

**ELEC60030 - Robotic Manipulation
Coursework Report
Group B1**

A report submitted in partial fulfilment of the module requirements for
ELEC60030 - Robotic Manipulation 2024 - 2025

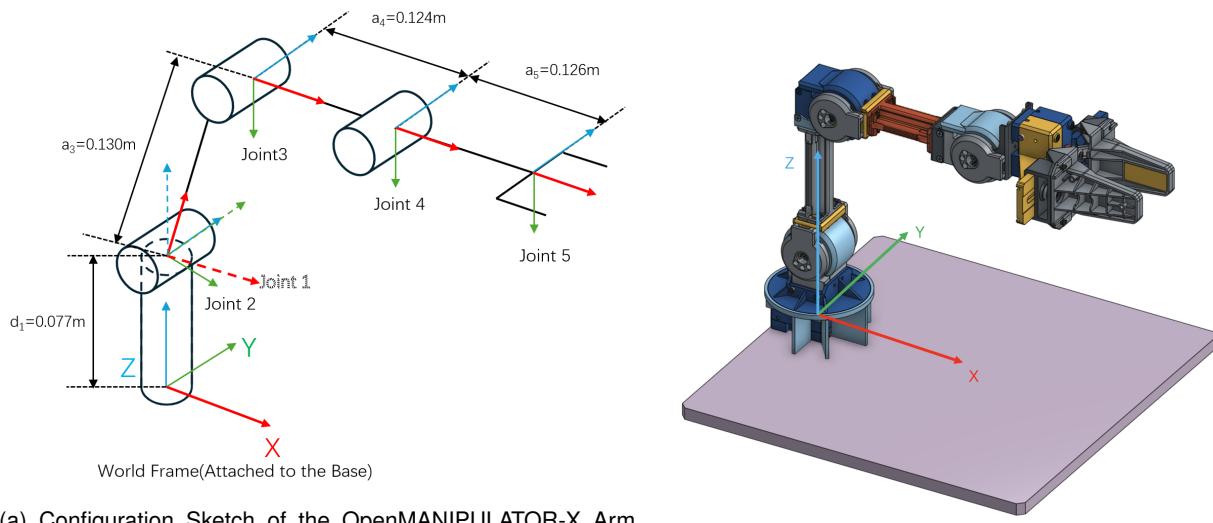
List of Contents

1	Task 1: Model the robot	1
1.1	Assigning Co-ordinate Frames and Creating DH Table for the Robot	1
1.2	Forward Kinematics and Graphical Simulation of the Robot	2
1.3	Trajectory Planning	3
2	Appendix	4
2.1	Inverse Kinematics	4
2.2	Task 3 3D Printed Objects	6

1 Task 1: Model the robot

1.1 Assigning Co-ordinate Frames and Creating DH Table for the Robot

In this coursework, all the robotic configurations and task completions were based on the OpenMANIPULATOR-X robot arm manufactured by Robotis, and the configuration sketch of this arm is plotted in Figure 1a, with all the co-ordinate frames assigned and attached to the joints. The world frame was attached to the robot arm's base. Frame choices for each joint aimed to match the real robot configuration and the positive direction of rotation for each actuator of the real arm, where the z-axis of the frames assigned to each joint aligns with their rotation axes, following the Right Hand Rule. Therefore, when at the default initial pose of the arm shown in Figure 1b, the z-axis of Joint 1 should align with the z-axis of the world frame, while the z-axis of Joint 2 to 4 should align with the y-axis of the world frame. Joint 5 only controls the opening and closing of the gripper, and it was only assigned a symbolic frame for the completeness of the DH table. Throughout the simulation in Task 1, the end effector was treated as a rod with no width and a length of 0.126m. The resulting DH table is shown in Table 1.



(a) Configuration Sketch of the OpenMANIPULATOR-X Arm with Frames Assigned

(b) Default Initial Pose of the OpenMANIPULATOR-X

Figure 1: Configurations of the OpenMANIPULATOR-X Robot Arm

Table 1: D-H Parameters

Joint	θ_i ($^{\circ}$)	α_i ($^{\circ}$)	a_i (m)	d_i (m)
1	0	0	0	0.077
2	$-\arctan\left(\frac{128}{24}\right)$	-90	0	0
3	$+\arctan\left(\frac{128}{24}\right)$	0	0.130	0
4	0	0	0.124	0
5	0	0	0.126	0

When it comes to the frames assigned to Joint 1 to 4, they all follow Craig's DH convention, where all the frames are put at the "head" of the links. For Joint 1, due to the special "stacking" configuration of Joint 1 and 2 and the fact that the motion of Joint 1 and 2 can be fully decoupled (will not affect each other), a tricky frame assigning technique was applied, which considered Joint 1 to be at the same height as Joint 2 (0.077m), to consist a 2 degrees of freedom (DOF) "shoulder" joint together with Joint 2. This height of 0.077m is reflected by the d_1 component in Table 1, indicating that the distance between links 1 and 0 (base) is 0.077m, measured along z wrt. the current frame. Special attention was paid to the α_2 component in Table 1, as it is directly related to the positive rotation directions of Joint 2 to 4. α_i reflects the angle between axes, measured about x wrt. the current frame, and here the angle between the rotation axes of Joints 2 and 1 is -90° , around the positive x-axis of Joint 1's frame. Under Craig's DH convention, it is hard to represent the length of Actuator 3, labelled as $L_{actuator3}$ in Figure 2, as this joint translation is not along the z-axis of the Joint 2 frame. Therefore, in Table 1, a_3 m which is the distance between the axes of

Joints 3 and 2, measured along x, is set to 0.130m to include the $L_{actuator3}$ components. Then the configuration between links 3 and 2 can be represented by θ_2 in Table 1, which is the original angle offset of Joint 2, measured about z wrt. the current frame. From the DH table view, the zero-offset position of Joint 2 should point towards the positive x direction in the world frame. Therefore, by looking at Figure 2 and considering the direction of the offset is opposite to the positive direction of rotation of Joint 2, it is clear that $\theta_2 = -\arctan(\frac{128}{24}) \approx -79.380^\circ$. For the same reason, from the DH table view, the zero-offset position of Joint 3 should point towards the dash line direction in Figure 2, and it is necessary to "drag" link 3 back to the horizontal default initial pose by the original angle offset of Joint 3 θ_3 in Table 1. However, this time the direction of the offset aligns with the positive direction of rotation of Joint 3, so $\theta_3 = +\arctan(\frac{128}{24}) \approx 79.380^\circ$. Finally, a_4 and a_5 , which represent the distances between the axes of Joint 4 and 3, and the axes of Joint 5 and 4, are simply equal to the length of the link between those axes, labelled with a_4 and a_5 in Figure 2.

From the practical view, the DH table in Table 1 is very accurate, providing precise end effector position calculation based on forward kinematics and matching the real-world frame very well.

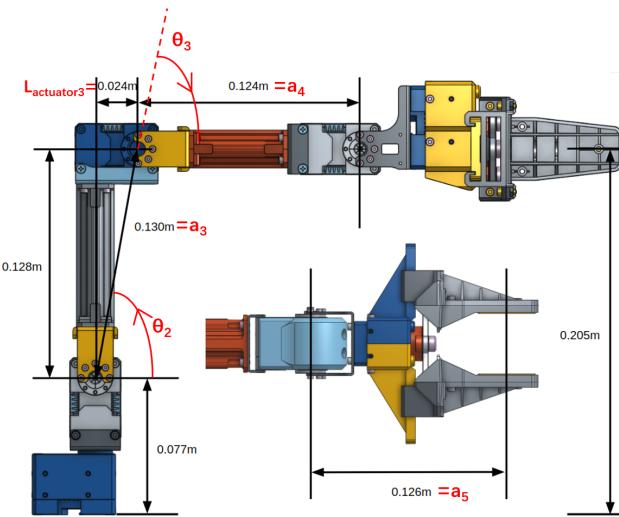


Figure 2: OpenMANIPULATOR-X Configuration Side View

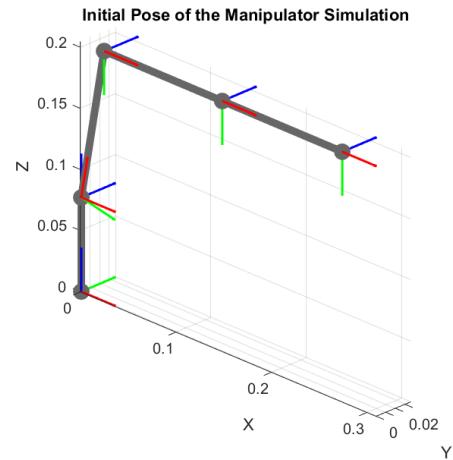


Figure 3: Initial Pose of the Manipulator Simulation

1.2 Forward Kinematics and Graphical Simulation of the Robot

As the DH table follows Craig's DH convention, the DH transformation matrix used in forward kinematics also follows Craig's convention:

$$T_i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Here, θ_i indicates each joint angles, and $c\theta_i = \cos(\theta_i)$, $s\theta_i = \sin(\theta_i)$. In this way, by adding the actuator rotating angles q_i of each joint with the original θ offsets defined in Table 1, the total joint angles can be easily determined. Then, after calculating the DH transformation matrix of each joint, starting with the identity matrix representing the base, for each joint i (from base to Joint 5), by multiplying the transformation matrix of the current joint by the previous cumulative transformation matrix, the robot's configuration can be easily determined. The cartesian position of the end effector in the world frame can be extracted from the first 3 components of the last column of the final cumulative transformation matrix after multiplying every joint, with respect to the x, y, and z coordinates, completing the forward kinematics. Furthermore, the cumulative matrix up to each joint was also extracted, and the first 3 components of the last column of those matrices exactly represent the x, y, and z coordinates of each joint. Hence, by connecting the real-time cartesian positions of each joint calculated by forward kinematics by `line()` objects in MATLAB, a

graphical simulation of the OpenMANIPULATOR-X robot arm was created. The simulation can be visualized in Figure 3, showing the default initial pose of the robot.

1.3 Trajectory Planning

After constructing the DH table and forward kinematics, an analytical inverse kinematics (IK) solution was also applied to the arm, shown in detail in [Inverse Kinematics](#) from [Appendix](#). The IK was very accurate, enabling 2 elegant trajectory planning strategies for the robot:

1. Cartesian Space Trajectory Planning: The cartesian space trajectory aims to plan a smooth, continuous and strictly straight-line trajectory between given starting and ending positions (represented by cartesian coordinates). Hence, quintic (fifth-degree) polynomial interpolations were used between the starting and ending positions in the form of: $p(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5$, with boundary conditions: $p(0) = p_0$ and $p(T_{\text{end}}) = p_{\text{end}}$ for the position; $\dot{p}(0) = 0$ and $\dot{p}(T_{\text{end}}) = 0$ for the velocity; and $\ddot{p}(0) = 0$ and $\ddot{p}(T_{\text{end}}) = 0$ for the acceleration. Using quintic interpolation guarantees not only smooth and continuous position and velocity variations across the whole trajectory but acceleration as well, which minimizes jerk and is better than cubic interpolation. Besides, quintic interpolation exactly ensures a straight-line trajectory, as the solution for each coordinate can be expressed as: $p_i(t) = p_{i,0} + f(t)(p_{i,1} - p_{i,0})$, $i \in \{x, y, z\}$, where $f(t)$ is a scalar function that varies from 0 at $t = 0$ to 1 at $t = T_{\text{end}}$. Since $f(t)$ only scales the difference $(p_{i,1} - p_{i,0})$, every point $p(t)$ lies on the straight line connecting $p(0)$ and $p(1)$. After quintic interpolation, IK is applied to all the interpolation points in the trajectory to output the corresponding joint angles to achieve these poses. In practice, the planned trajectory is discretized into a sequence of time steps containing the desired position at each instant, and the controller follows each step by position control.
2. Joint Space Trajectory Planning: Cartesian space trajectory planning ensures that the end-effector follows a straight line in the world frame, which is useful when accurate movements are required, like obstacle avoidance tasks. However, calculating IK for each interpolation point costs more computation power and time, and for trajectories covering a wide range without obstacles, it is usually enough to use joint space trajectories, which are faster to compute. Joint space trajectory planning only solves IK for the starting and ending positions, and the joint angles calculated are then used as the starting and ending joint angles of each joint. Hence, in the same way, by applying quintic interpolation between the starting and ending angles in the joint space, each joint has a specific trajectory with 0 angular velocity and angular acceleration boundary conditions and smooth and continuous angular velocity and angular acceleration variations. Every joint arrives at the ending position simultaneously. Thus the planned trajectory is also a "straight line" in joint space's view, but in cartesian space, it is a curve connecting the starting and ending positions.

As the IK solutions are usually not unique for a given position, continuity checks were applied to both cartesian space and joint space trajectories, where we calculate the difference between adjacent IK solutions and always choose the less different solution from the previous robot configuration.

The cartesian space trajectory and joint space trajectory planning are both demonstrated in our Task 1 simulation video, where cartesian space trajectories were used to plot squares of 10 x 10 cm in the XY, XZ and YZ cartesian planes, and joint space trajectories were used to plot a square in the XY plane, as an example.

2 Appendix

2.1 Inverse Kinematics

Our inverse kinematics (IK) aim to solve the joint angles of Joint 1 to 4, represented as q_1 to q_4 , according to given cartesian coordinates of the tip of the end effector, while treating the end effector as a rod with no width and a length of 0.126m. However, as we need to solve 4 joint angles for only 3 (q_x , q_y and q_z) coordinates, we need to assign an additional constraint to our system to achieve closed-form solutions. Hence, every time we calculate IK, we assign a required angle between the end effector and the horizontal direction, γ , to our system, as shown in Figure 5. This approach also helps with our practical trajectory planning, as we can manipulate the orientation in which the end effector interacts with the environment. For a given end effector position, calculating the joint angle of Joint 1, q_1 , is easy. As shown in Figure 4, in the x-y plane of the world frame, q_1 can be derived by:

$$q_1 = \arctan\left(\frac{p_y}{p_x}\right). \quad (1)$$

Besides, $p_r = \sqrt{p_x^2 + p_y^2}$ merged p_x and p_y onto the new r-axis.

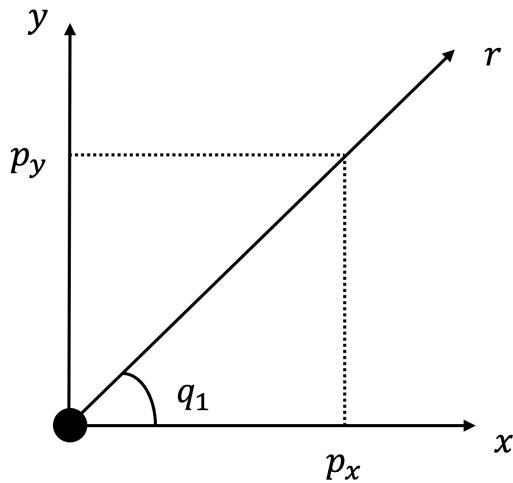


Figure 4

Figure 5 demonstrates the structure of the arm in the r-z plane of the world frame. According to the already known p_r , p_z , γ and q_1 , the p_{r4} and p_{z4} coordinates of Joint 4 can be easily derived by:

$$\begin{aligned} p_{r4} &= p_r - 0.126 \cdot \cos(\gamma) \\ p_{z4} &= p_z + 0.126 \cdot \sin(\gamma). \end{aligned}$$

Then we check the reachability of Joint 4 by checking whether $r \leq a_3 + a_4$ and $r \geq |a_3 - a_4|$, where $a_3 = 0.130$ and $a_4 = 0.124$ follow the definitions in Table 1. According to Figure 5, it is obvious that $L = \sqrt{p_{r4}^2 + (p_{z4} - 0.077)^2}$, and using cosine rule, we have $\beta = \pm \arccos\left(\frac{0.130^2 + 0.124^2 - L^2}{2 \times 0.130 \times 0.124}\right)$, corresponding to the two “elbow up” and “elbow down” solutions. Additionally, from Figure 2 it is clear that at the default initial pose, the angle between links 2 and 3 originally equals to $180^\circ - \theta_3 = 180^\circ - \arctan\left(\frac{128}{24}\right) = 100.620^\circ$. Therefore, q_3 can be derived by:

$$q_3 = 100.620^\circ - \beta. \quad (2)$$

Additionally, in Figure 5, φ can also be calculated by cosin rule: $\varphi = \arccos\left(\frac{0.130^2 + L^2 - 0.124^2}{2 \times 0.130 \times L}\right)$. Besides, δ in Figure 5 can be determined easily by the r-z coordinates of Joint 4: $\delta = \arctan\left(\frac{p_{r4}}{p_{z4} - 0.077}\right)$. Therefore, according to the

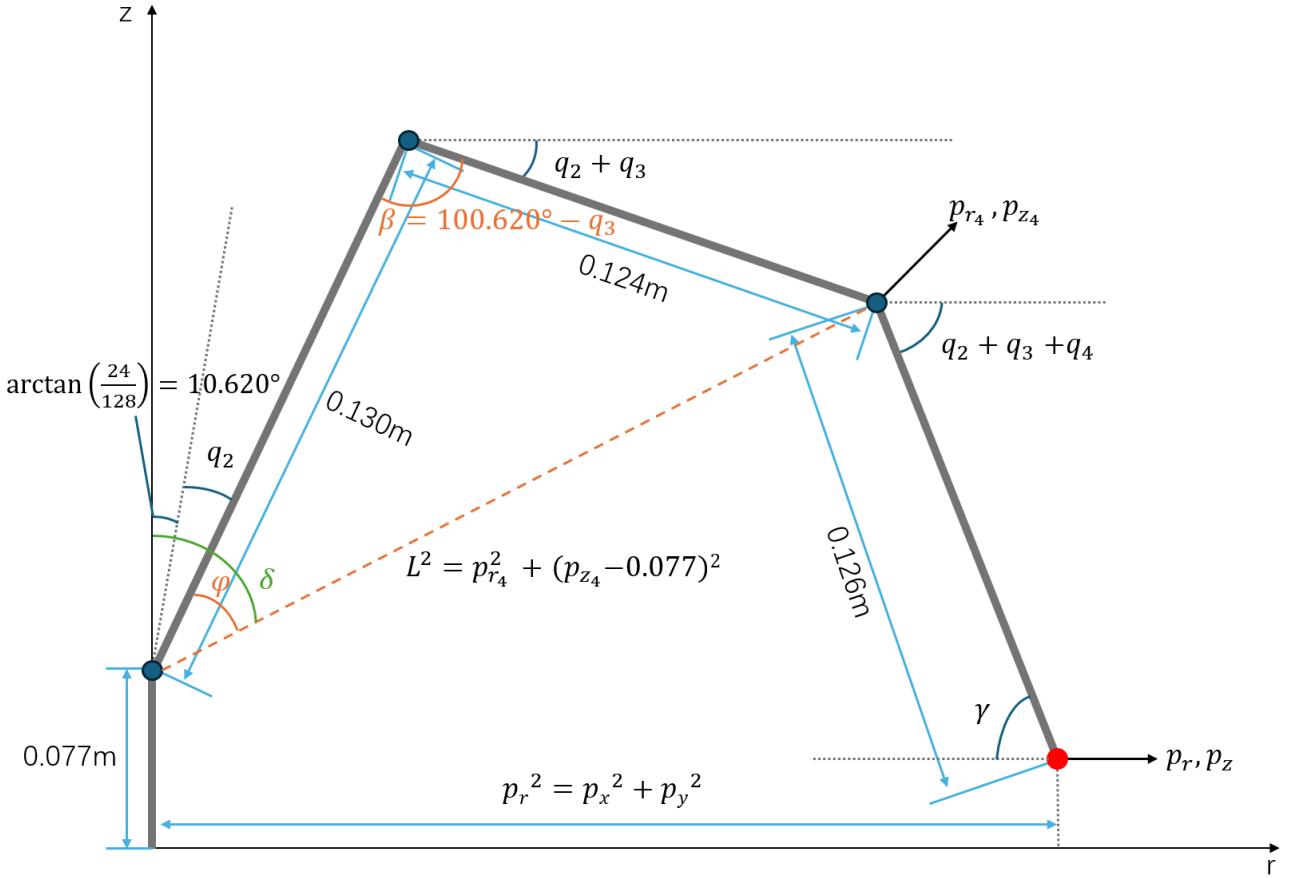


Figure 5: Configuration Sketch of the Arm in the r-z Plane of the World Frame

geometric relationships in Figure 5, q_2 can be derived by:

$$q_2 = \delta - \varphi - 10.620^\circ. \quad (3)$$

Finally, by noticing that the initial pose of the end effector is in the horizontal direction in the world frame, and its total deviation from the initial pose is caused by $q_2 + q_3 + q_4$, which equals the constant angle assigned between the end effector and the horizontal direction, γ . Therefore, q_4 can be derived by:

$$q_4 = \gamma - q_2 - q_3. \quad (4)$$

In conclusion, together, equations (1), (3), (2) and (4) form the analytical solutions to inverse kinematics of the system. After solving the equations, we review the z-coordinates of the end effector and Joints 3 and 4 to check whether they touch the ground. We finally discard the solutions that are lower than the ground (but allow them some tolerance considering that the links and end effector of the robot have their own thickness) or violate the joint limits assigned (depending on tasks).

2.2 Task 3 3D Printed Objects

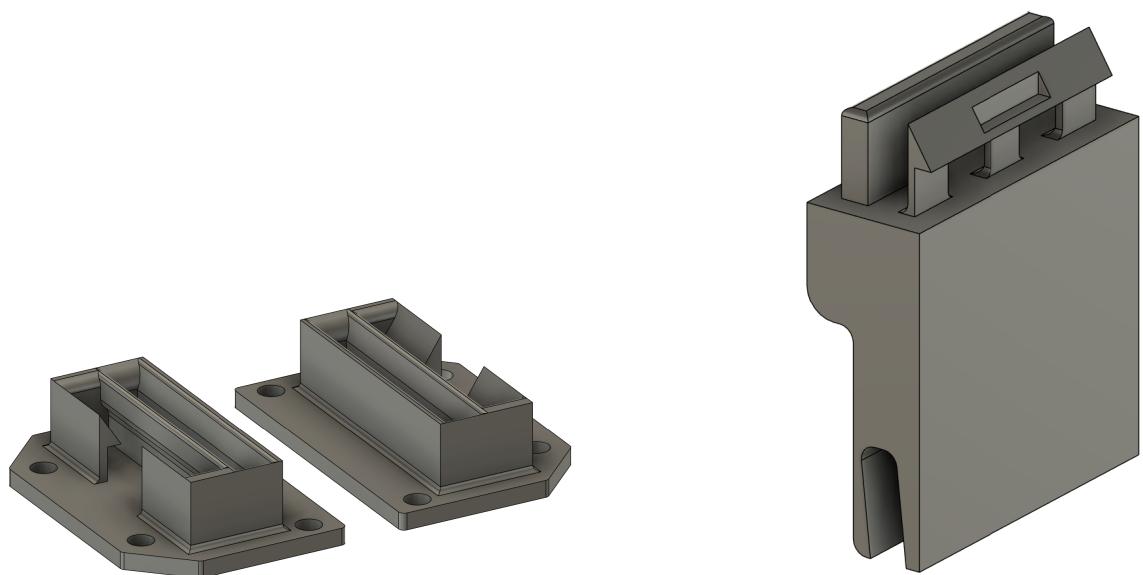


Figure 6: Changeable Gripper Base & First Gripper "Sweeper"

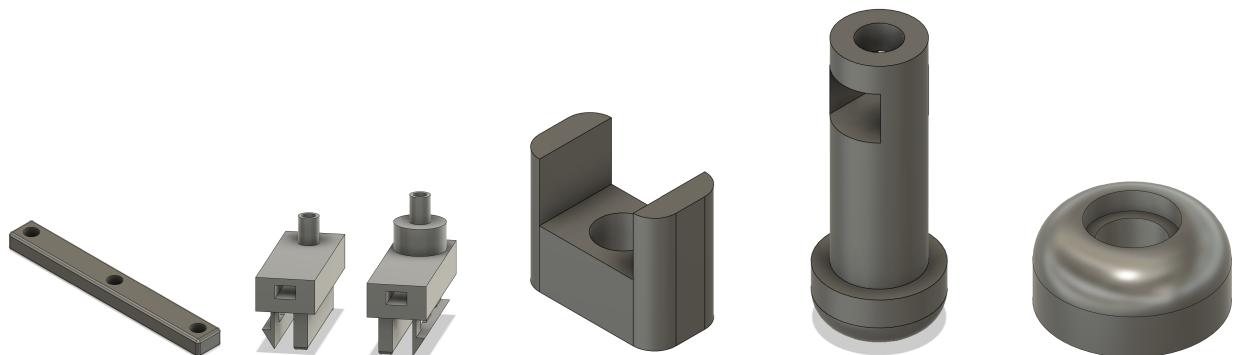


Figure 7: Second Gripper "Scissor" Main Structure

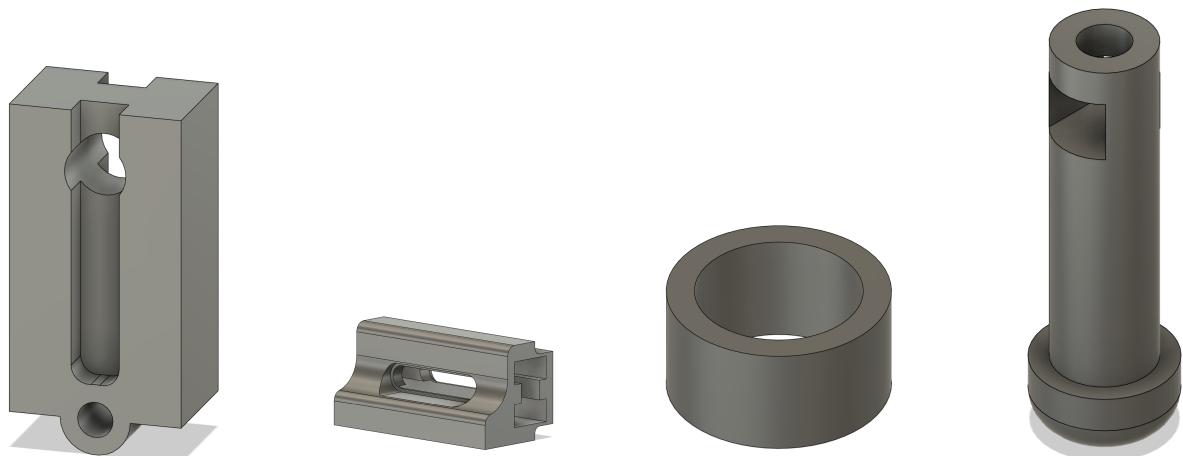


Figure 8: Second Gripper "Scissor" Grip Part Structure

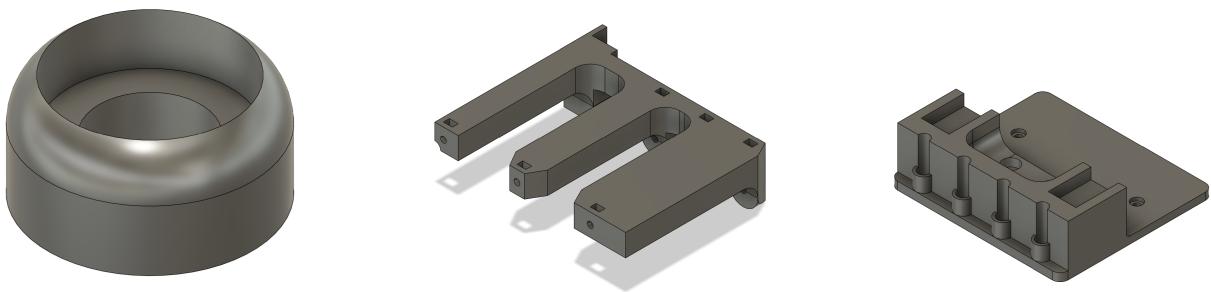


Figure 9: "Scissor" Gripper Holder from Acrylic Board Base



Figure 10: Module Designed Card Insert Holder & Gripper Removal Part

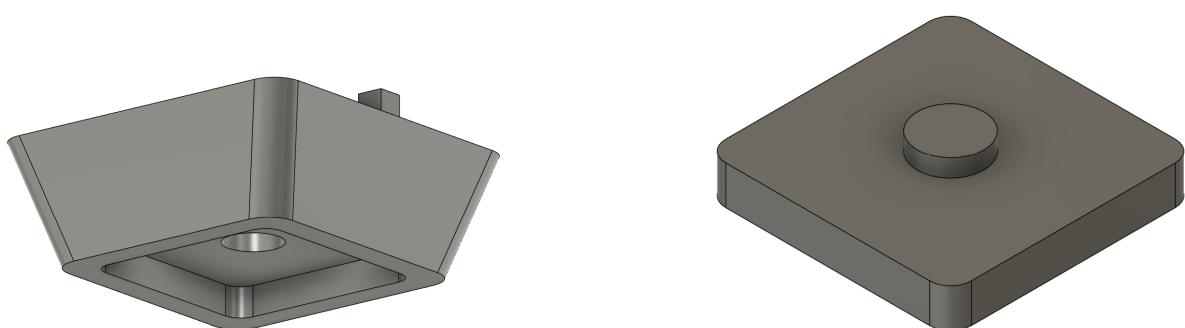


Figure 11: Acrylic Board Base Rising due to Mounted Modules. Base Pad TPU Mrinted for Damping



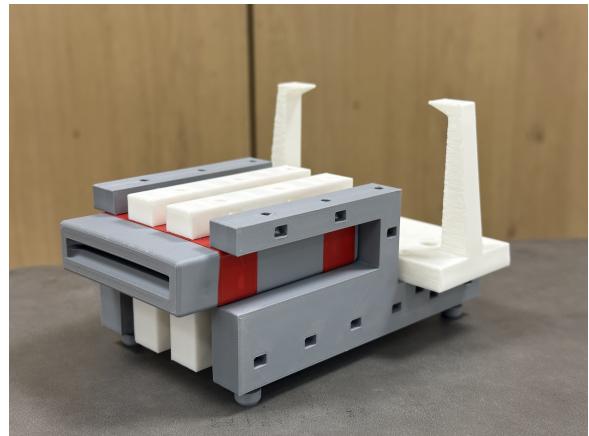
(a) "Scissor" holder station



(b) Two types of grippers used



(c) Acrylic board riser



(d) Gripper changing station & card holder

Figure 12: Assembled Configuration