Preliminary Design Report

Kent State University High-Power Rocket Club

Kent State University

Team Mentor: Mr. Steve Eves

Team Faculty Advisor: David B. Stringer, Ph.D.

dstring1@kent.edu 330.672.3953

Team Program Manager: Nelson Figueroa, AET

nfiguer2@kent.edu 330.423.5885

Team Members:

Adam Fertig; Safety Officer; afertig1@kent.edu, AE (330) 842-3396

Adrian Rivera; Propulsions Team Lead; ariver17@kent.edu, AE (787) 397- 0204

Tyler Williams, AE

Kaushik Anantha; Data Acquisition Team Lead; kanantha@kent.edu, AE (410) 802-7020

Alex Gajowski, CS

Michael Garlak, CS

Timothy Park, CS

Nick Stahl; Structures Team Lead; nstahl2@kent.edu, AET (706) 249-6251

Maddie Wilson, MECH

Nina Patterson, PH

Sydney Bihn, AE

Zack Helfer, AE

MECH: Mechatronics Engineering Technology

AE: Aerospace Engineering

AET: Aeronautical Systems Engineering Technology

CS: Computer Sciences

PH: Public Health

**Table of Contents**

Executive Summary.........................................................................................................................6

Competition Rocket Description.....................................................................................................7

Design........................................................................................................................................7

Predicted Flight Performance..........................................................................................................8

Propulsion..................................................................................................................................9

Roll Control Systems.....................................................................................................................12

Main Control Board.......................................................................................................................14

Radio Communication System......................................................................................................15

Mobius ActionCam........................................................................................................................16

Software.........................................................................................................................................16

Safety.............................................................................................................................................20

Mentor Report Form......................................................................................................................21

**Table of Tables of Figures**

Table 1: Table of Terms...................................................................................................................4

Table 2: Table of Symbols...............................................................................................................5

Table 3: Material Weight to Strength Comparison..........................................................................8

Table 4: Center of Pressure Spreadsheet.......................................................................................11

Table 5: Estimated Budget.............................................................................................................19

Figure 1: Preliminary Rocket Dimensions.......................................................................................7

Figure 2: J800T-L Acceleration vs Time.........................................................................................9

Figure 3: J800T-L Altitude vs Time................................................................................................9

Figure 4: J800T-L Velocity vs Time.............................................................................................10

Figure 5: J415W-L Acceleration vs Time.....................................................................................10

Figure 6: J415W-L Altitude vs Time ............................................................................................10

Figure 7: J415W-L Velocity vs Time............................................................................................10

Figure 8: Roll System A; Servo Actuated Canard System............................................................19

Figure 9: Roll System B; Ram Air Roll Orientation System.........................................................19

|  |  |
| --- | --- |
| **GLOSSARY** | |
| **TERM** | **DEFINITION** |
| **Arduino Nano** | A small, lightweight, and programmable, microcontroller |
| **C++** | An object-oriented, compiled, and low-level programming language |
| **Communications Device**  *(Also, Logging Device)* | Arduino Nano which receives data from the Control device, logs the data on an SD card, and transmits the data through an XBee Pro to a ground station. |
| **Control Device** | Arduino Nano which interacts with sensors, moves control surfaces, and sends data to the Communications Device |
| **I2C** *(Also I2C)* | Inter-Integrated Circuit: 2 wire communication protocol |
| **I/O** | Input/Output |
| **LED** | Light Emitting Diode |
| **Quaternion** | A method of representing a rotation by describing a vector about which the rotation occurs and the angle of the rotation |
| **RAM** | Random Access Memory |
| **SPI** | Serial Peripheral Interface: 4 wire communication protocol |

*Table 1: Table of Terms*

*Table 2: Table of Symbols*

|  |  |
| --- | --- |
| Symbol | Meaning |
|  | The rockets current roll |
|  | The target roll |
|  | The rockets roll rate |
|  | The rockets roll acceleration |
|  | The physical output (e.g. fin angle) |
| *I* | The rockets moment of inertia about its roll axis |
| *k* | The “spring constant” |
| *c* | The “damping term” |
|  | The torque on the rocket about the roll axis |
|  | The torque on the rocket about the roll axis due to the rocket’s passive aerodynamics |
|  | The torque on the rocket about the roll axis that the system is trying to achieve |
| ∆ | The difference between the torque due to passive effects and the torque the system is trying to achieve |

**Executive Summary**

The Kent State University High-Power Rocket Club is an Integrated Product Team comprised of members with varying majors. Our purpose is to compete in the 2017-2018 Space Grant Midwest High-Power Rocket Competition in support of our organizational goals:

“To advance the arts, sciences, and technology of aeronautics and astronautics, to develop professionalism and leadership of those engaged in these pursuits, and to promote community outreach of the sciences.”

**Overview**

This preliminary report provides

1. Flight performance prediction methodology and models.
2. Design descriptions.
3. Engineering solutions to challenges presented by this competition.

**Objective**

To successfully meet and/or exceed, within the 2018 competition time constraints, the standards and objectives required by the competition.

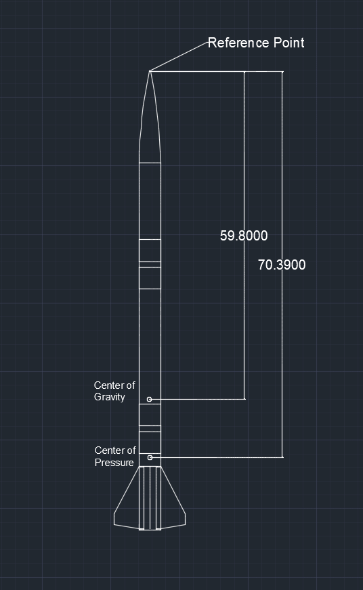
**Design Solutions**

The team’s design solutions are simple yet innovative and include

1. A non-commercial data collection package using components typically utilized in aeronautics.
2. Three proposed roll/orientation system designs.

**Competition Rocket Description**

Our rocket is a single body rocket with single-stage motors and a two-stage landing system. The airframe is a pre-glassed phenolic tubing with a phenolic motor mount, plastic nose cone, and two fiberglass couplers. The forward coupler houses our main non-commercial data package mounted on a 3D printed skid. The lower coupler houses our roll control system along with its operational computer mounted on a 3D printed skid. Each coupler utilizes fiberglass bulkhead plates and a pre-glassed phenolic switch band.



*Figure 1: Preliminary Rocket Dimensions*

**Design**

**Airframe**

The airframe design was chosen based on weight to strength ratio. The body is a 4-inch diameter pre-glassed phenolic tubing. A 3.9-inch inner diameter tubing was selected to maximize interior space for the roll control system. The pre-glassed phenolic tubing provides the high strength of a fiberglass body and the low weight of a phenolic tubing. Four fiberglass fins will add minimal weight while increasing stability and minimizing drag. The delta fins were chosen due to their proficiency during high-speed flight and ability to increase lift. Two nine-inch fiberglass couplers are housed inside of the rocket. The upper coupler with house the rockets non-commercial data package while the lower will house the roll control system and its computer.  The airframe of the rocket is made to be as aerodynamic as possible.

*Table 3: Material Weight to Strength Comparison*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Type | Product | I.D. (in) | O.D. (in) | Wall Thickness | Weight (oz.) | Weight per Length (oz./in) | Price per Length ($/in) | Strength |
| Fiberglass | 3" G12 Fiberglass Filament Wound Tube 48" Long | 3 | 3.14 | 0.07 | 34.075 | 0.71 | $1.90 | High |
| Blue Tube | 75mm Blue Tube | 3 | 3.1 | 0.05 | 20.106 | 0.42 | $0.62 | Med-High |
| Kraft Paper | 75mm LOC MMT | 3 | 3.16 | 0.08 | 13.827 | 0.41 | $0.44 | Low |
| Kraft Paper | 3.00in LOC Body Tube | 3 | 3.1 | 0.05 | 7.972 | 0.23 | $0.27 | Low |
| FGPT-3.0 | Pre-Glassed Phenolic Airframe Tubing | 3 | 3.19 | 0.1 | 23 | 0.64 | $2.64 | High |
| PT-3.0 | Phenolic Airframe Tubing | 3 | 3.13 | 0.06 | 12.1 | 0.34 | $0.53 | Med |
| QT-3.0 | Quantum Airframe Tubing | 3 | 3.15 | 0.07 | 15.2 | 0.42 | $0.61 | Med-High |
| Fiberglass | 4" G12 Fiberglass Filament Wound Tube 48" Long | 3.89 | 4.08 | 0.09 | 47.454 | 0.99 | $2.19 | High |
| Blue Tube | 98mm Blue Tube | 3.9 | 4.01 | 0.06 | 25.574 | 0.53 | $0.81 | Med-High |
| **FGPT-3.9** | **Pre-Glassed Phenolic Airframe Tubing** | **3.9** | **4.09** | **0.1** | **30** | **0.83** | **$2.92** | **High** |
| Kraft Paper | 3.9in (98mm) LOC Body Tube | 3.9 | 4 | 0.05 | 10.512 | 0.31 | $0.34 | Low |
| PT-3.9 | Phenolic Airframe Tubing | 3.9 | 4.02 | 0.06 | 15.3 | 0.43 | $0.58 | Med |
| PT-3.9-48 | Phenolic Airframe Tubing | 3.9 | 4.02 | 0.06 | 20.4 | 0.43 | $0.54 | Med |
| QT-3.9 | Quantum Airframe Tubing | 3.9 | 4.03 | 0.07 | 18 | 0.5 | $0.72 | Med-High |
| QT-3.9-48 | Quantum Airframe Tubing | 3.9 | 4.03 | 0.07 | 24.1 | 0.5 | $0.69 | Med-High |

**Camera System**

*Mobius Mini ActionCam* cameras are used for roll orientation verification, as required by the competition. Two downward facing cameras will be mounted on opposite sides of the rocket to keep drag even. Cameras will be mounted from the inside of the forward coupler.

**Predicted Flight Performance**

We calculated the rocket’s basic flight performance using accepted formulas based upon Newton’s second law. These calculations account for an overestimated drag estimation.

**[Eqn. 1]**

**[Eqn. 2]**

**[Eqn. 3]**

**[Eqn. 4]**

**[Eqn. 5]**

**[Eqn. 6]**

Using the values of:

d 0.1016 m

Newtons

0.8

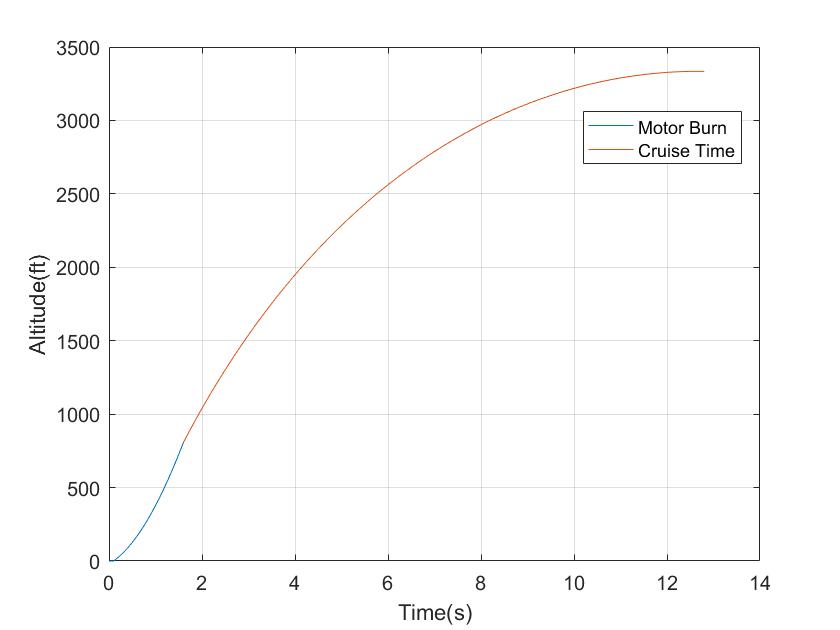
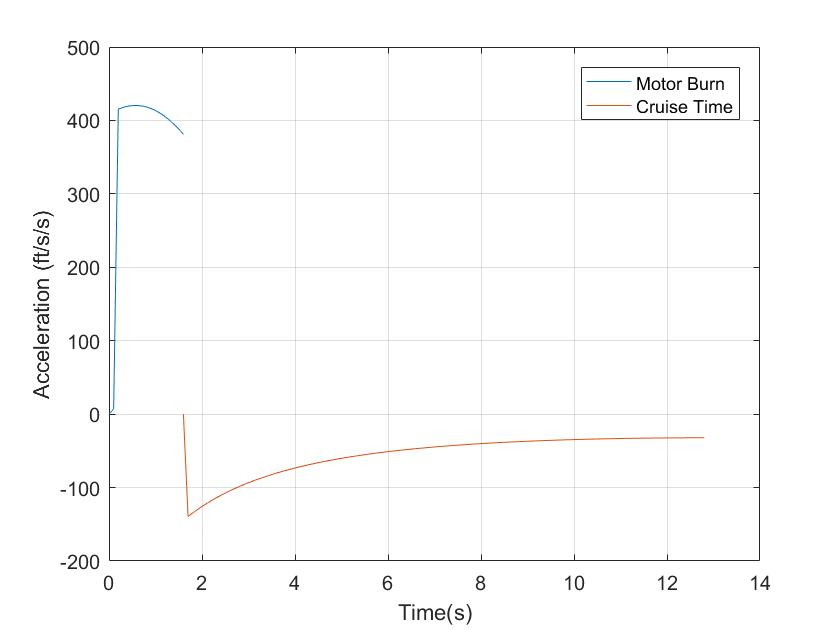
Initial Velocity

**Propulsion**

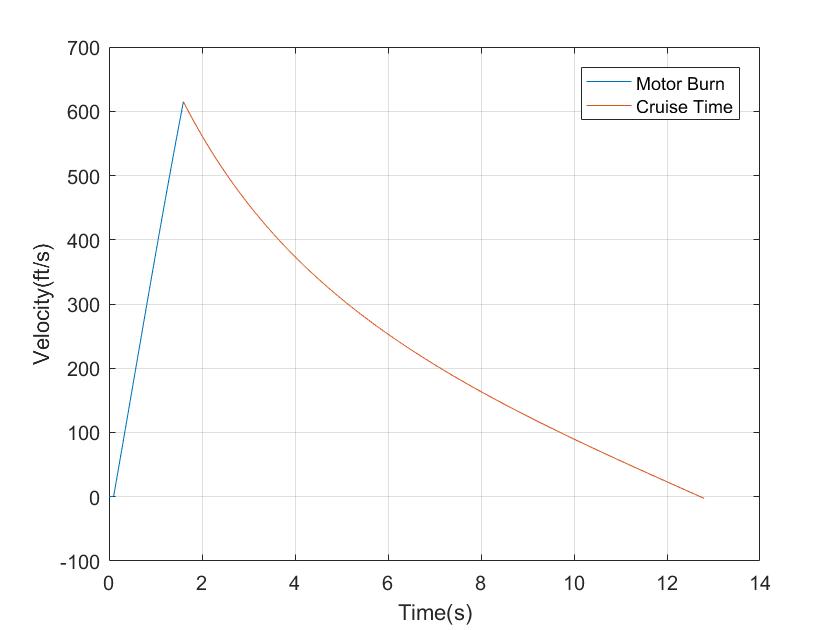
The J800T-L rocket motor was selected due to its higher average acceleration. The primary motor selection criteria was to maximize coast time while staying within altitude constraints. We found the J800T-L to be most effective in creating a longer coast time within our 13-pound weight goal. The J415W-L motor provides a close alternative to the J800T-L and can be used as a backup if the J800T-L provides a G-force that our electronics would not be able to handle, though we do not foresee this to be an issue.

***AeroTech* J800T-L**

Our apogee altitude for an AeroTech J800T-L motor corrected for drag using values from *Figure 2* is 3,334 ft., with a maximum velocity of 660.62 ft./s, and a maximum acceleration of 412.89 ft./.



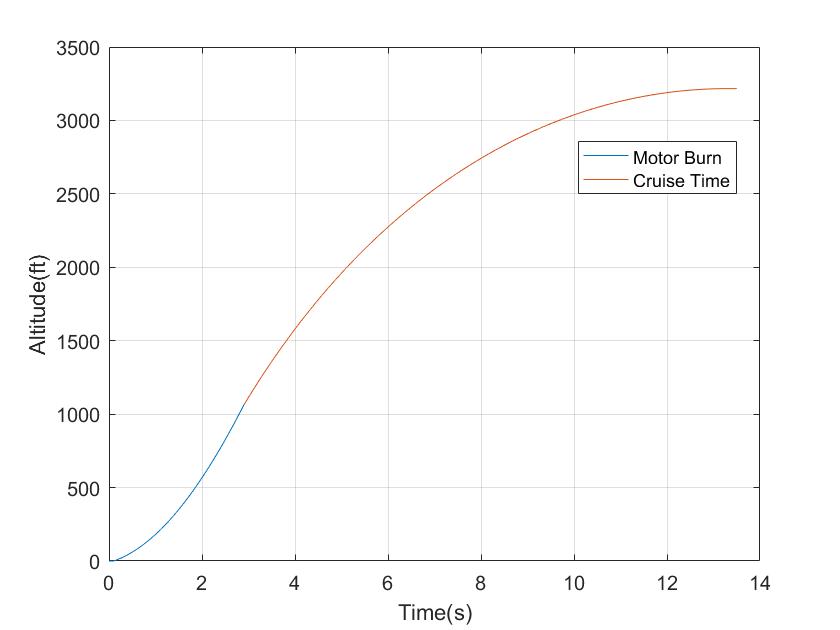
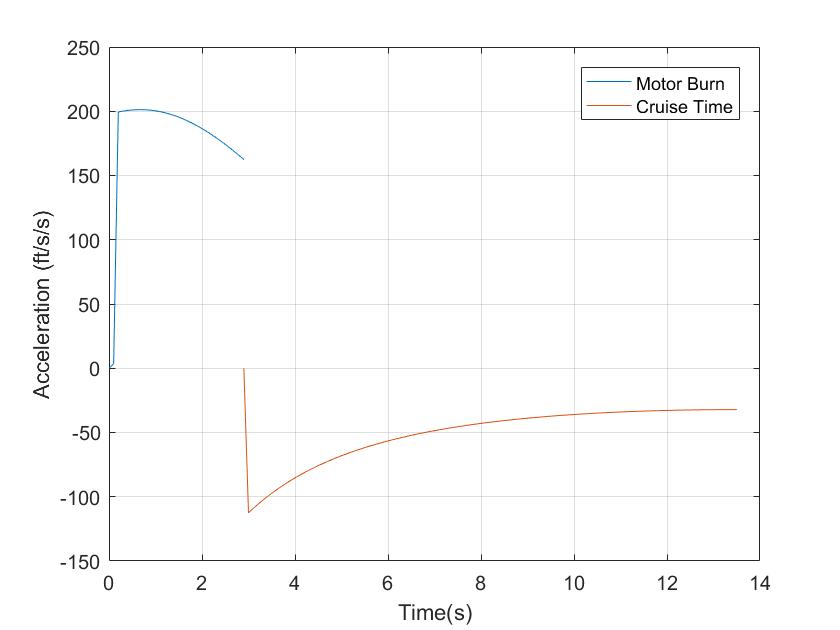
*Figure 2: J800T-L Acceleration v Time Figure 3: J800T-L Altitude v Time*



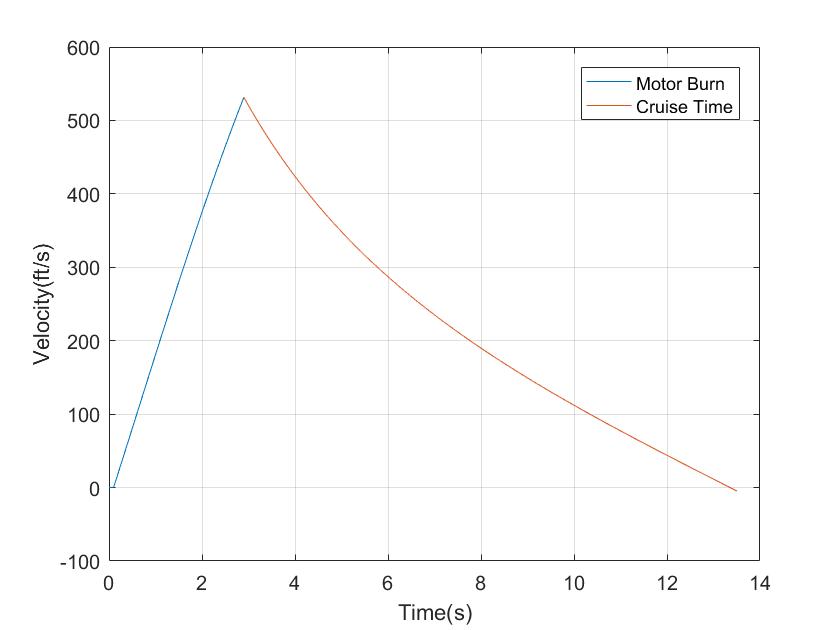
*Figure 4: J800T-L Velocity v Time*

***AeroTech* J415W-L**

Our apogee altitude for an AeroTech J415W-14A motor corrected for drag using values from *Figure 3* is 3,215.8 ft. with a maximum velocity of 531.43 ft./s and a maximum acceleration of 201.24 ft./.



*Figure 5: J415W-L Acceleration v Time Figure 6: J425W-L Altitude v Time*



*Figure7 : J415W-L Velocity v Time*

**Ejection Charge System**

The ejection charge system uses black powder for the charge. This was the initial choice due to reliability, low volumetric capacity, and effectiveness from a high total impulse. Its ability to ignite slightly quicker than other forms of ejection charges also played a major factor.

The black powder is encased in a small, cylindrical PVC cap, mounted on the top and bottom of the upper coupler. The nosecone and the lower part of the top coupler is held in place using #2 nylon sheer pins. The number of nylon pins used is determined by using Pascal’s Law, setting the pressure inside the parachute compartments equal to the quantity of force of the explosion over the area of said compartments. This calculation shows the relation of the sheer force of the pins to the affected area.

The amount of black powder used in the parachute compartments is determined by the Ideal Gas Law, using the molar mass of the black powder itself, the universal gas constant, and the additional parameters inside the rocket, and is set as a function of volume to the compartments themselves, which then provided the mass needed. We will also be ground testing the ejection charges as a measure of redundancy.

**Center of Pressure (CP) and Center of Gravity (CG)**

We calculated the center of pressure by finding each exterior component’s moment, and then multiplied each moment by the surface area. We then took the sum of the moments and divided the total moment by the total surface area to find the CP value. For ease of use, we used Excel to quickly find and compare CP values with different body tube sizes. All values were calculated using the tip of the nose cone as the reference point. Using an 18-inch upper body tube, we get a CP located 70.39 inches from the reference point. A *RockSim* model of the rocket was then used to validate our Excel sheet value. *RockSim* used the Barrowman Stability equation to get a CP value of 68.74 inches from the reference point. The center of gravity value of the rocket was calculated to be 59.75 inches and physically measured at 59.80 inches from the reference point. Our calculated static margin is measured to be a value of 2.66 while *RockSim* calculated our static margin at a value of 2.08. The difference in values between our calculations and *RockSim’s* calculations can be attributed to the different ways in which they were calculated.

**[Eqn. 7]**

Where *i =* Component = 1, 2, 3,...,*n*  **[Eqn. 8]**

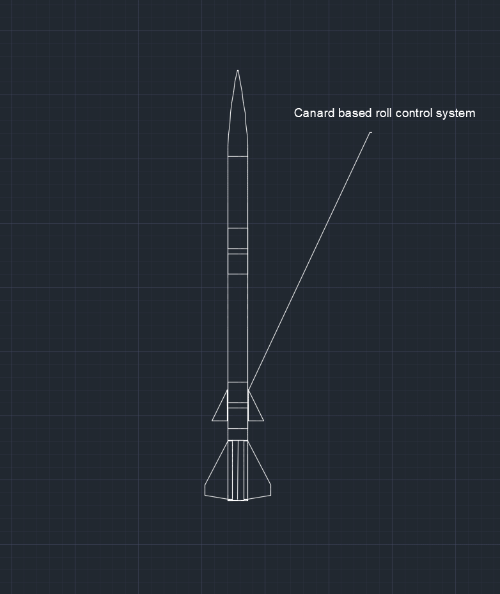
*Table 4: Center of Pressure Spreadsheet*

|  |  |  |  |
| --- | --- | --- | --- |
| Center of Pressure (Tip of the Nose is Reference Line) | | | |
| Component | Area | Center of Pressure (Individual) | Moment |
| Nose Cone | 33.5 | 11.16666667 | 374.0833333 |
| Upper Body | 72 | 25.75 | 1854 |
| Avionics Coupler | 4 | 35.25 | 141 |
| Mid Body | 116 | 50.25 | 5829 |
| Lower Coupler | 4 | 65.25 | 261 |
| Lower Body | 72 | 74.75 | 5382 |
| Fins | 29.7754375 | 79.58296156 | 9478.469992 |
|  | Area (total) | Center of Pressure | Moment (total) |
|  | 331.2754375 | 70.39324588 | 23319.55333 |

**Proposed Roll-Control Systems**

**Roll System A - Canard System**

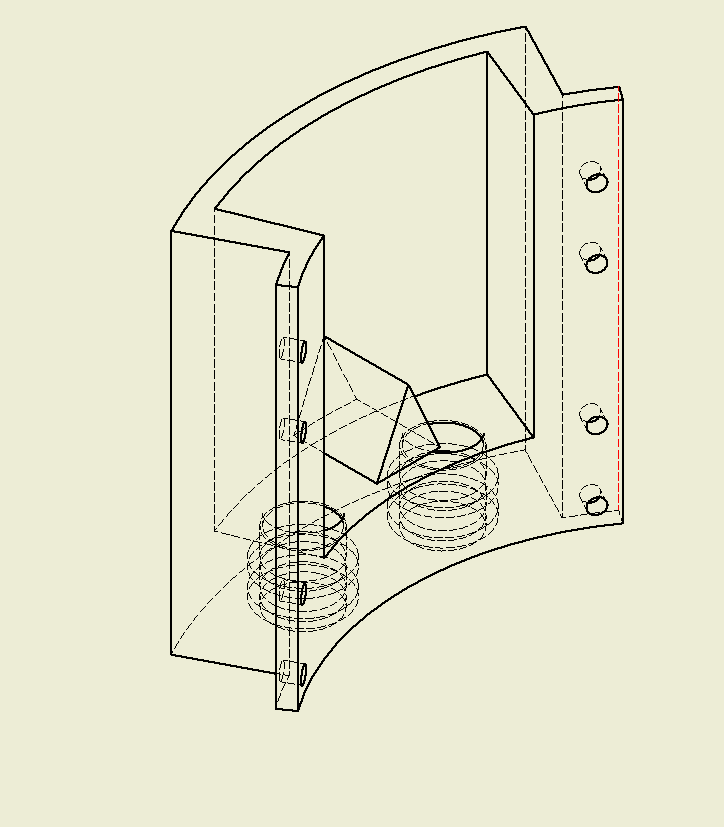
Two variable pitch canards posted below the center of gravity will rotate simultaneously to produce roll. The rotation will be actuated by an electronically controlled servo attached to a 3D printed system. The servo will rotate to a programed angle for a certain amount of time so that it may either stabilize or induce a specific roll-orientation on the rocket. Once a certain angle of orientation is achieved, the canards will quickly and shortly rotate in the opposite direction to act as a counter torque which will bring the roll to a stop.



*Figure 8: Servo Actuated Canard System*

**Roll System B - Ram Air System**

Two open air intakes will be positioned outside of the body of the rocket near the center of gravity parallel to the fins (on opposite sides of the rocket). Relative wind will be funneled into the inlets using the velocity of the rocket where the relative wind will be separated into two smaller pipe-like features. The relative wind will then be channeled down the airframe thorough tubes and then channeled perpendicularly to the fins. A servo will rotate a tongue depressor shaped object to cover one of the pipe-like features. This will cause the relative wind to be funneled down only one tube (per side of the rocket). This effectively creates angular momentum using a thrust vector perpendicular to the fins of the rocket using fluid dynamics.



*Figure 9: Ram Air Roll Orientation System*

**Roll System C - C System**

Four sixteen-gram C cartridges would be attached to individual C puncture pin fittings. Each puncture pin fitting would be connected to a short system of brass fittings to combine individual flows. The C gas output would then be controlled by a solenoid valve. Once released from the solenoid valve, the gas would be funneled down a system of tubes leading to opposing fins. Equal and opposite action from gas being shot out perpendicular to the fins would then induce a roll. This idea was not used due to excess weight from a solenoid that would be strong enough to control output pressure.

**Avionics Bays**

Our rocket houses two couplers, both of which having individual purposes. Each coupler is constructed from G10 fiberglass and will contain 3D printed sleds and bulk plates. 3D printed bulk plates increase ease of assembly when the couples are assembled and disassembled. Each sled has placeholders and standoffs imprinted for every electronics component to increase usability and ease of access.

**Recovery System**

To meet the design goal of a descent rate less than or equal to 24 ft./s, we selected a parachute size based on the weight to descent velocity ratio equation as follows:

**[Eqn. 9]**

Where;

Equation 1 then becomes: *(for a descent rate of only);*

, *Where;* **[Eqn. 10]**

Therefore, our selection consisted of parachutes having a minimum diameter of 1.82 meters or 72 inches.

Verifying with *RockSim*, we found our chute to be the correct diameter for a descent rate of less than 24m/s. Final verification will come from an Altimeter Two during pre-competition test launches.

**Main Control Board**

**Hardware**

**Main Board**

All devices in the main control assembly are mounted on a single circuit board in sockets.  This keeps cables (and therefore cable management) to a minimum while also allowing for easy replacement of damaged components

**Microcontrollers**

Programmable computation is provided by two Arduino Nanos whose purpose is to process sensor data, control the rocket’s roll, maintain contact with the ground, and log telemetry during flight.  The Arduino Nano was chosen due to its small size and widespread adoption among hobbyists and enthusiasts, meaning it is well supported and documented. In addition, it runs in real time (unlike more powerful, multitasking computers which rapidly switches which program is running at a given time, meaning that a given process [like our roll control system] may not actually be executing at a critical time).

Initially, a single Arduino Nano was planned to be used.  However, it became clear that the 2KB RAM provided would not be sufficient to accomplish all the requirements for the assembly, so the decision was made to use two Arduino Nanos.  The first - called the control device - is responsible for interfacing with sensors, physical output devices (e.g. servos, LEDs) and another Arduino Nano - called the communications device or logging device - to handle file I/O (for configuration loading, flight plan control, and data logging) and radio communications.  Data will be passed between the Arduinos during flight via I2C to ensure each has the information needed to accomplish its goals.

**Sensors**

**Barometric Pressure Sensor**

An Adafruit BMP280 (a Bosch BMP280 with a built-in PCB and 3.3V regulator) was chosen as an altimeter.  Adafruit’s tutorials and Arduino-compatible libraries justified the selection of this sensor over others.

**Absolute Orientation Sensor**

An Adafruit BNO055 (a Bosch BNO055 with a built-in PCB and 3.3V regulator) was chosen to provide information regarding the rocket’s acceleration and orientation.  Adafruit’s tutorials and Arduino-compatible libraries justified the selection of this sensor over others. This sensor includes accelerometers, magnetometers, and gyroscopes and performs sensor fusion and error filtering onboard.

**Radio Communication System**

**XBee XBP24-AWI-001**

The choice of radio transceiver was largely determined by the competition rules. XBees were chosen from Sparkfun electronics with quarter wave ground plane antenna preinstalled.  These works well for prototyping on the ground, however the antenna on the rocket will be replaced with a variant of a half wave dipole. The ground stations antenna may be outfitted with a directional antenna to ensure that the contact can be maintained with the rocket even at long range.

**Gravitech XBee add-on for Arduino Nano**

To connect the XBee to the Arduino, the Gravitech XBee add-on for Arduino Nano was selected because of its compactness and pre-assembled package which allows access to all the Arduino’s IO pins.

**SparkFun XBee Explorer Dongle**

To connect the ground station radio to a computer, a small USB dongle was chosen due to small size and ubiquitousness of USB ports.  If needed, the device can be separated from the computer and connected with a USB extension cord.

**Physical Output devices**

**Servos/actuators**

Some form of actuator will be required to drive the physical control system. However, it should be noted that the hardware and software as currently planned can be readily adapted to many different types of actuators.

**LEDs**

To provide a visual indication of the rockets control inputs, an RGB LED is positioned in view of one cameras. The LED can be set to any color, and will display a spectrum depending on the intended output torque (e.g. pure red for maximum clockwise torque, purple for maximum counter-clockwise torque). This will allow us to observe when the control system is providing input to direct the rockets rotation.

**Miscellaneous**

**Mobius Mini ActionCam**

According to competition guidelines, the launch, burn time, roll orientation pattern, and rocket separation must be recorded. The Mobius ActionCam was chosen given its video quality and battery life (with powerbank).

**Altus Metrum TeleMetrum GPS Payload**

The TeleMetrum Payload is a unique component and has capabilities far more than GPS tracking. Taking simplicity into consideration, the initial plan was to use a smartphone for tracking. Given the light weight, size, dual deploy capability, and other functions, the TeleMetrum was chosen over a smartphone. Apart from remote altitude measuring, the TeleMetrum has the following functions:

* Barometric Pressure Sensor
* Single axis accelerometer to verify rocket speed
* Dual-deploy capability; first at apogee and second charge ejection at 820 ft. (250 meters) (with ability to change altitudes of ejection)
* Ability to retrieve Real-time flight data

The TeleMetrum will function as the primary component for the ejection charges.

**Software**

**Technologies used**

**Programing language:**

All flight software is written in C++.  The C++ language was selected not only because it is the primary language supported by the Arduino, but also because it is a compiled language that supports low level functionality, works well in embedded applications, and supports object oriented programing.

**Arduino Libraries**

The Arduino compiler provides several functions not present in regular C++ for directly controlling the hardware.  The SoftwareSerial library is responsible for communicating with the XBee. The Wire and SPI libraries handle interactions between the control device, communications device, and other devices.  Furthermore, other standard Arduino libraries are used in the program.

**Sensor Libraries**

Both the BMP280 and BNO55 have manufacturer provided libraries which allow easy access to the data provided by the sensors.  They are used because they are written by the manufacturer and minimize the amount of work needed to get usable information from the sensors.

**Matrix math library:**

Since calculations based on rotation matrices must be performed, and these calculations are not directly supported in C++, a 3rd party library optimized for the Arduino is used.

**Software Architecture**

**Configuration file**

Some properties vary from flight to flight and from rocket to rocket, and since editing source code for each flight is not optimal these properties are stored as text in a configuration file, separated by a newline character, on an SD card, allowing for easier manipulation of data.  The logging device reads in properties and passes them to the control device one at a time until a specified number of properties is reached.

**Flight Plan**

Flight plan parsing will be performed on the communications device. The flight plan for the rocket will be provided as a string and represents the rotations and timing that the rocket must complete while ascending. For example, the string “#3;+0901000;-0001000;~1802000;” might be provided. The first part of the flight plan is the number of commands to execute, represented by the sequence “#<number\_of\_commands>;” which, in this case is 3. After that, the list of rotation angles and timings, separated by semicolons, follows. Each element in the list has three parts, the direction of rotation specified by a symbol, the angle, specified by the first three digits, and the time the rocket should take to make the turn.

The direction symbol could be one of three symbols, ‘+’, ‘-’, or ‘~’. ‘+’  represents clockwise rotation and ‘-’, counterclockwise rotation. ‘~’ tells the software to take the shortest possible rotation to get to the desired angle. The following three numbers represent the angle of the rocket in terms of degrees. The final four digits specify the roll time in milliseconds, which are the most accurate timing data available to our sensors. After completing the flight plan, the rocket holds the final orientation until it reaches the apogee.

Given the previous example, the rocket rolls clockwise to 90° in one second. It then turns counterclockwise in another second. Finally, it turns 180° in two seconds, most likely going counterclockwise to use its momentum. Also, using this format, one could specify that the rocket hold a position by using two commands, “+1801000;~1803500;”. This would tell the rocket to roll to 180° in 1 second and then to roll to its current position for another 3.5 seconds, essentially keeping it in the same place.

Providing a flight plan to the rocket happens one of two ways. By default, a flight plan will be pre-programmed and placed into the configuration file for the rocket. The plan can be verified manually before being loaded and verified by the rocket before flight. The second method sends a flight plan to the rocket remotely. To do this, a flight plan is generated and verified on the control computer, made possible by the parser being written in pure C++ to enable it to run across platforms. Then, the plan is transmitted over the radio to the rocket which will verify it. Should the flight plan fail parsing on the rocket, the default plan will be used.

**Communication**

The rocket and the control computer uses XBee radios to pass information back and forth. The rocket’s XBee is connected to an Arduino microcontroller and the Arduino SoftwareSerial library, which allows reading and writing bytes of data at a time. The control computer connects to the XBee through a USB adapter and is configured using the Digi XCTU software. Every message sent between the rocket and ground is prefixed with a single character to identify the source. Messages without Kent State’s identifier are ignored. The communication with the rocket is split into three sections: pre-launch, ascent, and descent. Each stage uses the radio connection differently, and the rocket controls the current stage.

During the pre-launch phase, after a valid flight plan that verifies on the control computer, the file is sent to the rocket. The communications device parses the plan and reports back to control device with a success or failure message. The launch can be delayed until the rocket confirms a correct flight plan. Otherwise, the rocket uses a preset flight plan.

While the rocket is ascending, sensors attached to the Arduino report at ~100Hz. When there is a sensor update, that data is passed to the ground station. To reduce the overhead of parsing messages from the ground, communication during ascent is one way.

After the rocket reaches apogee, two-way communication resumes with the control station sending problems for the rocket to solve. The rocket sends responses once the result has been computed. A simple interface is provided on the communications device to pass the raw message to the routine that solves the problem. This allows the implementation of problem solver to be done at a later time.

**Flight modes**

The behavior of the software varies based on the phase of flight (pre-flight or post-flight) the rocket is in at the time.  To accomplish this, the phase of flight is recorded as an integer (the flight mode) starting from 0 for the preflight phase, then 1 for powered flight, then 2 for the coast until near apogee (the part of the flight where the roll is actively controlled) and so on.  A switch statement is used to control the actual branching, ensuring that only the code intended to run during the rockets current phase of flight actually executes. A check is performed within the switch statement to determine if the next phase of flight has started (e.g. the rocket has a strong vertical acceleration and the launch rail is no longer present indicates that the boost phase has started). If it has, the flight mode is incremented.

**Roll Control**

The actual roll control algorithm runs on the control device.  The design philosophy of the roll control system itself is to make the rocket behave as though it were a critically damped harmonic oscillator along its roll axis (i.e., obeys the equations *I* + c + k( ) = 0 and 4*I*k = 0 with being the target roll (as provided by the flight plan)).  In this way, the rocket quickly and smoothly returns to the target roll after being disturbed from it, without overshooting.

In order to accomplish this, a is chosen on the ground to achieve a quick return to the target roll without exceeding the capability of the software to update or the roll control system to provide adequate torque.  From this (and the rockets roll moment of inertia (*I*)) values for *c* and *k* will be calculated.  In flight, the software uses the target roll provided by the flight plan, as well as the current roll and roll rate provided by the sensors, to calculate the torque  necessary to make the rocket obey the aforementioned equation of motion (in other words, = *c* k()).

The rocket is expected to be under some torque due to its passive aerodynamics, which is expected to be a function of its roll rate and speed relative to the air (denoted as (,*v*)).  Subtracting that torque from the target torque gives us the torque that must be provided by the control system (, ) = ).

Finally, the torque provided by the control system itself is expected to be a function of the airspeed of the rocket and the physical inputs to the system (e.g. the angle of the control surfaces), inverting this function with respect to the physical input allows us to calculate what that input should be in order to achieve the desired torque (e.g. (, v) = (, v)).  This value is used to control the servos or similar devices, ultimately controlling the physical roll control system.

**Data Logging**

The rocket logs altitude, airspeed (during ascent), and rotation information throughout its flight.  To accomplish this with minimum overhead, the control device transmits the quaternion provided by the BNO55, as well as the altitude provided by the BMP280, to the communications device, which logs this data on a micro-SD card. The communications device then transmits the data to a ground station via XBee.  Other information (roll, pitch, yaw, and airspeed) is calculated from this data on the control device during flight. To save bandwidth it is not directly transmitted to the communications device, stored, or transmitted to the ground station.

Additionally, independent data logging is done by an attached TeleMetrum.

**Auxiliary Features**

In addition to the primary mission of stabilizing and controlling the rockets roll during the ascent coast, the competition includes a bonus challenge of performing some simple tasks remotely on the rocket during decent (e.g. calculating arithmetic expressions, echoing text sent to the rocket but in reverse order, etc).  While any single type of these problems would be rather simple to solve, since the set of problems that could be included at similar complexity to the examples given is extremely large, it is not possible to write software to handle all of them ahead of time. For now, an interface is implemented which allows a raw message to be passed to a handler. The result is returned in a format which can be sent to the ground. Using this interface, a four-function calculator is implemented as a proof of concept. The expectation is that the handler changes after more details are given.

**Safety**

**Materials and Handling**

All hazardous materials, including but not limited to rocket motor reload kits, will be stored off campus with our mentor, Steve Eves. Mr. Eves is Level 3 Tripoli certified, and he will handle all storage and transportation of materials to ensure safety amongst our members. Mr. Eves will also be in charge of instructions at all of our test and competition launches, and he will have the final say when it comes to safety. He will be in charge of protocol if a motor is armed and does not ignite.

**Launch Procedures**

To ensure safety at launches, many procedures have been put in place to mitigate risk. Most importantly we will use the buddy system at all times, prelaunch, on the launch pad, and rocket recovery. The buddy system provides a second person in case of injury, and provides a second opinion. The group will be broken down into teams of rocket prep, launch pad set up, and recovery. Members will know their induvial and team jobs for every launch. A safety checklist will be prepared and filled out for every launch by the safety officer and will oversee all three teams along with Mr. Eves to double check every step before and after launch to guarantee safety along every step.

As per National Fire Protection Association (NFPA) code 1127, the launch team will consist of no more than five noncertified Tripoli or NAR members not including all certified members.

**Estimated Budget**

*Table 5: Estimated Budget*

|  |  |  |
| --- | --- | --- |
| ***Estimated Budget Delegation ($10,000)*** | | |
| **Delegated Component** | **Estimated Budget [$]** | **Budget Percent [%]** |
|
| Travel/Lodging | 3,800.00 | 38.00 |
| Lab Space/Overhead | 0.00 | 0.00 |
| Prototyping | 100.00 | 1.00 |
| Instructor Mentorship | 1,200.00 | 12.00 |
| Reports/Documentation Supplies | 100.00 | 1.00 |
| Propulsion | 1,200.00 | 12.00 |
| Structures | 1,100.00 | 11.00 |
| Data Acquisition Components | 1,600.00 | 16.00 |
| Registration Fee | 400.00 | 4.00 |
| Hardware/Software | 400.00 | 4.00 |
| Misc. | 100.00 | 1.00 |
| TOTAL | 10,000.00 | 100.00 |

