Electrical Engineering Department

California Polytechnic State University

Reaction Wheel Based Rocket Active Spin Stabilization

Tanuj Vemuri

Dr. Xiao-Hua (Helen) Yu

Cal Poly Space Systems

EE 462

17 February 2023

Abstract

Cal Poly Space Systems is the high-powered and experimental rocketry club at Cal Poly SLO. They aim to design, test, and launch fully integrated launch vehicles. In order to create efficient and reliable vehicles, stabilization systems must be implemented. One parameter that can be actively stabilized is the vehicle's roll. The roll of the vehicle can have noticeably effects on vehicle performance, as uncontrolled roll leads to decreased system stability in flight. Uncontrolled roll can also have negative impacts on the quality of onboard video, which is used for post-flight analysis and marketing.

Reaction wheels are a popular method for active control systems in the aerospace industry. Through the use of an actively controlled reaction wheel, roll stabilization can be achieved. The system is completely contained within the body of the vehicle and can be placed anywhere along the rocket's cylindrical axis. With this stabilizer, the club can achieve more reliable flight performance, and improved video capture by keeping the cameras pointed in a consistent direction during flight. The aim of this design is to develop a low complexity and cheap method to maintain roll stability.

Through a combination of simulation and physical testing, a reaction wheel stabilizer was designed for Cal Poly Space System's internal research vehicle. This vehicle had a diameter of 5 inches and a wet mass of 10kg. The system is centered on the STM32F411 and MPU-6050, the microcontroller and IMU selected for this project. Through the control of a burshed DC motor with attached aluminum wheel, the system was designed to stabilize roll on the cylindrical axis of the vehicle. Although simulations showed mitigation of roll, hardware testing highlighted possible issues with motor saturation.

Acknowledgement

I would like to take a moment to thank all of the wonderful people supporting me, without whom I could never have begun taking on such a project. Dr. Helen Yu, for beginning my interest in the field of control systems and supporting me throughout my time at Cal Poly as both a professor and advisor. Dr. Eric Mehiel, for supporting me in understanding the complex world of aerospace and helping to mesh my knowledge of control theory with aerospace applications. And finally, my friends in Cal Poly Space Systems, who for the last 4 years have given me a space to live up to my childhood dreams of building rockets.

Table of Contents

List of Figures

Chapter I. Introduction

Cal Poly Space Systems is the High-Powered and Experimental Rocketry club on campus. In their efforts to develop an efficient and reliable launch vehicle, the club requires active stabilization systems to be implemented. Vehicle roll is one parameter that must be stabilized, as it affects both vehicle trajectory, as well as the quality of video collected by the onboard systems. The club collects onboard video for both marketing and analysis purposes. For this reason, they require the video to be stable and ideally pointing in a constant direction.

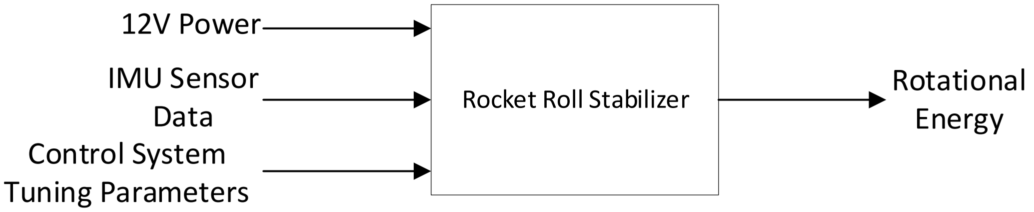


Figure 1 - System Block Diagram

Roll control can be achieved using a reaction wheel. When placed on the cylindrical axis of the vehicle, the reaction wheel can create the necessary angular momentum needed to cancel out those induced by aerodynamic forces. Using an accelerometer, the live movement of the vehicle can be recorded and acted upon. This data is then fed through a control algorithm to determine the correct speed needed from a DC motor. The reaction wheel is essentially a mass on this motor which, when spun, adds to the angular momentum of the system. In order to abide by the law of conservation of momentum, the entire rocket is forced to roll in the opposite direction. This is shown in the block diagram in figure 2.

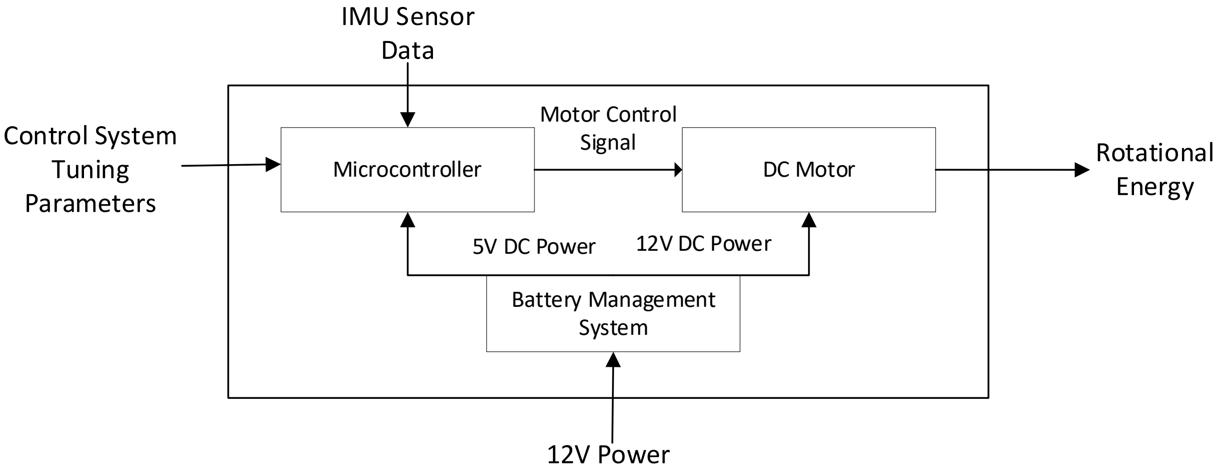


Figure 2 - System Functional Decomposition

This system represents a solution with low hardware complexity. The entire module is contained within the rocket. This method of stabilization also has the advantage of placement. As long as the wheel is radially centered withing the rocket, it can lie anywhere on the cylindrical axis, allowing for convenient placement. The reaction wheel system can also be entirely contained within the structure of the vehicle, eliminating any additional protrusions, which results in lower aerodynamic drag.

The reaction wheel-based approach presents a significant weight cost. In order to create the necessary momentum needed to control a fully loaded vehicle, the mass portion of the wheel would need to be accordingly massive. This also inhibits system scalability as the mass of the wheel required would increase with the overall weight of the rocket.

A different approach to the issue of roll stabilization is using passive stabilization systems. These systems force the rocket to roll in a single direction, increasing stability. This can be achieved through manipulation of the aerodynamic surfaces on the rocket, or through a similar reaction wheel system, which instead constantly spins in one direction. However, both approaches fail to consider the need for a stable video feed. By using an active system that reacts the rockets movement, it is possible to keep the onboard cameras pointed in a single direction.

An active stabilization system can also be implemented using motorized aerodynamic surfaces such as motorized fins [4]. This would reduce weight by eliminating the need for a reaction wheel allowing for an overall lighter-weight system. However, this approach leads to increased internal complexity of the vehicle, as data and control signals must travel from the avionics package to the bottom of the rocket where the fins are typically placed. The active reaction wheel system can be entirely self-contained and placed where convenient.

The reaction wheel based active spin stabilization system provides Cal Poly Space Systems with an uncomplicated and self-contained system, which allows for the club to achieve stable flight and video transmission.

Chapter II. Literature Review

Attitude control for spacecraft and launch vehicles is a heavily researched topic in the field of aerospace engineering. Several solutions have been studied and implemented by industry and student rocketry teams. Since the design covered in this project is meant for a collegiate rocketry club, student research was also considered for this review.

Reaction wheels are a common solution for attitude control in spacecraft. By imparting torque on to the dynamic system, reorientation of the satellite can be achieved. In a paper for the journal *Acta Astronautica*, Jeffrey T. King, a professor of Aerospace Engineering at the United States Naval Academy, outlines the considerations that must be made for the torque requirements of a reaction wheel system. In his study, King explores the agility of reaction-wheel control for multi-axis applications. The system outlined in the paper consists of multiple orthogonally oriented reaction wheels in order to achieve control authority in all axes. His findings illustrated that reaction wheel effectiveness is correlated with their orientation in regard to the system’s inertial moments [1].

Modeling of rocket dynamics has been well documented in the field of aerospace. A paper from the MIREA-Russian Technological University illustrates the mathematical model used to relate Euler’s equations to the Quaternion of the vehicle. This allows for the design and simulation of control schemes that account for the coupling of all three body axes in relation to the system’s dynamics. Using the MP-12 geophysical rocket as the focus of the simulation, PID control was designed and implemented to stabilize the pitching and yawing motion of the vehicle [2]. Similar analyses have been completed for the purpose of active guidance of rockets. Professor Shu Yang, from Northwestern Polytechnic University, explores the design of control schemes for actuated fins and canards. The paper explains that the dynamics of the vehicle are nonlinear, which leads to the need for more robust modeling and stability analysis [3].

One of the most common approaches to active roll control is using aerodynamic surfaces, such as fins and canards. A system implemented by students at University of Alabama in Huntsville used this approach. The students implemented an algorithm that used the vehicle’s angular velocity to determine the required angle of attack for the servos to set the fins to. This was paired with a PID controller to eliminate roll in the vehicle [4]. The system was defined as an individual module placed above the propulsion section of the vehicle structure. The team noted that this modular design allowed for easier testing and changes to the system. During testing of the system, it was determined that the fins could only handle a maximum angle of attack of ±5 degrees, as aerodynamic flow separation occurred at higher angles. This led the team to aim for a target roll rate that was non-zero, which unfortunately was not achieved. The main challenges faced by this team came in the integration of the system with the vehicle. Since the module was designed as a separate section of the body, getting a smooth interface and connection proved difficult.

Similar systems have been designed and implemented by students at Santa Clara University. These students implemented a Linear Quadratic Controller to stabilize an unstable 6 degrees of freedom model developed for the rocket [5]. This allowed them to simulate the optimal feedback gains for the system. While the system was never physically tested, simulations showed that the system would be successful once implemented.

The analyses completed by previous provided a foundation of analysis of vehicle dynamics. Many of the student led projects performed analysis through MATLAB and Simulink, using the 6DOF block from the Aerospace Toolbox. The fin analysis performed by students at the University of Alabama in Huntsville, informed this projects analysis of fins and their contribution to the vehicles roll during flight. While the Linear Quadratic Controller from the students at Santa Clara University served as reference for higher complexity control schemes, the difficulty of implementing this in a physical system made this a difficult approach for this project.

Chapter III. Background

Rockets and Launch Vehicles have three distinct axes of freedom: roll, pitch, and yaw. The stability of these three axes can have immense impacts on the flight performance of the vehicle. As a rocket accelerates during its thrust phase, the atmosphere applies a load to the structure of the vehicle. In order to maintain structural integrity, it is imperative that the lateral forces along the rocket are minimized [7]. By controlling the roll of the launch vehicle, it is possible to improve overall system stability. Figure 1 shows the simplified free-body diagram of the aerodynamic forces. As seen on the left and right, as the rocket loses pitch and yaw stability, a lift force is introduced. This additional force reduces flight performance and adds stress to the vehicles structure.

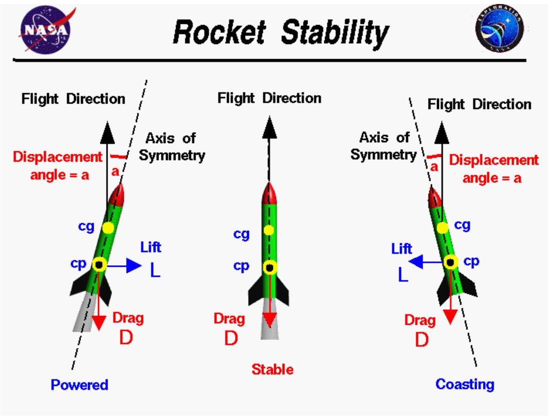


Figure 3 - Visualization of Aerodynamic Forces on Rocket [11]

Rolling occurs most commonly due to defects in fin construction. Tail fins are one of the biggest factors in system pitch and yaw stability, as they act as control surfaces throughout vehicle flight. However, even the slightest misalignment or deflection of these fins can induce roll for the rocket [10]. While purposely induced roll can improve stability, uncontrolled roll can have negative impacts on the pitch and yah stabilization of the vehicle. This would subject the structure of the rocket to excess forces, decreasing flight performance and reliability. Active roll control can be achieved using a simple single-input single-output control system that adjusts roll rate or roll angle. This is largely due to the roll axis of the rocket lying on its centerline.

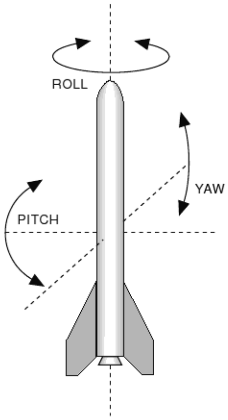


Figure 4 - Rocket Control Axes [9]

One way to control the roll rate of a rocket or launch vehicle is through the use of a reaction wheel. Reaction wheels are a form a flywheel, a device used to contain rotational energy and angular momentum. The primary difference between flywheels (also called momentum wheels) and reaction wheels is the ability to actively control the roll rate and angle [6]. Momentum wheels typically spin at constant high speeds, leading to stable systems that oppose attitude change. On the other hand, reaction wheels can spin up or spin down, allowing for greater attitude control.

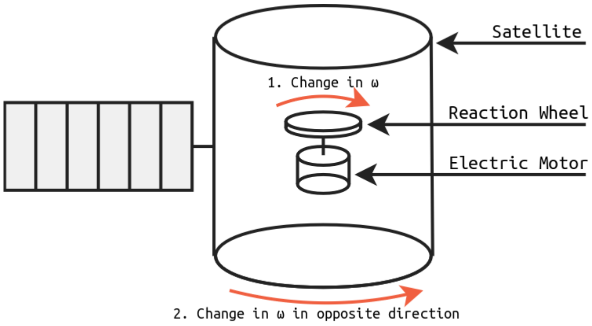


Figure 5 - Reaction Wheel in a Satellite [8]

Reaction wheels operate on the principles of conservation of momentum. The system consists of a motor and mass wheel. By rotating the mass, the reaction wheel induces an angular momentum on the overall system. In order to counteract this introduced angular momentum, the vehicle must rotate in the opposing direction, causing the total angular momentum of the vehicle to remain constant. Figure 3 shows a basic implementation of a reaction wheel on a satellite. In the figure, the term ω represents angular velocity.

Chapter IV. Simulation Design and Results

The proposed launch vehicle for this project was designed by Cal Poly Space Systems for an internal research project. The vehicle has a body tube diameter of 12.7cm and an overall mass of 10kg. Using OpenRocket®, a flight trajectory analysis tool for rockets, the vehicle was modeled and is shown below [12]. The tool allows for the various components of the rocket to be designed, such as the fins, which informed the aerodynamic analysis for the roll control simulation. In this design, the reaction wheel module is placed immediately above the center coupler.

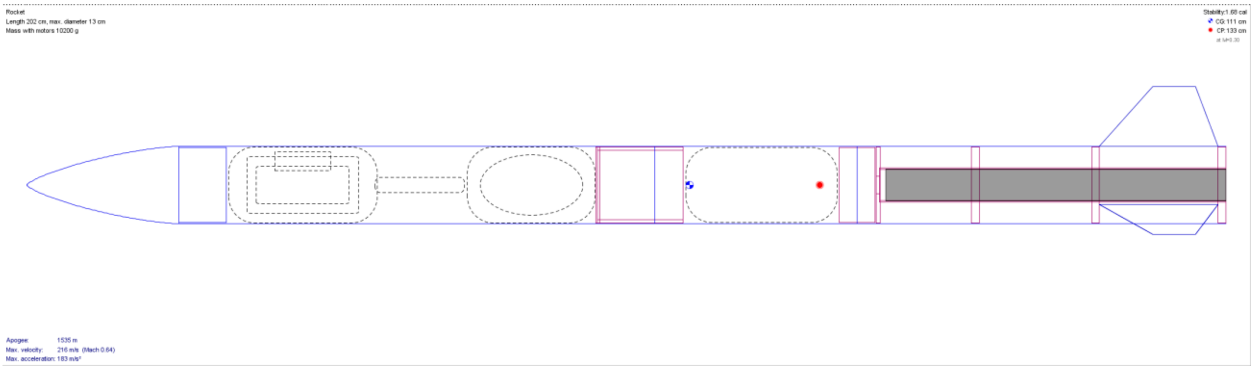


Figure 6 - Launch Vehicle Model

Design and simulation of the system was completed using MATLAB and Simulink. The center of the design rested on the 6 degrees of freedom (6DOF) module from the Aerospace block set. This module allowed for simulation of all three body axes, while offering several outputs, including rotational velocity and acceleration. The inputs to the 6DOF block are in the form of three-dimensional matrices of the system Forces and Moments.

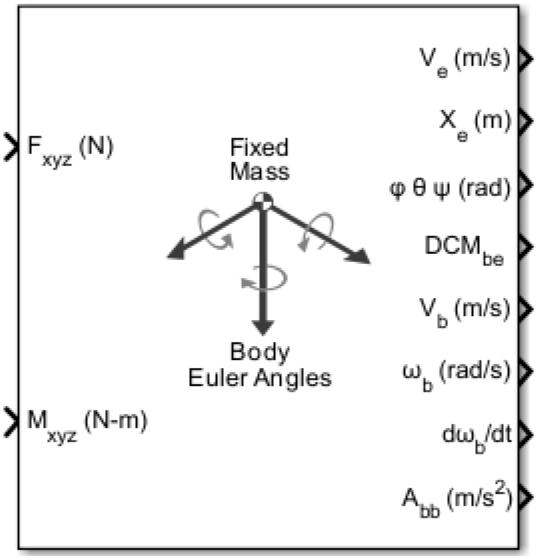


Figure 7 – 6DOF block in Simulink

The primary simplification made to the system was its discretization into two distinct operational modes. This must be done, as the mass of the system is variable due to the burning of the rocket motor. As such, the decision was made to model the system during the thrust phase, and after motor burnout as two separate models.

The thrust phase was further simplified by taking the average of several performance parameters to achieve a more linear model. One of these parameters was the rocket motor's. The motor being used by the rocket is the Cesaroni K1440-17A which has a burn time of 1.7s. Due to the short duration of the thrust phase, it was determined that simplification of the system would be acceptable. The thrust profile for this motor is shown in figure 4 below.

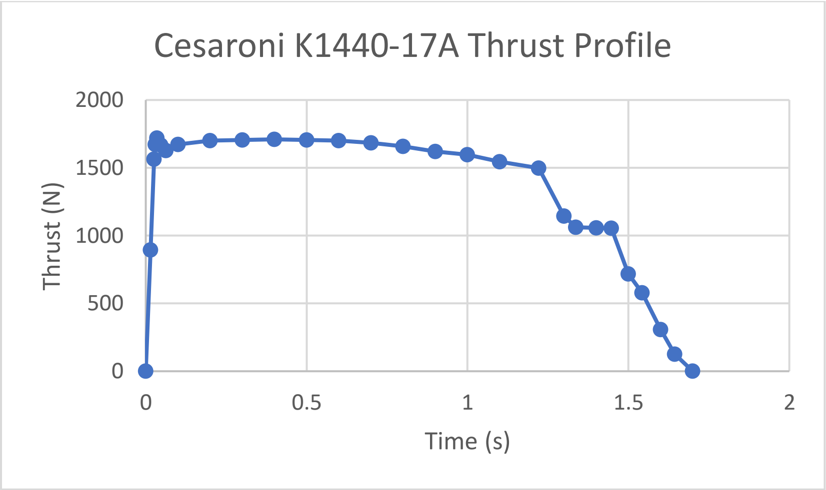


Figure 8 – Thrust profile for the Cesaroni K1440-17A Rocket Motor [13]

From this data, the average thrust was determined to be 1295.9N. This thrust was used as the propulsive force, which was input into the 6DOF block. This was the only force input used in the simulation. There were two moments that were simulated as inputs in the system. The rotational torque from the reaction makes up the first moment, and is the moment being controlled. The second moment is the aerodynamic moment of the launch vehicle. This rotational torque is caused by the lift generated by misaligned fins, akin to how wings generate lift for a plane. For the purposes of modeling, the fins we're assumed to be misaligned by 2 degrees in the same direction, meaning the moment generated by each wing would be equivalent.

First the coefficient of lift (C_L) was calculated using the area of each fin and the angle of attack, which as stated above was modeled as 2 degrees. This was then used in the following lift equation:

$$L = C_L * 0.5 * \rho * V^2 * A$$

For ρ, the density of air, the value was assumed to be equivalent to that of sea level. The vehicle velocity, denoted by V, was assumed to be the average velocity, as simulated using OpenRocket flight analysis software. Finally, the area of the wing, A, is the same as used in the calculation for C_L. Using a simple MATLAB function block, this was modeled as an input moment for the system.

The final input for the system was the control torque imparted by the reaction wheel. Using the angular rate output from the 6DOF block, an error signal was achieved. This is then run through a PID block in order to model a control moment. This was then added to the aerodynamic moment, to complete the moment input the system. The full Simulink setup is pictured below.

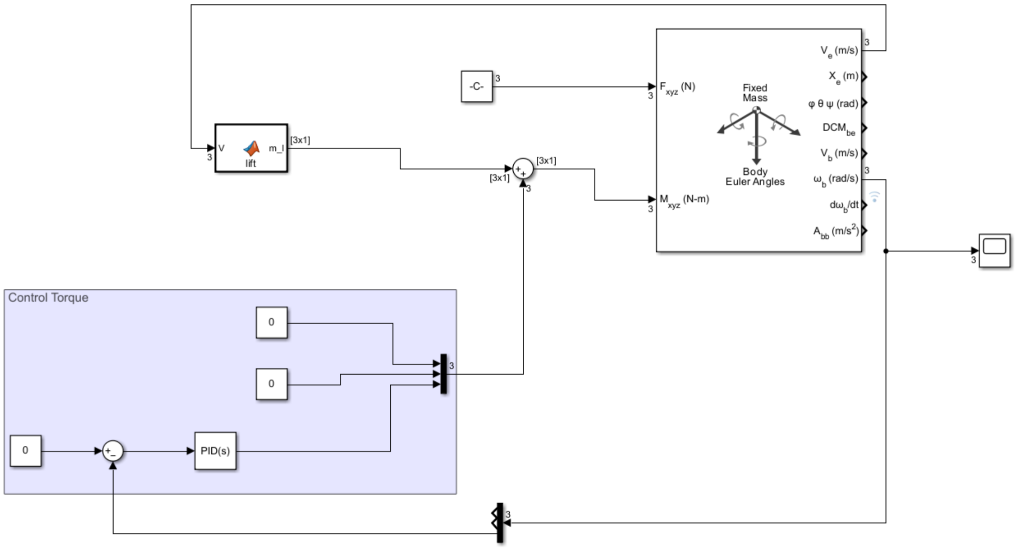


Figure 9 – Simulink Model

To measure the system behavior, the angular velocity output is run into an oscilloscope. The scope capture in figure 7 shows the uncontrolled behavior of the system. The roll rate in the cylindrical axis, which is shown in red, increases rapidly throughout the thrust phase of the flight. This is due to the acceleration of the motor thrust increasing the lift generated by the fins. Due to the nature of the simulation, the roll rates of the other 2 axis remain at 0.

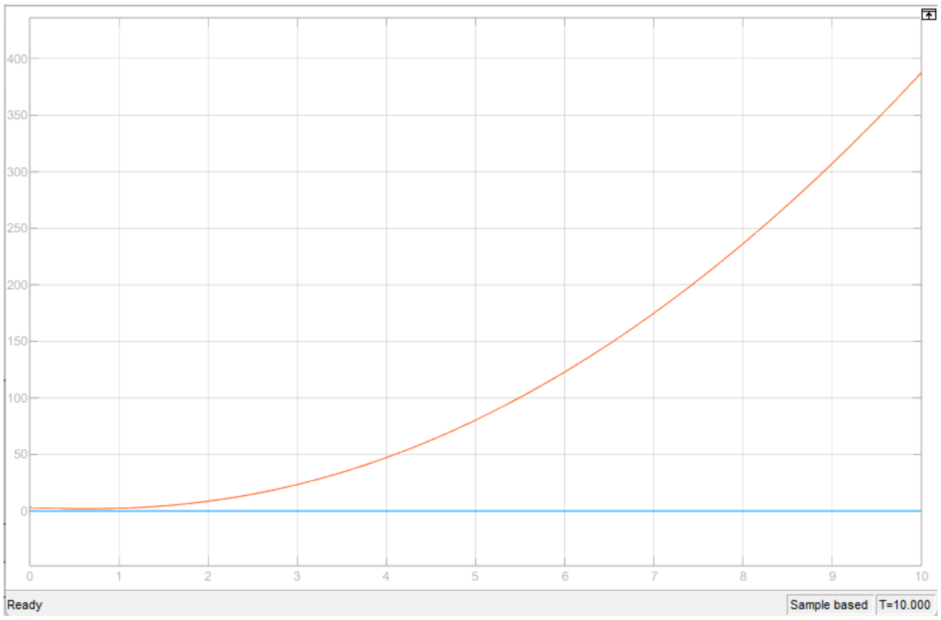


Figure 10 – Uncontrolled Angular Rate Output Capture

After applying a the PID control, the system behavior exhibits a significant change. The roll rate in the cylindrical axis is shown in orange in figure 8. There is an initial decrease in angular velocity, where the system begins to correct the initial spin upon take off, and a slight overshoot. However, the system does not eliminate the roll rate of the vehicle. Comparing the scaling of roll rate amplitude between the output captures, there is a significant decrease in roll rate after control is implemented. OpenRocket® simulations show that the time to apogee for the vehicle is approximately 14 seconds, at which point the tuned system only exhibits a roll rate of less than 1 rad/s.

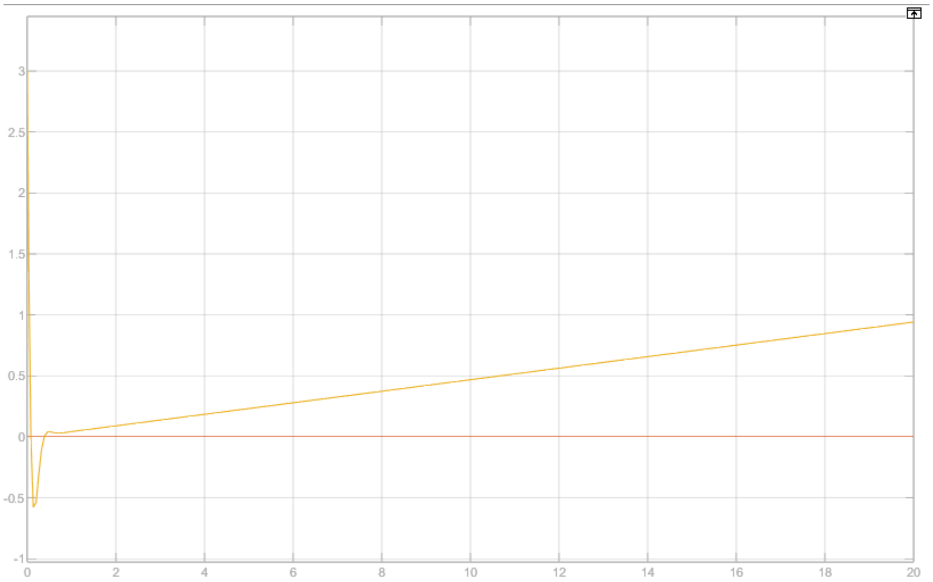


Figure 11 - Controlled Angular Rate Output Capture

Chapter V. Software Design

A simple PID controller was implemented using the Arduino IDE and STM32duino library on the STM32F411 “BlackPill” development board as well as the Adafruit MPU-6050 IMU. A flowchart for the program is shown below in figure 8. First, an I2C interface connection is initialized between the BlackPill and MPU-6050. Once this connection is established, the IMUs measurement parameters are set, including the gyroscope range and the digital lowpass filter bandwidth. A simple calibration is performed, sampling 1000 measurements and then averaging them to calculate an offset.

The board was programmed using the Arduino IDE 2.0 and the STM32duino library. This is an open-source library that implements ST microcontroller boards in the Arduino IDE [14]. Using the library allows for use of standard Arduino libraries for many features of the development board, such as serial interfaces, timers, and communication protocols. It also allows for the use of Arduino syntax for implementing basic GPIO pins. In conjunction with STM32duino, the Adafruit MPU6050 library was used in order to communicate with the IMU [15]. This greatly simplifies the initialization of the I2C connection between the BlackPill and MPU-6050 IMU.

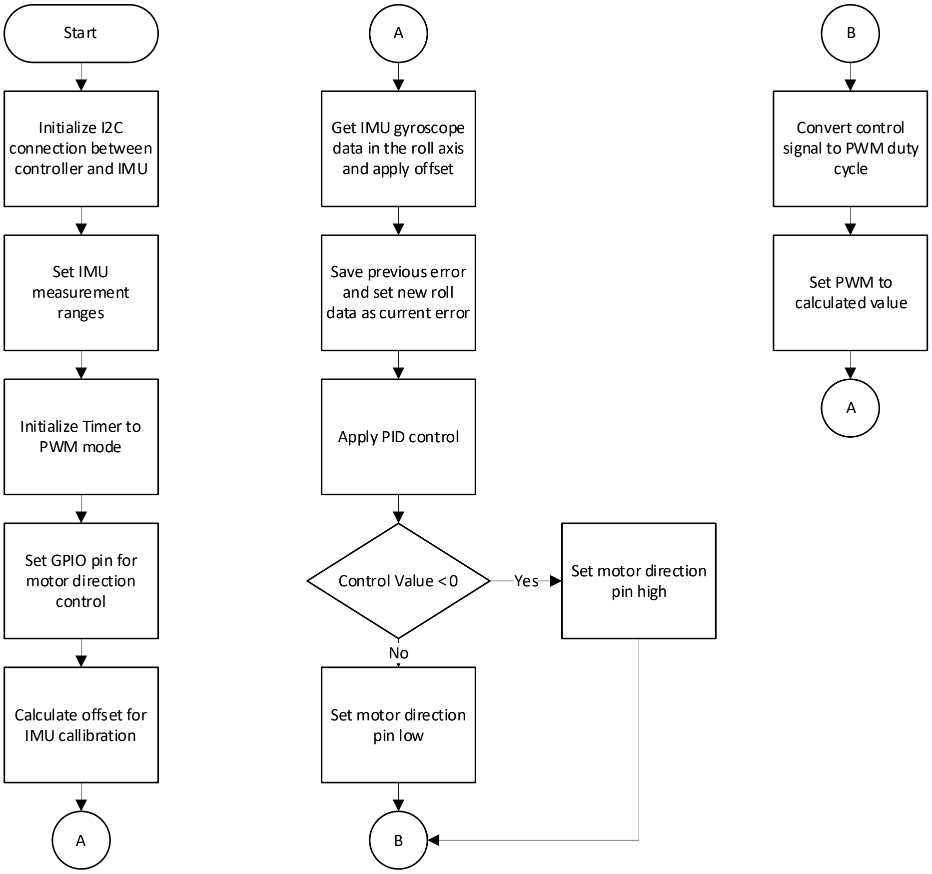


Figure 12 - Program Flowchart

The PID controller is implemented using a set point of 0, since the goal is for the vehicle roll to be eliminated. For this reason, the reading from the IMU is used directly as the current error signal. Motor direction is determined through the sign of the PID control output. A digital pin is set to high or low, which sends a signal to the motor controller in order to control the motor direction. The absolute value of the PID output signal is mapped to a PWM frequency, which is sent to the motor controller to control DC motor speed. This process is repeated, reading a new roll rate from the IMU gyroscope, and saving the previous roll rate as the previous error for use in derivative control.

Simple testing of the software was performed using the serial output terminal in the Arduino IDE. Initial testing involved simply printing IMU readings and toggling the BlackPill's onboard LED to determine direction. Once the I2C connection was established, the LED was enabled when counterclockwise spin was detected, which was indicated by negative readings from the IMU.

Chapter VI. Hardware Design and Results

The system was implemented using an STM32F411 microprocessor. This is an ARM based processor unit that is popular in the hobby RC and flight control communities. The development board, commonly known as the “BlackPill” provides several useful features for users, including 3.3V output, and a programmable button and LED. The chip contains multiple peripheral standards, such as I2C which is used for communicating with the IMU, and timers, which can be used to generate PWM signals for DC motor control.

Motor control and operation was implemented with the Pulolu DRV8874 Brushed DC motor driver. This chip allows for bidirectional control of a single brushed DC motor, accepting motor voltages from 4.5 to 37V, and providing a constant 2.1A of motor drive current. There are 2 main inputs from the microcontroller to control the driver. First, the EN pin receives a PWM signal, which determines the motor speed. As described in the previous section, the PWM duty cycle is determined by the PID control signal value. The PH pin on the driver determines the motor direction, and is controlled with a simple digital high or low signal. This was used to drive a Pulolu 37Dx50L 12V Metal Gearmotor. This is a brushed DC motor with a 10:1 gear ratio, offering a maximum torque of 4.9 kg·cm.

The final electrical component of the system was the Pulolu D24V10F5, a 5V step-down converter. This was necessary for the system to be operated off of a single voltage source, a 12V battery. The 5V output from the board provided power to the BlackPill microcontroller, which then has its own integrated 3.3V step-down converter to provide logic power to the MPU-6050 IMU and DRV8874 motor driver. A system interfacing diagram is shown below in figure 10. While a PCB was designed for the system, testing was conducted using perforated PCD prototyping boards. The designs for the PCB can be found in Appendix B.

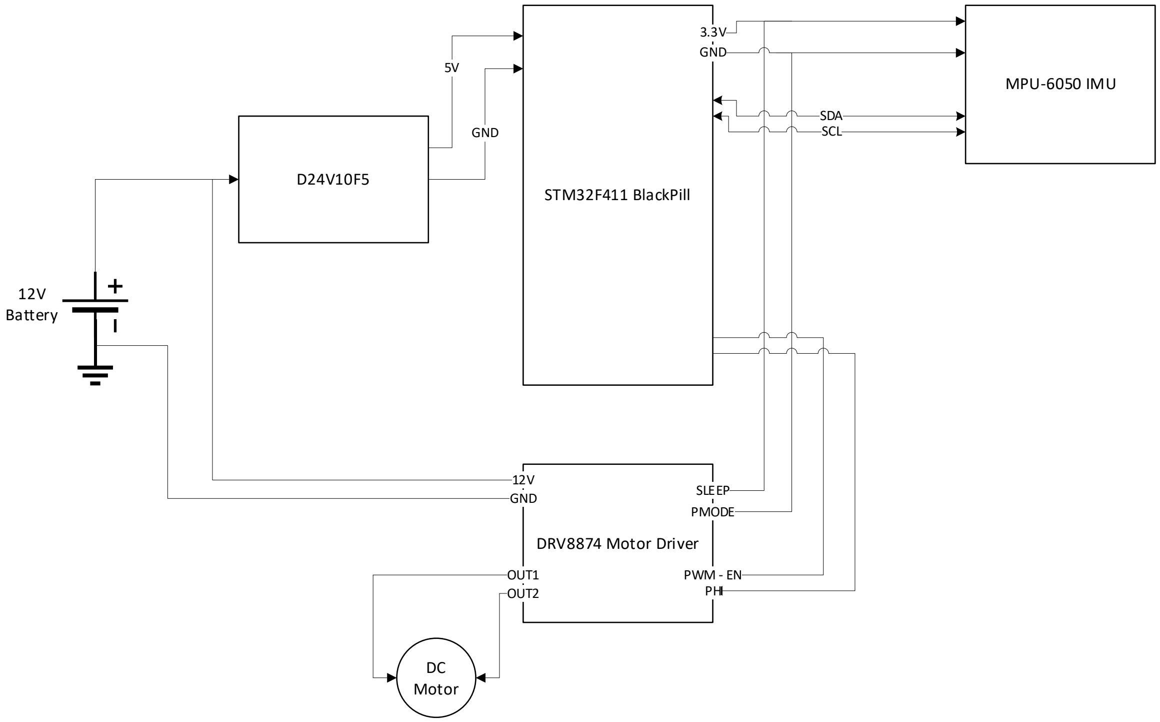


Figure 13 - System Interface Diagram

The final designed component was the reaction wheel. The wheel was created in SolidWorks® and has a major diameter of 4.5 inches and a thickness of 1 inch [16]. This size was informed by the inner diameter of the launch vehicle, which is 5 inches, allowing for maximization of wheel size, while still allowing for tolerances. Originally designed to be made of aluminum, manufacturing of this wheel was unfortunately not completed.

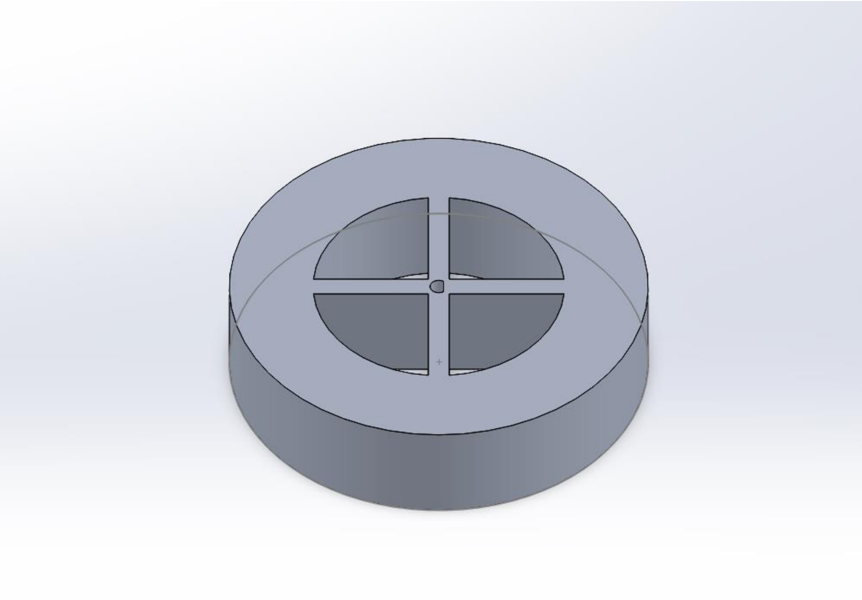


Figure 14 - SolidWorks Model of Reaction Wheel

Due to issues with the flight vehicle, fully integrated testing could not occur. However, simple ground testing of the system was performed. As a power source, a 12V Lead-Acid battery was utilized to support longer runtimes for testing and debugging. During testing, saturation of the reaction wheel was a repeated issue. This represents a failure of the system to provide a large enough torque to counteract the current roll rate. The causes the system to enter an infinite spin, as the reaction wheel fails to have an effect on the system. This unfortunately also caused damage to the motor driver, possibly due to high current draw. Upon further inspection, the motor was also hot to the touch, indicating higher power usage.

These issues with saturation may be an effect of component issues. One possible explanation would be in correctly chosen motor. As stated previously, the dc motor chosen for this system has a maximum torque 4.9 kg·cm. With an aluminum wheel instead of a PLA one, it may be the case that the DC motor would experience significant stalling, especially in cases where it must switch the direction of rotation. Another cause of saturation could be that the wheel itself may not have the required moment of inertia to impart enough torque and momentum on the system. The use of a different material may ameliorate this issue but could again run into the issue of stalling.

Chapter VII. Conclusions and Future Works

This project explores the practicality of using reaction wheel-based stabilization on a launch vehicle. Through simulation and testing, this was investigated as a possible solution to the application of roll stabilization. Overall, this project was an effective investigation into the practicality of reaction wheels for collegiate launch vehicles. One of the goals for this project was to explore reaction wheels as a cheaper alternative to previously researched topics of active roll control. This included methods such as active canard and fin control, which have been implemented by collegiate rocketry teams in the past. However, the system designed in this project minimizes the number of components and offers a more accessible method for such applications.

Several improvements could be made to the system, including further research on motor options and wheel materials, as well as full implementation of a PCB board for the integrated controller. Improvements could also be made to testing methods as well. Although a launch vehicle was not completed for this project, future testing could involve more integrated vehicle procedures. This could include wind tunnel testing to better characterize the roll caused by fins during flight, and the ability for the system to handle these angular velocities.

References

[1] J. T. King, “Increasing agility in orthogonal reaction wheel attitude control systems,” *Acta Astronautica*, vol. 177, pp. 673–683, Dec. 2020.

[2] A. S. Verkner and E. O. Guryanova, “The system of vertical movement stabilization of the geophysical rocket for the purpose of high-precision measurement,” *IOP Conference Series: Materials Science and Engineering,* vol. 747, no. 1, p. 012134, 2020.

[3] S. Yang, “Impact-point-based guidance of a spinning artillery rocket using canard cyclic control,” *Journal of Guidance, Control, and Dynamics,* vol. 43, no. 10, pp. 1975–1982, Jul. 2020.

[4] V. Braswell, H. Arcenaux, H. Strutzenberg, R. Taylor, and D. M. Lineberry, “UAH Student Launch Rocket Project: Roll Control of a sounding rocket through aerodynamic surfaces,” *53rd AIAA/SAE/ASEE Joint Propulsion Conference*, Jul. 2017.

[5] Guerrero, Valeria Avila; Barranco, Angel; and Conde, Daniel, "Active Control Stabilization of High Power Rocket" (2018). Mechanical Engineering Senior Theses. 81. https://scholarcommons.scu.edu/mech_senior/81

[6] A. Bitar, “Understanding reaction wheels.” [Online]. Available: https://aerospace.honeywell.com/us/en/about-us/blogs/understanding-reaction-wheels. [Accessed: 14-Oct-2022].

[7] A. Tewari, “Attitude Control of Rockets,” in *Advanced control of aircraft, spacecraft, and rockets,* Oxford: Wiley, 2011.

[8] C. Grassin, “Reaction wheel attitude control,” *Charles' Labs*. [Online]. Available: https://charleslabs.fr/en/project-Reaction+Wheel+Attitude+Control. [Accessed: 15-Oct-2022].

[9] “Rocket control,” *Science Learning Hub*. [Online]. Available: https://www.sciencelearn.org.nz/resources/387-rocket-control. [Accessed: 15-Oct-2022].

[10] “Rocket roll motion,” *NASA*. [Online]. Available: https://www.grc.nasa.gov/www/k-12/rocket/roll.html. [Accessed: 14-Oct-2022].

[11] “Rocket stability,” *NASA*. [Online]. Available: https://www.grc.nasa.gov/www/k-12/rocket/rktstab.html. [Accessed: 14-Oct-2022].

[12] “OpenRocket,” OpenRocket Simulator, https://openrocket.info/ (accessed Jun. 19, 2023).

[13] “Thrustcurve Hobby Rocket Motor Data,” ThrustCurve, https://www.thrustcurve.org/motors/Cesaroni/2372K1440-17A/ (accessed Jun. 2023).

[14] “Arduino core for STM32 MCUs,” STM32duino, https://github.com/stm32duino (accessed Jun. 19, 2023).

[15] Adafruit, “Arduino Library for MPU6050,” GitHub, https://github.com/adafruit/Adafruit_MPU6050 (accessed Jun. 19, 2023).

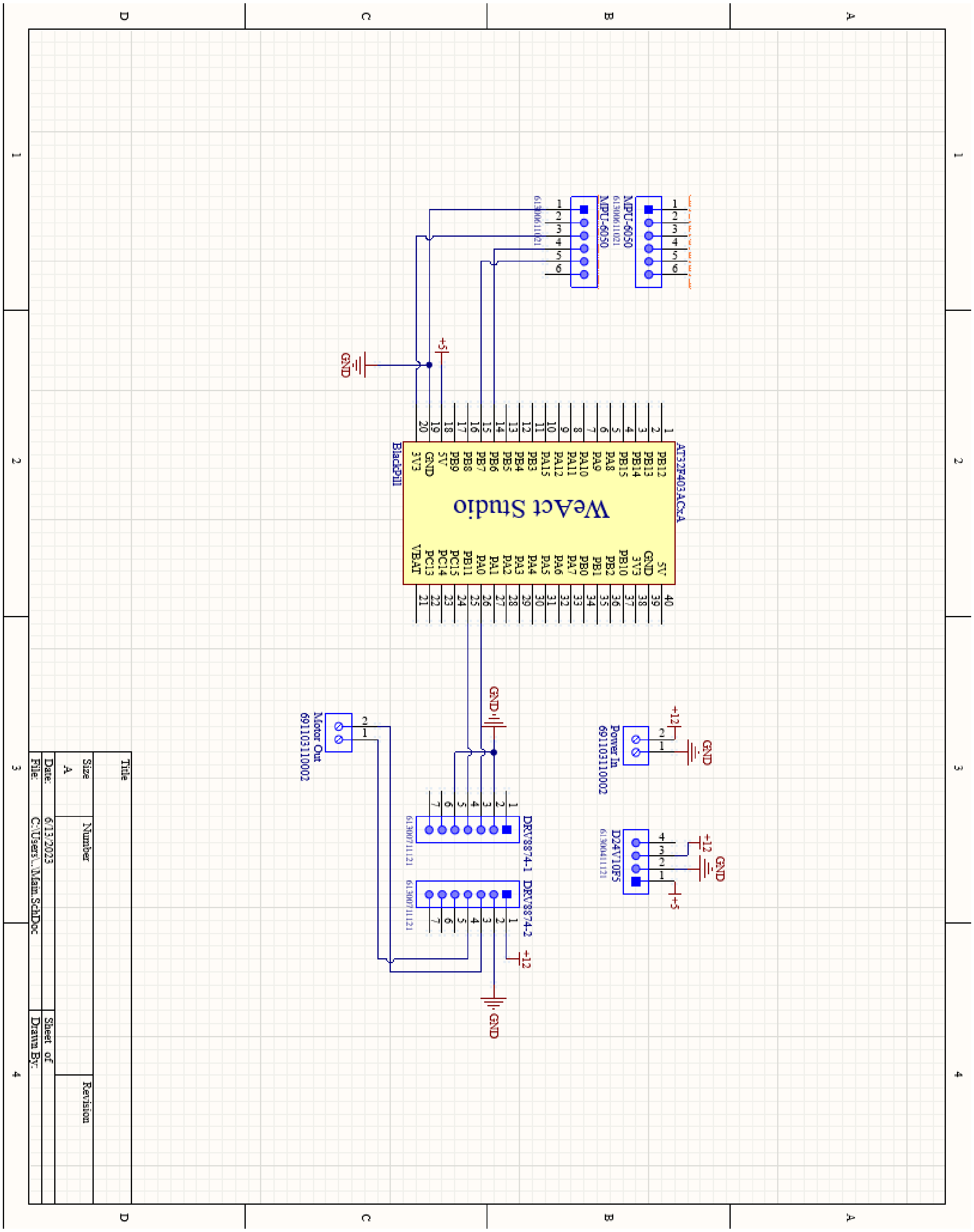
[16] “3D CAD Design Software,” SOLIDWORKS, https://www.solidworks.com/ (accessed Jun. 2023).

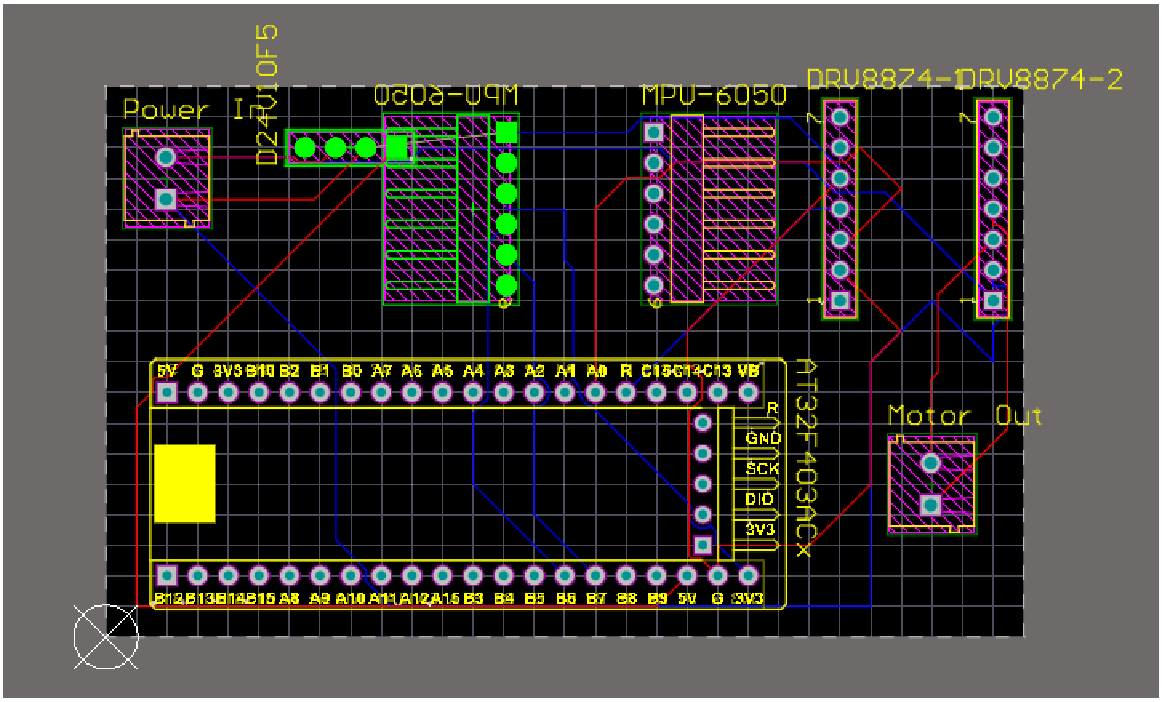
Appendix A – Program Code

```
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>

Adafruit_MPU6050 mpu;

//Initialize placeholder variables
int8_t current_error, previous_error = 0;
int16_t prop_error = 0;
int16_t total_integrated_error, integral_error = 0;
int16_t derivative_error = 0;
int16_t motor_control, motor_speed = 0;
int8_t Kp = 15;
int8_t Ki = 40;
int8_t Kd = 2;

void setup(void) {
  Serial.begin(115200);
  while (!Serial)
    delay(10);

  Serial.println("Adafruit MPU6050 test!");

  // Try to initialize!
  if (!mpu.begin()) {
    Serial.println("Failed to find MPU6050 chip");
    while (1) {
      delay(10);
    }
  }
  Serial.println("MPU6050 Found!");
  mpu.setGyroRange(MPU6050_RANGE_250_DEG);
  mpu.setFilterBandwidth(MPU6050_BAND_94_HZ);
  Serial.println("");
  delay(100);

  //Initialize Pin for motor direction control
  pinMode(B11, OUTPUT);
}
```

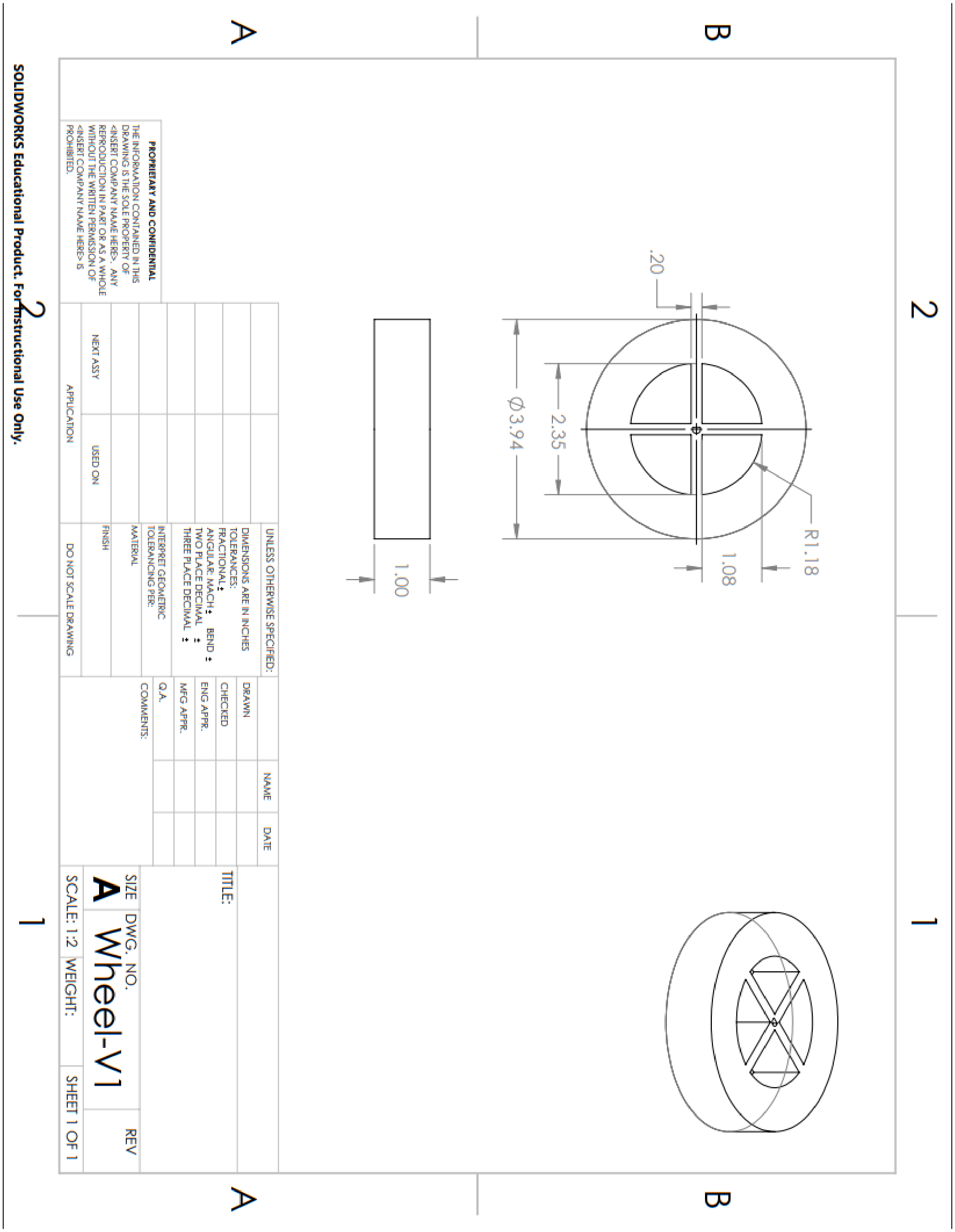
```
void loop() {

  /* Get new sensor events with the readings */
  sensors_event_t a, g, temp;
  mpu.getEvent(&a, &g, &temp);

  /* Print out the values for testing */
  Serial.print("Rotation X: ");
  Serial.println(g.gyro.x);

  /* Begin PID Control*/
  current_error = g.gyro.x;

  // Proportional gain
  prop_error = Kp * current_error;
  // Integral Error
  total_integrated_error += current_error;
  integral_error = Ki * total_integrated_error;
  // Derivative Error
  derivative_error = Kd * (current_error - previous_error);


  motor_control = prop_error + integral_error + derivative_error;
  if(motor_control < 1)
    digitalWrite(B11, HIGH);
  else
    digitalWrite(B11, LOW);

  //mapping motor_control value to PWM duty cycle
  motor_speed = map(abs(motor_control), 0, 32767, 0, 255);
  analogWrite(A0, motor_speed);

  //Save current error for derivative use
  previous_error = current_error;
}
```

Appendix B – PCB Schematic




Power In
D24V10F5
MPU-6050
MPU-6050
DRV8874-1
DRV8874-2
Motor Out
AT32F403ACX
5V G 3V3 B10 B2 B1 B0 A7 A6 A5 A4 A3 A2 A1 A0 R C15 C14 C13 VB
B12 B13 B14 B15 A8 A9 A10 A11 A12 A15 B3 B4 B5 B6 B7 B8 B9 5V G 3V3
R
GND
SCK
DIO
3V3

Appendix C – Reaction Wheel Mechanical Drawing



Appendix D – Bill of Materials

Item	Part number	Source	Quantity	Price
Microcontroller	STM32F411	Adafruit	1	$ 29.95
IMU	MPU-6050	Adafruit	1	$ 12.95
Motor Driver	DRV8874	Pololu	1	$ 9.95
Step-Down Voltage Regulator	D24V10F5	Pololu	1	$ 9.95
DC Motor	37Dx50L	Pololu	1	$ 32.95
Shipping and Handling				$ 28.28
Taxes				$ 9.42
Total				$ 133.45

Appendix E – Senior Project Analysis

Project Title: Reaction Wheel Based Rocket Active Spin Stabilization

Student's Name: Student's Signature: Tanuj Vemuri

Advisor's Name: Xiao-Hua (Helen) Yu **Advisor's Initials: Date**

Summary of Functional Requirements

This project designed a DC motor-based reaction wheel spin stabilizer for use in collegiate rocketry vehicles. The design provides a low complexity alternative to other approaches, such as canard control systems, and complex control schemes. It is based on readily available, hobbyist-grade components such as the STM32F411 Microcontroller and Adafruit MPU6050 IMU. The software for this project is also more accessible compared to traditional STM based applications, through its use of open-source libraries, like STM32duino which allows for programming of the device using the Arduino IDE. The system also establishes a cheaper alternative to other systems, by minimizing the total number of components needed for operation and allows for flexibility in key components such as the power source and DC motor.

Primary Constraints

This project was designed for Cal Poly Space Systems, the experimental rocketry club at Cal Poly SLO. The most significant challenge in undertaking this project was interfacing with the club and keeping up with changing requirements. One of the main limiting factors was the issue of weight. The vehicle proposed for this project had tight mass tolerances for the purpose of performance.

Economic

The primary impact of this project is its cost of human capital. The most complex portion of this project was the physical integration of the system. This not only includes the integration of individual components, but also the system as a whole into the launch vehicle. While the system is intended to be low complexity and adaptable, it requires significant testing and tuning to fully integrate into a vehicle. As such, the time invested into testing may often outlast the assembly and integration of the system. While the project is not meant to be a commercial product, it is meant to replicate and used in an open-source environment. As such, the financial gains from this project can only be analyzed from the lens of comparing it to alternative solutions. The costs of this project can be quantified in the materials and components. The final cost for this system was approximately $133.45, which includes the cost of all components as well as shipping.

The design and manufacturing of this system required several external tools, many of which are freely availably to students. Simulation of this system was conducted in MATLAB and Simulink which is free for students. Other required software for this system included PCB design programs, of which there are many free alternatives to the used Altium Designer, and the Arduino IDE for programming the microcontroller, which is also free. Some of the higher equipment costs come from manufacturing tools, such as soldering irons for construction on the PCB and 3D printers or mills and lathes for the reaction wheel.

Environmental

This project implements many commercial off the shelf electronics. The manufacturing of consumer electronics has a significant impact on the environment, as resource mines damage the ecosystems around them. Another issue that has become prevalent over recent years has been the disposal of electronics. Toxic elements are often dumped in landfills in third-world countries, where they are often picked apart by hand, exposing workers to dangerous conditions. Due to the global nature of the consumer electronics supply-chain, the acquisition of certain components is also a significant factor in the environmental impact of this project. Global shipping requires extensive use of fossil fuels and adds to the prevalence of greenhouse gases in the atmosphere, as well as pollution in the world's oceans.

Manufacturability

The main challenge in manufacturing this system comes from the reaction wheel. This represents the most significant equipment difficulty in the project. Depending on the material chosen for the wheel, the user would need access to either a 3D printer or mill and lathe if made of metal. While 3D printers have become more accessible over the years, even entry level options cost more than the components for this project. Making the reaction wheel out of a metal, or other machinable material would also require training and knowledge in the use of machine shop equipment.

Sustainability

This project contains several consumer electronic components, all of which are not made with sustainable materials. The elements that make of microprocessors are often toxic and non-recyclable. This is an issue inherent with the manufacturing of electronics, and in today's market is almost impossible to avoid. Similarly, the disposable of e-waste is a significant issue in the sustainability of electronics. While the system is built with minimal components, most of the components cannot be discarded in a sustainable manner.

Ethical

Misuse of this project could have significant ethical implications for the safety of others. While the system is designed for active stabilization of rockets, this could be modified to provide active control. Application of this system in that manner has both legal and ethical issues, specifically regarding the regulations of ITAR. The manufacturing of many consumer electronics also brings up ethical concerns in the treatment of workers. Many small-scale components are often manufactured in developing nations, due to the disregard of safety and fair treatment of workers. This could be counter-acted by sourcing components such as PCBs from ethical manufactures.

Health and Safety

Rocketry in general requires extensive safety procedures to keep people safe. This project aims to alter the behavior of a rocket during flight, albeit in an attempt to improve stability. However, improper tuning and testing of the device could lead to a de-stabilized flight trajectory. This could lead to unsafe flight of a launch vehicle, which poses harm to users.

Development

This project required significant research into the behavior of launch vehicles and aerodynamics. For the analysis of fins, knowledge and understanding of lift had to be obtained in order to simulate the contribution of fin deflection to a rocket’s roll. Similarly, the Aerospace Toolbox in Simulink was used to facilitate modeling of system dynamics. This project also presented an opportunity to implement open-source libraries while writing the software for control.