WSGC Elijah High Altitude Payload 2012 Summer Team Final Report

August 17th, 2012

Brock Boldus¹, Patrick Comiskey¹, Latisha Jones¹, Kaitlyn Mauk¹, Ben Peterson¹, & Matthew Weichart¹

¹Milwaukee School of Engineering

Advisor: Dr. William Farrow²

²Assistant Professor Department of Mechanical Engineering Milwaukee School of Engineering

Abstract

The 2012 Wisconsin Space Grant Consortium Elijah High Altitude Payload Team explored four different experiments this summer. We investigated the motion of the payload during flight, designed several systems for two cameras, utilized a magnetometer, and inspected the use of thermistors. Each experiment had numerous subcategories that resulted in more work than we had time for. Because of these unforeseen consequences, our team had to urgently attempt to make the deadline of the launch date. Through this haste, sections were overlooked; therefore we lost the cameras during flight and were not able to gather data from the magnetometer and thermistor experiments. However, this experience demonstrated a lot about the engineering process and working as a team. The internship taught us how to become better engineers through our mistakes and was an extremely worthwhile experience.

Introduction

The Wisconsin Space Grant Consortium (WSGC) is a group dedicated in getting students into science and engineering in the broad field of aerospace. We had the opportunity to work with the WSGC on the Elijah high altitude balloon project, which sends a weather balloon high into the atmosphere in order to conduct science and engineering experiments. The balloon takes the payload close to the edge of the earth's atmosphere. The balloon then pops and the payload descends on a parachute.

The only constraints for the project are those imposed by the Federal Aviation Association (FAA), the governing body over US Airspace. In accordance with FAA regulations for balloons,

¹ Financial support from Wisconsin Space Grant Consortium, special thanks to all those mentioned in "Acknowledgements" section

the payload must weigh only six pounds or less and the pressure on any square inch of the outer surface of the payload must not exceed 3/16 psi (FAA Regulations for Kites/Balloons, 1999).

This year, the team is elected to conduct experiments in stabilizing the payload, strength of the earth's magnetic field, the amount of radiation put off by the sun by measuring the temperature, and taking and comparing pictures with and without infrared filters. The stabilization of the payload includes two sub projects, controlling the tendency to tumble using a tail extension and controlling the spinning using CO_2 canisters.

Box-Tail/Drag-Kite

Early on one of the team's interests was to stabilize the payload during the flight. One of the projects that were born from this desire was the Box-Tail/Drag-Kite. The idea behind the Box-Tail/Drag-Kite is to extend the momentum arm of the whole payload, therefore requiring a bigger and stronger force to disturb the payload. It also adds a restoring force. As one side of the tail is pushed by wind, it will expose the opposite side to the same force, causing it to briefly go into periodic motion until it reaches a more stable orientation.

The first version of the Box-Tail/Drag-Kite was based on the casing from the previous year's casing and eight foot long flexible rods that extended behind the casing. Attached to the end of the rods was a large parachute that would be used to retard all of the forces acting on the payload and resultantly make the payload more inherently stable. The force of air acting on the drag kite would resist other forces acting on the payload, such as winds aloft and the length of the tail behind the payload would increase the moment of inertia making the whole thing harder to perturb.

However, there were major drawbacks to this plan. The eight foot rods would make the whole thing unwieldy and awkward to move in addition to just being excessively large so the rods were reduced from eight feet long to a more manageable four feet long. The parachute-like end also presented a significant problem as it could cause the whole payload to invert once it started to descend back towards Earth.

In order to deal with the issue of it flipping over, the parachute design was changed for a more traditional box-shape tail. The design would not resist the payload's upward movement like the previous design did, however as far as stabilization for side to side movement; it did just as well or better than the previous design. It was also significantly less likely to invert the whole payload than the previous design.

The rods for the Box-Tail/Drag-Kite were quickly decided to be made from fishing rods, which have proven to be able to withstand a lot of stress without breaking as well as being lightweight.

Fishing rods are also readily available in various lengths and actions, where the rod bends the most, so there were plenty of options for us to chose from. The team researched for fishing rod manufacturing companies based here in Wisconsin and came across St. Croix Rods, based in northern Wisconsin. The team got in contact with their director of engineering and was able to receive an in-kind donation of several blanks, rejected carbon-fiber rods that were not completely built, but the rods still offered the team everything the team was looking for the Box-Tail/Drag-Kite.

The carbon fiber rods were cut to the desired size using a dremel cutting tool and then fitted with nylon bolts so they could be attached to the rest of the payload more easily. The casing and the connection nexus that would be mounted on the bottom hemisphere of the payload were set up so that the rods would be ninety degree between each other and twenty degrees from the central axis.

The sides of the Box-Tail/Drag-Kite were decided to be made out of rip-stop nylon, the most popular material for parachutes since the Second World War and most recommended after kevlar and other similar materials currently used to make parachutes for the military. The sides of the Box-Tail/Drag-Kite were four inches in width and twenty-three and twenty one hundredths in length and set up to accommodate the angle of the rods. The sides were made out of rip-stop nylon that was donated by North Sails, a manufacturer of sails for sailboats operating out of Milwaukee and Chicago. The nylon was cut from a template and then pieces were then sewn together to make the sides of the Box-Tail/Drag-Kite with loops that would go over the rods. The nylon sides were then attached to the rods using Tear-Aid, a tear repair patch kit that bound very well to both the nylon and the carbon fiber rods.

Shape Determination and Connection Junction

In the pursuit for payload stabilization during flight, the shape of the payload is an integral consideration. There were many different hazards that the payload would experience that needed to be addressed. For example, crosswinds, drag force, cost, weight, center of mass, overall size, space for equipment, and ease of building are some of the parameters that had to be factored into the design.

The basic shapes that we persisted upon finding the best out of the bunch were the cube, sphere, cone, and cylinder. Because the payload experiences many different perils at different altitudes, we needed to gather information at various altitudes. Using data from *The Engineering Toolbox* (The Engineering Toolbox, 2012) and the *United States Centennial of Flight Commission* (U.S. Centennial of Flight Commission, 1999); we received crosswind speed with corresponding altitude, density, temperature, and an ascent rate. Compiling and processing the information

seemed quite a large task to undertake by hand; therefore, a *MATLAB* code was written to do the calculations.

After working some time on the code, for every data point of information found online, we had our different parameters on the four different shapes. Figure 1 below, shows the *Drag Force vs. Altitude* plot that the code displayed:



Figure 1: Drag Force vs. Altitude

Out of the four shapes, the sphere performed the best in terms of drag force. For each different parameter, something similar to this figure was communicated. After careful consideration, the sphere turned out to be the best shape for our application. We finalized the choice, and put forth a motion to start to build the spherical payload.

There are several different ways to connect the box tail to the payload, including right onto the outer sphere, a connection point inside the sphere, or various other methods. In regards to joining to the outside, physically creating the necessary geometry for a correct box kite would be very difficult. Lining everything up would be near impossible and separately connecting each carbon fiber rod adds unnecessary hassle. Therefore, we decided that it would be best to make a connection junction inside the payload.

Because of the complex geometry needed for the box kite and weight considerations, rapid prototyping our nexus seemed to be the best option. After furthering the project in other aspects, we decided that this connection junction could harbor the YEI 3-Space Sensor, along with the

camera mount carbon fiber rods, and the box kite carbon fiber rods. With that in our minds, we set out to make a structurally rigid, rapid prototyped part. After six different re-designs, the final variation that was built is seen in Figure 2 and 3:



Figure 2: Top view of the nexus with space for the YEI sensor and four spot to connect to the mounting board



Figure 3: Bottom view with the four larger holes at a 20 degree angle for the box kite and the two smaller ones for the camera mount

After retrieving the payload from launch, we saw that the connection junction performed flawlessly, and did everything that is was intended to do.

Magnetometer

At the beginning of this project, it was determined that the earth's magnetic field in the X, Y, and Z directions were to be experimentally found in relationship to altitude; these values would be measured in Gauss. Not much information could be found relating earth's magnetic field to altitude, except for over 119,000 meters -- most magnetic surveys occur under 300 meters due to

difficulties in acquiring data (Bayot, 2005). A goal from this side of the project was to investigate the relationship for altitudes approximately zero to ninety kilometers up. There was also the possibility of incorporating a Geiger Tube into the experiment to measure the radiation in the atmosphere in relationship to altitude as well, but that experiment was dropped due to the time required to process and ship that electronic component; it may not have arrived in time for the flight.

In order to measure the earth's magnetic field, a magnetometer was needed. There are different types of magnetometers that can come in 1, 2, or 3 axes. Different magnetometers are meant for different situations. Most basically, there are fluxgate, proton processors, and overhauser magnetometer (Bayot, 2005). The latter two were either meant for stable, ground situations or were too large for the purposes of this experiment, and certain fluxgate magnetometers are smaller in size and do not need a ground base, like particular digital and analog ones. Then, the major companies that sold fluxgate magnetometers whose products were most closely examined were Applied Physics, Stefan Mayer, and Bartington. During this time of research, a sheet with all of the projects' combined potential weight and budget was created. Due to these calculations, cost and weight were big factors to consider when trying to choose a magnetometer. The final magnetometer was chosen to be Magnetometer 113D from Applied Physics. It was a digital fluxgate magnetometer, weighing in at approximately 25 grams. Its minimum power input was +4.9V, which was compatible with the Arduino's capabilities. It came with six inch flying leads and could give off both ASCII and binary over the serial communication RS232 and TTL (Applied Physics Sytems, 2012). The earth's magnetic field averages about 50,000nT, or 0.5 Gauss, and it ranges from about 25,000nT at the Equator to 70,000nT around the poles (British Geological Survey, 2012). So since Model 113D had a range of $\pm 60,000$ nT with a 2nT resolution, it would be very sensitive to the altitude's magnetic differences (Applied Physics Systems, 2012).

Once the magnetometer came in the mail, a few wires had fallen off of the PC board. After they were resoldered, as the magnetometer was being dealt with, they continued to fall off randomly until the system was properly mounted with styrofoam. The system was not mounted with metal in a hopeful attempt to not mess up the earth's magnetic field readings. The wiring came prepared for the serial communication RS232, and the Arduino UNO was programmed with RS232 taken into consideration. After many struggles with trying to get the magnetometer and arduino to prove that they were talking through the computer, it was eventually found that the serial port was incorrect, and the wires were re-soldered to incorporate TTL communication, the real serial that the arduino communicates through. After determining that, it was a very fortunate situation that the board and magnetometer were not damaged through the testing because of the different voltages associated with these ports. By then changing some of the coding, the correct number of binary sets of numbers were showing up through the com port. Figure 4 shows how the code was written to only store the magnetic field in the X, Y, and Z directions in binary and not the rest of the nine bytes of information the magnetometer was picking up.

```
//if count is < 80 then continue gathering
  if (count/6 < targetDataReads) {
   Serial.println("Grabbing magnetometer data");
    //transmit "Osd<cr>"
   mySerial.write("Osd");
   mySerial.write(13);
    //wait for response ...
   int byteReceived = 1;
   while (byteReceived < 15) { //while still waiting for byte 15 to be received
     if(mySerial.available()){
       1/1
               SOT - 10
                              1 byte
               MX -
       1/2,3
                       0x#### 2 bytes
                                         store in EEPROM
       //4,5 MY -
                       0x#### 2 bytes
                                         store in EEPROM
       1/6,7
              MZ -
                      0x#### 2 bytes store in EEPROM
       1/8,9
              TEMP - 0x#### 2 bytes
       //10,11 ANA1 - 0x#### 2 bytes
       //12 Status 80
                              1 byte
       //13
             CS
                      0x##
                              1 byte
                      0x7FFF 2 bytes
       //14,15 EOT
       if (byteReceived > 1 && byteReceived < 8) {
         byte red = mySerial.read();
11
           Serial.println("good data");
11
           Serial.println(red,HEX);
         EEPROM.write(startEEPROMMagnetometer + count, red);
         count++:
       }
       byteReceived++;
     3
   }
```

Figure 4. Storage of only needed data on Arduino's EEPROM.

It was determined earlier on that the YEI accelerometer would need its own arduino memory if it was going to work through communication with the carbon dioxide canisters to the gyros too. Therefore, it was given its own arduino, and the thermistors and magnetometer were to split the memory on the other arduino. Given the amount of space available, binary was set to be used because if the data was recorded in ASCII, the number of readings per three hour flight would be significantly dropped. As it was, approximately only 72 readings could be taken during the flight in order not to write over itself in the memory, by just gathering the X, Y, and Z directions in Gauss. If there had been more time, extra EEPROM memory would have been purchased.

Some of the major programming victories include getting accurate data, setting a switch, and merging the magnetometer and thermistor codes. A switch needed to be set so that the arduino would give the computer its data without starting over and erasing the previous data. There was a button on the breadboard at first, and it was programmed so that when it was pushed down upon start up, the arduino would know to record data only. When it was not pushed down upon start up, the arduino would instantly just give the computer its last data. An LED was incorporated that would light up to signify when the system was ready to record data. Wiring became quite congested, as seen in Figure 5, so the program was rewritten to get rid of the button and change it to a wire that needed to be put in or pulled out, telling the arduino to gather data or give data. Because the battery supplies were estimated to only last three to four hours, and the flight was

approximately three hours long, another switch was made for the battery packs. A circuit was created so that when a nail was screwed into it from outside of the payload, it complete the circuit, turning the battery packs and LED lights on. Merging the thermistor and magnetometer codes was a difficult task, but it was merged so that both the thermistors and magnetometer were under the same delay. Because the EEPROM memory of one Arduino was being shared, there were three temperature readings for every one magnetometer reading.



Figure 5. Magnetometer and Thermistors connected on one Arduino board.

One of the biggest disappointments overall was after the testing, after all of the programming had worked well. The data showed up after an hour test of the payload hanging from a tree, and it made sense. Then, on the day of the launch, the green LED never lit up -- one of the wires connected to the battery pack had fallen off, not allowing the circuit to be complete.

In hindsight, another magnetometer, like the HMC5883L Triple Axis Magnetometer would have been more strongly considered because of the Arduino UNO tutorials online specifically for this product, including tilt compensation for compass directions formulas (Love Electronics, 2012).

However, this product was not chosen due to its lack of documentation online, and by not having any tutorials online for Magnetometer 113D, a conducive learning experience was enforced, especially since the given online manual was of very minimal help, at that. Having known nothing about programming before this experience, and having spent many hours trying to do it this summer, many of the essentials were learned at a quick pace without any formal instruction. If this project could be improved, more external EEPROM memory would have been bought to eliminate delays, and the payload would have hopefully been stabilized through the help of gyros and an accelerometer to eliminate spinning, which messes up magnetic field averages. Also, most definitely, better wire protection for soldered wires would be finalized.

Thermistors

In order to determine the amount of radiation the payload would receive from the sun as it rose 10K 1% Waterproof Thermistors were used. Thermistors were used because they are easy to use when measuring simple temperature readings and they are very adaptable as well. The thermistors were attached to one inch aluminum blocks with thermally conductive adhesive, and then glued into the payload, illustrated in Figure 6 below.





Figure 6: Thermistor glued to an aluminum block and then covered with aluminized mylar for insulation

It was found through research that thermistors would be the best sensor to record temperature due to the fact that thermistors are more manageable when it comes to programming and the sensitivity of the sensor itself. The arduino programming for the thermistors (Adafruit Learning System, 2012) was fairly simple yet tedious, the program allowed for the thermistor resistance to be converted into degrees Celsius as shown in Figure 7.

```
int thermistorpin = A0;
int seriesresistor = 10000;
int numsamples = 5;
int temperaturenominal = 25;
int bcoefficient = 3950;
int thermistornominal = 10000;
int samples[5];
void setup (void) (
  Serial.begin(9600);
3
void loop (void) (
  uint8_t i;
float average;
 for (i = 0; i < numsamples; i++)(
  samples[i]=analogRead(thermistorpin);
 delay(1000);
  average = 0;
for (i=0; i < numsamples; i++)(
average += samples[i];
  average /= numsamples;
  Serial.print("Average analog reading");
  Serial.println(average);
  average = 1023 / average -1;
  average = seriesresistor / average;
 Serial.print("Thermistor resistance");
  Serial.println(average);
float temp;
temp = average / thermistornominal;
temp = log(steinhart);
temp /= bcoefficient;
temp += 1.0 / (temperaturenominal + 273.15);
temp = 1.0 / steinhart;
temp -= 273.15;
Serial.print("Temperature");
Serial.print(temp);
Serial.println("C");
  delay(1000);
3
```

Figure 7: The thermistor program converting voltage to degrees Celsius.

The initial idea for the thermistors was to use them to receive temperature readings and then take those readings and mathematically convert those readings using the equation for radiation:

$$q = h_c * A * dT$$
 (Convective Heat Transfer, 2012)

q represents the amount of heat of heat transferred, A represents the area of the surface where the heat transfer takes place, h_c represents the heat transfer coefficient, and dT represents represents temperature difference. In order to make sure the thermistors could stand up to the cooler temperatures that the payload would come in contact with an experiment was performed using dry ice in order to drastically cool down the sensor. The experiment proved that the thermistors would function accurately in cooler temperatures; the data received from the dry ice experiment is shown below in Figures 3 and 4.

Average Analog Reading	Thermistor Resistance (Ohms)	Temperature (Degrees Celsius)	Time (Seconds)
560.2	12104.58	20.76	1.00
560.8	12133.28	20.71	3.04
562.0	12190.89	20.61	5.08
561.6	12171.65	20.64	7.12
562.4	12121.16	20.57	9.16
562.6	12219.81	20.56	11.2
563.4	12258.49	20.49	13.24
564.0	12287.58	20.43	15.28
565.0	12336.24	20.35	17.32
564.6	12316.75	20.38	19.36
565.0	12336.24	20.35	21.40
565.8	12375.33	20.28	23.44
566.0	12385.12	20.26	25.48
566.8	12424.37	20.19	27.52
567.0	12434.21	20.18	29.56
567.2	12444.05	20.16	31.60
568.6	12513.20	20.04	33.64
568.2	12493.41	20.07	35.68
569.0	12533.04	20.00	37.72

Table 1: Dry Ice Experiment Data



Figure 8: Graph displays the decrease in temperature with time, with a few anomalies based on the position of the thermistor in relation to the block of dry ice.

After the dry ice experiment proved that the thermistors could still perform in cooler temperatures the next step was to glue the thermistors into the payload and get them ready for the flight of the payload. As the payload was being prepped for launch there was a realization that there might be a problem with the wiring of the battery pack that was used to power the arduino board or the wiring of the green LED light that was used to identify whether or not the different applications, including the thermistors were connected and working. Unfortunately the balloon carrying the payload was already inflated and ready to be released, so it was known that there may have been a possibility of no recorded data. Upon retrieval it was in fact determined that one of the wires connected to the battery pack had come apart and there was no temperature data recorded from the flight of the payload.

The ideal design for the security of the thermistors proved to be flawless when tested in the dry ice experiment, but was inconclusive during the flight of the payload. A possibility for the improvement of the thermistor part of the experiment would be to find a way to accurately secure all wires inside of the payload, as well as create a more easily accessible way of fixing any wiring problem that may occur upon the pre-launch setup. Although error did not allow for the necessary data to be received, based on experimentation it is believable that the initial purpose for the thermistors would have been carried out.

Cameras and Camera Mount

Before beginning the camera research, the decision came down to what kind of pictures should be taken. Eventually it decided that in addition to video, normal photos and infrared photos would be taken as well for comparison. It was determined that the GoPro HD Hero camera for the video and Canon Powershot A480 cameras for the pictures would be the best cameras to use. These camera types were chosen because the previous payload team had used them, and there was a way to access them easily. The GoPro was used for video because it had better video quality.

Because the cameras would be in the air, the team needed some way to take pictures without pressing the shutter button. The answer was to use a time lapse function, where the camera is set to take a picture every time a certain number of seconds or minutes have passed. Since the A480 cameras did not come with a time lapse function, the Canon Hacking and Development Kit (CHDK) was used. CHDK was a program that can be downloaded into a memory card and installed into a Canon camera. This hacking program made changes that improved the camera, such as allowing the user to change the shutter speeds, download scripts, and change the distance the camera zooms in on the target. Using the CHDK wiki (CHDK.wikia.com, 2012), the team found the CHDK program for the Canon Powershot A480 and downloaded it into a memory card. The site "British Ideas" (British Ideas, 2012) linked to a time lapse script for the A480 that could be downloaded onto the same memory card that held CHDK. The site also had instructions on how to activate the time lapse function.

The team practiced taking apart and putting back together the A480 camera that was left over from the previous balloon team's work. When the inner workings of the camera were sufficiently understood, the camera's normal filter was replaced with an infrared one made from a film negative. Pictures of common items like pencils were taken with both the normal filter and the infrared filter for comparison. The film negative filter made the camera produce good infrared pictures, but the team wanted better quality. So using a plastic and a glass infrared filter instead was tried, in hopes to compare the two and see which filter was the best. The plastic filter was ordered from Edmund Optical and the glass filter was ordered from Thor Labs. Since both the filters were larger than the camera, a piece the size of the original filter had to be cut from both the plastic and glass filters.

Attempts at replacing the original filter with an infrared one were made not only on the original Canon A480 but on used Canon A480's purchased online. Though the modifications initially worked, after a while an unknown problem caused all modified cameras to stop working, no matter if an infrared or original filter was placed in them. Sometimes the lens would jam and make a grinding noise, other times the screen would fill with static and the pictures would turn out dark. Because the attempts at making infrared cameras was not working and the payload was

in danger of becoming overweight, it was decided that the payload would carry only two cameras: the GoPro and a non-infrared A480 for taking time lapse photos.

Tests were performed to see how the cameras would do in flight. Both the GoPro and an A480 camera with a CHDK time lapse program installed were given fully charged batteries and turned on to see how long it would take before their batteries were depleted or their memory cards filled up. (The A480 was set to take a photo every thirty seconds.) Both cameras lasted approximately three hours, the expected duration of the payload's flight. Because there was some doubt as to whether the GoPro would fare well under cold temperatures, it was placed in a freezer where it recorded for approximately three hours. The cold did not affect its operation.

Another large part of the project was the camera mounts, which held the cameras to the payload. Integral to the camera mount were servos. The servos were meant to hold the cameras stationary even when the payload was spinning and moving. They were also meant to move the cameras to record different areas of the ground and sky.

The servos that were chosen were the GS-1 servos. They were chosen because they have a built in gyro and keep a specific place in space by providing voltage (Dunehaven Systems, 2012). When designing the camera mounts, one common element among all the designs was how they would attach to the balloon. All of the possible designs were connected to two vertical carbon fiber rods that were attached to the connection junction.

When designing the mount, the team had to figure out how the cameras would attach to the servos. It was known that one of the servos had to attach to a "mount head" that would hold the camera, and it was also known that the mount head would have to have holes go through it, so that the cameras could be attached with screws. The A480 cameras had an area on the bottom so that they could be attached to tripods, so a screw that was the right size and threading for it was found. For the GoPro, a bike mount that attached to the camera was purchased and could also be screwed into the mount head.



Figure 9: A picture of the GoPro bike mount. Picture from http://www.tourcycling.com

After multiple brainstorming sessions, the team came up with the mount head pictured below. The two holes on the arm of the "T" correspond to the holes that the screws of the GoPro mount went through. The hole on the bottom is for the nylon screw that held the canon camera in place, as seen in Figure 10.



Figure 10: A picture of the mount head. Picture by Patrick Comiskey.

Figure 11 below is the final design for the camera mount. The swing arm allows the mount head to be turned on its side, and lets the cameras be pointed at downwards angles.



Figure 11: The final camera mount. Picture by Ben Peterson.

When the payload was launched, fishing line that was rated for 50 pound strength was tied to the cameras and to the i-bolt that went through the center of the payload as a failsafe. The team had originally intended on using braided metal wire, but a thick enough braided wire could not be found.

The day of the launch, the team could see that the servos worked and successfully kept the cameras in one position before periodically switching them to a new position. Unfortunately, the wheel of the top servo in the camera mount snapped sometime during flight. The cameras and the bottom portion of the mount were not recovered.

There are several improvements the team could have made to the camera mount system. First of all, contact information could have been placed on the cameras. That way, if they fell off the payload and someone found them, they could be returned. Also, the mount could have been redesigned so that so much weight was not put on one servo; or, the cameras could have been attached to the payload itself. In addition, the team could have used a failsafe made of a stronger material such as braided steel wire, instead of fishing line. Another improvement would be devising a test that simulates the payload falling end over end, so that the team could find out how that extra stress affects the payload and camera mount.

Pressurized CO₂ Jets

The goal to stabilize the payload inspired the idea to use pressurized jets. To use jets as stabilization on the exterior of the payload, they need to be mounted as a couple. This means that jets are mounted in pairs on opposite sides of the sphere and pointing in opposite directions. This is to provide a torque without a net linear force to the center of mass. Two jets together then provide resistive torque in one direction for one plane of geometry. To stabilize in both clockwise and counterclockwise directions for a single plane, two pairs of jets are needed. Furthermore, to stabilize in both directions in all three planes would take a minimum of six pairs of jets. For both simplicity and to save weight, the objective was decided to be stabilization in the horizontal plane only. In other words, the jets' purpose was to dampen any spinning about its vertical axis.

Theoretical Thermodynamic Analysis

To begin analyzing the force that can be achieved from the jets, the mass flow rate out of the jets must first be calculated using the Bernoulli Equation with the expression for mass flow rate of an orifice to achieve the equation shown below. This will now require information about the density and pressure of the CO_2 as well as the dimensions of the nozzle used and the outside pressure.

mdot =
$$A_2 \sqrt{\frac{2(P_1 - P_2)}{\rho(1 - (\frac{A_2}{A_1})^2)}}$$

Now with the mass flow rate, a method from NASA is used to calculate the force (Benson, 2008). Using that force and the distance from the center of mass, the moment created by both jets is found. This is then inserted into a modified expression for the rotational version of Newton's 2nd Law of Motion.

$$v_{2} = \frac{\text{mdot}}{\rho \cdot A_{2}}$$

$$v_{eq} = v_{2} + \frac{(P_{1} - P_{2})A_{2}}{\text{mdot}}$$

$$F = \text{mdot} \cdot v_{eq}$$

$$M = 2(F \cdot d)$$

$$dt = \frac{I}{M} \cdot d\omega$$

With the above methodology, the time needed to open the jets to counteract a specific rotational velocity can be found as long as the inner and outer pressures are known, the density of the CO_2 , and the mass moment of inertia. This method will successfully operate the CO_2 jets once.

This method cannot be used to continually operate the CO_2 jets because the pressure inside of the canister will change depending on a few factors. To solve this issue, we must solve for the final state of the CO_2 after a certain amount is released. In efforts to simplify the problem, the canister is assumed to be adiabatic (no release or absorption of heat to or from the environment). Using an unsteady flow analysis for an adiabatic system with no input or output work, the internal energy for state two can be calculated. Also, with the known loss of mass (using mass flow rate and total release time) and a fixed volume, the specific volume of state two can be calculated.

$$u_{2} = \frac{m_{1}u_{1} - mdot \cdot (dt) \cdot h_{1}}{m_{2}}$$

$$v_{2} = \frac{m_{2}}{V}$$

Because the CO_2 is a two phase system, three intensive properties need to be determined to define state two. To achieve the third and final property, an expression was taken from the 7th Edition <u>Thermodynamics - An Engineering Approach</u> by Boles. The modified expression is shown below.

$$\Delta T = \frac{1}{c_{v}} [(u_{2} - u_{1}) - (\frac{h_{fg}}{v_{fg}} - P)(v_{2} - v_{1})]$$

Now with state two fully defined, the pressure can be found on a continual basis. That being achieved, the jets can also be operated on a continual basis.

The above methodology is completely theoretical and has not been tested. By the completion of this theory, there was not enough time to bring the experiment to fruition by the launch date and the experiment was terminated. If time allowed, the theory should be simulated through a computer code. After that, the simulation should be experimentally tested and verified prior to use on the payload because the accuracy of the force calculations directly influences the effectiveness of the experiment as a whole.

Acknowledgements

Dr. William Farrow (Advising)
Dr. Matthew Traum (Thermodynamic insight)
Rich Hajny (Wiring assistance)
Jim Yauch (Camera Mount Prototype assistance)
Arthur Weborg (Software Engineer at Milwaukee School of Engineering)
Jason Brunner (Director of Engineering, St. Croix Rods)
Tom Pease (North Sails)

References

Adafruit Learning System. (2012). *Thermistor*. Retrieved 2012, from Adafruit Learning System: http://learn.adafruit.com/thermistor

Applied Physics Systems. (2012, March 19). *MODEL 113D*. Retrieved June 2012, from appliedphysics.com: http://www.appliedphysics.com/sites/default/files/documents/Model_113D.pdf

Bayot, W. (2005, January 17). *Practical Guidelines for building a Magnetometer by Hobbyists*. Retrieved June 2012, from http://perso.infonie.be/j.g.delannoy/BAT/IntroductiontoMagnetometerTechnology.pdf

Benson, T. (2008, July 11). *nasa.gov*. Retrieved August 2012, from Specific Impulse: http://www.grc.nasa.gov/WWW/K-12/airplane/specimp.html

British Geological Survey. (2012). *The Earth's Magnetic Field: An Overview*. Retrieved June 2012, from British Geological Survey: http://www.geomag.bgs.ac.uk/education/earthmag.html

British Ideas. (2012). *CHDK and Canon A480 Quick Start Guide*. Retrieved August 2012, from British Ideas: http://www.britishideas.com/2010/06/03/chdk-and-canon-a480-quick-start-guide/>.

CHDK.wikia.com. (2012). *What is CHDK?* Retrieved June 2012, from CHDK.wikia.com: http://chdk.wikia.com/wiki/CHDK

Convective Heat Transfer. (2012). Retrieved 2012, from The Engineering Toolbox: http://www.engineeringtoolbox.com/convective-heat-transfer-d_430.html

Dunehaven Systems. (2012). *GS-1 Gyro Servo*. Retrieved June 2012, from Dunehaven Systems: http://www.dunehaven.com/gs1.htm

FAA Regulations for Kites/Balloons. (1999, November 30). Retrieved August 2012, from chem.hawaii.edu: http://www.chem.hawaii.edu/uham/part101.html

Love Electronics. (2012). *Tilt Compensating a Compass with an Accelerometer*. Retrieved June 2012, from Love Electronics: https://www.loveelectronics.co.uk/Tutorials/13/tilt-compensated-compass-arduino-tutorial

The Engineering Toolbox. (2012). U.S Standard Atmosphere Air Properties in Imperial (BG) Units. Retrieved July 2012, from The Engineering Toolbox: http://www.engineeringtoolbox.com/standard-atmosphere-d_604.html

U.S. Centennial of Flight Commission. (n.d.). Retrieved July 2012, from centennialofflight.gov: http://www.centennialofflight.gov/essay/Theories_of_Flight/atmosphere/TH1G3.htm