}{\partial q_2} & \cdots & \frac{\partial p_y}{\partial q_6} \\ \frac{\partial p_z}{\partial q_1} & \frac{\partial p_z}{\partial q_2} & \cdots & \frac{\partial p_z}{\partial q_6} \end{bmatrix} \tag{54}
$$

2.6.2 Jacobian Rotational Matrix (Link 6)

Using the relationship $\hat{\omega}_6 = {}^{0}_{6}\dot{R}{}^{0}_{6}R^T$, rotation axes corresponding to each joint are identified . Since Link 6 motion depends on all joints $q_1, q_2, q_3, q_4, q_5, q_6$: After calculation, extract the skew elements in the $\hat{\omega}_6$ variable and constructs the rotation Jacobian:

Thus, the final Rotational Jacobian is:

$$
J_{\omega 6} = \begin{bmatrix} 0 & s_1 & s_1 & s_1 & -s_{234}c_1 & c_5 s_1 + c_{234}c_1 s_5 \\ 0 & -c_1 & -c_1 & -c_1 & -c_{234}s_1 & c_{234}s_1 s_5 - c_1 c_5 \\ 1 & 0 & 0 & 0 & c_{234} & s_{234}s_5 \end{bmatrix} \tag{55}
$$

3 Dynamics Calculation

3.1 Moment of Inertia

$$I_1 = \begin{bmatrix} \frac{1}{12}m_1(3r_1^2 + d_1^2) & 0 & 0 \\ 0 & \frac{1}{12}m_1(3r_1^2 + d_1^2) & 0 \\ 0 & 0 & \frac{1}{2}m_1 r_1^2 \end{bmatrix} \tag{56}$$

$$I_2 = \begin{bmatrix} \frac{1}{2}m_2 r_2^2 & 0 & 0 \\ 0 & \frac{1}{12}m_2(3r_2^2 + a_2^2) & 0 \\ 0 & 0 & \frac{1}{12}m_2(3r_2^2 + a_2^2) \end{bmatrix} \tag{57}$$

$$I_3 = \begin{bmatrix} \frac{1}{2}m_2 r_2^2 & 0 & 0 \\ 0 & \frac{1}{12}m_2(3r_2^2 + a_2^2) & 0 \\ 0 & 0 & \frac{1}{12}m_2(3r_2^2 + a_2^2) \end{bmatrix} \tag{58}$$

$$I_4 = \begin{bmatrix} \frac{1}{12}m_4(3r_4^2 + d_4^2) & 0 & 0 \\ 0 & \frac{1}{12}m_4(3r_4^2 + d_4^2) & 0 \\ 0 & 0 & \frac{1}{2}m_4 r_4^2 \end{bmatrix} \tag{59}$$

$$I_5 = \begin{bmatrix} \frac{1}{12}m_5(3r_5^2 + d_5^2) & 0 & 0 \\ 0 & \frac{1}{12}m_5(3r_5^2 + d_5^2) & 0 \\ 0 & 0 & \frac{1}{2}m_5 r_5^2 \end{bmatrix} \tag{60}$$

$$I_6 = \begin{bmatrix} \frac{1}{12}m_6(3r_6^2 + L_6^2) & 0 & 0 \\ 0 & \frac{1}{12}m_6(3r_6^2 + L_6^2) & 0 \\ 0 & 0 & \frac{1}{2}m_6 r_6^2 \end{bmatrix} \tag{61}$$

Where:

- m_i, r_i: Mass and radius of Link i.
- d_1, a_2, a_3, d_4, d_5: Denavit-Hartenberg length parameters derived from geometry.
- L_6: Thickness of the final wrist link (Wrist 3).

3.2 Lagrangian Dynamics Formulation

To derive the equations of motion for the UR10e robot, Euler-Lagrange method is applied. This energy-based approach describes the system's dynamics by analyzing the difference between the total kinetic energy and the total potential energy of the mechanism. The Lagrangian function $\mathcal{L}$ is defined as the difference between the total kinetic energy $\mathcal{K}$ and the total potential energy $\mathcal{P}$ of the system:

$$\mathcal{L}(q, \dot{q}) = \mathcal{K}(q, \dot{q}) - \mathcal{P}(q) \tag{62}$$

Where:

- $q \in \mathbb{R}^n$ is the vector of joint coordinates (generalized coordinates).
- $\dot{q} \in \mathbb{R}^n$ is the vector of joint velocities.
- The kinetic energy $\mathcal{K}$ is a quadratic function of the joint velocity vector $\dot{q}$.
- The potential energy $\mathcal{P}$ is a function of joint position q only and is independent of $\dot{q}$.

The general Euler-Lagrange equation for the i-th joint is given by:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathcal{L}}{\partial q_i} = \tau_i \tag{63}$$

where τ_i is the generalized force/torque applied at joint i.

Using the property that the kinetic energy can be expressed as $\mathcal{K} = \frac{1}{2}\dot{q}^T M(q)\dot{q}$, where $M(q)$ is the symmetric, positive-definite inertia matrix, the terms required for Lagrange's Equation (63) are derived as follows.

First, the partial derivative of the Lagrangian with respect to joint velocity yields the generalized momentum:

$$\frac{\partial \mathcal{L}}{\partial \dot{q}} = \frac{\partial \mathcal{K}}{\partial \dot{q}} = M(q)\dot{q} \tag{64}$$

Differentiating this term with respect to time introduces the time derivative of the inertia matrix:

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}} = M(q)\ddot{q} + \dot{M}(q)\dot{q} \tag{65}$$

Next, the partial derivative of the Lagrangian with respect to joint coordinates accounts for the spatial variation of inertia and the potential energy gradient:

$$\frac{\partial \mathcal{L}}{\partial q} = \frac{1}{2}\frac{\partial}{\partial q}(\dot{q}^T M(q)\dot{q}) - \frac{\partial \mathcal{P}(q)}{\partial q} \tag{66}$$

Substituting these partial derivative terms back into the general Euler-Lagrange equation, we obtain the explicit dynamical equation for the robotic arm:

$$M(q)\ddot{q} + \dot{M}(q)\dot{q} - \frac{1}{2}\frac{\partial}{\partial q}(\dot{q}^T M(q)\dot{q}) + \frac{\partial \mathcal{P}(q)}{\partial q} = \tau \tag{67}$$

To simplify this equation into a standard matrix form, we group the velocity-dependent terms and the position-dependent terms.

The terms involving the product of velocities are grouped into the Coriolis/centripetal vector $V(q, \dot{q})$:

$$C(q, \dot{q}) = \dot{M}(q)\dot{q} - \frac{1}{2}\frac{\partial}{\partial q}(\dot{q}^T M(q)\dot{q}) = \dot{M}\dot{q} - \frac{\partial \mathcal{K}}{\partial q} \tag{68}$$

The term derived from the potential energy represents the gravity vector $G(q)$:

$$G(q) = \frac{\partial \mathcal{P}(q)}{\partial q} \tag{69}$$

By combining the definitions above, the final closed-form dynamic equation for the manipulator is expressed as:

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) = \tau \tag{70}$$

Physical Interpretation:

- $M(q)\ddot{q}$: Inertial forces due to joint acceleration.
- $C(q, \dot{q})$: Coriolis and Centripetal forces arising from link velocities.
- $G(q)$: Gravitational forces acting on the robot links.
- τ: The vector of actuator torques required to produce the motion.

3.3 Derivation of the Inertia Matrix $M(q)$

The total kinetic energy $\mathcal{K}$ of an n-link robotic manipulator is defined as the sum of the translational and rotational kinetic energies of all individual links. For a rigid body system, this is expressed as:

$$\mathcal{K}(q, \dot{q}) = \frac{1}{2} \sum_{i=1}^{n} \left(m_i v_{c_i}^T v_{c_i} + \omega_i^T \mathcal{I}_i \omega_i \right) \tag{71}$$

Where:

- m_i: Mass of link i.
- v_{c_i}: Linear velocity vector of the center of mass (COM) of link i relative to the base frame.
- ω_i: Angular velocity vector of link i relative to the base frame.
- $\mathcal{I}_i$: Inertia tensor of link i expressed in the **base frame**.

The linear and angular velocities of each link can be expressed in terms of the joint velocities $\dot{q}$ using the link-specific Jacobians (J_{v_i} and J_{ω_i}):

$$v_{c_i} = J_{v_i}(q)\dot{q} \quad \text{and} \quad \omega_i = J_{\omega_i}(q)\dot{q} \tag{72}$$

The inertia tensor is typically given in the local frame of the link (denoted as ${}^{i}I_i$, which is constant). To express it in the base frame ($\mathcal{I}_i$), we use the rotation matrix $R_i(q)$ from the base to the link frame:

$$\mathcal{I}_i = R_i \, {}^{i}I_i \, R_i^T \tag{73}$$

Substituting the Jacobian relations and the inertia transformation into Equation (71):

$$\begin{aligned}\mathcal{K} &= \frac{1}{2}\sum_{i=1}^{n}\left[m_i(J_{v_i}\dot{q})^T(J_{v_i}\dot{q}) + (J_{\omega_i}\dot{q})^T(R_i\,{}^iI_i\,R_i^T)(J_{\omega_i}\dot{q})\right] \\ &= \frac{1}{2}\sum_{i=1}^{n}\left[\dot{q}^T\left(m_iJ_{v_i}^TJ_{v_i}\right)\dot{q} + \dot{q}^T\left(J_{\omega_i}^TR_i\,{}^iI_i\,R_i^TJ_{\omega_i}\right)\dot{q}\right]\end{aligned} \tag{74}$$

By factoring out the joint velocity vector $\dot{q}^T$ and $\dot{q}$, we obtain the quadratic form:

$$\mathcal{K} = \frac{1}{2}\dot{q}^T\left[\sum_{i=1}^{n}\left(m_iJ_{v_i}^TJ_{v_i} + J_{\omega_i}^TR_i\,{}^iI_i\,R_i^TJ_{\omega_i}\right)\right]\dot{q} \tag{75}$$

Comparing this result with the standard matrix form of kinetic energy $\mathcal{K} = \frac{1}{2}\dot{q}^TM(q)\dot{q}$, we identify the symmetric, positive-definite Inertia Matrix $M(q)$ as:

$$M(q) = \sum_{i=1}^{n}\left(\underbrace{m_iJ_{v_i}^TJ_{v_i}}_{\text{Translational contribution}} + \underbrace{J_{\omega_i}^T\left(R_i\,{}^iI_i\,R_i^T\right)J_{\omega_i}}_{\text{Rotational contribution}}\right) \tag{76}$$

3.4 Coriolis and Centripetal Forces via Christoffel Symbols

Once the Inertia Matrix $M(q)$ has been computed, the velocity-dependent terms (Coriolis and Centripetal forces) can be derived analytically using the Christoffel symbols of the first kind. This method provides a systematic way to construct the matrix $C(q,\dot{q})$ directly from the partial derivatives of the mass matrix elements.

The generalized force vector resulting from Coriolis and Centripetal effects, denoted as $C(q,\dot{q})$, can be expressed for the k-th joint as:

$$C_k(q,\dot{q}) = \sum_{i=1}^{n}\sum_{j=1}^{n}c_{ijk}\dot{q}_i\dot{q}_j \tag{77}$$

Where c_{ijk} represent the Christoffel symbols, defined as a function of the partial derivatives of the inertia matrix elements $m_{ij}(q)$ with respect to the generalized coordinates.

The Christoffel symbol c_{ijk} corresponds to the coefficient of the velocity product $\dot{q}_i\dot{q}_j$ in the equation of motion for joint k. It is calculated as:

$$c_{ijk} = \frac{1}{2}\left(\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k}\right) \tag{78}$$

A common convention to define the elements C_{kj} is:

$$C_{kj}(q,\dot{q}) = \sum_{i=1}^{n}c_{ijk}\dot{q}_i = \sum_{i=1}^{n}\frac{1}{2}\left(\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k}\right)\dot{q}_i \tag{79}$$

This analytical approach ensures that the resulting matrix $C(q, \dot{q})$ satisfies the skew-symmetric property of $\dot{M} - 2C$, which is crucial for control system stability analysis.

3.5 Gravity Vector Derivation via Jacobian

The gravity vector $G(q)$ represents the generalized torques required at each joint to compensate for gravity acting on the robot links. Instead of differentiating complex trigonometric functions of the potential energy directly, we can derive $G(q)$ using the Translational Jacobian matrices J_{v_i} computed in the kinematics section.

The total potential energy $\mathcal{P}(q)$ of the manipulator is the sum of the potential energies of all n links:

$$\mathcal{P}(q) = -\sum_{i=1}^{n} m_i \mathbf{g}^{T\,0} P_{COM_i}(q) \tag{80}$$

Where:

- m_i: Mass of the i-th link.
- $\mathbf{g} \in \mathbb{R}^3$: Gravity acceleration vector expressed in the base frame (typically $\mathbf{g} = [0, 0, -9.81]^T$).
- ${}^0 P_{COM_i}(q)$: Global position vector of the Center of Mass of link i.

The gravity term $G(q)$ in the dynamic equation is defined as the partial derivative of the potential energy with respect to the joint coordinates:

$$G(q) = \frac{\partial \mathcal{P}(q)}{\partial q} = \begin{bmatrix} \frac{\partial \mathcal{P}}{\partial q_1} & \dots & \frac{\partial \mathcal{P}}{\partial q_n} \end{bmatrix}^T \tag{81}$$

Applying the chain rule, we utilize the definition of the Translational Jacobian, which relates differential changes in joint angles to differential changes in Cartesian position ($\frac{\partial^0 P_{COM_i}}{\partial q} = J_{v_i}$):

$$\frac{\partial \mathcal{P}}{\partial q} = -\sum_{i=1}^{n} \left(\frac{\partial^0 P_{COM_i}}{\partial q} \right)^T (m_i \mathbf{g}) = -\sum_{i=1}^{n} J_{v_i}(q)^T (m_i \mathbf{g}) \tag{82}$$

Thus, the gravity vector $G(q)$ can be computed as the superposition of the gravitational forces acting on each link mapped into the joint space via the transpose of the Jacobian:

$$G(q) = -\sum_{i=1}^{n} J_{v_i}(q)^T m_i \mathbf{g} \tag{83}$$

Where:

- $J_{v_i}(q)^T$: The transpose of the translational Jacobian matrix for link i.

Physical Interpretation: This formula demonstrates the principle of virtual work. The term $J_{v_i}^T$ projects the Cartesian gravity force F_{g_i} acting on link i onto the joint axes, determining how much torque each joint must generate to support that weight.

3.6 Calculation of the Inertia Matrix $M(q)$

The system inertia matrix $M(q)$ is obtained by summing the contributions of kinetic energy from all six links ($k = 1 \dots 6$) projected into the joint space. The global formula is:

$$M(q) = \sum_{k=1}^{6} \left(\underbrace{m_k J_{v_k}^T J_{v_k}}_{\text{Translational}} + \underbrace{J_{\omega_k}^T \left(R_k I_k R_k^T \right) J_{\omega_k}}_{\text{Rotational}} \right) \tag{84}$$

Where:

- m_k, I_k: Mass and Inertia tensor (local) of Link k.
- R_k: Rotation matrix from Base to Link k (${}^0_k R$).
- J_{v_k}, J_{ω_k}: 3×6 Translational and Rotational Jacobians for the Center of Mass of Link k.

The above matrixes have been calculated in 7, 10, 17, 20, 29, 31, 42, 44, 48, 50, 54, 55, 56
The resulting matrix is a 6×6 symmetric positive-definite matrix:

$$M(q) = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} & m_{16} \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} & m_{26} \\ m_{31} & m_{32} & m_{33} & m_{34} & m_{35} & m_{36} \\ m_{41} & m_{42} & m_{43} & m_{44} & m_{45} & m_{46} \\ m_{51} & m_{52} & m_{53} & m_{54} & m_{55} & m_{56} \\ m_{61} & m_{62} & m_{63} & m_{64} & m_{65} & m_{66} \end{bmatrix} \tag{85}$$

Any element m_{ij} (representing the inertial coupling between Joint i and Joint j) is calculated by summing the contributions of all links k that are affected by both joints.

Let $J_{v_k}^{(i)}$ denote the i-th column of the Jacobian matrix J_{v_k} (and similarly for J_ω). The scalar value of m_{ij} is given by:

$$m_{ij} = \sum_{k=1}^{6} \left[m_k \left(J_{v_k}^{(i)} \right)^T J_{v_k}^{(j)} + \left(J_{\omega_k}^{(i)} \right)^T \left(R_k I_k R_k^T \right) J_{\omega_k}^{(j)} \right] \tag{86}$$

Note:

- Since the matrix is symmetric, $m_{ji} = m_{ij}$.

With the elements of the mass matrix m_{ij} determined, the Coriolis and Centrifugal matrix $C(q, \dot{q}) \in \mathbb{R}^{6\times 6}$ can be computed using Christoffel symbols of the first kind. 3.4

The Gravity vector $G(q) \in \mathbb{R}^6$, which accounts for the potential energy of the links, can be efficiently calculated using the linear Jacobian matrices J_{v_k}. Let $\mathbf{g} \in \mathbb{R}^3$ be the gravity acceleration vector (e.g., $\mathbf{g} = [0, 0, -9.81]^T$ defined in the base frame). 3.5

Finally, by combining the inertial, Coriolis/centrifugal, and gravitational terms, the complete closed-form dynamic equation of motion for the robotic manipulator is expressed as:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \tag{87}$$

where:

- $\ddot{q} \in \mathbb{R}^6$ is the vector of joint accelerations.
- $\dot{q} \in \mathbb{R}^6$ is the vector of joint velocities.
- $\tau \in \mathbb{R}^6$ represents the vector of generalized torques applied at the joints.

4 Control Design

4.1 Trajectory Planning

The control problem for the UR10e 6-DOF manipulator begins with trajectory planning in the task space (Cartesian coordinates), followed by Inverse Kinematics (IK) to transform these coordinates into the joint space. This process defines the set of target joint positions $q_{ref} \in \mathbb{R}^6$ for the low-level controller.

In this study, a circular trajectory on XY horizontal plane is defined as the reference path for the end-effector. The trajectory is parameterized by time t, with a fixed height z_{fixed}, a radius R, and a center point (c_x, c_y). The desired position of the end-effector $P_{EE}(t) = [x_{EE}(t), y_{EE}(t), z_{EE}(t)]^T$ is governed by the following equations:

$$\begin{cases} x_{EE}(t) = c_x + R \cdot \cos(\theta(t)) \\ y_{EE}(t) = c_y + R \cdot \sin(\theta(t)) \\ z_{EE}(t) = z_{fixed} \end{cases} \tag{88}$$

where $\theta(t)$ represents the angular progression along the circle, typically defined as a linear function of time $\theta(t) = \omega t$, with ω being the angular velocity.

Inverse Kinematics (IK) problem for the 6DOF UR10e robot is then solved using a geometric decoupling approach. This method divides the problem into two sub-problems:

- **Position IK:** Determining the first three joint angles (q_1, q_2, q_3) to position the wrist center (W) at the desired coordinates.

- **Orientation IK:** Determining the last three joint angles (q_4, q_5, q_6) to align the end-effector orientation with the desired target vector (specifically, pointing downwards).

Given the desired End-Effector position $P_{EE} = [x_E E, y_E E, z_E E]^T$ as a circular trajectory above and a vertical tool orientation, in this case will be downward in global -Z direction, the Wrist Center position P_W is first calculated by offsetting the tool length L_{tool} along the Z-axis and Wrist 2 length d_5:

$$P_W = \begin{bmatrix} x_W \\ y_W \\ z_W \end{bmatrix} = \begin{bmatrix} x_{EE} + d_5 cos(arctan(\frac{y}{x})) \\ y_{EE} + d_5 sin(arctan(\frac{y}{x})) \\ z_{EE} + L_{tool} \end{bmatrix} \tag{89}$$

4.1.1 Position IK

The first three joints determine the position of the wrist center. Based on the geometric projection of the robot arm onto the coordinate planes, the analytical solutions are derived as follows:

Base Joint (q_1): Calculated by projecting the wrist position onto the XY plane. The angle accounts for the geometric offset d_4 (shoulder offset) relative to the radial distance.

$$q_1 = \frac{\pi}{2} + \arccos\left(\frac{d_4}{\sqrt{x_W^2 + y_W^2}}\right) + \arctan\left(\frac{y_W}{x_W}\right) \tag{90}$$

Shoulder Joint (q_2): Determined using the law of cosines on the triangle formed by the upper arm (a_2), forearm (a_3), and the vector connecting the shoulder to the wrist.

$$q_2 = \frac{\pi}{2} - \arctan\left(\frac{z_W - d_1}{\sqrt{x_W^2 + y_W^2 - d_4^2}}\right) - \arccos\left(\frac{A^2 + a_2^2 + a_3^2}{2a_2 A}\right) \tag{91}$$

Where $A = \sqrt{x_W^2 + y_W^2 - d_4^2 + (z_W - d_1)^2}$ is the distance from the shoulder pivot to the wrist center.

Elbow Joint (q_3): Also derived from the law of cosines, representing the internal angle of the triangle formed by links a_2 and a_3.

$$q_3 = \pm \arccos\left(\frac{A^2 - a_2^2 - a_3^2}{2a_2 a_3}\right) \tag{92}$$

Note: The choice of sign for q_3 determines the "Elbow Up" or "Elbow Down" configuration.

4.1.2 Orientation IK

Once q_1, q_2, q_3 are known, the orientation of the third link (forearm) relative to the base frame, denoted as rotation matrix R_{03}, is determined. To achieve the desired end-effector orientation R_{target}, the wrist joints (q_4, q_5, q_6) must satisfy the rotation R_{36}:

$$R_{36} = (R_{03})^T \cdot R_{target} \tag{93}$$

Where R_{03} is calculated from the forward kinematics of the first three joints:

$$R_{03} = \begin{bmatrix} \cos(q_1)\cos(q_{23}) & -\cos(q_1)\sin(q_{23}) & \sin(q_1) \\ \sin(q_1)\cos(q_{23}) & -\sin(q_1)\sin(q_{23}) & -\cos(q_1) \\ \sin(q_{23}) & \cos(q_{23}) & 0 \end{bmatrix} \tag{94}$$

with $q_{23} = q_2 + q_3$.

For the specific task of keeping the tool pointing downwards (aligning the tool Z-axis with the world negative Z-axis), R_{target} is defined accordingly. The elements r_{ij} of the resulting matrix R_{36} are used to extract the wrist angles:

$$q_5 = \arccos(r_{33}) \tag{95}$$

$$q_4 = \arctan 2(r_{23}, r_{13}) \tag{96}$$

$$q_6 = \arctan 2(r_{32}, -r_{31}) \tag{97}$$

This analytical approach ensures that the robot follows the trajectory while maintaining the required vertical orientation for the end-effector.

4.2 Computed-Torque Control with PD Outer Loop

4.2.1 Introduction

Computed-Torque Control (CTC) is a special application of feedback linearization techniques for nonlinear systems. The fundamental idea is to utilize a nonlinear model of the robot manipulator to cancel out the nonlinear dynamics (such as inertia, Coriolis, centrifugal, friction, and gravity forces) in the feedback loop. By doing so, the complex nonlinear design problem is transformed into a simpler design problem for a decoupled linear system. Specifically, the resulting system behaves as a set of n decoupled double integrators, which can be easily controlled using linear control techniques such as Proportional-Derivative (PD) control.

4.2.2 Control Law Derivation

Consider the rigid-body dynamics of an n-link robot manipulator:

$$M(q)\ddot{q} + N(q, \dot{q}) + \tau_d = \tau \tag{98}$$

where:

- $M(q)$ is the $n \times n$ inertia matrix.

- $N(q, \dot{q}) = C(q, \dot{q})\dot{q} + G(q) + F_v(\dot{q})$ represents Coriolis/centrifugal, gravity, and damping friction terms.
- τ_d represents unknown disturbances.
- τ is the control torque vector.

Given a desired trajectory $q_d(t)$, the tracking error is defined as $e(t) = q_d(t) - q(t)$. The Computed-Torque Control law is chosen as:

$$\tau = M(q)(\ddot{q}_d + u) + N(q, \dot{q}) \tag{99}$$

Substituting this control law into the robot dynamics (assuming perfect modeling where $\tau_d = 0$), we obtain the linearized system:

$$\ddot{e} = u \tag{100}$$

Here, $u(t)$ is an auxiliary control signal designed to stabilize the error dynamics.

4.2.3 PD Outer-Loop Design

To ensure the tracking error converges to zero, a PD (Proportional-Derivative) controller is selected for the outer loop:

$$u = K_v\dot{e} + K_p e \tag{101}$$

Substituting u back into the control law, the overall input torque becomes:

$$\tau = M(q)(\ddot{q}_d + K_v\dot{e} + K_p e) + N(q, \dot{q}) \tag{102}$$

The resulting closed-loop error dynamics are given by the second-order linear differential equation:

$$\ddot{e} + K_v\dot{e} + K_p e = 0 \tag{103}$$

By choosing the gain matrices K_v and K_p to be positive definite (usually diagonal), the error system becomes asymptotically stable, ensuring that $e(t) \to 0$ as $t \to \infty$.

4.3 Implementation and Results

Following the definition of the reference trajectory in Section 4.1 and the derivation of the dynamic matrices $M(q)$, $C(q, \dot{q})$, and $G(q)$ in Section 3, a Computed-Torque Control (CTC) scheme was implemented for the UR10e 6-DOF manipulator.

The control architecture, illustrated in Figure 2, consists of an inner nonlinear cancellation loop (feedback linearization) and an outer Proportional-Derivative (PD) tracking loop. The control torque τ is computed as:

$$\tau = \hat{M}(q)(\ddot{q}_d + K_v\dot{e} + K_p e) + \hat{N}(q, \dot{q}) \tag{104}$$

where $\hat{M}$ and $\hat{N}$ represent the estimated dynamic model terms used for compensation, and $e = q_d - q$ is the tracking error.

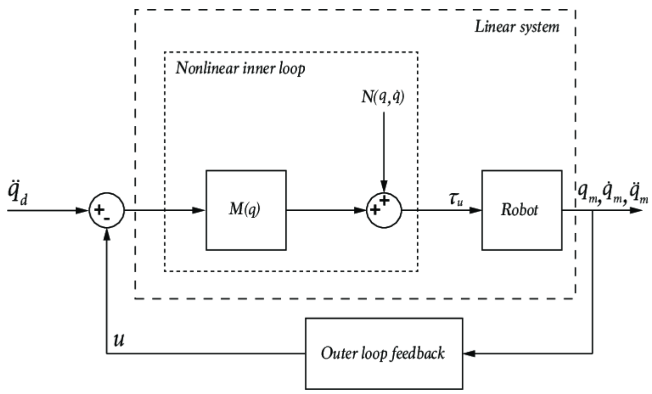

Figure 2: Block diagram of the Computed-Torque Control implementation with an outer PD feedback loop.

The tracking performance is evaluated through Mujoco simulation on VS-Code, Python 3.10, with the results presented below. Figure 3 illustrates the trajectory tracking in the joint space, while Figure 4 depicts the resulting path of the end-effector in Cartesian space.

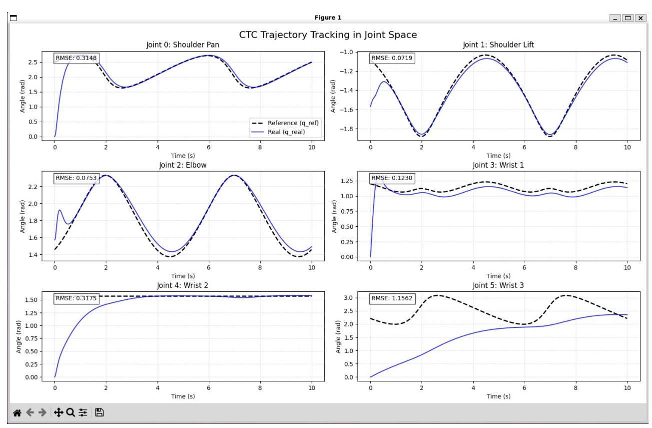

Figure 3: Trajectory tracking performance in joint space (q_{ref} vs q_{real}).

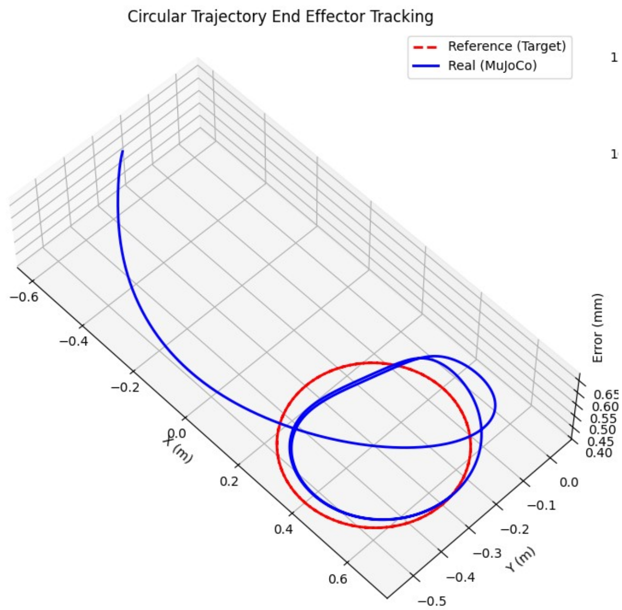

Figure 4: Cartesian trajectory tracking of the End-Effector (Circular Path).

4.3.1 Performance Analysis

While the controller successfully stabilizes the robot along the desired path, certain tracking errors are observed. These deviations are primarily attributed to model mismatches (parametric uncertainties) between the calculated dynamics and the actual robot model in the simulation:

- **Inertia Approximation:** The moment of inertia for each link was simplified as a uniform cylinder during the model calculation. This approximation introduces errors in the inertia matrix $\hat{M}(q)$, preventing the perfect cancellation of the non-linear inertial terms.
- **Unmodeled Friction:** The control law did not explicitly compensate for joint friction. In the simulation environment, undefined friction forces (Coulomb and viscous) act as disturbances, which the linear PD loop can only partially reject, leading to tracking lag.
- **Kinematic Configuration Mismatch:** Discrepancies between the theoretical Denavit-Hartenberg parameters used for Inverse Kinematics and the physical configuration of the robot model (e.g., offsets in the URDF description) contribute to the positioning error at the end-effector level. This demonstrates perfectly the real life performance of UR10e robot,

since there can't be absolute measurement for a real life system without special approximation method or AI support technique.

5 Momentum-Based Observer for Fault Detection

5.1 Theoretical Background

Traditional fault detection and isolation (FDI) methods often rely on comparing the measured joint accelerations with those predicted by the inverse dynamics model. However, numerical differentiation of position encoder signals to obtain acceleration introduces significant noise, making these methods impractical in real-world applications.

To overcome this limitation, a momentum-based observer is proposed. This method utilizes the generalized momentum of the robot, denoted as $p = M(q)\dot{q}$, to generate a residual vector r. The key advantage of this approach is that it transforms the second-order dynamic equations into a first-order form, thereby eliminating the need for acceleration measurements and inversion of the inertia matrix.

The residual vector $r \in \mathbb{R}^n$ is designed to behave as a first-order low-pass filter of the external disturbance or actuator fault. Under ideal conditions (no fault), r converges to zero. When a fault or collision occurs, r rises to estimate the magnitude of the disturbance torque.

5.2 Mathematical Formulation

Consider the rigid body dynamics of the UR10e manipulator:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau + \tau_{ext} \tag{105}$$

where τ is the control torque and τ_{ext} represents the unknown external torque (e.g., collision or actuator fault).

The generalized momentum is defined as:

$$p = M(q)\dot{q} \tag{106}$$

Differentiating p with respect to time and utilizing the skew-symmetric property of the matrix $(\dot{M} - 2C)$, the dynamics of the momentum can be expressed as:

$$\dot{p} = \tau + \tau_{ext} - g(q) + C^T(q, \dot{q})\dot{q} \tag{107}$$

Based on this, the residual vector $r(t)$ is defined utilizing an integral observer structure with a diagonal gain matrix $K > 0$:

$$r(t) = K\left[p(t) - \int_0^t \left(\tau(\sigma) - g(q) + C^T(q, \dot{q})\dot{q} + r(\sigma)\right) d\sigma - p(0)\right] \tag{108}$$

5.3 Implementation

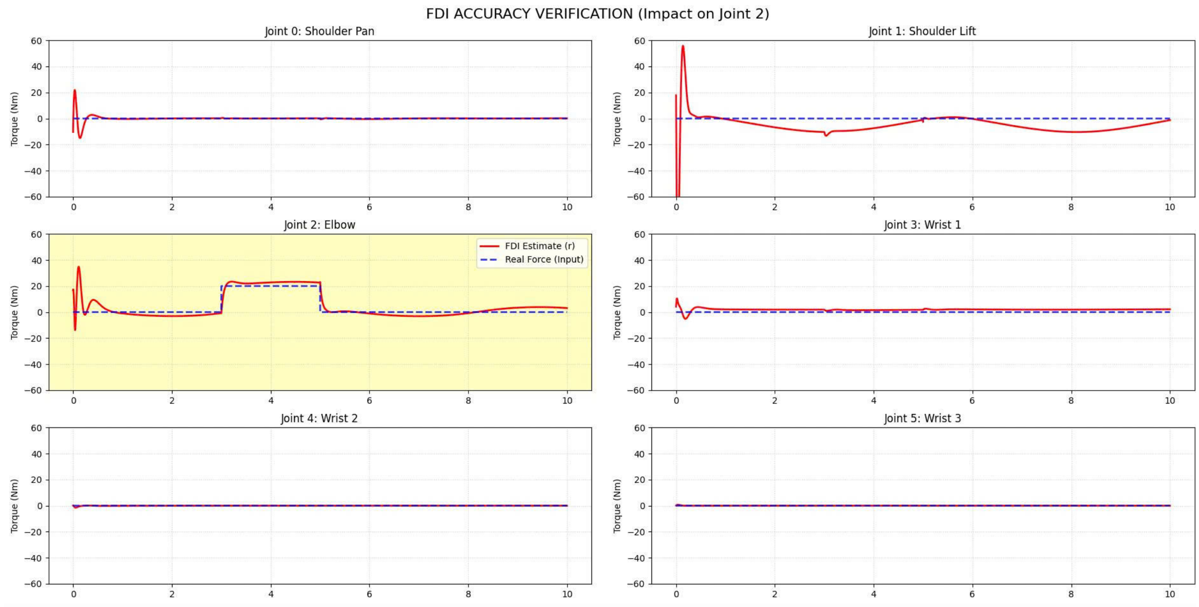

Figure 5: Momentum Observer

A 20Nm Momentum external torque is applied on Joint 2 to validate the effectiveness of the Momentum observer. The result shows that the observer can detect exactly the external force both at magnitude and duration. However, the residual line can't be at absolute 0 value when there's no applied external forces due to model mismatch.