Technical Engineering Report: Thrust Stand Development and Control System Tuning

NAVAIR Fellowship Fall 2024
Virginia Tech
malhar05@vt.edu
colleen23@vt.edu
jackm23@vt.edu

December 2024

Contents

1 Executive Summary

This report documents the progress made during the initial phase of the project aimed at designing genetic algorithms (GAs) to automatically tune control systems for small multi-rotor unmanned aerial vehicles (UAVs). Conducted under the NAVAIR Fellowship at Virginia Tech, the project's primary focus during this semester was on foundational learning and the renovation of the thrust stand to prepare for future implementation of advanced control algorithms.

Project Objectives

The main objectives of the fall semester were:

- To gain an understanding of UAV flight dynamics, including rigid body dynamics and control theory.
- To recalibrate and renovate the existing thrust stand for accurate and reliable testing.
- To set up and integrate the necessary software environments, specifically ROS2 and the Dynamixel SDK (DynamixelSDK, n.d.), on a Linux-based system (Odroid).

Key Findings and Achievements

- **Flight Dynamics and Control Theory Education:** Under the mentorship of Dr. L'Afflitto, the team studied rigid body dynamics, Euler equations, and proportional-integral-derivative (PID) control systems. This educational foundation is critical for the future development of control algorithms.
- **Thrust Stand Renovation:** A significant portion of the semester was dedicated to identifying and resolving mechanical issues with the thrust stand. The team discovered that the high mass of the thrust stand resulted in excessive moment of inertia, causing failures in the Yaw axis motor. Through collaborative efforts, including torque testing and CAD modeling, the stand was redesigned to reduce mass and improve performance.
- **Software Setup and Integration:** Successfully established a ROS2 workspace on the Odroid Linux-based system. This involved learning to use Colcon build tools and integrating the ACSL Flightstack (Gramuglia, Mugundan, and L'Afflitto, n.d.).
- **Motor Control Implementation:** The team leveraged the C++ built Dynamixel SDK to control all three axis motors simultaneously. Implemented a startup routine to position all motors to a zero reference point upon system initialization, ensuring consistent operation.

Critical Issues Encountered and Resolutions

- **ROS2 Workspace Challenges:** Initial difficulties were encountered in setting up the ROS2 environment on the Odroid system due to unfamiliarity with Linux-based systems. The team overcame these challenges by extensive research, learning how to use Colcon build tools, and troubleshooting installation issues with support from the ACSL graduate students.
- **Yaw Axis Motor Failure:** The Yaw axis motor was failing due to an excessive moment of inertia from the thrust stand's mass. The team conducted torque tests and used CAD software to model the thrust stand and drone, determining that significant mass reduction was necessary. The redesigned thrust stand now allows for improved motor responsiveness and control dynamics.
- **Integration of Motors:** Achieving simultaneous control of all three motors required an in-depth understanding of the Dynamixel SDK and hierarchical motor positioning. Through dedicated effort, the team successfully implemented code to control multiple motors and zero them to a preset position. This capability lays groundwork for more complex testing scenarios in the future.

Summary of Achievements

While the genetic algorithm component of the project has not yet been developed, the groundwork laid during this semester is essential for its future implementation. Key achievements include:

- A preliminary understanding of UAV flight dynamics and control systems, crucial for developing advanced control algorithms.
- Renovation and recalibration of the thrust stand, resolving critical mechanical issues that hindered performance.
- Successful setup and integration of necessary software environments, including ROS2 and the Dynamixel SDK, on the Odroid system.
- Implementation of motor control systems capable of initializing and synchronizing all axis motors.

Individual Contributions

- **Colleen Piccolo**
  This semester Colleen's contributions to the project included the following:
  - Helping setup the ROS2 workspace on the Odroid.

- Developing the MATLAB scripts to calculate torque and to process the data gathered during the sinusoidal and step input testing of the thrust stand.
- Helping Luca look for new motors for the Yaw DOF of the thrust stand.
- Led the team's disassembly of the wire configuration in the thrust stand for the ISE lab to be able to cut holes in the stand.
- Worked on torque tests with Luca after initial tests. Helped Malhar with initial torque tests.
- Contributed to the team's mid-year report and mid-year presentation. Primarily worked on sections 5, 6, 7, 11, and 12 of the report. Built the Referencees.bib and references section.

• **Jack McGuire**
This semester Jack's contributions to the project were the following:

- Updating and maintaining an accurate thrust stand CAD assembly
- Obtaining an accurate matrix of inertia for the thrust stand by doing things like centering the thrust stand assembly at the origin to ensure accurate matrix of inertia calculations.
- Adding real-world mass data to CAD parts from manufacturer documentation
- Creating CAD interface for the motor torque test
- Contributing to the team's mid-year report and mid-year presentation.

• **Malhar Mahajan**
This semester Malhar's contributions to the project were the following:

- Set up the Dynamixel SDK workspace on the Odroid.
- Helped set up the ROS2 workspace on the Odroid.
- Created all motor testing scripts in C++ (e.g., zeroing, sinusoidal inputs, step inputs).
- Constructed the torque stand for motor characterization and ran initial torque tests.
- Researched and ordered parts (excluding the motor) for the lightened thrust stand.
- Compiled the mathematical equations for the thrust stand, integrating them into the slide deck and report.
- Created the initial skeleton/template for the report and worked on Sections 1, 2, 3, 7, and 8.

Future Work

Looking ahead, the team plans to:

- Reassemble the thrust stand.
- Begin the development of genetic algorithms for tuning PID and adaptive control systems for UAVs.
- Integrate the GA-based control algorithms with the thrust stand and UAV hardware for testing and validation.
- Continue studies in control theory and adaptive control systems under Dr. L'Afflitto's guidance.
- If time allows, conduct extensive testing to evaluate the performance improvements offered by GA-tuned control systems compared to traditional methods.

Conclusion

In conclusion, this semester's efforts have been pivotal in establishing the necessary foundation for the project's success. The team's focus on education, mechanical renovation, and software integration has addressed critical issues and set the stage for the development and implementation of genetic algorithms in UAV control systems on a redesigned thrust stand. The progress made provides a strong platform for achieving the project's ultimate goals in the upcoming semesters.

2 Introduction

- **Background of the NAVAIR fellowship and its significance:**
  The NAVAIR fellowship provides three well-qualified undergraduate students with the opportunity to experience research, collaborate with graduate students in an academically robust environment, and contribute to the success of the thrust stand project.
- **Structure of the report:**
  This report provides a detailed overview of the team's accomplishments from this semester. It starts with a breakdown of different components addressed this semester including: the Fundamentals of Control Systems, the Thrust Stand Mechanics, Software System Integration, and the Experimental Setup and Testing. The report concludes with the team's challenges they resolved as well as the results of the motor characterization.

3 Project Background

In the initial laboratory sessions, the team focused on fundamental control system concepts and their applications to UAV dynamics. This learning phase included theoretical discussions and practical derivations, laying a foundation for advanced control algorithm development. Key topics covered include:

3.1 Fundamentals of Control Systems

The fundamentals of control systems are essential for understanding and designing stable, responsive UAV systems. This section details the mathematical concepts and their applications to the thrust stand and UAV dynamics.

- **PID Control Derivation:**
  The Proportional-Integral-Derivative (PID) control equation was derived to enhance UAV stability and responsiveness. It is expressed as:

  $$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt},$$

  where:

  - $e(t) = r(t) - y(t)$: The error, defined as the difference between the desired setpoint $r(t)$ and the actual system output $y(t)$.
  - K_p, K_i, K_d: Proportional, integral, and derivative gains, respectively, which control system responsiveness and stability.

  The PID controller dynamically adjusts UAV control inputs based on real-time errors, past errors, and the rate of error change.

- **Dynamic and Inertial Matrices:**
  The positive-definite inertia matrix I plays a critical role in characterizing rotational dynamics, quantifying resistance to angular acceleration. The Jacobian matrix $J(\phi(t), \theta(t), \psi(t))$ relates Euler angle rates $(\dot{\phi}(t), \dot{\theta}(t), \dot{\psi}(t))$ to angular velocities $(\omega_x(t), \omega_y(t), \omega_z(t))$:

  $$\omega(t) = J(\phi(t), \theta(t), \psi(t))\dot{\Theta}(t), \quad \dot{\Theta}(t) = \begin{bmatrix} \dot{\phi}(t) \\ \dot{\theta}(t) \\ \dot{\psi}(t) \end{bmatrix}.$$

  To ensure numerical stability and avoid gimbal lock, the 3-2-1 rotation sequence was used for transformations between local and global reference frames.

- **Dynamic Equations:**
  The thrust stand's rotational dynamics are governed by Euler's equations of motion:

  $$\dot{\omega}(t) = I^{-1} \left[M(t) - \omega(t) \times I\omega(t) \right],$$

  where:

  - $M(t)$: Applied external moments as a function of time.

– $\omega(t)$: Angular velocity vector at time t.

This equation allows for the calculation of required torques to counteract inertial effects during UAV testing.

- **Matrix Rotations as Functions of Time:**
The team derived time-dependent rotation matrices for the x-, y-, and z-axes, which describe how UAV orientation changes dynamically:

$$R_x(t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x(t) & -\sin\theta_x(t) \\ 0 & \sin\theta_x(t) & \cos\theta_x(t) \end{bmatrix},$$

$$R_y(t) = \begin{bmatrix} \cos\theta_y(t) & 0 & \sin\theta_y(t) \\ 0 & 1 & 0 \\ -\sin\theta_y(t) & 0 & \cos\theta_y(t) \end{bmatrix},$$

$$R_z(t) = \begin{bmatrix} \cos\theta_z(t) & -\sin\theta_z(t) & 0 \\ \sin\theta_z(t) & \cos\theta_z(t) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Using these matrices, the UAV's global orientation at time t can be determined through the 3-2-1 sequence:

$$R(t) = R_z(t)R_y(t)R_x(t).$$

This sequence provides critical insight into the complex interplay between UAV dynamics and control inputs over time.

These mathematical principles, integrated with experimental setups like the thrust stand, enable precise characterization and testing of UAV control systems. They form the theoretical foundation for future development and implementation of advanced control algorithms.

3.2 Applied Analysis: Torque and Moment Calculations

A MATLAB script was developed to calculate the moments generated about the Pitch, Roll, and Yaw axes, incorporating inertia matrix effects and angular velocities. The angular velocity matrices were provided by Mattia and Giri. The script outputs insights into torque requirements under varying angular speeds and accelerations.

3.3 Insights from Torque Calculations

The MATLAB script provided critical insights into the dynamics of the UAV and thrust stand system:

- **Torque Scaling with Angular Velocities:** The analysis revealed that torque demands increase significantly with higher angular velocities, underscoring the necessity for accurate motor torque calculations to ensure UAV stability and control.
- **Inertial Property Optimization:** By examining the moments about different axes, the team identified the excessive inertial loads imposed by the thrust stand, particularly on the Yaw motor. This insight led to the redesign of the thrust stand to optimize its inertial properties and reduce torque demands.
- **Enhanced Design and Motor Compatibility:** The calculations directly informed the redesign of the thrust stand and motor selection process. The refined design ensured compatibility with the MX-106T motor's torque capacity, improving overall system performance and reliability.

4 Thrust Stand Mechanics

4.1 Thrust Stand Overview

The thrust stand is designed to mimic the drone's movements in three degrees of freedom (3DOF): Roll, Pitch, and Yaw. This setup allows for controlled testing of the UAV's control systems by simulating rotational movements without the risks associated with free-flight testing. The stand provides precise measurements of the drone's response to control inputs, which is essential for tuning and validating control algorithms.

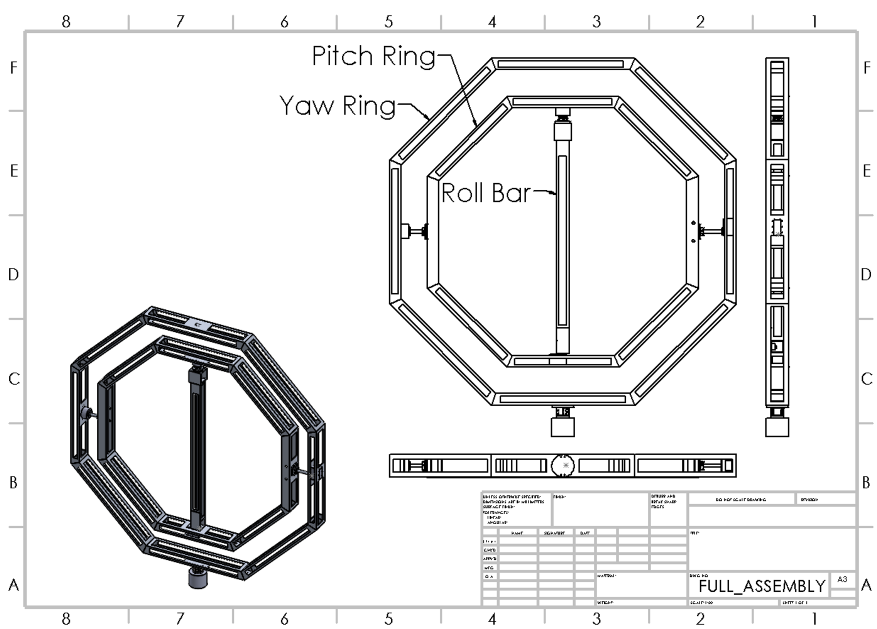


Figure 1: Thrust stand setup with 3 degrees of freedom.

The primary components of the thrust stand include:

- **Rotational Axes Rings:** Structures allowing rotation about the Roll, Pitch, and Yaw axes.
- **Actuation Motors:** Motors controlling each axis to induce the desired rotational movement. Includes internal sensors measuring angular positions, velocities, and accelerations for feedback.
- **Computing Unit:** An Odroid Linux-based system handling processing and control tasks.

4.2 Mechanical Design and CAD

The mechanical design was developed using SolidWorks CAD software, focusing on reducing the moment of inertia (Moment of Inertia) while maintaining structural integrity. The center of mass of the thrust stand was adjusted to be in the physical middle of the thrust stand. This allows for a more even mass distribution and thus a lower moment of inertia. All parts of the CAD were tuned to match real-world mass properties so that SolidWorks could accurately calculate the matrix of inertia. This design approach improves system responsiveness, reduces the risk of motor overloads, and supports precise control system adjustments. Additionally, to measure the matrix of inertia at any angle, mates were added to the thrust stand assembly, so that it moves inside SolidWorks the same way it does in the real-world.

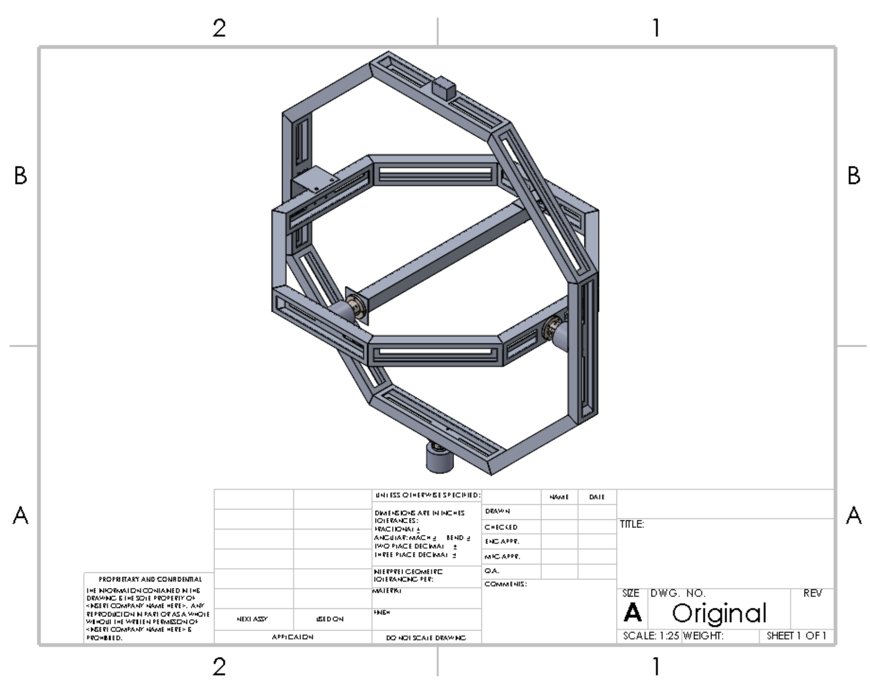


Figure 2: Initial CAD model of the thrust stand assembly.

4.2.1 Materials Selection and Structural Considerations

Materials were chosen based on a balance of strength, weight, and ease of manufacturing. Key design considerations included:

- **Weight Reduction:** The thrust stand was made of aluminum because of its lightweight yet strong properties. To further decrease mass while ensuring that the thrust stand was able to withstand operational stresses, the team performed finite element analysis simulations to strategically place holes in the Roll, Pitch, and Yaw rings. the team also used smaller gimbal rods and slip rings to further reduce the mass. Overall, the modifications reduced the thrust stand's mass by 28.7% bringing the mass from 19.67 to 14.02 kilograms.
- **Ease of Assembly:** The stand's components were designed for quick and simple assembly, enabling easy transport, setup, and reconfiguration when needed. The thrust stand breaks up into 3 main components: the Pitch ring, Yaw ring, and Roll bar.

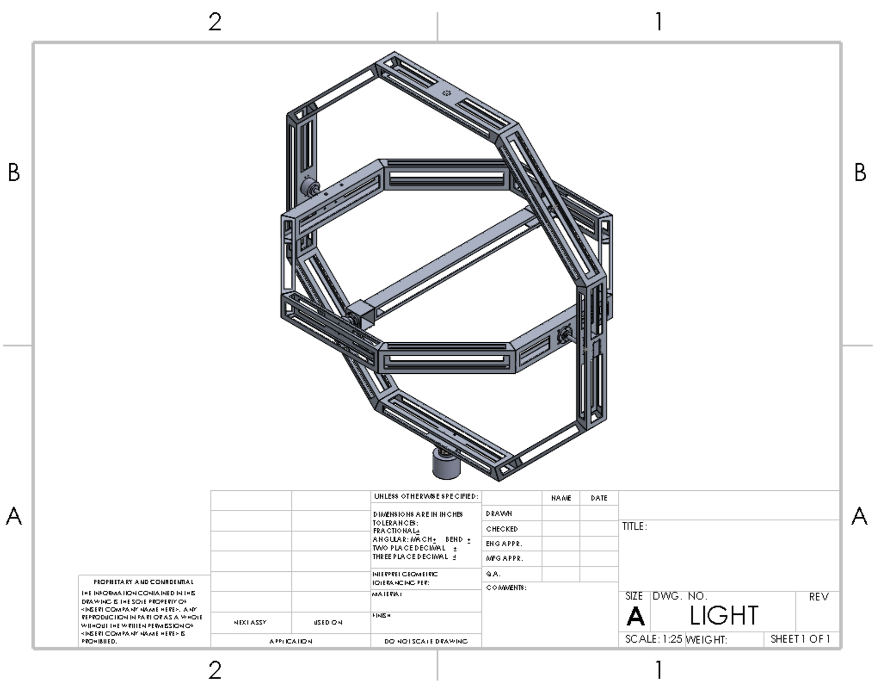


Figure 3: Lightened CAD model of the thrust stand assembly after mass-reducing modifications.

5 Motor and Gear Analysis

5.1 Selection of Motors

Initially, MX-106T servo motors were used for all three axes due to their high torque and compatibility with the Dynamixel SDK. Specifications for the MX-106T found on the Robotis website ("Dynamixel MX106-T", n.d.) include:

- **Stall Torque:** 8.4 N m
- **Input Voltage:** 10 – 14.8 V
- **No-load Speed:** 45 rpm

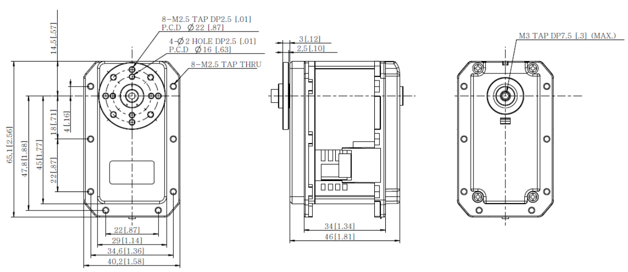


Figure 4: Drawing of MX-106T motor available from ("Dynamixel MX106-T", n.d.)

The MX-106T motor also weighs less than a pound-force ("Dynamixel MX106-T", n.d.), which makes them lightweight and easy to handle. The motor is equipped with an encoder ("Dynamixel MX106-T", n.d.), which allows it to know its position. Further motor options were explored when it was determined that the motor's Yaw axis would need to sustain a high torque that would possibly not work with the MX-106T's stall torque and continuous torque. Important things to consider when looking at the motors included their required operating voltage, their weight, whether they were AC or DC, and whether they had supporting software. Resources utilized include the DigiKey website and assistance from Luca Nanu, the doctorate student helping the team with the project. The total cost includes the estimated cost of a driver to support the motor. Below find a table of the motors that were discovered during this research phase. Note the link for the driver for LRPX32 Brushless was not found in the Excel sheet, but the cell's value was set to $175.00 so it was added to the total price of the motor which can be found through the link:

Motor Name	Total Cost (USD)	Hyperlinks
PH42-020-S300-R	$1649.90	Dynamixel
YM070-210-R051-RH	$2,389.90	Dynamixel
24V Planetary DC Gear Motor (22mm)	Unknown	Motor
LRPX32 Brushless DC Planetary Gear Motor	$872.00	Motor

Initially, the YM070-210-R051-RH motor was considered. However, due to its expensive price other options were explored. Ultimately, it was determined that the best course of action is to remove mass from the thrust stand, which subsequently will reduce the thrust stand's Moment of Inertia about the Yaw axis and therefore the torque. A replacement gear set was ordered for the MX-106T Yaw axis motor because the gears on it were damaged. The gearset cost the project $127.70, which is significantly less expensive and less risky than spending funds on a new motor.

5.2 Torque Testing and Moment of Inertia Calculations

It was observed that the Yaw axis motor failed under the load due to a high Moment of Inertia. To investigate, the team conducted torque tests using a makeshift torque stand to see what torque the motor could handle and calculated the Moment of Inertia using SolidWorks.

Torque Stand Test Results The torque stand allowed the team to measure the actual torque output under various loads. Data from test trials in the clockwise direction showed the actual torque output is less than the output as given in the MX-106T datasheet ("MX-106T/R Datasheet", n.d.).

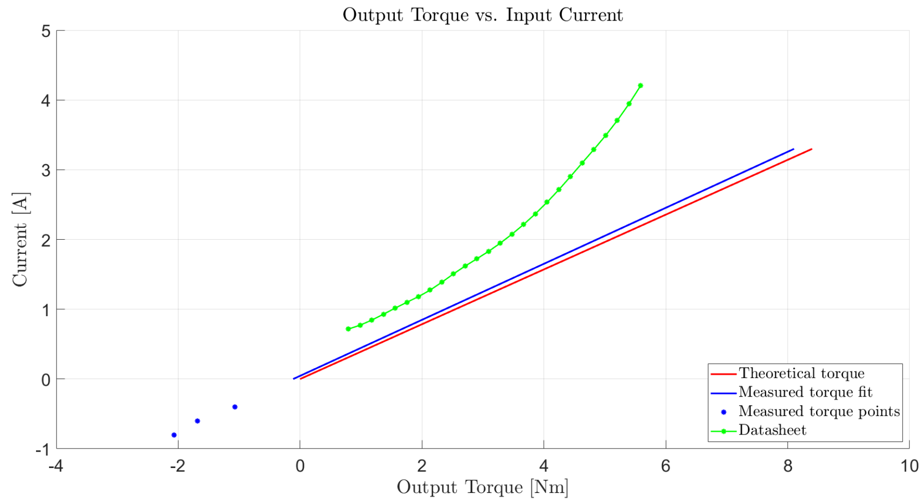


Figure 5: Calculated Torque trend line compared to MX-106T datasheet

Initial Moment of Inertia Calculations The calculated moments of inertia on the thrust stand prior to modifications, including prior to the correction of the material properties, were the following (these were taken at 90 degrees). The Yaw Matrix definitely includes the drone in the CAD but it is unclear whether it is included for the Pitch and Roll Matrices. The units of these matrices are kg·m^2. The torques can be calculated using the MATLAB file titled "Old_Moment_Inertia_Calcs.m":

Yaw Matrix

$$\begin{bmatrix} 3.74 & 0.00 & 0.00 \\ 0.00 & 2.37 & 0.00 \\ 0.00 & 0.00 & 3.05 \end{bmatrix}$$

Pitch Matrix

$$\begin{bmatrix} 0.84 & 0.00 & 0.00 \\ 0.00 & 0.81 & 0.00 \\ 0.00 & 0.00 & 1.61 \end{bmatrix}$$

Roll Matrix

$$\begin{bmatrix} 0.02 & 0.00 & 0.00 \\ 0.00 & 0.18 & 0.00 \\ 0.00 & 0.00 & 0.18 \end{bmatrix}$$

- **Roll Axis Torque:** 0.1000 N·m
- **Pitch Axis Torque:** 4.0500 N·m
- **Yaw Axis Torque:** 7.6250 N·m

The high Yaw Axis Torque in these initial rudimentary calculations prompted the team to start exploring options to reduce it or get a stronger motor. After modifications, these pre-modification values were recalculated because the initial CAD model they were performed on did not have accurate material properties. Once the material properties had been updated, the inertia matrices were taken at 90 degrees, included the CAD model of the drone, and the ones used for this calculation, again with units of kg·m^2, are:

Yaw Matrix

$$\begin{bmatrix} 4.14 & 0.01 & 0.02 \\ 0.01 & 3.76 & -0.01 \\ 0.02 & -0.01 & 2.50 \end{bmatrix}$$

$$\begin{bmatrix} 2.50 & -0.01 & 0.02 \\ -0.01 & 3.76 & 0.01 \\ 0.02 & 0.01 & 4.14 \end{bmatrix}$$

Pitch Matrix

$$\begin{bmatrix} 0.74 & 0.01 & 0.00 \\ 0.01 & 1.70 & 0.00 \\ 0.00 & 0.00 & 1.12 \end{bmatrix}$$

$$\begin{bmatrix} 1.12 & 0.00 & 0.00 \\ 0.00 & 1.70 & 0.01 \\ 0.00 & 0.01 & 0.74 \end{bmatrix}$$

Roll Matrix

$$\begin{bmatrix} 0.09 & 0.02 & 0.00 \\ 0.02 & 0.34 & 0.00 \\ 0.00 & 0.00 & 0.39 \end{bmatrix}$$

$$\begin{bmatrix} 0.39 & 0.00 & 0.00 \\ 0.00 & 0.34 & 0.02 \\ 0.00 & 0.02 & 0.09 \end{bmatrix}$$

Please note that the x and z axes were initially flipped with respect to Roll and Yaw due to the axis becoming inverted during the CAD modeling process after being corrected. The matrices were then changed to match the initial orientation of the inertia matrices. Below are the recalculated initial values using the above, corrected inertia matrices, using the MATLAB file titled "Old_Updated_Moment_Inertia_Corrected_Axis.m":

- **Roll Axis Torque:** 1.9500 N·m
- **Pitch Axis Torque:** 8.5002 N·m
- **Yaw Axis Torque:** 10.5889 N·m

These Torque values were calculated with only one axis moving at a time. For example, the Torque of the Yaw axis is what the motor for Yaw experiences only when the Yaw axis is moving. The high Moment of Inertia value in the Yaw axis contributed to the motor's inability to control the Yaw axis effectively.

5.3 Resolution of Motor Issues

To reduce the Moment of Inertia and resolve the motor failure, the following steps were taken:

1. **Redesigning Components:** Added weight-reducing holes to the rings.

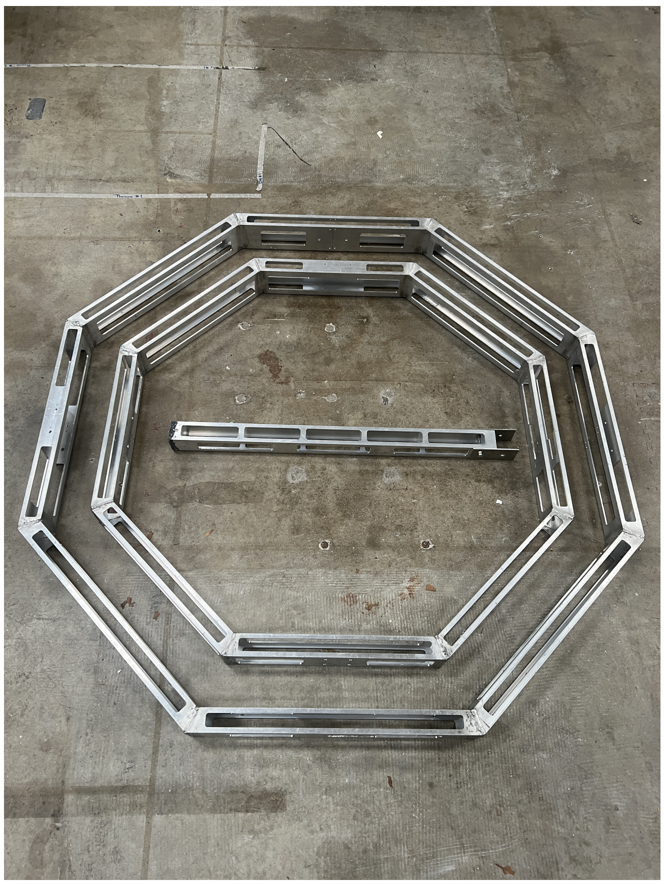
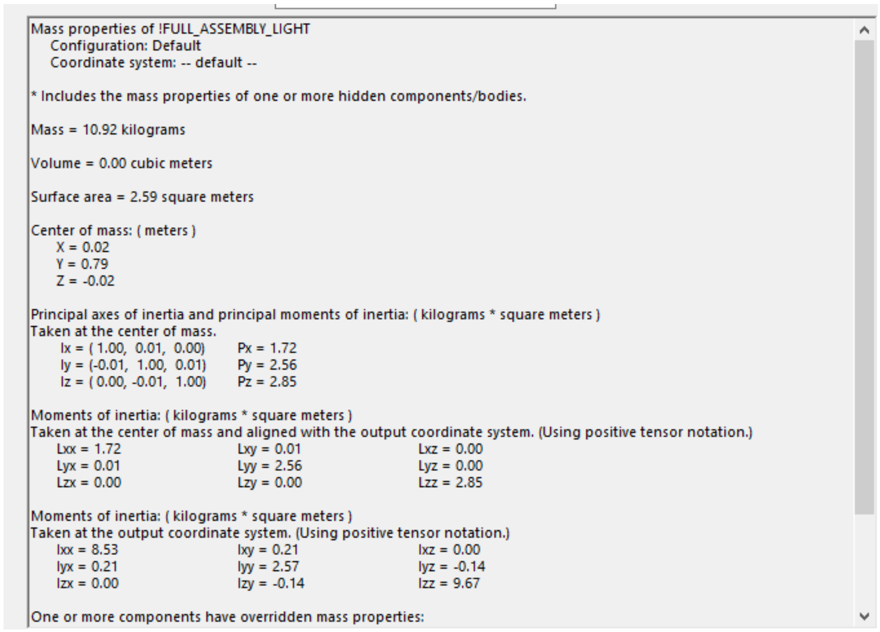


Figure 6: Modified rings with weight-reducing holes.

2. **Structural Integrity:** Holes were strategically placed to ensure structural integrity was maintained despite material removal.

3. **Updated Moment of Inertia Calculations:** The new Yaw Axis Torque has reduced noticeably to 7.1250 N·m, which means the modifications fulfilled their intention. The below image was taken at 90 degrees.



Mass properties of !FULL_ASSEMBLY_LIGHT
Configuration: Default
Coordinate system: -- default --

* Includes the mass properties of one or more hidden components/bodies.

Mass = 10.92 kilograms

Volume = 0.00 cubic meters

Surface area = 2.59 square meters

Center of mass: (meters)
X = 0.02
Y = 0.79
Z = -0.02

Principal axes of inertia and principal moments of inertia: (kilograms * square meters)
Taken at the center of mass.
Ix = (1.00, 0.01, 0.00) Px = 1.72
Iy = (-0.01, 1.00, 0.01) Py = 2.56
Iz = (0.00, -0.01, 1.00) Pz = 2.85

Moments of inertia: (kilograms * square meters)
Taken at the center of mass and aligned with the output coordinate system. (Using positive tensor notation.)

Lxx = 1.72	Lxy = 0.01	Lxz = 0.00
Lyx = 0.01	Lyy = 2.56	Lyz = 0.00
Lzx = 0.00	Lzy = 0.00	Lzz = 2.85

Moments of inertia: (kilograms * square meters)
Taken at the output coordinate system. (Using positive tensor notation.)

Ixx = 8.53	Ixy = 0.21	Ixz = 0.00
Iyx = 0.21	Iyy = 2.57	Iyz = -0.14
Izx = 0.00	Izy = -0.14	Izz = 9.67

One or more components have overridden mass properties:

Figure 7: Updated Moment of Inertia values from SolidWorks after modifications.

4. **Re-testing Motors:** The motors were tested again with the reduced Torque, and the Yaw motor operated without stalling.

5.4 Final Motor Selection and Justification

The decision to continue using the MX-106T motors was based on:

- **Lightened Structure:** The motor may function better with the lightened thrust stand.
- **Compatibility:** Existing integration with the Dynamixel SDK and control systems.
- **Cost-effectiveness:** Eliminate the need for the purchase of new motors and instead spend it on gears.

5.5 Section Remarks

By addressing the excessive Moment of Inertia through mechanical redesign, the team proactively addressed the motor issue related to the moment of inertia. The introduction of weight-reducing holes in the structural components decreased the load on the motors. This optimization not only prevented motor failures but also enhanced the responsiveness and accuracy of the control system, providing a reliable platform for future control algorithm development and testing.

6 Software and System Integration

6.1 ROS2 Workspace Setup

- **Installation and configuration of ROS2 on Odroid:**
  Mattia and Giri installed the OS onto the Odroid for the undergrads to use. The first skill developed while working with the Odroid was learning how to SSH into the Odroid. Once that was completed, the ACSL Flightstack (Gramuglia, Mugundan, and L'Afflitto, n.d.) was cloned and installed on the Odroid. Then, the ROS2 workspace was built on the Odroid referencing the instruction manual provided to the team. The team encountered and overcame difficulties in setting up the ROS2 workspace.

6.2 Dynamixel Workspace Configuration

- **Using Dynamixel Wizard:**
  Before using the Dynamixel SDK, the team used the Dynamixel Wizard, available on the Robotis website ("Dynamixel MX106-T", n.d.). This Wizard has an easy to use User Interface and provides several ways to control the motors including current, position, and velocity control. It also provides direct access to write to different registers on the motors.
- **Setting up Dynamixel SDK in C++ on Odroid:**
  Utilizing a Github repository found by Giri, the team was able to install Robotis' Dynamixel Software Development Kit (SDK) on the Odroid (DynamixelSDK, n.d.). The SDK proved to be a valuable resource for the team containing example code used in reference for development of the motor control scripts as well as providing functions with ease of access to control the motors.
- **Testing motor control: Moving individual and multiple motors.**
  The motors were first tested with the Dynamixel Wizard. Then, they were tested using scripts developed leveraging the Dynamixel SDK. During testing with the Dynamixel SDK, graphs were generated to observe the lag across different axis. When these trials were conducted, only the degree of freedom of interest was prompted to move. Note that the lag for the Yaw axis is noticeably problematic, likely due to the high moment of inertia. The plots shown below were generated using the MATLAB file titled "DataProcessing.m." Additionally, the step input graphs shown below were generated using the MATLAB file now titled "StepFunction.m." These plots were generated using data from October 20, 2024.

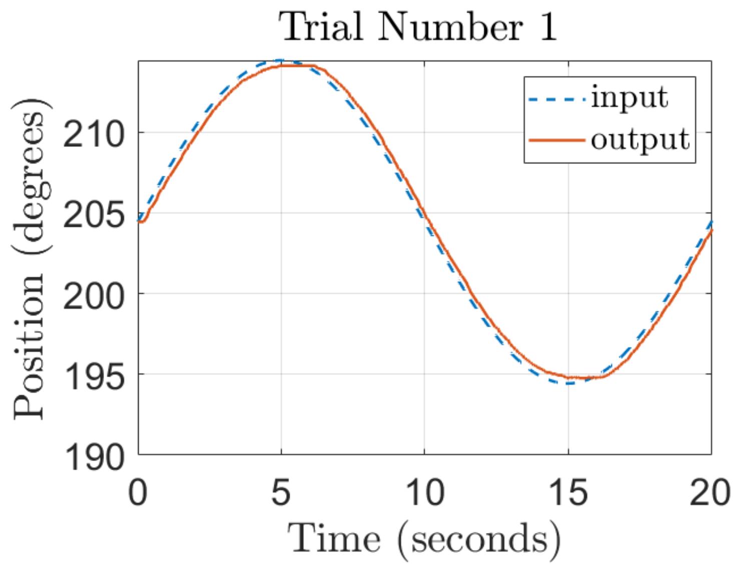


Figure 8: Roll Axis Lag using Dynamixel SDK

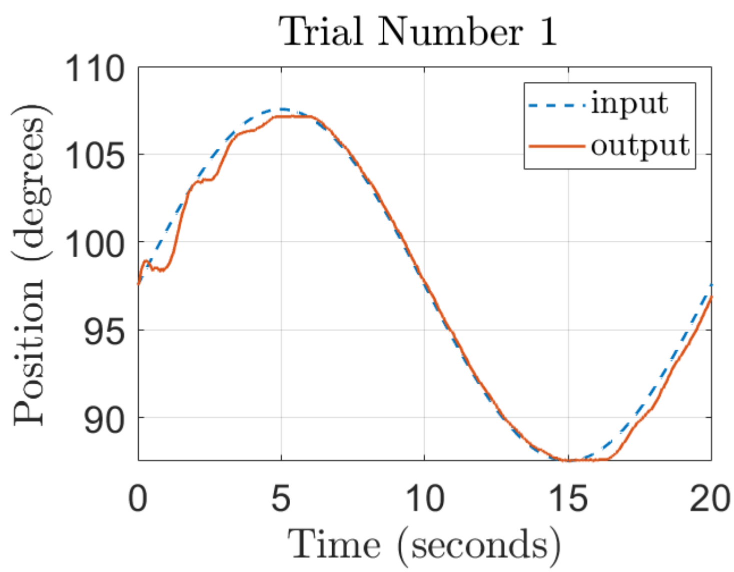


Figure 9: Pitch Axis Lag using Dynamixel SDK

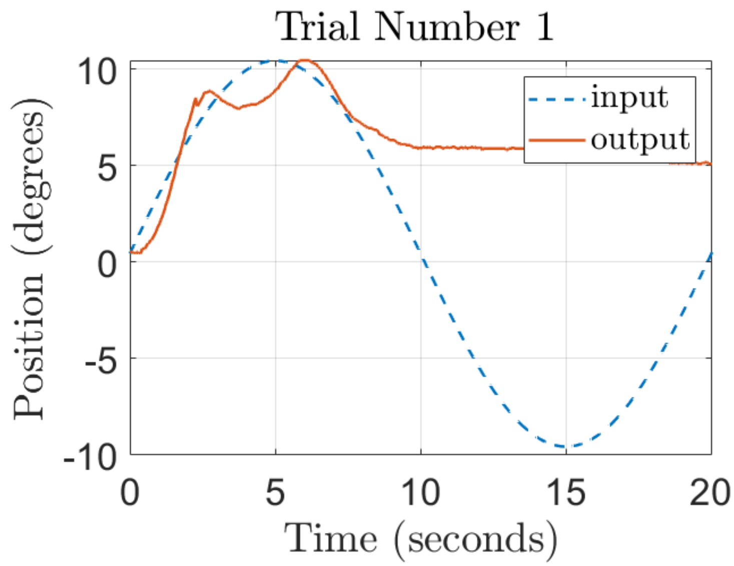


Figure 10: Yaw Axis Lag using Dynamixel SDK

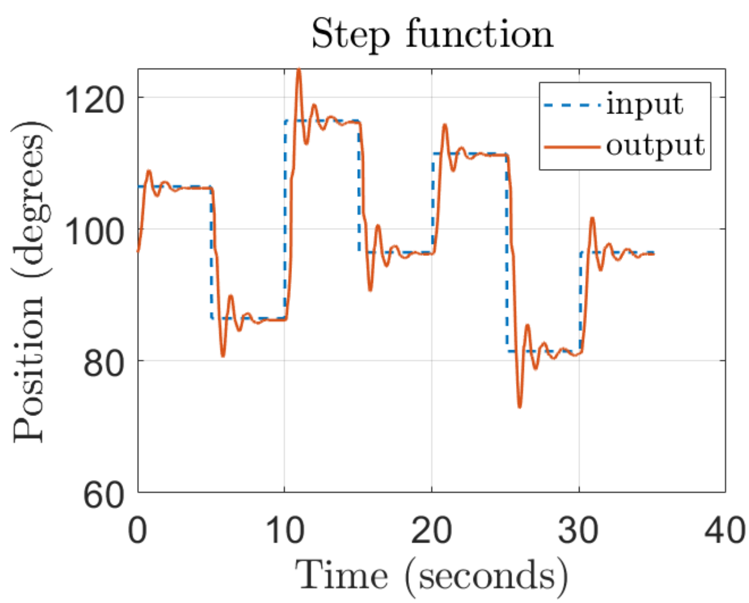


Figure 11: Pitch Step Response

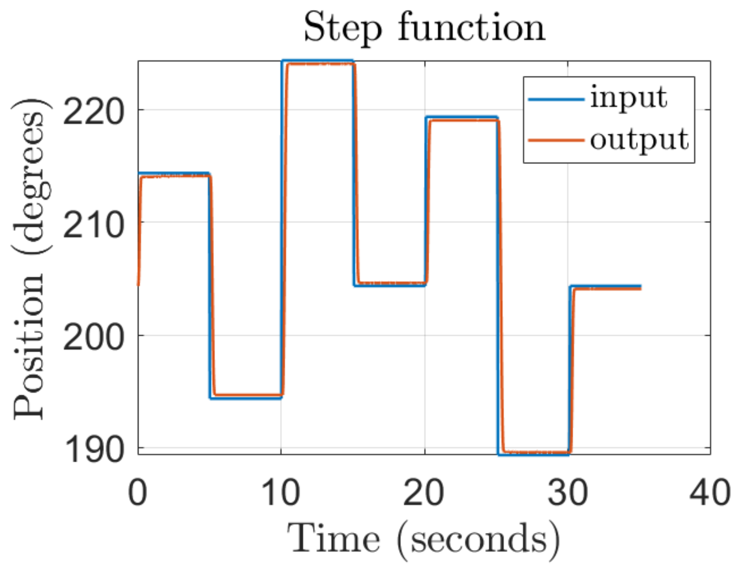


Figure 12: Roll Step Response

7 Experimental Setup and Testing

7.1 Thrust Stand Characterization

- **Development of a torque stand:**
  To characterize the torque of the MX-106T motor on the Roll axis, a test setup was constructed. A CAD interface was developed for the motor to connect with the rod used for testing. Initially, a CAD built pillar was considered to support the motor for testing. However, this idea was scrapped in favor of using a simpler solution using supplies in the lab. The torque stand can be seen below.

Figure 13: Torque Stand to characterize MX-106T motor

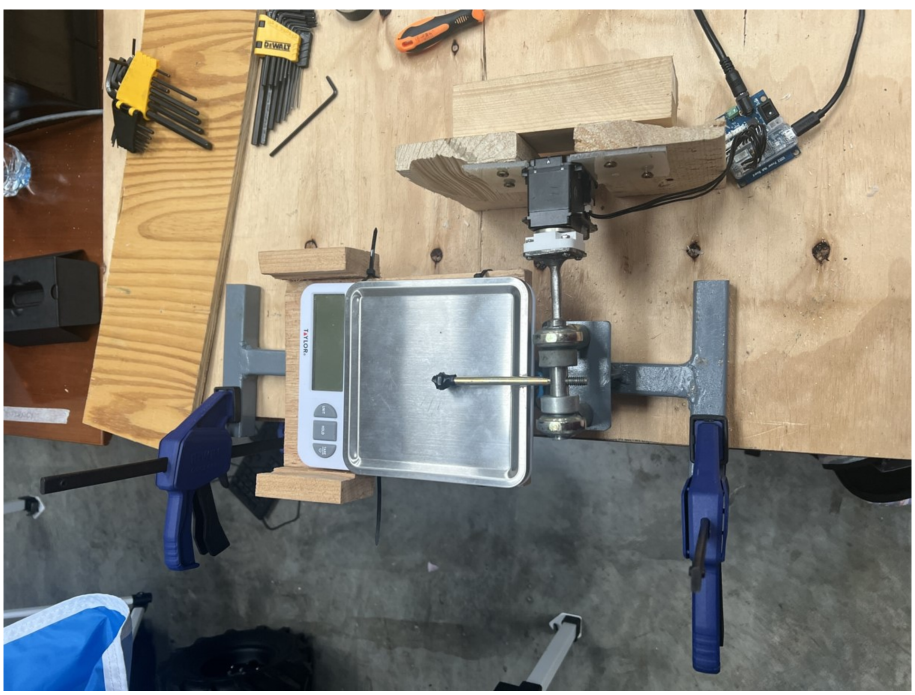


Figure 14: Above View of Torque Stand featuring CAD interface

- **Data collection methodologies:**
  To collect the data, one team member adjusted the current in the Dynamixel wizard, while the other team member approximated the measured mass value on the scale to get the

mass value to calculate the force. Multiple trials at each current were conducted to take the average of the measured values to calculate the torque.

- **Results of characterization:**
As mentioned earlier, characterization was difficult due to the fluctuations of the scale. However, an interesting trend was observed when the direction of torque rotation changed. When torque rotated in the counter-clockwise direction, the trend significantly changed, as can be seen in the figures below generated using the "motor_characteristic.m" file and values calculated from data gathered in the lab that was put into the files: "Torque_cw_times_calcs.csv" and "Torque_ccw_times_calcs.csv".

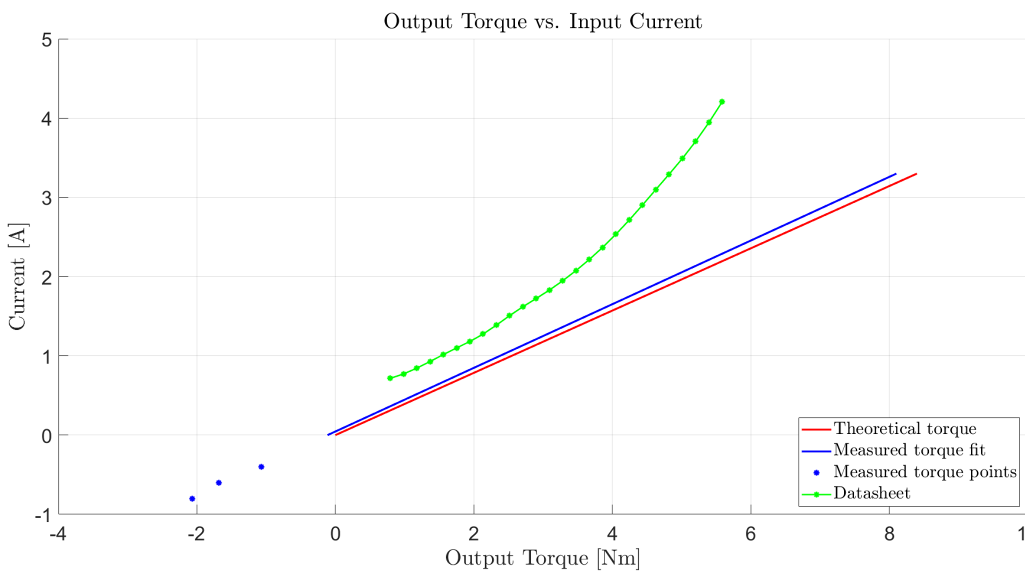


Figure 15: Torque characterization in clockwise direction

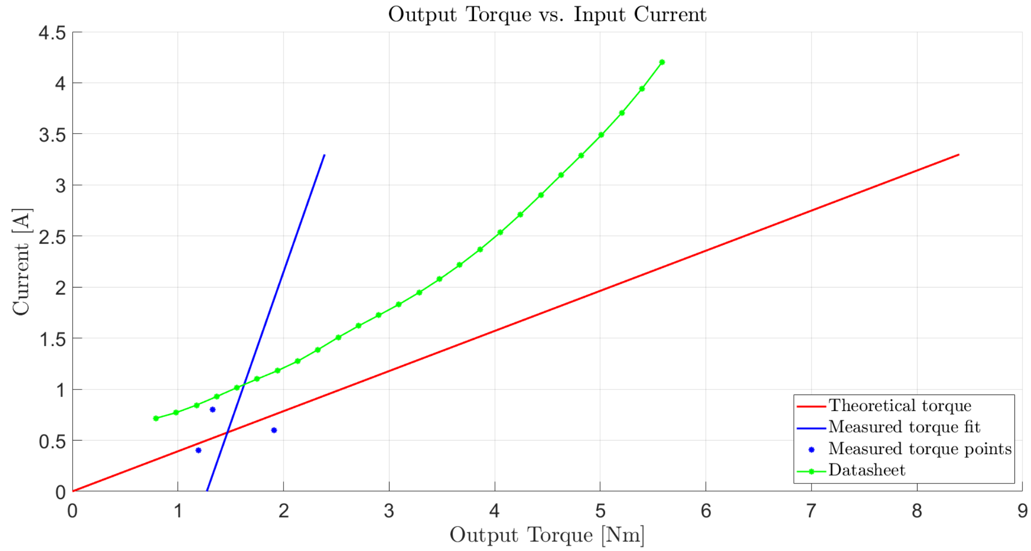


Figure 16: Torque characterization in counterclockwise direction

7.2 Motor Control Testing

- **Initial motor movement tests using DynamixelSDK:**
  The initial tests were simply verifying if the motors could move.

- **Simultaneous control of all ring motors:**
  All three motors were tested using the DynamixelSDK to verify if they could move simultaneously. All three motors did move simultaneously.

- **Input response tests: Sinusoidal and step inputs:**
  Sinusoidal and Step inputs were programmed to run on each of the motors. The sinusoidal inputs ranged in frequency from 0.5 Hz to 0.25 Hz in 0.05 Hz intervals, and amplitudes of 5.0 degrees to 25.0 degrees in 5.0 degree intervals. The purpose of this was to determine the difference between the inputs and outputs of the motors. For example, the Sinusoidal response of the Pitch axis seen below has a slight delay in response. Step inputs were not tested on Yaw, due to Yaw's unstable reaction to sinusoidal inputs. To graph the step and sinusoidal inputs and output responses, two MATLAB scripts were developed: "DataProcessing.m" and "StepFunction.m".

- **Zeroing the motors:**
  After the Sinusoidal and Step inputs were tested, Dr. L'Afflitto suggested that the team look into calibrating the motors when testing so that it would return to zero before each test. This was accomplished through the development of the some C++ scripts. Two of them are `zerod_position.cpp` and `zerod_position01.cpp`, which can be found on the Odroid. The motors move to predetermined encoder values that are experimentally

determined to make the motors level. In late October, the sinusoidal trials on the 3 Degrees of Freedom were re-conducted utilizing this zeroing ability.

- **Analysis of motor performance and identification of failures:**
As can be seen in Figures 8 through 10, the degree of freedom with the closest output to the input function was the Roll axis. This is not surprising due to the Roll axis bearing the least of the inertia generated as it is only dealing with the inertia of the Roll degree of freedom. Yaw performed the worst, which prompted further investigation into the Yaw degree of freedom motor and physical setup. It was found that the inertia of the structure may be require too powerful of a torque to overcome for the current Dynamixel MX-106T motor. This calculation was done through a MATLAB script developed. This prompted the structure to be re-evaluated and attempts are being made to lighten it.

8 Challenges and Resolutions

- **Mechanical issues: Yaw ring motor failure and thrust stand modifications.**
The challenges faced by the large Moment of Inertia of the thrust stand has been addressed primarily by the CAD redesign and the ordering of lighter parts. Additionally, the frame of the thrust stand shakes when the Yaw motor is moved quickly.

- **Software integration hurdles: Ensuring seamless communication between ROS2 and Dynamixel SDK.**
Challenges encountered during the installation and setup of the ROS2 workspace were overcome by leveraging available resources and support from the ACSL graduate students. The first challenge was the proper setup of the ROS2 workspace with the ACSL flightstack. The benefit of overcoming this challenge is that the team learned how to better read an instruction manual for software setup and how to utilize available resources.

- **Data accuracy and reliability in torque measurements:**
The accuracy of the torque measurements is rough at best. When testing, the scale values displayed when struck by the motor would fluctuate. Therefore, to refine testing the team began to take the measurements at rough intervals of 5 seconds. Additionally, when the force of the motor at a current of 2 Amperes was applied, it exceeded the weight measurable by the scale. Despite the rough estimation, the torque measurements were valuable in providing the team with insight into how the motor's torque relates to the torque in the MX-106T datasheet.

9 Results and Discussion

9.1 Torque Stand Performance

- **Summary of torque measurements and motor performance:**
  The torque measurements gathered compared to the MX-106T datasheet varied depending on the direction of the rotation of the torque. This method of testing should be explored further.

- **Impact of thrust stand modifications on overall system performance:**
  The overall system performance will be evaluated next semester. The goal is to lessen the torque exerted on the motors of the thrust stand in order to have them be within the torque capabilities of the MX-106T motors.

10 Conclusion

- **Recap of project objectives and achievements:**
  This semester heavily focused on identifying existing issues with the thrust stand which are motor issues and a large moment of inertia. Achievements this semester include researching potential motor replacements in the event that the Yaw motor continues to be problematic, becoming familiar with the thrust stand, characterization of the Roll motor's torque capabilities, and a CAD design of the drone to be used on the thrust stand.

- **Summary of key findings and their significance:**
  It was determined that the option to fully replace the Yaw motor would be an expensive investment for the project. Therefore, at this time, the team will be pursuing a cheaper option by replacing the damaged gears of the Yaw motor with gears purchased from Robotis.

- **Future work and recommendations for further development:**
  Future work next semester will be focused on implementing control algorithms developed by Dr. L'Afflitto, Luca, Giorgio this semester. Additionally, the thrust stand will be reassembled when the team returns from winter break.

11 Acknowledgements

The team would like to thank the following individuals for their contributions to this project.

Dr. Andrea L'Afflitto:

- Weekly team meetings and providing guidance for project development.

- Modeling the equations of motion of the thrust stand.
- Designing the controllers for the thrust stand.

Graduate Students of ACSL:

- Luca Nanu (CAD, torque testing, control algorithm development, "motor_characteristic.m" and "figure_plot.m" files)
- Giri Mugundan (Software)
- Mattia Gramuglia (Software)
- Giorgio Orlando (Control algorithm development)

12 References

References

Dynamixel MX106-T (n.d.). `https://www.robotis.us/dynamixel-mx-106t`. Online; Accessed: December 23, 2024.

DynamixelSDK (n.d.). *Dynamixel SDK GitHub Repository.* `https://github.com/ROBOTIS-GIT/DynamixelSDK/tree/master`. Online; Accessed: December 21, 2024.

Gramuglia, M., G. Mugundan, and A. L'Afflitto (n.d.). *ACSL-flightstack.* `https://github.com/andrealaffly/ACSL-flightstack`. Online; Accessed: December 21, 2024.

MX-106T/R Datasheet (n.d.). `https://emanual.robotis.com/docs/en/dxl/mx/mx-106/#drawings`. Online; Accessed: December 22, 2024.

A. L'Afflitto. (2017). *A Mathematical Perspective on Flight Dynamics and Control*, Springer, London, UK. The foreword has been kindly provided by Professor R. Beard. Gramuglia, M., Kumar, G.M., & L'Afflitto, A. (2024, January). *A Hybrid Model Reference Adaptive Control System for Multi-Rotor Unmanned Aerial Vehicles*, AIAA SciTech, Orlando, FL.

Gramuglia, M., Kumar, G.M. & L'Afflitto, A. (2024, July). *Two- layer Adaptive Funnel MRAC with Applications to the Control of Multi-Rotor UAVs*, International Workshop on Robot Motion and Control, Poznan, Poland.

Gramuglia, M. (2023). *Design and high-fidelity simulations of an adaptive control system for X8-copters employed for payload delivery.* MS thesis, Polytechnic of Milan, Italy.

A Appendix A: Parts List

Quantity	Item	Price per unit	Cost	Hyperlink
2	Precision Weights	21.98	43.96	Amazon1
1	Gearset	127.70	127.70	Dynamixel
2	Slip Rings	55.84	111.68	Amazon2
1	Aluminum Rods	14.99	14.99	Amazon3
2	Flange-Mounted Shaft Support for 1/2" Shaft Diameter, 2024 Aluminum	92.81	185.62	McMasterCarr1
1	Multipurpose 6061 Aluminum Rectangular Tube, 1/8" Wall Thickness, 2-1/2" High x 2-1/2" Wide Outside, 1/2 ft length	15.91	15.91	McMasterCarr2
1	Multipurpose 6061 Aluminum Rectangular Tube, 1/16" Wall Thickness, 1-1/2" High x 1-1/2" Wide Outside, 1 ft length	5.59	5.59	McMasterCarr3
2	Silver Corner Bracket, 1" Long for 1" High Rail T-Slotted Framing	7.92	15.84	McMasterCarr3
4	Silver Corner Bracket, 1.5" Long for 1.5" High Rail T-Slotted Framing	8.80	35.20	McMasterCarr4