

# **Obtaining a Horizontal Panoramic Image in a High-Power Rocket after Landing**

Kathryn E. Lenz, David A. Juckem, Eric S. McDaniel, Jens K. Carter, Shawn R. Schumacher  
*University of Wisconsin – Fox Valley, Menasha, WI*

## **Abstract**

During the 2017-2018 Collegiate Rocket Launch (CRL) competition, teams were required to design and construct a high-power rocket meeting certain design criteria. The goals for the 2018 competition were that it would complete a safe flight, reaching as close as possible to the target apogee height of 3,000 feet, as well as take a 360° horizontal image of the surrounding landscape after landing. The UW-Fox Valley team, the Rocketeers, designed a three-inch diameter, thin-walled fiberglass airframe at a final length of 67 inches that completed three safe flights on competition day. In order to get the 360° image, a Raspberry Pi micro-computer was used with a Pi Camera, which took multiple pictures that would be post-processed and stitched together. To orient the camera correctly, three legs were attached to the bottom of the booster section of rocket which were released at the first ejection after apogee. These legs were designed to orient the booster section to land upright. The camera module, located at the top of the booster, would then take multiple pictures of the surrounding landscape.

## **1. Introduction**

For the second consecutive year, the UW-Fox Valley Rocketeers had the opportunity to compete in the Wisconsin Space Grant Consortium (WSGC) Collegiate Rocket Launch (CRL) competition. The 2018 competition goals were to complete a safe flight as close as possible to a 3,000 foot apogee, as well as take a horizontal panoramic image upon landing. Along with these goals for the rocket flight competition, teams were scored for multiple other deliverables, including multiple progress reports, community outreach, project management and a final presentation to show a panel of judges and fellow competitors the constructed rocket and the design work put into it.

## **2. Design methodology**

With such an open-ended goal, there were many possible options for the team to proceed with for the competition. First, it was decided for the initial design stages to split the objective into two parts: a landing system and a camera system. The two separate systems were brainstormed and decision matrices were used to determine the best to proceed with.

**2.1 Airframe** The airframe design needed to take into consideration that the camera needed to have a clear field of view to be able to take a panoramic horizontal view as required by the competition. To meet this design parameter, multiple designs were considered and put into a decision matrix.

The first design considered was creating a leg system to ensure that a section of the rocket would stand upright, and the camera would take pictures from the top of the upright section, which was termed the landing gear system. Initially, using a separating dual deploy system was considered so that the tether wouldn't cause any problems with the rocket staying upright. The team was unable to move forward with that particular part of the design as it was a competition requirement that all sections of the rocket be tethered. Similar to this idea, the team also considered a set of legs that

would deploy after landing, pushing the rocket into a vertical position so that the photos could be taken from the top of the upright section.

Another design considered was creating a drone system that would deploy from the rocket on landing. This drone would contain the imaging system and would take an aerial panoramic image. An offshoot of this design was a drone system that took a whole section of the rocket and design it as a drone system. This idea was also thrown out due to the inability to do a dual deploy system. Less developed ideas included a ballistic rocket, which was immediately thrown out due to high power rocketry rules and safety concerns, as well as a glider system to help orient a section of the rocket.

To come to a final decision, the team considered pros and cons of each design and put each method into a decision matrix to determine which was the best to move forward with. The criteria considered were performance, durability, complexity, price and aesthetics. For each method, the performance and durability were considered the most important criteria and were weighted the heaviest due to that. In order to maximize the points the team was able to obtain at competition, the system needed to be able to perform reliably. Additionally, due to the unpredictability of the landing site at Bong Recreational airfield, it was important to make sure the system was durable and could have repeat flights for testing and for competition.

After determining from the decision matrix that the landing gear was the design to move forward with, the team had to decide which section of the rocket to use for the imaging system. Taking into consideration the center of gravity (CG) and center of pressure (CP), the landing gear was designed to be attached to the booster stage of the rocket. Since this section of the rocket already has a low CG, it was determined this would lead to a stable touch down of this section vertically. Having the legs so close to the bottom of the rocket also helped with the stability of the rocket as it added more perpendicular drag to the rear of the rocket. This design also gave a good vantage point for a top mounted camera to take quality pictures.

**2.2 Imaging System** For the imaging system, multiple options were brainstormed and discussed to determine the best method. One of the things the team had to determine was whether they wanted an internal, or external camera system. Another set of options the team had to choose between was whether to have multiple cameras to cover the 360° area, one camera that would rotate to take different images to cover the entire area, or whether to just use one 360° camera to cover the whole area. Some of the criteria that were discussed when deciding which method to use were the cost, the quality of the pictures it would take, the durability of the system, the modularity of the system, and the complexity of the system.

A rotating camera system was ultimately chosen because of its cost efficiency and picture quality. It was decided to use a Raspberry Pi Zero and accompanying Pi Camera to capture the images. Even though a 360° camera would have simplified the system, it would have come at a large monetary expense and provided low quality "fish-eye" lens images. When it was compared to the chosen camera module, the image quality and cost efficiency was sufficiently greater. In addition, the versatility of the Raspberry Pi Zero system allowed for multiple tasks to be completed, including rotating the camera in a durable, low complexity system.

The final design was a camera module that rotated a Pi-Camera around the vertical axis of the rocket. A Raspberry Pi-Zero controlled the Pi-Camera, a continuous rotation servo, and two micro-switches that allowed the camera module to function optimally.

### 3. Airframe Construction

Once the general design approach was determined with the decision matrix, the team chose a Madcow DX3 3" kit [5], shown in Figure 1, as the core of the rocket. Many characteristics of this kit made it attractive to the team. Being constructed primarily of fiberglass, it offered a high strength to weight ratio combined with the inherent resistance to moisture. This was important to the team members due to the unpredictability of the conditions of the landing site at Bong.



Figure 1: Parts from DX3 3" kit

Preliminary design analysis using Rock-Sim [1] software indicated a length of 81 inches, a diameter of 3.1 inches and a weight between 8 and 9 pounds would provide the flight performance desired. Simulations indicate a J500G motor with a total impulse of 722.664 N·s would provide the required thrust with this design to attain 3000

ft altitude above ground level. A weight margin was incorporated to help zero in on the altitude by adding or subtracting weight based on test flight performance prior to the competition.



Figure 2: Fillets being applied to fins

The fin design called for three fins with a total surface area of 65 square inches. Construction of the fins is shown in Figure 2. Initial designs used the fins as the mounting point for the landing gear legs. Analysis of this design raised questions concerning the landing impact loads being translated to the airframe thru the fins. It was determined that the risk of damaging the fins during a hard landing was too great as repair of said fin would prove to be problematic, especially if it occurred close to or during the CRL launch. A separate machined aluminum hinge was designed with a built-in extension spring. This was bolted to the aft end of the airframe where the aft motor mount ring was located. This area is one of the strongest on the

airframe due to the multiple airframe components all being bonded together at this location. It also provided a reasonably easy repair avenue should it have become necessary.

The rocket came with a 54 mm motor mount, however the motor being used was 38 mm. While adapters were available for this situation it would have added weight, so a 38 mm mount tube was incorporated. This required new fins to be cut with a longer internal tab for bonding to the motor

mount. The bulkheads for the ejection duct were also cut at this time. All of these components were bonded to the airframe tube using rocketpoxy.

The avionics pod was next on the list. Once the sled was designed for the avionics the total length of the pod was known. The doubler was added and a hole cut for the master switch. End bulkheads were constructed and holes were drilled for the hardware required to hold the whole pod together and provide mounting for the shock cords. Wiring was straight forward and done using acceptable methods and practices found in industry.

The nose cone was assembled next after a GPS mount was designed and 3D printed. Bulkheads and transition tubes were bonded into place and hardware added for the shock cord attachment. With all the major subassemblies completed for the airframe and the parachutes chosen the team determined final lengths for the drogue and main parachute cavities. Several iterations of performance with Rocksim were performed during this phase to ensure flight performance was not being compromised. Shear pins and removable retention rivets were employed to hold the various airframe components together.

#### **4. Avionics Construction**

The team decided to use a RRC3 [6] altitude controller from Missile works powered with a 2 cell 800mah LiPo battery. The small size and excellent data acquisition made it the perfect choice for a dual deployment rocket. Provisions for the WSGC Raven [3] with a separate 9-volt power source and an external switch to turn these systems on at the launch pad was also incorporated in the avionics sled. The sled was a custom designed and 3D printed mount to provide efficient and lightweight mounting along with safe and reliable circuit management.

An Eggtimer [2] GPS locator system with an Eggtimer receiver provided tracking data to a laptop computer. The transmitter was mounted, with a dedicated power source, in the nosecone of the rocket. This was necessary to mitigate the effects of any metal in the rest of the rocket from attenuating the transmitter affecting the range and accuracy of the GPS system.

#### **5. Payload Construction**

The camera payload was designed to be a separate autonomous module. Initially a Basic Stamp was intended for the microprocessor control. After further evaluation during the preliminary design phase it was decided the Raspberry Pi [7] offered better camera options and file structure.

The camera payload was located at the top of the main booster stage to be held as high as possible above ground level after landing. This provided the best possible view of the horizon. The rules required an ejection duct to allow the motor ejection charge access to the drogue chute chamber without impacting the payload. A sub diameter (38 mm motor mount tube ) ejection duct was used around which the camera module would rotate. With these design parameters in place it was possible to begin the payload module design.

First all the main and support components were modeled in Solidworks 3D CAD. These consisted of the Raspberry Pi Zero, 3.7-volt LiPo battery, a step-up shim used to convert 3.7 volts to 5V for the Pi, a Raspberry Pi Camera Module V2, a microswitch to power the unit when the drogue chute

deployed, and a microswitch used to pause the rotation at the required points for photo acquisition. The actuation switch was used to eliminate the risk of the battery being drained while waiting an unknown length of time on the pad with the power on before the launch.

The main payload housing rotated on a Delrin bearing that incorporated a Geneva mechanism chosen to convert the servo rotation to intermittent movements required for the camera photo acquisition points. This bearing was mounted to the ejection duct and retained the module on the airframe. It also contained eight actuation protrusions for the microswitch to pause the rotation and actuate the camera. Two aluminum plates mounted the housing to the Delrin bearing and provided a bearing surface to rotate upon.

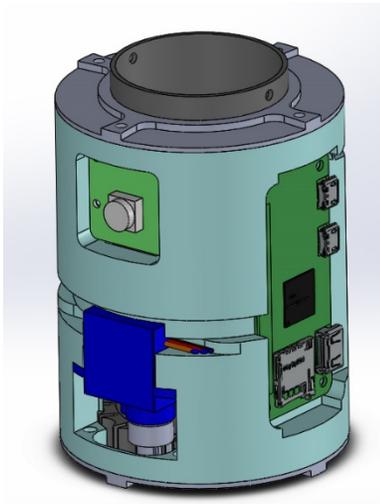


Figure 3: Final Solidworks model for camera module

Once all the components were modeled a solid was generated that represented the volume available for the camera module, shown in Figure 3. All the components were located within this volume and cavities were modeled along with mounting provisions for the components. While keeping the manufacturing process in mind, a 3D printer was used. Some areas that could be improved were noted and iterations of the housing was manufactured after incorporating the noted design changes. Using both 3D CAD modeling and rapid manufacturing techniques allowed the design to progress rapidly from concept to a working assembly.

## 6. Imaging System

The imaging system was controlled by a Raspberry Pi Zero, which was connected to a 3.7-volt Lithium polymer battery (LiPo), a power converter or shim, a continuous rotation micro servo, a Raspberry Pi Camera Module V2, and two microswitches that powered the Raspberry Pi and allowed for the camera to intermittently photograph images. Each component was housed in the main payload housing. A script was developed in Python 3.4 to make the parts work cohesively.

After the first event, the rocket separated which activated the first micro switch that was connected to the power supply. This first switch allowed for power to flow from the LiPo which powered the Raspberry Pi Zero and allowed it to begin its boot process. After the operating system booted, a Bash shell script automatically launched, which called for the main Python script to be interpreted. A predetermined timer began which accounted for the estimated time it took for the booster section to land. Once the rocket had landed and the timer reached zero, the program executed its main function, which consisted of twenty-four iterations. The camera would take an image, store it in a directory while incrementing its file name, and activate the servo motor until the second micro switch was activated by a node from the inner Delrin tubing. The node denoted when the camera was to stop, incrementing each rotation  $45^\circ$ . The full imaging system made three complete rotations. After the twenty-fourth iteration, the Raspberry Pi instructed the operating system to shut down. Once the rocket was retrieved, a team member could either access the images directly from the Raspberry Pi or extract the images from the device to an alternative PC. The team would then evaluate all the images from the landing scene and select the most consistent stream of eight images that completes a panoramic image and stitch them together using proprietary software.

## **7. Flight Testing**

The first test flight was completed on March 25<sup>th</sup>, at the Bong State Recreational Area airfield. The team flew the rocket to see how it would perform in regards to reaching target apogee and to get an idea for how the landing gear might be improved. The test flight did not go as planned, however, and the rocket took some minor damage. After preparing the rocket according to the checklist, the rocket was brought up to the Range Control Officer and loaded onto the pad. The rocket flew straight and was very stable on the way up. At its 2,458' apogee, the drogue parachute deployed along with the legs. Everything proceeded as expected until the rocket continued to fall on the drogue parachute far below the set 500' main deployment altitude. The rocket landed under the drogue, falling roughly twice as fast as it should have been and sustaining minor damage to the legs. After a close inspection, it was determined that the motor backup had ejected the drogue parachute and both ejection charges failed to ignite. It was found that a double charge failure and positive false tests on the resistance checks had occurred. It was determined tightening up the allowable tolerances for the resistance tests would keep this incident from happening again.

## **8. Imaging System Testing**

The software's algorithm was designed on paper and was initially implemented in a sandboxed testing environment. User keystrokes were used to simulate readings from the microswitch, where if the user typed a "1", the program would interpret that as a closed-circuit which activated the servo motor, or a "0" which was interpreted as an open-circuit. The software was then tested in a more realistic environment by running the script in command line from the Raspberry Pi's native operating system Raspbian, a distribution of Linux. After successful tests, verbose readouts used for debugging were removed and the program's interpretation of user keystrokes were replaced with the actual readouts from the GPIO, which monitored the microswitch's state and controlled the continuous servo motor's functionality. The camera module performed successfully in the final tests, taking photos starting at IMG\_01.jpg, incrementing the name counter, saved into the directory, and iterated a total of twenty-four times.

## **9. Final Design Modifications**

To meet the competition goal, the rocket had to gain 542' in altitude with the same amount of total thrust as the aforementioned test flight, a gain of over 20%. To accomplish this, the team turned their attention to the weight of the rocket. The team first looked at the rigging. With 20' of cordage and eight pieces of tackle, it was heavy and overdesigned. The D-loops were reduced to the next lightest available, still retaining more than double the strength calculated in a worst-case scenario. The eye-bolts used to secure lines to the bulkheads were already the smallest available, so the excess thread was trimmed, conferring a small weight reduction. The largest weight savings were realized with two major upgrades. First, the team sourced and ordered a Fruity Chutes Iris light chute [4], netting over one pound in weight reduction. Second, with this smaller, lighter chute, the forward bay was reduced by 8". A little over two pounds were reduced in total. The team spent hours sanding, clear-coating, and polishing the airframe to drastically reduce the aerodynamic drag on the length of the rocket. The final rocket schematic is shown in Figure 4.

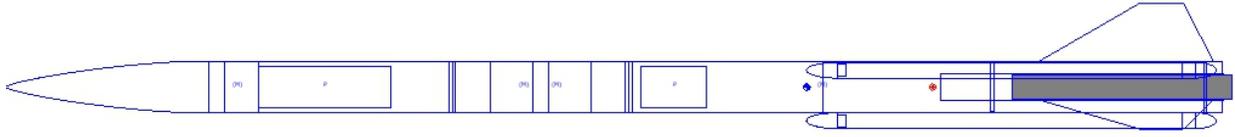


Figure 4: Final Rocket Schematics with CG (left) and CP (right) marked

## 10. Predicted Performance

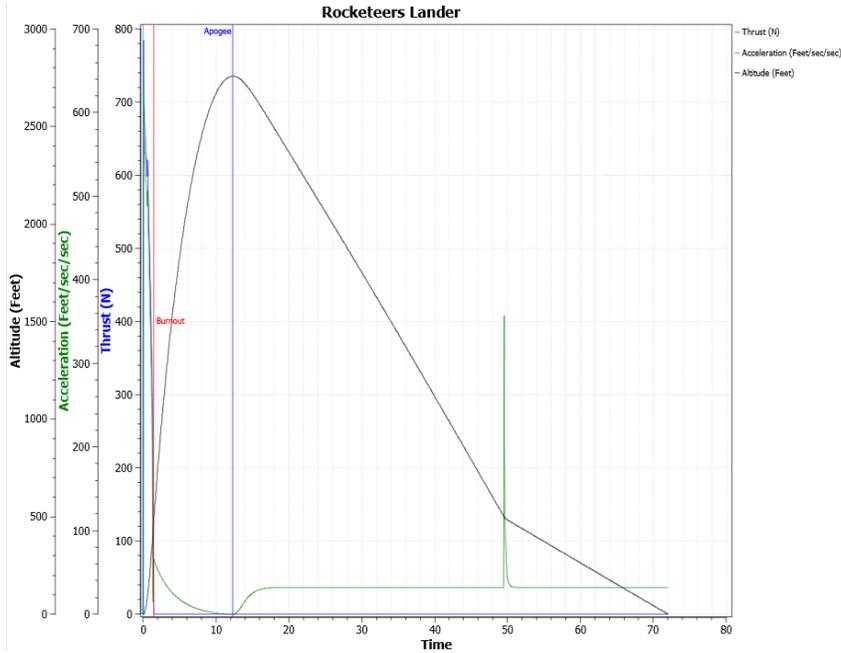


Figure 5: Simulation for first test flight of the rocket

Simulations indicated that after modifying the rocket to make it as light as possible, its weight was still just above what would be needed to successfully reach the 3,000' mark. A sample simulation for the first test flight is shown in Figure 5. The next step was to lower the drag. The drag coefficient on the rocket is a somewhat difficult thing to predict. With the known weight and motor impulse, a reasonably close estimate of the current drag coefficient was obtained for comparison.

After researching and relying on previous experience, a reasonable estimate of the drag coefficient would be attainable for the painted and polished version of the rocket. Loading this into the software left the prediction near 3,100'. From previous experience and research, it was known that the motors can have  $\pm 12\%$  total thrust and the Raven used for the competition flight logger only read down to every 11' from the launch pad. With these things combined, the prediction of around 3,100' put the team comfortably in the neighborhood of the goal of 3,000'.

## 11. Competition - Rocket Performance

On competition flight day, the rocket completed three straight flights, with the rocket landing in flyable condition. The flights completed were consistent with the calculations, simulations, and prior test flights conducted. Desired flight paths were achieved for all of the flights, with only slight variations due to the wind, which caused it to drift back towards the launch site each of the

three launches. In regards to the altitude, the first flight of the day was the best, reaching an apogee height of 3,041 feet. The final rocket design can be seen in Figure 6.

## 12. Competition - Recovery System Assessment

The recovery system was dry-tested and ground-ignited numerous times during the test phase of the build. On the final testing day, however, some major design flaws were found. On the first launch, the drogue parachute deployed violently, partly ejecting the main parachute and causing the rear bay to be expanded enough that when the main charge went off, it could no longer produce enough pressure to fully eject the parachute. It was later found that the electronic charge and the motor delay charge went off at nearly the same time, effectively doubling the force of the ejection. The rocket came down on drogue only, but fortunately landed amongst the trees, softening the blow, and took no damage. The second flight suffered a different problem, but with a similar effect. Upon launch, the parachute slid to the rear of the compartment. This compressed the charge wires, splitting the wire and again sending the rocket to the ground on drogue alone. Again, the rocket sustained no damage and was ready to fly soon after. The third flight, again, had a similar issue. This time, the flight computer gave a false positive on the main parachute charge. The rocket came down on drogue. The soft ground at Bong softened the impact slightly and allowed the rocket to survive one last time. At the end of the day three successful recoveries were done and the rocket proved that it was built well enough to withstand much more than a perfect landing.



*Figure 6: Final rocket design with landing gear legs deployed*

## 13. Competition - Imaging System Performance

For all three launches, the team found that the camera module worked successfully after landing. The software was timed appropriately and ran its course, rotating the module until all photos were stored in memory. The team unfortunately struggled with the clarity of images taken from the Raspberry Pi Camera. The photos from the first launch were mildly blurry and lacked definition, however they were the most promising photos from the overall competition. The photos from the second and third launch were severely overexposed and completely unrecognizable. Between launches, attempts were made to modify the source code to improve the images' clarity. The script imported prewritten libraries for the camera which provide functions for developers to modify how images were taken, such as shutter speed and exposure.

The team found minimal success from adjusting the source code, suggesting that the issue with the photos' clarity may have been a hardware issue.

The team suffered its largest setback with the imaging system after the competition ended. All images, excluding a few hidden images that were circumstantially redirected, were irreversibly corrupted. The Raspberry Pi's operating system Raspbian used the ext4 filesystem which is native to UNIX-based operating systems, however unrecognizable to Microsoft Windows, which operates under the NTFS architecture. After the competition, the team tried to extract the photos on to a computer running Windows by using third-party software which would recognize the ext4 Linux filesystem within Windows as a virtual machine containing an alternative file architecture. This program was consistently unstable and would repeatedly mount and unmount the external card which contained the launch day images. Improper dismounts of peripheral hard drives often risk data corruption or even a complete loss of data. The program then unexpectedly crashed corrupting all of the images within the directory that the Raspberry Pi's software had access to. Photos from only the second launch were later recovered due to a circumstantial decision during the competition to redirect the second launch's images into an alternate folder beyond the scope of the imaging system's software, however the photos still suffered from the lack of clarity and overexposure.

Attempts were made to recover the corrupted images by using free software to dissect the images' hexadecimal values and restoring them by modifying the file's header values. The team originally believed that while the header metadata was corrupted, the data representing the rest of the image would be recoverable. This was found to not be true as the image's values read 00 from start to finish, showing absolutely no recoverable data. The team later tested the imaging system as a proof-of concept and successfully extracted the images using an alternative method.

## 14. Conclusion

The Rocketeer's rocket, seen in image 7, reached an apogee height only 41 feet off from the target apogee height on competition day. The rocket came close to the apogee height due to multiple test launches completed prior to the competition. There were problems with the landing gear at the competition, and it was something that had still been undergoing changes throughout the test launches. The camera system, which had been working with clear photos when testing on the ground, proved to not work as well on competition day. While it rotated as expected, the photos were either blurry or overexposed. Then, after leaving the launch site, the image files were lost.



*Figure 7: The team recovering the rocket after the third successful launch*

While some things didn't go as well as expected, the team stayed positive and focused on how the things that didn't work could be improved for next year. As it was only the second year competing, the Rocketeers know that there is a lot of room to improve and to continue to grow in future competitions.

### **15. Acknowledgment**

The team would like to acknowledge and thank the NASA Wisconsin Space Grant Consortium for this opportunity to compete in the CRL competition. Additionally, the team would like to acknowledge the support they received from the University of Wisconsin-Fox Valley's Engineering Club and the University of Wisconsin-Platteville's Collaborative Engineering Program. The team also appreciates the guidance from their faculty adviser, Dr. Warren S. Vaz and would like to thank their industry mentor, David A. Juckem, for his expert suggestions.

### **References**

- [1] Apogee Components. RockSim v9, retrieved from <[https://www.apogeerockets.com/Rocket\\_Software/RockSim](https://www.apogeerockets.com/Rocket_Software/RockSim)>, 2018.
- [2] Eggtimer Rocketry. Eggfinder GPS Tracking System, retrieved from <<http://www.eggtimerrocketry.com/page21.php>>, 2018.
- [3] Featherweight Altimeters. Raven III Altimeter, retrieved from <<https://www.featherweightaltimeters.com/>>, 2018.
- [4] Fruity Chutes. Iris Ultra, retrieved from <<https://fruitychutes.com/>>, 2018.
- [5] Madcow Rocketry. 3" Fiberglass DX3, retrieved from <<https://www.madcowrocketry.com/3-fiberglass-dd-dx3/>>, 2018.
- [6] Missile Works. RRC# Altimeter System, retrieved from <<https://www.missileworks.com/products/>>. 2018.
- [7] Raspberry Pi. Camera Module v2, retrieved from <<https://www.raspberrypi.org/products/camera-module-v2/>>, 2018.