SPACE
Space and Celestial Explorations, Inc.
2778 Hayward Ave
Ann Arbor, MI 48109-2140

April 18, 2017

Professor Peter D. Washabaugh, NASA Representative
3028 Francois-Xavier Bagnoud Aerospace Building
1320 Beal Avenue
University of Michigan
Ann Arbor, MI 48109-2102

Dear Professor Washabaugh,

We are writing to enclose our report on the sea-level prototype for NASA's high-Earth blimp. NASA has requested design proposals for a blimp that conducts surveillance on Earth at an altitude of 10 km. They have specified that this blimp must carry a 100-g camera as its payload, have a system mass of less than 1000 g, store into a 23 cm x 56 cm x 91 cm box when deflated, and be constructed of only materials supplied by NASA. As a result, we were asked to design, build, and test a sea-level prototype of such a blimp; research Earth's environment at 10 km altitude and its implications on our design; and propose adaptations for it to fly at that altitude.

Over the past five weeks, we have designed and constructed a sea-level prototype blimp that meets all of NASA's stated requirements. Our sea-level prototype is a remotely controlled blimp with a C-class shaped envelope and a gondola made of bass and balsa wood. It also includes a 10 V power system and two motors with seven inch propellers. After testing our sea-level prototype, we found that it has a maximum speed of 1.31 m/s. In addition, it can complete NASA's speed test in 35.91 seconds, NASA's reconnaissance test in 46.82 seconds, and stay aloft for 8 hours and 10 minutes. However, we realized that a blimp is not a practical vehicle to conduct surveillance at 10 km altitude due to jet streams. Thus, we have instead proposed an uncontrolled balloon in place of a blimp at 10 km altitude. This balloon would have no propulsion system and thus float freely in the wind. The envelope would be made of Polyester Mylar Laminate and the electronic system would be powered by solar panels. All information is detailed in the enclosed report.

Thank you for your support of our project. The resources that you have provided proved to be vital to the construction of our blimp. Without your support, we would never have been able to produce such results. In addition, it was a great honor to work for you and the rest of NASA. This was a once-in-a-lifetime opportunity that would have been impossible without you.

If you have any questions, please contact us at (239) 248-0550 or fostbenj@umich.edu.

Sincerely yours,

Seamus Callaghan   Alex Chen   Benjamin Foster   Ryan Johnston   David Koelzer

Sea-Level Prototype of NASA's High-Earth Airship: Design, Performance, Adaptations for 10-km Above Earth

by
Seamus Callaghan
Alex Chen
Benjamin Foster
Ryan Johnston
David Koelzer
Engineering Interns

SPACE
Space and Celestial Explorations, Inc.
2778 Hayward Ave
Ann Arbor, MI 48109-2140

Report Submitted to
Professor Peter D. Washabaugh,
Representative, NASA
3028 Francois-Xavier Bagnoud Aerospace Building
1320 Beal Avenue
University of Michigan
Ann Arbor, MI 48109-2102

April 17, 2017

Table of Contents

Table of Contents

List of Figures

List of Tables

FOREWORD

The National Aeronautics and Space Administration (NASA) has requested proposals for an Earth-surveillance blimp to fly at an altitude of 10 km. The blimp must:

- Carry a 100-g camera as its payload
- Stay aloft for at least one year
- Have a mass of less than 1000 g
- Store into a 23 cm x 56 cm x 91 cm box when deflated
- Be constructed of only materials supplied by NASA
- Complete NASA-specified speed and maneuverability test courses in qualifying times

Task

Space and Celestial Explorations (SPACE) has asked our team of engineering interns to complete the following tasks for NASA's proposal:

- Design and build a sea-level prototype for their Earth-surveillance blimp
- Test the sea-level prototype's speed and reconnaissance abilities
- Propose ways to adapt it for Earth at 10-km altitude

We have completed these tasks and presented our findings and conclusions to a NASA representative.

Purpose

The purpose of this report is to provide details and supporting documentation for:

- The design of our sea-level prototype
- The performance of our sea-level prototype
- The new material, size, and mass for our proposed redesign of the sea-level prototype for 10-km altitude above Earth

SUMMARY

We have successfully designed and built a sea-level prototype of NASA's high-Earth blimp. In addition, we have also conducted speed, reconnaissance, and time aloft tests on the sea-level prototype. Afterwards, we researched Earth's environment at 10-km altitude and developed a proposal for a high-Earth version of our blimp.

Design of Sea-Level Prototype

Our sea-level prototype is a remotely controlled blimp with a C-class shaped envelope. Its gondola is made of a combination of bass wood and balsa wood and houses all the blimp's electrical components. The gondola also has a fin made of polymeric membrane material with a wooden support structure to stabilize the blimp during flight. Furthermore, the blimp is powered by two battery packs producing a total voltage of 10 V. The batteries are connected to a microcontroller that provides differential thrust to allow the blimp to turn and a MOSFET board that distributes power to two motors. These motors are mounted 30 inches apart along the axis of the blimp's center of mass and produce thrust to propel the blimp forward. They can also be deflected 180 degrees by a mechanism consisting of a servo and two gears with a 2:1 ratio, allowing the blimp to change its altitude. Lastly, the blimp can receive radio signals through its receiver and can be controlled via a transmitter.

Performance of Sea-Level Prototype

We conducted three tests using our sea level prototype. The first was a speed test in which we assessed the time it took for our blimp to fly to and back from a point about 20 m away. Our sea-level prototype blimp was able to complete the speed test in an average of 34 seconds. During this test, we also determined that our blimp could reach a maximum speed of 1.31 s. The second test was a reconnaissance test in which we assessed the time it took for our blimp to fly under a laser and up to a target, take a picture of the target, and return to its starting location. Our blimp was able to complete the reconnaissance test in an average of 59 seconds. The last test was to determine how long our blimp could stay aloft without refilling the envelope with helium. We found that our blimp could do this for 8 hours and 10 minutes.

Adaptations for 10-km Above Earth

Due to high wind speeds of up to 123 m/s (275 mph), no means of propulsion would be powerful enough to control a blimp at 10-km altitude on Earth. Therefore, a blimp is not a practical vehicle to conduct surveillance in this environment. As a result, we propose changing the design to an uncontrolled spherical balloon that would be freely propelled by winds. It would have an insulating foam gondola covered by solar panels, an electric heater, and an infrared camera. We expect it to have a mass of 3200 g, a volume of 8.78 m^3, and a radius of 1.28 m. Since the mass exceeds the 1000-g limit, we would need to request a higher mass limit from NASA.

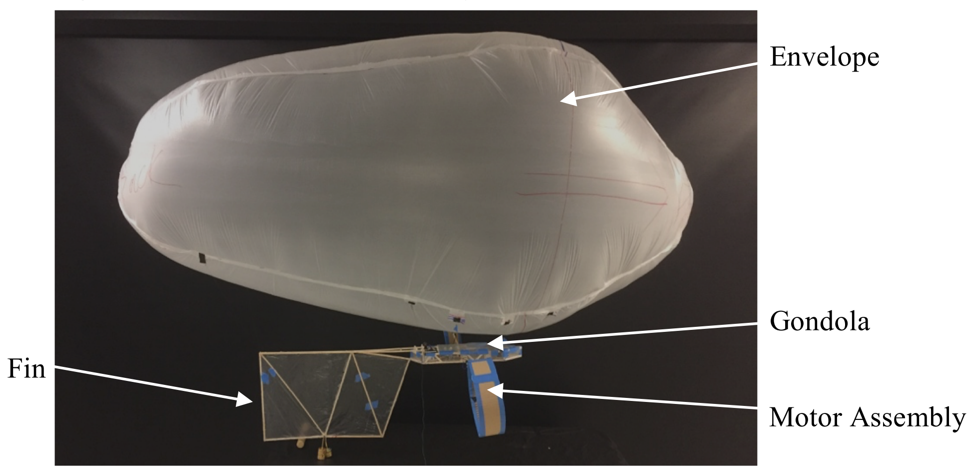
DETAILS
In this section, we will begin by describing the mass budget of our sea-level prototype and the design and rationale for each of its subsystems. We will also discuss the performance of our sea-level prototype and conclude by outlining our proposed balloon for Earth at 10-km altitude.

Design of Sea-Level Prototype
We designed our sea-level prototype as a remotely controlled blimp. It meets all of NASA's requirements, weighing less than 1000 g and storing into a 23 cm x 56 cm x 91 cm box when deflated. It was also constructed using only materials provided to us by NASA. These materials include:

- Polymeric membrane material
- One 4-channel HiTech "Feather Lite" 72 MHz FM receiver
- One Gravitec-Arduino Nano microcontroller with power distribution board
- One dual channel MOSFET motor control board
- One HiTech HS-55 servo
- Two Astroflight "Firefly" motors and propellers
- Eight NiMH size-AAA batteries in protective enclosures
- Bass and Balsa wood
- Electrical wiring

The sea-level prototype consists of several subsystems integrated together. These subsystems include the envelope, gondola and fin, servo mechanism, propulsion system, electrical system, transmitter controls, and microcontroller program. The masses of each component of our blimp is summarized in a mass budget. Each subsystem was designed to maximize the speed and maneuverability of the blimp while keeping construction reasonably simple. Figure 1 gives an overview of our design.

Figure 1: Overview of Earth Prototype



Envelope Designed to Reduce Drag: The envelope is the part of the blimp that is inflated with helium to provide the blimp with the buoyancy it needs to stay aloft. The rest of the blimp is suspended beneath it by shroud lines.

The envelope has a C-class shape (Figures 2 and 3) with a fineness ratio (the ratio of its length to its maximum diameter) of 2.0. This shape was chosen because it has a low drag coefficient of approximately 0.04 [1], which means the force of drag on the envelope is relatively low, so the blimp can reach a higher maximum speed.

Figure 2: Preliminary Envelope Design

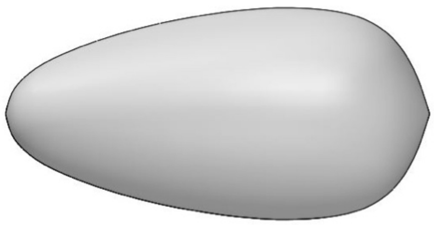


Figure 3: Fully Constructed Envelope.

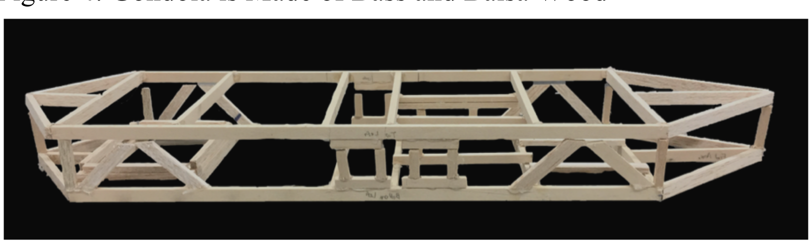
The envelope was constructed from four panels of polymeric membrane material sealed using a hot-jaw sealer. The shape of the panels was determined using a CAD program. The upper two seams were folded inside the envelope to reduce drag and helium leakage, while the lower two seams were left on the outside of the balloon for ease of construction and for attaching the shroud lines. One of the lower seams has a nozzle attached to it for inflation and deflation, which is clipped shut with a paperclip while in use. Four shroud lines on each side connect the lower seams of the envelope to the corners of the gondola, four more connect to the motor arms, and two more connect to the fin. This keeps the envelope firmly connected to the rest of the blimp, and reduces instability.

The envelope has a mass of 96.62 g, a length of approximately 1.8 m, and a volume of approximately 0.713 m^3. When inflated with helium, this provides a gross lift of 7.55 N, which is more than enough to lift the blimp.

Gondola Houses Electronics and Fin Provides Stability: The gondola (Figure 4) is the structure that is suspended below the blimp envelope that supports the propulsion, electrical, and control systems. It was constructed from a mixture of balsa wood, which has a low density, and bass wood, which has higher strength, held together with hot glue.

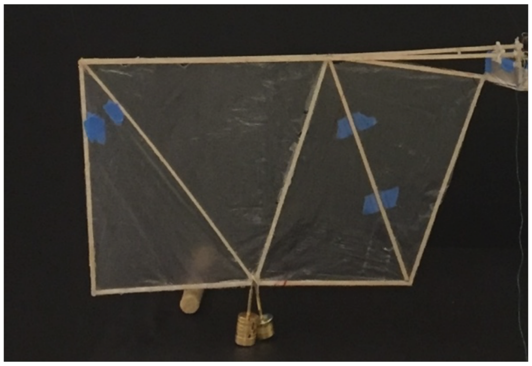
The gondola has an aerodynamic design, and the bottom and sides have a skin made from the same material as the envelope. This shape and skin reduce drag, which helps the blimp reach a higher maximum speed.

Figure 4: Gondola is Made of Bass and Balsa Wood



The fin (Figure 5) is a structure protruding off the back of the gondola that stabilizes the blimp during flight. It consists the same material as the envelope supported by a frame of basswood and balsawood. The fin also supports the ballast, as that is the ideal position to locate the ballast mass in order for the blimp to remain level with the ground. The total mass of the gondola and the fin (not including the ballast) is 77.55 g.

Figure 5: Fin Provides Stability (Ballast also pictured)



Servo Mechanism Allows for Pitch Adjustment: The servo mechanism (Figure 7) uses a small, precision servo motor to position of the motor ducts to control vertical thrust. The mechanism uses two pulleys with a 2:1 gear ratio connected by one timing belt. This allows the 90º servo to provide 180º of rotation to the motor ducts.

Figure 6: Overview of Gondola with Location of Servo Mechanism Highlighted

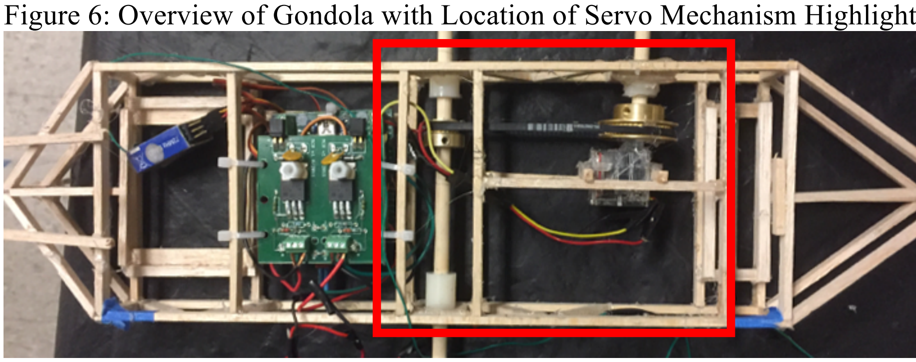
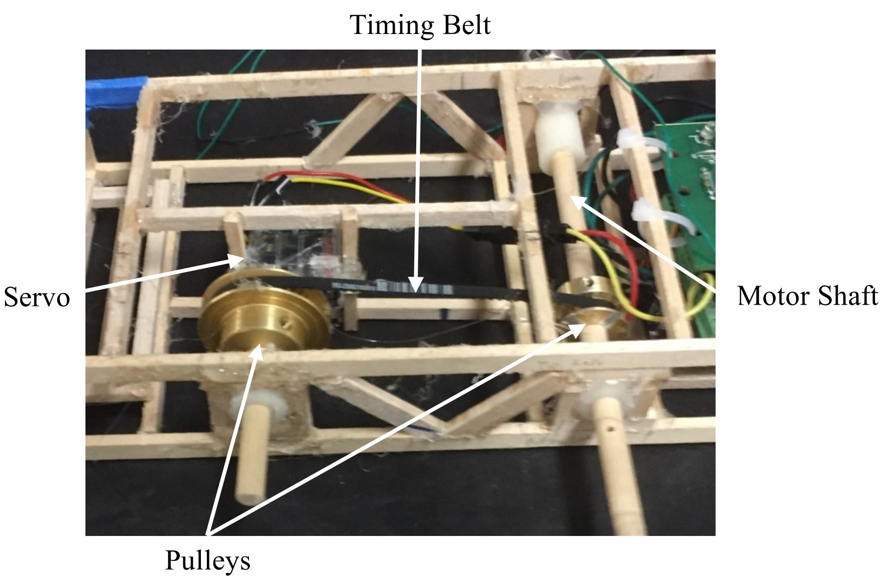


Figure 7: Close-up of Servo Mechanism



Propulsion System Provides Thrust: The propulsion system (Figure 8) provides thrust to propel and steer the blimp. It consists of two motorized, 7-inch propellers inside of ducts mounted on shaft arms that extend them 15 inches off the sides of the gondola.

The ducts are used to protect the propellers from collisions with objects, as well as to increase the efficiency of the propellers by directing air in a specific direction. The edges of the ducts are taped over to reduce drag caused by rough edges. Extra ballast was added to one of the motor arms to keep the propulsion system balanced, as one of the motors had a heat sink attached to it, giving it extra mass.

The arms are connected to the shaft of the servo mechanism by collars, which can be detached to allow the removal of the arms so that the blimp meets NASA's compact storage requirements.

Figure 8: Motor Ducts Spaced Far Apart to Increase Torque When Turning

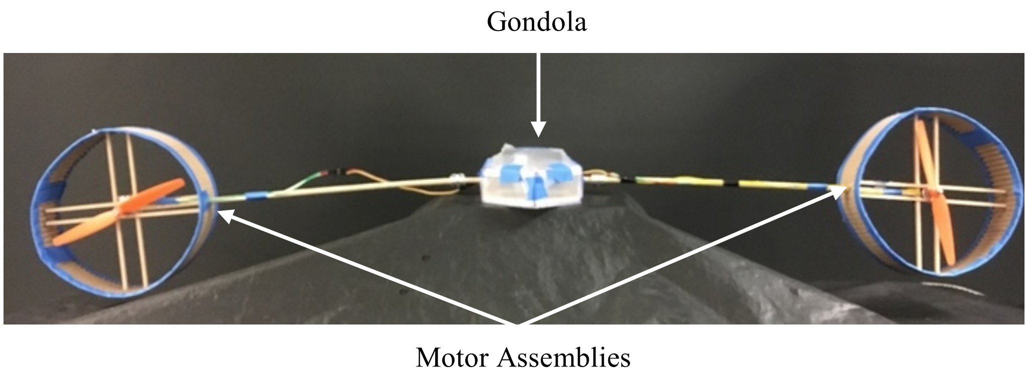
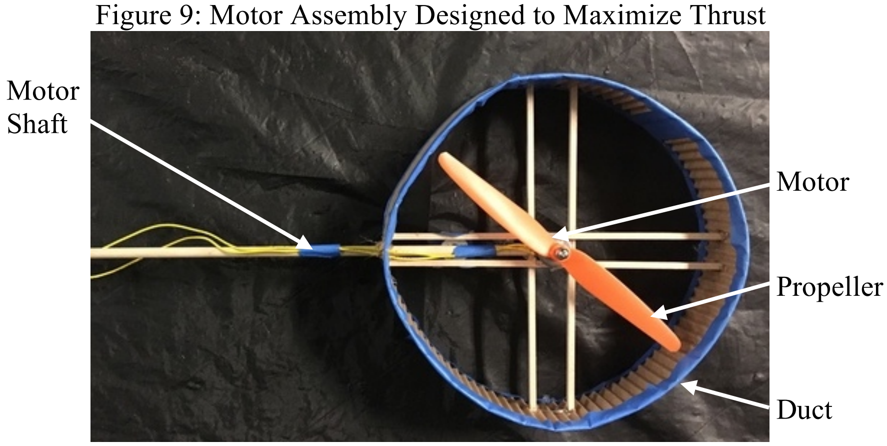


Figure 9: Motor Assembly Designed to Maximize Thrust



Electrical System Provides Power and Controls Blimp: The electrical system was 10-volts and its main components were a microcontroller, a MOSFET motor controller board, and a radio receiver. The microcontroller was used to control the thrust of the motors as well as the deflection of the servo while the MOSFET board was used to distribute the correct amount of power to the motors. Figures 10, 11, and 12 show an overview of the electrical system.

Figure 10: Block-Diagram Shows How Electrical System Processes Signals From Transmitter (Wiring is Simplified)

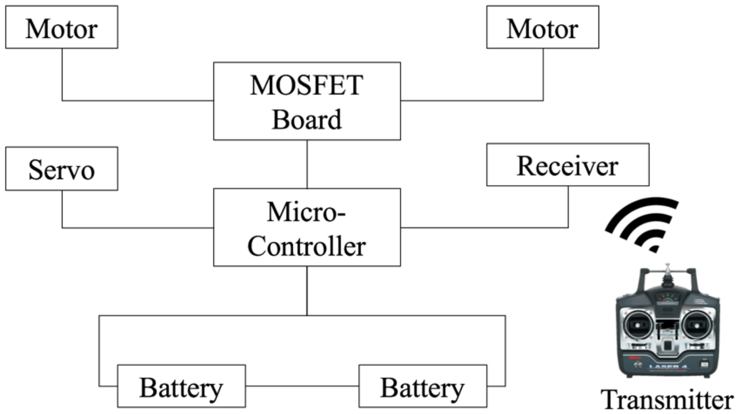


Figure 11: Electrical Components Spaced to Distribute Weight Throughout Gondola

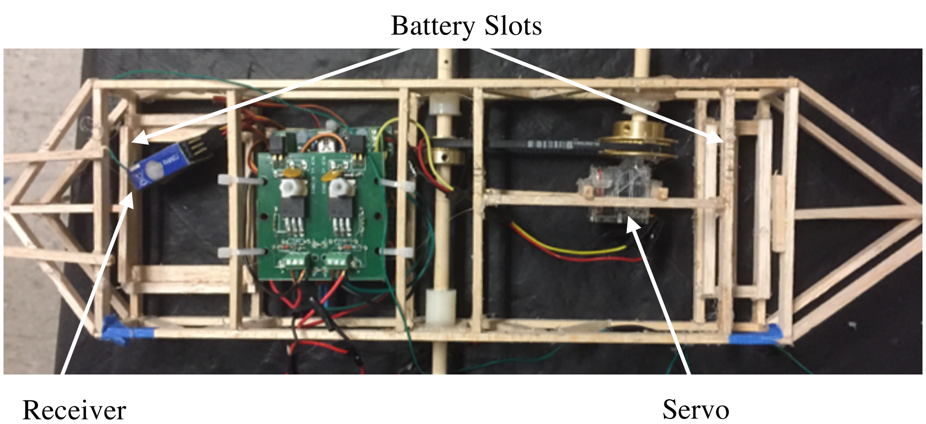
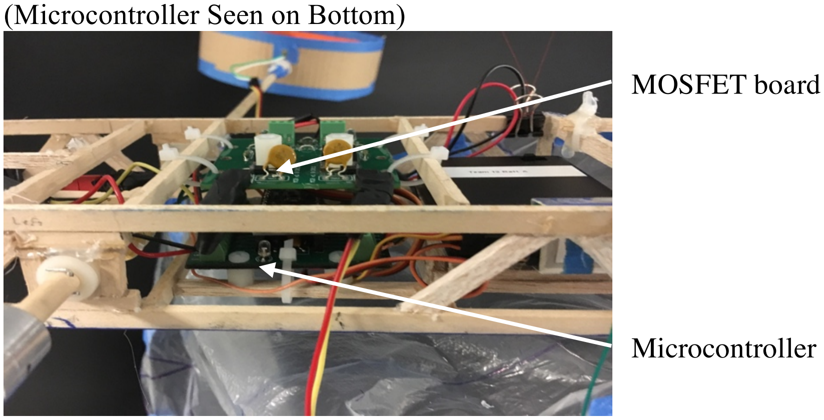


Figure 12: Microcontroller and MOSFET Board Compacted to Conserve Space (Microcontroller Seen on Bottom)



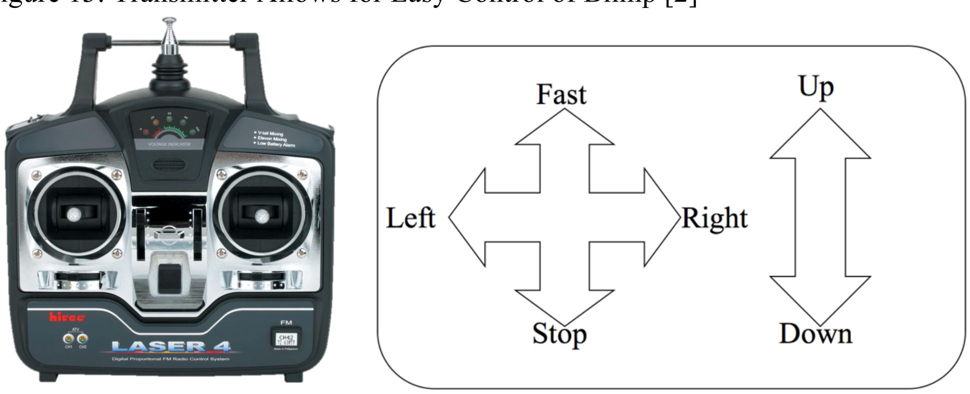
The microcontroller acted as the main input for all the wiring, every component interfaced with the microcontroller in some way. Our microcontroller included a terminal adapter that added pins that allowed for voltage pass-through from the batteries. This allowed us to connect the MOSFET board's power and ground pins to the terminal adapter and its signal pin to the microcontroller. Both the servo and the receiver were powered of the two 5-volt pins on the microcontroller and had their signals routed into the digital pins on the microcontroller. We were sure to put the servo signal into a pin that supported the servo library we used (D11). At first, we attempted to use a servo that was modified for 360-degree rotation but it wasn't entirely working and was to unreliable to be used in our blimp. We switched to a servo of the same design but without the modifications and relied on a mechanical system for increasing its range, as described above in the servo mechanism section.

The MOSFET board acted as a power distribution system for the motors. Our original design did not include a MOSFET board to minimize the mass of our electrical system, but we later realized we would need the board to use our 10-volt electrical system to its full potential. Thus, the MOSFET board handles signal from the microcontroller and is passed the full 10-volts of power through a pin on our microcontroller's terminal adapter. The purpose of the MOSFET board can be summarized as combing both the power and signal for the motors in a way that the motors can understand.

Our electrical system's hub is the microcontroller which touches each electrical component on the blimp. Two batteries are used to supply 10-volts of power to the blimp. A receiver is used to read signal from a radio transmitter. A MOSFET board is used to combine signal and power for the motors. A servo is used to rotate the motors to change their direction of thrust.

Transmitter Used to Control Blimp: For the pilot to better control the blimp, we decided to use each of the channels on the transmitter to control blimp movement. One of the main problems each of us saw in the demonstration blimp is that the transmitter channels controlled individual components rather than overall blimp movement. For this reason, we decided to use the microcontroller in between the receiver and the components to have controls that were more intuitive. Figure X outlines what we matched each channel on the transmitter to control.

Figure 13: Transmitter Allows for Easy Control of Blimp [2]



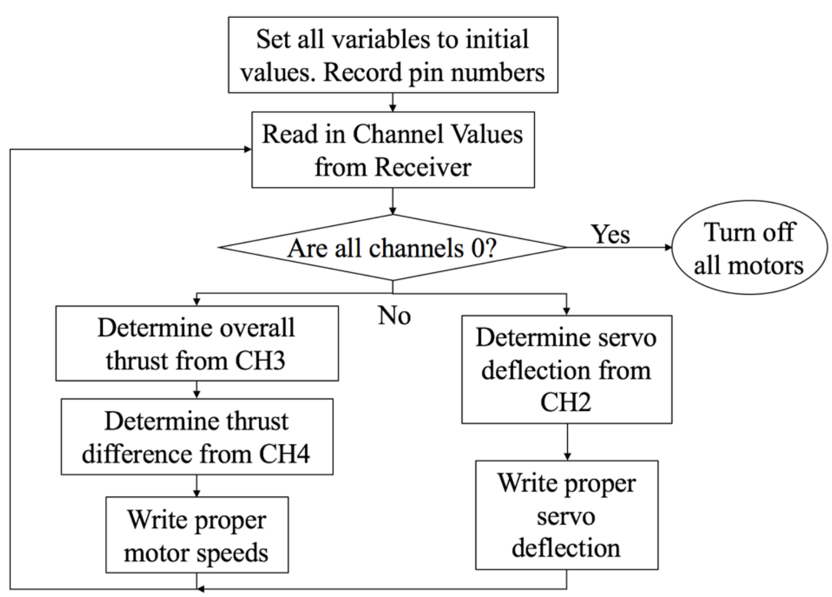
Each blimp movement is achieved by a manipulation of the motor speed or servo. Each movement command is sent to the microcontroller which then determines which component to control and how to control it for the desired movement. To move the blimp faster or stop it, the microcontroller will increase or reduce signal to both MOSFET board's motor signal pins. To move the blimp to the left and right, the microcontroller will decrease the signal to one of the motor pins on the MOSFET board. To go left the right motor signal decreases and to go right the left motor signal decreases. In this way one side of the blimp is propelled more than the other which results in a turning motion. To move the blimp up and down, the microcontroller controls the deflection of the servo. To go up the servo will change the rotation of the motors such that their thrust goes upwards; to go down the servo will change the rotation of the motors such that their thrust goes downwards. Since the servo can also rotate between directing thrust fully up and fully down, it is possible to change the elevation of the blimp while also having some thrust used to control the speed and turning of the blimp.

The purpose of including a microcontroller was to make the blimp more intuitive to control and our pilot believes that the ease of use of the controls was integral to his success. Once the program for the microcontroller was coded it would have also been very easy to change the channels in any way that the pilot felt comfortable, this allows for a wider range of pilots to be comfortable with the controls.

Microcontroller Serves as Hub for Electrical Components: Our main use for the microcontroller was as a hub for all the electrical components in which we could change how the inputs of the system (receiver) affected the outputs of the system (motors and servo). As discussed above, we needed a system that could define movements of the blimp in terms of commands for the motors and servo; our microcontroller program is what we used to complete this task. We ran the program of an Arduino Nano with an ATmega328 chip to minimize the weight of the system and since we were only running simple tasks, the computing power decrease of the smaller board did not seem to impact performance.

Our microcontroller program follows examples made by the Arduino company. First, all "pins" are given more recognizable names so that we know where each component is plugged into the microcontroller. Variables are then set up to record the channel values from the receiver. These are global variables so they can be used at any point in the program. The setup section of the program then runs, which "attaches" each component to the pin we defined for it before and will label the component as an input or output so the microcontroller knows if it should be sending or receiving values. Then the loop section of the program runs which first gets whatever values the receiver is seeing and then uses these values to determine what to tell each of the other components to do. The loop section will then run anywhere from 30 to 60 times per second to readjust the components as the receiver signals change. Appendix A shows our actual microcontroller program with comments for each step as well as a more technical summary. A flowchart of this program is shown in figure 14.

Figure 14: A flowchart representation of our microcontroller program



To make each input translate into motion for the blimp we had to do some computation in between reading in the input values and sending out the output values. The servo's computations are relatively simple and shown in the "setServo()" function in Appendix A. The code takes the variable that has the value read from channel two on the receiver, which can be anywhere from about 800 to about 2000, and then "maps" this range to a corresponding value from about 1000 to 2500, a range of values that the servo better understands. There is also a condition that checks if all channel values are 0 which indicates that the transmitter is off; if this occurs, the servo will simply remain at a 90 degree rotation (the middle of its range of motion). The values that we computed can then be written to servo directly through its signal input. In this way, we effectively define the joystick fully down to be servo tilted all the way in one direction and the joystick fully up to be servo tilted all the way in the other direction; movement of the joystick between these values will move the servo proportionately.

The computations for the motors are slightly more complicated but follow the same general pattern as the servo does. These computations are shown in the "setThrust()" function in Appendix A. The code takes variables with values read from channel 3 and 4 on the receiver which can be anywhere from about 800 to about 2000, and then uses the same map function described above to map these values to a corresponding range of 0 to 255 which the motors understand. There is also a condition that checks if all channel values are 0 which indicates that the transmitter is off; if this occurs, the motors will shut completely off. The values from channel 3 are used to set the overall power for the motors which makes channel 3 act as a "throttle". The values from channel 4 are used to subtract from the channel 3 value, however, only one motors power is subtracted from at a time, which motor is dependent on the side to which the channel 4 joystick is pushed to. In other words, moving the channel 4 joystick will make one motor more powerful than another which results in a turning motion.

Our microcontroller provides great functionality for the system, however, there are still some imperfections in the electrical system which prevent it from reaching its full potential. One of the major problems we had when testing in the atrium was signal noise from other sources. We could tell that this was occurring because sometimes the motors and servo would act erratically if our transmitter was low on battery or off. A possible solution to this would be to implement a communications system that had less crosstalk than our current radio transmitter. Another problem we encountered was that it was difficult to define the bounds for values that we would get from the transmitter. This resulted in our motors turning slightly instead of being completely off when the joystick was down. We ultimately decided that this didn't effect flight so we didn't change it. However, it would be nicer to have a more precise radio control that had more constant bounds. A final problem we had with the system was that each motor responded slightly differently to power, this resulted in one motor spinning slightly faster than the other. This was fixed with the trim tab on transmitter which is designed for exactly that problem. In summary, the biggest improvement we could have made to the system was to have a more precise and direct form of communication to the blimp.

The microcontroller was integral in accomplishing our goals of having the movement of the blimp be simplified down to actions done with the two joysticks on the transmitter. Without the microcontroller working to simplify the flight process we could not have flown as efficiently in the competition as we did.

Total System Mass is Less Than 1000-g Limit: The sea-level prototype has a total system mass of 715 g, which is under the 1000-g mass limit set by NASA. We designed the blimp to be as light as possible because a lighter blimp can achieve a higher acceleration with the same amount of thrust. However, we included 75 g of ballast to account for differences in air density as well as unforeseen extra mass that we might have had to add during construction. Table 1 lists the mass budget for our blimp.

Table 1: Blimp Weighs Less Than 1000-g Limit

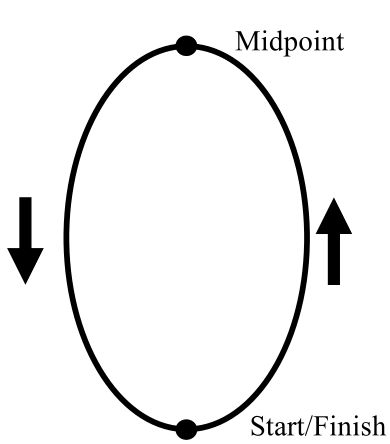
Subsystem	Component	Mass (g)
Payload	Camera	100.00
Lifting System	Envelope	96.62
Structural System	Gondola	55.70
	Tail	19.50
	Gondola Skin	2.35
Propulsion System	Motor with Heat Sink	28.90
	Motor without Heat Sink	24.54
	7-inch Propeller (2)	5.60
	Duct/Motor Mount	48.84
Navigation System	Servo	8.44
	Motor Deflection Mechanism	19.88
Electrical System	Battery Pack (2)	144.32
	Microcontroller	29
	MOSFET Control Board	18.08
	Wiring	28.26
	Receiver	9.63
Ballast		75
Total		715

Performance of Sea-Level Prototype

After the sea-level prototype was fully constructed it was flown through a series of tests and managed to achieve qualifying times for all tests. During these tests our group learned how the sea-level prototype performed in flight and collected data about its speed and maneuverability. The sea-level prototype was able to quickly change elevation as well as rapidly make sharp turns, however, it was discovered that the sea-level prototype had difficulties flying straight and making wide turns. Unfamiliarity with how the sea-level prototype flew contributed to preventable errors in flight and played a large part in its overall performance.

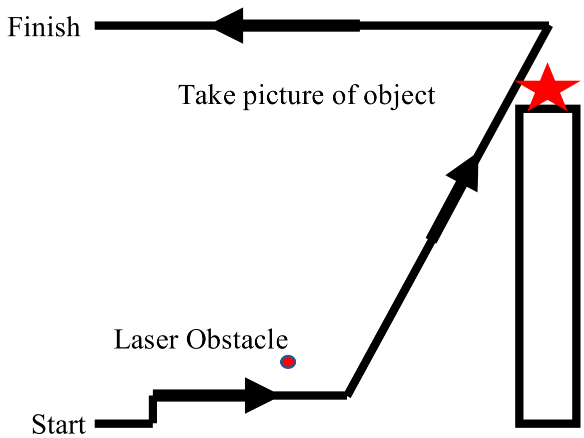
Sea-Level Prototype Completes Speed Test in 35.91 Seconds: The speed test involved taking the sea-level prototype through the FXB atrium to a point twenty meters away then returning to the start. Our sea-level prototype did not place within the top three times for the speed test and our best time during competition was 35.91 seconds. It was nearly impossible to fly straight for more than a few meters, which resulted in longer competition times.

Figure 15: Rendition of NASA Designed Speed Course



Sea-Level Prototype Completes Reconnaissance Test in 46.82 Seconds: The reconnaissance test involved navigating the sea-level prototype below a laser line then rising above a ladder to take a photo of masses in a box. After the picture was taken the number of masses in the image were counted and then checked for accuracy. Our sea-level prototype completed the course in 46.82 seconds; which was the third fastest time in reconnaissance test overall. As the batteries in both the blimp and receiver lost power the sea-level prototype became more difficult to control, which resulted in poorer performance in later trials.

Figure 16: Rendition of NASA Designed Reconnaissance Course



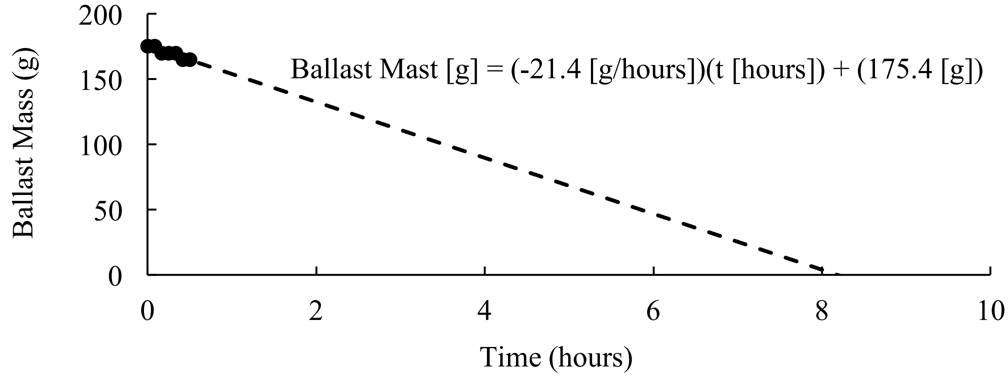
Maximum Speed of Sea-Level Prototype is 1.31 m/s: Over three timed trials the sea-level prototype achieved a maximum speed of 1.31 m/s. This was determined by marking out ten meters then timing how long the prototype took to traverse this distance. After the competition, it was determined that the motor duct's construction played a large impact in the amount of thrust produced by the propellers. If the inner surface of the duct was smooth or the motor mount obstructed less area in the duct it is likely that the sea-level prototype would have achieved greater thrust and greater maximum speed.

Table 2: Sea-Level Prototype Achieves Qualifying Times

Trial #	Speed Test Time (s)	Reconnaissance Test Time (s)	Maximum Speed (m/s)
1	40.60	46.82	0.98
2	35.91	57.82	1.31
3	40.65	88.56	1.25
Average	39.1	64.4	1.18

Sea-Level Prototype Can Stay Aloft For 8 Hours and 10 Minutes: Over a time period of thirty minutes, the amount of ballast required to keep the envelope neutrally buoyant was measured every five minutes. The ballast mass vs. time data was then plotted in a scatterplot and a trendline was drawn using the data. The maximum time aloft was determined to be 8 hours and 10 minutes by extrapolating the trendline until the ballast mass was 0 g. Due to time constraints, the data could not be taken past thirty minutes. More data would result in a more accurate trendline and possibly a new time aloft.

Figure 17: Sea-Level Prototype Stays Aloft Approximately Eight Hours



Atmospheric Conditions at 10-km Above Earth are Hostile
At 10-km, Earth's atmosphere is significantly colder and less dense than the atmosphere at 255-m, and there are extremely fast-moving winds, known as jet streams, at this altitude.

Temperatures are Much Colder at 10-km Altitude: At 10-km altitude, the Earth's atmosphere is significantly colder than at our testing altitude of 255-m. According to Figure 18, Earth's atmospheric temperatures at 10-km altitude range from a minimum of approximately 200 K to a maximum of approximately 260 K. Using equation 1 below, our group found the expected average temperature at 10-km altitude is 223.15 K [1].

Figure 18: Earth's Atmospheric Temperatures Vary with Altitude [1]

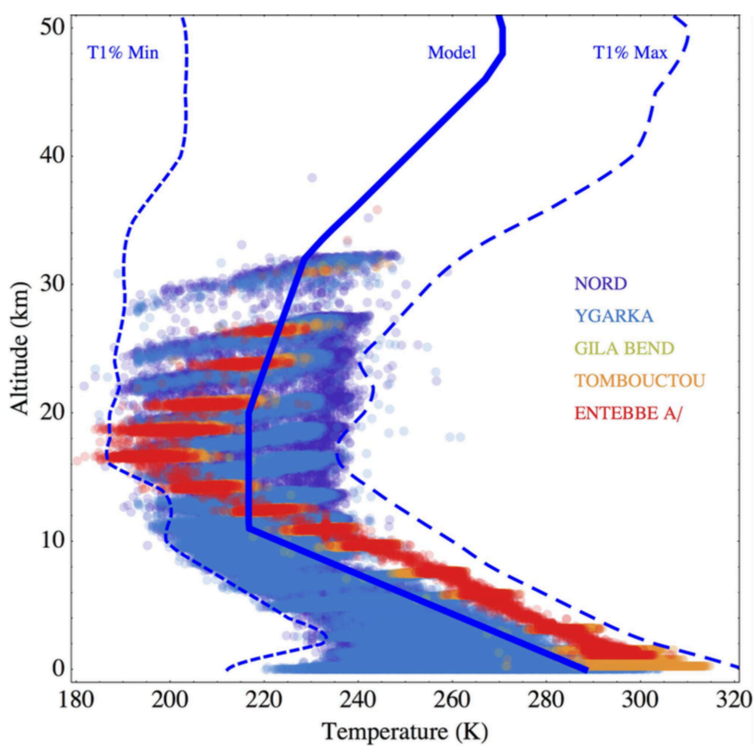


Table 3: Constants for 10-km Altitude Temperature Calculations

Variable Symbol	Variable Name
T	Temperature
C_T	Lapse Rate
h	Altitude above Surface

$$T(h) = T(0m) - C_T h \qquad \text{(eq. 1) [1]}$$
$$T_{avg}(10000m) = 288.15\ K - 0.0065\frac{K}{m} * 10000m = 223.15\ K$$

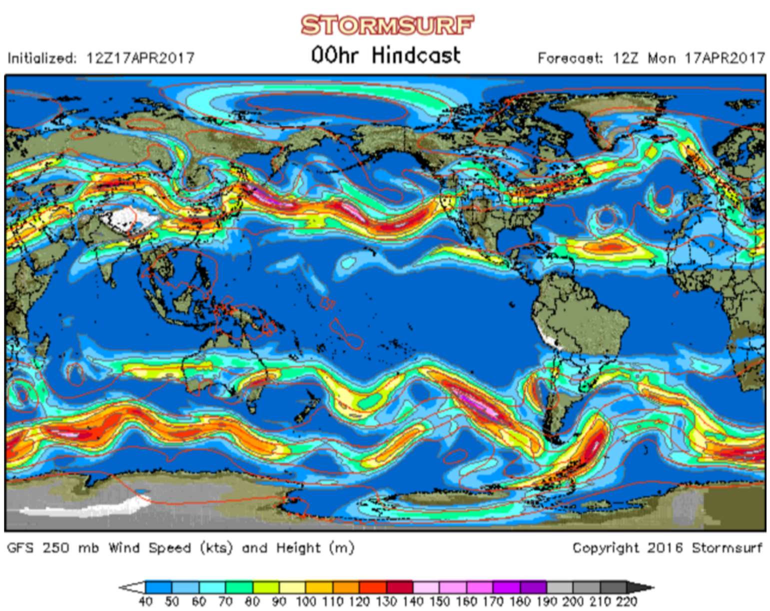
Atmospheric Density Much Lower at 10-km Altitude: The atmosphere at 10-km altitude is significantly less dense than the atmospheric density at sea-level. According to the International Civil Aviation Organization standard atmosphere, the average atmospheric density at 10-km is 0.4137 kg/m^3 [3]. This model-predicted density is approximately 34% of the atmospheric density at sea-level.

$$\frac{\rho_{10\text{-km}}}{\rho_{Earth}} = \frac{0.4137\frac{kg}{m^3}}{1.225\frac{kg}{m^3}} = 0.34 \qquad \text{(eq. 2)}$$

Severe Winds at 10-km Altitude: At 10-km altitude, there are extremely fast winds called jet streams. Jet streams are seasonal in nature, with varying locations based on the time of year and atmospheric temperatures, and they occupy a swath of the atmosphere from 8-km to 15-km altitude, and their typical maximum wind speed is approximately 275-mph, or 123-m/s [4]. Figure 19 below shows a map of the jet streams for April 17, 2017, with wind speeds in knots [5]. Equation 3 below gives the conversion between knots, miles per hour, and kilometers per hour for reference.

$$\text{Speed (kts)} * 1.51\frac{mph}{kts} = \text{Speed (mph)} \qquad \text{(eq. 3)}$$

Figure 19: Map of Global Jet Streams for April 17, 2017 [5]



Hostile Atmospheric Conditions at 10-km Altitude Require Redesign of Sea-Level Prototype
As a result of the atmospheric conditions mentioned in the previous section, an upscale of our sea-level prototype for NASA's blimp will not be suitable for flight at 10-km altitude. The NASA provided propulsion system's approximately 1-N thrust does not provide enough acceleration to have any effect against the wind's force, as our sea-level prototype achieved a maximum speed of 1.31 m/s, which as equation 6 demonstrates below, is approximately 1% of the maximum wind speeds at 10-km altitude.

$$\frac{v_{max,\ prototype}}{v_{max,\ wind}} = \frac{1.31\frac{m}{s}}{123\frac{m}{s}} = 0.0107 \qquad \text{(eq. 6)}$$

As our control system is dependent upon this propulsion system, he extreme wind speeds at this altitude will make our prototype blimp design impossible to control in pitch, and yaw. Furthermore, the structural stability of our blimp is extremely questionable given the high forces which the wind speeds will impart on the blimp.

Other propulsion systems, such as combustion engines and jet turbines, may provide enough thrust to overcome the high winds at 10-km altitude and have some degree of control authority for the blimp, however their fuel requirements preclude our blimp remaining aloft for the one year timeframe defined by NASA.

The current NASA-provided polymeric membrane material for the envelope cannot maintain structural integrity at even the maximum temperature at 10-km altitude, -33 ºC, so a new envelope material will be required for flight at the NASA-specified altitude. Furthermore, our gondola construction methods using NASA-supplied balsa and bass wood and hot glue will become extremely brittle at this temperature, so new materials will be required throughout the blimp for operation at 10-km altitude. Furthermore, the extremely low temperatures at 10-km altitude will severely impair the performance of our NASA-supplied power system, consisting of rechargeable AAA batteries, which produced significantly less power in colder environments, which in turn will reduce the effectiveness of our electrical system as a whole.

Two Redesign Options Considered: Taking these considerations into account, our sea-level prototype blimp will require a total redesign to meet any of NASA's requirements for the blimp at 10-km altitude. Two separate paths for a redesign were considered by our group, neither of which would fulfill all of NASA's requirements.

We considered requesting a change in the operational altitude of our blimp, which would require an upscale of our existing system, along with new materials for the gondola and envelope to accommodate the extremely cold temperatures at altitude. Furthermore, this redesign would require new power and propulsion systems, to provide adequate power for a one-year flight endurance, and to provide adequate thrust in the thinner air at a high altitude on Earth.

The other redesign considered would remain at the NASA-specified 10-km operating altitude, however, due to the high winds at this altitude, we would remove any propulsion and control system from the blimp, leaving it as an unpowered balloon carried by the wind. Due to the removal of the propulsion system, the envelope does not need to be directional, so the envelope would be changed to a spherical design to maximize the mass to volume ratio of the envelope.

Both designs were considered in the context of their ability to meet NASA's requirements for the high-Earth altitude blimp, with our chosen design driver being our ability to meet NASA's specified altitude requirement. In this regard, the unpowered balloon concept was considered to be more preferable than the rescaled blimp, which would require a change in altitude to avoid the high winds at 10-km. Furthermore, the unpowered balloon will have a significantly lower power supply requirement, as compared to a powered blimp. The envelope design and construction for the smaller balloon will also be simplified, and the mass to volume ratio of a spherical envelope is more ideal than the rescaled blimp's C-class envelope, which is an important consideration, given the likely higher density of the new envelope material.

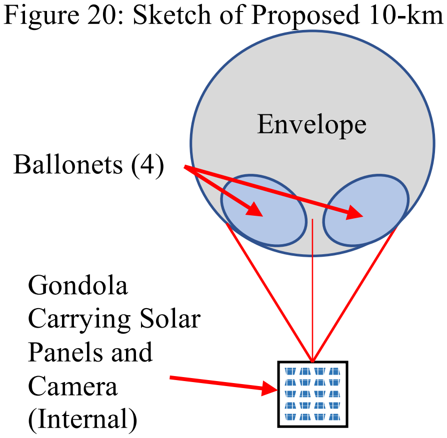
Both designs will likely exceed NASA's 1-kg system mass limit, however, the balloon design will be significantly lighter than the rescaled blimp design, due to the absence of propulsion systems and the power supplies. Neither design is likely to fit within the storage constraints imposed by NASA, due to the need for a larger envelope to accommodate the lower density of air at high altitudes on Earth. Furthermore, the unpowered balloon will not be controllable, as its position will be controlled by the surrounding winds.

Although neither of the designs which we considered would be able to fulfill all of NASA's requirements, our group determined that an unpowered balloon with a spherical envelope would be most able to meet NASA's overarching requirement for a blimp which can operate at 10-km altitude above Earth.

Unpowered Balloon Design Recommended for 10-km Altitude

Due to the atmospheric conditions at 10-km altitude above Earth, our group is proposing a complete redesign of our system as an unpowered balloon. Our group has determined that the materials provided by NASA will be unsuitable for this altitude, given the extremely cold environment, and we are requesting that NASA supply us with new materials, including extruded polystyrene foam for the gondola and a NASA/ Dupont developed polyester fabric/film laminate for a new, spherical envelope. The redesigned envelope will have a radius of 1.28-m, and its volume of 8.78-m^3 is approximately 10 times larger than the volume of our sea-level prototype. Unfortunately, due to the redesigned systems and new envelope material and size, our design has a projected mass of 3200-g, which exceeds NASA's 1000-g mass limit, and therefore, we are requesting an increase in the mass budget for our design. Furthermore, we are unable to meet NASA's speed and maneuverability requirements due to our redesign's lack of propulsion.

Figure 20: Sketch of Proposed 10-km Balloon (Note: Sketch Not to Scale)



New Envelope Material Required: At a 10-km altitude in Earth's atmosphere the current material is entirely unsuitable. To fix this, we propose a new NASA/Dupont material constructed with "a bi-laminate of woven polyester fabric and thin homogeneous film of polyester" that has been tested on numerous NASA long endurance balloons. The envelope will be pressurized to avoid excess lifting gas loss and allow the balloon to stay aloft for one year. The density of this new material will be 0.03 kg/m^2, which will allow us to stay close to the original mass limit. The new material can also withstand up to -80°C, which will be suitable for 10-km above Earth. [6]

Figure 21: Example of NASA Long-Endurance Balloon using NASA/Dupont Envelope Material [7]

New Power System Required: The NASA-supplied battery packs for our sea-level prototype will not be suitable for our redesign for 10-km altitude, as they will not operate efficiently at the extremely cold temperatures at this altitude, and they will not be able to provide enough endurance to allow the balloon to operate for one year. To allow the power system to have one year of endurance, our redesign incorporates exterior-mounted solar panels and internal battery systems designed for "CubeSat," or nanosatellite, applications. These systems are designed to survive years of exposure to extreme temperature gradients in the vacuum of space, and are relatively efficient due to the mass limits imposed on nanosatellite systems. Furthermore, the systems will be able to withstand buffeting by the jet streams at 10-km, as they are designed to withstand high loads and accelerations as the systems are launched into orbit.

New Simplified Gondola Required: The gondola materials utilized in our sea-level prototype of NASA's blimp are unsuitable for the extremely cold environment at 10-km altitude, and our prototype gondola design is unnecessarily complicated, given the unpowered nature of our balloon redesign. To this end, our redesign incorporates a cube-shaped gondola suspended directly below our envelope, and is constructed out of extruded polystyrene (XPS) foam. The XPS foam construction will result in a sturdy and lightweight gondola, with the additional benefit of providing some degree of insulation for the components housed within from the external temperature.

The gondola will carry any required transmitters and receivers to track the balloon and retrieve data from the system, as well as the NASA-supplied camera payload, and the batteries required to power these systems. The solar panels, used to recharge the batteries carried internally while the balloon is in daylight, will be mounted on the four exterior sides of the gondola. Mounting the majority of the electronic components, and the camera payload, inside the fuselage will allow us to better protect them from the previously mentioned jet streams at 10-km altitude. Due to the extremely cold temperatures at this altitude, the gondola will also likely require heaters to maintain the electrical components at a reasonable temperature, which will also be powered by the batteries.

Ballonets Maintain Altitude: To control altitude of the balloon we have incorporated ballonets into our design. By adding or removing air from the ballonets the pressure and average density of the envelope can be changed in flight. This allows the balloon to maintain a constant altitude 10-km altitude as required by NASA and stay pressurized. [1]

Envelope Size Increases Significantly: Our unpowered balloon design for 10-km altitude has a projected system mass of 3200-g, which unfortunately exceeds NASA's 1000-g limit for the system mass. Table 4 below shows our current prediction for the system mass of the redesigned balloon.

Table 4: Balloon Exceeds 1000-g Limit

Component	Mass (g)
Imaging System	380
Gondola Structure (XPS Foam)	1000
Electrical System	500
Envelope	770
Margin	500
Total	3200

Using the expected 3200-g system mass, we were able to determine the radius of our redesigned envelope. Due to the unpowered nature of the balloon, a spherical envelope was chosen for the 10-km altitude balloon design, as this design maximizes the mass-to-volume ratio of the envelope, providing the most lift per unit mass.

Table 5: Constants and Variables for Redesigned Envelope Calculations

Symbol	Description	Value
m_{system}	System mass (without envelope)	2.38 kg
$\sigma_{material}$	Area density of envelope material	$0.0375\ \frac{kg}{m^3}$
ρ	Atmospheric density (general)	-
h	Height above sea level	10000 m
ρ_0	ISA sea level density (general)	-
$\rho_{air\ 0}$	ISA sea level density for air	$1.225\frac{kg}{m^3}$
$\rho_{He\ 0}$	ISA sea level density for helium	$0.169\frac{kg}{m^3}$
T	Atmospheric temperature	-50 K
T_0	ISA sea level temperature	288.15 K
C_T	Specific Temperature	$0.0065\frac{K}{m}$
ρ_{air}	Air density at altitude	$0.5135\ \frac{kg}{m^3}$
ρ_{He}	Helium density at altitude	$0.0564\ \frac{kg}{m^3}$
V	Envelope volume	0.890 m^3
g	Gravitational acceleration	$9.8\ \frac{m}{s^2}$
SA	Envelope surface area	20.6 m^2
r	Envelope radius	1.28 m

Equations 7 and 8, below, reiterate the standard atmosphere-derived equations for modifying atmospheric density for altitude.

$$\rho(h) = \rho_0 \left(\frac{T(h)}{T_0}\right)^{4.3} \qquad \text{(eq. 7)}$$

$$T(h) = T_0 - C_T h \qquad \text{(eq. 8)}$$

Equation 8 is used to find the average atmospheric temperature at 10-km, or 10000-m, as in Equation 9, and this calculated value of -50-K can then be utilized, along with the values for the densities of air and helium at sea-level from Table 5, above, in Equation 7 to calculate the modeled densities of air and helium at 10-km, $0.0564\ \frac{kg}{m^3}$ and $0.5135\ \frac{kg}{m^3}$, respectively.

$$T(10000\ m) = T_0 - C_T * 10000\ m = -50\ K$$

$$\rho_{He}(10000\ m) = \rho_{He\ 0}\left(\frac{T(10000\ m)}{T_0}\right)^{4.3} = 0.0564\ \frac{kg}{m^3}$$

$$\rho_{air}(10000\ m) = \rho_0\left(\frac{T(10000\ m)}{T_0}\right)^{4.3} = 0.5135\ \frac{kg}{m^3}$$

The radius of the redesigned envelope is calculated by setting the net lift of the system equal to 0, and solving for the radius, which is a common term in the volume and surface area terms, in Equation 9, below. The calculated radius of our redesigned, spherical envelope is 1.28-m.

$$L_{Net} = L_{Gross} - W_{He} - W_{System} - W_{Envelope} \qquad \text{(eq. 9)}$$

$$0 = \rho_{air}Vg - \rho_{He}Vg - m_{system}g - \sigma_{material}SA$$

$$0 = (\rho_{air} - \rho_{He})(\frac{4}{3})\pi r^3 - m_{system} - \sigma_{material}4\pi r^2$$

$$r = 1.28\ m$$

Redesign Cannot Meet all NASA Requirements: Our redesigned, unpowered balloon to fulfill NASA's requirements for a blimp at an altitude of 10-km above Earth has a projected mass of 3200-g, which unfortunately exceeds NASA's 1000-g mass limit, and therefore, our group is requesting an increased mass budget. Furthermore, our group recommends that NASA consider a new imaging system for operations at 10-km altitude due to the high cloud cover which exists below this altitude. An infrared camera, such as the Optris PI lightweight, which is designed for aerospace applications, would allow our balloon to perform reconnaissance duties for NASA regardless of the weather below its altitude, and we have considered the increased mass of this system in our projected mass budget [8]. Unfortunately, due to the high-speed jet streams which exist at 10-km altitude, our redesigned balloon will be propelled by the winds, so we will be unable to control its position, meaning we cannot meet NASA's speed and maneuverability requirements for the system at 10-km altitude.

CONCLUSIONS

We have successfully designed and built a sea-level prototype of NASA's high-Earth blimp. In addition, we have also conducted speed, reconnaissance, and time aloft tests on the sea-level prototype. Afterwards, we researched Earth's environment at 10-km altitude and developed a proposal for a high-Earth version of our blimp.

Design of Sea-Level Prototype

Our sea-level prototype is a remotely controlled blimp with a C-class shaped envelope. Its gondola is made of a combination of bass wood and balsa wood and houses all the blimp's electrical components. The gondola also has a fin made of polymeric membrane material with a wooden support structure to stabilize the blimp during flight. Furthermore, the blimp is powered by two battery packs producing a total voltage of 10 V. The batteries are connected to a microcontroller that provides differential thrust to allow the blimp to turn and a MOSFET board that distributes power to two motors. These motors are mounted 30 inches apart along the axis of the blimp's center of mass and produce thrust to propel the blimp forward. They can also be deflected 180 degrees by a mechanism consisting of a servo and two gears with a 2:1 ratio, allowing the blimp to change its altitude. Lastly, the blimp can receive radio signals through its receiver and can be controlled via a transmitter.

Performance of Sea-Level Prototype

We conducted three tests using our sea level prototype. The first was a speed test in which we assessed the time it took for our blimp to fly to and back from a point about 20 m away. Our sea-level prototype blimp was able to complete the speed test in an average of 34 seconds. During this test, we also determined that our blimp could reach a maximum speed of 1.31 s. The second test was a reconnaissance test in which we assessed the time it took for our blimp to fly under a laser and up to a target, take a picture of the target, and return to its starting location. Our blimp was able to complete the reconnaissance test in an average of 59 seconds. The last test was to determine how long our blimp could stay aloft without refilling the envelope with helium. We found that our blimp could do this for 8 hours and 10 minutes.

Adaptations for 10-km Above Earth

Due to high wind speeds of up to 123 m/s (275 mph), no means of propulsion would be powerful enough to control a blimp at 10-km altitude on Earth. Therefore, a blimp is not a practical vehicle to conduct surveillance in this environment. As a result, we propose changing the design to an uncontrolled spherical balloon that would be freely propelled by winds. It would have an insulating foam gondola covered by solar panels, an electric heater, and an infrared camera. We expect it to have a mass of 3200 g, a volume of 8.78 m^3, and a radius of 1.28 m. Since the mass exceeds the 1000-g limit, we would need to request a higher mass limit from NASA.

REFERENCES

[1] Washabaugh, P.D. (2017). University of Michigan ENGR 100-700 Lab Manuals Winter 2017.
[2] HiTec Laser4 4CH 555 Transmitter. (n.d.). Retrieved April 18, 2017, from http://www.gravesrc.com/hitec-laser4-4ch-555-transmitter-with-3-hs-81.html
[3] International Civil Aviation Organization and Langley Aeronautical Laboratory. (1952, November 10). Report 1235: Standard Atmosphere-Tables and Data for Altitudes to 65,800 Feet.
[4] NOAA. (2017). The Jet Stream . Retrieved April 5, 2017, from National Weather Service JetStream: http://www.srh.noaa.gov/jetstream/global/jet.html
[5] STORMSURF. (2017, April 4). Weather Model - Global Jet Stream Wind and 250 mb Pressure. Retrieved April 17, 2017, from http://www.stormsurfing.com/cgi/display_alt.cgi?a=glob_250
[6] Said, M. A. (1999). Mechanical Behavior of Fabric-Film Laminates. NASA Goddard Spaceflight Center.
[7] NASA. (2016, December 20). Scientific Balloon. (M. Johnson, Editor) Retrieved April 6, 2017, from Columbia Scientific Balloon Facility: https://www.nsbf.nasa.gov/balloons.html
[8] Optris GmbH. (2017). PI Lightweight. Retrieved April 7, 2017, from Optris Infrared Thermometers: http://www.optris.com/pi-lightweight-netbox
[9] Language Reference. (n.d.). Retrieved April 15, 2017, from https://www.arduino.cc/en/Reference/HomePage

HONOR CODE

We have neither given nor received aid on this assignment nor have we concealed a violation of the honor code.

Seamus Callaghan 4/18/2017
Alex Chen
Benjamin Foster
Ryan Johnston
David Koelzer

APPENDIX A: MICROCONTROLLER PROGRAM AND COMMENTS

```
/*
 * main.ino
 * Code designed for mapping transmitter values intuitively to motors
 *
 * Ryan Johnston (ryanpj@umich.edu)
 * git@gitlab.eecs.umich.edu:ryanpj/Blimp-MotorMapping.git
 */

#include <Servo.h>            //Import the servo library for use

/*
 * CODE SUMMARY
 * The code above the setup() function is meant to firstly define the pins
 * on the microcontroller (where each input/output is plugged in). Secondly
 * it sets up variables for recording and setting values for each input or
 * output. The setup() function then takes this information and uses
 * Arduino syntax to correctly show them to the microcontroller. For the
 * rest of the program the loop() runs multiple times per second. It will
 * first read in values with the readInputs() function then use these to
 * compute what signal to send to the MOSFET board and the servo (through
 * functions setThrust() and setServo() respectively.
 */

//Define which pins we are using for INPUTS and OUTPUTS
//All 23 other pins are not used and nothing is plugged in
const int ch1Pin = 1;         //Not used-pin defined to remember connection
const int ch2Pin = 2;         //Used for speed control (INPUT)
const int ch3Pin = 3;         //Used for altitude control (INPUT)
const int ch4Pin = 4;         //Used for turning control (INPUT)
const int leftMotorPin = 5;   //Used for left motor (OUTPUT)
const int rightMotorPin = 6;  //Used for right motor (OUTPUT)
const int servoPin = 11;      //90 degree servo (OUTPUT)

int ch2Value;                 //Setup a value for recording receiver ch2
int ch3Value;                 //Setup a value for recording receiver ch3
int ch4Value;                 //Setup a value for recording receiver ch4
int servoValue = 90;          //Initial servo value of 90 degrees (middle)
int servoMicroseconds = 0;    //Setup a value for fine servo controlling

Servo deflectionServo;        //Initialize a servo using the Servo library

// the setup function runs once when you press reset or power the board
void setup() {
  //Uses the pin numbers from above to setup the program for running
  pinMode(13, OUTPUT);              //Setup a pin for the LED
  pinMode(ch2Pin, INPUT);           //Setup a pin for the receiver ch2
  pinMode(ch3Pin, INPUT);           //Setup a pin for the receiver ch3
  pinMode(ch4Pin, INPUT);           //Setup a pin for the receiver ch4
  //Setup a pin for the left motor (MOSFET)
  pinMode(leftMotorPin, OUTPUT);
  //Setup a pin for the right motor (MOSFET)
  pinMode(rightMotorPin, OUTPUT);
```

```
  deflectionServo.attach(servoPin); //Setup a pin for the servo
  //Write an initial value of 90 to the servo
  deflectionServo.write(servoValue);

  Serial.begin(9600);               //Begin serial output for debugging

}

// the loop function runs continuously forever(many times per second)
void loop() {
 //Calls three functions (each is implemented below under "HELPER METHODS")
  readInputs();   //Assign inputs from the receiver to variables
  setThrust();    //Compute the thrust to send to the MOSFET
  setServo();     //Compute the position to send to the servo

  digitalWrite(13, LOW); //Keeps the LED on the Arduino board off
}

//HELPER METHODS

//Computes servo deflection from given channel values read by readInputs.
void setServo(){
  //Set the servo to 0 if the transmitter is off
  if(ch2Value==0 && ch3Value==0 && ch4Value==0){
    deflectionServo.write(90);      //Moves the servo to a middle position
    return;                         //Don't run the rest of the function
  }

  //"Map" deflection from ch2 values (TUNE CH2 MAX/MIN HERE)
  servoMicroseconds = map(ch2Value,800,2000,1000,2500);
  //servoMicroseconds are used for more precise control, for an explanation
  of how servos are controlled in this manner see the language reference[9]
  deflectionServo.writeMicroseconds(servoMicroseconds);
}

//Computes thrust from given channel values read by readInputs
void setThrust(){
  //Shutoff the motors if the transmitter is off
  if(ch2Value==0 && ch3Value==0 && ch4Value==0){
    analogWrite(leftMotorPin, 0);   //0 is the off value for the motor
    analogWrite(rightMotorPin, 0);
    return;                         //Don't run the rest of the function
  }

  //Overall throttle is how much power to send to both motors
  //Map overall throttle from ch3 (TUNE CH3 MAX/MIN HERE)
  int throttle = map(ch3Value,800,1650,0,255);

  //Variables for how much to subtract from current throttle (Used for
  turning)
  int rightMotorDiff = 0;
  int leftMotorDiff = 0;


  //CH4 Center ~1300 (This is what value is read when the joystick is in
```

```
  the middle)
  int center = 1300;
  //If the joystick is to the left, subtract power from the left motor to
  initiate a left turn
  if(ch4Value<center){
    rightMotorDiff = 0;
    leftMotorDiff = (center-ch4Value);
  //If the joystick is to the right, subtract power from the right motor to
  initiate a right turn
  } else {
    rightMotorDiff = (ch4Value-center);
    leftMotorDiff = 0;
  }

  //Do the actual subtraction of computed values above
  int leftThrottle = throttle-leftMotorDiff;
  int rightThrottle =  throttle-rightMotorDiff;

  //Keep within motor power bounds (accounts for receiver glitches/
  imperfect bounds)
  if(leftThrottle>255)
    leftThrottle = 255;
  if(leftThrottle<0)
    leftThrottle=0;
  if(rightThrottle>255)
    rightThrottle = 255;
  if(rightThrottle<0)
    rightThrottle=0;


  //Write computed values to motors
  analogWrite(leftMotorPin, leftThrottle);
  analogWrite(rightMotorPin, rightThrottle);

  //DEBUG Output
  Serial.print("Throttle: ");
  Serial.print(throttle);
  Serial.print(" | ");
  Serial.print("Right Motor Diff: ");
  Serial.print(rightMotorDiff);
  Serial.print(" | ");
  Serial.print("Left Motor Diff: ");
  Serial.print(leftMotorDiff);
  Serial.println();
}

//Reads values from the controller and assigns to global variables for use
in other functions
void readInputs(){
  ch2Value = pulseIn(ch2Pin,HIGH);  //Read in value from ch2 on receiver
  ch3Value = pulseIn(ch3Pin,HIGH);  //Read in value from ch3 on receiver
  ch4Value = pulseIn(ch4Pin,HIGH);  //Read in value from ch4 on receiver


  //DEBUG Output
```

```
  Serial.print("CH2 Value: ");
  Serial.print(ch2Value);
  Serial.println();
  Serial.print("CH3 Value: ");
  Serial.print(ch3Value);
  Serial.println();
  Serial.print("CH4 Value: ");
  Serial.print(ch4Value);
  Serial.println();
}
```

APPENDIX B: CALCULATIONS FOR VARIOUS PREDICTED ENVELOPE PROPERTIES

Table 6: Variables and Constants for Envelope Calculations

Symbol	Description	Symbol	Description
θ	Parameterization variable	ρ_0	ISA sea level air density
L	Envelope length	ρ_{air}	Air density
V	Envelope volume	ρ_{He}	Helium density
t	Integration variable	ρ_{env}	Area density of envelope material
N	Number of envelope sections	m_{other}	System mass (without envelope)
A_{env}	Surface area of envelope	m_{env}	Envelope mass
T	Atmospheric temperature	L_{net}	Net lift
T_0	ISA sea level temperature	L_{gross}	Gross lift
C_T	Specific Temperature	W	Total weight
h	Height above sea level	g	Gravitational acceleration
D	Drag	F_T	Thrust
C_D	Drag coefficient	A_p	Propeller area
U	Maximum velocity	η	Motor efficiency
A_{craft}	Blimp cross-sectional area	P	Power through motors

Parameterization of envelope surface in terms of length:

$$\Phi(x,y,z) = \big(x(\varphi,\theta), y(\varphi,\theta), z(\varphi,\theta)\big) = \left(L\tfrac{1}{2}\left(\cos\varphi + \tfrac{1}{4}\cos^2\varphi\right),\ L\tfrac{1}{4}\sin\varphi\cos\theta,\ L\tfrac{1}{4}\sin\varphi\sin\theta\right);$$
$$0 \le \theta \le 2\pi, \qquad 0 \le \varphi \le \pi$$

Expected envelope volume in terms of length:

$$\vec{g} = (x,0,0),\ \ \therefore\ \vec{\nabla}\cdot\vec{g} = 1$$
$$V = \oiiint dV = \oiiint \vec{\nabla}\cdot\vec{g}\, dV = \oiint \vec{g}\cdot\left(\Phi_\varphi \times \Phi_\theta\right)d\theta d\varphi\ = \frac{\pi}{24}L^3$$

Required envelope length for neutral buoyancy at 10% below average air density:

$$0.9\ \rho_{air}V = \rho_{He}V + \rho_{env}A_{env} + m_{other}$$
$$0.9 * 1.20\,\frac{kg}{m^3} * \frac{\pi}{24} * L^3 = 0.169\,\frac{kg}{m^3} * \frac{\pi}{24} * L^3 + 1.169 * 0.022\frac{kg}{m^2} * L^2 + 0.668\ kg, \quad \therefore L = 1.85\ m$$

Volume:

$$V = \frac{\pi}{24}L^3 = \frac{\pi}{24} * 1.85^3\ m^3 = 0.829\ m^3$$

Envelope surface area:

$$A_{env} = 1.169 * L^2 = 1.169 * 1.85^2\ m^2 = 4.00\ m^2$$

Envelope mass:

$$m_{env} = \rho_{env}A_{env} = 0.022\,\frac{kg}{m^2} * 4.00\ m^2 = 0.088\ kg$$

Parameterization of envelope cross section in terms of length:

$$\vec{f}(\theta) = \big(f_x(\theta),\ f_y(\theta)\big) = \left(L\tfrac{1}{2}\left(\cos\theta + \tfrac{1}{4}\cos^2\theta\right),\ L\tfrac{1}{4}\sin\theta\right)$$

Parameterization of envelope section planform border in terms of envelope length:

$$\vec{c}(\theta) = \left(\int_0^\theta dl(t)\,,\ n(\theta)\right) = \left(\int_0^\theta \sqrt{f_x'^2(t) + f_y'^2(t)\cos\left(\frac{\pi}{N}\right)^2}\,dt,\ f_y \sin\left(\frac{\pi}{N}\right)\right) =$$

$$\left(L\int_0^\theta \sqrt{\frac{1}{4}\sin(t)^2\left(1+\cos t+\frac{1}{4}\cos(t)^2\right)+\frac{1}{16}\cos(t)^2\cos\left(\frac{\pi}{N}\right)^2}\,dt\,,\ L\frac{1}{4}\sin\theta\sin\left(\frac{\pi}{N}\right)\right)$$

Envelope surface area in terms of envelope length (calculated numerically):

$$A_{env} = 2N\int_0^\pi n(\theta)dl(\theta) = 1.169 * L^2$$

Center of lift (measured from back of envelope):

$$\bar{x} = \frac{5}{8}L + \frac{\int_0^\pi f_x(\theta) * f_y(\theta)^2 * f_x{}'(\theta)\,d\theta}{\pi L^3/24} = \frac{17}{40}L = \frac{17}{40} * 1.85\text{ m} = 0.786\text{ m}$$

APPENDIX C: PYTHON CODE FINDS ENVELOPE SURFACE AREA AND PLANFORM COORDINATES

This code takes the length of our desired envelope and the number of desired planforms as input and outputs envelope length, volume, selected coordinates of planform border, planform area, and envelope surface area, with plot of planform shape.

Input and Calculations

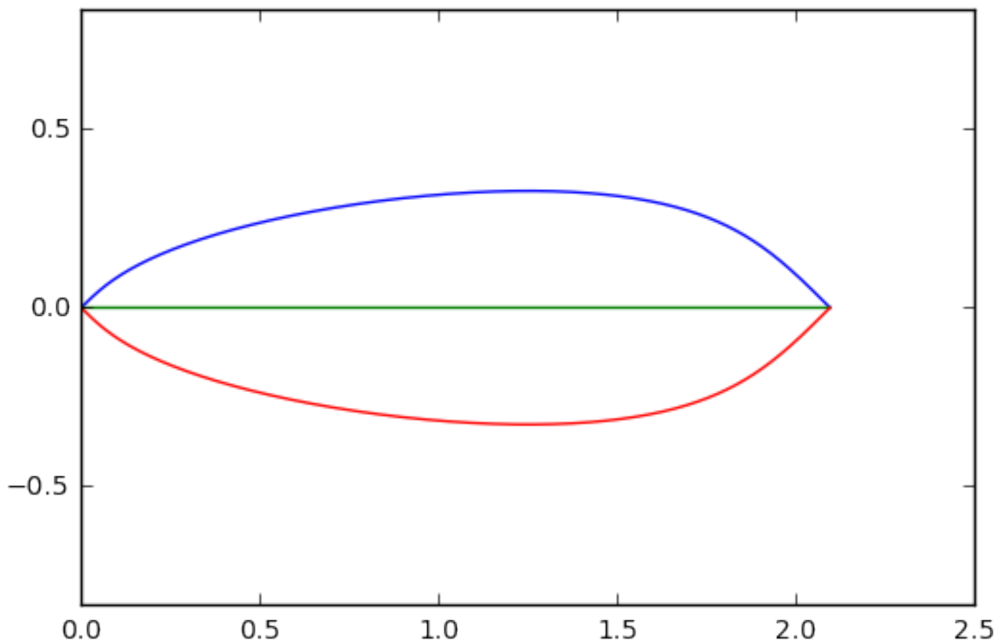
```
[In]:

import numpy as np                         ## numpy is a library that
inclues most of the necessary numerical functions
import matplotlib.pyplot as plt            ## this is the library will be
used to plot the planform

L = 1.85  #Set length of the envelope
N = 4.    #Number of envelope sections

#Set up the parameterization variable (angle theta)
steps = 10001
d_theta = np.pi / (steps - 1)
theta = np.arange(steps) * d_theta

#Calculate volume using analytically derived equation
V = np.pi * L**3. / 24.

print "length =", L, "meters"
print "Volume =", V, "m^3"

#create arrays to hold planform border x and y coordinates
plan_x = np.zeros(np.size(theta))
plan_y = np.zeros(np.size(theta))

#fill in y coordinates of planform border
plan_y = L * 1./4. * np.sin(theta) * np.sin(np.pi/N)

#create array to hold differentials of planform area
d_A = np.zeros(np.size(theta))

#for use in construction
print "Planform border coordinates:"

#iterate
i = 0
while i < steps:
    #t is a shorter way of saying theta[i]
    t = theta[i]

    #calculate differential of arc length for the x-coordinates of the
     planform outline
    d_l = L * np.sqrt(1./4. * (np.sin(t))**2. + 1./4. * np.cos(t) *
     (np.sin(t))**2. + 1./16. * (np.cos(t) * np.sin(t))**2. + 1./16. *
     (np.cos(t) * np.cos(np.pi / N))**2.) * d_theta
```

```
    #set x-coordinates for planform outline to be previous x-coordinate
     plus change in x (arc length differential).
    plan_x[i] = plan_x[i - 1] + d_l

    #since the above step is recursive, for x_plan[0], it attempts to
     access plan_x[-1], which doesn't exist.  This sets the first x-
     coordinate to 0
    if (i == 0):
        plan_x[i] = 0

    #differential of surface area
    d_A[i] = d_l * plan_y[i]

    #print enough planform border coordinates for use in construction,
     spaced out at regular intervals of theta
    if (i % ((steps - 1)/20) == 0):
        print "x =", plan_x[i],"y =", plan_y[i]

    i += 1

#calculate planform and envelope surface area, print
plan_A = 2 * np.sum(d_A)
print "Area of planform:", plan_A, "m^2"
total_A = N * plan_A
print "Total surface area:", total_A, "m^2"

#plot shape of planform.  Units are in meters
plt.plot(plan_x, plan_y)
plt.plot(plan_x, np.zeros(np.size(theta)))
plt.plot(plan_x, -1. * plan_y)
plt.axis('equal')
plt.show()
```

Output

```
[Out]:

length = 1.85 meters
Volume = 0.828807774387 m^3
Planform border coordinates:
x = 0.0 y = 0.0
x = 0.0547422312815 y = 0.0511598403566
x = 0.125967162812 y = 0.101059955654
x = 0.220090123802 y = 0.148471639444
x = 0.336768156903 y = 0.192227458722
x = 0.472546650996 y = 0.23125
x = 0.622435624596 y = 0.264578398803
x = 0.780738095747 y = 0.291391999342
x = 0.941635886872 y = 0.311030561783
x = 1.09966210916 y = 0.323010519542
x = 1.25008413946 y = 0.327036886299
x = 1.38919368578 y = 0.323010519542
x = 1.51449738463 y = 0.311030561783
x = 1.62480577938 y = 0.291391999342
x = 1.72022262066 y = 0.264578398803
```

```
x = 1.8020331546 y = 0.23125
x = 1.87247457089 y = 0.192227458722
x = 1.93435843217 y = 0.148471639444
x = 1.99056082607 y = 0.101059955654
x = 2.04353857355 y = 0.0511598403566
x = 2.09512438208 y = 4.00504676009e-17
Area of planform: 1.00046029111 m^2
Total surface area: 4.00184116445 m^2
```



APPENDIX D: PROJECT MANAGEMENT TOOLS

Figure 22: All Deadlines for Project Met

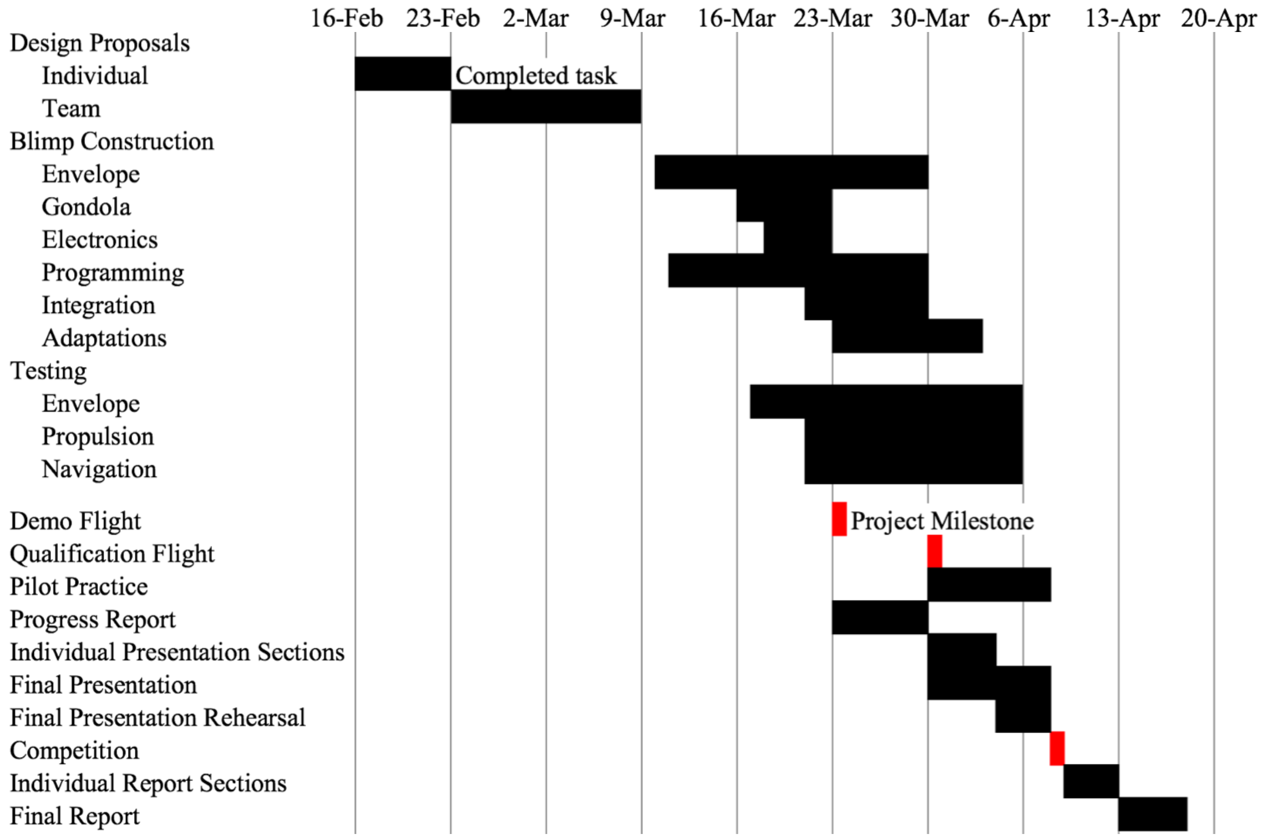


Table 7: Project Roles Distributed Evenly Among Group Members

Name	Construction Management	Hands-On Construction	Reporting Management
Seamus Callaghan	Envelope	Envelope Construction, Gondola Assembly	Oral Progress Pres
Alex Chen	Electronics	Electronics Wiring, Envelope Construction	Final Written Report
Benjamin Foster	Gondola	Gondola Assembly, Motor Assembly	Oral Design Pres
Ryan Johnston	Controls	Microcontroller Programming, Envelope Construction	Final Oral Pres
David Koelzer	Tail	Motor Assembly Gondola Assembly	Final Written Report
All	Create and present all reports and presentations		