Optimizing rocket stability through a combination of thrust vector control and aerodynamic design

Daniel Ghasemfar

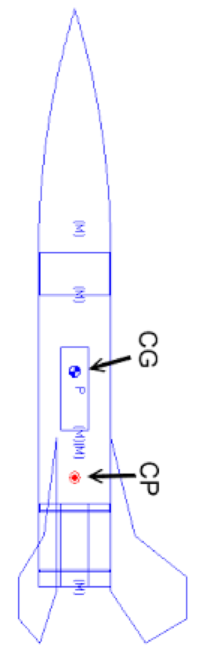
Background

The Problem of Rocket Stability

One of the greatest challenges a rocket faces as it ascends into the unknown is the problem of small-scale vibrations and more macro-scale deviations in flight path. At the same time, SpaceX is pioneering a herculean feat of engineering: creating a reliable landing rocket. The keystone technology in allowing the Falcon 9 to land 96% of the time is the Thrust Vector Control at the bottom of the rocket. When combined with the adjustable grid fins, the rocket can correct any errors on its descent to Earth. For this research project I would like to explore the feasibility of TVC and deliberate aerodynamic design in optimizing rocket stability, both macro and micro.

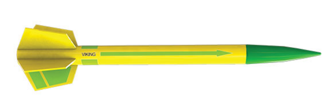
Aerodynamic Factors

Currently the primary way rockets remain stable in flight is by lowering the center of pressure (COP) below the center of mass (COM). As a rocket flies upwards, every inch of it's surface is affected by a different drag vector. The sum of all these vectors acts on one point, the COP. However, because of the laws of Newtonian Mechanics, the rocket rotates around it's COM. Therefore if the COP is above the COM, you essentially have an inverted pendulum.



COP is practically found in a wind tunnel or with CAD simulations, but to aid understanding it helps to understand the formulas discovered by James S. Barrowman in 1965. The gist of the calculations is the same as finding the COM. You take a surface integral of the pressure field vector times the distance from the stern of the ship, and divide by the total surface integral of just the pressure vectors. This is why almost all rockets have fins. More area at the bottom results in more force from pressure at the bottom, hence a lower COP and a more stable rocket.

$$\vec{F} = \oint p \, \vec{n} \, dA$$

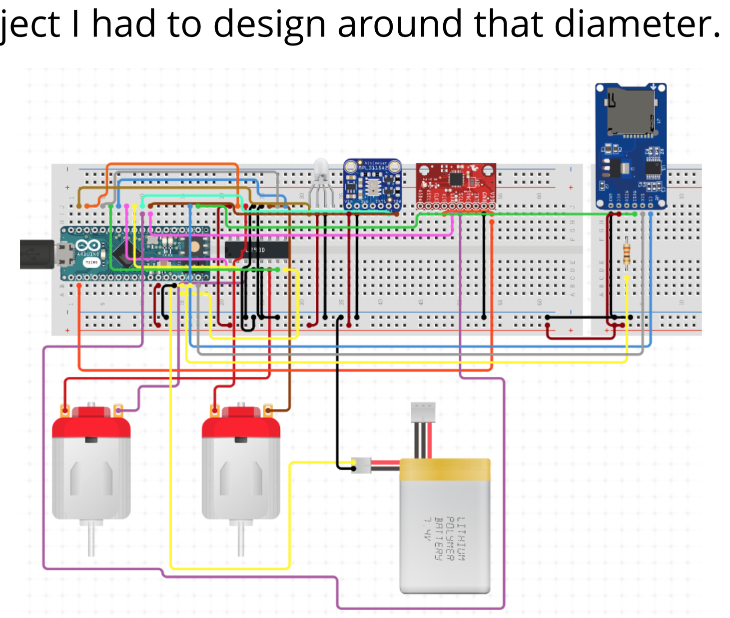


Project Design

Electronics and Rocket Motors

The first thing I considered in my project was the scale and power of the motor I would be using. The most powerful motor available without Tripoli Certification is a G motor, but because I was looking for a cheaper option I opted for the 40 Newton Second E motor. This motor is most common in a 29 mm format, so later in my project I had to design around that diameter.

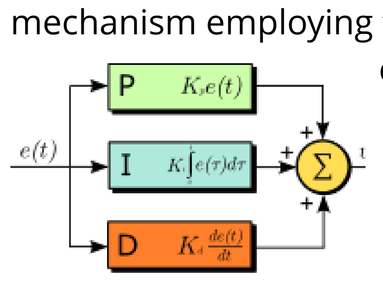
After deciding on the motor I chose the rest of my electronics. I needed two servos that were light, responsive, and cheap, so I chose the MG90 stepper motor. I went with the well-tested and cheap MPU6050 accelerometer for my input and a higher end Altimeter from Apogee Rockets.



All of the computation is done on an Arduino Uno and data is written to an SD card for post-flight analysis.

Software

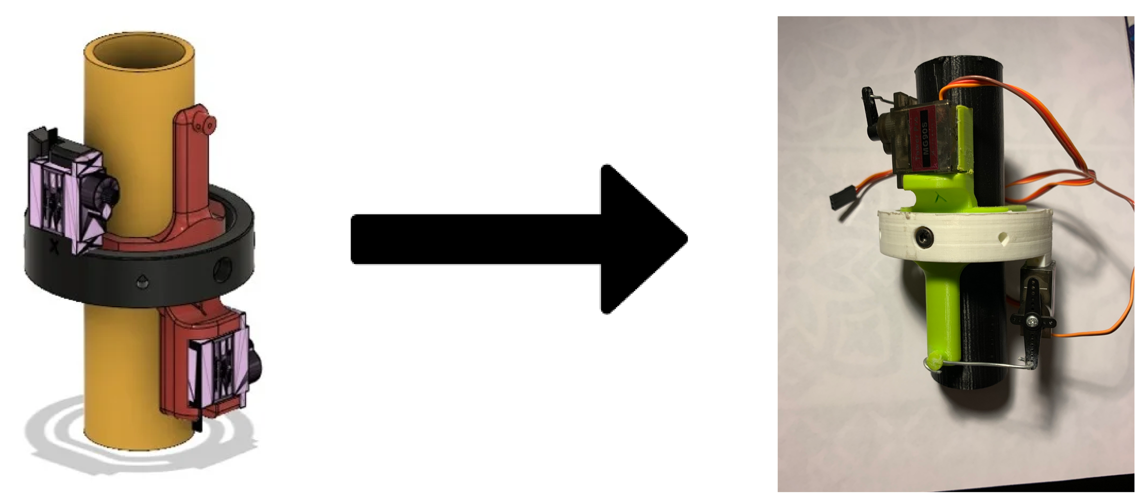
The code for this project was relatively straightforward. At the heart and soul of the whole balancing algorithm is a separate PID controller for both the X and Y axis. A Proportional-Integral-Derivative controller is a real-time control loop mechanism employing feedback that is widely used in applications requiring continuously modulated control. The Proportional term is simply a constant multiplied by the error, the Integral term is another constant multiplied by the integral from launch t=0 to time t, and the derivative term is a constant multiplied by the rate-of-change of the function.



Methods and Results

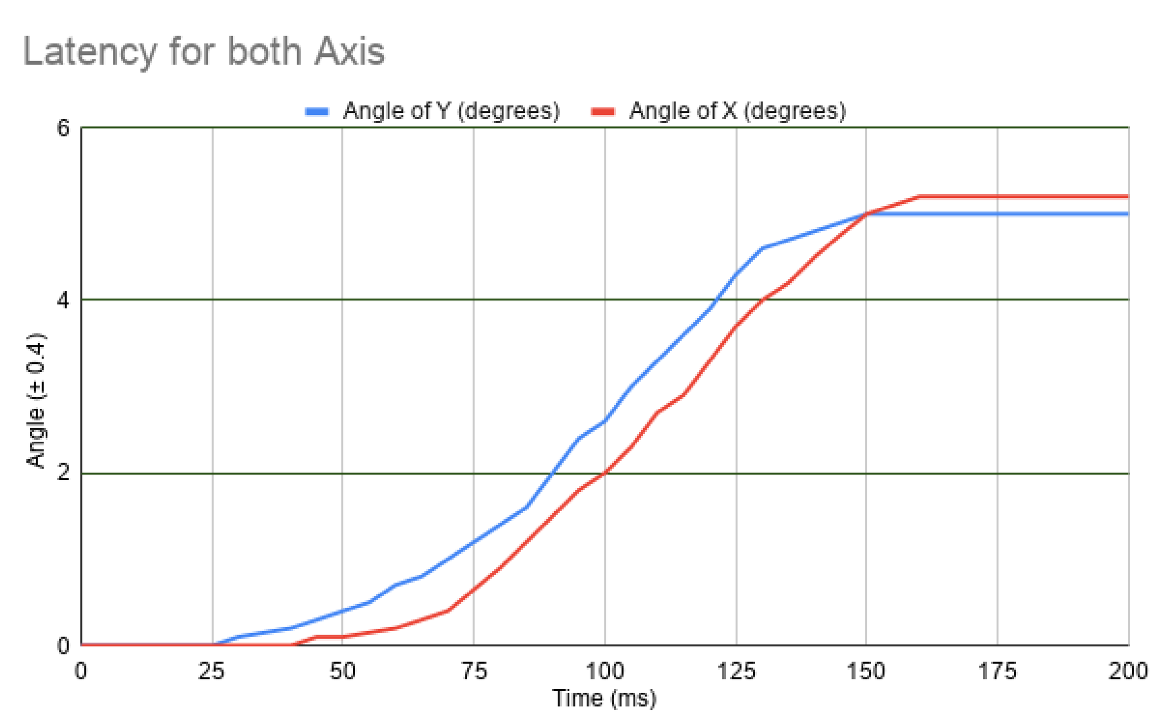
Dual Axis Thrust Vectoring Mount

Although I understood the mechanics of a dual axis vectoring mount and could likely create one in CAD, finding Rafael Guida's 3D printable STL files on Thingiverse gave my project a huge head start. Fortunately, the dimensions already fit the 29 mm E motor and the MG90 stepper motors. Unfortunately due to support structure and print instructions, it took me four tries to get the perfect shapes. After sanding down some surfaces, connecting some M3 screws, super-gluing the MG90 motors into the appropriate slots, and using steel gauge as a connection between the motors and the housing, I had a working 2D mount.



Latency Tests

Now that I had a working TVC mount, I wanted to make sure that it was quick enough to successfully implement a PID controller and hopefully make my rocket more stable. Ideally it would complete an action in 100 milliseconds or less, and it would not oscillate around the desired output. I set up my iPhone's slow motion camera directly above the TVC, along with a protractor and a light that would signal the moment an action is sent from the Arduino. I manually observed the angle over time in a video-editor and found this latency curve. The input for both tests was 5 degrees and the error is ±0.4 degrees.

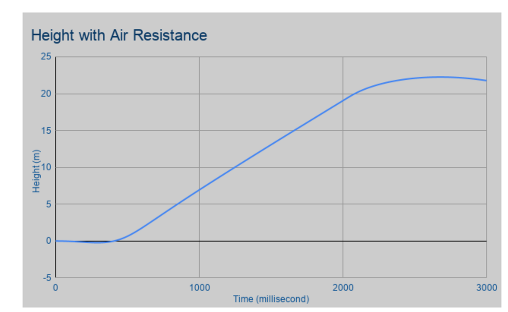


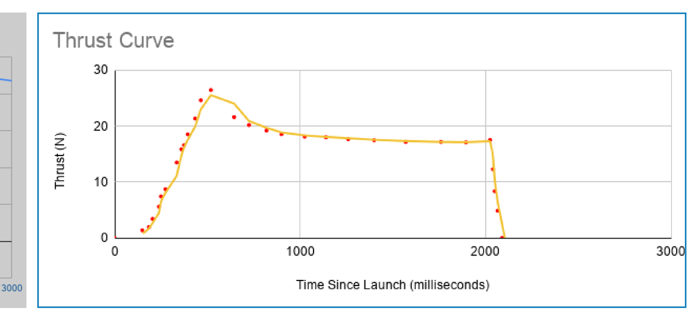
Both curves show a motion-profile or sigmoid-like curve, most likely because of the Servo.C library I ran on the Arduino. There is about 25 milliseconds of lag between when the signal is sent and when the casing moves. This is a combination of data lag and the motors overcoming the friction between the screws and the PLA. The X-axis was about 10 milliseconds slower, likely because it is both heavier and the screws on it are a little tighter.

Range Calculation and Simulation

Before launching my actual rocket and potentially loosing all my expensive components, I made a simulation of the ideal flight. To begin with, I derived a second-order, linear but non-separable differential equation to model the flight of the rocket with aerodynamic drag. The key factors are that a) the mass of the rocket changes as it looses fuel and b) the drag has a square-relationship to the velocity. The differential equation to be solved is shown below. Using Google Sheets to model the system gives me the data below that.

$$Mdu = [(p - p_0)A - Mgcos\theta - F_D]dt + vdm$$

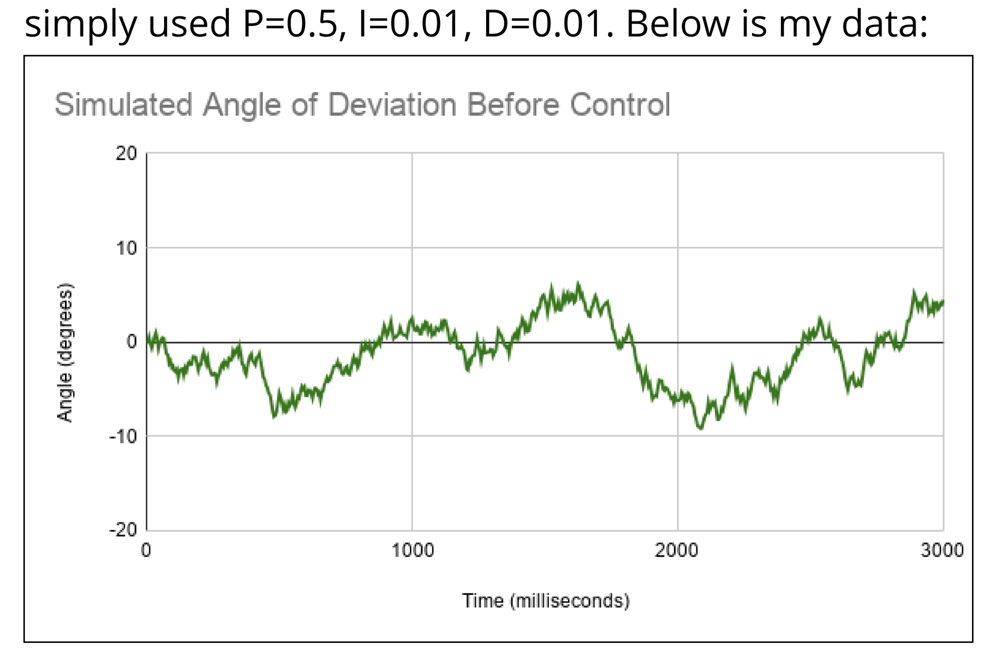


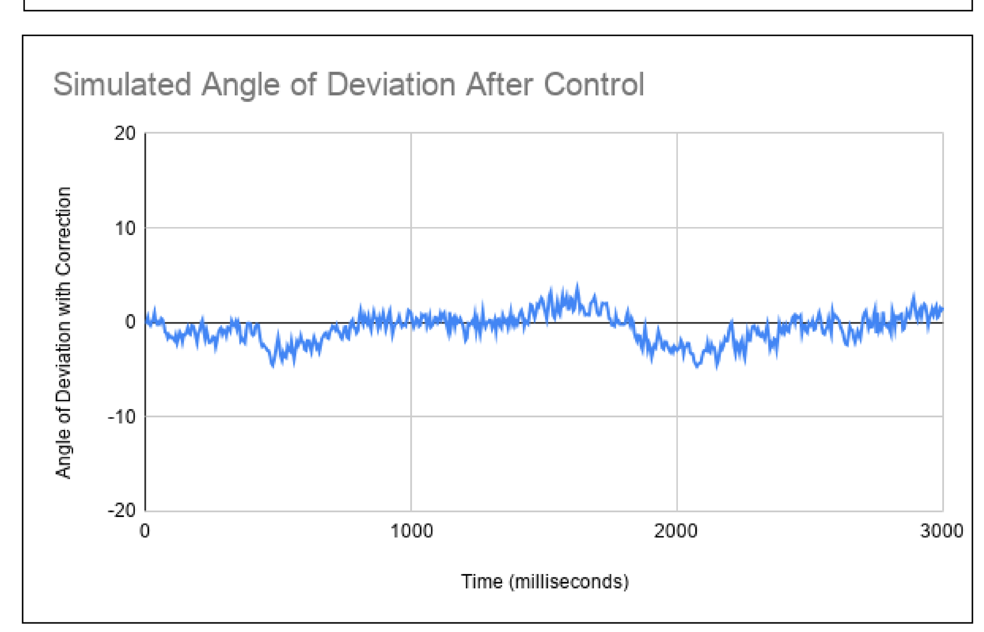


The graph above assumes a total rocket mass of 300 grams and a drag coefficient of 0.3 however both of these constants have yet to be determined. Furthermore it also assumes the thrust given by the curve to the right (collected from the Estes E30 data sheet) even though my motor will be more powerful.

PID Controller Simulations

To help me predict the result of a PID controlled motor on the ascending part of the flight, I simulated a turbulent flight and then tested the effects of vectored thrust. I used a Perlin Noise type algorithm to generate large-scale deviations (caused by wind, off-center COM, or a on-perpendicular initial launch angle) and I used random numbers in the range [-1,1] to add small, local deviations (caused by motor vibrations, harmonic oscillation, or turbulent flow). I did not tune my controller and simply used P=0.5, I=0.01, D=0.01. Below is my data:





Conclusion

Clearly in simulation the use of a TVC mount improves rocket stability. The standard deviation (which could be thought of as how far away the rocket is on average from 0 degrees) for the control was 7.261 but for the PID launch was a 3.342. Visually we can see that, because of the latency, the PID did not eliminate small-scale instability. Instead I will try to do that with custom printed fins.

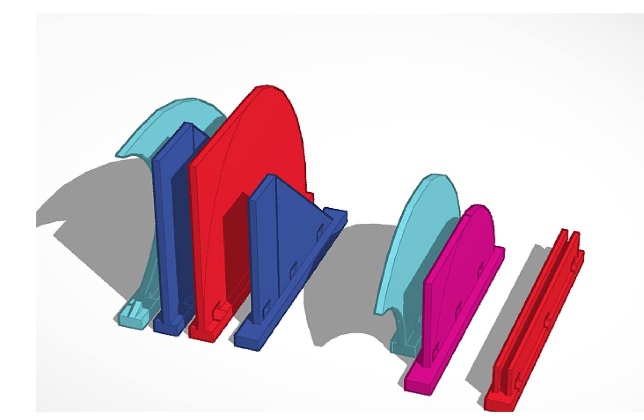
Future Works

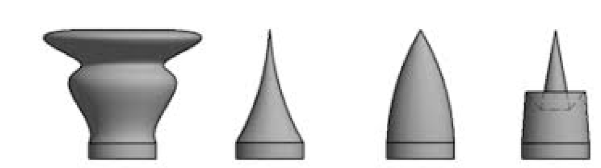
Physical Launch

The most urgent goal right now is to launch my model rocket. I am currently waiting for all the right components to arrive but once they do I will have one static test, to make sure my 3D printed TVC mount does not melt, and then one real launch, to make sure my electronics and code are all implemented correctly. Before that however I will tune my PID values on the ground by feel.

Aerodynamic Designs

After a successful launch my next goal would be to look into designing some 3D printed fins of different shapes and finding a metric to rate them by. I have already collected a lot of ideas from past designs, including: variations on the flat design, spirals, grid fins, mini-cones.





In addition to different fins I would also be curious to see what nose-cone design pairs best with the thrust vector control I am using.

Software

Using a PID controller as my correction algorithm is the most obvious choice, however in my future work I could work on implementing some more complex algorithms. I have researched Kalman filters in the past and it seems like a good choice for a launch (because there is a measured state and a calculated state). I could also experiment with guidance in simulation later on.

Works Cited

Creighton, Jolene. "Tonight's SpaceX Launch Is Breaking Records. Here's What You Need to Know." Futurism, Futurism, 6 Mar. 2018, futurism.com/spacex-falcon9-50th-flight-tonight.

The Theoretical Prediction of the Center of Pressure https://www.apogeerockets.com/downloads/PDFs/barrowman_report.pdf

Guida, Rafael. "Thrust Vectoring Mount for 29 Mm Rocket Engines (Compatible with mg90s and sg90) by RafaGuida." Thingiverse, 10 Jan. 2019, https://www.thingiverse.com/thing:3343523.

Haeussermann, Walter. "Developments in the Field of Automatic Guidance and Control of Rockets." NASA/ADS, The Bendix Corporation, Huntsville, Ala., 30 Apr. 1981, http://adsabs.harvard.edu/abs/1981JGCD....4..225.

Apogee Components, Inc©. "Technical Publication 16 - What Type of Fin Shape Is Best." Apogee Rockets, Model Rocketry Excitement Starts Here, https://www.apogeerockets.com/technical_publication_16.

Guerrero, Valeria Avila, et al. Active Control Stabilization of High Power Rocket. Santa Clara University Scholar Commons, July 2018, https://scholarcommons.scu.edu/cgi/viewcontent.cgi?article=1080&context=mech_senior.