Robot Kinematics and Dynamics - Project Report

Module name: ELE00150M-S2-A Robot Kinematics and Dynamics
Project: Designing, Modeling, Control and Simulation of a 4 joint robot manipulator

Objective

The objective of this project is to design, model, control and simulate a robot manipulator with four degrees of freedom. This report demonstrates the development and control of a PRRR manipulator and modeling of its kinematics and dynamics.

Design

In this project a 4 joint manipulator has been designed with 4 degrees of freedom. The manipulator has one prismatic and 3 revolute joints, likely designed for educational, prototyping, or light industrial applications.

The key design elements consist of:

1. A stable base which provides foundational support for the arm.
2. A prominent vertical column extending upwards from the base. This element incorporates a prismatic joint, allowing the entire body to perform linear sliding motion along the vertical axis.
3. The arm consists of two main segments or links connected by revolute joints (pivot points). These segments allow the arm to extend and retract, providing reach and positioning capabilities.
4. At the very end of the arm, there's a distinct L-shaped end effector, with the offset. A very good use case of the offset for the end effector, by using it as a gripper to grasp an object, or a sensor that would take a reading from a distance, which can be a very good use case for obstacle avoidance application.

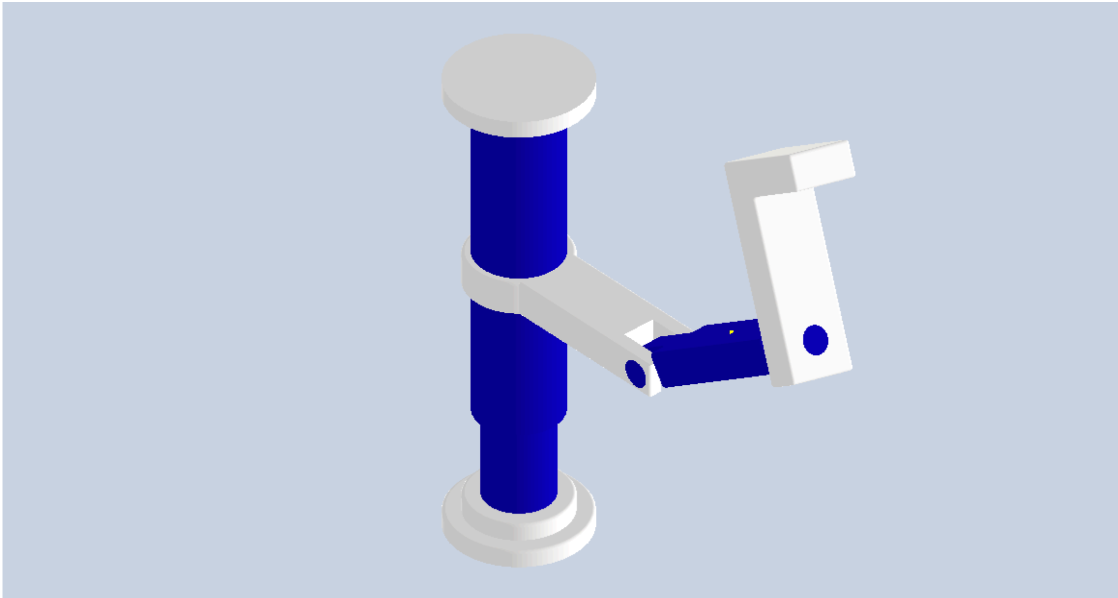


Fig: Assembly of the PRRR manipulator

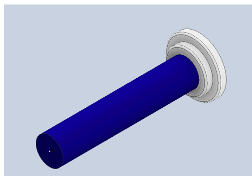
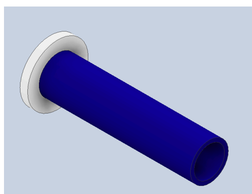
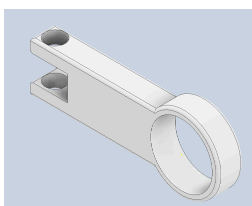
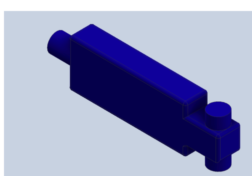
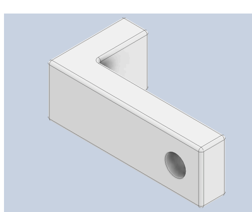
Design Applications

The robot manipulator can primarily be used for manipulation and positioning within a defined workspace. Certainly it can be used in applications of:

- **Pick-and-Place Operations:** Transferring items between locations.
- **Assembly Tasks:** Grasping and positioning components to carry out fundamental assembly operations.
- **Material Handling:** Light-duty handling of small components or materials.

- **Research and Education:** Providing a forum for studying kinematics, control systems, robotics, and fundamental automation concepts. It is appropriate for experiments and demonstrations due to its straightforward design.
- **Testing:** Testing ideas for increasingly intricate robotic systems is known as prototyping.

Dimensions

Link		Joint Type	Length (mm)	Offset(mm)	Dimensions(mm)
1		Prismatic	1000	0	Diameter = 200
2		Prismatic	1000	0	Inner Diameter = 200 Outer Diameter = 250
3		Revolute	Length = 500 Width = 100	0	Joint diameter = 80 Outer Diameter = 300 Inner Diameter = 250
4		Revolute	Length = 500 Width = 100	0	Joint Diameter= 80
5		Revolute	Length = 500 Width = 100	Length = 300 Width = 100	Joint Diameter = 80

Forward Kinematics

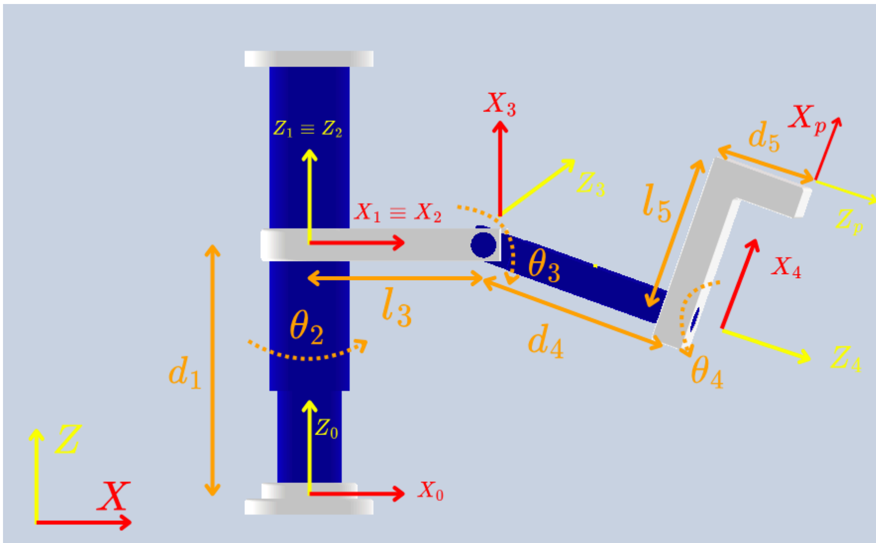


Fig: Diagram of the model with Joint Coordinate frames

DH Parameters

i	α_{i-1}	a_{i-1}	d_i	θ_i
$i = 1$	0	0	d_1	0
$i = 2$	0	0	0	θ_2
$i = 3$	-90°	l_3	0	θ_3
$i = 4$	-90°	0	d_4	θ_4

Homogeneous Transformations

The different Transformation matrices are as follows:

- T01 - First Homogeneous Transformation from frame 0 to 1

T01 =

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- T12 - Second Homogeneous Transformation from frame 1 to 2

T12 =

$$\begin{pmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- T23 - Third Homogeneous Transformation from frame 2 to 3

T23 =

$$\begin{pmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & l_3 \\ 0 & 0 & 1 & 0 \\ -\sin(\theta_3) & -\cos(\theta_3) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- T34 - Fourth Homogeneous Transformation from frame 3 to 4

T34 =

$$\begin{pmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ -\sin(\theta_4) & -\cos(\theta_4) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- T02 - Homogeneous transformation from frame 0 to 2

T02 =

$$\begin{pmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- T03 - Homogeneous transformation from frame 0 to 3

T03 =

$$\begin{pmatrix} \cos(\theta_2)\cos(\theta_3) & -\cos(\theta_2)\sin(\theta_3) & -\sin(\theta_2) & l_3\cos(\theta_2) \\ \cos(\theta_3)\sin(\theta_2) & -\sin(\theta_2)\sin(\theta_3) & \cos(\theta_2) & l_3\sin(\theta_2) \\ -\sin(\theta_3) & -\cos(\theta_3) & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- T04- Homogeneous transformation from frame 0 to 4

T04 =

$$\begin{pmatrix} \sin(\theta_2)\sin(\theta_4) + \cos(\theta_2)\cos(\theta_3)\cos(\theta_4) & \cos(\theta_4)\sin(\theta_2) - \cos(\theta_2)\cos(\theta_3)\sin(\theta_4) & -\cos(\theta_2)\sin(\theta_3) & \cos(\theta_2)\,\sigma_1 \\ \cos(\theta_3)\cos(\theta_4)\sin(\theta_2) - \cos(\theta_2)\sin(\theta_4) & -\cos(\theta_2)\cos(\theta_4) - \cos(\theta_3)\sin(\theta_2)\sin(\theta_4) & -\sin(\theta_2)\sin(\theta_3) & \sin(\theta_2)\,\sigma_1 \\ -\cos(\theta_4)\sin(\theta_3) & \sin(\theta_3)\sin(\theta_4) & -\cos(\theta_3) & d_1 - d_4\cos(\theta_3) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where

$$\sigma_1 = l_3 - d_4\sin(\theta_3)$$

- T4P - Homogeneous Transformation from 4 to end effector(P)

T4P =

$$\begin{pmatrix} 1 & 0 & 0 & l_5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- T0P - Homogeneous Transformation from base 0 to end effector(P)

T0P =

$$\begin{pmatrix} \sigma_1 & \cos(\theta_4)\sin(\theta_2) - \cos(\theta_2)\cos(\theta_3)\sin(\theta_4) & -\cos(\theta_2)\sin(\theta_3) & l_5\,\sigma_1 + \cos(\theta_2)\,\sigma_4 - d_5\cos(\theta_2)\sin(\theta_3) \\ \sigma_2 - \sigma_3 & -\cos(\theta_2)\cos(\theta_4) - \cos(\theta_3)\sin(\theta_2)\sin(\theta_4) & -\sin(\theta_2)\sin(\theta_3) & \sin(\theta_2)\,\sigma_4 - l_5\,(\sigma_3 - \sigma_2) - d_5\sin(\theta_2)\sin(\theta_3) \\ -\cos(\theta_4)\sin(\theta_3) & \sin(\theta_3)\sin(\theta_4) & -\cos(\theta_3) & d_1 - d_4\cos(\theta_3) - d_5\cos(\theta_3) - l_5\cos(\theta_4)\sin(\theta_3) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where

$$\sigma_1 = \sin(\theta_2)\sin(\theta_4) + \cos(\theta_2)\cos(\theta_3)\cos(\theta_4)$$

$$\sigma_2 = \cos(\theta_3)\cos(\theta_4)\sin(\theta_2)$$

$$\sigma_3 = \cos(\theta_2)\sin(\theta_4)$$

$$\sigma_4 = l_3 - d_4\sin(\theta_3)$$

Inverse Kinematics

The inverse kinematics problem involves determining the values of the joint variables that result in a desired end-effector position. In this configuration, The revolute joints θ_2, θ_3 and θ_4 and the prismatic joint d_1 are considered active.

With this our objective is to get the positions of these joints, for a given position of the end effector. Using the symbolic forward kinematics matrix T0P, which provides the homogeneous transformation from the base to the end-effector, we substitute specific joint angle values to evaluate the position of the end-effector under known configurations.

Based on the T0P transformation matrix, the individual values of these joint positions have been derived as follows:

1. θ_3:

```
% for theta_3
theta_3 = acos(-T0P_ideal(3,3))
```

Output:

theta_3 = $\pi - \mathrm{acos}(r_{33})$

2. θ_4:

```
% for theta_4

r31 = -cos(theta4)*sin(theta3);  % first expression
r32 = sin(theta3)*sin(theta4); % second expression

%theta_4 doesn't depend on theta3, since after dividing r31 and r32, the
%sine terms will cancel and will be left with r31/r31 = - tan(Theta4)
%Thus:
theta_4 = atan2(-r32, r31)
```

Output:

theta_4 = $\mathrm{atan2}(-\sin(\theta_3)\sin(\theta_4), -\cos(\theta_4)\sin(\theta_3))$

The output has theta4 in the lhs, because it has not resolved the r31 and r32 parameters. Since both numerator and denominators have $Sin(\theta_3)$, eventually it will cancel out. Thus theta4 will purely depend on the values of r31 and r32, as in the following equation:

$$\theta_4 = \mathrm{atan2}(-r_{32}, r_{31})$$

3. θ_2:

```
% for theta_2

% Compute A and B:
A = sin(theta4);
B = cos(theta3) * cos(theta4);

% Compute R and α: (since, Rcos(theta_2 - alpha) = r11)
R = sqrt(A^2 + B^2);
alpha = atan2(A, B);

%if r11/R > 1, no solution possible
% else if r11/R <= 1:
% two primary solutions for theta_2:
theta2_sol1 = alpha + acos(T0P_ideal(1, 1) / R)
theta2_sol2 = alpha - acos(T0P_ideal(1, 1) / R)
```

Deriving theta 2 from T0P(1, 1), this equation can be re written as

$$Rcos(\theta_2 - \alpha) = T0P_{11}$$

Where R and alpha are the variables from the trigonometric form of a rotation matrix element.

The result depends on the value of $T0P_{11}$/ R. If it was greater than 1 then no solution is possible. This will mean that the pose is outside the reachable workspace. It can also mean the solution violates some mechanical or linkage constraint

Output:

theta2_sol1 =

$$\mathrm{acos}\left(\frac{r_{11}}{\sqrt{\cos(\theta_3)^2\cos(\theta_4)^2+\sin(\theta_4)^2}}\right)+\mathrm{atan2}(\sin(\theta_4),\cos(\theta_3)\cos(\theta_4))$$

theta2_sol2 =

$$-\mathrm{acos}\left(\frac{r_{11}}{\sqrt{\cos(\theta_3)^2\cos(\theta_4)^2+\sin(\theta_4)^2}}\right)+\mathrm{atan2}(\sin(\theta_4),\cos(\theta_3)\cos(\theta_4))$$

4. d_1:

```
% for d1
d1 = T0P_ideal(3, 3) + (d4 + d5)*cos(theta3) + l5*cos(theta4)*sin(theta3)
```

Output:

d1 = $r_{33}+\cos(\theta_3)\ (d_4+d_5)+l_5\cos(\theta_4)\sin(\theta_3)$

Working out on an example of a given position and orientation of the end effector:

For an example, the position of the end effector is:

```
% Define a desired pose T0P_desired (example)
xp = 0.3; yp = 0.1; zp = 0.2; %(unit: meters)
```

And the desired orientation is:

```
R_desired = 3×3
     0     0    -1
     0    -1     0
    -1     0     0
```

The transformation matrix for the end effector would be

```
T0P_desired = 4×4
         0         0   -1.0000    0.3000
         0   -1.0000         0    0.1000
   -1.0000         0         0    0.2000
         0         0         0    1.0000
```

Now after solving for θ_2, θ_3, θ_4 and d_1 from the equations derived previously, we get the values:

```
th_3 = 1.5708
th_4 = -3.1416
theta2_sol1 = -0.4636
theta2_sol2 = -3.6052
d1 = -0.2000
```

where, the angles are represented in radians and distances are mentioned in meters.

Jacobian

The approach is based on symbolic matrix operations using the transformation matrices derived during forward kinematics.

The manipulator consists of a combination of revolute and prismatic joints, and the Jacobian is structured to reflect how joint velocities influence the linear and angular velocity of the end effector.

1. **Rotation Matrices Extraction**: The rotation matrices R01,R02,R03,R04 were extracted from the homogeneous transformation matrices T01,T02,T03,T04. These matrices are used to express the joint axes in the base frame.
2. **Joint Axes in Base Frame**: Each joint axis (assumed to align with the z-axis of its respective frame) is transformed into the base frame.

$$z_i = R_{0i} * (0;\ 0;\ 1)$$

3. **Origin Positions Calculation**: The position of each joint and the end effector is obtained by extracting the position vector (fourth column) from the transformation matrices

```
O0P = T0P(1:3,4);
O0O2 = T02(1:3,4);
O0O3 = T03(1:3,4);
O0O4 = T04(1:3,4);
```

4. **Position Differences for Linear Component**: Displacement vectors from joints to the end effector were calculated to compute the linear velocity components:

```
O2P = O0P - O0O2;
O3P = O0P - O0O3;
O4P = O0P - O0O4;
```

5. **Jacobian Assembly**: The Jacobian matrix is built column by column. For revolute joints, linear velocity components are computed via cross products between joint axes and displacement vectors. Angular components are directly taken as the joint axes. For prismatic joints, this structure is adapted accordingly:

```
J=[z1,cross(z2,O2P),cross(z3,O3P),cross(z4,O4P);...
    zeros(3,1),z2,z3,z4]
```

The output:

J =

$$\begin{pmatrix} 0 & \sigma_1 - \sin(\theta_2)\,\sigma_5 + \sigma_3 & -\cos(\theta_2)\,\sigma_6 & \sin(\theta_2)\sin(\theta_3)\,\sigma_7 - \cos(\theta_3)\,(\sigma_1 + \sigma_3) \\ 0 & \sigma_2 + \cos(\theta_2)\,\sigma_5 - \sigma_4 & -\sin(\theta_2)\,\sigma_6 & -\cos(\theta_3)\,(\sigma_2 - \sigma_4) - \cos(\theta_2)\sin(\theta_3)\,\sigma_7 \\ 1 & 0 & \sin(\theta_2)\,(\sigma_1 + l_3\sin(\theta_2) - \sin(\theta_2)\,\sigma_5 + \sigma_3) - \cos(\theta_2)\,(\sigma_2 - l_3\cos(\theta_2) + \cos(\theta_2)\,\sigma_5 - \sigma_4) & \cos(\theta_2)\sin(\theta_3)\,(\sigma_1 + \sigma_3) + \sin(\theta_2)\sin(\theta_3)\,(\sigma_2 - \sigma_4) \\ 0 & 0 & -\sin(\theta_2) & -\cos(\theta_2)\sin(\theta_3) \\ 0 & 0 & \cos(\theta_2) & -\sin(\theta_2)\sin(\theta_3) \\ 0 & 1 & 0 & -\cos(\theta_3) \end{pmatrix}$$

where:

$\sigma_1 = l_5\ (\cos(\theta_2)\sin(\theta_4) - \cos(\theta_3)\cos(\theta_4)\sin(\theta_2))$

$\sigma_2 = l_5\ (\sin(\theta_2)\sin(\theta_4) + \cos(\theta_2)\cos(\theta_3)\cos(\theta_4))$

$\sigma_3 = d_5\sin(\theta_2)\sin(\theta_3)$

$\sigma_4 = d_5\cos(\theta_2)\sin(\theta_3)$

$\sigma_5 = l_3 - d_4\sin(\theta_3)$

$\sigma_6 = d_4\cos(\theta_3) + d_5\cos(\theta_3) + \sigma_8$

This Jacobian captures the relationship between joint velocities and the spatial velocity of the end-effector.

After computing the full manipulator Jacobian mapping joint velocities to the end-effector linear and angular velocities, we also calculate **intermediate Jacobians** for the **centers of mass** of individual links in the 2D case:

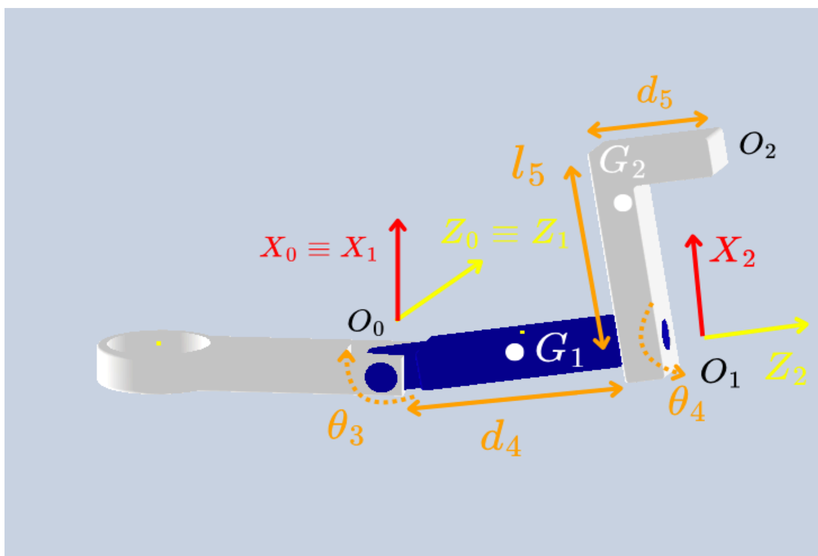


Fig: Diagram of the 2D case

1. **Compute CoM Position in the new Base Frame:** Computed the global position of each CoM in the 2D case by transforming its local coordinates through the respective transformation matrix.

```
v = T01 * [0; lg1; 0; 1]
O0G1 = v(1:3, 1)
```

```
v = T02*[lg2; 0; 0; 1]
O0G2 = v(1:3, 1)
```

2. Getting Relative Vectors and Rotation Matrices:

```
O1G1 = O0G1 - [0; l3; 0]
R01 = T01(1:3, 1:3)
z1 = R01*[0;0;1]
```

```
R02 = T02(1:3,1:3);
z2=R02*[0;0;1];
```

3. Build Jacobian for both the links

```
J1 = [cross(z1, O1G1), zeros(3, 1); z1, zeros(3, 1)]
J2=[cross(z1,O1G2), cross(z2, O2G2);z1, z2]
```

4. Resulting Jacobians:

```
J1 =
```

$$\begin{pmatrix} l_3 - \mathrm{lg}_1 \cos(\theta_3) & 0 \\ -\mathrm{lg}_1 \sin(\theta_3) & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{pmatrix}$$

```
J2 =
```

$$\begin{pmatrix} l_3 - d_4 \cos(\theta_3) - \mathrm{lg}_2 \cos(\theta_4) \sin(\theta_3) & -\mathrm{lg}_2 \cos(\theta_3) \sin(\theta_4) \\ \mathrm{lg}_2 \cos(\theta_3) \cos(\theta_4) - d_4 \sin(\theta_3) & -\mathrm{lg}_2 \sin(\theta_3) \sin(\theta_4) \\ 0 & -\mathrm{lg}_2 \cos(\theta_4) \cos(\theta_3)^2 - \mathrm{lg}_2 \cos(\theta_4) \sin(\theta_3)^2 \\ 0 & -\sin(\theta_3) \\ 0 & \cos(\theta_3) \\ 1 & 0 \end{pmatrix}$$

These Jacobians are used to track internal link motions.

Dynamics of the Manipulator

Centrifugal and Coriolis Vector and the Mass Matrix

In this part of the analysis, the Mass matrix and the Centrifugal and Coriolis vectors for the manipulator were computed, focusing on the last two joints (joints 3 and 4), with the first two joints fixed at zero position.

1. ***Jacobian Partitioning:*** The Jacobians for the linear and angular velocities of the centers of mass of the last two links were divided into their components:

   - Jv1 and Jw1 represent the linear and angular velocity Jacobians of the first link (joint 3), extracted from the previously computed intermediate Jacobians.

     Jv1 =

     $$\begin{pmatrix} l_3 - \mathrm{lg}_1 \cos(\theta_3) & 0 \\ -\mathrm{lg}_1 \sin(\theta_3) & 0 \\ 0 & 0 \end{pmatrix}$$

     Jw1 =

     $$\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{pmatrix}$$

   - Jv2J and Jw2 similarly correspond to the second link (joint 4).

     Jv2 =

     $$\begin{pmatrix} l_3 - d_4 \cos(\theta_3) - \mathrm{lg}_2 \cos(\theta_4) \sin(\theta_3) & -\mathrm{lg}_2 \cos(\theta_3) \sin(\theta_4) \\ \mathrm{lg}_2 \cos(\theta_3) \cos(\theta_4) - d_4 \sin(\theta_3) & -\mathrm{lg}_2 \sin(\theta_3) \sin(\theta_4) \\ 0 & -\mathrm{lg}_2 \cos(\theta_4) \cos(\theta_3)^2 - \mathrm{lg}_2 \cos(\theta_4) \sin(\theta_3)^2 \end{pmatrix}$$

     Jw2 =

     $$\begin{pmatrix} 0 & -\sin(\theta_3) \\ 0 & \cos(\theta_3) \\ 1 & 0 \end{pmatrix}$$

2. ***Inertia Matrix Definition:***

Symbolic inertia tensors I1 and I2 were defined for each link as diagonal matrices with principal moments of inertia as symbolic variables. Masses m_1 and m_2 of the two links were also symbolically defined.

```
I1 =
```

$$\begin{pmatrix} I_{1,1} & 0 & 0 \\ 0 & I_{1,2} & 0 \\ 0 & 0 & I_{1,3} \end{pmatrix}$$

```
I2 =
```

$$\begin{pmatrix} I_{2,1} & 0 & 0 \\ 0 & I_{2,2} & 0 \\ 0 & 0 & I_{2,3} \end{pmatrix}$$

3. *Mass Matrix Computation*

The manipulator's Mass matrix M was computed symbolically by combining contributions from each link's translational and rotational kinetic energy

- Translational kinetic energy terms are formed by $m_i J_{vi}^T J_{vi}$.
- Rotational kinetic energy terms are formed by $J_{wi}^T R_{0i} I_i R_{0i}^T J_{wi}$, where R_{0i} are the rotation matrices from base to link frames.

The total Mass matrix is the sum of these terms for both links, simplified symbolically for clearer expressions.

```
M = simplify(m_1 * transpose(Jv1) * Jv1 + transpose(Jw1) * R01 * I1
* transpose(R01) * Jw1 + m_2 * transpose(Jv2) * Jv2 + transpose(Jw2)
* R02 * I2 * transpose(R02) * Jw2)
```

where M is the mass matrix.

Output:

```
M =
```

$$\begin{pmatrix} I_{1,3} + m_2\,(d_4\cos(\theta_3) - l_3 + \mathrm{lg}_2\cos(\theta_4)\sin(\theta_3))^2 + m_1\,(l_3 - \mathrm{lg}_1\cos(\theta_3))^2 + m_2\,(d_4\sin(\theta_3) - \mathrm{lg}_2\cos(\theta_3)\cos(\theta_4))^2 + I_{2,2}\cos(\theta_4)^2 + I_{2,1}\sin(\theta_4)^2 + \mathrm{lg}_1{}^2\, m_1 \sin(\theta_3)^2 & \sigma_1 \\ \sigma_1 & m_2\,\mathrm{lg}_2{}^2 + I_{2,3} \end{pmatrix}$$

where

$$\sigma_1 = \mathrm{lg}_2\, m_2 \sin(\theta_4)\,(d_4 - l_3\cos(\theta_3))$$

4. *Centrifugal and Coriolis Vectors*

These vectors represent the coupling gradients and centrifugal reduction of the joints, and the impact of motion of one joint onto the other and vice-versa.

To obtain the Coriolis matrix terms, partial derivatives of the Mass matrix entries with respect to joint variables θ3 and θ4 were computed symbolically. Two auxiliary matrices, CC1 and CC2 which are the components of the Coriolis and centrifugal force vector, were calculated based on standard formulas involving these derivatives to capture Coriolis and centrifugal forces acting on the joints.

CC1:

ans =

$$\begin{pmatrix} \sin(\theta_4)\ \left(-m_2\cos(\theta_4)\,\mathrm{lg}_2{}^2 + l_3\,m_2\sin(\theta_3)\,\mathrm{lg}_2 + I_{2,1}\cos(\theta_4) - I_{2,2}\cos(\theta_4)\right) & l_3\,\mathrm{lg}_2\,m_2\sin(\theta_3)\sin(\theta_4) \\ \dfrac{\sin(2\,\theta_4)\ \left(m_2\,\mathrm{lg}_2{}^2 - I_{2,1} + I_{2,2}\right)}{2} & 0 \end{pmatrix}$$

CC2:

ans =

$$\begin{pmatrix} \dfrac{\sin(\theta_4)\ \left(-4\,m_2\cos(\theta_4)\,\mathrm{lg}_2{}^2 + 5\,l_3\,m_2\sin(\theta_3)\,\mathrm{lg}_2 + 4\,I_{2,1}\cos(\theta_4) - 4\,I_{2,2}\cos(\theta_4)\right)}{2} & 0 \\ 0 & 0 \end{pmatrix}$$

Total Potential Energy and Gravity Vector

The symbolic expressions for potential energy are computed using the center of mass of each link, and the gravitational acceleration acting in the negative X-direction (i.e., −9.81 m/s² along the X-axis).

1.Potential Energy of the links

```
% Potential Energy of the 1st link
V1 = -(m_1 * [-vpa('9.81'),  sym(0),sym(0)] * O1G1)
% Potential Energy of the 2nd link
V2 = -(m_2 * [-vpa('9.81'),  sym(0),sym(0)] * O1G2)
```

Where, O1G1 and O1G2 are the coordinates of the Center of Masses of the respective links

Output:

$$V1 = -9.81\, \mathrm{lg}_1\, m_1 \sin(\theta_3)$$

$$V2 = -9.81\, m_2\, (d_4 \sin(\theta_3) - \mathrm{lg}_2 \cos(\theta_3) \cos(\theta_4))$$

2. Total Potential Energy

The Total Potential Energy is given by :

$$V = V_1 + V_2$$

$$V = -9.81\, m_2\, (d_4 \sin(\theta_3) - \mathrm{lg}_2 \cos(\theta_3) \cos(\theta_4)) - 9.81\, \mathrm{lg}_1\, m_1 \sin(\theta_3)$$

3. Gravity Vector

The Gravity Vector yields the symbolic torque contributions from gravity acting on the active joints, which can be later used in the dynamic control model.

To derive the gravity vector G, we compute the partial derivatives of the total potential energy with respect to the joint angles θ3 and θ4:

```
Gravity = simplify([diff(V, theta3); diff(V, theta4)])
```

Output:

$$\text{Gravity} = \begin{pmatrix} -9.81\, m_2\, (d_4 \cos(\theta_3) + \mathrm{lg}_2 \cos(\theta_4) \sin(\theta_3)) - 9.81\, \mathrm{lg}_1\, m_1 \cos(\theta_3) \\ -9.81\, \mathrm{lg}_2\, m_2 \cos(\theta_3) \sin(\theta_4) \end{pmatrix}$$

Control in Joint Space

The objective is to design and implement a Proportional-Derivative (PD) controller for regulating the joint positions of the robotic manipulator in joint space.

The CAD model of the robotic manipulator was created and fully constrained in a corresponding Simulink model of the manipulator, including all joints and rigid bodies.

PD Controller Design

The PD control law for each joint is given by:

For revolute Joints: $\tau_i = K_{p,i}(\theta_{d,i} - \theta_i) + K_{d,i}(\overline{\theta}_{d,i} - \overline{\theta}_i)$

For prismatic Joints: $F_i = K_{p,i}(x_{d,i} - x_i) + K_{d,i}(\overline{x}_{d,i} - \overline{x}_i)$

Where:

τ_i - torque applied to revolute joint *i* (Nm)

F_i - force applied to prismatic joint *i* (N)

$\theta, \overline{\theta}$ - angular position and velocity (rad, rad/s)

$x, \overline{x}$ - linear position and velocity (m, m/s)

$K_{p,i}$, $K_{d,i}$ - Proportional and Derivative gains

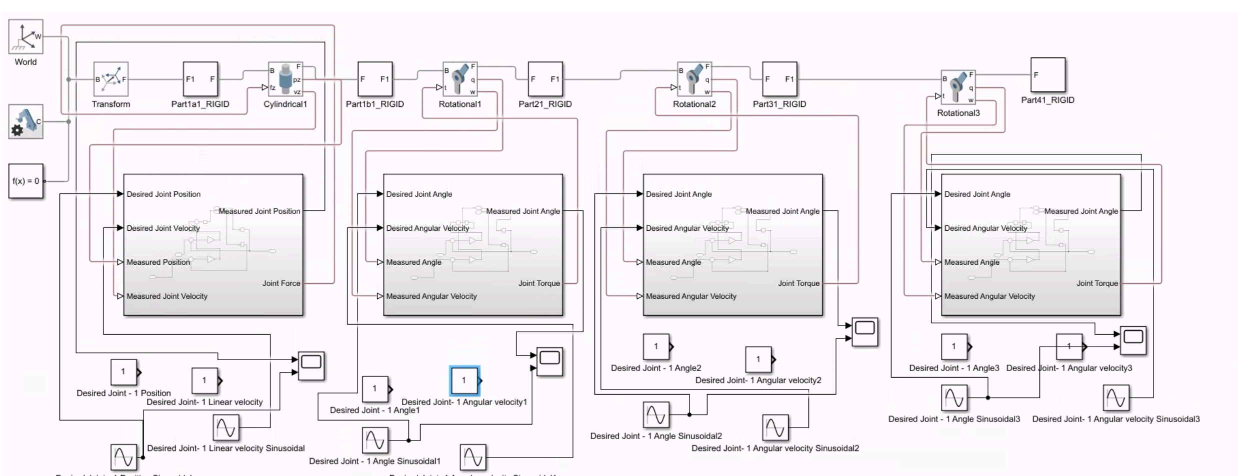


Fig: Control Diagram of all the joints

Simulink Implementation

- A time-varying signal was used to specify the desired joint position $\theta_{d,i}$ or $x_{d,i}$
- The sum blocks are used to compute the error and its derivative.
- Gain blocks are applied for proportional and derivative gains.
- The resulting control signal was fed to the torque port for revolute joints and force port for prismatic joints.
    - Joint 1: Prismatic

      $K_{p,1} = 300000, K_{d,1} = 2000$

    - Joint 2: Revolute

      $K_{p,2} = 500, K_{d,2} = 20$

    - Joint 3: Revolute

      $K_{p,3} = 2000, K_{d,3} = 1700$

    - Joint 4: Revolute

      $K_{p,4} = 1500, K_{d,4} = 50$

Simulation and Results

The PD controller was tested on the imported model by applying step commands to each joint. Joint positions and velocities were monitored and compared with the desired profiles. The controller was able to track the desired joint positions accurately with minimal overshoot and acceptable settling time.

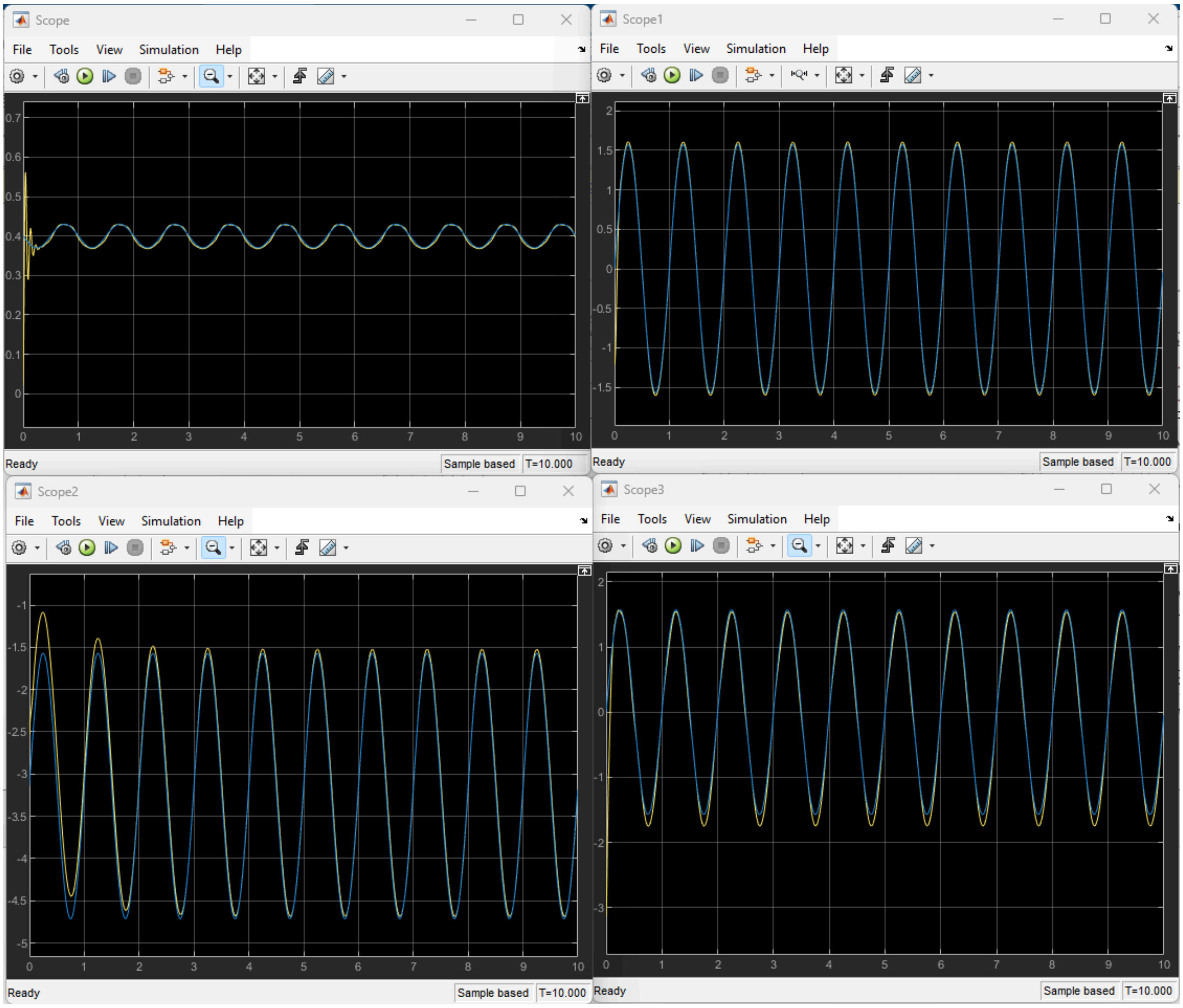


Fig: Graphs show the simulation graphs for each joint. Scope0 represents the graph for the prismatic joint, while Scope1, 2 and 3 represent the graph for the revolute joints.

Thus, the PD controller was successfully designed and implemented in joint space for the robotic manipulator. The controller demonstrated effective regulation of both revolute and prismatic joint positions in simulation, laying the foundation for potential hardware implementation.

Conclusion

The modelling, analysis, and control of a 4-DOF robotic manipulator have all been covered in detail in this report. In order to determine the connection between joint parameters and end-effector position, we first developed the robot's kinematic and dynamic models. From there, we derived both forward and inverse kinematics. These expressions served as the foundation for later control tasks and were symbolically validated.

After that, a joint-space PD controller was added to the Simscape environment, allowing for fine-grained control over the manipulator's motion. To ensure physical consistency throughout the model, special attention was paid to applying torques for revolute joints and forces for prismatic joints correctly.

According to simulation results, the PD controller tracked desired joint positions accurately, had reasonable response times, and had little overshoot. A comprehensive workflow that can be used in simulation and real-world deployment is demonstrated by the combination of feedback control, mechanical modelling, and symbolic derivation.

This work offers hands-on experience with model-based design using Simscape in addition to reinforcing theoretical concepts in robot kinematics and dynamics. The techniques and resources employed here provide a strong basis for the development of more sophisticated control strategies in the future.