Sprint Flight 6
Failure Analysis

Video of the flight can be found here: https://youtu.be/EXtvKCnIm6w?t=2587

Introduction

On May 20th, 2020, the BPS.space Sprint rocket lifted off the pad at 10:33 and 49 seconds, CT. The vehicle remained stable in yaw, pitch, and roll during the 1-second Ori Only phase of flight, then entered FSF(Full State Feedback) mode, where position and velocity were to be corrected for.

220 milliseconds into the FSF phase of flight, the vehicle began an aggressive pitch rotation, turning negative on the Y body axis. At this point, the vehicle began moving uprange toward the camera, in the +Z direction, due to the negative Y axis pitch.

- At T+2 seconds the vehicle was pitched -10.8 degrees around the Y axis, +1.5m in the Z direction
- At T+2.5 seconds the vehicle was pitched -22 degrees around the Y axis, +3.1m in the Z direction
- At T+3 seconds the vehicle was pitched -38 degrees around the Y axis, +6.3m in the Z direction
- At T+3.5 seconds the vehicle was pitched -46 degrees around the Y axis, +10m in the Z direction
- Also at T+3.5 seconds the vehicle began losing roll control, quickly rolling to -70 deg and back
- For the remainder of the flight, the vehicle continued in the same direction. It accumulated a total travelled distance of +90m on the Z axis, and -60m on the Y axis, reaching an altitude of 60m.

Initial Findings

The behavior of the vehicle initially suggested the presence of an axis rotation or alignment error. The behavior of the Y axis, coupled with the distance travelled in the +Z directly suggested a positive feedback loop. In this case, the greater the deviation in the Z axis position, the greater the Y axis pitch to attempt to correct for that deviation. While this was true to some extent, the problems discovered post flight are more complex than a simple axis polarity flip.

Fundamental Errors

The Sprint rocket and AVA flight computer use a right handed coordinate system to define the axes of rotation and translation. +X extends out the nose of the vehicle, +Z extends out the front plane of the AVA flight computer, toward the camera in most shots, and +Y extends out the left of the AVA flight computer when facing it, to the left of the camera in most shots.

Copyright 2020, Barnard Propulsion Systems

Measurements enter the system in multiple coordinate frames, and here we will primarily focus on two. The inertial frame, which represents the orientation of the vehicle on the launch pad just before liftoff. And the body frame, which travels along with the vehicle during flight, as it yaws, pitches, and rolls. All guidance and navigation is computed in the inertial frame, then any nonzero setpoints are rotated into the body axis through the roll of the vehicle in order to control it effectively.

The first error that I found, and the first error that is mathematically encountered in the control system, was the inertial axis polarity. While upwards travel of the vehicle did correspond with +X readings, both the Z axis, and the Y axis position and velocity measurements were reversed from what they should have been. This can be seen in the flight data and telemetry, where the +Z uprange translation was displayed as -Z translation, and -Y translation was displayed as +Y translation.

Double Negatives

Interestingly, this on its own did not cause any problems. Other than confusing flight data, the vehicle had no trouble caused by this axis reversal. That is because when rotating measurements from the inertial frame into the body frame, the axes were flipped again. This means that as the vehicle travelled in the +Z direction, while it incorrectly saw -Z movement in the inertial frame, it correctly saw +Z movement in the body frame. The two bad rotations cancelled eachother out, which helps explain why the error was not caught before launch. The correct axes and movements were being fed into the control system, as there were no guidance commands in the inertial frame(just driving position and velocity to zero), and the body axis measurements used in the actual control loop were correct.

Full State Feedback Control

This is where the heart of the error took place, and the issue is somewhat complex. AVA uses a full state feedback controller to keep the vehicle on course. The states in the state vector are, YZ orientation, YZ gyro rates, YZ position, and YZ velocity. It is an 8 state vector, which neglects the roll orientation, and altitude of the vehicle. The roll controller is a PID, and the altitude is a fundamentally uncontrolled state since we do not have control over the thrust of the solid rocket motor.

When multiplying the state vector with the K matrix(gain matrix), we get two outputs, which represent torque in newton meters. This is the torque that the controller decides will drive our system from any given current state, to zeros all around. Those torque measurements are then converted into the appropriate TVC angle by taking into account the vehicle moment arm, and its current thrust. Let's do a very basic overview of how the gyroscope section of the controller works.

Gyro Control Example

Consider a full state feedback controller, but neglect everything except the Y axis gyro control.

1. We measure an angular rate of +0.2 radians per second
2. Assume a gyro gain of 0.5
3. +0.2rps gets multiplied by the +0.5 gain to create a torque output of +0.1 N·m
4. +0.1 N·m is not going to stabilize the system!

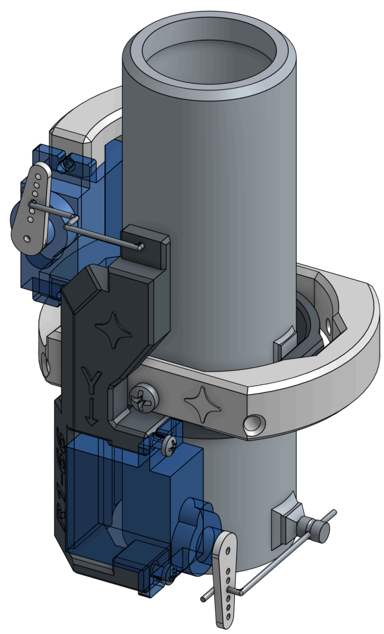
Why is it not correct? The torques and angular rates have matching signs, or polarity. A positive torque will create a positive angular rate. A negative torque will create a negative rate. So a feedback controller which sees a positive angular rate, and outputs a positive torque on that axis is, in every sense, a positive feedback controller. That system will become unstable very quickly.

Copyright 2020, Barnard Propulsion Systems

AVA's Current Control Setup

AVA's full state feedback controller was set up in the same way as the Gyro Control Example above. When feeding in a positive angular rate or orientation, we had positive gains on angular rate and orientation, which lead to positive torque outputs. Ordinarily this would not work, however, as mentioned, the torque outputs do not get directly fed into the servos of the thrust vector control mount. They first go through several steps, which we'll look at here. During one of these steps, it's fundamentally required that both the Y and Z control output signs are flipped so we have a negative feedback loop, which is what we want.

1. The torque output from the controller is converted into TVC angle, signs stay positive
2. The TVC alignment integrator is added, signs stay positive
3. The TVC angle is limited to ±5 degrees, signs stay positive
4. The TVC angle is scaled to the TVC mount gear ratio, creating the servo write value, **signs do NOT stay positive**
    a. The Y axis output is **flipped negative** at this step
    b. The Z axis output **remains positive** at this step
5. The servo write value is added to the respective TVC servo midpoint, usually around 90. Values stay at their current polarity/sign
6. The servo write value plus the midpoint is sent as a PWM signal to the respective TVC servo.



The reason for this sign flip at step four has to do with how the BPS R2 TVC hardware is set up.

- When the servo which controls the Y axis rotates in a **positive** direction, it tilts the mount in a **negative** Y rotation, which creates a **positive** Y torque on the vehicle. Thus, for whatever torque is desired on the Y axis, **no sign flip *should* be required.**
- When the servo which controls the Z axis rotates in a **positive** direction, it tilts the mount in a **positive** Z rotation, which creates a **negative** Z torque on the vehicle. Thus, for whatever torque is desired on the Z axis, **a sign flip *is required.***

This does not initially make sense, given the description of how the control system works. A sign flip on the Y axis should be NOT BE required, and a sign flip on the Z axis SHOULD BE required. The flight software is set up in the opposite way. We flip the sign on the Y axis, and leave the Z axis alone. Why?

The problems all cancel out when we remember that the FSF torque controller on AVA is set up incorrectly for our scenario. Positive angular rates generate positive torque commands, which means we must always flip the signs somewhere down the line. At the risk of boring the reader, let's walk through the steps very clearly.

Copyright 2020, Barnard Propulsion Systems

1. An example angular rate of +0.2rps on the Y axis is sensed
2. +0.2rps is multiplied by +0.5, which is the Y axis gain, resulting in a torque output of +0.1N · m
3. +0.1N · m is converted to an angle for the Y axis servo. We will pretend this is +3 degrees
4. +3 degrees has the alignment error added to it, we assume this is 0.
5. +3 degrees is saturation limited to ±5 degrees, which does not change the value
6. +3 degrees is scaled to the correct gear ratio, which is in this case, 6:1, resulting in +18 degrees
    a. At this step, all the positive values are flipped negative for the Y axis. +18 becomes -18.
7. -18 degrees is added to the Y axis midpoint, which we will assume is 90. Resulting value is 72.
8. 72 degrees is written to the Y axis servo. This is a negative rotation from the center point of 90.
9. The -18 degree rotation, rotates the motor mount **positive** on the Y axis.
10. The **positive rotation** of the motor mount creates a **negative torque** on the vehicle because the side force enters **below** the center of mass.
11. **This shows how the errors cancel out because of how the servos are mounted.**

Similarly, these steps occur for the Z axis, but without the sign reversal at step 6a. This also closes, and the system remains stable.

When The Errors Stack Up To High, You Lose The Game Of Jenga…

Even with all the errors and discrepancies discussed here, the rocket should still fly well, right? This is a correct assumption when the full state feedback controller only takes orientation and angular rate into account. However, things become more complicated when horizontal position and velocity are added into the mix. Once again, I'd like to walk through how the controller works, step by step. In this example, we are trying to control the position along the Z axis, by changing the attitude on the Y axis.

1. A deviation of +1m is measured on the Z axis
2. The position gain is 0.5, so when multiplied through the K matrix, the resulting torque is +0.5N · m
3. The K matrix is set up so that torque is commanded on the Y axis, not the Z axis.
    a. Translational motion along one axis must be corrected with an angular change on the alternate, orthogonal axis
4. +0.5N · m is commanded on the Y axis
5. +0.5N · m is converted to an angle in the TVC mount - we will pretend this is +3 degrees
6. +3 degrees has the alignment error added to it, we assume this is 0.
7. +3 degrees is saturation limited to ±5 degrees, which does not change the value
8. +3 degrees is scaled to the correct gear ratio, which is in this case, 6:1, resulting in +18 degrees
    a. **Here, all the positive values are flipped negative for the Y axis. +18 becomes -18.**
9. -18 degrees is added to the Y axis midpoint, which we will assume is 90. Resulting value is 72.
10. 72 degrees is written to the Y axis servo. This is a negative rotation from the center point of 90.
11. The -18 degree rotation, rotates the motor mount **positive** on the Y axis.
12. The **positive rotation** of the motor mount creates a **negative torque** on the vehicle because the side force enters **below** the center of mass.
13. **NEGATIVE TORQUE ON THE Y AXIS WILL SEND US FURTHER POSITIVE ON THE Z AXIS**

At last we have arrived at the problem!

Copyright 2020, Barnard Propulsion Systems

Sprint Flight 6
Failure Analysis

BPS.SPACE

Summing Up The Problem

There are several issues to fix here, some of which did not technically cause the problem, but are underlying silent bugs. The primary issue is that the FSF controller is not set up correctly. Commanding positive torque when we see positive angle or angular rates is incorrect, and attempting to correct for that issue down the line is not only wrong, but is the root cause of the failure of Sprint Flight 6.

The controller was haphazardly set up, in a way that seemed correct on the ground. All control outputs were tested in isolation, and the entire full state feedback controller was never put together all at once. This created a perfect storm of circumstance to miss this issue. I tested position and velocity gains and directions at one point, while testing gyro and angle gains at a separate point. Moreover, I did not step through every part of the controller’s process.

Had I done a more thorough review of the torque outputs from the controller in reference to the coordinate definitions selected for this vehicle, I likely would have caught this error before launch.

Action Items

Going forward, there are several steps to take to ensure something like this cannot happen again. They are listed here, roughly in order of when each problem should be looked at and resolved.

1. Walk through each transformation of GNSS data
    a. Does Lat Lon correspond to the correct axis signs when the vehicle is pointed true north?
    b. When rotated through ground heading, does the inertial data hold up as well?
        i. Test position and velocity along Y and Z axes
    c. Rotate inertial GNSS data to body axis GNSS data.
        i. At 0 deg roll angle, inertial and body axis measurements in the Y and Z axes must match entirely
        ii. They are currently the complete opposite. Which, while that did not cause problems on its own, is extremely problematic in general
2. Reverse polarity/sign of angular rate and orientation gains
    a. Then work through all post FSF steps to ensure we create the correct torque commanded by the FSF controller
3. Walk through, step by step, the Y and Z axis position and velocity torque generation, and which gains must have their polarity flipped.
    a. Through somewhat fast analysis, based on the current coordinate definitions, Z position gain should be positive, and Y position gain should be negative

This concludes the failure analysis of Sprint Flight 6. The next flight attempt is expected to happen within one week from writing this. Written on May 23, 2020.

Copyright 2020, Barnard Propulsion Systems