Robotic Arm That's Here to Butter You Up

MAE C263A – Kinematics of Robotic Systems Final Project

Group Number 12

David Qiao

Eduardo Salazar

Emre Ustuner

Hyeongtaek Nam

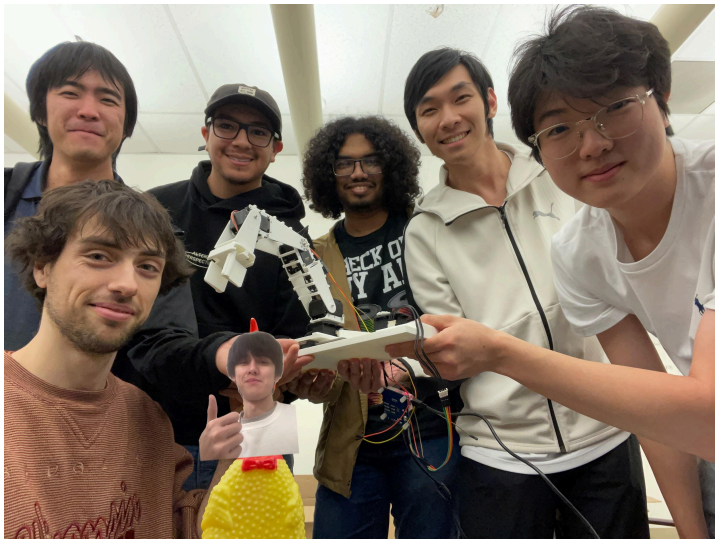
Sanskar Anil Nalkande

Trevor Oshiro

Yong-Ping Su

Instructor: Dr. Dennis Hong

December 7, 2024



1 Introduction

The purpose of this project is to design and control a robotic manipulator capable of performing a simple but precise task: passing the butter. The chosen design consists of a 5-DOF (Degrees of Freedom) robot arm that utilizes Dynamixel MX-28AR servo motors as actuators to move the arm in three-dimensional space with a servo driven gripper to grab objects. This robot is designed to meet the functional requirement of reaching out and passing butter within a constrained environment, with particular attention to design efficiency and kinematic accuracy.

The challenge of this project lies not only in the mechanical design and the selection of actuators but also in deriving and implementing the kinematic equations to allow precise movement of the arm. Forward and inverse kinematics are central to this task, as they will determine the positional control of the manipulator.

Through this project, we aim to explore the complexities of robotic manipulation, from theoretical concepts to practical implementation, all while keeping in mind the real-world task of butter passing.

2 Robot Design

2.1 Robot Structure

The manipulator we have designed is a 5-DOF robot arm. A 5-DOF configuration provides sufficient degrees of freedom to achieve the necessary range of motion to interact with objects at varying distances within a specified workspace. The robot consists of four revolute joints (controlling the arm) and a prismatic joint (controlling the base). All joints are powered by Dynamixel MX-28AR servo motors. The gripper/end effector is powered by an SG90s micro servo which is controlled by an Arduino uno. A CAD model of the robot can be seen in Figure 1.

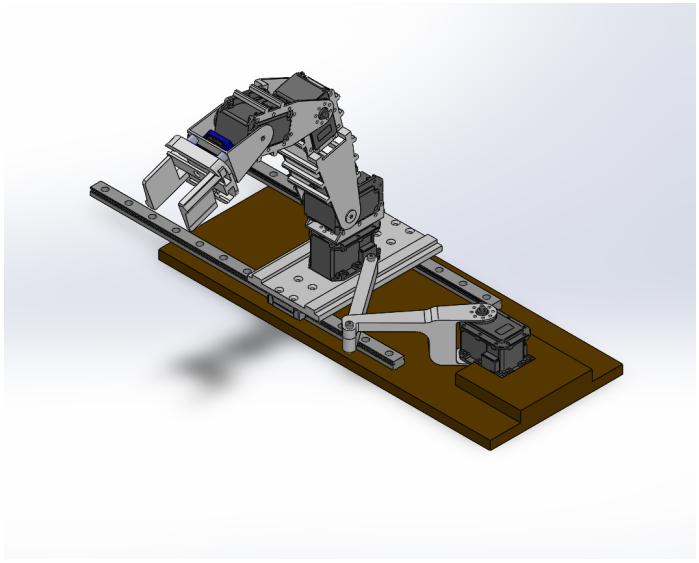


Figure 1: CAD model showing the configuration of robot arm

2.2 Workspace Analysis

In its initial design, the pass-butter robot's workspace was limited to a hemispherical region with a 30 cm radius (Figure 5). Within this area, the stationary robotic arm could pick up and deliver butter, ensuring accessibility and convenience for diners.

To overcome the limitations of this fixed workspace, a linear guide was integrated below the arm platform, enabling linear motion and significantly expanding the robot's operational range. This modification transformed the robot from stationary to mobile, allowing it to serve a larger area across the dining table.

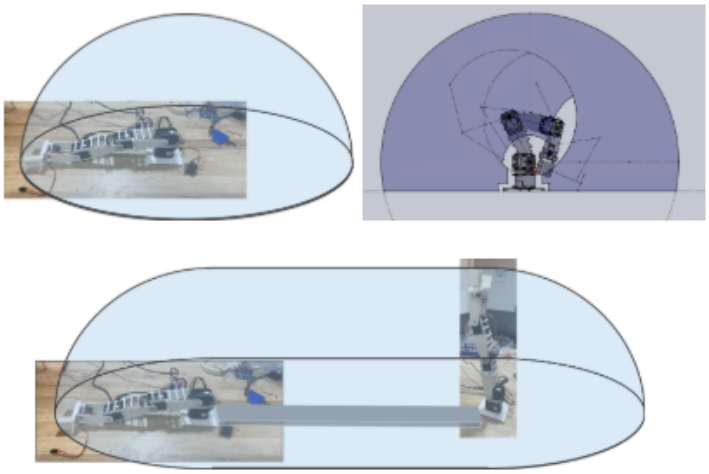


Figure 2: (Left) Ideal workspace of the robotic arm (Right) Actual workspace due to physical constraints on system (Note: Slice is taken for side of planned operation)

However, there are some areas we can't access due to the limitation of the mechanism as highlighted in the simlulated workspace in the figure above. In conclusion, this enhancement illustrates the potential for integrating linear motion systems into robotic designs, demonstrating how a relatively simple mechanical addition can significantly expand a robot's capabilities and adaptability in practical applications.

2.3 Link Length and Joint Configuration

Each joint in the manipulator is connected by a link, and the length of these links is constrained by the motor torque capabilities and physical interference considerations. The links needed to be long enough to allow the end-effector to reach the desired locations but not so long as to cause instability or exceed torque limits described by the motor's manufacturer. The joints are as follows (and are pointed out on the robot in Figure 4):

- Moving Base (Joint 1): Prismatic joint; increases reachable workspace.

- Pivot (Joint 2): Revolute joint; controls rotation in x-y plane.
- Shoulder (Joint 3): Revolute joint; controls ascent, descent and extension.
- Elbow (Joint 4): Revolute joint; controls ascent, descent and extension.
- Wrist (Joint 5): Revolute joint; orients manipulator perpendicular to surface.
- End-Effector: A gripper that can pick up butter, controlled by an arduino uno.

2.4 Design Challenges

The Gripper is controlled by a single motor. A rack and pinion mechanism is used to control the closing and opening of the gripper. One of the main issues we ran into was adjusting for printer tolerance, since 3D printing a gear required tighter tolerance than normal. We also had to increase the robustness of the overall design by bridging arm linkages and increasing the size of the baseplate, since it would move around too much during operation to perform consistently.

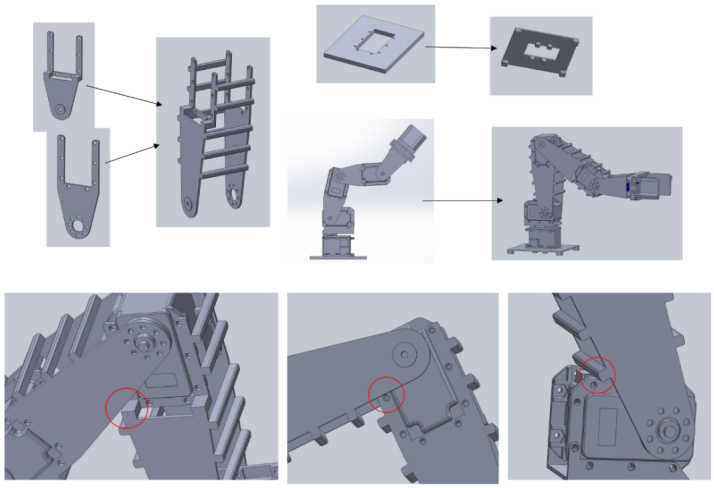


Figure 3: Bottom row illustrating physical constraints between joints. Top row displaying design updates to allow for a more robust system.

3 Static Force Analysis

To evaluate the structural integrity of our robotic arm under load, we conducted a static simulation using SolidWorks FEA. The purpose of this analysis was to examine how the weight of the target object, in this case, a piece of butter, affects the mechanical components of the robot arm. In the simulation setup, the base of the robotic arm was fixed to simulate a stable foundation. A downward force of 0.5 N, corresponding to the weight of a 50 g butter block, was applied to the gripper. The resulting stress distribution is illustrated in Figure 4.

The analysis revealed that the stress is primarily concentrated in the pivot located above motor 2 and below motor 3. However the point of highest deformation is still under the materials yield strength. This observation suggests that the current pivot design is robust enough for completing our task many times over.

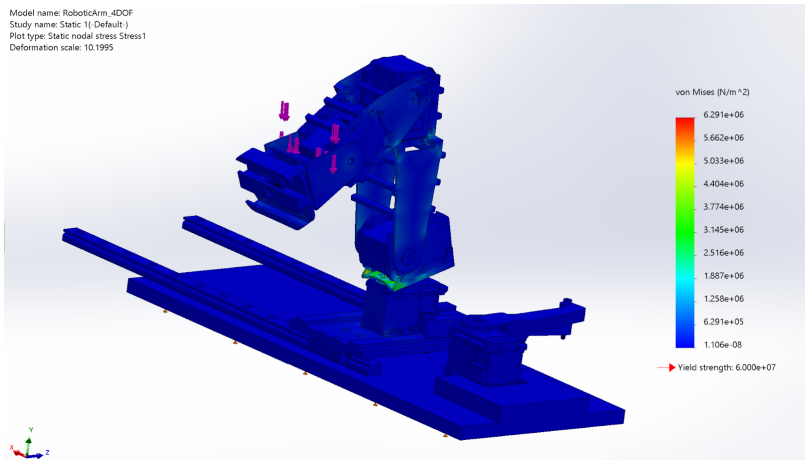


Figure 4: SOLIDWORKS Deformation Analysis

To ensure the motors had enough torque to pick up the butter and withstand its own weight we performed a static force analysis. The arm orientation shown in Figure 4 was chosen since this would result in the payload being furthest from the point at which the motor applies torque. All the motors used on the arm are identical, so the motor above the pivot was focused on since it would have to work against the weight of the entire arm. From this analysis it was determined that the robot could easily pick up a 50 g block of butter. Exact calculations are shown in the appendix.

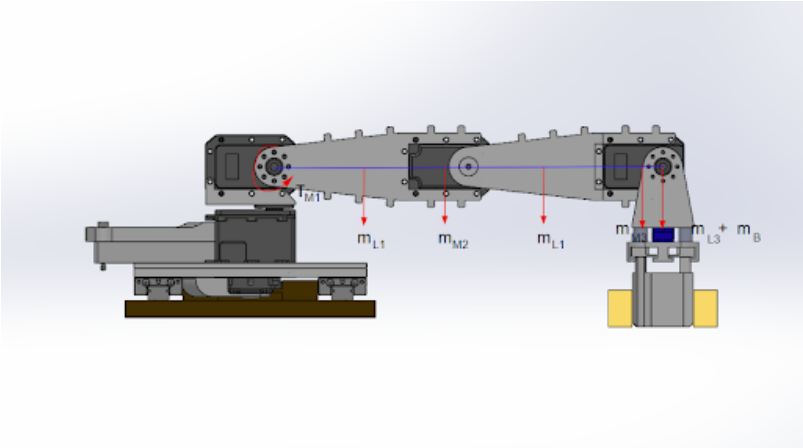


Figure 5: Force diagram for the report

4 Forward Kinematics

The Denavit-Hartenberg (DH) parameter is illustrated in the figure below. Notably, the z-axis of motor 3 is oriented in the opposite direction (rotated by 180°) compared to the z-axes of the motors positioned above and below it.

This deliberate rotation of motor 3 serves an important purpose: maintaining the symmetry of the robotic arm. By reversing the orientation of motor 3, the design ensures that any offsets are effectively canceled out, resulting in a configuration where the net offset is zero. This symmetry is crucial for preserving the arm's balance and enhancing its overall performance and functionality.

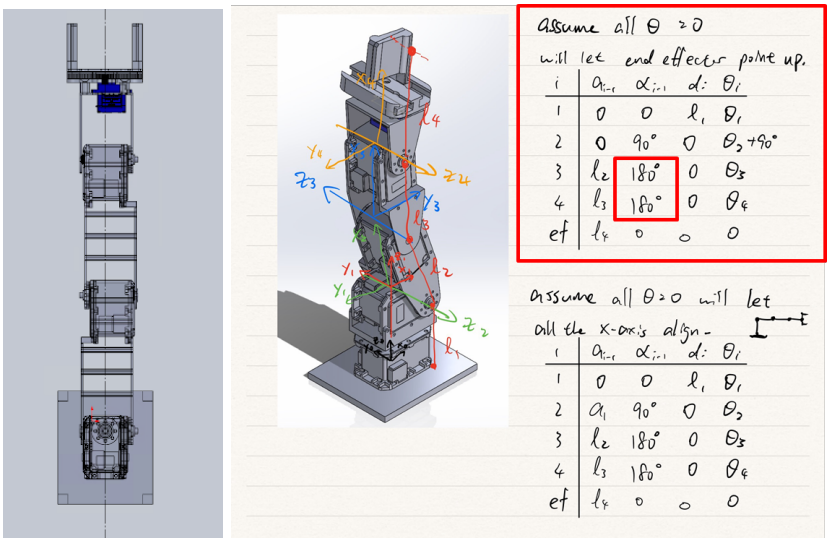


Figure 6: DH Table

5 Inverse kinematics

Inverse kinematics (IK) is the mathematical process of determining the joint angles of a robotic arm required to achieve a specific position and orientation of its end-effector.

The inverse kinematics solution for the robot arm is derived using a geometric method. The approach begins by taking the desired end-effector position ($p = [x, y, z]$) and orientation as input. The goal is to compute the joint angles $[\theta_1, \theta_2, \theta_3, \theta_4]$ that will position the end-effector at this desired location while satisfying the constraints of the robot's kinematic structure.

The steps for calculating the joint angles are as follows:

- Joint 1: To calculate θ_1, the desired end-effector position in the horizontal plane is projected onto a 2D plane. Using the polar coordinates of the projected point, t1 is calculated as the angle of rotation in the base plane.

$$\theta_1 = atan2(y, x) \tag{1}$$

  Where x and y are the projected end-effector coordinates in the plane.

- The second and third joints (θ_2 and θ_3) are responsible for positioning the arm in the vertical direction and controlling the reach of the arm. These joint angles are calculated using a combination of trigonometric equations that depend on the projection of the end-effector position in 3D space. Specifically, the law of cosines is applied to solve for the angles. θ_3 is calculated using the following equation derived from the geometry of the arm:

$$\cos(\theta_3) = \frac{r_4^2 - l_2^2 - l_3^2}{2l_2l_3} \tag{2}$$

Where r_4 is the distance from the base to frame 4. The value of θ_3 is then computed

$$\sin(\theta_3) = \sqrt{1 - \cos^2(\theta_3)} \tag{3}$$

$$\theta_3 = atan2(\sin(\theta_3),\, \cos(\theta_3)) \tag{4}$$

The second joint angle θ_2 is determined by combining the angles α and β, which are based on the geometry of the arm:

$$\alpha = atan2(z_4, r_4) \tag{5}$$

$$\cos(\beta) = \frac{r_4^2 - l_2^2 - l_3^2}{2l_2r_4} \tag{6}$$

$$\beta = atan2(\sin(\beta),\, \cos(\beta)) \tag{7}$$

Where, α is the angle that accounts for the vertical reach, and beta adjusts the position of the arm in the horizontal plane. The final value of θ_2 is adjusted for the configuration of the arm:

$$\theta_2 = \alpha + \beta - \frac{\pi}{2} \tag{8}$$

- The last joint θ_4 controls the orientation of the end-effector. Given that the wrist joint follows a simple rotational path, θ_4 is computed based on the required end-effector orientation and the sum of previous joint angles:

$$\theta_4 = \gamma - \theta_2 - \frac{\pi}{2} + \theta_3 \tag{9}$$

Here, γ is the required orientation angle of the end-effector.

The final joint angles $[\theta_1, \theta_2, \theta_3, \theta_4]$ are calculated by solving these equations sequentially, ensuring that each joint contributes to the overall position and orientation of the end-effector. The solution uses the forward kinematics model to validate the joint angles, ensuring that the calculated positions and orientations are correct.

6 Software Overview

We developed all of our software in MATLAB. We initialize an interface with the Dynamixel SDK using the example matlab code given to us. We use this structure to send joint angle commands to the 5 dynamixel motors in our robot, read real time motor angles feedback, and adjust settings of the motors. For our end effector gripper, we use a servo motor that is controlled separately by an Arduino board. We interface with the Arduino board in MATLAB by setting up another serial

port connection to the Arduino board. For our main control software, we developed two operating modes for the robot. We ask the user to input the mode (1 or 2) they like to use in the matlab's command window.

For mode 1, we set a predetermined set of end effector coordinates as our desired trajectory for the robot to follow, then we compute inverse kinematics and feed the resulting joint angles to the dynamixel sdk. We also decide which step should the end effector gripper close or open. At the same time, we read current motor angles from the dynamixel sdk, and use forward kinematics to plot the robot's real time configuration in top view and side view.

For mode 2, we have user manual control of the robot, where the user uses a keyboard to change the end effector's desired position, base slider's position, and open/close the end effector gripper. We have a key pressed and released callback function to monitor the user's keyboard activities. The end effector's desired position gets updated in the callback function, and then fed to the inverse kinematics to compute the joint angles that gets passed to the dynamixel sdk. We also plot the top view and side view of the robot in real time using the manually changed end effector position.

7 Singularity Discussion

We don't have a singularity.

8 Simulation

As the robot's chosen locations involve a specified orientation of the end-effector, the simulation also included an experimental approach for both determining the workspace, and testing the specified coordinates. Within this project, both parts of the simulation were executed through plots generated from MATLAB code. Each grasping trajectory of the robot was specified to have an orientation with the end-effector link oriented normal to the table surface, so incremented positions were iterated over through our inverse kinematics function to generate a plot for a reachable workspace for a specific orientation. As the robot also faced workspace constraints from physical constraints in the link shape, limits for each joint angle were also applied as criteria for determining if a point would be reachable.

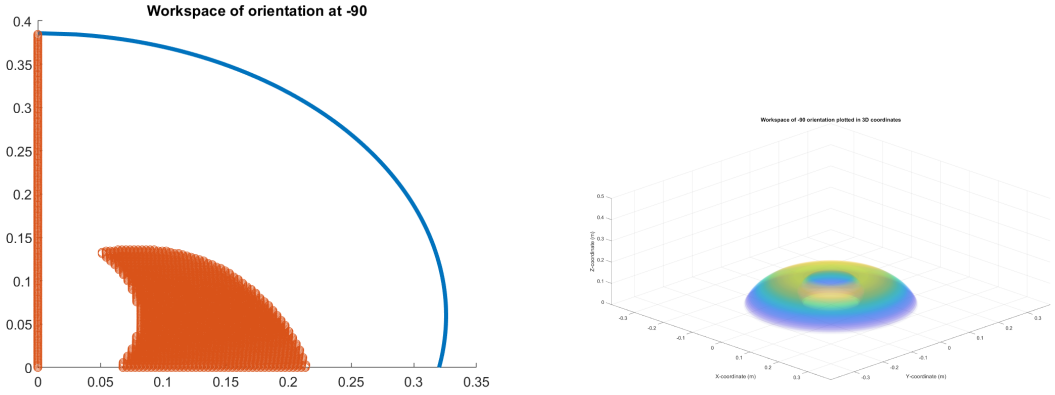


Figure 7: Reachable workspace (right) and boundaries (left)

As the full extent of the arm represents the maximum distance it can reach, the boundaries for the workspace were set with a circle (blue line in the plots) with a radius defined as the sum of the three links connected to the end effector, and the center was offset from zero by the offset value from the distance between the first two joints. As the first joint, aside from the prismatic joint, was the only actuator influencing the positions in the x-y plane, revolving the figures about the z-axis would give the full workspace in 3D.

9 Performance Analysis in Comparison with the Commercial Products

We compare our robot arm to the FANUC CRX-5iA Cobot. CRX-5iA is a robot arm with a 5kg payload and 994mm reach, which is the smallest robot arm in this company. A gripper could be added as its end-effector. The cost of this robot arm is approximately $30,000–$40,000. The robot excels in this task with its precise gripper control and integrated vision system, ensuring consistent performance and adapting to variations in the size or placement of the butter.

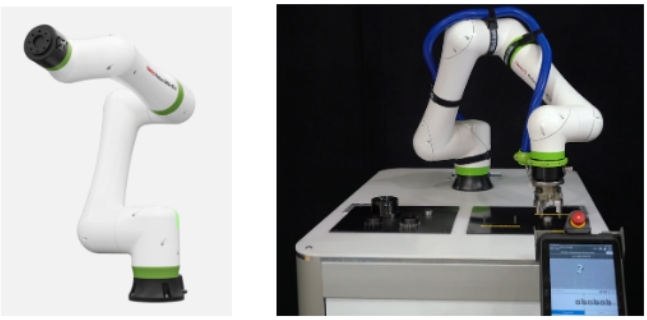


Figure 8: Simluated workspace cross-section

While our butter-passing robot arm is a cost-effective and versatile solution for prototyping and simple applications, the FANUC CRX-5iA offers better precision, reliability, and integration capabilities for industrial tasks. For a butter-picking task requiring consistent grip and placement accuracy, the FANUC CRX-5iA might be the superior choice, albeit at a significantly higher cost, with a fixed base without linear motion. The butter-passing robotic arm is an excellent alternative for environments where budget constraints or educational goals are a priority.

10 Conclusion

The Butter Robot project successfully demonstrates the potential of robotic manipulation for everyday tasks, such as passing the butter, using a 4-DOF manipulator powered by Dynamixel MX-28AR servo motors. Through careful design and iterative testing, we have developed a functional robotic arm that meets the primary objective of efficiently delivering butter across a table in a controlled environment.

The design process involved addressing key challenges such as selecting appropriate actuators, determining link lengths, and configuring joint angles to optimize the arm's range of motion and torque limits. We utilized a combination of revolute and prismatic joints to achieve a versatile range of movement, allowing the robot to reach various locations within its workspace.

The Butter Robot stands as an excellent solution for educational purposes and low-cost prototyping. Its simple design, ease of use, and adaptability make it an ideal choice for environments where cost and versatility are more important than absolute performance. Moving forward, the system could be further refined by enhancing the gripper design, improving motion control algorithms, and exploring additional mobility mechanisms.

In conclusion, the Butter Robot project showcases the feasibility of applying robotic systems to small-scale tasks, providing valuable insight into both the complexities and possibilities of robotic manipulation. As technology continues to advance, projects like this lay the foundation for the development of more sophisticated robotic systems capable of performing an increasing range of everyday tasks with efficiency and precision.

11 References

1. https://crx.fanucamerica.com/crx-5ia
2. https://github.com/thomashiemstra/fred
3. Course Module: MECH&AE C263A

12 Appendix

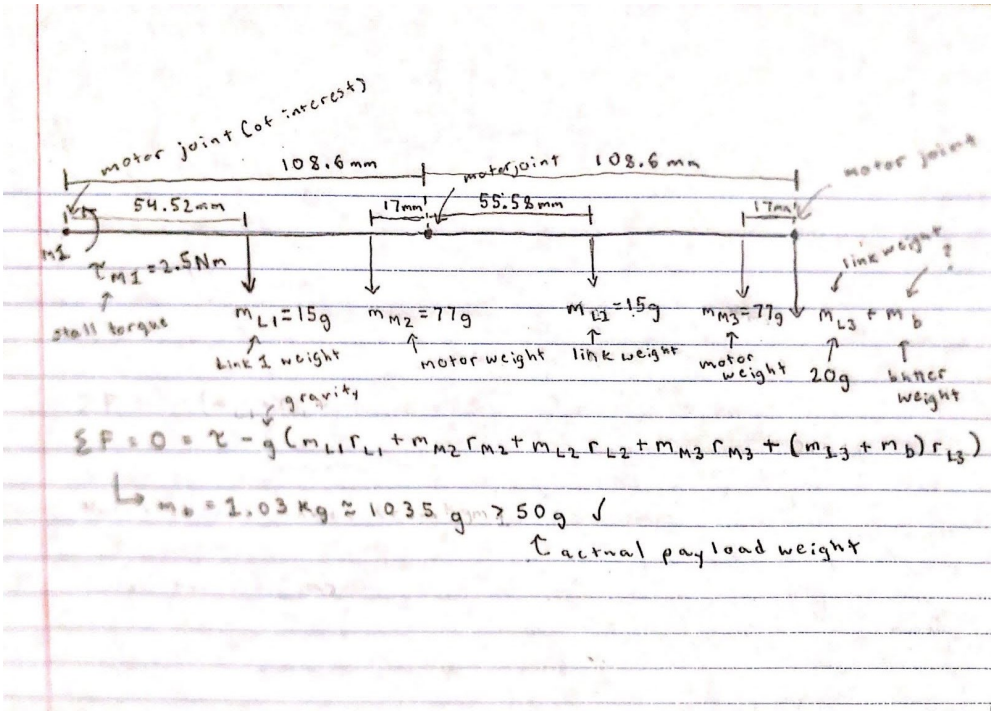


Figure 9: Static Force Diagram Analysisn