ECE 3111

Control Design & Simulation of Quadcopter Dynamics

Alexander Gmuer

December 16, 2020

Introduction

The quadcopter is one of the more well known and useful aerial drones, and its use only continues to rise. Instrumental to this has been computer simulation tools, like MATLAB and Simulink, which improve the efficiency and design of the quadcopter as a control system. The goal of this paper is to document the development of a quadcopter system from first principles, and then demonstrate the system response when the quadcopter takes off from the ground and stabilizes at a height of 5 meters.

Linearization

Given the diagram shown above of the quadcopter in space, non-linear equations of motion can be derived which describe the mechanics of the system.

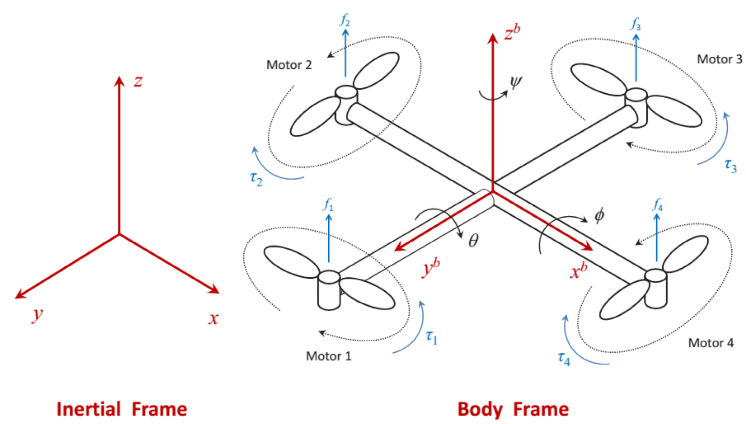

Figure 1: The Quadcopter inertial reference frame

These equations relate the square voltage motor inputs u_1, u_2, u_3, and u_4 to the positions, velocities, and accelerations in the x, y, and z directions - as well as the pitch (ϕ), roll (θ), and yaw (ψ) of the quadcopter with respect to its inertial reference frame:

$$\begin{aligned}
\dot{x} &= v_x \\
\dot{y} &= v_y \\
\dot{z} &= v_z \\
\dot{v_x} &= -\frac{k_d}{m}v_x + \frac{kc_m}{m}(\sin\psi\sin\phi + \cos\psi\cos\phi\sin\theta)(u_1+u_2+u_3+u_4) \\
\dot{v_y} &= -\frac{k_d}{m}v_y + \frac{kc_m}{m}(\cos\phi\sin\psi\sin\theta - \cos\psi\sin\phi)(u_1+u_2+u_3+u_4) \\
\dot{v_z} &= -\frac{k_d}{m}v_z - g + \frac{kc_m}{m}(\cos\theta\cos\phi)(u_1+u_2+u_3+u_4) \\
\dot{\phi} &= \omega_x + \omega_y(\sin\phi\tan\theta) + \omega_z(\cos\phi\tan\theta) \\
\dot{\theta} &= \omega_y(\cos\phi) + \omega_z(-\sin\theta) \\
\dot{\psi} &= \omega_y(\sin\phi/\cos\theta) + \omega_z(\cos\phi/\cos\theta) \\
\dot{\omega_x} &= \frac{Lkc_m}{I_{xx}}(u_1-u_3) - \left((\frac{I_{yy}-I_{zz}}{I_{xx}}\right)\omega_y\omega_z \\
\dot{\omega_y} &= \frac{Lkc_m}{I_{yy}}(u_2-u_4) - \left(\frac{I_{zz}-I_{xx}}{I_{yy}}\right)\omega_x\omega_z \\
\dot{\omega_z} &= \frac{Lkc_m}{I_{zz}}(u_1-u_2+u_3-u_4) - \left(\frac{I_{xx}-I_{yy}}{I_{zz}}\right)\omega_x\omega_y
\end{aligned}$$

In order to create a robust model which is able to be analyzed by typical control methods, the equations of motion above must be linearized. We shall consider the system equilibrium point at which $x = y = z = 0m$, and $\phi = \theta = \psi = 0°$. The equations of motion can then be put in a linear form:

$$\begin{aligned}
\dot{x} &= v_x \\
\dot{y} &= v_y \\
\dot{z} &= v_z \\
\dot{v_x} &= -\frac{k_d}{m}v_x + \\
\dot{v_y} &= -\frac{k_d}{m}v_y + \\
\dot{v_z} &= -\frac{k_d}{m}v_z - g + \frac{kc_m}{m}(u_1+u_2+u_3+u_4)
\end{aligned}$$

$$\begin{aligned}
\dot{\phi} &= \omega_x \\
\dot{\theta} &= \omega_y \\
\dot{\psi} &= \omega_z(\cos\phi/\cos\theta) \\
\dot{\omega_x} &= \frac{Lkc_m}{I_{xx}}(u_1-u_3) \\
\dot{\omega_y} &= \frac{Lkc_m}{I_{yy}}(u_2-u_4) \\
\dot{\omega_z} &= \frac{Lkc_m}{I_{zz}}(u_1-u_2+u_3-u_4)
\end{aligned}$$

Having now created a list of approximate linear equations which describe the quadcopter system near equilibrium, it can now be modeled as a Multi-Input, Multi-Output (MIMO) system using state space representation in the time domain.

Modeling the System: State Space & Transfer Function

Before constructing the state space quadcopter model, it is important to remind ourselves of the desired inputs and outputs for this system. This is a matter of design and perspective, as most modern systems are complex, with engineering teams often only responsible for a single subsystem. In the case of the quadcopter, our model is designed to take squared voltages for each of the four motors as inputs, then use those to control the attitude, (roll, pitch, and yaw) as well as the altitude and and position.

Looking at the linearized equations of motion, we see that u_1, u_2, u_3, and u_4 are the input motor voltages, and the outputs are x, y, z, ϕ, θ, and ψ. This means that this system will have 4 inputs and 6 outputs. In the state space model, this means that the input $\mathbf{u}$ vector will contain the squared motor voltages u_1, u_2, u_3, and u_4 - while the output $\mathbf{y}$ vector will contain the x, y, z, ϕ, θ, and ψ state variables. With this, as well as the linearized equations of motion, we can construct the state space representation:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$$
$$\mathbf{y} = C\mathbf{x} + D\mathbf{u}$$

Where the matrices A and B are given by:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{-kd}{m} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{-kd}{m} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{-kd}{m} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{kc_m}{m} & \frac{kc_m}{m} & \frac{kc_m}{m} & \frac{kc_m}{m} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{Lkc_m}{I_{xx}} & 0 & -\frac{Lkc_m}{I_{xx}} & 0 \\ 0 & \frac{Lkc_m}{I_{yy}} & 0 & -\frac{Lkc_m}{I_{yy}} \\ \frac{Lbc_m}{I_{zz}} & -\frac{Lbc_m}{I_{zz}} & \frac{Lbc_m}{I_{zz}} & -\frac{Lbc_m}{I_{zz}} \end{bmatrix}$$

While the D matrix is zero, and C is given by:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

The matrix dimensions of A, B, and C are thus:

$$\begin{aligned} A &: 12 \times 12 \\ B &: 12 \times 4 \\ C &: 6 \times 12 \end{aligned}$$

Using the matrix equation $T_{ij}(s) = C_i(sI - A)^{-1}B_j + D$, a 6×4 transfer function matrix T can be calculated, such that:

$$\mathbf{y} = T\mathbf{u}$$

Where we have T to be:

$$T = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{3}{25s(2s+1)} & \frac{3}{25s(2s+1)} & \frac{3}{25s(2s+1)} & \frac{3}{25s(2s+1)} \\ \frac{3}{2s^2} & 0 & -\frac{3}{2s^2} & 0 \\ 0 & \frac{3}{2s^2} & 0 & -\frac{3}{2s^2} \\ \frac{1}{40s^2} & -\frac{1}{40s^2} & \frac{1}{40s^2} & -\frac{1}{40s^2} \end{bmatrix}$$

Since the first two rows of the transfer function matrix are zero, and correspond to the x and y responses, for the later parts we will consider T as a 4×4 matrix composing of the transfer functions for the z, ϕ, θ, and ψ variables.

Control Conversion

Using the relation $\omega_i^2 = c_m u_i$ for $i = 1, 2, 3, 4$, the input squared voltages u_i can be related to the squared angular velocities of the corresponding motors. The angular velocities of the motors can be further related to the z, ϕ, θ, and ψ inputs via the following mapping:

$$\begin{bmatrix} u_z \\ u_\phi \\ u_\theta \\ u_\psi \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & -0.5 & 0 & 0 \\ 1 & 0 & -0.5 & -0.5 \\ 0 & 0 & 0.5 & -0.5 \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}$$

Using this mapping matrix M, the transfer function matrix T can be converted so that it can take inputs u_z, u_ϕ, u_θ, and u_ψ through the following steps:

$$\begin{aligned}
\mathbf{y} &= T\mathbf{u} \\
\mathbf{u'} &= Mc_m\mathbf{u} \\
\implies \mathbf{u} &= M^{-1}\frac{1}{c_m}\mathbf{u'} \\
\implies \mathbf{y} &= TM^{-1}\frac{1}{c_m}\mathbf{u'}
\end{aligned}$$

Where $\mathbf{u'} = [u_z\ u_\phi\ u_\theta\ u_\psi]^T$. This transformation results in the following transfer function matrix:

$$H = \begin{bmatrix} \frac{3}{125000s(2s+1)} & 0 & 0 & 0 \\ 0 & -\frac{3}{20000s^2} & -\frac{3}{20000s^2} & 0 \\ 0 & \frac{3}{20000s^2} & \frac{3}{20000s^2} & -\frac{3}{10000s^2} \\ 0 & \frac{1}{20000s^2} & 0 & \frac{1}{20000s^2} \end{bmatrix}$$

Taking the transfer functions along the diagonal entries, we can attain a set of open loop transfer functions which can model the four control variables:

$$\begin{aligned}
\frac{Z(s)}{U_z(s)} &= \frac{3}{125000s(2s+1)} \\
\frac{\phi(s)}{U_\phi(s)} &= -\frac{3}{20000s^2} \\
\frac{\theta(s)}{U_\theta(s)} &= \frac{3}{20000s^2} \\
\frac{\psi(s)}{U_\psi(s)} &= \frac{1}{20000s^2}
\end{aligned}$$

PID Control Design & Analysis

Figures 5-7 show the Bode plots of the four transfer functions from MATLAB. It can be seen that the magnitude of all the transfer functions is less than 0dB when the phase plots cross 180°, so each subsystem is stable for any input gain desired. Since the phase and gain margins are infinite, it shows the systems are either stable or marginally stable. Consulting the root locus plots in Figures 8-11 confirm this, as all loci sketched are either in the left half of the imaginary plane - indicating negative real part and system stability - or they are on the imaginary axis - indicating marginal stability.
The next step of the system design involves determining how P, PI, and PID controllers could influence the stability of the system. Luckily, since all of the above four transfer functions

have either one or two poles at the origin, as shown in the root loci, they are each either Type 1 or Type 2 systems - which means that in the absence of any proportional gain, the system as a whole is stable or marginally stable in response to a standard step input. By adding a proportional gain or P controller, the system would be stable for the z translation, but it will still be marginally stable in terms of the attitude angles. Therefore a P controller would be sufficient to stabilize the Z translation subsystem, however a PI controller will be necessary for stability for attitudinal stability of the quadcopter.
Despite the fact that a P and PI controller are all that is necessary for stability in our system, I opted to design the system with a PID controller in order to control the transient response more precisely, as well as to eliminate steady state error in the system response. Adding an integrating I term to the system allows for the system to handle steady state error, while adding the derivative D term allows the ability of controlling the settling time, as well as percent overshoot of the system. In short, utilizing a PID controller is good for systems which have specific parameter and transient response requirements.
The block diagram of the system, with PID controllers added for each of the four transfer functions, can be seen below:

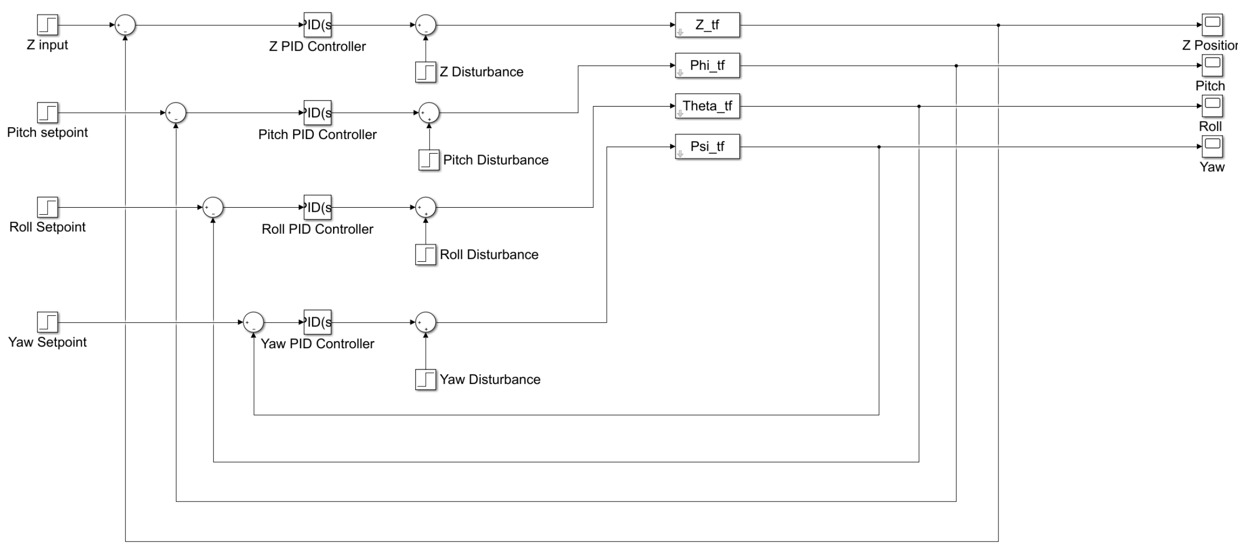

Figure 2: A Block Diagram of the Quadcopter System in Simulink

PID Design

The table below displays the gain factors associated with the PID controllers in for the Z translation, Pitch, Roll, and Yaw subsystems:

	Z Translation	Pitch	Roll	Yaw
K_p	117450	-19618	21511	707609
K_i	21317	-3656	4198	144602
K_d	161472	-25848	27066	850263

Using these gain values, the system responses were graphed versus time. In the case of the Z translation, the system was provided a step input of magnitude 5m, which allows us to see the system response starting at ground level (0 meters) and rising then stabilizing at 5 meters. The pitch, roll, and roll responses were given as inputs the desired 0° angles, then given a disturbance step input.

Closing Thoughts

The quadcopter model created using the methods of this paper is effective based on the transient responses produced in response to step inputs. However, there are a couple of important drawbacks which may impact the implementation of such a system.

First, while the gain factors used in the PID controller are acceptable for use in simulation, it may cause issues when it comes to a physical device. In the case of a quadcopter, having such large amplification factors is likely not feasible, due to the large power amplifiers one would need in order to achieve them. These would likely add significant extra weight to the quadcopter, and so in a realistic implementation it may be wise to sacrifice the very low settling times I achieved for the pitch, roll, and yaw controls in order to not add significant power and weight demands.

Another limitation of this design is the lack of control in the x and y directions. This was largely a matter of simplifying the control architecture, since movement in the z direction combined with stabilization while hovering is difficult in its own right. However, The model does not allow for adjustments back to equilibrium in these directions, due to the transfer functions evaluating to be zero based on the linearized equations of motion. If x and y were also considered as control variables, the system would become under-actuated. This is due to the fact that only 4 inputs (motor voltages/speeds) would need to control 6 outputs.

Further developments on this system could fix these shortcomings and enhance the applicability of the model to real quadcopters, however this model has been effective in the scope that it was designed.

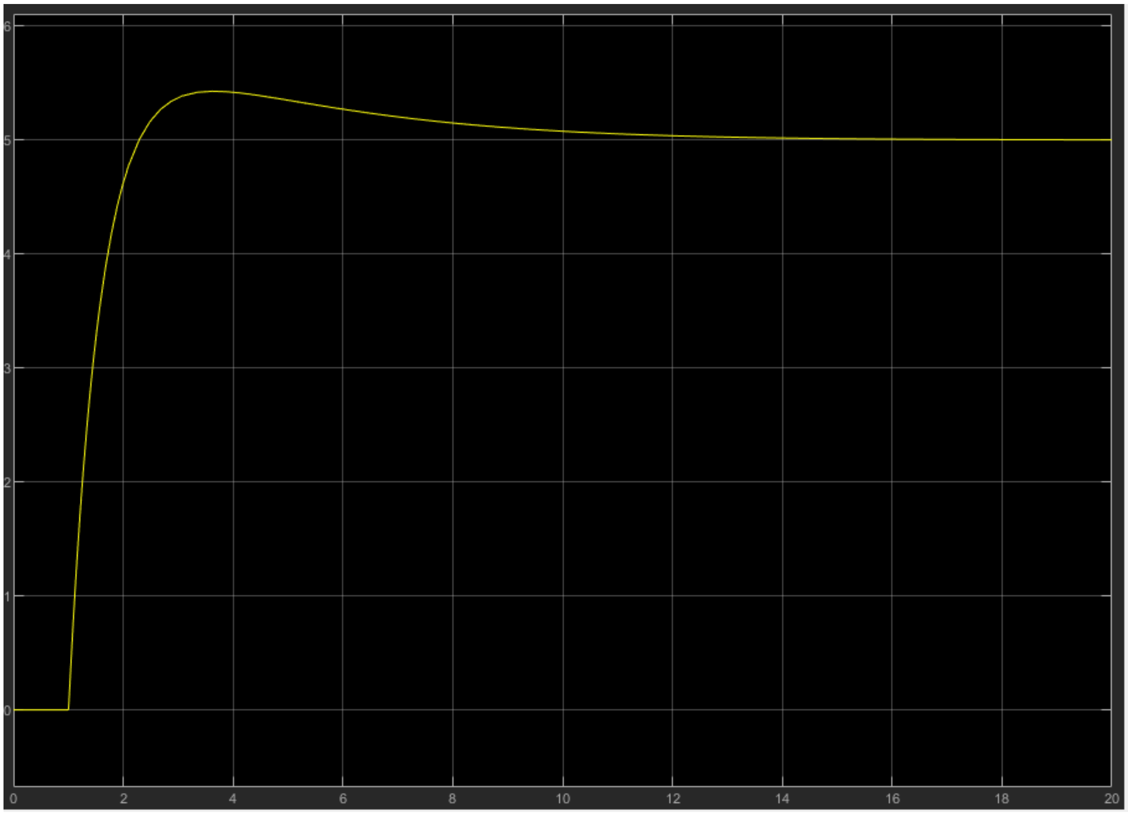

Figure 3: Z Response to Step input of magnitude 5 meters

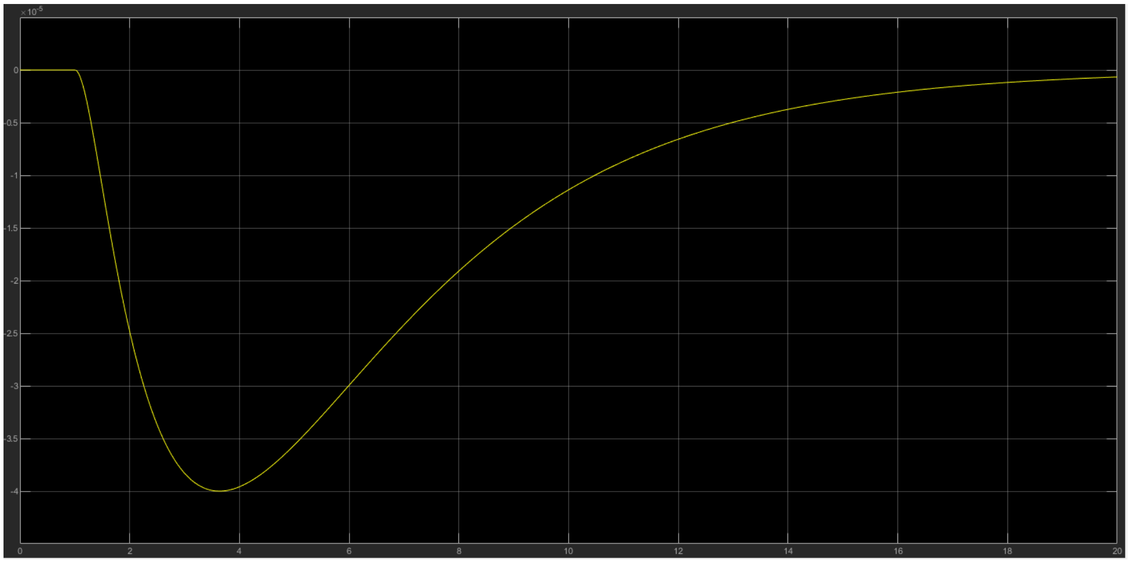

Figure 4: Pitch Response to Unit Step Disturbance

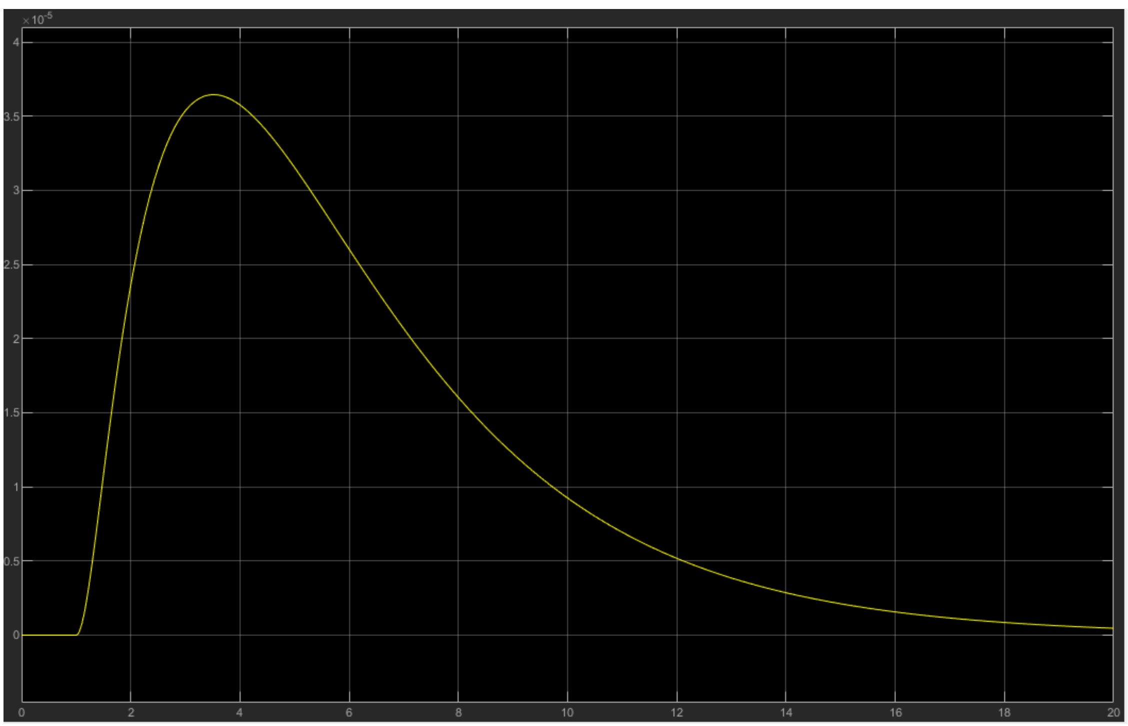

Figure 5: Roll Response to Unit Step Disturbance

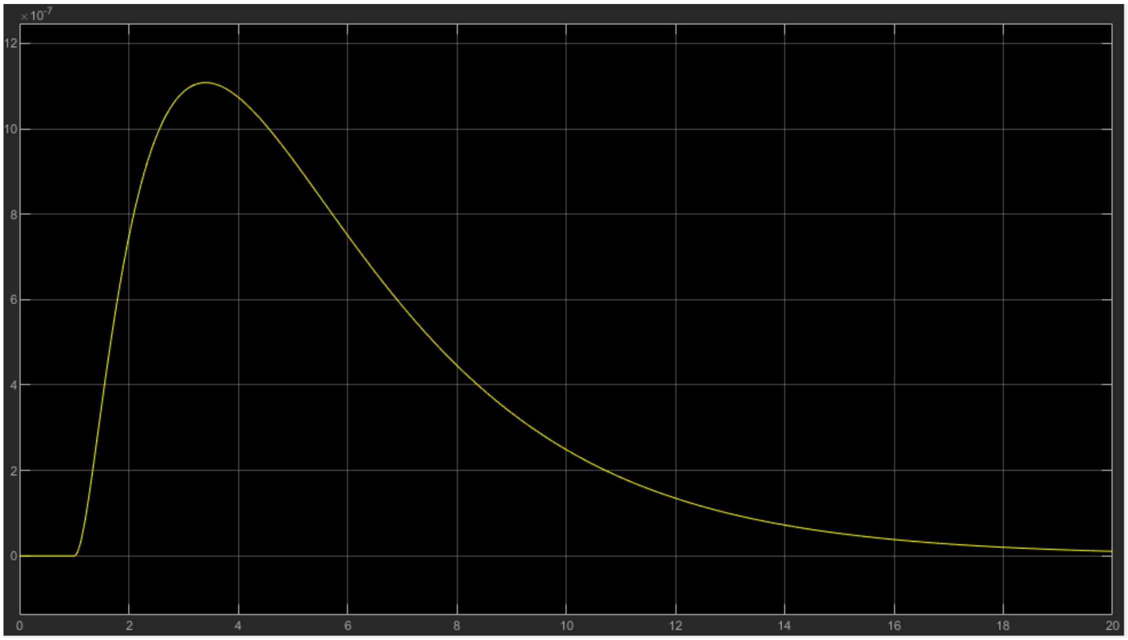

Figure 6: Yaw Response to Unit Step Disturbance

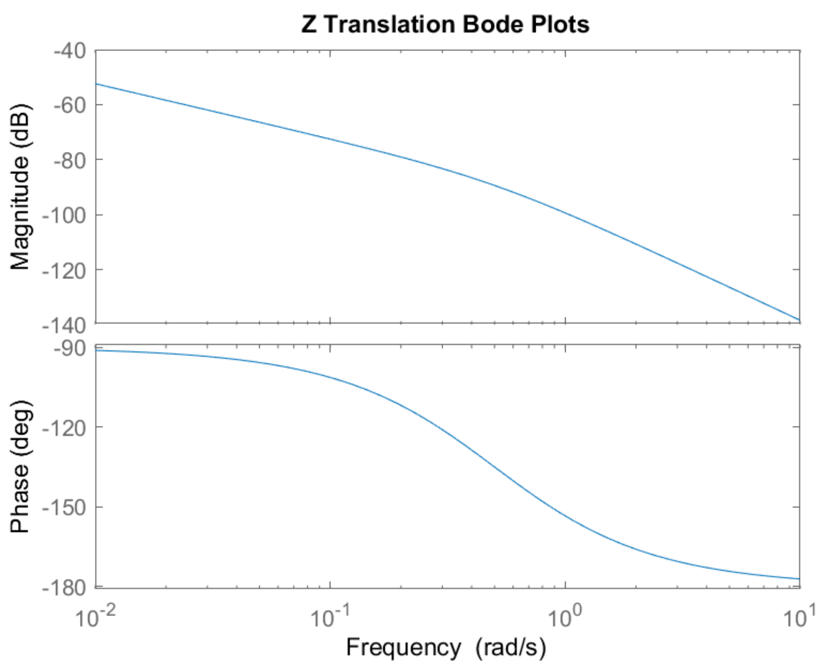

Figure 7

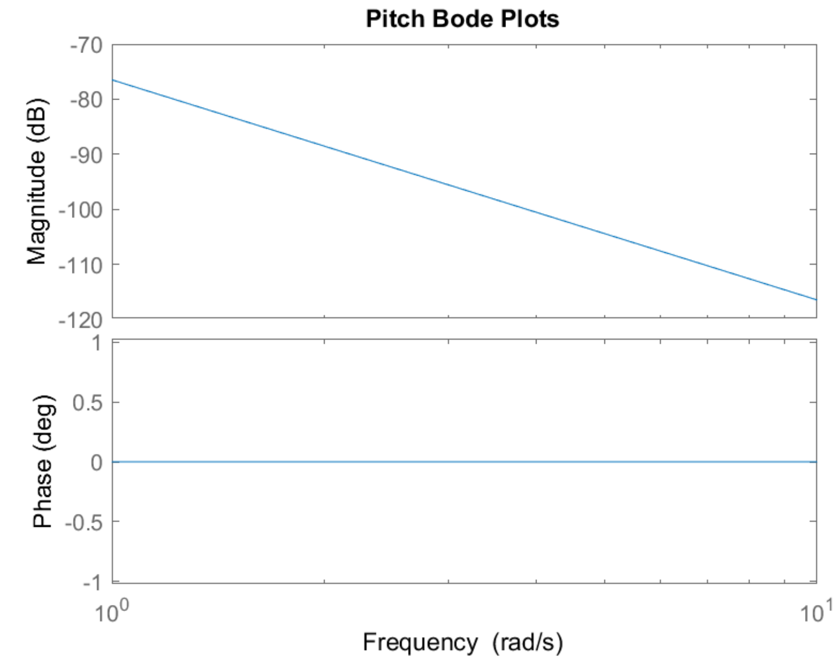

Figure 8

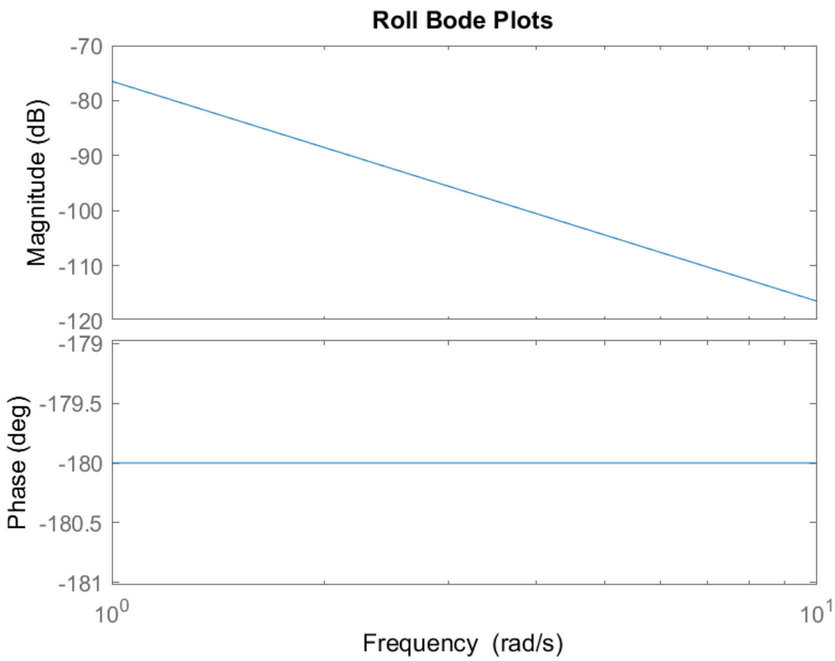

Figure 9

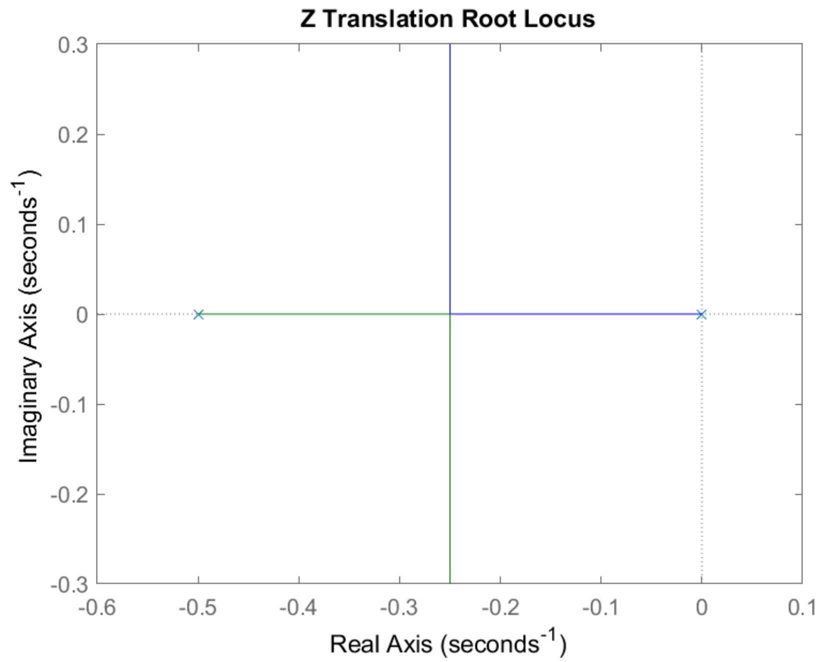

Figure 10

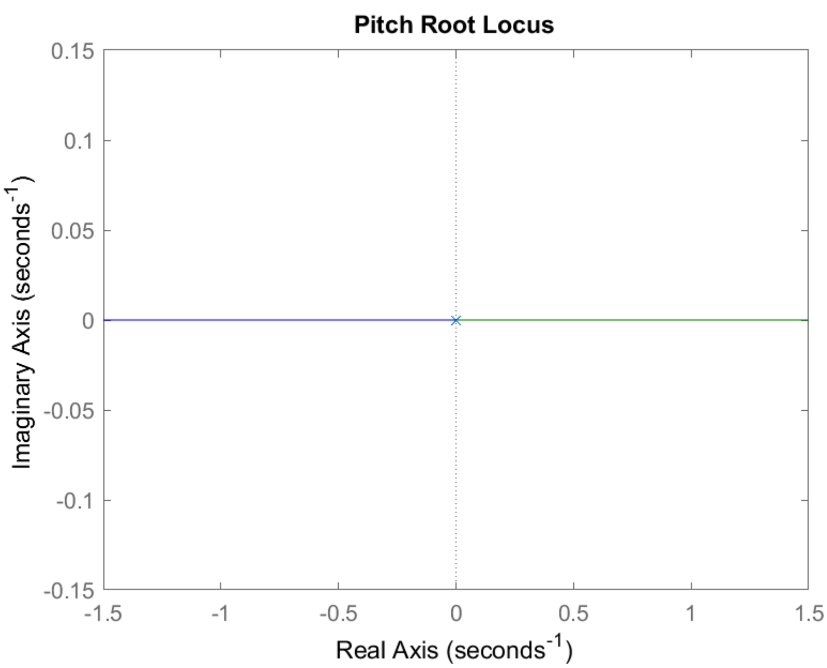

Figure 11

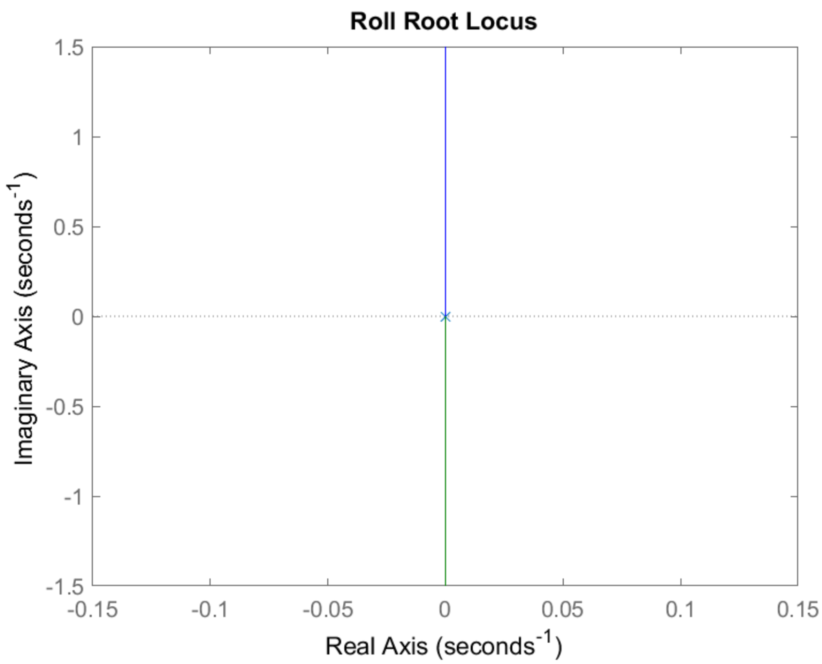

Figure 12

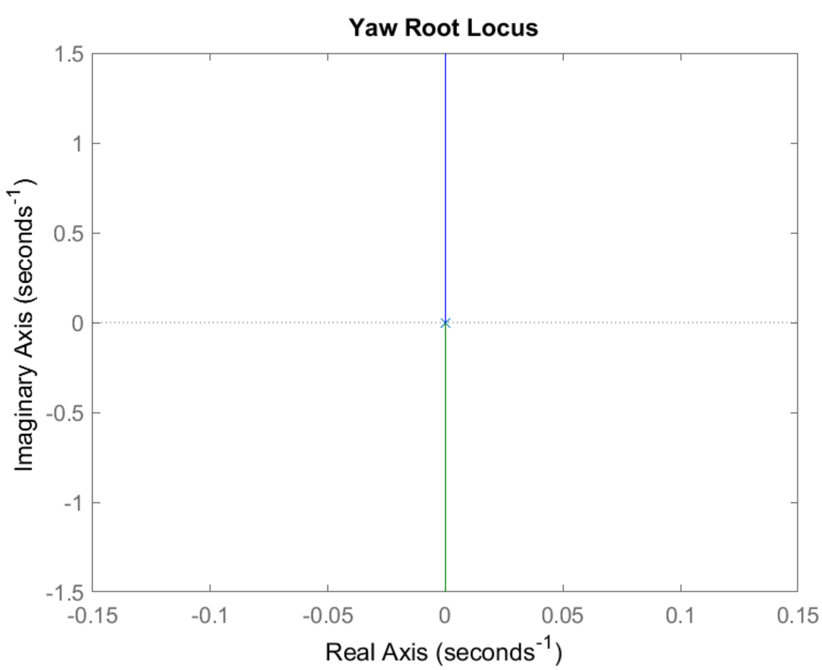

Figure 13