Preliminary Design Report

Wolfram Green

Rocket Team at the University of Minnesota - Twin Cities

Mentor:	Gary Stroick	952-201-3002 president@offwegorocketry.com
Faculty Advisor:	Joseph Nichols	612-626-5496 jwn@umn.edu
Student lead:	Jacob Grunklee	612-231-8773 grunk012@umn.edu

Members: Max Jetzer, Sophia Vedvik, Anna Cramblitt, Emma Lenz, Hannah Reilly, Ryan Thomas, Will Busacker, Paul Gross, Kyle Roberts, and Casey Carlson.

Table of Contents

Executive Summary

The 2017-2018 Minnesota Space Grant Consortium Midwest Competition (henceforth "Midwest Competition") challenges teams to build a high-power rocket with an active orientation-control system. The rocket must fly to 3,000ft and be safely recovered at least twice. In flight, it must be able to hold a constant roll orientation and roll to a series of predetermined compass angles. As a bonus challenge, teams may implement Xbee-based radio telemetry to communicate with the rocket both in flight and on the pad.

To meet these objectives, the Rocket Team at the University of Minnesota Twin Cities will construct a 72" long, 4" diameter rocket, dubbed "Wolfram Green." Wolfram will apogee around 3,600ft AGL on an Aerotech J800-T and is expected to experience a maximum of 13 seconds of actuating time after burnout and before apogee. Stability will exceed 1.25 calibers throughout all phases of ascent. The launch vehicle was designed and simulated in OpenRocket.

To actively control roll, pivoting elliptical canards will be actuated through a gear set by four independent servo motors. Special care will be taken to keep the rocket CG above the leading edge of the canard and to passively stabilize the canards to a vertical position in the event of a power or control failure. The servos will be driven according to a Linear Quadratic (LQ) state-space controller, a type of Linear Time-Varying (LTV) controller. This controller determines its current attitude and roll rate based on readings from a custom sensor package including IMU, magnetometer, and gyroscope. An attitude solution and control outputs are computed in a real-time environment called the Real Time Executive for Multiprocessor Systems (RTEMS), an open-source operating system platform. This operating system provides extremely precise timing, allowing the controller to actuate and log flight data at 100 Hz, while sampling sensors at up to 400 Hz. The Roll Control Module is based on a Raspberry Pi Zero microcontroller.

A second Raspberry Pi Zero will serve as Wolfram's Flight Computer. This microcontroller will relay data to and from the ground via a 2.4 GHz Xbee Pro module, actuate an 3 LEDs to indicate direction of actuation, and record video via an outward-facing Raspberry Pi Camera Module V2. A mirror system will be used to bring both the ground and LED indicator within the camera's field of view. All avionics are mounted in the coupler on a set of custom-printed, stacking PCBs.

Roll Control - Project Overview

Objective

Rocket Team has been independently developing a roll control system for over a year prior to the announcement of this competition, with the goal of building a functioning system for use on the Midwest Competition rocket in addition to a 6 inch diameter, 10-14 foot tall rocket with a target altitude of 30,000 feet. The subsystem has thus been designed to be easily transferable to other rockets designed to be stable with canards. The Midwest Competition objectives are seen as fitting within prior objectives set by Rocket Team's Roll Control Project.

To accomplish these goals the subsystem encompasses a controller, attitude determination system, supporting structural design, and rocket integration. Within each sub group there are many components that require testing and verification prior to building the subsystem. These components include, but aren't limited to, the plant dynamics, magnetometer calibration, expected torque from the canards, and structural stress.

Motivation

There are two main sources of motivation driving the Roll Control Project. The first is to provide hands-on experience in rocket attitude control.The project provides undergraduates a chance to start learning about control even before they have the opportunity to take formal classes in the subject (often restricted to junior or senior year). Also, industry is increasing their use of control systems and is looking for students who have had experience working with real systems beyond what is taught in the classroom.

A second motivation for the project is the desire for a smoother flight. Canard-based active roll control has the drawback of reducing stability, since controlling the roll rate to zero eliminates any stability gained from the phenomenon known as spin stabilization. However, reducing roll rate limits stress on the on board electronics and improves the reliability of radio communication. As an added bonus, onboard video becomes easier to view, resulting in film that highlights Rocket Team's accomplishments. Active roll control is desired by Rocket Team in all our competitions, but satisfies Midwest Competition requirements as well.

Roll Control Project Requirements

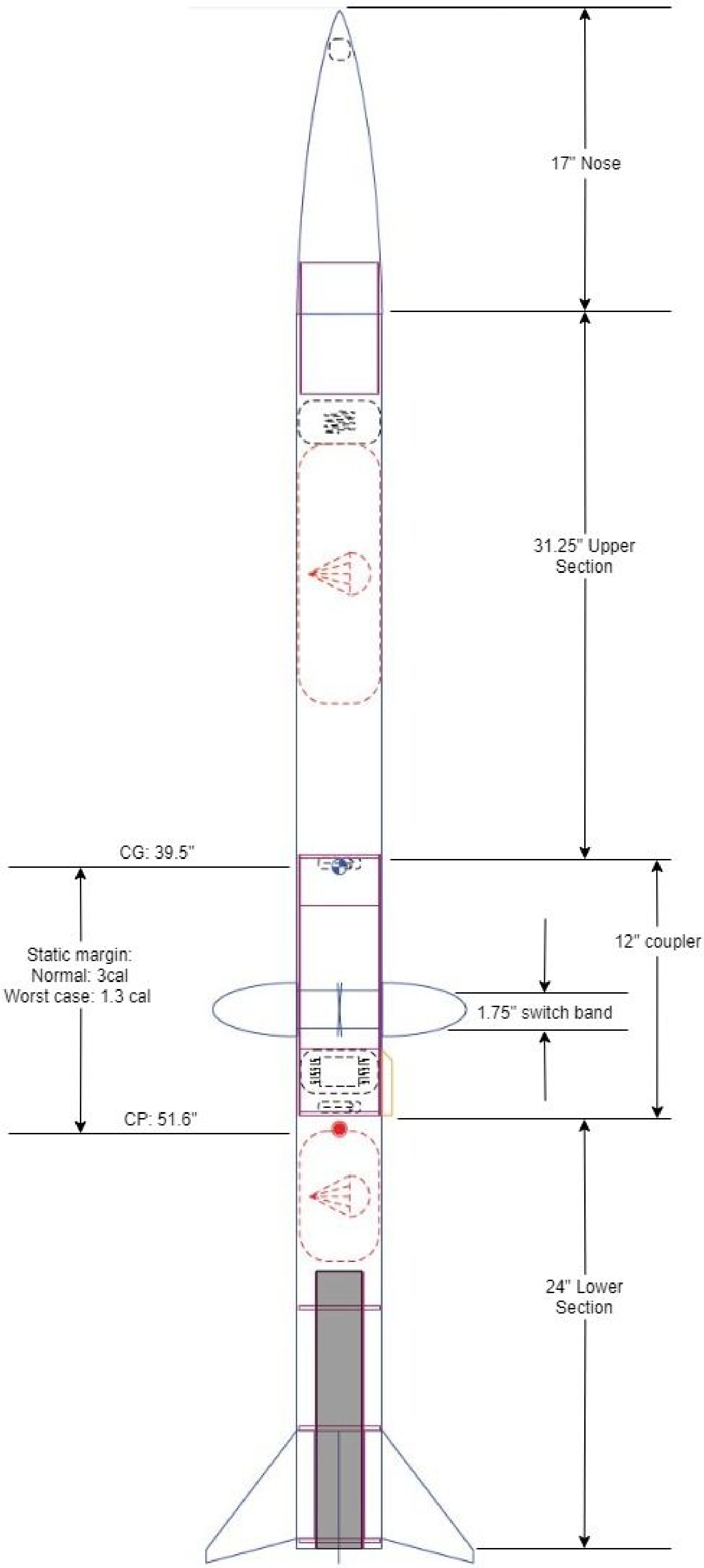
The subsystem requirements can be broken up into two groups: those that define the success criteria and minimum operating requirements, and those that define the controllable regime. These two groups shall be referred to as requirements and constraints, respectively.

Table 1: Roll Control Module Requirements	
Parameter	**Range**
Controller Frequency	[49:51] Hz
Attitude Accuracy	[-10, 10] degrees
Steady State (1s hold)	[-10, 10] degrees
Fin Deflection	[-4, 4] degrees

The requirements are listed in Table 1. Controller frequency is the speed the main control loop operates at and is controlled by RTEMS. Since this is a real time operating system, the allowable range can be small. Attitude accuracy defines how far from “truth" the attitude solution is. This affects the range of our steady state value since the accuracy defines the minimum knowledge available. The controller defines a reference direction it tries to orient the rocket towards. The steady state range defines the allowable error between the actual pointing and the reference for success. It also defines the time necessary to be within this range for successful pointing to be accomplished. The steady state requirement is more important for the Midwest competition since that requires multiple attitudes to be achieved in a single flight. The canard deflection is the maximum amount a single canard can deflect. This is both a self-imposed and hardware limitation, because the torque calculations rely on the small angle approximation, and because the servo motor has a predefined range, respectively.

Table 2: Roll Control Module Constraints	
Parameter	**Range**
Tilt angle	[0, 20] degrees
Controllable Velocity	[100, 165] m/s
Density, Range	[0, 10] %
Density, Counts	[0, 5] counts
Altitude, Range	[0, 20] %
Altitude, Counts	[0, 5] counts
Roll Rate	[0, 720] deg/s

The constraints listed in Table 2 represent the flight conditions necessary for the controller to activate. Tilt angle is the angle between the nose of the rocket and the vertical direction, which is always a positive value. For safety, the controller will cease to actuate if the tilt angle exceeds this constraint. A minimum controllable velocity is determined based upon canard geometry and approximate speed off the rail. Since the canards rely on aerodynamics to generate a torque, there is a minimum velocity needed to exceed the expected disturbance.. The upper limit is simply the maximum velocity expected during the flight. Since the density is used with the torque calculations, an acceptable range between the current value and last value must be defined. This limits the chance that a wrong value will be used. Also defined are a maximum number of times a wrong value can be computed before a sensor failure is assumed, and a maximum roll rate. The roll rate constraint prevents the controller from operating in extreme cases or if the controller fails and begins to increase the roll rate. Similarly, altitude is used to



estimate acceleration from motion, so limits must be defined for this. It is important to note that these restrictions on density, altitude, and roll rate are applied after the sensor data selection criteria. Since the data is sampled at 400 Hz but the controller only runs at 50 Hz, the sensor data can be combined to prevent a bad value from ruining the output. This is done by taking a simple average of the readings each iteration of the control loop.

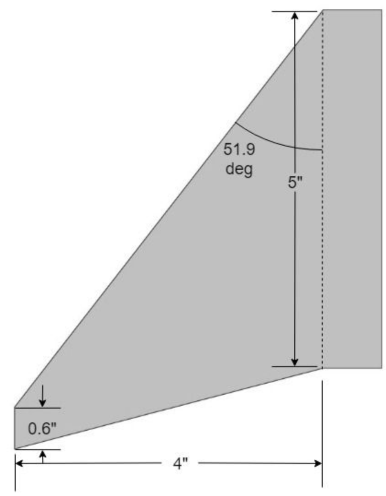
Launch Vehicle

The launch vehicle for the Midwest Competition (left) was designed, optimized, and simulated in OpenRocket. Wolfram Green will meet competition objectives by its sturdy, canvas-phenolic construction, an Aerotech J800-T motor, and a dual-deploy recovery system. Overall dimensions include a 24” lower section, housing the motor and drogue; a 2” switch band with holes for through-mounting the canard set; and a 31” upper section containing a main parachute.

Airframe

Wolfram sports a commercial wound-fiberglass nose cone, made by Wildman rocketry. The Von Karman shape profile minimizes drag for a 3.9” diameter base. Purchasing a nose was by far the simplest option; Rocket Team has laid up fiberglass nose cones in the past, but with limited success. The nose contributes 17” to the rocket’s total length.

The airframe, purchased from MAC Performance Rocketry, is 3.9” diameter, 1/16” canvas phenolic. Although Rocket Team has worked with composite tubes in the past, canvas was preferred in this case for being stronger, lighter, and more workable than fiberglass and cheaper than carbon fiber. Rocket Team has had success with canvas in previous years, particularly in the 2017 Midwest Competition. Historically, Rocket Team has competed on minimum-diameter rockets, such that the inner diameter of the airframe matched the outer diameter of the engine casing. However, the numerous electronics and mechanical components of the roll control module necessitated a larger diameter.



Stationary rear fins (right) are cut from ⅛” G10 fiberglass sheet. They will be epoxied directly to both the airframe and to the engine mount through slots in the lower tube. Fin shape parameters (shown below) were optimized to reduce drag and increase rocket stability using OpenRocket.

Propulsion

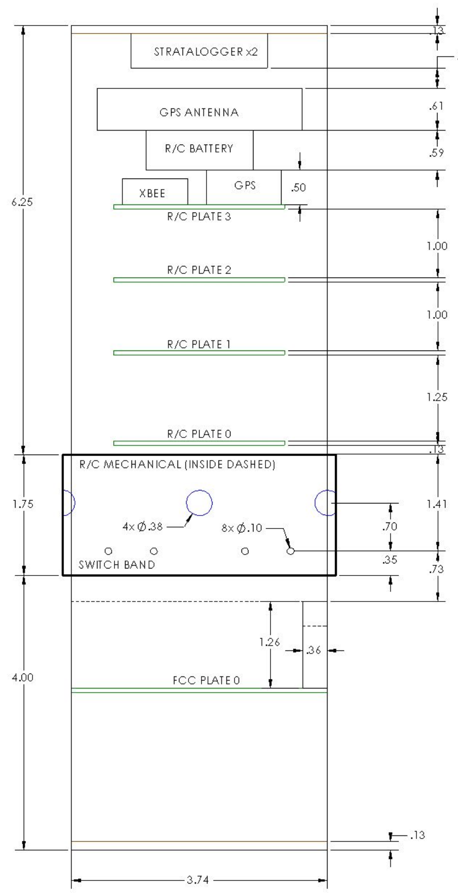
This year’s competition rewards rockets for reaching 3000 ft AGL, but no higher. The competition rules limit selection to either I- or J-class motors. At 5.9kg dry mass, Wolfram requires a motor on the upper end of this spectrum. Initially, the best motor option seemed to be the Aerotech J460T-6. However, revised weight budgeting suggested that a larger motor was needed. Although motors between the J460 and J800 have existed at some point in time, the J800 appears to be the smallest motor larger than the J460 that is currently commercially available.

Another factor considered in selecting a motor was that motors with longer and skinnier profiles would have lower moments of inertia about the roll axis compared to stubbier motors. A larger roll moment might assist the first flight challenge, helping the rocket resist changes in roll orientation. Conversely, a smaller roll moment might assist the second flight challenge. However, it was decided that a smaller roll moment would be most desirable because it will require less control effort on both flights.

Recovery System

Since it carries an especially heavy (and expensive!) payload, a functional recovery system is essential to Wolfram’s success. The rocket will utilize dually-deployed parachutes, with a 36” drogue deploying at apogee and a 66” main deploying at 750 ft AGL. [10] The first separation point will be between the lower section and coupler. The second separation will be at the nose. A redundant pair of StratoLogger Altimeters will trigger ejection charges. These will be mounted 1 each on the inside of each coupler bulkhead, both wired to ejection charges in both the upper and lower sections. Terminal blocks on the bulkheads and internal wiring will be used to limit struggles with wiring ejection chargers. The motor ejection charge will back up ejection at apogee. Both parachutes will be wrapped in flame-retardant protector sheets when packed in the rocket to prevent melting or burning.

Avionics Bay Mechanical Design



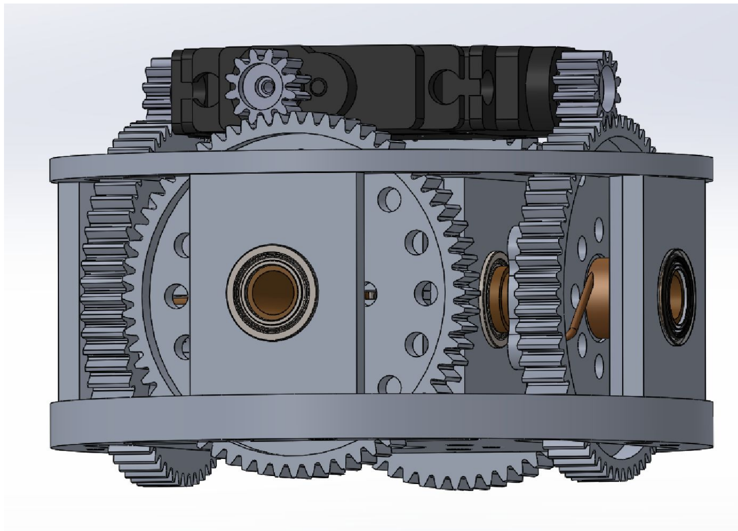
All of Wolfram's active roll components and electronics fit in a 12in coupler. These include the roll control mechanical structure, roll control PCBs, flight computer PCB, camera, and altimeters. Rocket Team typically mounts electronics on vertical (in-plane with the threaded rods that secure the bulkheads) sleds. This option is very easy to design and assemble. However, the roll control mechanical structure is cross-sectionally mounted, separating the coupler into sections. Placing electronics on a sled would therefore require a 90° interface at the servo connection point. A sled would also provide at most $24in^2$ of mountable area above the switch band, some of would be unusable due to the curvature of the tube.

Instead, the current coupler (right) layout maximizes mounting area by placing components on stacked cross-sectional plates. This configuration provides over $25in^2$ of mountable area above the switch band, nearly all of which is utilized. Everything is anchored to a pair of aluminum plates in the switch band. Above this, a Raspberry Pi Zero, custom sensor suite, batteries, and GPS antenna will mount to three $2.5in^2$ PCBs. The plates will be separated by standoffs and interconnected by extra-long male-female header pairs. A circular top plate which matches the inner diameter of the coupler will stabilize the stack laterally. Below the switch band, the Flight Computer will consist of a separate PCB mounted in similar fashion.

As a result of carefully planning the coupler physical layout (see schematic above), the rocket will meet the Midwest Competition requirement of a neat and well-organized av bay. The final product will only have discrete wires for battery, servos, and UART data between computing modules. All other components will be compactly mounted on PCBs.

Canard Set Internal Structure

The main mechanical structure of the roll control module will consist of a set of 4 steel canard fins, each mounted through the coupler wall via a shaft to a gear set, mounted to an internal steel frame (below). The current iteration builds on Rocket Team's previous working mechanism by incorporating 4 independent servo motors instead of a central one. While this adds complexity to the design, independent control contributes towards goals of the Roll Control Project.



To construct this component, we will purchase most components including gears, springs, bushings, and screws. However, many components including the main plates, servo brackets, fins and fin shafts will be machined the team.

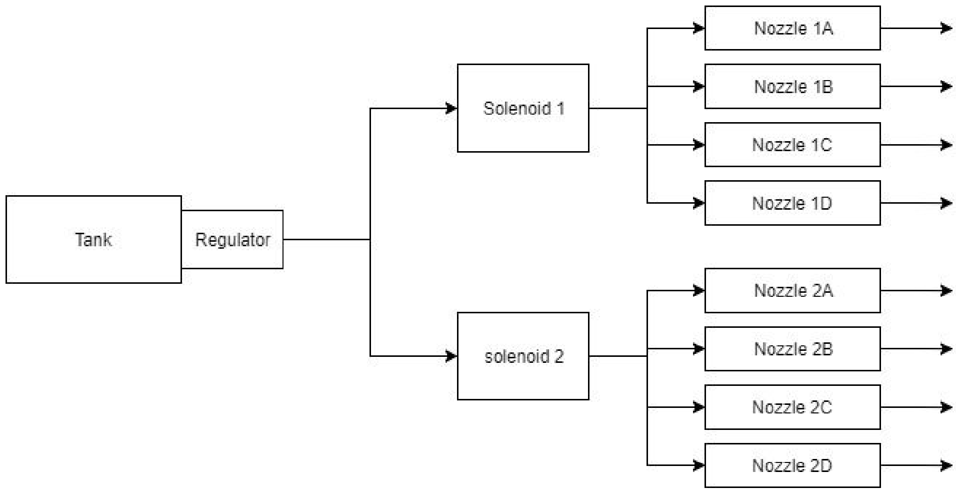
Finding a way to securely attach the fins after inserting the structure in the coupler has proven a major design challenge. One previous solution (permanently attaching the fins) required cutting long insertion slits in the coupler, which severely compromised the rocket's structure. Instead, the current design uses a set screw to attach the canards after the structure is in the coupler.

The next issue comes from the gearing; small misalignments cause the gears to have extra freedom known as "slop". This must be minimized since there is only a total range of 8 degrees of deflection. To accomplish this, the number of gears has been minimized, and the ratio maximized to achieve the required torque. Springs will also be used to preload the gears.

Alternate Plan - RCS Thruster System

In order to have an alternative actuation method for controlling roll, our team is independently developing a compressed-air thruster system. Virtually no previous work has been done by Rocket Team on this type of system, so it is currently the backup plan. The thruster system is also bulkier and less likely to exert an effective control torque compared to the fin system. However, it could prove more efficient at greater altitudes where fins are less effective.

The thruster system will be fed from a high-pressure compressed air tank. Such tanks are commercially available in light, low-profile packages, often marketed for high-performance paintball. Our tank is rated to hold 45 cm^3 of air at a maximum of 4500 psi. It was purchased with a regulator that brings output pressure down to 450 psi. From the regulator, air passes through a series of high pressure tubes and splitters that carry air to a pair of nozzle sets. An electronic solenoid valves will control each set. See schematic below:



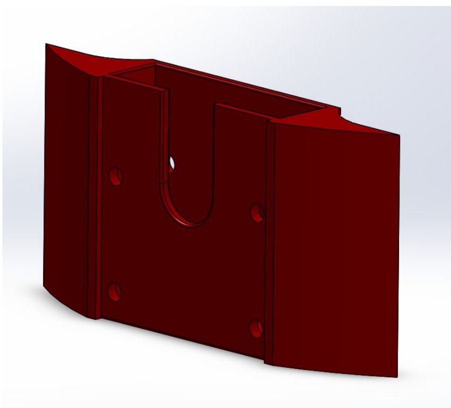
Thus far, research has been conducted to find any sort of work that has been done previously on this type of thruster system. Most of the previous work found was used for orbital attitude control of satellites, but we have still found a useful amount of information for our purposes from other groups. The initial design will not necessarily be capable of fitting in a 4in diameter rocket. The purpose of this prototype will be to test our models of the system and to see how accurately the performance of the system can be predicted. In the future, a focus will be placed on miniaturizing the system and making it more feasible for use in a rocket.

The next steps that need to be taken include finalizing the parts that need to be ordered and machining parts that cannot be ordered, such as nozzles. After obtaining the necessary parts, we will rig a system to test the impulse generated by our thrusters and solenoid actuation speeds, and thus we will know what sort of moments we can generate on a rocket. This data can then be used to confirm or update the models.

If this thruster system is to be used in the future, it is important to have sufficient testing and models to reliably predict flight performance. If this system is to replace the usual fin system on a rocket, we cannot sacrifice reliability or effectiveness. Otherwise there is no reason to use this type of system. It is important to maximize the efficiency of this thruster system, through testing and modelling, so it can improve the performance of the rocket.

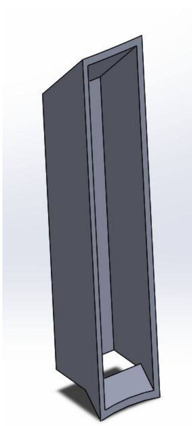
DREAM

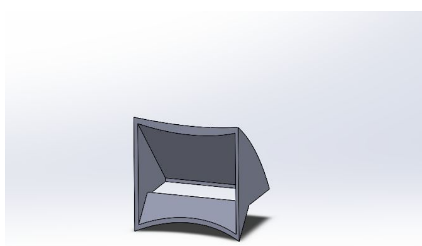
This year’s challenge requires the rocket to equip a down-facing camera and monitor control efforts by viewing a set of indicator LEDs. Our solution, the Dual Reflective Effect of Actuation Monitor (DREAM) uses a set of two mirrors to keep all electronics inside the coupler. This is desirable since our flight computer will be a Raspberry Pi Zero (see Electrical Design), which is built to interface with the Raspberry Pi Camera Module V2 via a short ribbon cable. The V2 is one inch square, helping minimize DREAM's size inside the rocket. The edges of the mount are rounded to be flush with the interior of the rocket. The camera is secured into the mount, which will be epoxied to the coupler interior.

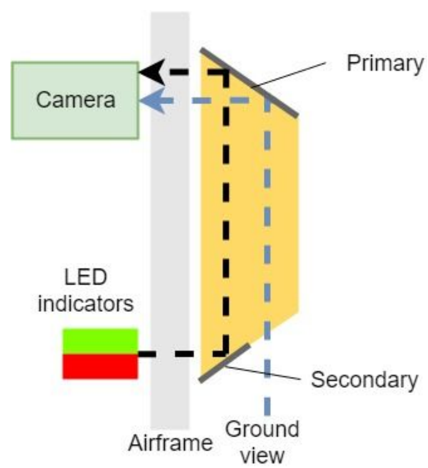


The camera sits in a mount (left) facing outwards. DREAM uses a primary mirror to redirect the field of vision, with a smaller secondary mirror to reflect back the image of the LEDs. Both of these mirrors will make 45 degree angles with the rocket. The rightmost image below is a schematic view of our mirror-camera system, showing how the camera will be able to capture images of the ground and the LEDs with the assistance of DREAM.

Below on the left is the first design of the mirror mount. We originally designed the mirror mount to have a length of 3.28", width of 0.57", and protrude 0.6" from the rocket. After taking test videos with this design, we found the field of view to be extremely small, so we increased its width to 1.5" and protruding dimension to 1.2". We also removed the long tube between the primary and secondary mirror. The middle image is a model of our final mirror mount design. The mounts were designed in SolidWorks and 3D printed in PLA Plastic.







We also considered alternative camera and mirror system designs. One design was to use a prism since we've had success with them in the past. However, a mirror would provide a clearer image. We also discussed placing the indicator LEDs on the outside of the rocket and using a Mobius to mount the camera, which would eliminate the need for mirrors. However, this design greatly complicated the electrical wiring to the LEDs because of its extension outside the body of the rocket. Because of this, the costs of this design outweighed its benefits, so the DREAM became a reality.

Electrical Hardware Design

The electrical design of Wolfram Green is divided into 2 distinct computing modules: the Roll Control Module and the Flight Computer. As mentioned, the Roll Control Module strictly performs data collection, computation and actuation of the canard set. The flight computer fulfils the remaining competition objectives including downward video and radio telemetry. The Flight Computer thus helps maintain the modularity of the roll control system (a Roll Control Project goal). It also reduces computational overhead on the roll control microcontroller. Furthermore, producing multiple systems allows more members to gain electronics experience.

Roll Control Module

The Roll Control Module utilizes a Raspberry Pi Zero as its main flight computer with a Teensy 3.2 as a supporting microcontroller. The Pi is responsible for the state machine and general executive tasks. This includes the attitude determination system, the controller, obtaining sensor data, and logging data to a GPS. It sends some key information to the Teensy for communication with the Flight Computer and servo control. Meanwhile, the Teensy gets data from the GPS and Flight Computer, controls the servos, and relays the information to the Pi and Flight Computer.

The sensor package consists of an Inertial Measurement Unit (IMU) that has triads of rate gyros, magnetometers, and accelerometers. On board is also a pressure sensor, temperature sensor, and GPS. All sensors feed into the attitude determination algorithm, and the gyro data also gets sent to the control algorithm. Detailed specifications for the sensors used can be found in their respective data sheets, but only the part numbers are listed below for brevity.

Table X: Roll Control Module Hardware

Item	Part Number	Item	Part Number
Accelerometer	ADXL 377	Temperature	LM61BIZ
Gyroscope	L3GD20H	GPS	Copernicus II
Magnetometer	LSM9DS1	Servo	HS-82MG
Pressure	HSCMNNN1.6BAAA3		

Flight Computer

The Flight Computer is a separately functioning Raspberry Pi Zero that handles radio up- and down-link, video recording, plus backup sensing and data logging. Onboard sensors will include IMU, vibration, and photoresistors. The BNO55 IMU will be used to cross-check the attitude solution from Roll Control. Adafruit Vibration sensors are included primarily as a “research” interest. This information will be logged but not broadcast to the ground. The photoresistors will detect light that enters the rocket as a result of separation at apogee. This information is useful to cross-check against other flight data. The Flight Computer also handles telemetry for the "bonus" portion of the competition (See Bonus Challenge Section).

Controller Design

As part of the Roll Control Project, a Linear-Quadratic-Regulator (LQR) controller is being designed and implemented. The controller outputs will be realized using canards and constrained by the aerodynamic forces experienced throughout flight. Since aerodynamic forces are velocity dependent, the controller must be robust to plant disturbances. The following sections outline the plant dynamics, the current approach to the controller design, and results from the most recent testing. Quaternions are used throughout this paper, and are defined as $q = [q_0,\ q_{vec}]^T$ with q_0 being the scalar term. This report assumes the reader is familiar with quaternions and their math, a more detailed discussion can be found in [5].

Plant Dynamics

The rocket attitude model is described by its orientation (quaternions) and angular velocities (rigid body dynamics). The quaternion attitude kinematics are given by

$$\dot{\mathbf{q}} = \frac{1}{2}\Xi(\mathbf{q})\omega = \frac{1}{2}\Omega(\omega)\mathbf{q}$$

quaternions, where $\Omega(\omega)$ is given by

$$\Omega(\omega) = \begin{bmatrix} 0 & -w_x & -w_y & -w_z \\ w_x & 0 & w_z & -w_y \\ w_y & -w_z & 0 & w_x \\ w_z & w_y & -w_x & 0 \end{bmatrix}$$

and ω is the angular velocity of the body-fixed frame with respect to the inertial frame with components $\omega = [\omega_x,\ \omega_y,\ \omega_z]^T$. The quaternion describes the orientation of the spacecraft using a body-fixed frame with respect to the inertial frame. The dynamics for a rigid body rocket are given by

$$J\dot{\omega} = [\omega\times]\,J\omega + \mathbf{L},$$

where J is the moment of inertia matrix, L is the applied torque, and $[\omega\,\times]$ is a skew-symmetric matrix of angular velocity components given by

$$[\omega\times] = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix}$$

These equations are then combined and written in the standard state space form,

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$$
$$\mathbf{y} = C\mathbf{x} + D\mathbf{u}$$

With our state vector x being given by

$$x = \begin{bmatrix} w_1 & w_2 & w_3 & q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\mathsf{T}$$

The rigid body dynamics and quaternion altitude kinematics are combined to create the A and B matrices, they compose a state space representation of the plant. Due to the nonlinear nature of dynamics and quaternions, they are linearized using the equation given below.

$$\delta\dot{\mathbf{x}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\delta\mathbf{x} + \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\delta\mathbf{u}$$

This produces the following A matrix, which is a function of the state vector, meaning not only is the problem nonlinear but also time varying.

$$A(\mathbf{x}) = \begin{bmatrix} 0 & \frac{I_2-I_3}{I_1}\omega_3 & \frac{I_2-I_3}{I_1}\omega_2 & 0 & 0 & 0 & 0 \\ \frac{I_3-I_1}{I_2}\omega_3 & 0 & \frac{I_3-I_1}{I_2}\omega_1 & 0 & 0 & 0 & 0 \\ \frac{I_1-I_2}{I_3}\omega_2 & \frac{I_1-I_2}{I_3}\omega_1 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2}q_1 & -\frac{1}{2}q_2 & -\frac{1}{2}q_3 & 0 & -\frac{1}{2}w_1 & -\frac{1}{2}w_2 & -\frac{1}{2}w_3 \\ \frac{1}{2}q_0 & -\frac{1}{2}q_3 & -\frac{1}{2}q_2 & \frac{1}{2}w_1 & 0 & \frac{1}{2}w_3 & -\frac{1}{2}w_2 \\ \frac{1}{2}q_3 & \frac{1}{2}q_0 & -\frac{1}{2}q_1 & \frac{1}{2}w_2 & -\frac{1}{2}w_3 & 0 & \frac{1}{2}w_1 \\ -\frac{1}{2}q_2 & \frac{1}{2}q_1 & \frac{1}{2}q_0 & \frac{1}{2}w_3 & \frac{1}{2}w_2 & -\frac{1}{2}w_1 & 0 \end{bmatrix}$$

The calculated B matrix is constant, as are C and D.

$$B = \begin{bmatrix} \frac{1}{I_1} & 0 & 0 \\ 0 & \frac{1}{I_2} & 0 \\ 0 & 0 & \frac{1}{I_3} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

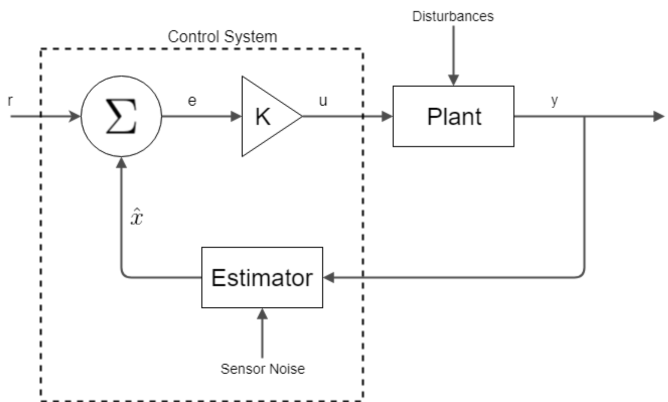
The C and D matrices are defined as

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Our control input u is given by

$$u = \begin{bmatrix} T_1 & T_2 & T_3 \end{bmatrix}$$

and comes from the canard fins. These controller inputs are translated into a fin deflection angle that is then given to servos. This forms the basis of the control loop outlined in the block diagram.



Where the plant encompasses the dynamics as well as the actuators. Then the estimator block represents the full state estimator (discussed in greater detail later in the report).

Approach

In order to track a reference state, an LQR controller is used. Full state feedback is assumed because a full state estimator (mechanized by an extended Kalman filter) has been designed to accomplish this goal. For testing purposes the estimator has not been included. However, once a controller has been finalized, it will be tested with the estimator to ensure the system remains controllable. Since this controller has no integrator there will be some steady state error however, this has been observed to be small during initial testing; a result to be verified through flight tests.

Attitude Determination System

The algorithm used for these missions is based off the paper published by Professor Gebre-Egziabher from the University of Minnesota on attitude determination for Micro Aerial

Vehicles (MAV). Some adaptations have been made to account for use in high acceleration applications, and a short summary of the methods will be given here but for a detailed description see [1]. This method works by optimally combining information from rate gyros with accelerometer and magnetometer data. To accomplish this an extended Kalman filter[1] is used to obtain the attitude solution, where the measurement update calculates a correction rotation to be applied by using a vector matching technique. Because of the vector matching relies on a measurement of gravity, the acceleration from motion must be removed from the accelerometer measurements. This leads to the adaptations implemented since a rocket will undergo acceleration much greater than that of gravity.

First, the dynamics are propagated using three rate gyros to integrate the attitude quaternion by equations below, as explained in [3].

$$\dot{\mathbf{q}}_k = \frac{1}{2}[\omega^{\times}]\mathbf{q}_k$$

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \Delta t \cdot \dot{\mathbf{q}}_k$$

Where $[\omega^x]$ is a skew symmetric matrix given by the following.

$$[\omega^{\times}] = \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix}$$

Then the state transition matrix Φ is constructed along with the discrete process noise C_d.The state transition matrix is simply given by a matrix exponential as seen in the following.

$$\Phi = e^{\Delta t F}$$

Next the covariance is propagated by the equation given below. This tends to increase the uncertainty since it is only based on the system dynamics. Propagating the quaternion and covariance is done every time the determination algorithm is run, but the remaining portion of the algorithm can be by passed in case of loss of GPS, and magnetometer or accelerometer.

$$P_k^- = \Phi P_k \Phi^T + C_d$$

This is where the high acceleration must be accounted for, and also where the algorithm deviates from that outlined in the [1]. To accomplish this a Kalman filter was designed to optimally combine information from a pressure sensor, temperature sensor, and GPS. It works by calculating the altitude from pressure and temperature readings then combining this information with GPS velocity to obtain acceleration estimates. Again, a full discussion of Kalman filtering has been omitted but can be found in [2]. To do this the dynamics are given by the following.

---

[1] See [2] for a discussion of Kalman filtering

$$F = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ & 0_{3\times3} & & 0 & & I_{3\times3} & \\ & 0_{3\times3} & & 0 & & I_{3\times3} & \end{bmatrix}$$

Where the acceleration is modeled as uncorrelated process noise because the underlying model for jerk is unknown. This requires the filter to be tested independently to obtain reasonable values for the Q matrix which has been determined to be,

$$F = \begin{bmatrix} 0_{4\times4} & & 0_{4\times3} & \\ & 10 & & \\ 0_{3\times4} & & 10 & \\ & & & 10 \end{bmatrix}$$

The state dynamics have then been descritized and used to calculate the Kalman gain and obtain acceleration estimates similar to the process described in the larger extended Kalman filter. Test results can be found in a later section. It's important to note that the dynamics are only valid in an inertial reference frame, and thus, the Earth Centered Inertial (ECI) frame is used. This also means the resulting estimate must be rotated into the body frame before it can be removed from the measured acceleration.

The reference magnetic field (this comes from the International Geomagnetic Field Reference Model, for details see [4]) and accelerometer vector are rotated into the body frame using the quaternion obtained from the gyro integration to get an estimate for comparison to the sensor readings. Then the estimated acceleration from motion is removed from the measurement in the body frame to obtain the sensors measurement of gravitational acceleration. Next the difference between the sensor readings and the measurements is found, this is used as the effective measurement, z .Then the sensor noise covariance, R_z , is found, and the Kalman gain is calculated using

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_z)^{-1}$$

Then the state and covariance are updated with the effective measurements (equations given below). Since the update calculates a small correction, q_e , to the attitude obtained from the gyros, the small angle approximation is used, for quaternions this means $q_0 = 1$. Then q_e is combined with q and the gyro biases are added to get the final state (equations below).

$$P_{k+1} = (I_{6x6} - K_k H_k) P_k^-$$

$$\begin{bmatrix} \mathbf{q}_{e1:3} \\ \mathbf{b_e} \end{bmatrix} = K\mathbf{z}$$

$$\mathbf{q}_{k+1} = \mathbf{q}_k \otimes \mathbf{q}_e$$

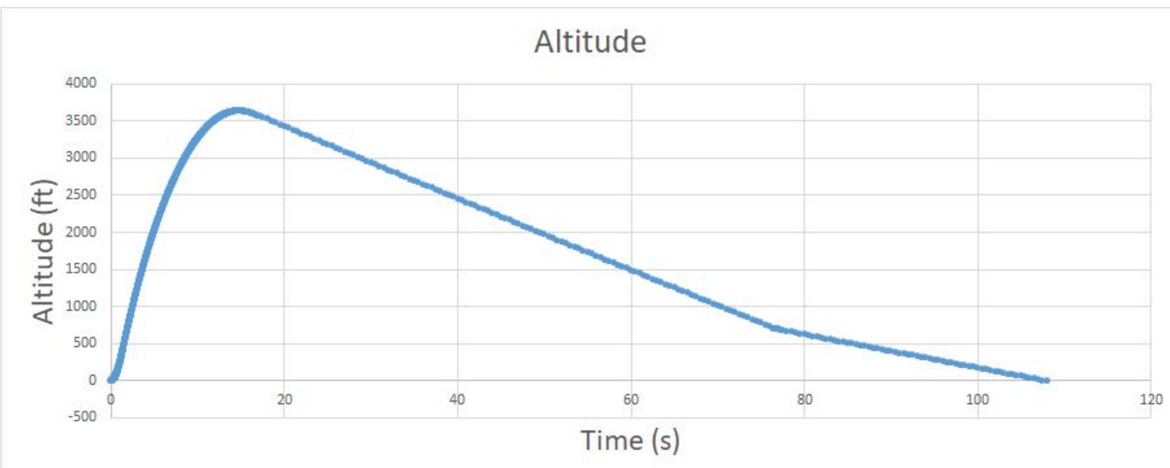
$$\mathbf{b}_{k+1} = \mathbf{b}_k + \mathbf{b}_e$$

Predicted Performance

Flight Profile

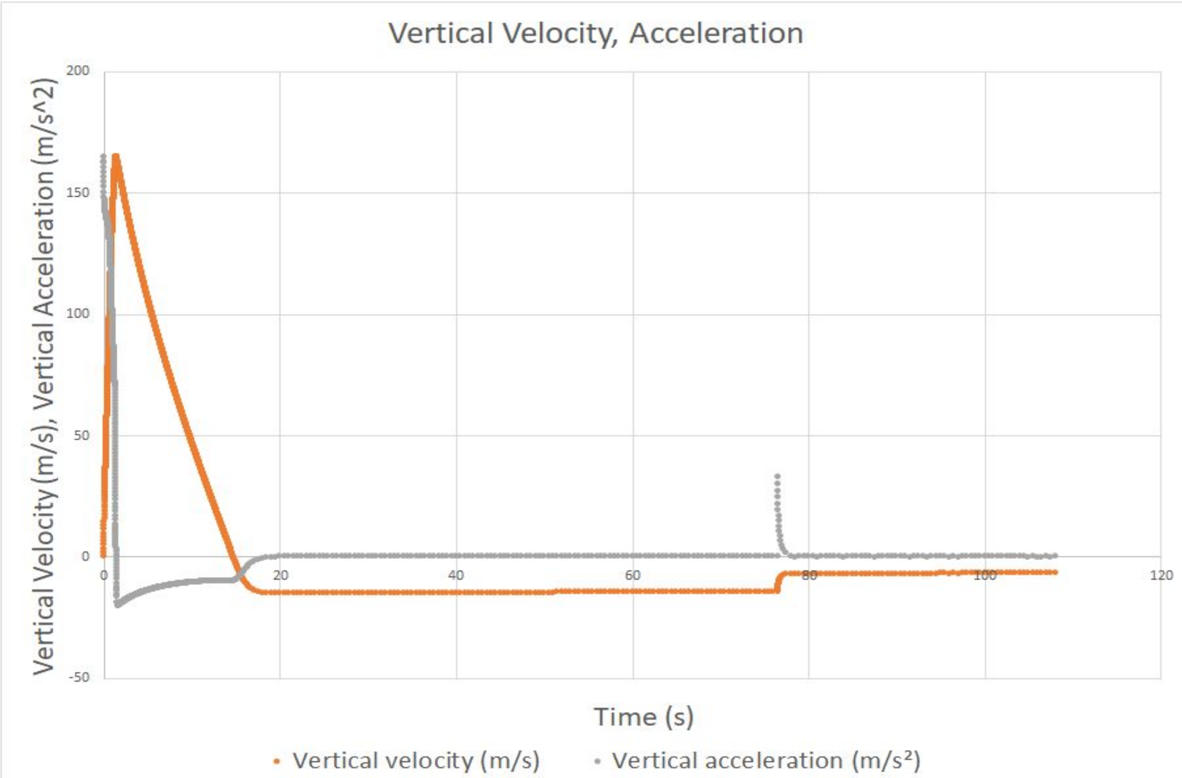
Wolfram Green's expected flight profile has been simulated in OpenRocket. Flying on a J800-T motor, apogee is simulated to occur around 3,600ft. This altitude allows a very wide margin of error above the competition requirement of 3,000ft, which is desirable in case of poor flight conditions or unanticipated weight additions. On March 17th, we plan to have a test launch in order to assess the reliability and accuracy of our roll control system.

Peak velocity is simulated at 165 m/s or mach 0.49. For the purposes of creating a sufficient control torque with the actuating canards, even higher velocities would be desirable. We hope to achieve this by minimizing mass when possible during construction. On descent, the rocket is expected to expected to fall at about 48 ft/s under drogue. After main deploys, the rocket will slow to about 18ft/s until touchdown (used [10]). This is safely within the competition requirement of less than 24 ft/s descent.



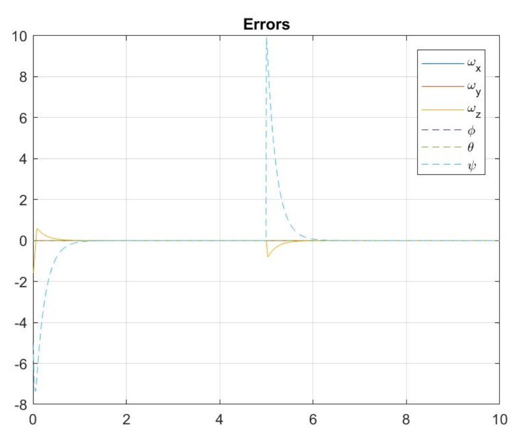
Vertical acceleration peaks at 165 m/s^2, or 16.8 Gs. This peak acceleration is not expected to overstress any rocket components. The thrust-to-weight ratio off the rail is calculated to be about 15, safely above competition requirements.

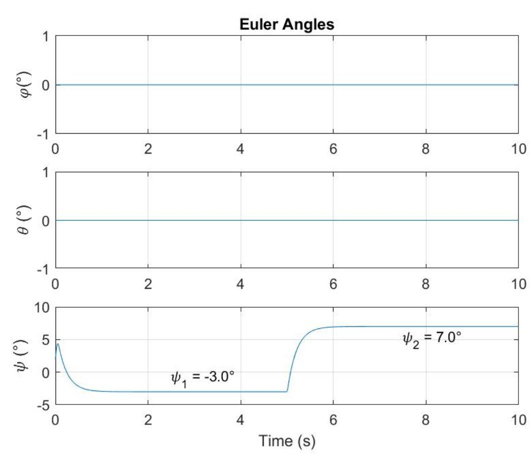
Also of interest in regards to roll control are the rocket's simulated worst-case roll rate due to imprecise construction. To simulate this, rear fins were canted 0.25° and the simulation run in OpenRocket. The canard cant necessary to overcome the resulting torque was determined to be within the 4° maximum canard deflection angle.



Controller

The left plot shows the difference between the reference commands and the real state of the system. A working controller would push these to zero, and as can be observed they are close to zero. There is some steady state error but this was expected since no integrator was present. In the simulations the error is observed to be small but flight testing is needed to assess real world performance.

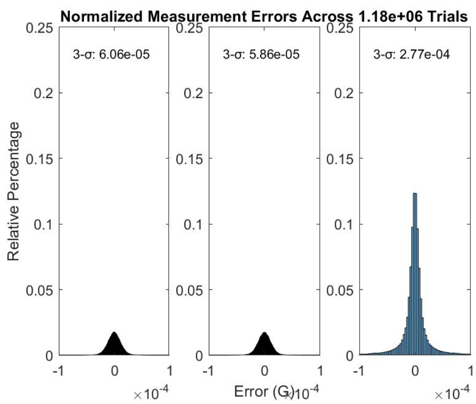




As can be seen from the right most plot, 2 reference points were used and the system managed to track both. It can also be seen that the controller has a settling time of less than 1 second. This is ideal since the controllable time is short and multiple references must be tracked.

Work has started on other methods such as an LQR with integral action [6] [8], or an iterative LQR [7] to remove the steady state error but at the moment it is unclear if these will be utilized for the competition since the error is already small. An initial study appears to reveal the system is unstabilizabled when the integrator gets augmented to state but further review is required before it is eliminated from consideration.

Magnetometer Calibration

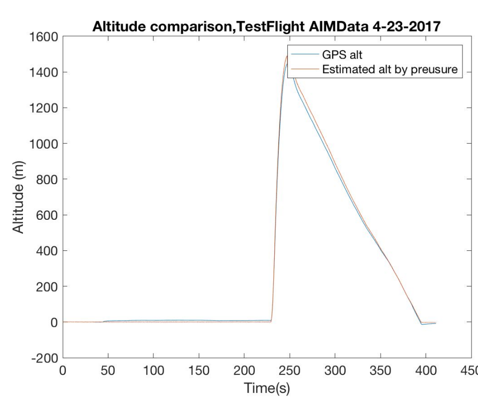


Another important aspect of the attitude determination algorithm is magnetometer calibration. The purpose of this is to account for environmental error in the local magnetic field, these come from ferromagnetic materials near the sensor, as well as changing magnetic fields generated by other electronics or radios. A full discussion of the technique used here can be found in [9], only the independent test results are presented here.

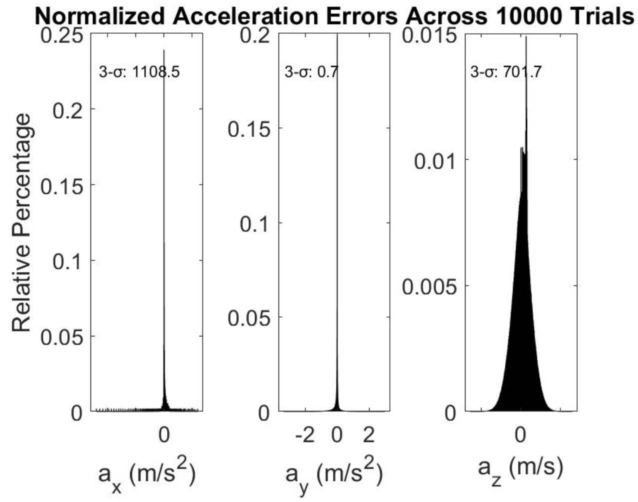
Synthetic measurements were generated and Monte Carlo simulations were conducted to characterize the errors for each axis. As shown in the figure above, each set of normalized errors has a mean of zero. They also have a 99.7% confidence bound on the order of 10^{-4} gauss which is on the order of 10s of nano Tesla. For comparison, the Earth's magnetic field has a strength on the order of 10^{4} nano Tesla along a compass direction. This provides an acceptable amount of error on each sensor, and will improve the overall accuracy of the attitude determination algorithm.

Altitude Measurements

In order to estimate acceleration, position must be known. Because GPS is slow, pressure and temperature are used to calculate altitude, and this process must be verified. The figure outlines the true and estimated altitudes as measured above ground level. The observed difference is small compared to apogee. These estimates were taken on their own to characterize the performance of the method and associated process noise. It will later be combined with the Kalman filter to obtain estimates for acceleration. The Kalman filter will remove the sensor noise and model the process noise to obtain an optimal estimate of the acceleration.



Acceleration Estimation



Since the attitude determination relies on vector matching with the accelerometer it's necessary for the sensor to be approximately stationary. In the case of the rocket the acceleration from motion must be carefully estimated in order to remove this from the sensor measurements. A Kalman filter was designed to combine altitude and velocity measurements while modeling acceleration as uncorrelated process noise. The results of Monte Carlo simulations can be seen in the figure. While the x-axis appears to be an anomaly, this could be due to how the simulator supplied the data. Of note is the z-axis acceleration as this experiences the most acceleration throughout the flight. The errors are zero mean, and a 99.7% confidence of 701.7 m/s^2 which is approximately 3.5 times the max acceleration expected. Since the GPS only updates at 1 Hz the majority of the corrections come from the noisey altitude estimates. Without more advanced filtering or more expensive sensors estimating such high acceleration is expected to be problematic. Simulations with the full attitude determination system will determine if this range is acceptable.

Safety

Planned Av Bay Assembly Procedure

1. Attach verticals to aluminum baseplate. Mount bushings and gears.
2. Attach aluminum top plate and servo brackets.
3. Attach servos, threading wires to top of system.
4. Plug in servos and attach upper connection plate.
5. Attach lower connection plate. Mount camera, screw switches and flight computer.
6. Connect screw switch wires to connection plate and flight computer.
7. Fit unit into coupler and insert screws. Insert threaded rods.
8. Plug assembled roll control pcb stack into connection plate.
9. Test all battery voltages.
10. Mount and connect batteries to terminal blocks.
11. Thread ejection charge wires through the stack, attach to Stratologgers.
12. Thread separation detection through lower bulkhead.
13. Cap coupler with bulkheads and screw down the threaded rod.

Pre-flight Checklist

1. Inspect all shock cord connection points for upper and lower sections.
2. Inspect all wiring connections and zip-tie attachments.
3. Power altimeters on to test proper wiring and deploy settings.
4. Power all avionics off.
5. Wire all charges to the terminals.
6. Screw upper section to coupler.
7. Fold main parachute into upper body tube and tape tracker to shock cord.
8. Insert recovery into upper section.
9. Insert nose and upper shear pins.
10. Insert coupler in lower section. Ensure ensure camera is fixed and faces primary mirror.
11. Insert lower shear pins.
12. Assemble and install motor.
13. Place rocket on launch rail.
14. Power up Raspberry Pi’s.
15. Arm upper Stratologger, waiting for proper beep signals (three sequential beeps)
16. Arm lower Stratologger, waiting for proper beep signals (three sequential beeps)
17. Visual inspection
18. Photograph(s) of rocket on pad.
19. Photograph(s) of team members with rocket.
20. Launch!

Post-flight Checklist

1. Locate rocket and inspect for any damage.
2. Check that there are no remaining live charges.
3. Take a picture of the rocket.
4. Power off electronics.
5. Disconnect spent charges from electronics.
6. Bring recovered rocket to post flight check in table.
7. Give Altimeter 2 to competition official.
8. Retrieve and deliver relevant flight data to competition officials.

Model Rocket Flights

New members of the Rocket Team participated in a program called the Grand Opening Project to Help Educate Rocketeers, or GOPHER. In this program, the new members were split into teams, each of which was assigned an experienced team member to be their mentor. These teams then built L1 class dual-deployment high power rockets. These rockets flew last October (image below) on Aerotech DMS I280 Dark Matter motors. All rockets were recovered!


Budget

Item	Cost	Quantity	Subtotal
Flight Computer			
Raspberry Pi Zero	$5.00	2	$10.00
RPi v2 Camera Module	$29.99	1	$29.99
2.4GHz Xbee Pro Radio Module	$32.00	4	$128.00
12” x 12” acrylic-mylar mirror	$8.99	1	$8.99
Roll Control Mechanical Structure			
Belleville disc spring, (94065K26)	$2.70	1	$2.70
Torsion spring, 90 deg. LHT, 2. 15 in-lbs, pack of 6(9271K584)	$5.01	2	$10.02
Metal for machined parts	$83.00	1	$83.00
Hub gear, 32P, 64T(615194)	$12.99	4	$51.96
Face tapped clamping hub, 0.770” pattern, 5/16’ bore(545592)	$5.99	4	$23.96
Flanged ball bearing, 5/16" ID, 1/2" OD, pack of 2(535046)	$1.99	5	$9.95
HS-82MG Servo(32082S00)	$19.99	4	$79.96
32P, 24T C1 Spline Servo Mount Gear(615274)	$14.99	4	$59.96
Roll Control Electronics			
Screw switches	$2.95	6	$17.70
Blue LED	$0.52	25	$13.00
Red LED	$0.39	25	$9.75
Green LED	$0.52	25	$13.00
Gyro	$12.50	1	$12.50
Xbee Headers	$0.95	3	$2.85
Tall back plane headers	$2.95	20	$59.00
RPi headers	$1.50	2	$3.00
Mag	$14.95	3	$44.85
Micro SD card headers	$3.95	2	$7.90
⅝” standoffs	$0.56	25	$14.00
1” standoffs	$0.60	40	$24.00
3.8K Resistor ¼ W Axial	$0.10	10	$1.00

Right angle headers	$5.53	3	$16.59
Pressure sensor	$35.48	3	$106.44
2x20 Male Headers	$0.71	15	$10.65
LM1085 ADJ	$1.92	5	$9.60
12 Bit ADC	$2.82	10	$28.20
1K Resistors ¼ W Axial	$0.10	100	$10.00
1M Resistors ¼ W Axial	$0.10	50	$5.00
Battery Holder	$4.48	2	$8.96
Heat Sinks	$0.21	10	$2.10
Backplane TFSM	$2.95	5	$14.75
SMA Crimp Connector	$2.23	2	$4.46
2Pin Jumper	$0.10	10	$1.00
4-40 Screws (⅜)	$1.60	1	$1.60
4-40 Screws (⅝)	$2.10	1	$2.10
6-32 Nuts	1.24	1	1.24
Teensy 3.2	$19.80	3	$59.40
Copernicus II	$74.95	1	$74.95
Standard Female Headers	$1.50	10	$15.00
GPS Antenna	$12.95	1	$12.95
18650 Battery	$6.90	8	$55.20
18650 Charger	$13.00	2	$26.00
6-32 Threaded Rod	$3.89	2	$7.78
Launch Vehicle			
4in phenolic body tube	2	$84.95	$169.90
4in phenolic coupler	1	$13.00	$13.00
12" long x 3.9in dia fiberglass coupler	12	$2.60	$31.20
Wildman 4in nose cone	1	$69.00	$69.00
G10 fiberglass sheet	1	$30.00	$30.00
Commercial nose cone	1	$69.00	$69.00
54mm canvas phenolic motor mount	2	$17.35	$34.70
Propulsion			
Aerotech J850 test motors	2	$92.99	$185.98
Aerotech J850 competition motors	2	$92.99	$185.98
Aerotech 54/1280 motor casing	1	$95.00	$95.00
54mm forward closure	1	$45.00	$45.00
General			
Competition fee	1	$400.00	$400.00
Lodging	3	$600.00	$600.00
Van Rental	2	$416.00	$416.00
Shipping	1	$100.00	$100.00
GRAND TOTAL:			**$3,639.77**

References

[1] D. Gebre-Egziabher and G. H. Elkaim, “Mav attitude determination by vector matching,” IEEE Transactions on Aerospace and Electronic Systems, vol. 44, pp. 1012–1028, July 2008.

[2] J. L. Crassidis and J. L. Junkins, Optimal Estimation of Dynamic Systems. Boca Raton, FL : CRC Press, 2 ed., 2012.

[3] R. W. Beard, Small Unmanned Aircraft: Theory and Practice, pp. 254–259. Princeton University Press, 2012.

[4] E. Thebault and etc. all, “International geomagnetic reference field: the 12th generation,” Earth, Planets and Space, pp. 1–19, 2015.

[5] J. Diebel, “Representing attitude: Euler angles, unit quaternions, and rotation vectors,” tech. rep., Stanford University, October 2006.

[6] F. Zheng-ping, Z. Ji-mao, and R. Allen, “Design of continuous and discrete lqi control systems with stable inner loops,” Journal of Sjamgjao Jiaotong University, vol. E-12, no. 6, pp. 787–792, 2007.

[7] International Conference on Informatics in Control, Automation, and Robotics, Iterative linear Quadratic Regulator Design for Nonlinear Biological Movement Systems.

[8] K. J. Astrom and R. M. Murray, Feedback Systems, pp. 7.1–7.35. Princeton University Press, October 2016.

[9] D. Gebre-Egziabher, “Magnetometer autocalibration leveraging measurement locus constraints,” Journal of Aircraft, vol. 44, pp. 1361–1368, July-August 2007.

[10] “Parachute Descent Rate Calculator,” Fruity Chutes. Accessed 3 March 2018. https://fruitychutes.com/help_for_parachutes/parachute-descent-rate-calculator.htm.

Communications Challenge

Objective

This year's special communication challenge requires incorporation of a 2.4 GHz XBee Pro radio module on the non-commercial sensor suite. Rocket Team will implement all three bonus challenge options, namely: A) transmitting alternate orientation commands to the rocket while it is on the pad; B) transmitting orientation information to a ground station while the rocket is in flight; and C) sending commands and receiving replies from the rocket during flight.

Hardware Design

The 2.4GHz Xbee Pro Module is available in available in two main versions, one which supports WiFi mesh networks and one supporting only point-point or point-multipoint communication. For simplicity, the point-point Xbee Pro Module was selected. This module was also cheaper.

The flight computer (as opposed to the roll control module) will handle all bonus challenge tasks. This device will consist of a Raspberry Pi Zero microcontroller. The Xbee, Pi and other flight computer sensors will be mounted to a single custom-printed circuit board. These sensors alone will not be sufficient to accurately determine rocket attitude, so the flight computer will receive attitude and roll data from the roll control module via a wired UART connection.

Software Design

Rocket Team has two primary concerns in regards to the software side of the communication challenge. The first is signal interference due to too many Xbees on the same bandwidth. The best way to mitigate this problem might be to leave all other teams' Xbees (and other WiFi devices) powered off. Otherwise not much can be done to mitigate this issue.

Our second concern regards the 2-byte "security code," which we plan to supplement with a third byte in in the upper 128-256 range (extended ASCII), both on up- and downlinks.

Bonus Challenge A: Reprogramming on the Pad

The first component of the bonus challenge requires the roll orientation instructions to be overwritten after power-up while the rocket sits on the pad. Instructions will be provided at launch. A python script will be used in the ground station to enter ot parse these commands before sending them to the rocket. Reprogram codes will be marked with a special start character (different than the "security code") to authenticate this action.

When the flight computer receives a reprogram command, it will relay the data to the roll control module. The new instructions will be stored and utilized in the controller, but will not erase the default values.

Bonus Challenge B: Flight Data Broadcast

The second component of the bonus challenge is to broadcast flight data to the ground from the rocket during flight. While they are powered on, both the roll control module and flight computer will log sensor data to onboard SD cards. In addition, the roll control module will select and send a 256-byte message via wired UART to flight computer on a 50Hz frame (this frame will in fact be precisely 50Hz due to the deterministic timing of RTEMS). The flight computer will not have the same precise timing abilities, but will operate as close to 50Hz as possible. In each time frame, flight computer will record its own sensor data and compile packages for telemetry to ground. Running at 115200 baud, we expect to be able to send 256 bytes per 50Hz time frame, plus the default start/stop byte every ninth byte.

We plan to deliberately interpret the data as pure binary when we send it to the radio and receive it on the ground. While this reduces the immediate readability of the data, it allows very densely packed messages. This is desirable especially since we want to complete telemetry transactions as quickly as possible during flight. This task will run concurrently with the task for bonus challenge C, below.

Bonus Challenge C: Flight Math

The final component of the bonus challenge involves broadcasting a 5-character string to the rocket containing both digits and mathematical operators. The flight computer will perform the operations and send the result back to the ground. In order to maximize the frequency at which this and other data can be sent over radio, we plan to compress the uplink data before sending it to the rocket.

For each instruction of form

`‘a#b@c’`

where `a`, `b`, `c` are digits and `#`, `@` are both arbitrarily the `+`, `-` or `*` operators. Based on the competition rules the characters will initially be represented in ASCII, so the command is 5 bytes long. On the ground (a standard laptop computer), the digit characters will be converted to binary such that,

`‘a’->0bxxxx, ‘b’->0byyyy, ‘c’->0bzzzz`

Where `x`, `y`, and `z` are arbitrary binary digits (0 or 1). In binary, 4 bits are sufficient to represent up to 16 unique values. Since only 10 digit characters exist, this range is sufficient. The operators will be encoded into a different, custom 4-bit code as follows. We denote the bits of this encoding with the letter `w`.

Operators	4-bit encoding	Operators	4-bit encoding
'--'	0b0000	'++'	0b0100
'-+'	0b0001	'+*'	0b0101
'-*'	0b0010	'*-'	0b0110
'+-'	0b0011	etc.	

We compile all of the above into the following transmission packet:

`0bxxxxyyyy 0bzzzzwwww`

This 2-byte code contains the same information as the original, 5-byte character string. The flight computer will unpack this code, perform the desired operations, and produce an integer result. This result will usually only take a single byte to represent. However, in some cases where two multiplication operators are passed, the result may be as large as `9*9*9` or `0x2D9`. We plan to make the telemetry calls flexible so that the ground station knows to listen for 2 bytes when these cases occur, but default to a single byte under typical circumstances.