PLANET

Planetary Landings, Atmospheres, and Nebulae: Explorations and Testing
709 N. University Avenue
Ann Arbor, MI 48105

Dear Dr. Washabaugh:

As you know, NASA wants to send a blimp to the planet Venus to survey the surface. In order to be considered suitable for deployment, this blimp has a variety of parameters that it must meet. It is to carry a 100-gram camera as a payload, and it must be speedy, maneuverable, and made only of NASA materials. NASA also stipulated that the entire structure must fit in a 23 cm x 56 cm x 91 cm container and weigh less than 1000 grams total. Before sending anything to Venus, however, NASA wants a contractor team to design, build, and test an Earth-based prototype that must meet the same criteria as the Venus design. Thus, you have contracted us to construct a blimp and test its performance in speed and recon tests, and then report back to you on the results of our efforts. Also, you tasked us with adapting our design for deployment on Venus. The report that follows presents our blimp design, Earth performance, and adaptations for Venus.

To summarize, our blimp design consists of a 2:1 fineness ratio spheroid envelope attached to a wooden gondola, which folds for easy storage. To fly, our design incorporates vectored thrust and an Arduino microcontroller. Most importantly, our blimp is under the 1000-gram mass limit, and it meets all of NASA's requirements.

In terms of performance, our blimp achieved an average time of 25 seconds on the speed test, with its fastest time being 22 seconds. On the recon test, our blimp's average time was 63 seconds, with a fastest time of 47 seconds. Also, it is important to note that our blimp can stay aloft, according to our calculations, for 102 minutes. We believe that, based on these results, our blimp will work well if deployed on Venus.

To take it to Venus, we have determined that, keeping lift constant, we can reduce the blimp's size by a geometric scale factor of $S_G = 0.26$. Additionally, NASA's membrane material will not survive in Venus' atmosphere, so we recommend using a new material called Polybenzoxazole. Finally, the batteries and wood will not work on Venus either. We have no suggestions thus far to replace the wood, but solar energy is a likely a good substitute for batteries.

Lastly, and most importantly, our team would like to thank you for giving us the incredible opportunity to work with you on such a critical mission. We appreciate your support, and we look forward to working with you again in the near future.

Sincerely,

Sam Cipriani   Aaron Crasner   Case Kittel

Steven Sloboda   A.J. Van Kainen

Earth Prototype of NASA's Venus Blimp

Sam Cipriani
Aaron Crasner
Case Kittel
Steven Sloboda
AJ Van Kainen

Engineering Interns
Team 06: Rocket
PLANET Corporation

Report submitted to
Professor Peter D. Washabaugh, NASA Representative
3028 François-Xavier Bagnoud Aerospace Building
1320 Beal Avenue
University of Michigan
Ann Arbor, MI 48109-2140

Monday, December 10, 2012

Table of Contents

List of Figures

List of Tables

FOREWORD

NASA is planning on sending an exploration blimp to Venus and needs an Earth-based prototype of the airship that meets the following criteria:

- Carries a100 gram payload
- Is speedy and maneuverable
- Uses NASA supplied materials
- Is stored in a 23 cm x 56 cm x 91 cm container
- Has a total system less than 1000-grams
- Meets all NASA requirements

Thus, we were contracted by NASA to design, build, and test an Earth prototype blimp and adapt our design to Venus.

The purpose of this report is to present our blimp design, Earth performance, and adaptations for Venus.

SUMMARY

The following is a summary of our blimp project's design, performance, and Venus adaptations.

Design

Our prototype blimp design consists of a spheroid envelope with a wooden gondola structure suspended underneath that collapses for storage. The blimp is designed to be both maneuverable and fast. The airship uses an Arduino microcontroller, has vectored thrust, and has a system mass of 794 grams which is under the 1000-gram limit. Our blimp design, shown in Figure 1 on page 3, meets all of NASA's requirements.

Performance

Upon completion, we subjected our Earth prototype blimp to several tests to determine performance. We conducted a speed test which involved flying in a straight line to a set distance, and then returning to the start-finish line. Our average speed test time was 25 seconds. A reconnaissance mission, requiring us to maneuver an obstacle course, and capture a clear image of a designated target was also flown. Our average reconnaissance time was 63 seconds. Additionally, we determined that the blimp is capable of staying aloft for 102 minutes. The performance of the blimp in these tests demonstrates its superior ability to meet NASA's needs.

Venus Adaptations

The geometric scale factor, S_G, for our Venus blimp is 0.26. The NASA polymeric membrane will not survive Venus's harsh atmospheric conditions so we recommend the use of Polybenzoxazole (PBO) for the envelope material on the Venus blimp [1]. The wood and other materials used to construct the Earth prototype will not withstand the conditions on Venus either. We plan to make use of solar cells to recharge the batteries.

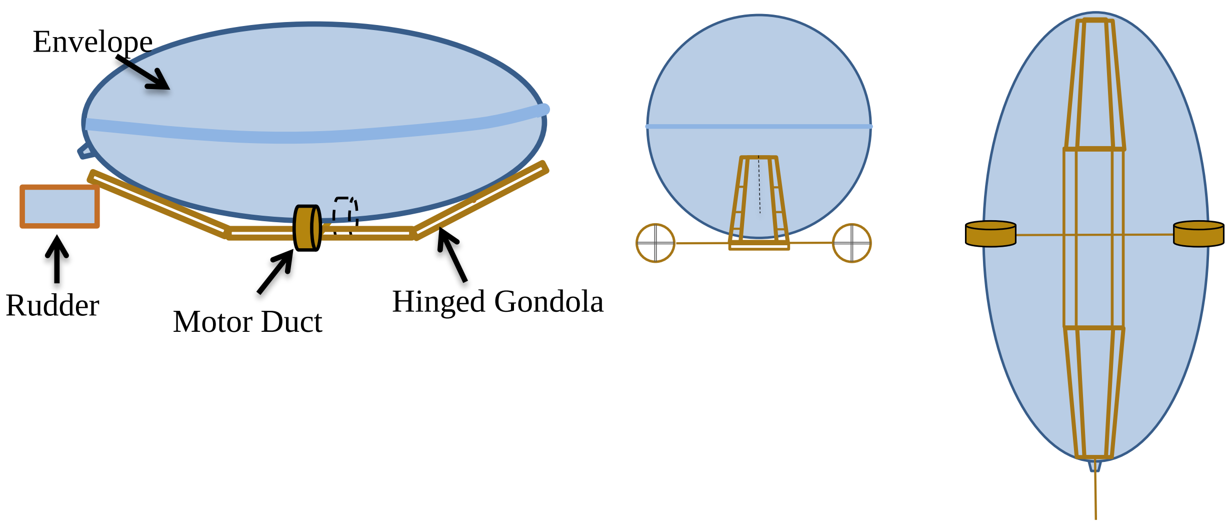
DETAILS

The subsequent text outlines the following: a detailed account of our blimp's design, the mass budget, the earth prototype blimp's performance, and the necessary adaptations and recommendations for Venus.

Blimp Design Overview

The prototype blimp design consists of a 2:1 fineness ratio (length to width), prolate spheroid, polymeric membrane envelope that is suspended by shroud lines above a wooden support structure as seen in Figure 1 below. Our blimp's shape allows for a compromise between maneuverability and speed. The blimp's support structure is hinged and features a removable rudder, allowing the blimp to be stored in the required enclosure. Our airship's envelope is pressurized with helium gas. The blimp is powered by two gondola-mounted propeller and motor systems with the ability to produce 180 degree vertically vectored thrust and also differential thrust. Additionally our blimp incorporates an Arduino microcontroller to simplify the user interface of the controls. Figure 1 below illustrates our blimp's design. The blimp has a system mass that totals to 794g out of the 1000 gram maximum, as seen in Table 1 on page 8.

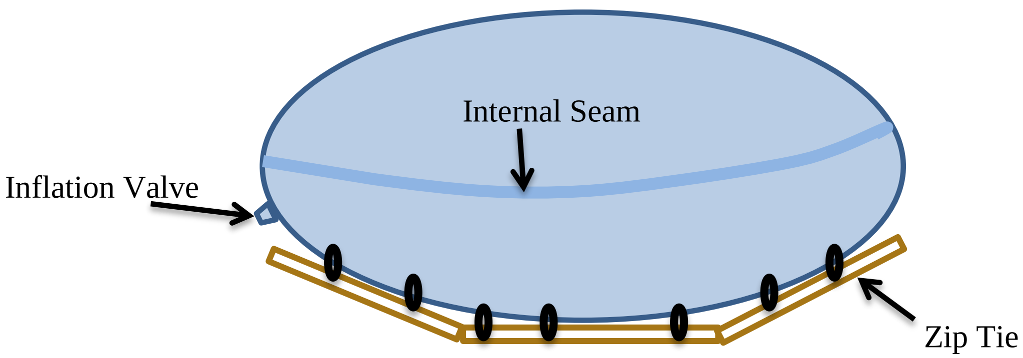
Figure 1: Blimp Design from Side, Front, and Bottom



Envelope Design: The envelope design for our blimp is approximately a 2:1 fineness ratio prolate spheroid as. Our envelope is an approximation of an ellipsoid, having a length of 2.06m (four times the radius) and a width of 0.82m (twice the radius).This design sufficiently maximizes speed without sacrificing maneuverability. The envelope is constructed out of five panels of NASA's supplied polymeric membrane which are all hot-jaw sealed together. Four of these panels are of equal size, which provides the majority of the 3-D shape for the envelope. An inflation valve is located near the bottom, and at the aft of the airship. These design features are shown in Figure 2 on page 4.

Attachment: An additional smaller panel has membrane loops that are hot-jaw sealed in between the envelope layers. Balsa wood rods are then threaded through this "pocket" and connected to the gondola via zip ties as shown in Figure 2 below. This attachment method reduces the total drag on the blimp system because the area in between the gondola and envelope is greatly reduced. Internal seams are also utilized, which further decreases drag on the blimp allowing the blimp to fly efficiently through the air.

Figure 2: Envelope Shape and Zip Tie Attachments



Envelope Lift: When inflated with Helium, the envelope provides sufficient lift for the blimp system. The envelope is designed to make the blimp slightly positively buoyant, and have the capability of adding ballast mass to achieve neutral buoyancy. Because the addition of extra ballast mass to the system greatly reduces speed, we strived to be flyable with as little extra mass as possible. In order to accomplish this, we calculated an approximate envelope radius needed to satisfy the net lift equation at neutral buoyancy. These calculations are outlined below.

Approximating our envelope's shape to be a cylinder that is twice as tall as its diameter, we set up an equation to solve for its volume:

$$V_{envelope} = (\pi r^2)h \qquad \text{[Eq. 1]}$$

since the height will be twice the length of the diameter it is 4 times the radius, so we plug in 4r for h and get the full volume:

$$V_{envelope} = 4\pi r^3 \qquad \text{[Eq. 2]}$$

Now, approximating the same cylinder, we set up the equation to solve for the envelope's surface area:

$$A_{envelope} = 2\pi rh + 2\pi r^2 \qquad \text{[Eq. 3]}$$

plugging in 4r again for h, we get:

$$A_{envelope} = 2\pi r(4r) + 2\pi r^2 = 10\pi r^2 \quad \text{[Eq. 4]}$$

Setting up the net lift equation by incorporating each essential component we get:

$$L_{net} = (\rho_{air} - \rho_{He})(g)(V_{envelope}) - (\sigma_{membrane})(A_{membrane}) - (M_{system})(g) \quad \text{[Eq. 5]}$$

by plugging in all of the values and making net lift zero, we have:

$$0 = (1.225\text{-}0.169)(9.8)(4\pi r^3) - (0.01725)(10\pi r^2) - (0.794)(9.8) \quad \text{[Eq. 6]}$$

and finally, we solve to obtain an approximate value for our envelope's radius:

$$\mathbf{r = 0.515\ m}$$

Gondola and Structures Design: The gondola support system holds the electronics, payload, mechanisms, and propulsion. The main outline of the gondola is composed of six trusses (three on each side of the gondola) of equal length. These trusses are comprised of basswood chords with balsa wood cross webbing. On each the left and right side of our structure, there are three trusses connected via hinges to one another. This design allows the entire structure to be folded up, so it is able to fit in the required container measuring 23 cm x 56 cm x 91 cm. Before flight, small wood pieces are placed in between the trusses to establish a maximum angle of hinge movement. These wooden stops prevent the gondola structure from becoming completely flat during flight, allowing the structure to fit the contour of the envelope more closely.

The gondola is held together in the middle by several balsa wood beams and smaller additional trusses composed of both balsa and basswood. This ensures that the gondola system is sturdy enough to bear the desired load without substantial deformation. A support beam mounted to the center of the gondola structure houses the Arduino microcontroller, receiver, and tilt mechanism. The dowel rod connecting the motors runs along this center support beam. These components have been arranged to make the center of mass in the center of the gondola.

Toward the rear of the gondola is the structure that houses the turn mechanism. This structure consists of small balsa wood pieces arranged in a rectangular shape. This rectangular shape allows for the turn mechanism to fit tightly, and is connected to the end of the gondola via hinge. Our fabricated gondola structure is shown below in Figure 3 (un-folded for flight with a duct) and in Figure 4 (folded for storage).

Figure 3: Unfolded Gondola

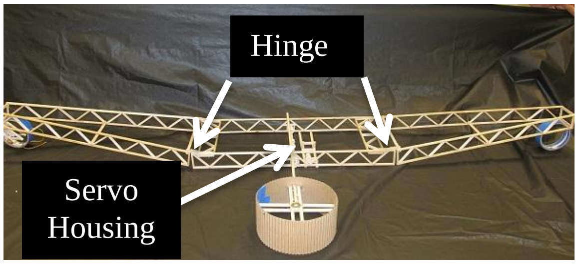
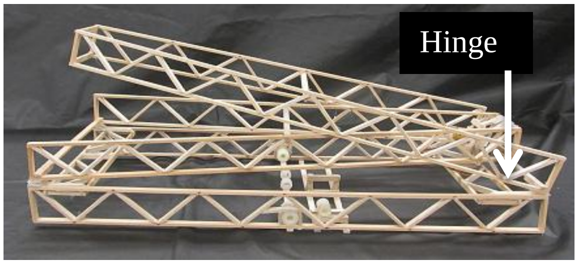


Figure 4: Folded Gondola



Mechanisms and Propulsion Design: Our blimp is propelled by two ducted propellers, driven by motors, one on each side of the craft. There are two main mechanisms on our blimp design: the motor tilt angle mechanism and the rudder angle mechanism. The tilt mechanism is what gives the blimp 180° direct vectored thrust, or the ability to change the direction of the propelling force from straight up to straight down. The rudder angle mechanism contributes to the blimp's turning ability, changing the angle of the rudder relative to an axis drawn from front to back through the blimp's center. The maximum angle from the center axis that the rudder can achieve on each side is 90°. Differential thrust, running the left and right motors at different speeds, contributes to the blimp's ability to turn as well by changing the speed at which each side of the blimp moves. The powerful combination of direct vectored thrust, differential thrust, and a rudder maximizes the blimp's maneuverability as it allows for the craft to quickly change its direction of movement.

Motor tilt mechanism: The motor tilt mechanism consists of a long dowel that supports a ducted propeller and motor on each end, two identical, small pulleys, a belt, a servomechanism, and a small support rod. The servomechanism and the long dowel are mounted near the center of the gondola. The servomechanism has a pulley, a belt, and a support rod attached to it. The support rod's function is to hold the pulley and servo in place, allowing the servo to turn about a stable, fixed axis that is not affected by the tension of the belt on the pulley. The belt connects a pulley on the long dowel and the pulley on the servomechanism, translating the 180° rotation of the servo to the long dowel and the ducted propeller/motor. This configuration is shown in Figure 5 below. The dowel is cut in half and its two halves are attached via an aluminum collar so that the dowel can be disassembled and stored in the required enclosure.

Rudder angle mechanism: The rudder angle mechanism consists of a servomechanism, a pulley, a dowel, a large nylon bushing, a balsa wood support triangle, and a membrane covered, wooden frame rudder. The servomechanism is mounted on the underside of the rear of the gondola. The servo has a pulley attached to it and faces downward. The pulley has set screws and a hollow inside that allows the dowel to fit inside and be tightened in or removed as needed. The dowel is glued to end of the rudder and has a freely rotating bushing around it near the end opposite of the servo. The bushing is glued to the support triangle which is then zip-tied to the gondola frame. The rudder mechanism is shown in Figure 6 below. This structure allows the servomechanism to rotate the rudder in an 180° range of motion (90° from each side of the center line) while still providing structural support.

Figure 5: Tilt Mechanism

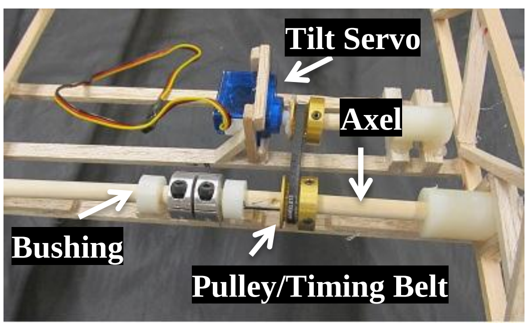
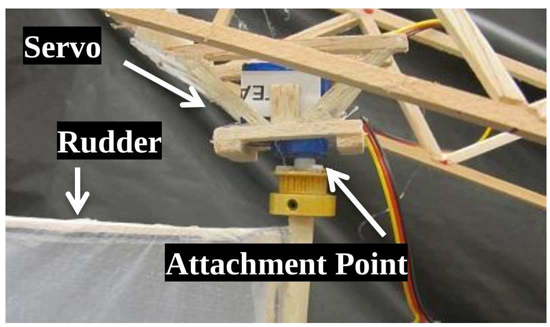


Figure 6: Rudder Angle Mechanism



Propulsion: There is a cardboard duct, a plastic propeller, and an electrical motor mounted on the end of the long dowel on either side of the blimp. Each duct features a cross-shaped, balsa wood support structure inside that houses the motor to which the propeller is attached. This is shown in Figure 7 on page 7. The duct is composed of a 7.62 cm (3 in) wide strip of cardboard formed into a cylinder with a diameter of 16.51 cm (6.5 inches), and is sealed with adhesive tape. The duct diameter, 16.51 cm (6.5 inches), is slightly larger than that of the propeller, 15.24 cm (6 inches), so that the propeller blades do not hit the walls, and so that the airflow is concentrated, improving propeller efficiency. The wooden support structures are made of thin beams of balsa glued to the inside edge of the duct and to one side of a washer, which is aligned with the axis of the cylindrical duct. The purpose of the beams is to support the circular shape of the duct and to hold the washer, which is the attachment point for the motor. The motor has a threaded portion that allows it to be tightened on to the washer with a nut, holding the motor in place so that the propeller blades are positioned a small distance inside the duct. In this position, the propeller is protected by the duct and its generated airflow will be concentrated within the duct which allows for greater efficiency than a free propeller for the speeds at which the blimp will operate. The ducts can rotate 180̊ to optimize full range of motion, as shown below in Figure 8.

Figure 7: Propulsion System

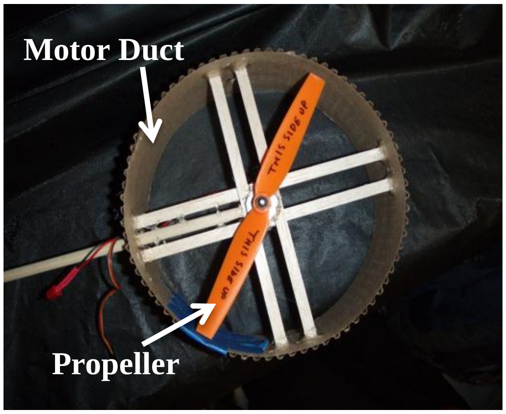
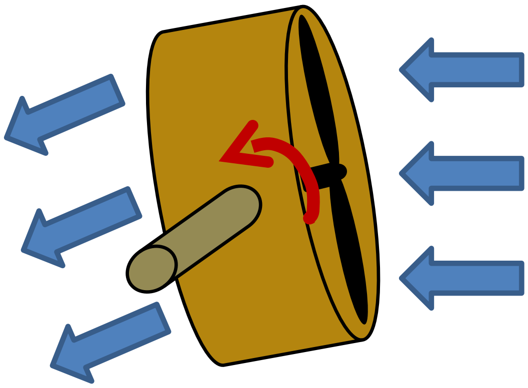


Figure 8: Ducts Rotate 180̊



Circuitry and Programming: The blimp’s circuitry is powered by two battery packs attached to each other in series. They are connected in parallel to two motors and an Arduino microcontroller. A circuit diagram of this arrangement is shown in Figure 9 on page 8.

The remote controller transmits control data to the receiver on board the blimp. The controls are laid out in the following way: the left joystick controls the total available thrust of the motors; the right joystick (vertical) controls the tilt angle of the motors; the right joystick (horizontal) controls the blimp’s direction (left and right) These controls are illustrated in Figure 10 on page 8.

Figure 9: Circuit Wired in Parallel

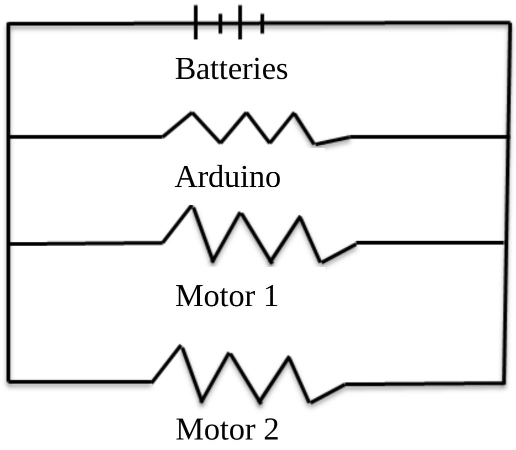
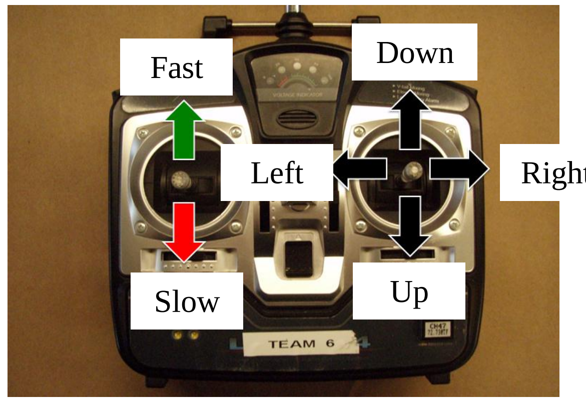


Figure 10: Controller Configuration



The receiver then pipes the control data into the Arduino. The code on board the Arduino is maps the thrust, tilt, and turning control inputs to the following mechanical outputs, and is shown in Appendix A on page A1:

1. The turning control input data is parsed to control both a rudder and the relative thrust generated by each motor. Depending on the joystick and trim tab positions, one motor turns faster (and thus generates more thrust) than the other at the same time that a rudder on the back rotates to provide even greater maneuverability. For example, if the pilot wishes to make the fastest possible right turn, the left motor is set to 100% available thrust, the right motor is at 50% available thrust, and the rudder at the back is turned all the way to the right. The reason the weak-side motor only ranges from 50% to 100% is to provide the pilot with less-sensitive (and therefore more stable) controls. The blimp is still highly maneuverable.

2. The tilt control input data is parsed to control the angle the motors themselves make against level. Depending on the joystick and trim tab positions, the program rotates the motors to face the floor (full down) or to face the ceiling (full up) and anywhere in between. The servo on board grants us 180° of rotation, which again drastically improves our blimp's maneuverability.

3. The thrust control input data is parsed to control the total thrust available to either motor. Depending on the joystick and trim tab positions, the total available thrust is swept from no thrust to the max thrust that the motors can generate.

Note that the turning control function only allows the dominating motor to achieve 100% available thrust, not 100% total thrust. The only condition under which the dominating motor in a turn achieves 100% total thrust is when the thrust control input is set to 100% as well.

System Mass Under 1000-gram Limit: The total mass of our blimp design is under the 1000-gram limit imposed by NASA as shown in Table 1 below. The mass budget is divided into several categories—Propulsion, Body, Envelope, Payload, Power, and Ballast—that define the main components of the blimp.

Table 1: System Mass Under 1000-g Limit

System	**Component**	**Mass (grams)**
Propulsion	Ducts, motors, wooden beams	99.55
Body	Gondola, electronics, mechanisms	342.00
Envelope	Envelope, attachment rods	79.00
Payload	Camera	100
Power	Batteries	98.88
Ballast	-----------	50
	Total	**794 g / 1000 g**

Earth Blimp Tests and Performance

Because we completed our blimp well before the competition date, we had ample time to test our blimp's in-flight performance. This allowed us to gather much data to assess what the main obstacles were on the course and also provided numerous opportunities for our pilot to refine his technique. Because of the advantages of early flight and the subsequent time for practicing, our blimp's performance was very respectable overall.

Speed Test: The speed test was a timed test that required flying between two lines placed twenty meters apart, turning around, and then flying back to the starting point, at which point the test time was recorded. One of the main challenges in this test was achieving stable flight, which mainly became an issue if the controls were too sensitive or if the structural design of the blimp caused it to experience asymmetric forces. Another big challenge was the turning maneuver that was required in order to return to the starting point and complete the test. This turn could be troublesome if the blimp's controls were not designed or programmed correctly. For example, propellers that are closer to the center of mass will provide less torque, and therefore a slower turn. Additionally, if the controls are not programmed such that one motor overpowers the other, then the blimp will not turn as effectively either. In general, these problems could be mitigated via design changes and the pilot's adjustments during flight.

Speed Test Performance: In the speed test, our blimp flew well, and our timed trials prove that. Our fastest time was 22 seconds, with our average time being 25 seconds. Our fastest time was the result of much practice and was the last of a series of test flights. The data from our speed test flights is shown in Table 2 on page 10.

The biggest contributing factors to our blimp's low times on the speed test were correctly programmed controls, a tail that turned to reduce drag when turning, and a pilot who figured out good flying techniques and the best way to turn. The way we designed the controls on the transmitter was also quite intuitive, thus easing the pilot's job.

Table 2: Speed Test Average Time is 25 Seconds

Trial	Time (s)
1	26
2	29
3	23
4	22
Average	**25**

Flight Speed Test: As part of our agenda, we also had to test our blimp's flight speed, which we did by flying the blimp over a distance of 20 meters and dividing the distance by the time the blimp took to traverse it. Performing several trials, we determined that our blimp's fastest speed is approximately 1.4 m/s, with an average speed of 1.2 m/s. The data from this testing is shown in Table 3 on page 10.

Table 3: Average Speed is 1.2 Meters per Second

Trial	Speed (m/s)
1	1.1
2	1.4
3	1.1
Average	**1.2**

Recon Test: The recon test was a timed test that required flying along the same course as the speed test, except with much different parameters. The blimp had to fly down the course and under a laser beam, and then over a ledge. On top of the ledge, a box was placed with a specific number of objects arranged inside. Our job was to fly close enough over this box with the camera to get clear video of the inside on a live monitor, which we had the ability to rewind. After successfully getting a clear video, the blimp had to fly back to the starting point, at which point the test time was recorded. We were then required to use this video to freeze-frame at a clear image, and then count the number of objects inside.

There were a number of challenges in this test, all of which were nontrivial obstacles to success. Firstly, the same challenges of straight flight and turning seen in the speed test remained important factors. Then, the need to fly under a laser and then over a ledge presented a new challenge in terms of our blimp's controllability. This required much more precise controls and piloting, as well as the ability to effectively climb and descend. Next, it was difficult to get the blimp close enough to the box such that a clear video could be taken. The camera had to be quite close to the box, adding yet another layer of difficulty to the course, and therefore more skillful piloting.

Additionally, there were other obstructions including an air jet that the blimp had to avoid. Lastly, the clarity of the camera's images could cause errors when counting the number of items in the box. There was a steep penalty of multiplying the test time by a factor of (1 + error) for miscounting. For example, a count that is off by one would yield a time-doubling penalty; a count that is off by two would yield a time-tripling penalty, and so on. Overall, the combination of this and the numerous other obstacles made for a very difficult test, one that required a great deal of control over the blimp's flight.

Recon Test Performance: Despite the numerous challenges listed above, our blimp performed very well in our recon tests. Because we had sufficient time to practice flying, we handled the obstacles with relative ease. After running several trials, our fastest time was 47 seconds, with our average time being slightly higher at 63 seconds. The data from our speed test flights is shown below in Table 4 on page 11.

Table 4: Recon Test Average Time is 63 Seconds

Trial	**Time (s)**
1	71
2	69
3	47
4	65
Average Time (s)	**63**

There are quite a few factors that go into the timing of the recon test. Air gusts in the room, as well as small pilot errors, in addition to the many aforementioned obstacles, can have compounding effects. Thus, it is certainly possible that our slower times were the results of these combined problems. Therefore, we believe that our fastest time of 47 seconds is, in fact, repeatable. It was only possible due to the amount of practicing that our pilot did, and subsequent practices can only serve to improve the chances of successes like that one. Additionally, we were able to successfully get a clear picture of the items in the box as shown below in Figure 11. A test flight is shown in Figure 12.

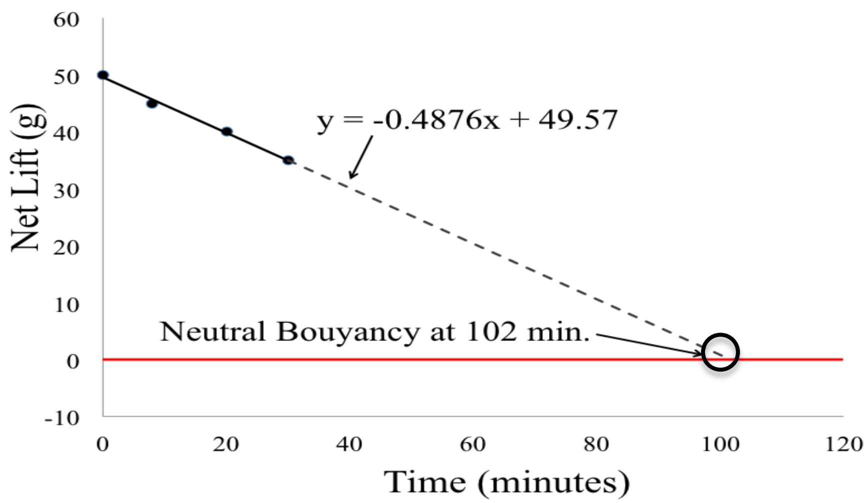
Figure 11: Blimp Provides Clear Image of Box

Figure 12: Blimp in Test Flight

Time Aloft: To determine how long our blimp could feasibly stay aloft, we had to perform a helium diffusion test. This basically involves leaving our blimp inflated with ballast mass attached for a long time and, each time the blimp achieves negative lift, periodically removing ballast to return to neutral buoyancy. This steady decrease in the ballast mass that provides neutral buoyancy is interpreted as the rate, in grams/minute, at which helium diffuses through the porous envelope material.

Time Aloft Test Performance: We found, after completing testing, that our blimp can stay aloft for approximately 102 minutes. Following the above testing model for about 30 minutes, we were able to gather data that was quite linear. This allowed us to extrapolate the data to the point at which our blimp's net lift would equal zero, making it neutrally buoyant. This is the point at which our blimp is considered to no longer be capable of staying aloft. The graph displaying this data is shown in Figure 13 on page 12.

Figure 13: Blimp Stays Aloft for 102 Minutes



Venus Deployment

NASA has tasked us with deploying our blimp in the harsh Venusian atmosphere. Venus has extremely hot temperatures (ranging from 740-770 Kelvin), a high atmospheric density (50x that of Earth), and sulfuric acid clouds [2]. The following section details our blimp's scaling calculations and material adaptations to Venus. The atmospheres of Earth and Venus are compared below in Table 5.

Table 5: Venus Atmosphere: Hot, Dense, Acidic

Condition	Earth	Venus
Avg. Temperature Range	285-290	735-770
Surface Density (kg/m^3)	1.225	65.48
Other Conditions	N/A	Acidic

Scaling Calculations: Because the planet Venus has a very high atmospheric pressure compared to that of Earth's, the envelope size for our blimp will be considerably smaller. We decided to scale the blimp such that the net lift would be zero, making the blimp neutrally buoyant. Shown below are the calculations necessary to determine the geometric factor for scaling our blimp to Venus.

First we need the constants shown in Equations 7:

$g_{Venus} = 8.87$ m/s2 $\qquad \rho_{AtmVenus} = 65.48 \qquad \rho_{HeVenus} = 6.03$ [Eq. 7]

Setting up the net lift equation by incorporating each essential component we get:

$$L_{net} = (\rho_{Atm} - \rho_{He})(g)(V_{envelope}) - (\sigma_{membrane})(A_{membrane}) - (M_{system})(g) \quad \text{[Eq. 8]}$$

by plugging in all of the values and making net lift zero, we have:

$$0 = (65.48–6.032)(8.87)(4\pi)(S_G*0.384)^3–(0.01725\pi\ (S_G*0.384)^2–(0.739)(8.87) \quad \text{[Eq. 9]}$$

Finally, solving for the geometric scale factor, we get:

$\mathbf{S_G = 0.26}$

The geometric scale factor for our blimp is 0.26. The new length and width can be found by multiplying our Earth-based dimensions by the scale factor. In doing so, we find a length of 0.4004-m and a width of 0.2002-m required to achieve neutral buoyancy on the planet Venus.

New Materials: Polybenzoxazole is a lightweight, space membrane. Independent studies have shown that PBO is flame resistant, and can withstand temperatures up to 923 Kelvin without suffering structural damage [1]. 923 Kelvin is greater than Venus' highest temperature, so a blimp envelope of this material would survive the intense heat of the planet. Polybenzoxazole has a high hydrolytic stability, meaning it is highly resistant to organic solvents. This will allow the blimp's envelope to survive the acidic atmospheric conditions for as long as possible. After reviewing the characteristics of Polybenzoxazole, we recommend this new membrane material for the construction the blimp that will be deployed on Venus.

Additionally, the wood and other materials used to construct the Earth prototype will not withstand the conditions on Venus, so we recommend that heat and corrosion resistant materials be used for the construction of the Venus blimp. We also recommend the use of solar cells to recharge the blimp's batteries.

CONCLUSION

In conclusion, our Earth prototype blimp design employs a 2:1 spheroid envelope with a wooden gondola structure that is hinged for collapsible storage. The blimp is strategically designed to be both quick and pilotable during flight. The airship is powered by vectored thrust, which is regulated by an Arduino microcontroller. Our blimp design has a total system mass within the 1000-gram limit and adheres to all of NASA's requirements.

A reconnaissance mission and speed tests were flown to determine the performance of our Earth prototype blimp. The times of each trial were respectable and consistent. The maximum speed was very reputable, especially for the fact that our blimp can turn on a dime. By calculating the diffusion rate, we deem that the blimp system will stay aloft for the duration of the desired mission. The performance of our blimp as a whole was ample proving that our design was satisfactory to NASA's task.

Because of Venus' harsh atmospheric conditions, we will be able to scale the blimp with a geometric scale factor of 0.26. We recommend the use of Polybenzoxazole [1] as a new envelope material, as the proposed NASA polymeric membrane will not survive the alien planet. Other materials, such as the wood used to fabricate the gondola, will not be suitable for the conditions on Venus either. Solar cells will be mounted to the craft into to recharge the batteries.

REFERENCES

1. Yavrouian, A., Yen, S. P. S., Plett, G., & Weissman, N. (n.d.). High Temperature Materials for Venus Balloon Envelopes. California Institute of Technology, Pasadena, CA. <http://www2.jpl.nasa.gov/adv_tech/ballutes/Blut_ppr/tempmatl.pdf>

2. <http://csep10.phys.utk.edu/astr161/lect/venus/atmosphere.html>

APPENDICES

Appendix A: Arduino Code

```
#include <Servo.h>


/********************************************************/
/*                Initialize Variables                  */
/********************************************************/

double motor1;                          //Rigt Side
double motor2;                          //Left Side

const double THR_MAX = 2047;            //Max chanel 3 value - thrust
const double THR_MIN = 871;             //Min chanel 3 value

const double TIL_MAX = 2087;            //Max chanel 2 value - tilt
const double TIL_MIN = 900;             //Min chanel 2 value

const double TUR_MAX = 2078;            //Max chanel 1 value - turn
const double TUR_MIN = 905;             //Min chanel 1 value


const double TIL_MID = (TIL_MAX-TIL_MIN)/2;      //Mid tilt angle value
const double TUR_MID = (TUR_MAX-TUR_MIN)/2;      //Mid turn value

const double EX_HIGH = (TUR_MAX-TUR_MIN)-25;
const double EX_LOW = 15;

Servo tiltServo;
Servo turnServo;
Servo m1;                       //m1 on pin 11
Servo m2;                       //m2 on pin 3
double thrustm1=0;
double thrustm2=0;




/********************************************************/
/*                Initialize Motors                     */
/********************************************************/
```

```
void init_motors()
{
	//sweeping input to the motors initializes them
	int a;
	for (a = THR_MIN; a<THR_MAX; a+=10)
	{
		m1.writeMicroseconds(a);

		delay(1);
	}
	for (a = THR_MAX; a>THR_MIN; a-=10)
	{
		m1.writeMicroseconds(a);
		delay(1);
	}
	for (a = THR_MIN; a<THR_MAX; a+=10)
	{
		m2.writeMicroseconds(a);
		delay(1);
	}
	for (a = THR_MAX; a>THR_MIN; a-=10)
	{
		m2.writeMicroseconds(a);
		delay(1);
	}
}


/*********************************************************/
/*		Startup Sequence				*/
/*********************************************************/

void startup()
{
	//This mostly all looks cool...
	//Turn Servo
	turnServo.write(0);
	delay(1000);
	turnServo.write(180);
	delay(1000);
	turnServo.write(90);

	//Tilt Servo
	tiltServo.write(0);
	delay(1000);
```

```
        tiltServo.write(180);
        delay(1000);
        tiltServo.write(90);




        int a;
        for (a = THR_MIN; a<THR_MAX; a+=10)
        {
                m1.writeMicroseconds(a);
                delay(5);
        }
        delay(100);
        for (a = THR_MAX; a>THR_MIN; a-=10)
        {
                m1.writeMicroseconds(a);
                delay(5);
        }
        delay(300);
        for (a = THR_MIN; a<THR_MAX; a+=10)
        {
                m2.writeMicroseconds(a);
                delay(5);
        }
        delay(100);
        for (a = THR_MAX; a>THR_MIN; a-=10)
        {
                m2.writeMicroseconds(a);
                delay(5);
        }
        delay(300);
}

void setup()
{
/*********************************************************/
/*              Setup Inputs/Outputs                     */
/*********************************************************/

        tiltServo.attach(2);            //Tilt servo on D2
        turnServo.attach(5);            //Turn servo on D5

        pinMode(10,INPUT);              //Channel 1 Input - Turn
        pinMode(9, INPUT);              //Channel 2 Input - Tilt
```

```
        pinMode(8, INPUT);              //Channel 3 Input - Thrust
        pinMode(3, OUTPUT);             //Motor 2
        pinMode(11, OUTPUT);            //Motor 1
        m2.attach(3);
        m1.attach(11);

        //Sweep the controls for a test
        init_motors();
        startup();

        Serial.begin(9600);
}

void loop()
{

/*******************************************************/
/*              TURN CONTROLS                          */
/*******************************************************/

        double turnval=pulseIn(10,HIGH);        //Get input from receiver ch1

        //turnval mapped to servo range
        double motorturn=map(turnval,TUR_MIN,TUR_MAX,0,180);

        turnval = turnval-TUR_MIN;              //0<turnval<TUR_MAX-TUR_MIN
        double rat = 100*abs((TUR_MID-turnval)/TUR_MID);        //here 0<rat<100
        rat = map(rat,0,100,0,50);              //rat gets mapped from 0 to 50
        turnServo.write(motorturn);

        if ((turnval>TUR_MID) && (turnval<EX_HIGH))     //Normal right turn
        {
                motor1=100-rat;                 //min value is 50, max is 99.999
                motor2=100;
        }
        else if((turnval<TUR_MID) && (turnval>EX_LOW))  //Normal left Turn
        {
                motor2=100-rat;                         //same min/max as ^
                motor1=100;
        }
        else if(turnval==TUR_MID)                       //Exactly straight on
        {
                motor1=100;
                motor2=100;
        }
```

```
	else if(turnval>=EX_HIGH)				//Extreme right turn
	{
		motor1=0;
		motor2=100;
	}
	else if(turnval<=EX_LOW)				//Extreme left turn
	{
		motor1=100;
		motor2=0;
	}


/*******************************************************/
/*		TILT CONTROLS				*/
/*******************************************************/

	double tiltval = pulseIn(9,HIGH);			//Input from ch2
	double motortilt = map(tiltval,TIL_MIN,TIL_MAX,180,0);	//mapped to servo
	tiltServo.write(motortilt);


/*******************************************************/
/*		THRUST CONTROLS				*/
/*******************************************************/

	double thrustval = pulseIn(8,HIGH); //Get input from receiver - ch3
	double thrustrat = map(thrustval,THR_MIN,THR_MAX,0,100);

	double pm1 = (thrustrat*motor1)/100;		//percentage thrust to each motor
	double pm2 = (thrustrat*motor2)/100;			//ranges from 0 to 100

	thrustm1 = map(pm1,0,100,THR_MIN,THR_MAX);
	thrustm2 = map(pm2,0,100,THR_MIN,THR_MAX);

	m2.writeMicroseconds(thrustm2);			//Writes thrust to m2
	m1.writeMicroseconds(thrustm1);			//Writes thrust to m1


}
```

Honor Code: We have neither given nor received aid on this assignment.

Samuel Cipriani, Aaron Crasner, Case Kittel, Steven Sloboda, AJ Van Kainen