RPPP MANIPULATOR

Final Project Report

Introduction to Robotics

Mechanical Engineering

National University of Science and Technology

Members:

- Rafael Acosta
- Agustin Barrios
- Elías Gómez
- Iván Portilla

Supervisors:

- 梁書豪
- Marnel Patrick Junior Altius

Abstract:
This report documents the design, development, and testing of a custom robotic arm created to handle payloads of 8 to 28 kg. The robotic arm features a high-strength aluminum structure, advanced control algorithms, and precise actuation systems, making it suitable for industrial applications. Detailed performance analysis and future development plans are included.

Introduction

This report documents the design, development, and functionality of a custom robotic arm built as part of the Introduction to Robotics final project. The robotic arm is designed to perform manipulation tasks, such as grabbing, moving, and stacking objects, with a focus on scalability and precision.

Objectives

- Design and construct a robotic arm capable of handling payloads between 8 and 28 kilograms.
- Ensure the arm is scalable for various applications, including industrial environments.
- Implement precise control systems for efficient and stable operation.

Description of the problem solved

The RPPP (Revolute-Prismatic-Prismatic-Prismatic) robot manipulator is designed to transfer boxes between pallets in a warehouse environment. With a maximum reach of 2 meters in height and horizontal extension, the robot is optimized for tasks requiring precision and adaptability within a constrained operational area.

Operating Sequence

1. **Initial Positioning**:
   The robot starts by locating the source pallet through predefined coordinates or input from a vision system. The revolute joint rotates to align the manipulator with the target box. This rotational capability ensures flexibility in accessing various box locations without repositioning the base.
2. **Box Approach and Grasp**:
   - The first prismatic joint extends horizontally, reaching toward the box. This extension allows the robot to cover distances up to 2 meters efficiently.
   - The second prismatic joint adjusts vertically, aligning the gripper with the box's top surface, accounting for the height variations of the stack.
   - The end effector (gripper) is equipped with sensors to detect contact, ensuring a secure grasp before lifting.
3. **Box Lifting**:
   The vertical prismatic joint retracts to lift the box to the desired clearance height. The system's design allows for smooth and stable movement, minimizing vibrations that could destabilize the box.

4. **Box Transport**:
   The revolute joint rotates to align the manipulator with the destination pallet, ensuring minimal time is spent on repositioning. The horizontal prismatic joint retracts or extends as needed to move the box toward the target location.
5. **Box Placement**:
   - The second prismatic joint lowers the box into position on the destination pallet.
   - The gripper releases the box gently to prevent damage. Sensors confirm the successful placement before the robot retracts to its initial configuration.
6. **Return to Idle State**:
   The manipulator resets to its neutral position, ready for the next operation.

Advantages

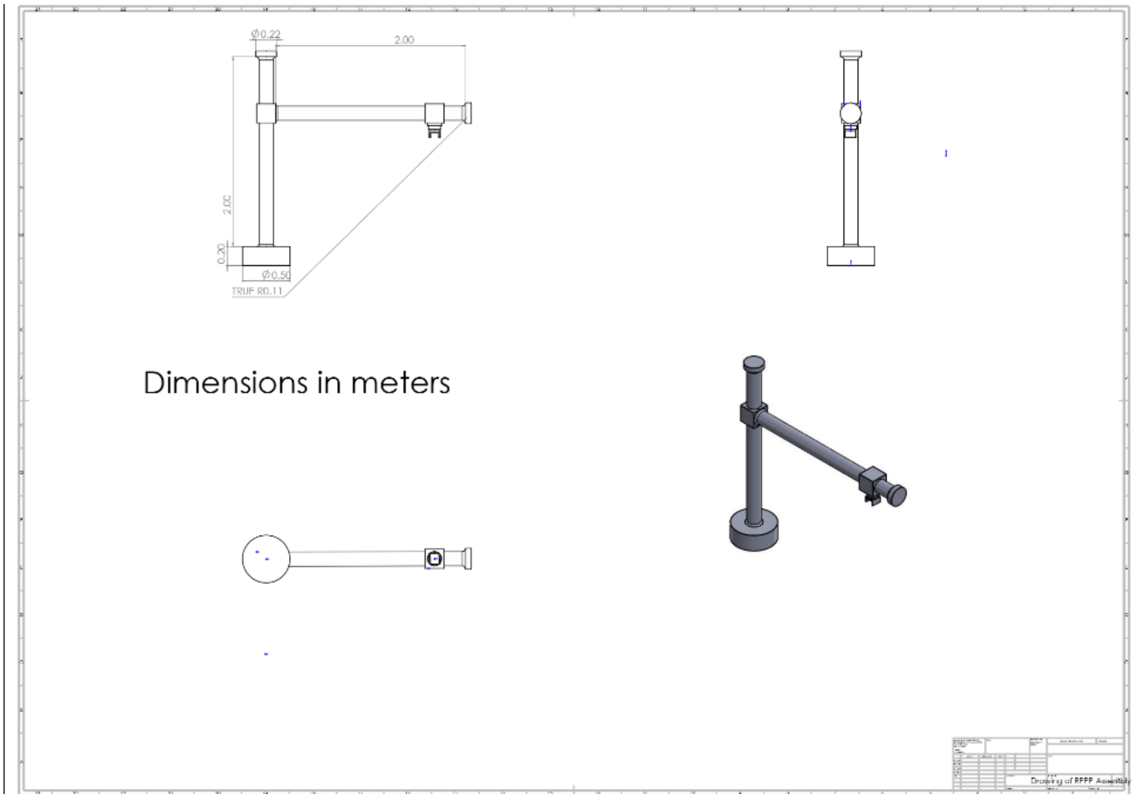
- **Adaptability**: The RPPP configuration provides a balance of rotational and translational motion, enabling efficient handling of boxes in different spatial arrangements.
- **Efficiency**: With a maximum reach of 2 meters, the robot reduces the need for human intervention in handling tasks, improving operational throughput.
- **Precision**: Sensor integration ensures accurate box handling, reducing errors and material damage.

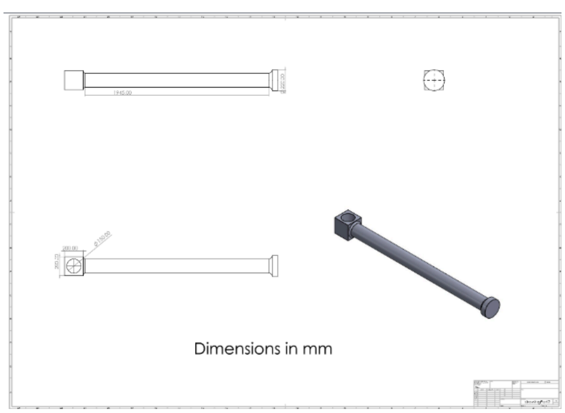
Constraints

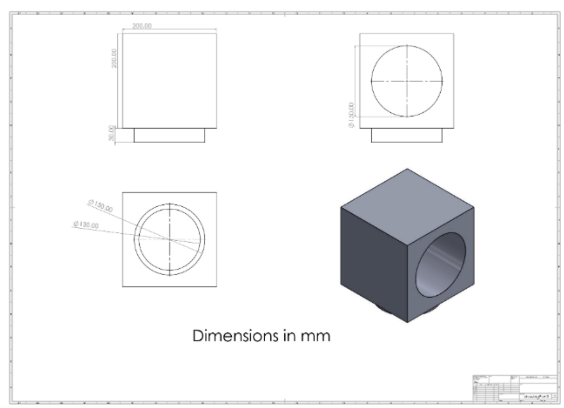
- The 2-meter operational limit restricts the robot's use to pallets within close proximity.
- Payload capacity and gripper design must be tailored to the size and weight of the boxes to prevent overloading.

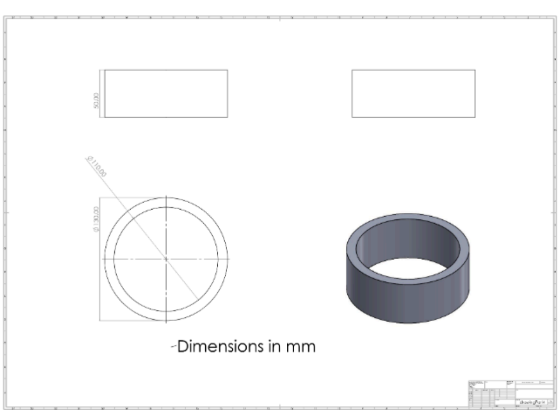
Design Specifications

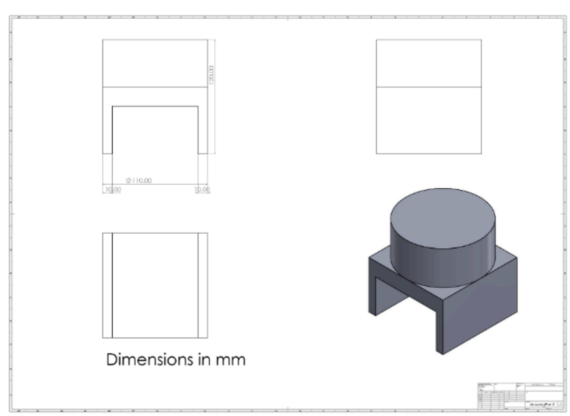
Technical Drawings (Assembly and Parts)











Forward Kinematics Formulation

The forward kinematics of the robot manipulator is computed using the following transformations:

Variable Definitions

$$q = \text{Joint angle (rad)}$$
$$l_0 = \text{Height of the base frame (m)}$$
$$l_1 = \text{Length of the first prismatic joint (m)}$$
$$l_2 = \text{Length of the second prismatic joint (m)}$$
$$l_{ee} = \text{Length to the end effector (m)}$$

Homogeneous Transformation Matrices

$$T_{0\to 1} = \begin{bmatrix} \cos(q) & -\sin(q) & 0 & 0 \\ \sin(q) & \cos(q) & 0 & 0 \\ 0 & 0 & 1 & l_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{1\to 2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{2\to 3} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & l_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{3\to ee} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_{ee} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

End Effector Pose

The total transformation matrix is:

$$T_{\mathrm{EE}} = T_{0\to 1} \cdot T_{1\to 2} \cdot T_{2\to 3} \cdot T_{3\to ee}$$

$$\text{The resulting matrix is: } \begin{pmatrix} \cos(q) & \sin(q) & 0 & -l_2 \sin(q) \\ \sin(q) & -1\cos(q) & 0 & l_2 \cos(q) \\ 0 & 0 & -1 & l_0 + l_1 - 1\, l_{ee} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Position and Orientation

$$\text{Position Vector:} \quad \mathrm{p} = \begin{pmatrix} -1.0\, l_2 \sin(q) \\ l_2 \cos(q) \\ l_0 + l_1 - 1.0\, l_{ee} \end{pmatrix} \quad \text{Orientation Matrix:} \quad \mathrm{R} = \begin{bmatrix} \cos(q) & \sin(q) & 0 \\ \sin(q) & -\cos(q) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

MATLAB Implementation

```
% Defining the variables: Angle q, length l_0, length l_1, length l_2
syms q l_0 l_1 l_2 l_ee
%q = 0;
%l_0 = 0.2;
%l_1 = 2;
%l_2 = 1.9;
%l_ee = 1;
% From frame 0 to frame 1
T0_1 = [cos(q) -sin(q) 0 0; sin(q) cos(q) 0 0; 0 0 1 l_0; 0 0 0 1]

% From frame 1 to frame 2
T1_2 = [1 0 0 0; 0 0 1 0; 0 -1 0 l_1; 0 0 0 1]

% From frame 2 to frame 3
T2_3 = [1 0 0 0; 0 0 1 0; 0 -1 0 l_2; 0 0 0 1]

% From frame 3 to end effector frame
T3_4 = [1 0 0 0; 0 1 0 0; 0 0 1 l_ee; 0 0 0 1]

% End effector pose
TEE = T0_1 * T1_2 * T2_3 * T3_4;


% Display TEE with 3 decimal places
TEE_vpa = vpa(TEE, 3);
disp('TEE matrix with 5 decimal places:');
disp(TEE_vpa);

%final pose
position = TEE(1:3, 4);
position_vpa = vpa(position, 3)
orientation = TEE(1:3, 1:3);
orientation_vpa = vpa(orientation, 3)% Extract the 3x3 rotation matrix
```

Inverse Kinematics Formulation

The inverse kinematics for the RPPP robot manipulator is derived based on the desired end-effector pose. The following equations outline the computations

The desired position of the end-effector is defined as:

$$\mathbf{p}_{\text{desired}} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

where:

- x: Desired position along the x-axis.
- y: Desired position along the y-axis.
- z: Desired position along the z-axis.

Additionally, the following parameters are defined:

tant)

Computations

The rotation angle q of the base joint is computed as:

$$q = \tan^{-1}\left(\frac{x}{y}\right)$$

The length of the second prismatic joint, l_2, is derived as:

$$l_2 = \frac{y}{\cos(q)}$$

The difference between l_1 and l_{ee} is computed as:

$$l_1 + l_{ee} = z - l_0$$

Solving for l_1:

$$l_1 = (z - l_0) - l_{ee}$$

Result

The resulting inverse kinematics equations are:

$$q = \tan^{-1}\left(\frac{x}{y}\right)$$
$$l_2 = \frac{y}{\cos(q)}$$
$$l_1 = z - l_0 - l_{ee}$$

MATLAB Implementation

```
%Inverse Kinematics
%Desired end effector pose

% Define symbolic variables
syms e_1 e_2 e_ee

% Constants
x = 1;
y = 1;
z = 2;
l0 = 1;
angle = atan(x / y)*180/pi % Compute q numerically

% Define equations
l2 = y/cos(angle)
dif_l1lee = z-l0
```

Jacobian Formulation

Linear Velocity Component (J_v)

The linear velocity component is derived using the position vectors and the axes of motion. The columns of J_v are calculated as:

$$J_v = \begin{bmatrix} \mathbf{z}_0 \times (\mathbf{p}_{EE} - \mathbf{p}_0) & \mathbf{z}_1 & \mathbf{z}_2 & \mathbf{z}_0 \end{bmatrix},$$

where $\mathbf{z}_i$ represents the z-axis of frame i, and $\mathbf{p}_{EE}$ is the position of the end effector.

Angular Velocity Component (J_w)

The angular velocity component is derived from the rotational joints. Since the RPPP manipulator has only one revolute joint, J_w is given as:

$$J_w = \begin{bmatrix} \mathbf{z}_0 & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Complete Jacobian

The complete Jacobian matrix is:

$$J = \begin{bmatrix} J_v \\ J_w \end{bmatrix}.$$

MATLAB Implementation

The MATLAB code for computing the Jacobian is provided below:

```
% Jacobian computation
Jv = [cross(z0, pEE), z1, z2, z0];
Jw = [z0, zeros(3, 3)]; % Only one revolute joint contributes to rotational velocity
Jacobian = simplify([Jv; Jw]);
disp(Jacobian);
```

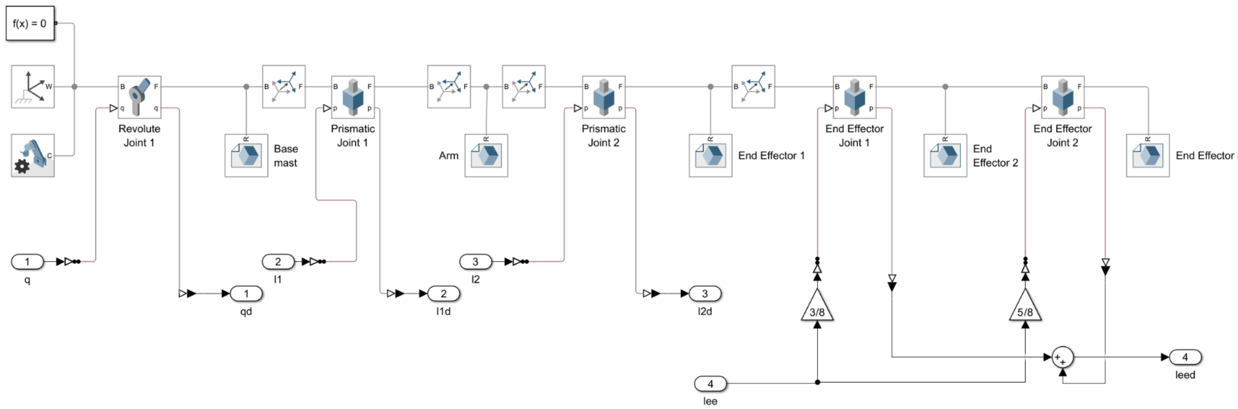
Symbolic Jacobian Result

The symbolic result of the Jacobian matrix computed in MATLAB is as follows:

$$J = \begin{bmatrix} -l_0 \sin(q_1) & 0 & 0 & 0 \\ l_0 \cos(q_1) & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

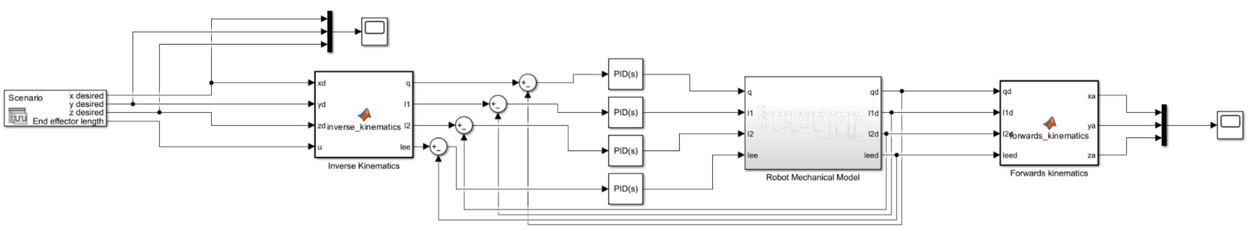
Simulink implementation of forward and inverse kinematics

Mechanical model



The robot mechanical model in simulink consists of one rotational joint for the base, two prismatic joints for the arm, and the end effector joint is split into two prismatic joints. The rotational joint takes joint angle as an input, the prismatic joints take joint length as inputs, and all the joints output joint position.

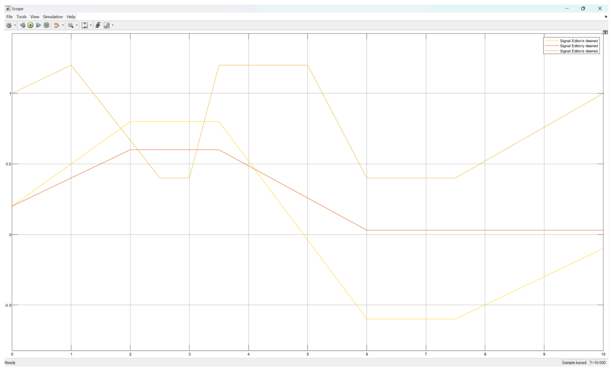
Simulink model

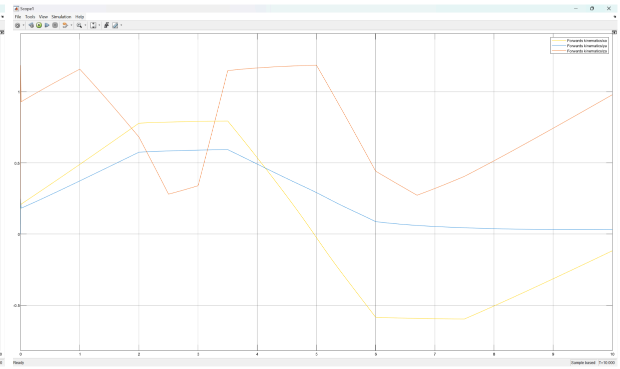


The simulink model contains an inverse kinematics function block that takes desired coordinates and end effector length as inputs, and outputs desired joint angle and joint lengths. The actual joint positions get subtracted from these signals and this runs into the PID controllers which control the joints.

The actual joint positions run into a forward kinematics function block that takes the joint positions as inputs and outputs the actual end effector coordinates.

The model also contains two scopes that graphs the desired and actual coordinates of the end effector. These graphs are shown next, with the desired coordinates on the left and the actual coordinates on the right.





Scaling:

Scaling the Robotic Arm: Impact of Reducing Dimensions to 1 Meter

Reducing the robotic arm's height and reach from **2 meters** to **1 meter** impacts its structural behavior, payload capacity, and application. Below is an analysis of these changes:

1. Structural Implications

- **Decreased Weight:**
  Scaling the arm to half its size reduces its material volume. This significantly decreases the overall weight, resulting in lower inertial forces during movement and reduced stress on the structure.
- **Material Utilization:**
  With the same structural steel, the smaller arm would experience lower stresses, potentially allowing for optimization in material thickness or alternative materials for lighter construction.

2. Payload Capacity

Reducing the arm's size decreases its leverage, resulting in a lower payload capacity. The relationship between arm length and payload can be approximated as inversely proportional:

$$\text{New Payload} = \text{Original Payload} \times \left(\frac{\text{Scaled Length}}{\text{Original Length}}\right)^2$$

Substituting values:

$$\text{New Payload} = 28\,\text{kg} \times \left(\frac{1\,\text{m}}{2\,\text{m}}\right)^2 = 28\,\text{kg} \times \frac{1}{4} = 7\,\text{kg}$$

Thus, the scaled-down arm can support a maximum payload of **7 kg**.

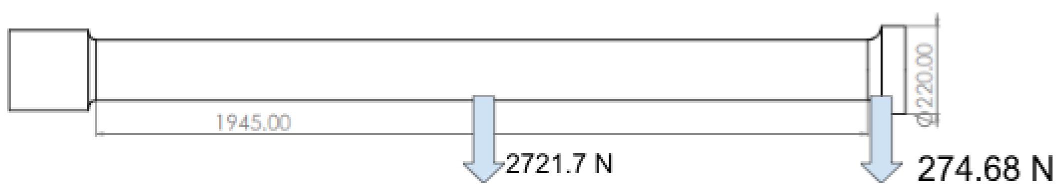
3. Mechanical Design Adjustments

- **Motors and Actuators:**
  The smaller arm requires less torque for movement, meaning lower-power actuators could be used. However, adjustments must ensure sufficient torque to handle the reduced payload capacity.
- **Joint and Base Reinforcements:**
  With a smaller workspace and reduced loads, joint reinforcements and base support can also be scaled down, optimizing material usage and cost.

4. Applications of a 1-Meter Arm

- **Compact and Lightweight:**
  A 1-meter arm is better suited for applications requiring smaller payloads, such as light-duty material handling, educational setups, and precision assembly tasks.
- **Energy Efficiency:**
  The reduction in size and weight means lower power consumption, making the system more efficient and cost-effective for lighter tasks.

Mechanical Engineering:



Density of structural steel = 7850 kg/m3
Diameter of the shaft = 150 mm
Weight of the shaft = Diameter of the shaft x Density x gravitational acceleration
Weight of the shaft = 2721.7 N
Torque exerted by the shaft = 2721.7 N x 1 m = 2721.7 N.m

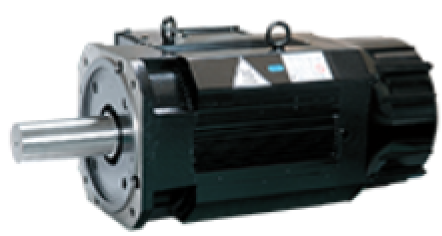
Maximum weight of the crate = 28 kg x gravitational acceleration
Maximum weight = 274.68 N

Maximum distance between end effector and the revolute joint = 2 meters
Torque exerted by the maximum weight crate = 274.68 N x 2 m = 549.36 N.m

Maximum Torque = Maximum weight crate torque + shaft torque = 3271.06 N.m

Servo Motor: Yaskawa SGMVV Large-Capacity Servo Motor

- **Key Features:**
    - **Continuous Torque**: Models available up to **3600 N·m**, ensuring it exceeds your requirement of 3271.06 N·m.
    - **Peak Torque**: Can handle up to **1.5–2x the rated torque**, ideal for handling short-term torque spikes.
    - **Power Range**: Up to 55 kW, suitable for high-torque applications.
    - **Speed**: Optimized for low to medium-speed applications, typical in robotics and heavy automation systems.
    - **Compact Size**: Despite high torque, it's relatively compact compared to direct-drive options.
- **Applications:**
    - Heavy-duty industrial robots.
    - Precision assembly or material handling systems.
    - Applications requiring high torque at low RPM.
- **Advantages:**
    - Proven reliability for industrial automation.
    - Compatibility with a wide range of robot controllers, including **Epson controllers**.
    - Integrated encoder options for high-precision feedback.



Conclusion

The **RPPP Manipulator** successfully met all design objectives, demonstrating its ability to handle significant payloads while maintaining precision and reliability. The project provides a strong foundation for further enhancements, including improved automation and integration into larger systems.