

# Filtering of High-Powered Rocket Pose Through Sensor Fusion with Computer Vision Horizon Estimation

Tim Player and Makoto Nara

16 April 2019

## 1 Introduction

This paper documents the successful modification of the E178 class rocket to support a payload of two video cameras. These cameras support precise localization by allowing for the determination of rocket attitude via an optical horizon estimation as shown in Fig. 1.

The rocket, which flies on a 38 millimeter J-class solid-propellant motor, achieved supersonic speeds and apogee in excess of 12,000 feet.



Figure 1: Visual horizon identification scheme during rocket flight. Approximate field of view of opposite-facing cameras is shown in red. Camera output is shown in sub-figures (b) and (c).

### 1.1 Class Rocket

The class rocket kit (based on the Mad Cow Go Devil 38 mm minimum diameter rocket) provided a stable development platform capable of supersonic flight on a J-class motor. The rocket includes pyrotechnic recovery ejection and an electronics bay containing an AIM XTRA, Featherweight Raven, and slots to attach external thermistors. Fig. 2 shows the class rocket.

### 1.2 Mechanical Alterations to Class Rocket

Using the results of Lab 4, we knew that we had to make our fillets with milled fiber.

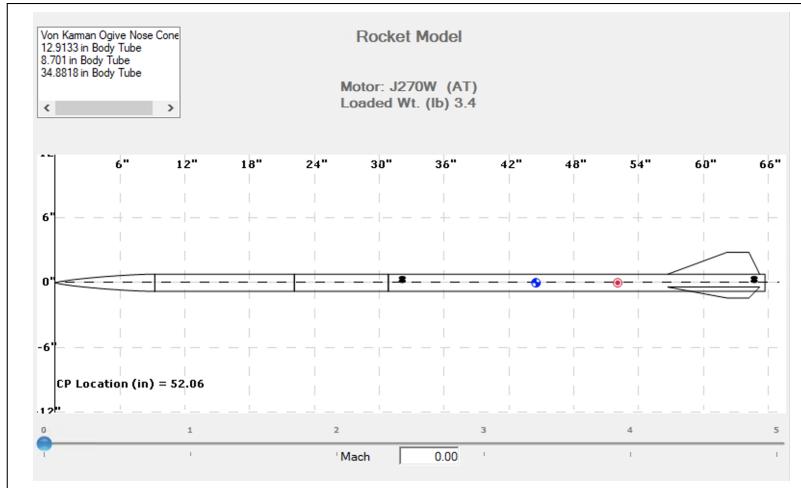


Figure 2: Model of class rocket generated in RASAero II.

We added a camera bay to the base rocket. 2 #50 size holes were drilled to secure the camera mount within the rocket, and two 0.25" holes were drilled to create a space for the cameras to look out of. The section was machined out of a standard main chute section and shortened to minimize possible shifting of the camera mount.

## 2 Computer Vision System

### 2.1 3D Printed Camera Holder

The camera mount was 3D printed to minimize production time. A model, shown in Fig. 3, was constructed in Autodesk Fusion and went through several iterations before finally settling on this design. The design is symmetric, and thus able to hold 2 cameras. The three small stands are there to interface with pre-existing holes in the camera PCB, using #0 screws. The slot in the middle is to hold one 3.7V, 1000 mAh battery for power both cameras. The power cable for that battery had to be modified so that it interfaces with the cameras. The main issue that we had to contend with for the first launch was keeping the battery secured. During our first launch, when the rocket either impacted upon the ground or when the Kevlar snapped taut during the main or drogue ejection, the battery came out and became disconnected. This killed the cameras and didn't save any of the data on them. The two loops on the top had wire fed through them and secured over the battery preventing it from coming loose. We also had to secure the lenses relatively securely and that was achieved using some rubber cement on the back to keep tight. The lenses were also secured with rubber cement in order to keep them from coming out of focus as the rocket went through various vibrations and shocks during the whole launch process.



Figure 3: 3D printed camera holder with slot for battery.

### 3 Mathematical Formulation of the Attitude Problem

The problem of estimating the rocket’s attitude from camera output is governed by the underlying projective geometry of the camera-world system. Consider the elementary graphics equation relating individual points in 3D space to camera pixels,

$$\mathbf{y} = f(\mathbf{x}) \quad (1)$$

$$= H(R|\mathbf{t})\mathbf{x}. \quad (2)$$

Here,  $\mathbf{y} = (x_{image}, y_{image}, 1)^T$  gives the position of the image on the camera plane, and is a function of  $\mathbf{x} = (x_{object}, y_{object}, z_{object}, 1)^T$ , the position of the object.  $H$  is the lens distortion matrix relating points on the ideal image plane to the actual pixel output.  $R$  is the rotation matrix between the camera orientation and global coordinates, and  $\mathbf{t}$  is the translation vector  $(x, y, z, 1)^T$  defining the position of the camera.

Eq. 1 , implies a general approach to the problem of visual pose estimation. The camera pose matrix  $(R|\mathbf{t})$ , which is an element of  $M_{4 \times 3}$ , may be estimated by the correct identification of five spatially distinct features. This technique is known as “direct linear transformation”. That approach, which is akin to triangulation, is subject to the same sensitivity problem as conventional 2D triangulation: the features must be separated by a large angular distance in order to properly identify the camera pose within reasonable error. The addition of more point pairs increases the certainty of the estimation.

The problem of extracting the pose of a camera from a single view is common in the computer vision and robotics literature, and the general approach described above has been efficiently implemented in the limiting case of structure-from-motion as “visual odometry” and “visual simultaneous localization and mapping”. These approaches are very clever, but we implemented a simpler scheme to recover only pitch and yaw instead of the full pose.

By inspection of Figure 4, it can be seen that the slope and height of the horizon in a camera image represent respectively the pitch and yaw of the rocket. If the rocket pitches into the page, then the horizon will slope clockwise in the starboard camera image, and if the rocket yaws to the left, then the horizon will rise in the page; the opposite is true for the port camera.

We make the simplification that the lens distortion matrix  $H$  can be neglected, which is reasonable given that the fisheye lens results in only minimal distortion of lines passing near the center of the image (Fig. 1. Thus, in theory, raw camera output can be converted to rocket pitch and yaw measurements by identifying the slope and height of the horizon in the image, inferring the corresponding camera attitude, and correcting for the relative rotation between the camera and rocket.

If we’d had more time, we possibly could have used OpenCV features like ORB (Oriented and Rotated Brief) to pick out key features in each frame and then find their matches in the next frame, which would have given us a sense of how the rocket rotated about its long axis. ORB is used in applications like stitching large panorama images together, so we have reasonable confidence that we would have been able to get it working with enough time.

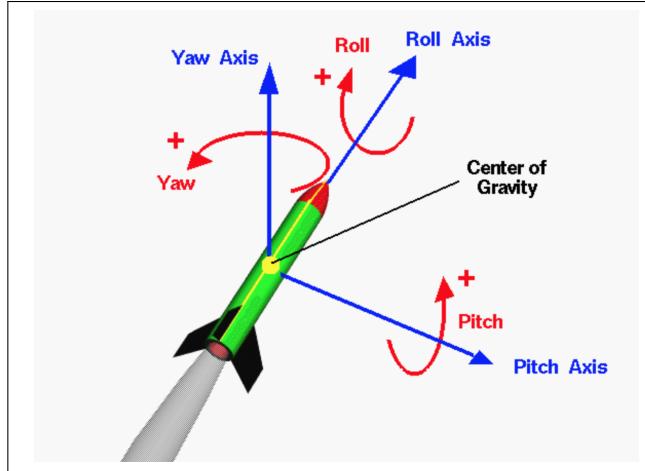


Figure 4: Attitude axis conventions. Camera 1's aperture lies on the pitch axis.

### 3.1 Computer Vision Algorithm for Horizon Estimation

An algorithm to automatically detect the horizon was implemented in OpenCV. The algorithm works by translating the image to an HSV (hue, saturation, and value) colorspace and identifying the sky. The edges of the sky region are then identified, and a straight line is fitted to the horizon. More specifically, the algorithm

- Makes a mask that includes the sky but not the ground.
- Finds the edges of that mask.
- Picks the longest edge. Assumes that contains the horizon.
- Crops the image to only examine the central rectangle (and not the edges of the circular window).
- Uses Hough line transform to identify a long straight line.

An example set of computer vision intermediate steps is shown in Fig. 5.

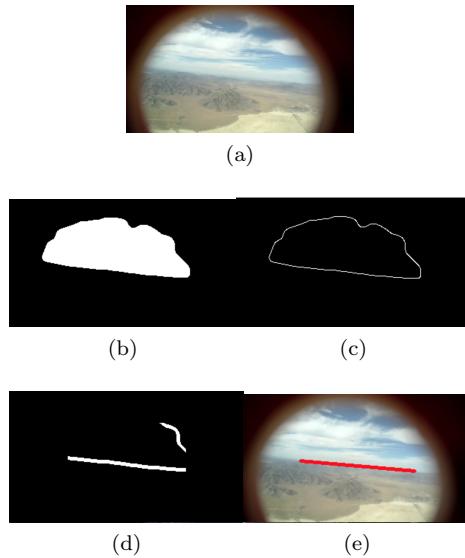


Figure 5: Computer vision algorithm for identifying the horizon in an image.

## 4 Results

### 4.1 CV Algorithm Results

Fig. 5 shows that the algorithm was capable of correctly identifying the horizon, and indeed, videos of the identified horizon consistently track what a human interpreter identifies. However, the correspondence is not perfect, and the writers recommend either applying a low-pass filter to the horizon results or hand-correcting erroneous horizon estimates.

### 4.2 Flight Characterization

Here are most of the required figures and analysis.

### 4.3 Characterization of motor performance

The flight that we analyzed had a CTI J530, a fairly strong motor, but not quite the strongest motor we launched on. Using the code we wrote for assignment 7/8, which builds upon Professor Spjut's code, determined the following thrust and mass curves. As can be seen, launch detection didn't work quite perfectly and it was around 0.3 seconds in that the launch event actually happened. This could have been due to the slightly damaged igniter that we used. We also calculated an average thrust of 243.52 N (compared to a listed 530 N) and total impulse of 1810.6 Ns (compared to a listed 1115 Ns). The average thrust is low and the total impulse is high. The average thrust might be low due to the damaged igniter that we used, causing the rocket to not fully ignite, and the high total impulse is likely due to imprecision in the trapezoidal summation of thrust curve. Also likely a large factor is that somewhere along the way, we lost our motor.

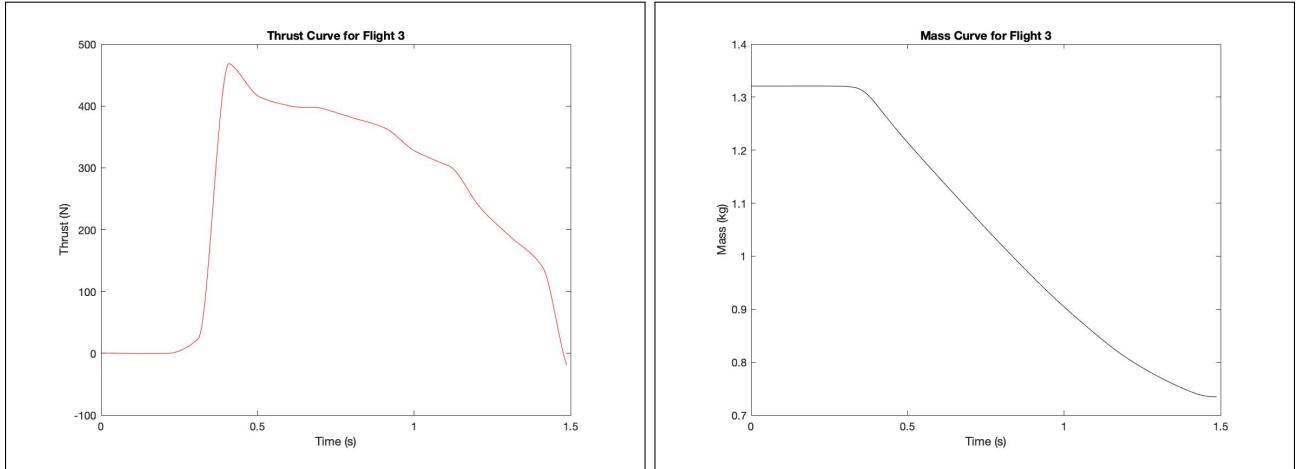


Figure 6: Flight 3 Thrust and Mass Curve

### 4.4 Characterization of flight performance

We had a couple interesting issues during this launch. First of all, our motor retainer ring appears to have fallen off (or perhaps never been attached in the first place) and so our motor was lost at some point. However, the flight of the rocket was so smooth that it is not possible to determine when the motor casing loss occurred even when inspecting video footage from the flight or sensor output. Within the flight data itself, there are a couple oddities that must be accounted for. The apogee time is remarkably hard to locate as even though the drogue deployed and the charges clearly fired, there is no indication of such an event on the z accelerometers. In addition, one of the thermistors was lost on the way down, most likely due to an impact between the electronic bay and another section of the rocket, and thus only one thermistor was able to be used in the characterization of the temperature of the atmosphere as a function of altitude. After determining the temperature and speed of sound as a function of altitude, it was a simple matter of determining the coasting region of the flight and windowing the time (in our case from 1.5 seconds to 5 seconds) so that lower speed weren't captured to get the best characterization of  $C_D$  vs Mach number (13) Our rocket only briefly went within the transonic region and never went quite supersonic.

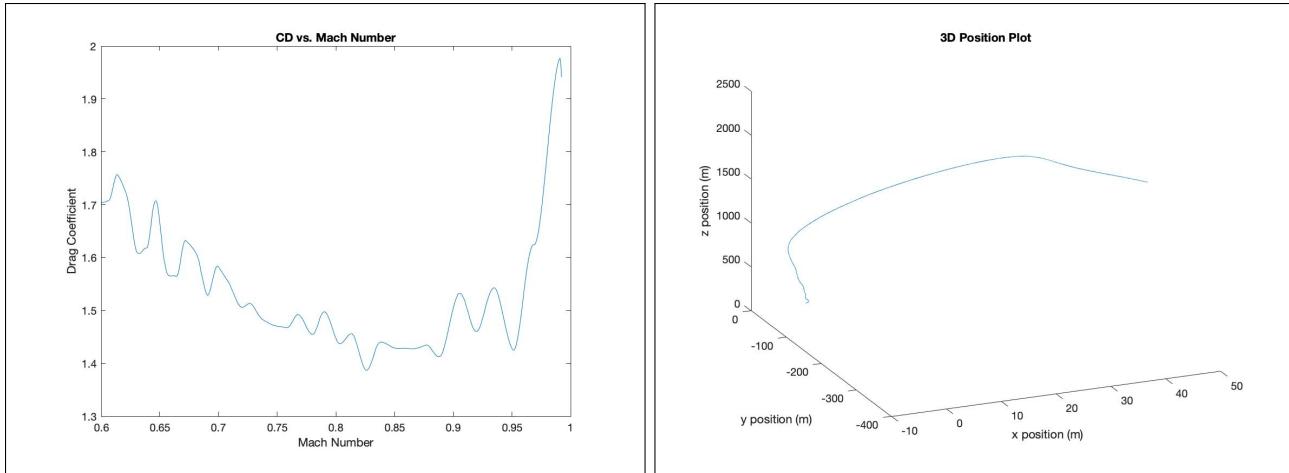


Figure 7: CD vs Mach Number and 3D trajectory

The trajectory of our rocket is a little odd, not as much weather cocking as would be expected especially considering the wind carried the rocket about a mile east once it deployed its drogue. The couple kinks in the trajectory could either be the rocket entering regions with different wind direction and behavior or also from the rocket losing its motor after burnout. Our sensors calculated an apogee of 2742.8 m or 8998.7 ft, a little low, but again likely due to our damaged igniter and insecure motor casing. It is prudent to note at this point that when watching our rocket launch it appeared to cato, with the smoke trail from the motor vanishing well before burn out should have occurred.

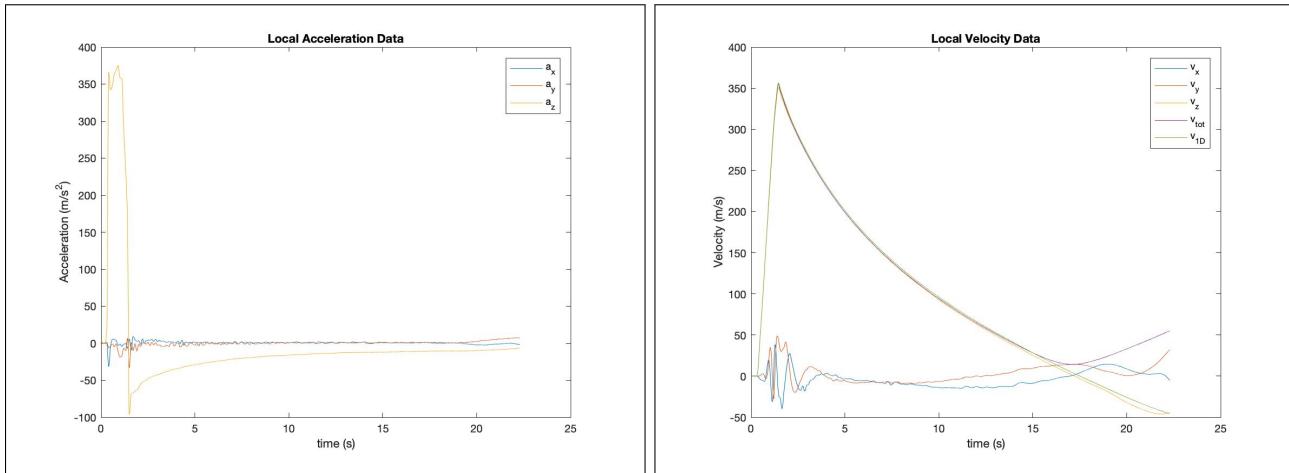


Figure 8: Local Acceleration and Velocity

Above are the local acceleration and velocity profiles, obtained from the code that Prof Spjut provided to us. We see that the flight was relatively stable and we don't see too much acceleration in either lateral direction and the velocity follows the profile we would expect with the exponential decay as the drag slows down the rocket after burn out. We see some disturbances in the x-y direction at burn out but we're unsure if this is just behaviour from burnout or actually from the motor casing being ejected.

Above we see the global acceleration and velocity profiles and by comparing against the local data, we can see that the rocket was rotating as it was experiencing some change around burnout, but that was an acceleration in purely one direction in the global sense. Below we can see the altitude in MSL and AGL, we can see that the rocket performed relatively nominally in terms of drogue and main deployment with a clear kink at around 225 m AGL indicating the ejection and deployment of the main.

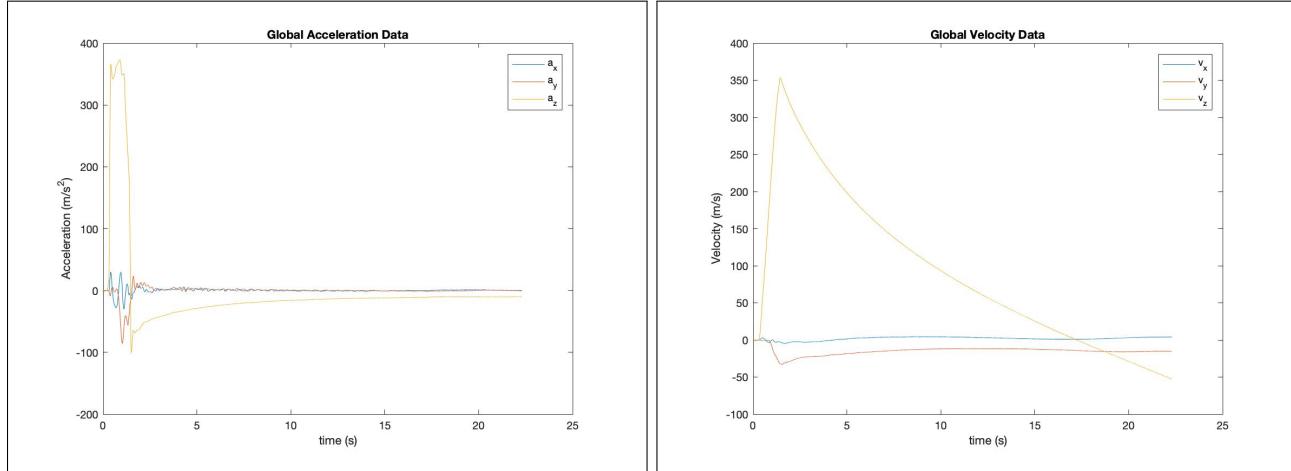


Figure 9: Global Acceleration and Velocity

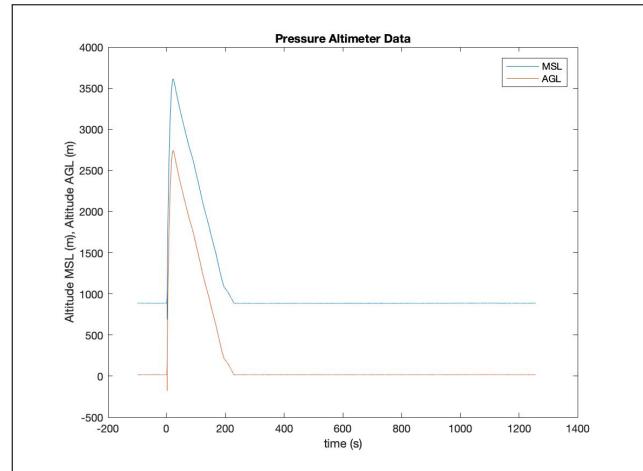


Figure 10: Local Acceleration and Velocity

## 4.5 Sensor Performance Characterization

Below we can see a couple of graphs showing the performance of the thermistors. It can be seen that one of the thermistors snapped off during the descent so the second graph shows just the one working thermistor. The measured temperatures are not quite exactly following the standard lapse rate, but they are quite close. It could have to do with various factors like the thermistor holders, which were black, absorbing a lot of heat while sitting on the pad on what was quite a hot day, and slowly bleeding that heat off as the rocket descended, messing with the thermistor readings.

On the following page we can also see the pressure vs the standard atmospheric model and as before we can see that it overlaps virtually perfectly.

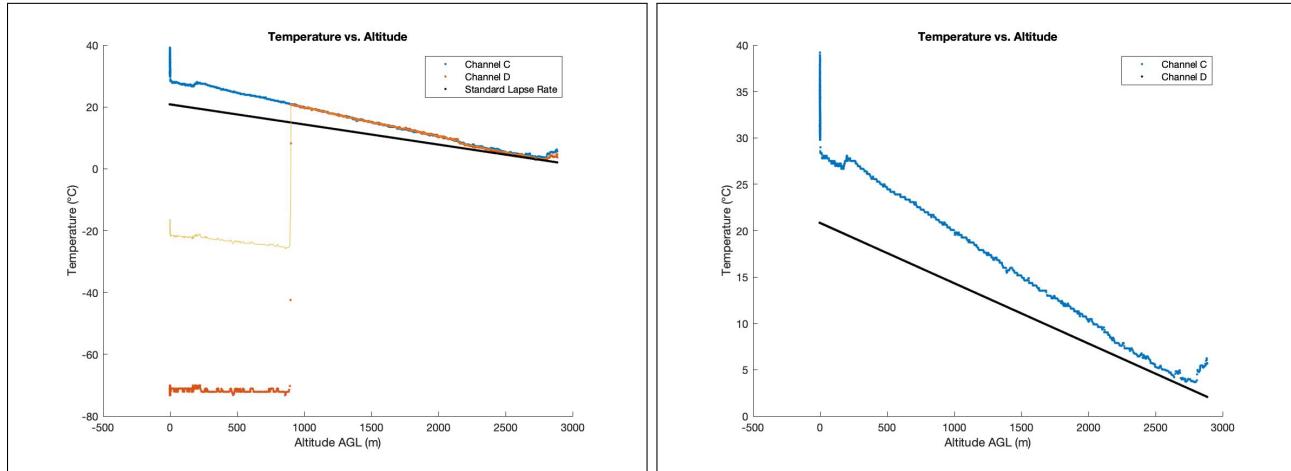


Figure 11: Global Acceleration and Velocity

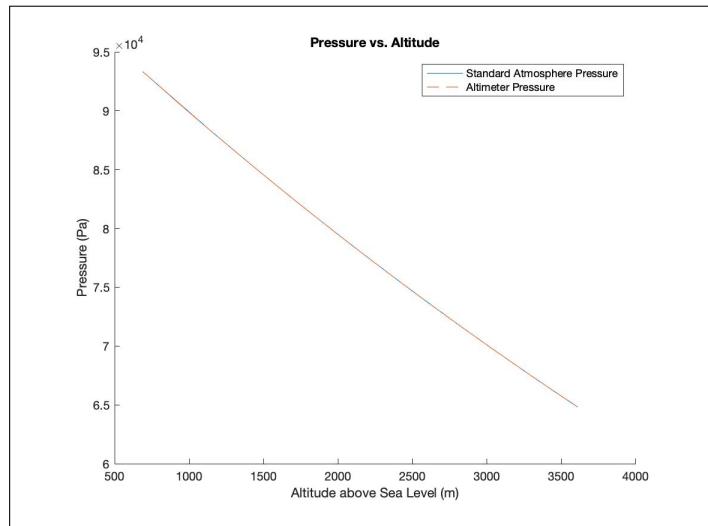


Figure 12: Global Acceleration and Velocity

Above we can see the PSD plots of the accelerometer data. The sample rate of the accelerometer was determined to be 10 Hz based on the associated time data. On the X and Y PSD plots there is not much activity, there are a couple small humps, but since the sampling frequency is so low it probably has nothing to do with the resonant frequency of the rocket body and probably more to do with changes in air speed/direction that caused oscillations in the rocket movement. The Z PSD has a much more interesting behavior, most likely caused by the motor behavior how the thrust provided by the motor is certainly not linear or constant.

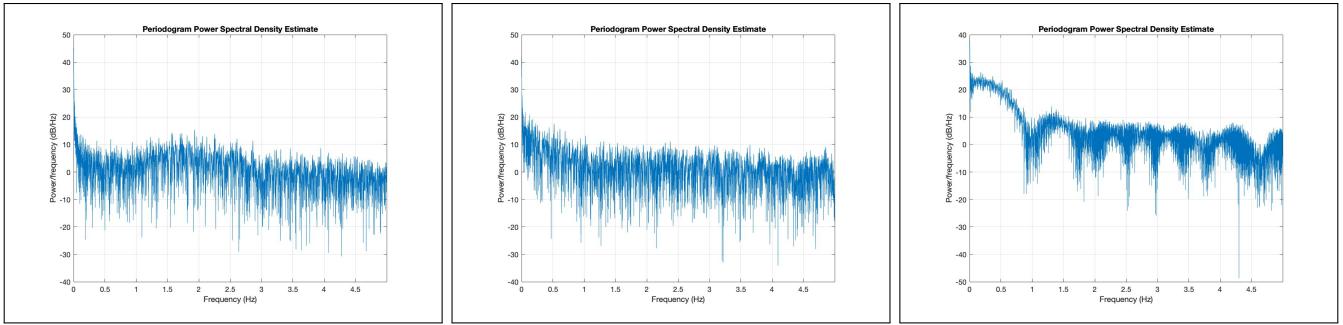


Figure 13: X, Y, and Z PSD plots

## 4.6 Rocket Performance

As discussed above, the rocket recovery systems performed pretty nominally but we didn't quite get the altitude or speed we expected due to issues with how the motor was loaded. Despite the added weight of the camera and the drag of the open holes for the camera lenses, the rocket performed admirably well.

## 5 Future Work

In the future, the output from the horizon estimation program could be incorporated into a state estimation scheme. To do so, we would need to

- a. Determine the yaw and pitch from the horizon estimates by accounting for the rotation of the camera relative to the rocket
- b. Experimentally determine the uncertainty associated with the visual yaw and pitch measurements, and verify that the errors are normally distributed.
- c. Feed the visual yaw and pitch estimates into the existing pose-estimation Kalman filter after fusing them with gyroscope and magnetometer data (perhaps through another Kalman or Monte Carlo-based filter)

Additionally, we could apply a correction for the lens distortion matrix. By performing the above steps, we would hope to improve on the existing state estimation procedure.

## References

- [1] E. R. Spjut, “Orientation & Position,” *E178 (formerly E190AJ): High-Power Rocketry*, Feb-2019.
- [2] E. R. Spjut, “Determining Noise Specifications on Accelerometers and Rate Gyros,” *E178 (formerly E190AJ): High-Power Rocketry*, Feb-2019.