Quintin Nelson

Kade Carlson

AERSP 304

4/6/22

Project 2 Report

Participation:

Quintin Nelson: 50%

Kade Carlson: 50%

We pledge that we have neither given nor received assistance on this project.

Signed: Quintin Nelson

Signed: Kade Carlson

Problem Formulation:

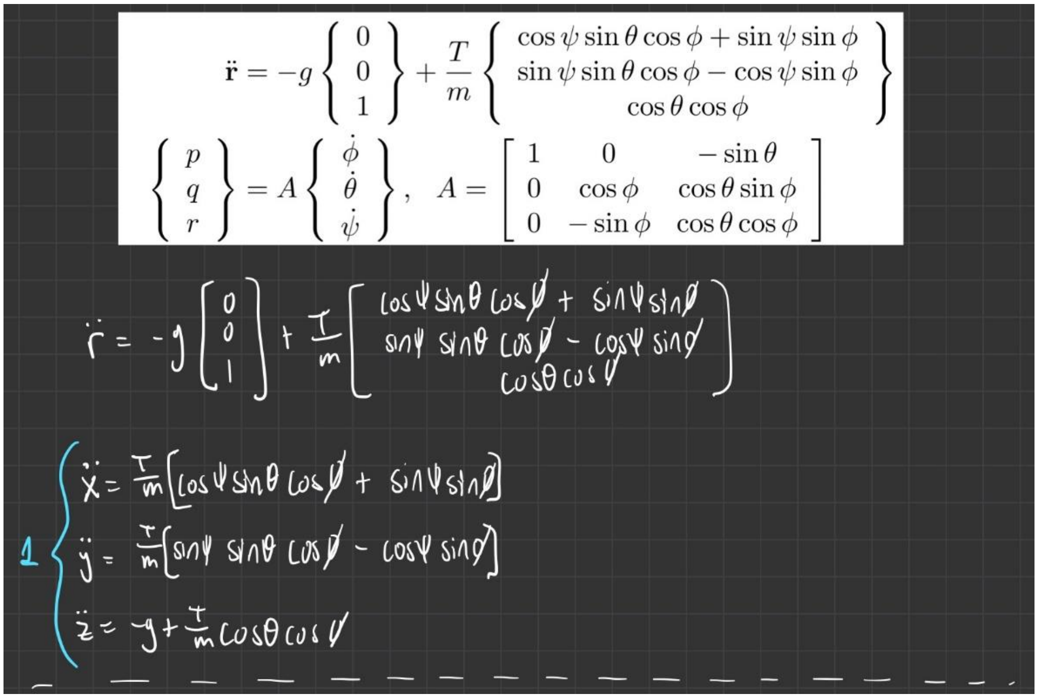
The objective of this project was to analyze how a six-degree of freedom quadcopter's motion is affected by several different rotor velocities. The four quadcopter rotors control roll, pitch, yaw, and elevation of the craft. To understand the resultant motion of the quadcopter, position, body velocity, angles of rotation, and the angular body rates were found.

Solution Methodology:

Given initial conditions and equations for translational acceleration; Euler angle change over time; body angular accelerations; roll, pitch and yaw moments; and angular velocities, ODE45 was utilized to numerically determine the motion of the quadcopter over time. Since ODE45 requires first-order ODEs, several equations of motion had to be reduced from second-order. This, in addition to the fact that many variables show up across multiple equations, a state space was created.

To do so, each equation of motion was manipulated to translate from matrix form to ODE form.

First, translational acceleration was broken up into x, y, and z components, as shown in Figure 1.



$$\ddot{\mathbf{r}} = -g\begin{Bmatrix}0\\0\\1\end{Bmatrix} + \frac{T}{m}\begin{Bmatrix}\cos\psi\sin\theta\cos\phi+\sin\psi\sin\phi\\ \sin\psi\sin\theta\cos\phi-\cos\psi\sin\phi\\ \cos\theta\cos\phi\end{Bmatrix}$$

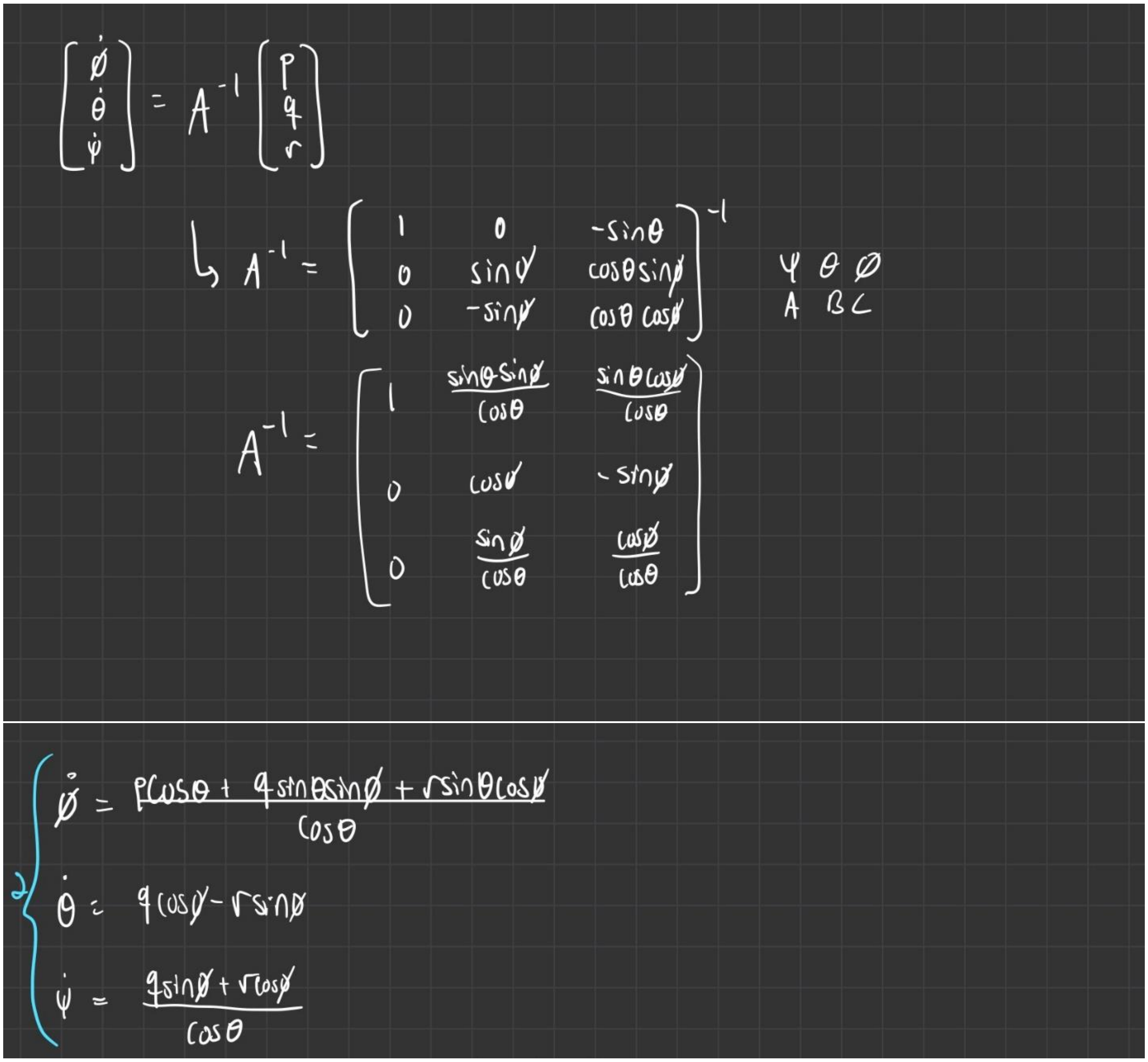
$$\begin{Bmatrix}p\\q\\r\end{Bmatrix} = A\begin{Bmatrix}\dot\phi\\ \dot\theta\\ \dot\psi\end{Bmatrix},\quad A=\begin{bmatrix}1&0&-\sin\theta\\0&\cos\phi&\cos\theta\sin\phi\\0&-\sin\phi&\cos\theta\cos\phi\end{bmatrix}$$

$$\ddot{r} = -g\begin{bmatrix}0\\0\\1\end{bmatrix} + \frac{T}{m}\begin{bmatrix}\cos\psi\sin\theta\cos\phi+\sin\psi\sin\phi\\ \sin\psi\sin\theta\cos\phi-\cos\psi\sin\phi\\ \cos\theta\cos\phi\end{bmatrix}$$

$$1\begin{cases}\ddot{x} = \frac{T}{m}\left[\cos\psi\sin\theta\cos\phi+\sin\psi\sin\phi\right]\\ \ddot{y} = \frac{T}{m}\left[\sin\psi\sin\theta\cos\phi-\cos\psi\sin\phi\right]\\ \ddot{z} = -g+\frac{T}{m}\cos\theta\cos\phi\end{cases}$$

Figure 1. Translational acceleration

Next, the equations for the time derivatives of the Euler angles were derived, as shown in Figure 2.



$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = A^{-1}\begin{bmatrix}p\\ q\\ r\end{bmatrix}$$

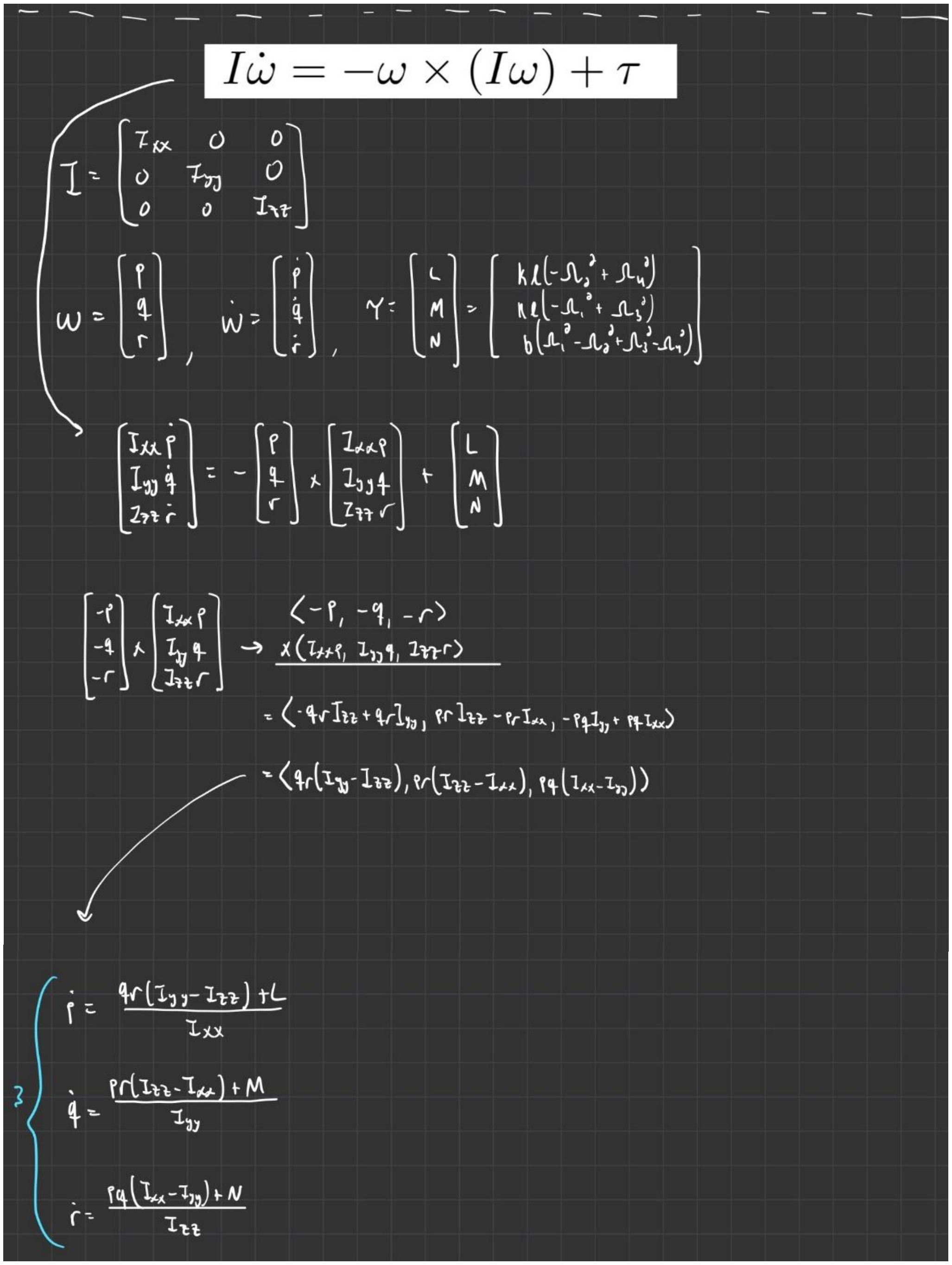
$$\hookrightarrow A^{-1} = \begin{bmatrix}1 & 0 & -\sin\theta\\ 0 & \sin\phi & \cos\theta\sin\phi\\ 0 & -\sin\phi & \cos\theta\cos\phi\end{bmatrix}^{-1} \qquad \begin{array}{ccc}\psi & \theta & \phi\\ A & B & C\end{array}$$

$$A^{-1} = \begin{bmatrix}1 & \frac{\sin\theta\sin\phi}{\cos\theta} & \frac{\sin\theta\cos\phi}{\cos\theta}\\ 0 & \cos\phi & -\sin\phi\\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta}\end{bmatrix}$$

$$2\left\{\begin{aligned}\dot{\phi} &= \frac{p\cos\theta + q\sin\theta\sin\phi + r\sin\theta\cos\phi}{\cos\theta}\\ \dot{\theta} &= q\cos\phi - r\sin\phi\\ \dot{\psi} &= \frac{q\sin\phi + r\cos\phi}{\cos\theta}\end{aligned}\right.$$

Figure 2. Time derivatives of the Euler angles

Next, the time derivatives of the angular body rates were found, as shown in Figure 3.



$$I\dot{\omega} = -\omega \times (I\omega) + \tau$$

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}$$

$$\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \quad \dot{\omega} = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}, \quad \tau = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} kl(-\Omega_2^2 + \Omega_4^2) \\ kl(-\Omega_1^2 + \Omega_3^2) \\ b(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{bmatrix}$$

$$\begin{bmatrix} I_{xx}\dot{p} \\ I_{yy}\dot{q} \\ I_{zz}\dot{r} \end{bmatrix} = -\begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx}p \\ I_{yy}q \\ I_{zz}r \end{bmatrix} + \begin{bmatrix} L \\ M \\ N \end{bmatrix}$$

$$\begin{bmatrix} -p \\ -q \\ -r \end{bmatrix} \times \begin{bmatrix} I_{xx}p \\ I_{yy}q \\ I_{zz}r \end{bmatrix} \rightarrow \frac{\langle -p, -q, -r \rangle \times \langle I_{xx}p, I_{yy}q, I_{zz}r \rangle}{}$$

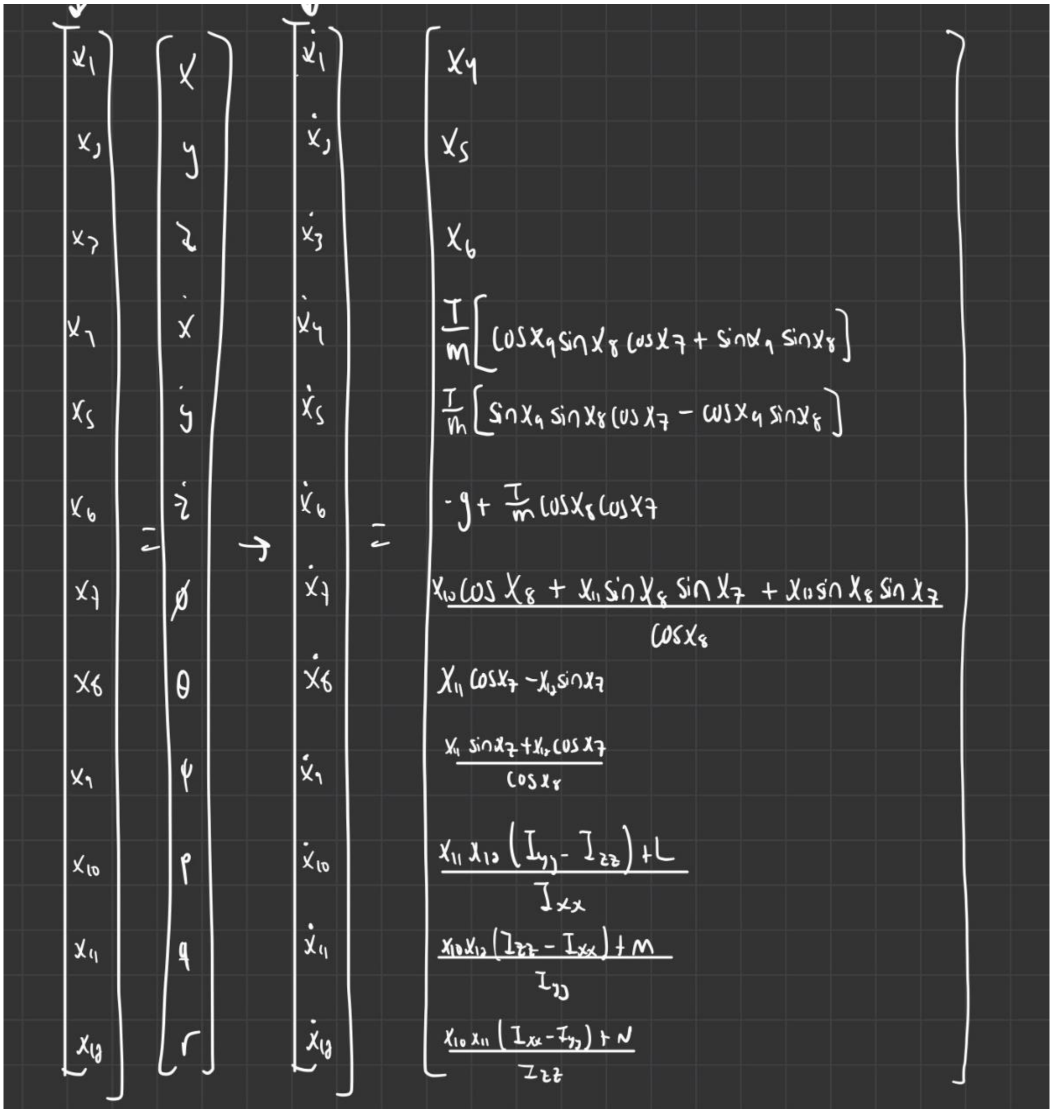
$$= \langle -qrI_{zz} + qrI_{yy},\ prI_{zz} - prI_{xx},\ -pqI_{yy} + pqI_{xx} \rangle$$

$$= \langle qr(I_{yy} - I_{zz}),\ pr(I_{zz} - I_{xx}),\ pq(I_{xx} - I_{yy}) \rangle$$

$$3 \begin{cases} \dot{p} = \dfrac{qr(I_{yy} - I_{zz}) + L}{I_{xx}} \\[2ex] \dot{q} = \dfrac{pr(I_{zz} - I_{xx}) + M}{I_{yy}} \\[2ex] \dot{r} = \dfrac{pq(I_{xx} - I_{yy}) + N}{I_{zz}} \end{cases}$$

Figure 3. Time derivatives of the body angular rates

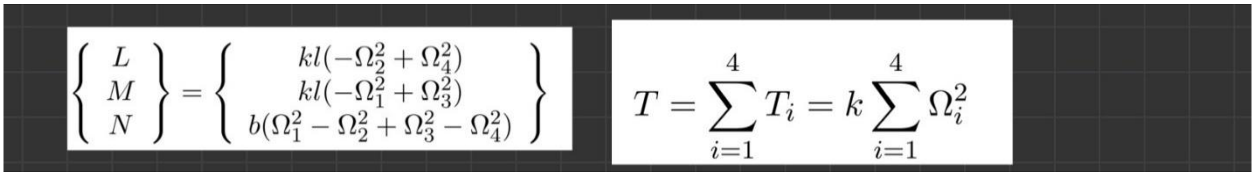
Finally, the three above derivations, in addition to the considered x,y,z values, a large state space was formed. As seen in Figure 4, all of the variables were related to each other.



$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \phi \\ \theta \\ \psi \\ p \\ q \\ r \end{bmatrix} \rightarrow \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \dot{x}_{10} \\ \dot{x}_{11} \\ \dot{x}_{12} \end{bmatrix} = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ \frac{T}{m}\left[\cos x_9 \sin x_8 \cos x_7 + \sin x_9 \sin x_8\right] \\ \frac{T}{m}\left[\sin x_9 \sin x_8 \cos x_7 - \cos x_9 \sin x_8\right] \\ -g + \frac{T}{m}\cos x_8 \cos x_7 \\ \frac{x_{10}\cos x_8 + x_{11}\sin x_8 \sin x_7 + x_{12}\sin x_8 \sin x_7}{\cos x_8} \\ x_{11}\cos x_7 - x_{12}\sin x_7 \\ \frac{x_{11}\sin x_7 + x_{12}\cos x_7}{\cos x_8} \\ \frac{x_{11}x_{12}\left(I_{yy} - I_{zz}\right) + L}{I_{xx}} \\ \frac{x_{10}x_{12}\left(I_{zz} - I_{xx}\right) + M}{I_{yy}} \\ \frac{x_{10}x_{11}\left(I_{xx} - I_{yy}\right) + N}{I_{zz}} \end{bmatrix}$$

Figure 4. State space of the system

The state space can now be used to find the motion of the quadcopter using ODE45. The quadcopter simulation runs between 1 and 6 seconds. Six events occur, each one lasting 1 second each. The only thing that changes in the state space between each event is L, M, N, and T. These depend on the velocities of each of the four rotors. These four forces and moments are calculated using the equations in Figure 5.

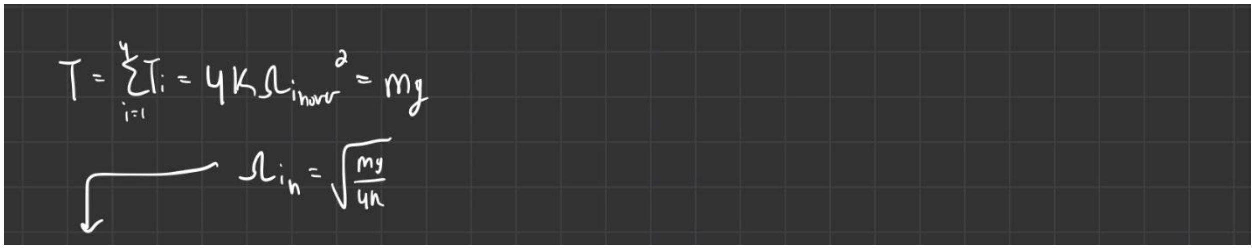


$$\left\{ \begin{array}{c} L \\ M \\ N \end{array} \right\} = \left\{ \begin{array}{c} kl(-\Omega_2^2 + \Omega_4^2) \\ kl(-\Omega_1^2 + \Omega_3^2) \\ b(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{array} \right\}$$

$$T = \sum_{i=1}^{4} T_i = k \sum_{i=1}^{4} \Omega_i^2$$

Figure 5. L, M, N, and T equations

The different angular velocity equations for each event were used to find the angular velocity of each rotor to be used in the equations in Figure 5. Some events only provided angular velocities for two rotors. In this case, the missing angular velocities were assumed to be running at omega hover, which is calculated in Figure 6.

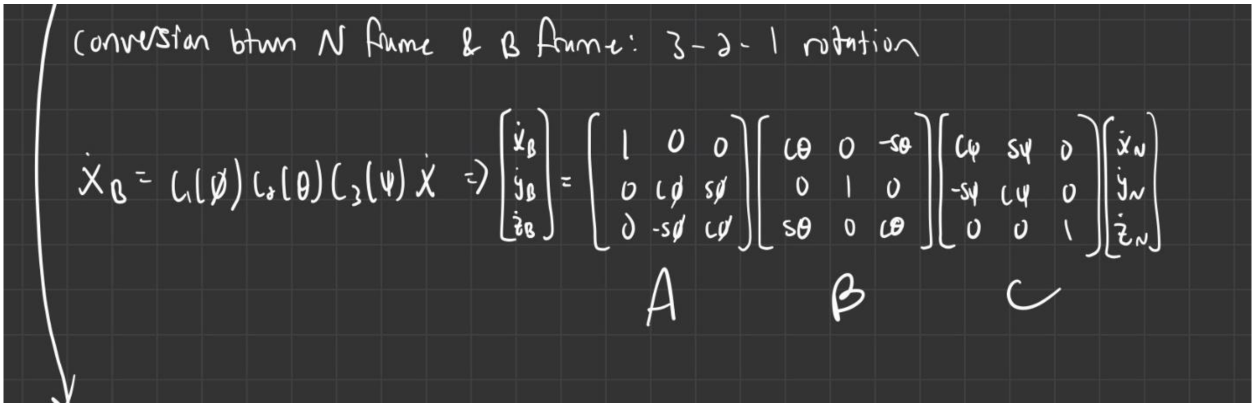


$$T = \sum_{i=1}^{4} T_i = 4K\Omega_{i_{hover}}^{2} = mg$$

$$\Omega_{i_h} = \sqrt{\frac{mg}{4k}}$$

Figure 6. Omega hover derivation

Since the problem statement requires the velocity to be in the body frame, DCMs were utilized to convert the inertial velocity to the body frame. The system had a 3-2-1 rotation, so the DCM sequence is as shown in Figure 5.



conversion btwn N frame & B frame: 3-2-1 rotation

$$\dot{X}_B = C_1(\phi)C_2(\theta)C_3(\psi)\dot{X} \Rightarrow \begin{bmatrix} \dot{x}_B \\ \dot{y}_B \\ \dot{z}_B \end{bmatrix} = \underset{A}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{bmatrix}} \underset{B}{\begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{bmatrix}} \underset{C}{\begin{bmatrix} c\psi & s\psi & 0 \\ -s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}} \begin{bmatrix} \dot{x}_N \\ \dot{y}_N \\ \dot{z}_N \end{bmatrix}$$

Figure 7. DCM for velocity

The initial reference frame position, body velocity, Euler angles, and angular velocity were then graphed verses time for the entire 6 seconds of the simulation.

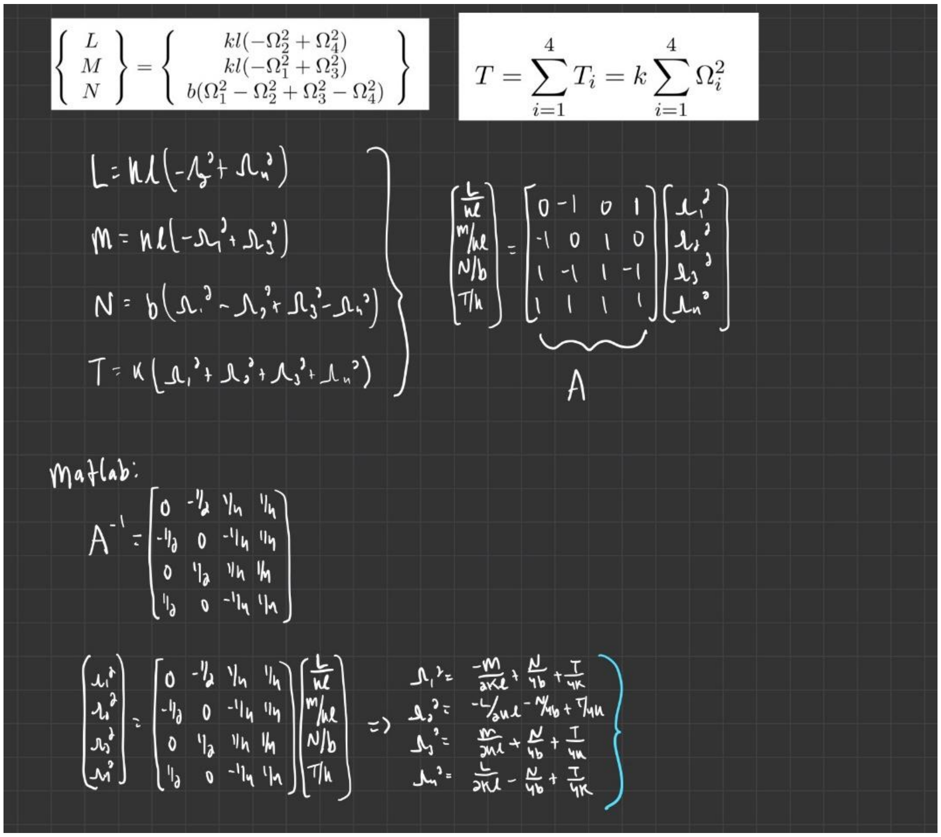
The second part of the problem involved using an autopilot for the quadcopter. T, L, M, and N are now calculated using the equations in Figure 8.

$$
\begin{aligned}
T &= (g + (\dot{z}_r - \dot{z}) + (z_r - z)) \frac{m}{\cos\phi\cos\theta} \\
L &= I_{xx}\left((\dot{\phi}_r - \dot{\phi}) + (\phi_r - \phi)\right) \\
M &= I_{yy}\left((\dot{\theta}_r - \dot{\theta}) + (\theta_r - \theta)\right) \\
N &= I_{zz}\left((\dot{\psi}_r - \dot{\psi}) + (\psi_r - \psi)\right)
\end{aligned}
$$

Figure 8. Autopilot equations

Given initial conditions, and with a desired reference altitude, z_r, of 10m and the rest of the reference values of 0, the state space equation can additionally be used to find the motion of the quadcopter during a 120 second period.

Note that, as per Figure 8, now the state space now does not depend on the angular velocity of the rotors. The angular velocity can still be calculated using the equations in Figure 5. The finalized equations, shown in Figure 9, were used to plot angular velocity over the time frame.



$$
\left\{ \begin{matrix} L \\ M \\ N \end{matrix} \right\} = \left\{ \begin{matrix} kl(-\Omega_2^2 + \Omega_4^2) \\ kl(-\Omega_1^2 + \Omega_3^2) \\ b(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{matrix} \right\}
\qquad
T = \sum_{i=1}^{4} T_i = k \sum_{i=1}^{4} \Omega_i^2
$$

$$
\left.\begin{aligned}
L &= kl(-\Omega_2^2 + \Omega_4^2) \\
M &= kl(-\Omega_1^2 + \Omega_3^2) \\
N &= b(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \\
T &= k(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2)
\end{aligned}\right\}
\qquad
\begin{bmatrix} \frac{L}{kl} \\ M/kl \\ N/b \\ T/k \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix}}_{A} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix}
$$

Matlab:

$$
A^{-1} = \begin{bmatrix} 0 & -1/2 & 1/4 & 1/4 \\ -1/2 & 0 & -1/4 & 1/4 \\ 0 & 1/2 & 1/4 & 1/4 \\ 1/2 & 0 & -1/4 & 1/4 \end{bmatrix}
$$

$$
\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} 0 & -1/2 & 1/4 & 1/4 \\ -1/2 & 0 & -1/4 & 1/4 \\ 0 & 1/2 & 1/4 & 1/4 \\ 1/2 & 0 & -1/4 & 1/4 \end{bmatrix} \begin{bmatrix} \frac{L}{kl} \\ M/kl \\ N/b \\ T/k \end{bmatrix}
\Rightarrow
\left.\begin{aligned}
\Omega_1^2 &= \frac{-M}{2kl} + \frac{N}{4b} + \frac{T}{4k} \\
\Omega_2^2 &= -L/2kl - N/4b + T/4k \\
\Omega_3^2 &= \frac{M}{2kl} + \frac{N}{4b} + \frac{T}{4k} \\
\Omega_4^2 &= \frac{L}{2kl} - \frac{N}{4b} + \frac{T}{4k}
\end{aligned}\right\}
$$

Figure 9. Derivation of autopilot omega equations

Again, initial reference frame position, body velocity, Euler angles, and angular velocity were then graphed verses time for the entire duration seconds of the simulation. In addition, the angular velocities were graphed over time as well.

Discussion of Results:

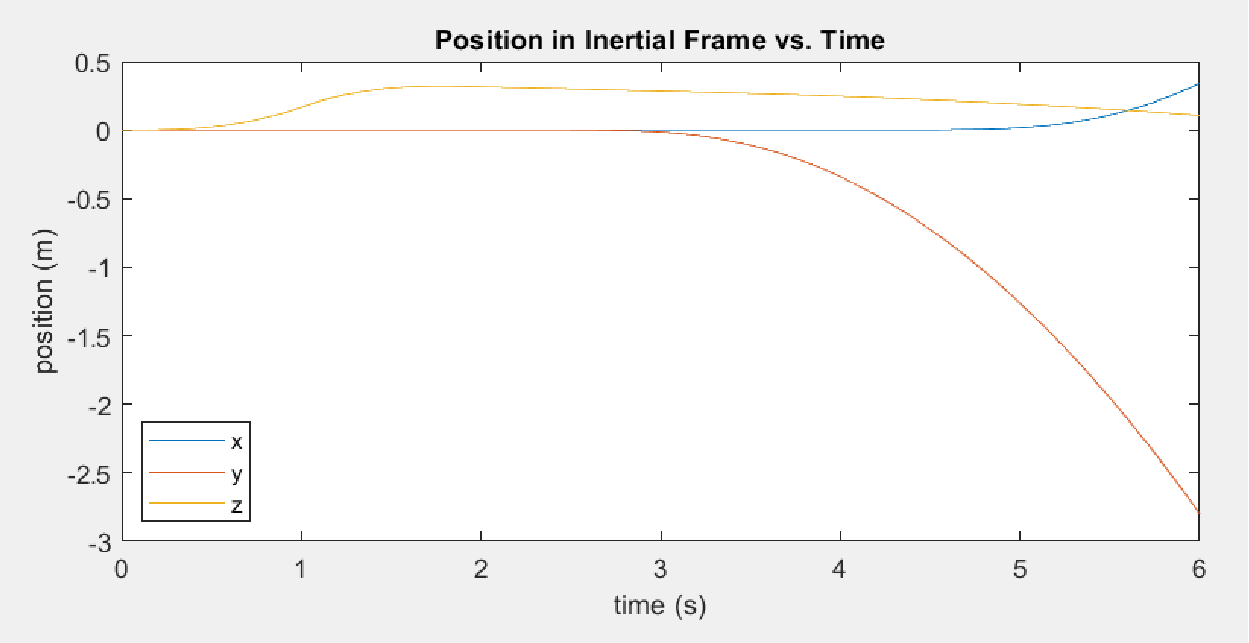


Figure 10: Position vs. Time

The z position of the graph increases rapidly in the first few seconds. This was the time that the drone was ascending so this is as expected. Then eventually it levels off and just steadily decreases, this is likely because the velocity in the z-direction goes to zero over this time. The y-position changes because of the roll motion and the x changes because of the pitch motion. The pitch motion was done in the last second so the x position did not change until t = 5s.

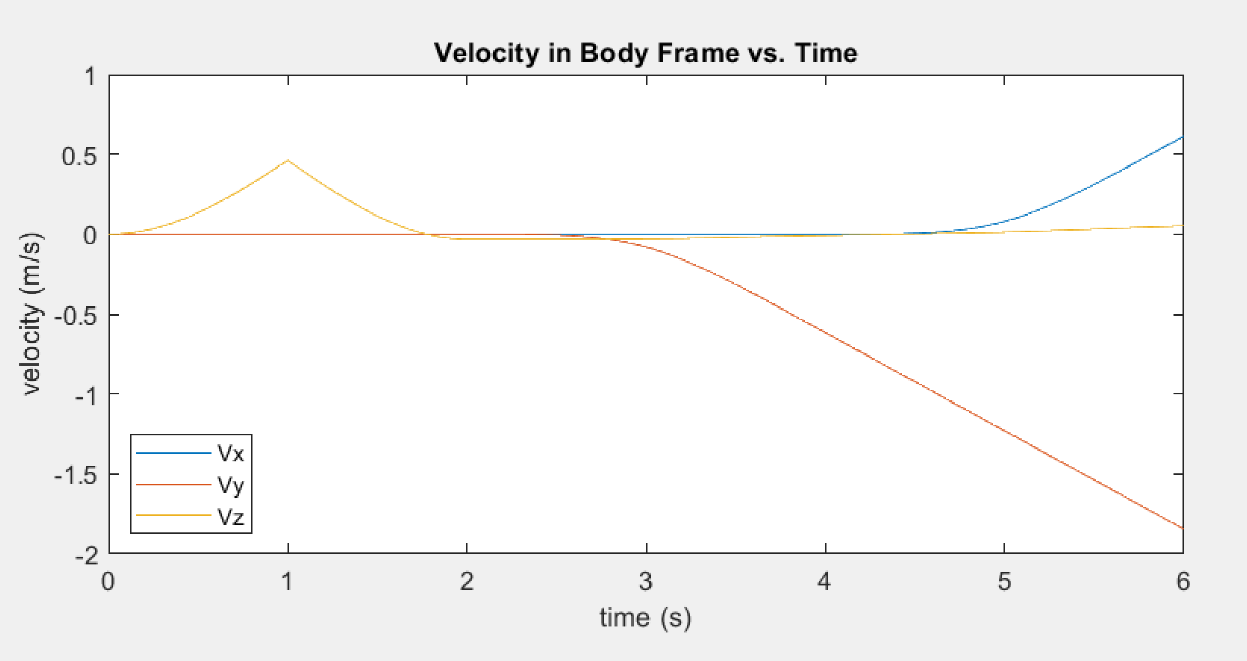


Figure 11: Velocity vs. Time

The velocities follow a similar pattern to their respective positions as expected. The z-velocity spikes to the hover point and then decreases and hovers around zero. This decrease in velocity will explain the decrease in the z-position. Vy and Vx both increase during the respective times of their motions which is expected because this is what their positions do as well.

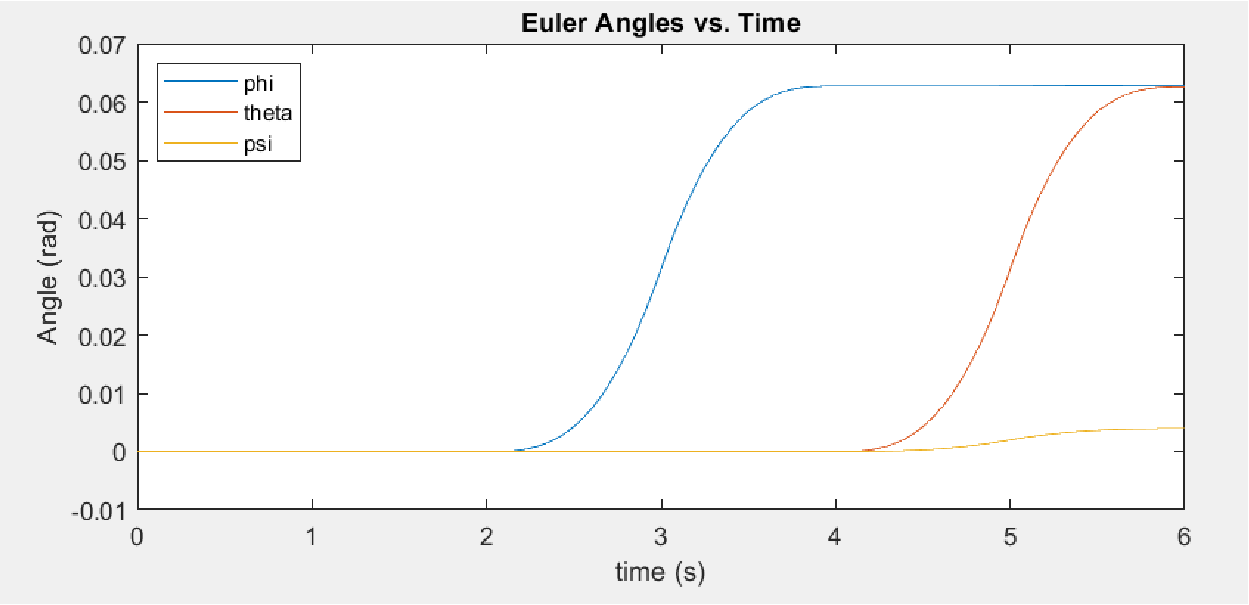


Figure 12: Angle vs. Time

The angles will change as the different pitch and roll moments are performed by the quadcopter. The angles of the quadcopter will start at zero and increase over time until it stops at an angle where it can continue its motion. This is shown in the graphs above where the quadcopter starts at an angle of zero radians, increases beginning at 2 seconds for phi and 4 seconds for theta, and then stops at an angle of .065 radians at 4 seconds for phi and 6 seconds for theta.

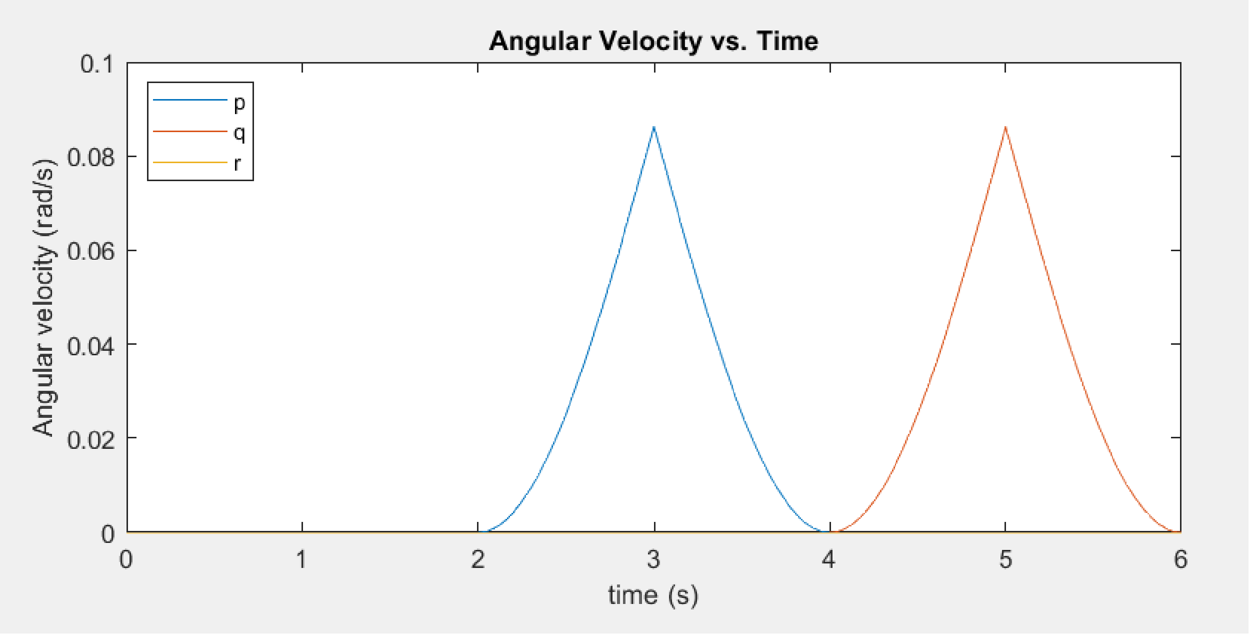


Figure 13: Angular Velocity vs. Time

The angular velocity graphs were as expected, following the same pattern as the position graphs. The angular velocities increase to increase the angle and then decrease to decrease it to a final position where the velocity becomes zero. There is a spike around 3 and 5 seconds which is where the angular rates began to transition from acceleration to deceleration.

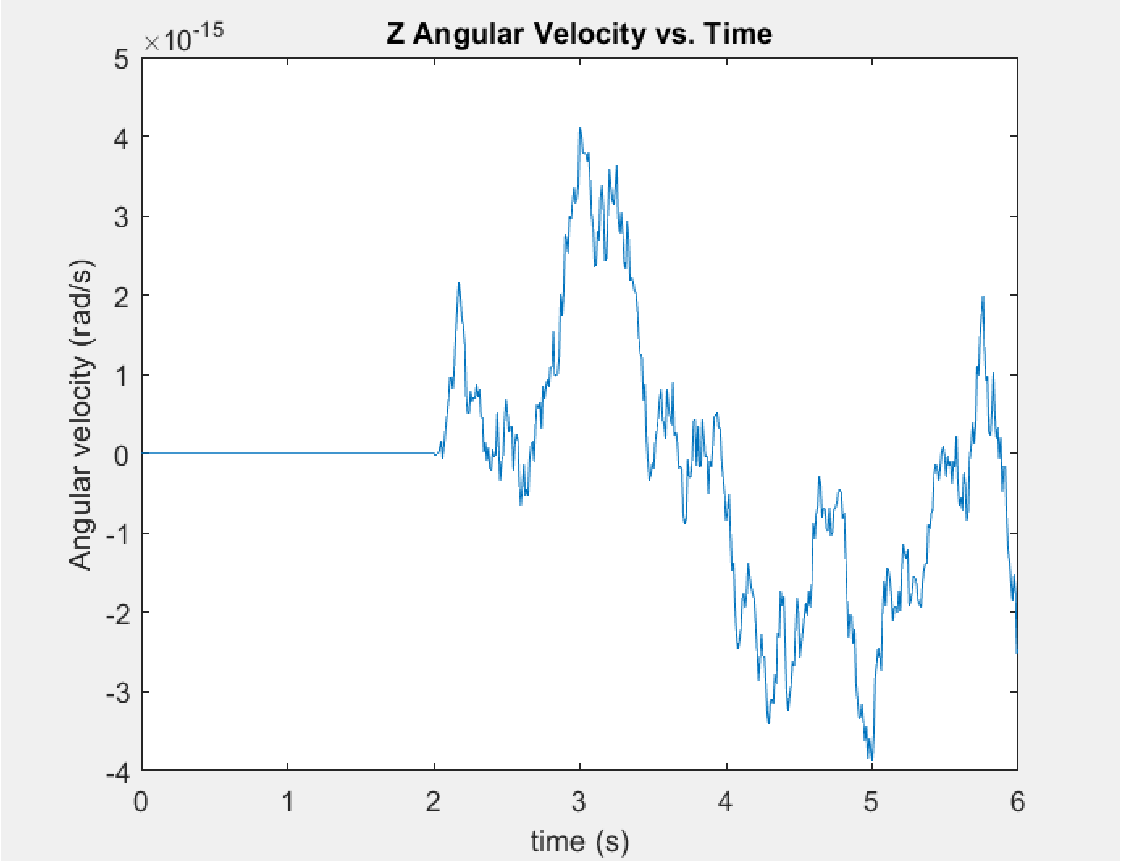


Figure 14: Error in vibrations

This graph displays some type of error. This error can come from noise due to small vibrations in the motion of the drone.

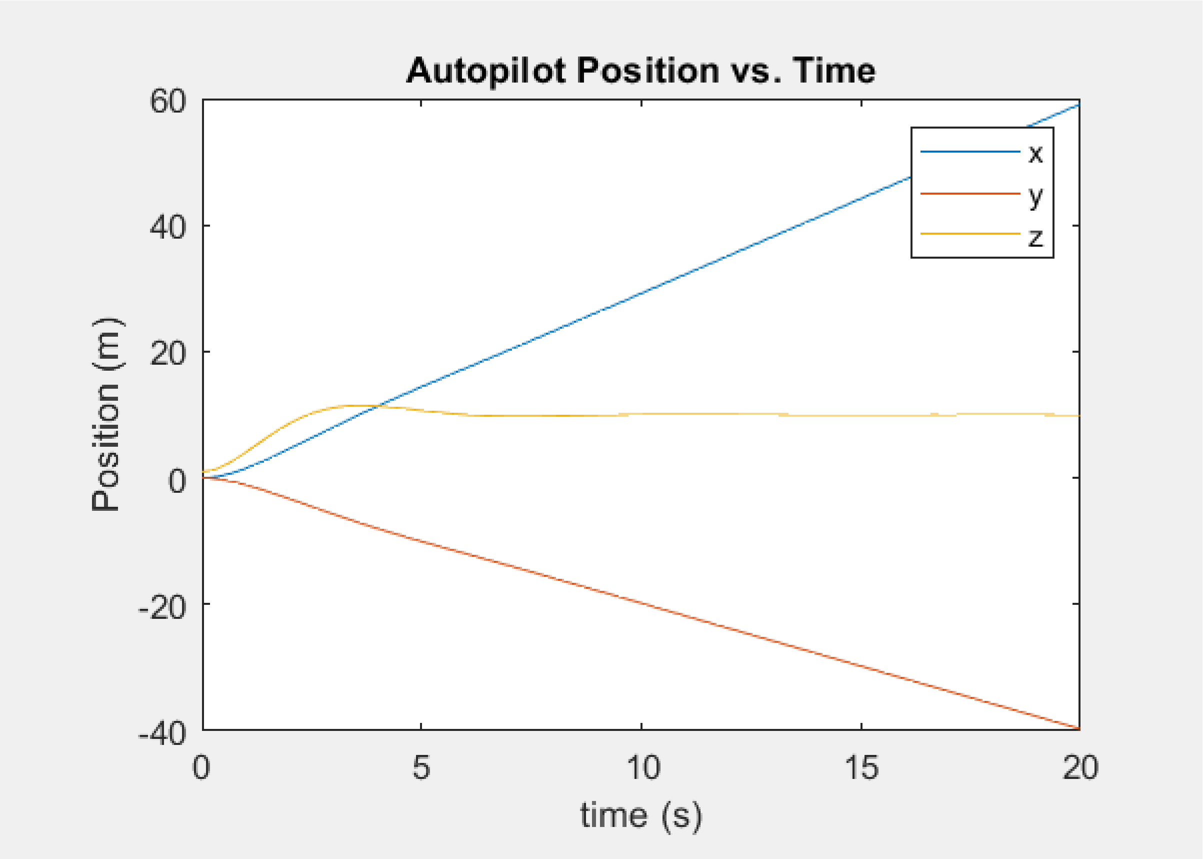


Figure 15: Autopilot Position vs. Time

The autopilot positions in the first 20 seconds shoot up linearly in the positive x-direction and negative y-direction. The z-direction gets to a peak and then decreases into a steady position. The autopilot found values of x and y that gave a steady state z but no longer as a reference to x and y so those directions increase.

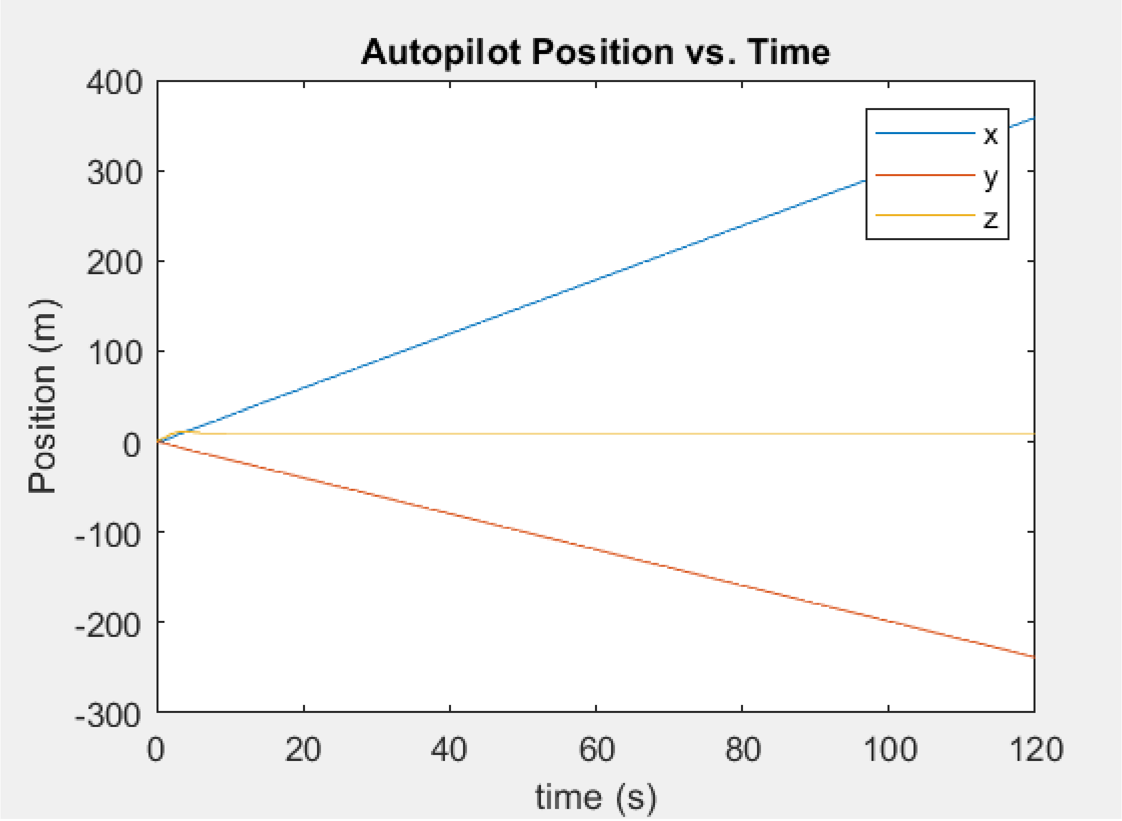


Figure 16: Autopilot Position vs. Time for 120 seconds

This is the autopilot position for 120 seconds. This is consistent with the previous graph where the drone stays at 10 meters for the entire duration. Both x and y are still shooting off but this is something that the autopilot cannot control.

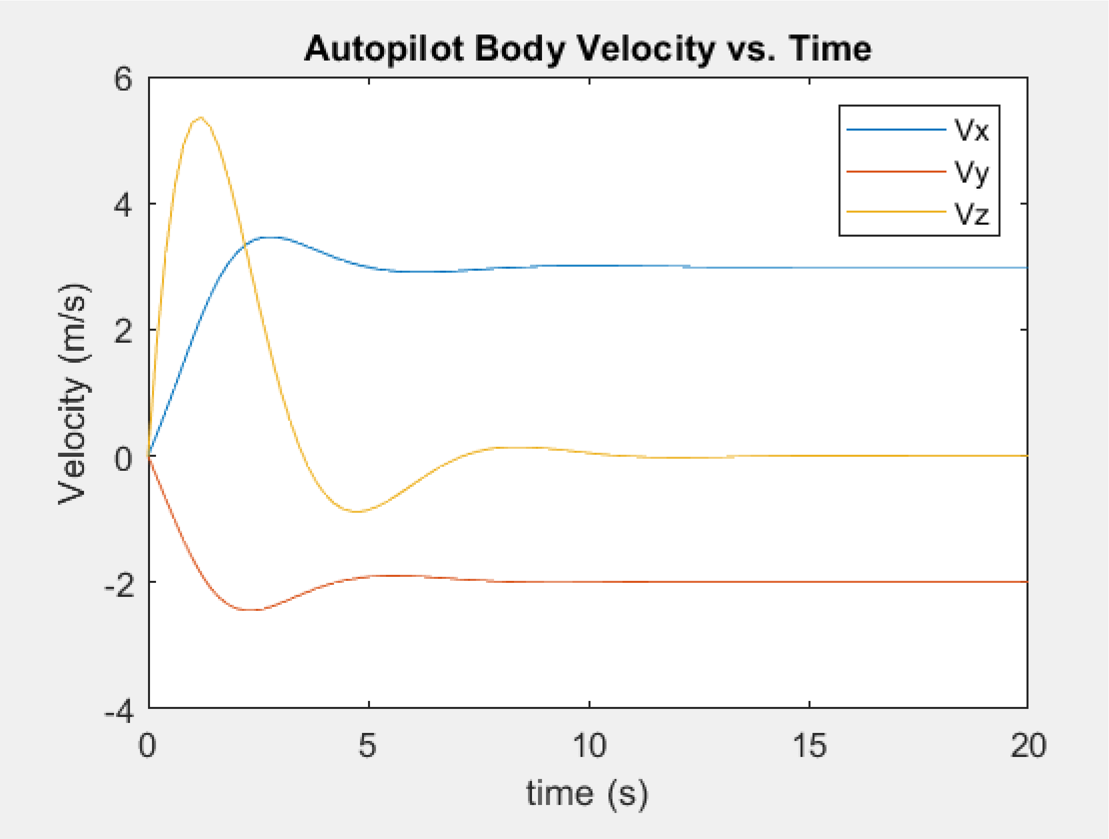


Figure 17: Body Velocity vs. Time

All velocities oscillate and then stabilize. The z-velocity stabilizes at 0 m/s which is expected since the position has to stay the same. The slopes of the position graphs are linear so it is expected that they will stabilize at a constant non-zero velocity which is shown in the figure. The z-velocity peaks much higher than the others. This is probably because the drone is reaching that z-position faster than the other components.

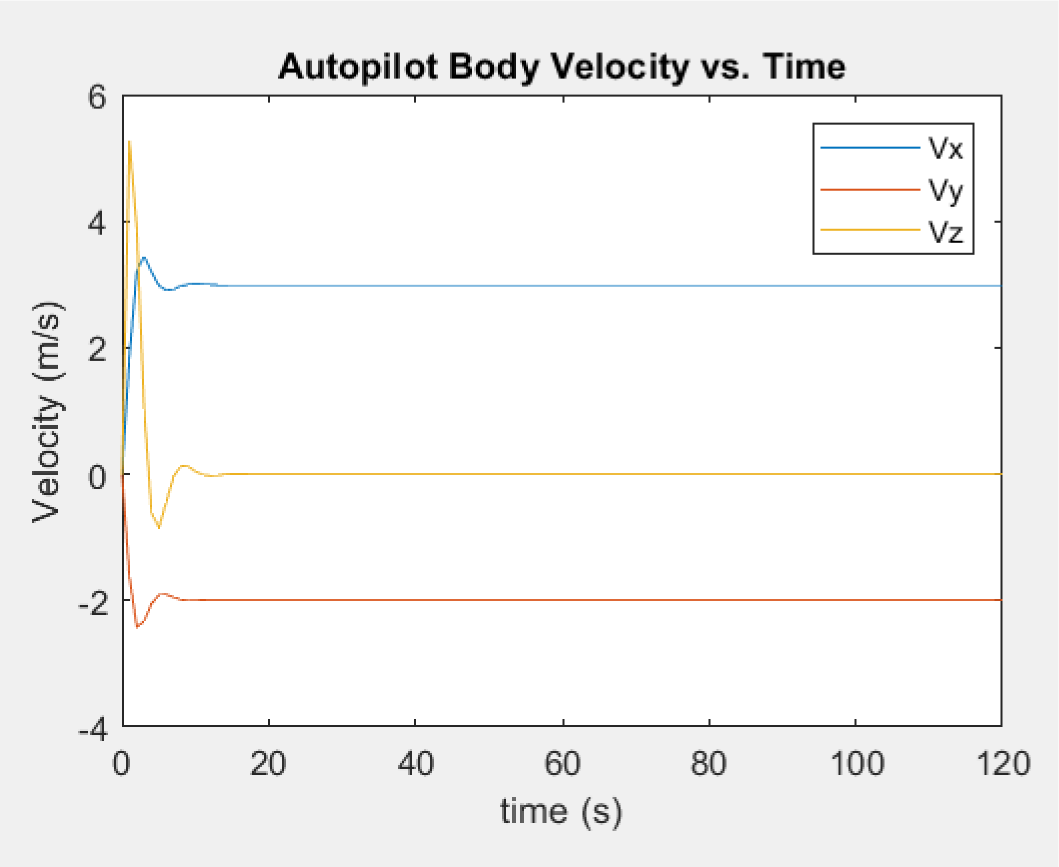


Figure 18: Autopilot Body Velocity vs. Time

The velocities for 120 seconds are shown in the figure. These are consistent with the previous graph and stabilize completely over 120 seconds.

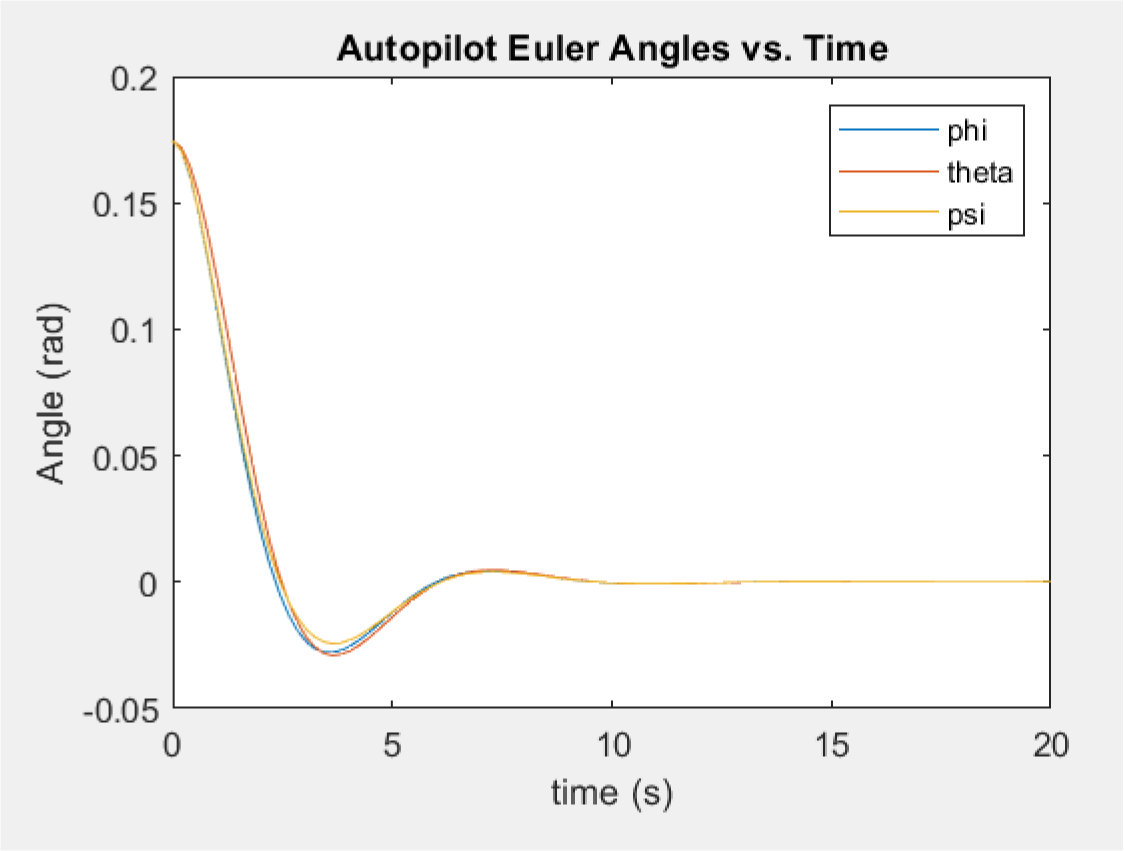


Figure 19: Autopilot Euler Angles vs. Time

The Euler Angles phi, theta, and psi all stay somewhat equal through the entire time period. They start at some angle and then oscillate until they are steady at 0 angle. This is expected since eventually it will hover without moving.

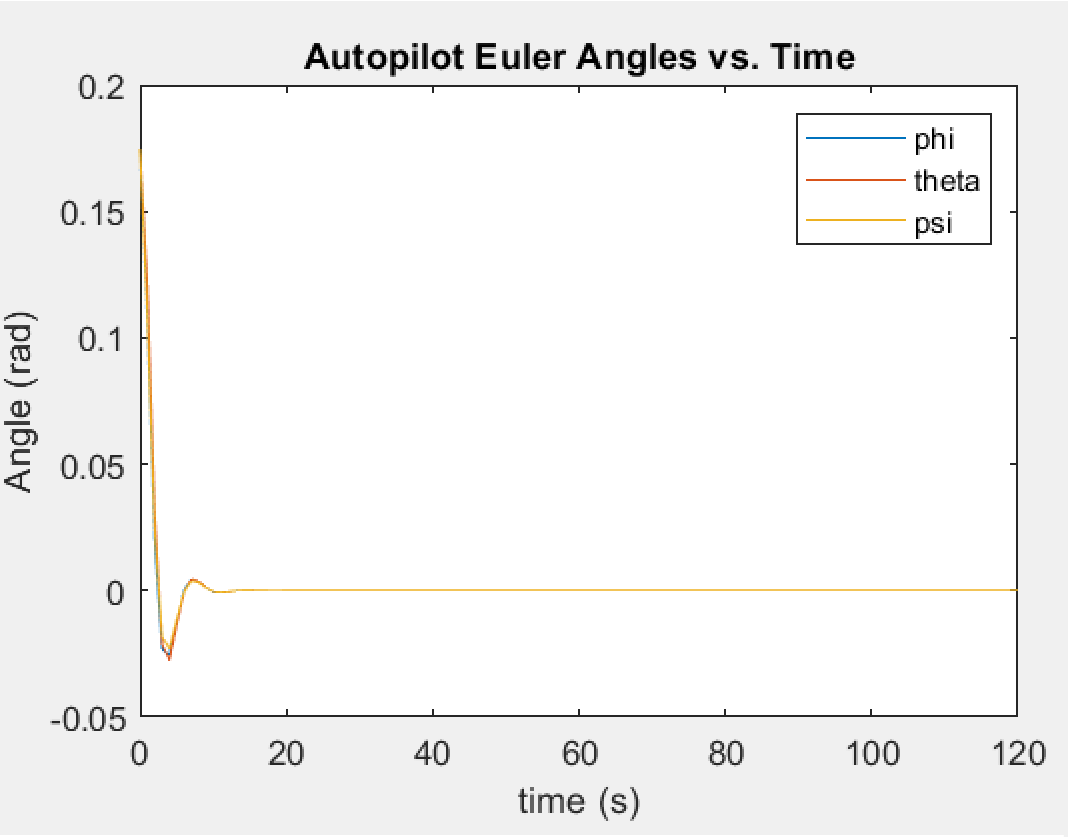


Figure 20: Autopilot Euler Angles vs. Time

The angles and the time are still consistent with the 20 second graph.
Eventually the angles in all directions stabilize at zero for the remainder of the time.

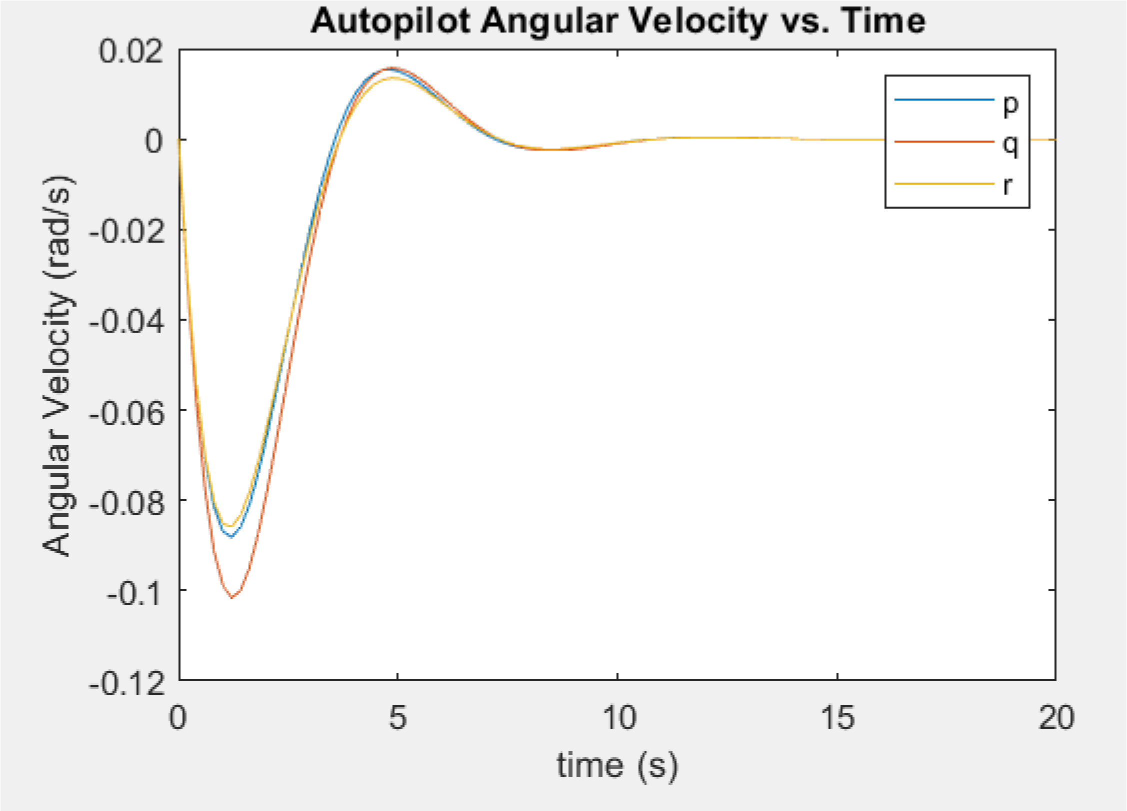


Figure 21. Autopilot Angular Velocity vs. Time

Figure 21 is a closer look at the first 20 seconds of the simulation. All of the angular velocities dip into the negatives to correct the positive angles that were a part of the initial conditions. The p and r values hit a minimum of about -.085 rad/s while the q value dips a little lower to about -.1 rad/s. This is likely to correct for the overshoot from an already biased initial rotation. All values hit a steady state value of 0 rad/s at around 20 seconds.

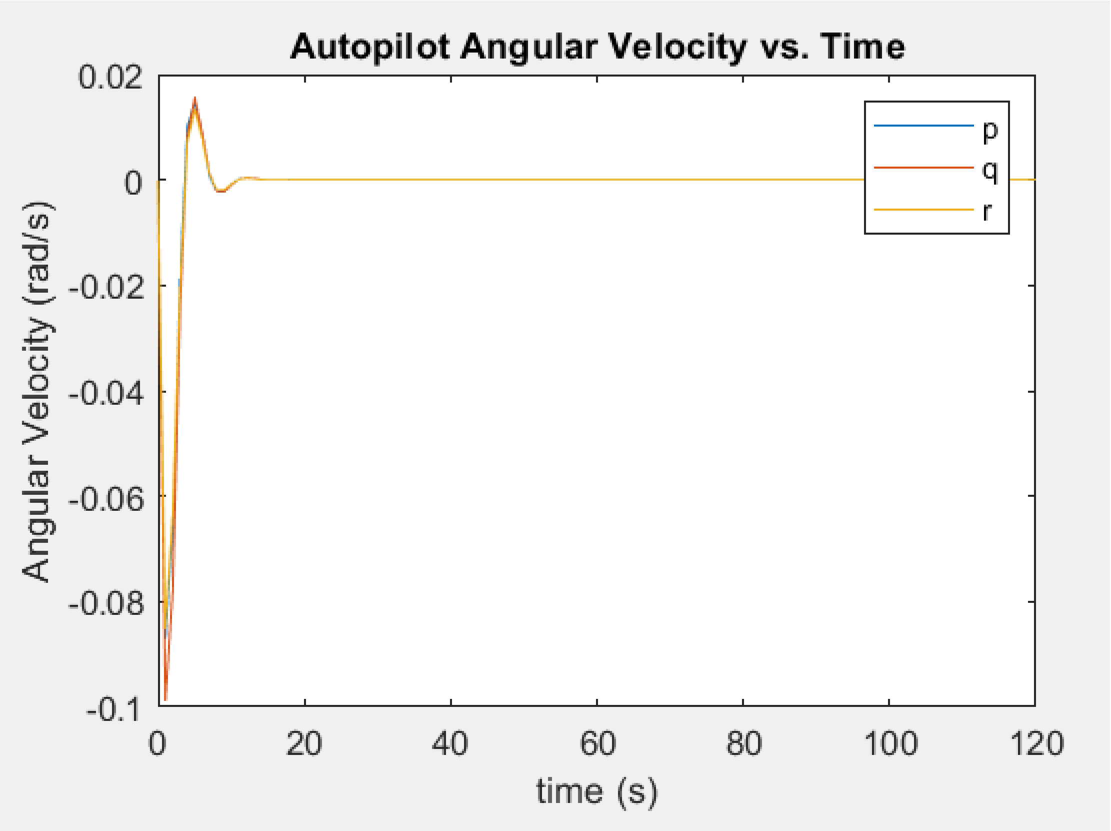


Figure 22. Autopilot Angular Velocity vs. Time

Figure 22 shows the angular velocity of the quadcopter over the entire simulation time.

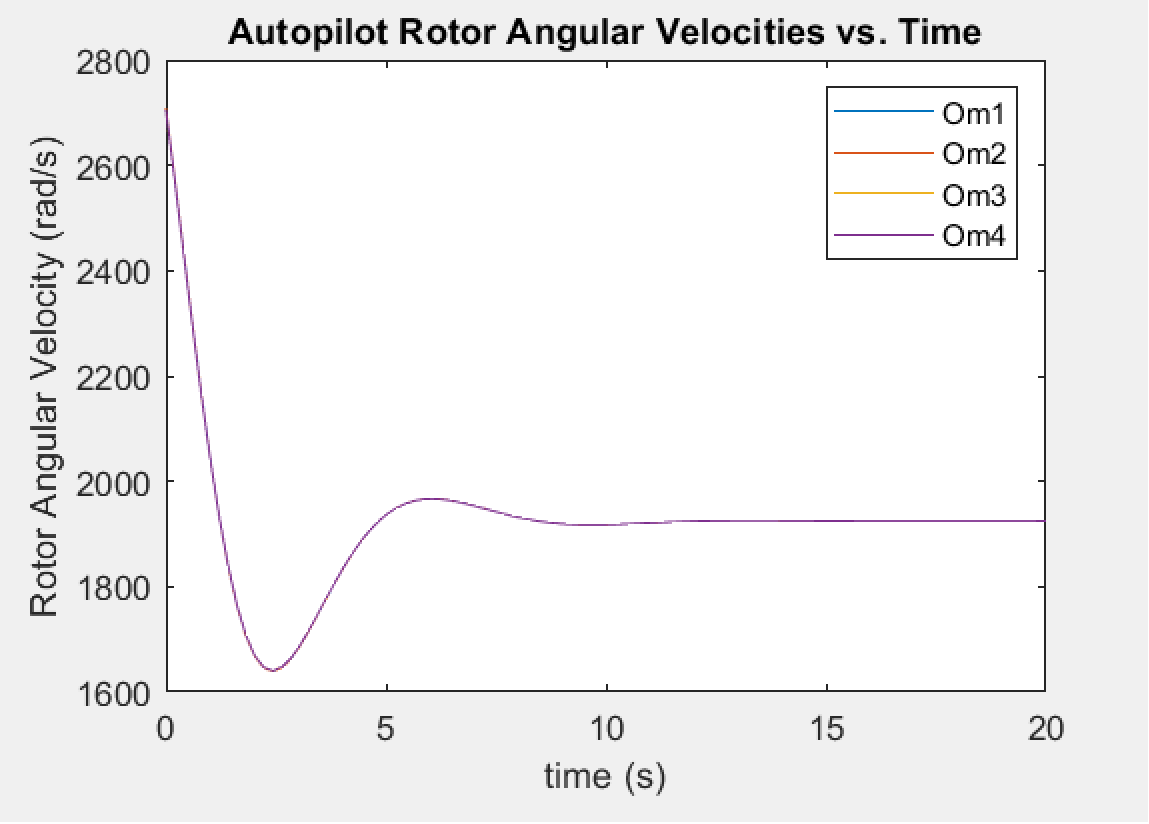


Figure 23. Autopilot Rotor Angular Velocity vs. Time

Figure 23 is a closer look at the first 20 seconds of the simulation. All of the angular velocities dip into the negatives to correct the positive angles that were a part of the initial conditions. The p and r values hit a minimum of about -.085 rad/s while the q value dips a little lower to about -.1 rad/s. This is likely to correct for the overshoot from an already biased initial rotation. All values hit a steady state value of 0 rad/s at around 20 seconds.

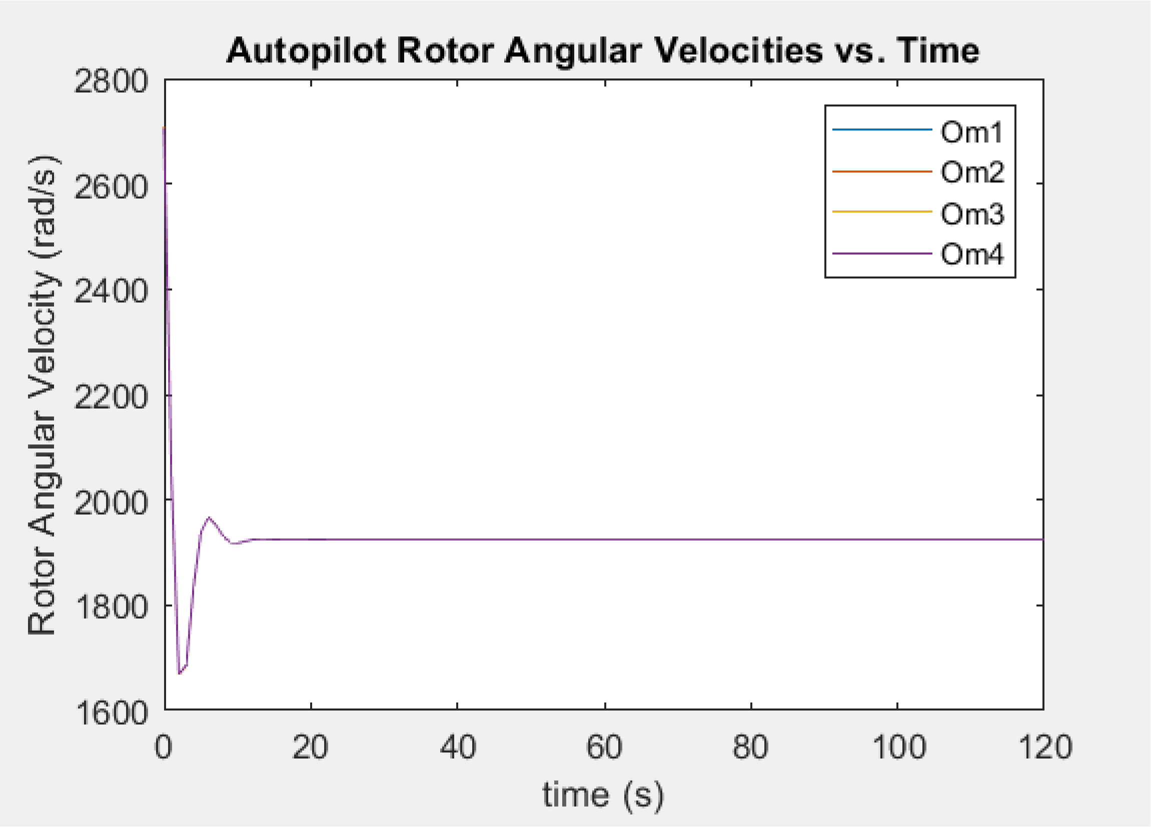


Figure 24. Autopilot Rotor Angular Velocity vs. Time

Figure 24 shows the angular velocity of the quadcopter over time. It stabilizes very early in the process.