Controlling of Fundamental UAVs

(Unmannned Aerial Vehicles)

Carrie Yan
Supervised by: Prof. Hugh H.T Liu

Flight Simulation and Control Lab
Engineering Science, Aerospace
Universities of Toronto

August 24^{th}, 2015

Outline

I. Introduction

- Fundamental UAVs – X shape Quad rotors
- No control surfaces, all motions are controlled by 2 types of identical unidirectional rotors.
- Each rotor can provide a thrust and a torque.
- front motor M1 & rear M3 rotate counterclockwise
- left motor M2 & right M4 turn clockwise.
- Objective:
  model the quad rotors and control for tracking problems

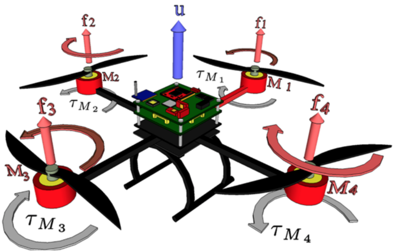


Figure: 3.1 Quadrotor

II.Abbreviation

m	mass of quadrotor
g	gravity
I_x, I_y, I_z	moment of inertia in each direction
$\vec{R} = C_1(\phi)C_2(\theta)C_3(\psi)$	Euler rotation matrix in 1-2-3 sequence[1]
W	angular velocities conversion matrix [2]

body frame		inertial frame	
u,v,w	velocities	$\dot{x},\dot{y},\dot{z}$	velocities
$\vec{\Omega} =$ p,q,r	angular velocities	$\vec{\dot{\eta}} = \dot{\phi},\dot{\theta},\dot{\psi}$	angular velocities
T	main thrust	$\vec{\xi} =$ x,y,z	positions
τ_x, τ_y, τ_z	roll,pitch,yaw torques	$\vec{\eta} = \phi, \theta, \psi$	Euler angles

[1]See Appendix A. for details

[2]See Appendix A for details

III.Model in body fixed frame

- $$\begin{bmatrix} -mgsin(\theta) \\ mgcos(\theta)sin(\phi) \\ T + mgcos(\theta)cos(\phi) \end{bmatrix} = \begin{bmatrix} m\left(\dot{u} + qw - rv\right) \\ m\left(\dot{v} + ru - pw\right) \\ m\left(\dot{w} + pv - qu\right) \end{bmatrix}$$
- $$\begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} I_x\dot{p} - \left(I_y - I_z\right)qr \\ I_y\dot{q} - \left(I_z - I_x\right)pr \\ I_z\dot{r} - \left(I_x - I_y\right)pq \end{bmatrix}$$

- Conversion between body and inertial frame
- $$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R^{-1}\begin{bmatrix} u \\ v \\ w \end{bmatrix}$$
- $$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = W^{-1}\begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

III.Model in earth inertial frame

Lagrangian Method

Lagrangian: $\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \left(\frac{\partial L}{\partial q}\right) = \vec{F}_{nonconservative}$

Generalized coordinates q: $\vec{\xi} = (x, y, z)$ and $\vec{\eta} = (\phi, \theta, \psi)$

1. For translation motion (coordinates x,y,z)
   - Kinematic Energy $T = \frac{1}{2}m\dot{\xi}^T\dot{\xi}$
   - Potential Energy $U = mgz$
   - Nonconservative force $\vec{F} = \vec{R}^T \vec{F_{body}} = \vec{R}^T \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix}$

Result: $\begin{bmatrix} m\ddot{x} \\ m\ddot{y} \\ m\ddot{z} \end{bmatrix} = \begin{bmatrix} T\left(sin(\phi)sin(\psi) + cos(\phi)cos(\psi)sin(\theta)\right) \\ T\left(cos(\phi)sin(\theta)sin(\psi) - cos(\psi)sin(\theta)\right) \\ Tcos(\theta)cos(\phi) - mg \end{bmatrix}$

2. For rotation motion (coordinates ϕ, θ, ψ)
   - Kinematic Energy $T = \frac{1}{2}\vec{\Omega}^T I \vec{\Omega} = \frac{1}{2}\dot{\vec{\eta}}^T J \dot{\vec{\eta}}$, where $J = W^T I W$
   - Potential Energy U=0
   - Nonconservative torque $R^T \vec{\tau}_{body}$

Result:

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = J^{-1} \left\{ R^T \vec{\tau}_{body} - \left[\dot{W}^T I W + W^T I \dot{W} - \frac{1}{2} \frac{\partial}{\partial \vec{\eta}} \left(\dot{\vec{\eta}}^T J \right) \right] \dot{\vec{\eta}} \right\}$$

From Newton Second Law, $\vec{F} = m\vec{a}$

Result: $$\begin{bmatrix} m\ddot{x} \\ m\ddot{y} \\ m\ddot{z} \end{bmatrix} = R^T \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} + \vec{F}_{body} = R^T \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}$$

Also, $I\dot{\vec{\Omega}} = -\vec{\Omega} \times I\vec{\Omega} + \vec{\tau}$, where $\vec{\Omega} = W\dot{\vec{\eta}}$

Result: $$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \ddot{\vec{\eta}} = I^{-1}W^{-1}\left(-I\dot{W}\dot{\vec{\eta}} - W\dot{\vec{\eta}} \times IW\dot{\vec{\eta}} + \vec{\tau}_{body}\right)$$

III.Modelling

Summary

- Input: main thrust T, torques τ_x, τ_y, τ_z
- Output: positions x,y,z
- State Vector $\begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \phi \\ \theta \\ \psi \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$

For model verification,

Choose:

Simulation time = 10s

T = sine wave with bias m*g, frequency 1 rad/s, amplitude 0.02 N

τ_x = sine with no bias, amplitude 0.0001 N/m, frequency 1 rad/s

τ_y = sine with amplitude 0.0001 N/m and frequency 2 rad/s

τ_z = step at time 1s with final value 0.0001 N/m

III.Modelling

Open Loop Response Comparison

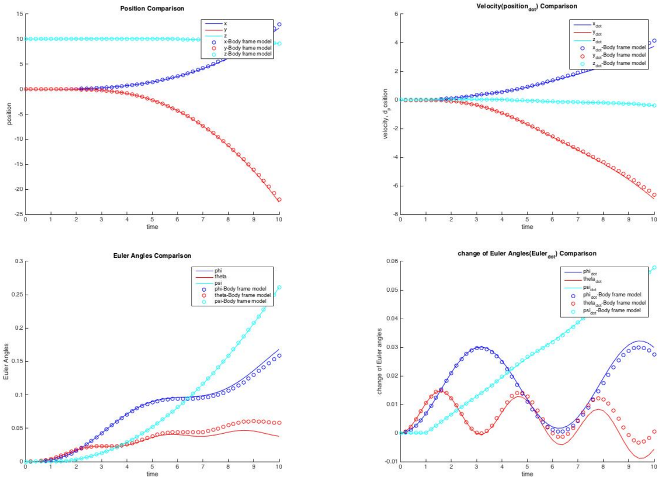


Figure: 11.1 Open Loop Response Comparison Result

III.Model

Difference

Modelling in the body fixed frame and in the earth inertial frame with Newtonian method look similar.
Difference:
the order of state space vector updates and the conversion between body frame and inertial frame.

- Every time, state space vectors are updated to next time based on previous state information.
- body frame modeling does conversion after updates
- inertial frame modeling does conversions first

That is EVERYTHING about the model.

IV.Stability

Method of Analysis

- Necessity of Controllers
  - Plant is Stable?
    If No, need **stabilizing controllers** to stabilize the model
    If Yes, need **rate controllers** to enhance the performance of plant?
- **Impulse Response Analysis**
  - Stable: at infinite time, response reduced to 0
  - Marginally Stable: at infinite time, response reduced to a finite number
  - Unstable: response unbounded
- **Poles Position Analysis** via eigenvalue of state vector A
  - Stable: all poles in left-hand plane
  - Marginally Stable: some poles lie on imaginary axis, while no right-hand plane poles
  - Unstable: there exists right-hand plane pole

IV.Stability

Linearization

- Equation of Motion used (Newtonian Method)
  It is of the non-linear form $\dot{\vec{x}} = f(\vec{x})$
- $$\begin{bmatrix} m\ddot{x} \\ m\ddot{y} \\ m\ddot{z} \end{bmatrix} = \begin{bmatrix} T\left(sin(\phi)sin(\psi) + cos(\phi)cos(\psi)sin(\theta)\right) \\ T\left(cos(\phi)sin(\theta)sin(\psi) - cos(\psi)sin(\theta)\right) \\ Tcos(\theta)cos(\phi) - mg \end{bmatrix}$$
- $$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \ddot{\vec{\eta}} = I^{-1}W^{-1}\left(-I\dot{W}\dot{\vec{\eta}} - W\dot{\vec{\eta}} \times IW\dot{\vec{\eta}} + \vec{\tau}_{body}\right)$$
- Linearized around operating point via calculations of Jacobian
- Operating Point: Hovering Mode
  $T = m * g,\ \tau_x = 0, \tau_y = 0, \tau_z = 0$

State Space Vector $\begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} & \phi & \theta & \psi & \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$
$= \begin{bmatrix} 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$

IV.Stability

Linearized Model

$$\underline{\dot{x}} = \underline{A}\underline{x} + \underline{B}\underline{u}$$

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \phi \\ \theta \\ \psi \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9.81 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -9.81 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \phi \\ \theta \\ \psi \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1.2346 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 147.9290 & 0 & 0 \\ 0 & 0 & 147.9290 & 0 \\ 0 & 0 & 0 & 63.2911 \end{bmatrix} \begin{bmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}$$

IV.Stability

Linearized Model

$$\underline{y} = \underline{C}\underline{x} + \underline{D}\underline{u}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \phi \\ \theta \\ \psi \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \underline{0}_{3\times4} \end{bmatrix} \begin{bmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}$$

1

[1] Calculations are carried out by Matlab symbolic analysis

Open Loop Response Comparison

For model verification, compare the linearized model with full model

Choose:

Simulation time = 10s

T = sine wave with bias m*g, frequency 1 rad/s, amplitude 0.02 N

τ_x = sine with no bias, amplitude 0.0001 N/m, frequency 1 rad/s

τ_y = sine with amplitude 0.0001 N/m and frequency 2 rad/s

τ_z = step at time 1s with final value 0.0001 N/m

Note: Differences should be rather small when the disturbances are small.

IV.Stability

Open Loop Response Comparison Result

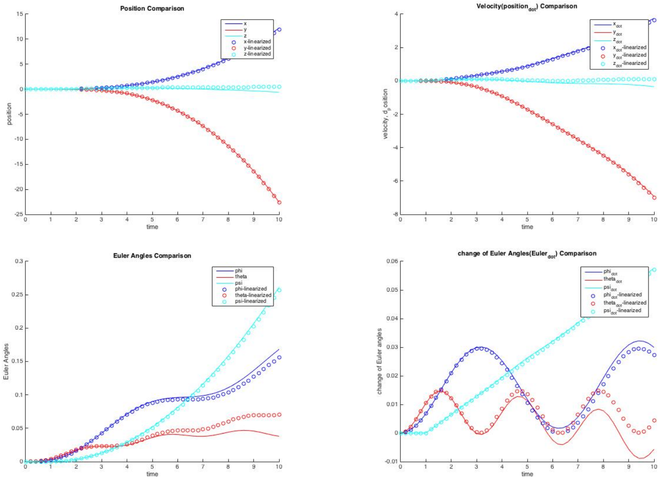


Figure: 18.1 Open Loop Response Comparison Result

- Reason for linearization: stability analysis
- Eigenvalue of State Matrix A:
  eigenvalue = [0 0 0 0 0 0 0 0 0 0 0 0]
  System is marginally stable
  Controller in need
- Reason for linearization: relationship between inputs, outputs and state parameters
- For example, which parameter dominates the influence on main thrust T?
  By linearization, we can get $T \propto \ddot{z}$, so feedback information for controllers of Thrust Channel, would be about the position z, velocity $\dot{z}$ and acceleration $\ddot{z}$

V. Controllers

SISO system

PID controllers are chosen for this design.

First, MIMO system are categarized into 4 SISO systems.

$$\frac{d}{dt}\begin{bmatrix} z \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} z \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ 1.2346 \end{bmatrix} T$$

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 9.81 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 147.929 \end{bmatrix} \tau_y$$

$$\frac{d}{dt}\begin{bmatrix} y \\ \dot{y} \\ \phi \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -9.81 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} y \\ \dot{y} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 147.929 \end{bmatrix} \tau_x$$

$$\frac{d}{dt}\begin{bmatrix} \psi \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ 63.2911 \end{bmatrix} \tau_z$$

V. Controllers

Physical Explanation

Operating Point: Hover Mode, where $T = m * g$, $\tau_x = 0$, $\tau_y = 0$, $\tau_z = 0$

State Space Vector $\begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} & \phi & \theta & \psi & \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T =$
$\begin{bmatrix} 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$

- 6 relationship: $\ddot{z} \propto T$, $\ddot{\phi} \propto \tau_x$, $\ddot{y} \propto \phi$, $\ddot{\theta} \propto \tau_y$, $\ddot{x} \propto \theta$, $\ddot{\psi} \propto \tau_z$
- For $\ddot{z} \propto T$ and $\ddot{y} \propto \phi$ and $\ddot{x} \propto \theta$ and $\ddot{\psi} \propto \tau_z$, It is straightforward that the linear/angular accelerations are proportional to the corresponding force/torque.
- For $\ddot{y} \propto \phi$ and $\ddot{x} \propto \theta$,
  For example, when the quadrotor tilts from hovering mode for a small pitch angle θ, keeping the thrust equal to $m * g$, then decompose the thrust force in inertial frame. The component in x-direction is $m * gsin(\theta)$, according to Newton's Law, the acceleration in x-direction $\ddot{x}$ is related to $gsin(\theta)$. For small angle, $sin(\theta) \approx \theta$. Hence, $\ddot{x}$ is proportional to θ.
  Similarly, $\ddot{y}$ is proportional to ϕ.

V. Controllers

Controller Schematics

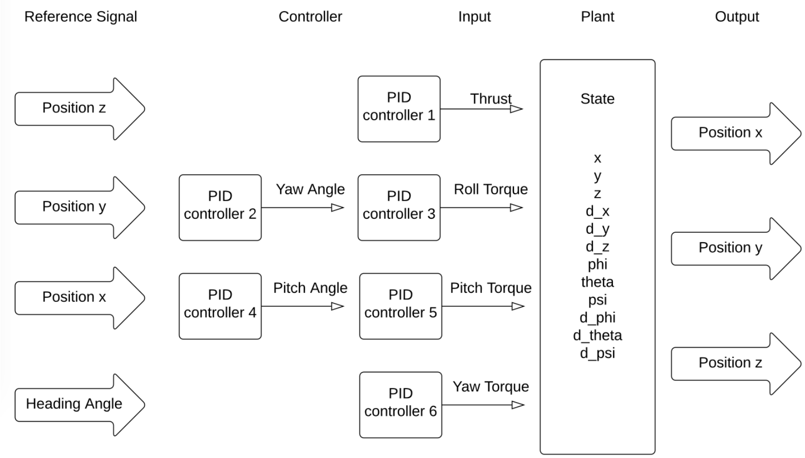


Figure: 21.1 Controller Schematics

V. Controllers

Inner Loop & Outer Loop

- In reality, the sensors can only sense the **accelerations** with gyroscope or the **positions** with the help of Global Positioning System (GPS)
- Normally, the inner loop is for faster regulation, the feedback are linear/angular velocities; while the outer loop is for slower regulation, and the feed back are positions of vehicle or Euler angles.

V. Controllers

Controller Schematics with Inner Loop & Outer Loop

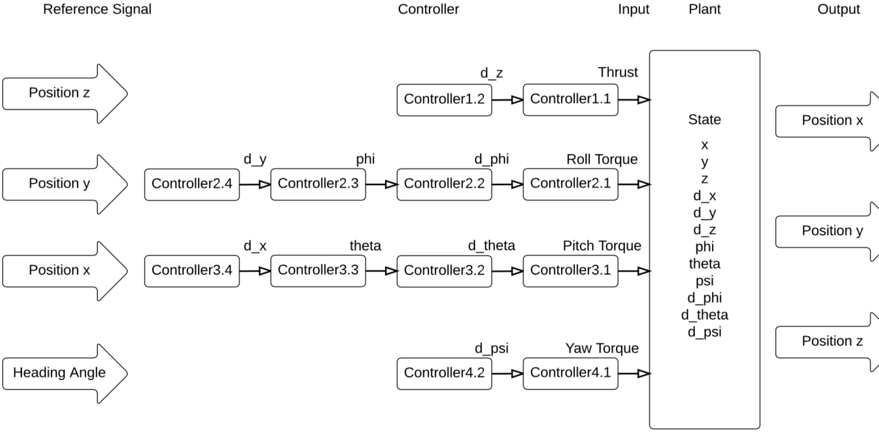


Figure: 23.1 Controller Schematics with inner & outer loop

Summary: 4 channels (Thrust, roll, pitch, yaw torques)

V. Controllers

Offset $m * g$ for thrust channel

- Linearization is performed around operating point, when thrust equals to $m * g$ and three torques to be zero. Our control is feedback error control. Using PID controllers, at steady state (infinity time), the feedback error is expected to be zero constantly. The proportional/integral/derivative of zero are all zeros. Thus, input of thrust is zero. However, we are expecting the input to be gravity of the vehicles to keep the vehicle in steady state, namely hovering.
- On the other hand, in the linearized model $\delta\dot{\vec{x}} = A\delta\vec{x} + B\delta\vec{u}$, controllers regulate the variation of space states and inputs, instead of the space states and inputs themselves. That is why we need an offset $m * g$ as the "initial condition" for thrust channel input.

After controllers schematics finalized, in need of a method to tune PID controllers parameters

- Trial-and-Error Method (recommended)
- Ziegler Nichols Method (constrained)
  –Unit Step Response
  –Frequency Response
- Root Locus Method (accurate in theory but complicated, troublesome)

Ziegler Nicholos Unit Step Response

- Apply a unit step input to the plant and obtain response, figure out the values of delay time L, time constant T. [1]
- This method only applies when the step response is an S-shaped curve.
- When the plant involves neither integrator(s) nor dominant complex-conjugate poles,then such a unit-step response curve may look S-shaped.

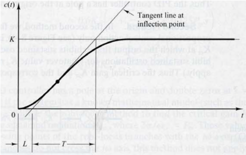


Figure: 26.1 S-curved Step Response

- Comment: Plant is in form of double integrators $\frac{num}{s^2}$. NOT WORK.

[1] See Appendix B. for detailed PID parameters values

Ziegler Nicholos Frequency Response

- Apply a step response and increase proportional gain K until the system becomes marginally stable and continuous oscillations, then the corresponding gain & period of oscillation are called ultimate gain K_u and ultimate period P_u. [1]

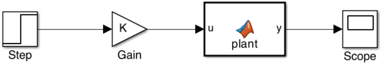


Figure: 27.1 Scheme of ZN frequency response method

- Comment: Plant is in form of double integrators $\frac{num}{s^2}$. NOT WORK.

[1] See Appendix B. for detailed PID parameters values

From the linearized model Equation of Motions (EMs), we can get Summary of the Transfer Functions(TFs) via Laplace Transform:

Linearized EMs	TFs	TFs with inner&outer loop [1]	
$\ddot{x} = 9.81 * \theta$	$\frac{X}{\Theta} = \frac{9.81}{s^2}$	$G_{3.3} = \frac{\dot{X}}{\Theta} = \frac{9.81}{s}$	$G_{3.4} = \frac{X}{\dot{X}} = \frac{1}{s}$
$\ddot{y} = -9.81 * \phi$	$\frac{Y}{\Phi} = \frac{-9.81}{s^2}$	$G_{2.3} = \frac{\dot{Y}}{\Phi} = \frac{-9.81}{s}$	$G_{2.4} = \frac{Y}{\dot{Y}} = \frac{1}{s}$
$\ddot{z} = 1.2346 * T$	$\frac{Z}{T} = \frac{1.2346}{s^2}$	$G_{1.1} = \frac{\dot{Z}}{T} = \frac{1.2346}{s}$	$G_{1.2} = \frac{Z}{\dot{Z}} = \frac{1}{s}$
$\ddot{\phi} = 147.929 * \tau_x$	$\frac{\Phi}{T_x} = \frac{147.929}{s^2}$	$G_{2.1} = \frac{\dot{\Phi}}{T_x} = \frac{147.929}{s}$	$G_{2.2} = \frac{\Phi}{\dot{\Phi}} = \frac{1}{s}$
$\ddot{\theta} = 147.929 * \tau_y$	$\frac{\Theta}{T_y} = \frac{147.929}{s^2}$	$G_{3.1} = \frac{\dot{\Theta}}{T_y} = \frac{147.929}{s}$	$G_{3.2} = \frac{\Theta}{\dot{\Theta}} = \frac{1}{s}$
$\ddot{\psi} = 63.2911 * \tau_z$	$\frac{\Psi}{T_z} = \frac{63.2911}{s^2}$	$G_{4.1} = \frac{\dot{\Psi}}{T_z} = \frac{63.2911}{s}$	$G_{4.2} = \frac{\Psi}{\dot{\Psi}} = \frac{1}{s}$

[1] $G_{i,j}$ denotes the j^{th} TF of i^{th} channel

V. Controllers

Root Locus–Thrust & Yaw Channel

- Thrust & Yaw Channels are similar to each other.
- 2 loops of controllers
- TFs for Thrust: $G_{1.1} = \frac{\dot{Z}}{T} = \frac{1.2346}{s}$ $G_{1.2} = \frac{Z}{\dot{Z}} = \frac{1}{s}$
- TFs for Yaw: $G_{4.1} = \frac{\dot{\Psi}}{T_z} = \frac{63.2911}{s}$ $G_{4.2} = \frac{\Psi}{\dot{\Psi}} = \frac{1}{s}$

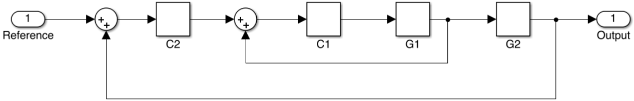


Figure: 29.1 Controllers and Transfer Functions

Note that $G_n = \frac{v_n}{s}$ are TFs and $C_n = K_n \frac{s-i_n}{s}$ are controllers.

v_n are given by TFs derived before
i_n are chosen manually
K_n are given by root locus plot by selecting the gain corresponding to most rapid stable response.

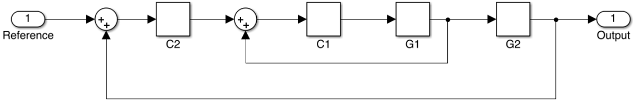
V. Controllers

Root Locus–Thrust & Yaw Channel

	v_1	i_1	K_1	v_2	i_2	K_2
Thrust	1.2346	0.1	0.324	1	0.1	0.1
Yaw	63.2911	0.1	0.0632	1	0.1	0.1

[1]

Note that $G_n = \frac{v_n}{s}$ are TFs and $C_n = K_n \frac{s - i_n}{s}$ are controllers.
v_n are given by TFs derived before; i_n are chosen manually; K_n are given by root locus plot by selecting the gain corresponding to most rapid stable response.



[1] See Appendix for plots details

V. Controllers

Root Locus–Roll & Pitch Channel

- Roll & Pitch Channels are similar to each other.
- 4 loops of controllers
- TFs for Roll: $G_{2.1} = \frac{\dot{\Phi}}{T_x} = \frac{147.929}{s}$ $G_{2.2} = \frac{\Phi}{\dot{\Phi}} = \frac{1}{s}$
  $G_{2.3} = \frac{\dot{Y}}{\Phi} = \frac{-9.81}{s}$ $G_{2.4} = \frac{Y}{\dot{Y}} = \frac{1}{s}$
- TFs for Pitch: $G_{3.1} = \frac{\dot{\Theta}}{T_y} = \frac{147.929}{s}$ $G_{3.2} = \frac{\Theta}{\dot{\Theta}} = \frac{1}{s}$
  $G_{3.3} = \frac{\dot{X}}{\Theta} = \frac{9.81}{s}$ $G_{3.4} = \frac{X}{\dot{X}} = \frac{1}{s}$

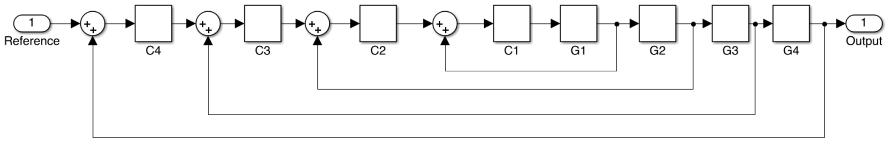


Figure: 31.1 Controllers and Transfer Functions

V. Controllers

Root Locus–Roll & Pitch Channel

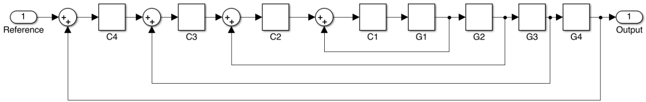
	v_1	i_1	K_1	v_2	i_2	K_2
Roll	147.929	0.1	0.002704	1	0.1	0.1
Pitch	147.929	0.1	0.002704	1	0.1	0.1

[1]

	v_3	i_3	K_3	v_4	i_4	K_4
Roll	-9.81	0	-0.01	1	0.1	0.03
Pitch	9.81	0	0.01	1	0.1	0.03

Note: for tuning $C_{2,3}$, plot negative root locus plot.

Note that $G_n = \frac{v_n}{s}$ are TFs and $C_n = K_n \frac{s - i_n}{s}$ are controllers.
v_n are given by TFs derived before; i_n are chosen manually; K_n are given by root locus plot by selecting the gain corresponding to most rapid stable response.



[1]See Appendix for plots details

V. Controllers

Parameters Comparison

Channel	Trial & Error			Root Locus		
Controller	K_P	K_I	D	K_P	K_I	K_D
Thrust $C_{1.1}$	1.5	0.01	0.05	0.324	0.0324	0
Thrust $C_{1.2}$	0.3	0	0	0.1	0.01	0
Roll $C_{2.1}$	0.05	0	0.0005	0.002704	0.0002704	0
Roll $C_{2.2}$	0.5	0.001	0	0.1	0.01	0
Roll $C_{2.3}$	-0.02	0	0	-0.01	0	0
Roll $C_{2.4}$	0.1	0	0.01	0.03	0.003	0
Pitch $C_{3.1}$	0.05	0	0.0005	0.002704	0.0002704	0
Pitch $C_{3.2}$	0.5	0.001	0	0.1	0.01	0
Pitch $C_{3.3}$	0.02	0	0	0.01	0	0
Pitch $C_{3.4}$	0.1	0	0.01	0.03	0.003	0
Yaw $C_{4.1}$	0.01	0	0.001	0.0632	0.00632	0
Yaw $C_{4.2}$	0.15	0	0.02	0.1	0.01	0

Note that filter coefficient $N = 100$ was chosen.

V. Controllers

Parameters Comparison

- Root locus method is based on linearized model.
  Only valid when the disturbances are rather small.
- Also, Root Locus methods only provide information about the proportional and integral parts, we have no clues about the derivative parts (and filter coefficients).
- In need of adjustments of PID parameters obtained by Root Locus method, to optimize the behaviors.

VI. Simulation

Longitudinal Manipulation with controllers tuned by Root Locus

Objective:

initial condition: Hovering Mode

desired position: $x_d = 0$, $y_d = 0$, $z_d = 30$, Heading Angle $= 0$

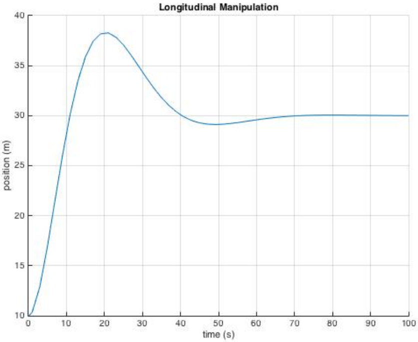


Figure: 35.1 Altitude Manipulation

VI. Simulation

Lateral Manipulation with controllers tuned by Root Locus

Objective:

initial condition: Hovering Mode

desired position: $x_d = 0$, $y_d = 0$, $z_d = 10$, Heading Angle $= 10^\circ = 0.174$ rads

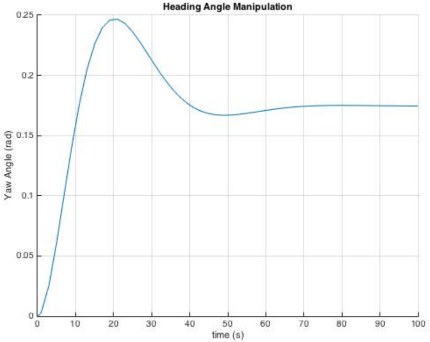


Figure: 36.1 Heading Angle Manipulation

VI. Simulation

x position Manipulation with controllers tuned by Root Locus

Objective:

initial condition: Hovering Mode

desired position: $x_d = 30$, $y_d = 0$, $z_d = 10$, Heading Angle $= 0$

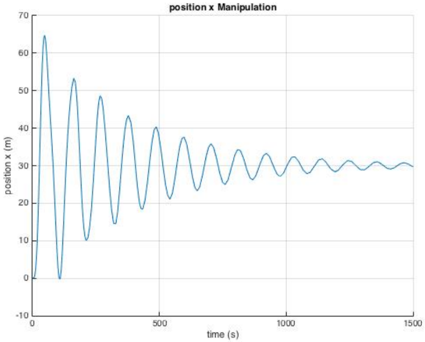


Figure: 37.1 Heading Angle Manipulation

VI. Simulation

y position Manipulation with controllers tuned by Root Locus

Objective:

initial condition: Hovering Mode

desired position: $x_d = 0$, $y_d = 40$, $z_d = 10$, Heading Angle $= 0$

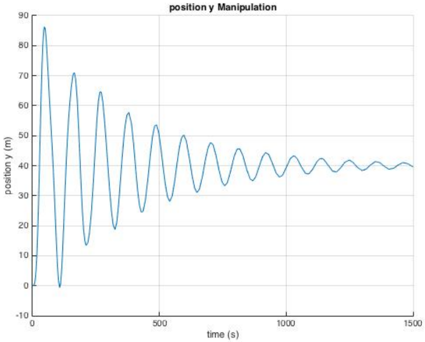


Figure: 38.1 Heading Angle Manipulation

VI. Simulation

Limitation of PID controls tuned by Root Locus

- Time invariant controllers
- Behaviors might be aggressive

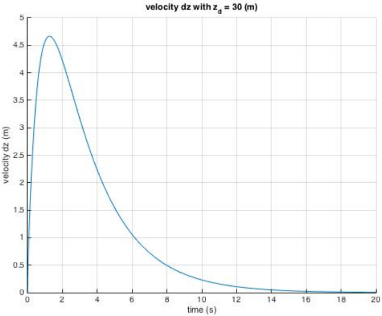


Figure: 39.1 Velocity of z with $z_{desire} = 30m$

Note that: for position z,
rise time =5.770971s and settling time =11.867515s.

VI. Simulation

Limitation of PID controls tuned bu Root Locus

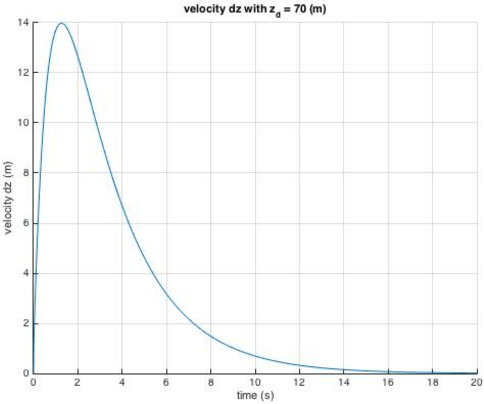


Figure: 40.1 Velocity of z with $z_{desire} = 70m$

Note that: for position z,
rise time =6.443427s and settling time =12.517229s.

VI. Simulation

Limitation of PID controls tuned bu Root Locus

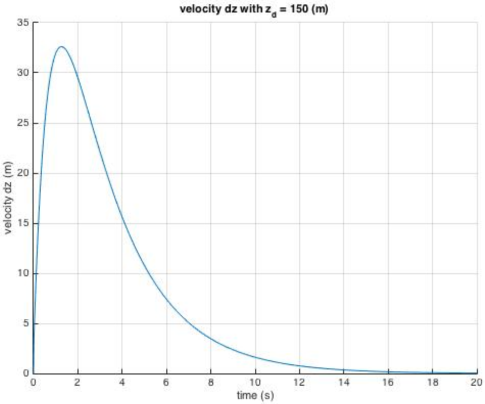


Figure: 41.1 Velocity of z with $z_{desire} = 150m$

Note that: for position z,
rise time =6.29885s and settling time =12.744253s.

VII. Reality Application

- In reality, we cannot directly control the main thrust T and three torques τ_x, τ_y, τ_z
- Relation between 4 thrusts and 4 controlled items:

$$\begin{bmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} T_1 + T_2 + T_3 + T_4 \\ (T_1 - T_3)\, l \\ (T_2 - T_4)\, l \\ k\,(T_1 - T_2 + T_3 - T_4) \end{bmatrix}$$

- 4 unknowns with 4 equations, SOLVEABLE!

Thank you for your attention!

Reference

Appendix A

Rotation Matrix

$$R = \vec{C}_{BI} = \begin{bmatrix} cos\psi cos\theta & sin\psi cos\theta & -sin\theta \\ cos\psi sin\theta sin\phi - sin\psi cos\phi & sin\psi sin\theta sin\phi + cos\psi cos\phi & cos\theta sin\phi \\ cos\psi sin\theta cos\phi + sin\psi sin\phi & sin\psi sin\theta cos\phi - cos\psi sin\phi & cos\theta cos\phi \end{bmatrix}$$

Rotation matrix $\vec{R}$ is orthonormal, i.e $\vec{R}_T = \vec{R}_{-1}$

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

Appendix A

Body Frame Method

Known from classic mechanics, for first derivative, $\vec{v}^{\bullet} = \vec{v}^{\circ} + \vec{\omega} \times \vec{v}$

$$\begin{bmatrix} -mgsin(\theta) \\ mgcos(\theta)sin(\phi) \\ T + mgcos(\theta)cos(\phi) \end{bmatrix} = \begin{bmatrix} m\left(\dot{u} + qw - rv\right) \\ m\left(\dot{v} + ru - pw\right) \\ m\left(\dot{w} + pv - qu\right) \end{bmatrix}$$

$$\text{Force in inertial frame} = \begin{bmatrix} -mgsin(\theta) \\ mgcos(\theta)sin(\phi) \\ T + mgcos(\theta)cos(\phi) \end{bmatrix}$$

$$\text{Velocity in body frame} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

$$\text{Angular velocity between two frames} = \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

Appendix B

Ziegler Nichols Method

- Unit Step Response

Type of Controller	K_p	K_I	K_D
P	$\frac{T}{L}$	0	0
PI	$\frac{0.9T}{L}$	$\frac{L}{0.3}$	0
PID	$\frac{1.2T}{L}$	$2L$	$0.5L$

- Frequency Response

Type of Controller	K_p	K_I	K_D
P	$0.5K_u$	0	0
PI	$0.45K_u$	$\frac{P_u}{1.2}$	0
PID	$0.6K_u$	$0.5P_u$	$0.125P_u$

Appendix C

Root Locus

- Slowest pole dominates the behavior of response.
- Closer the left hand plane poles to the imaginary axis, slower the response is.
- Choose the value of gain K to make the slowest pole as far from the imaginary axis as possible. (as left as possible)
- Controller 1 gain is easy to determinate by calculating the gain corresponding to break-in points.
- Rest of controllers gain are found by using System Identification Toolbox.
  type "controlSystemDesigner(system)" in command window

Appendix C

Root Locus Plot–Thrust

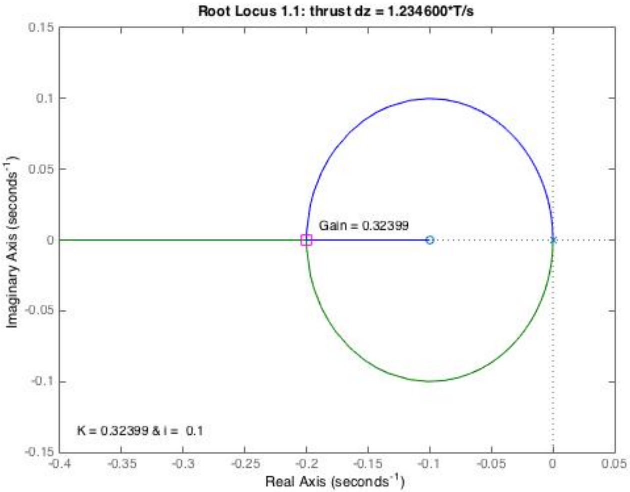


Figure: Root Locus Plot (Thrust channels)

Appendix C

Root Locus Plot–Thrust

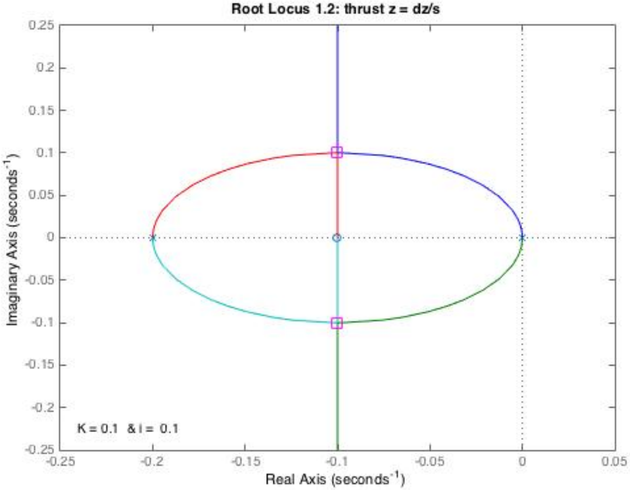


Figure: Root Locus Plot (Thrust channels)

Appendix C

Root Locus Plot–Yaw

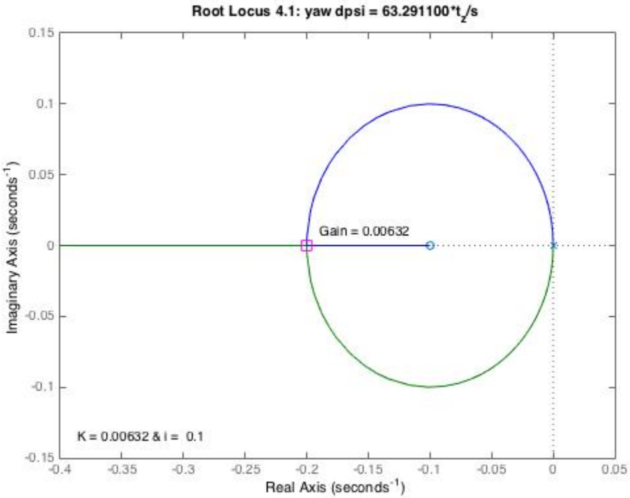


Figure: Root Locus Plot (Yaw channels)

Appendix C

Root Locus Plot–Yaw

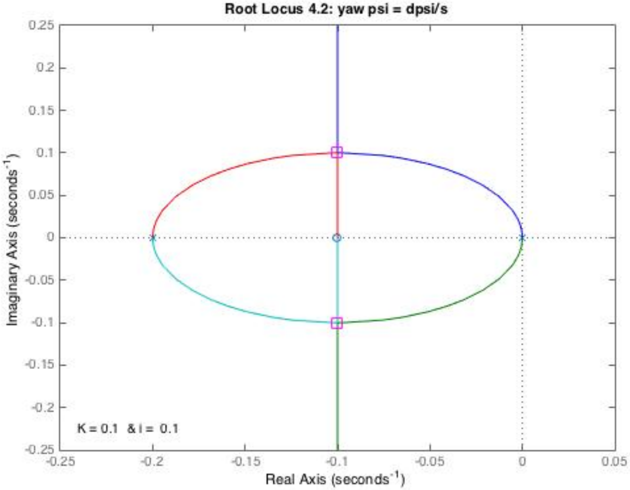


Figure: Root Locus Plot (Yaw channels)

Appendix C

Root Locus Plot–Roll & Pitch

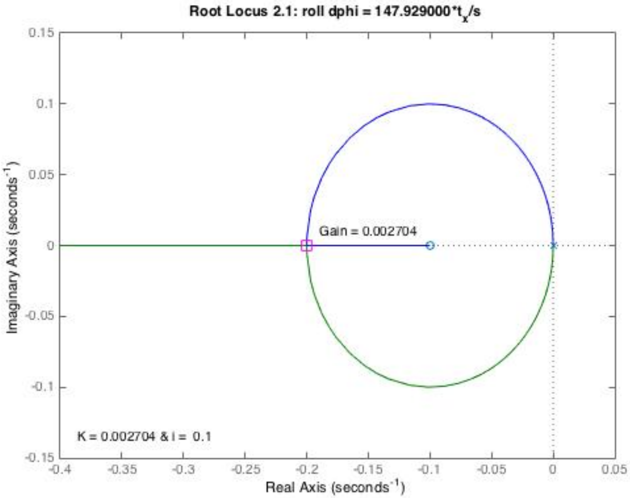


Figure: Root Locus Plot (Roll & Pitch channel)

Appendix C

Root Locus Plot–Roll & Pitch

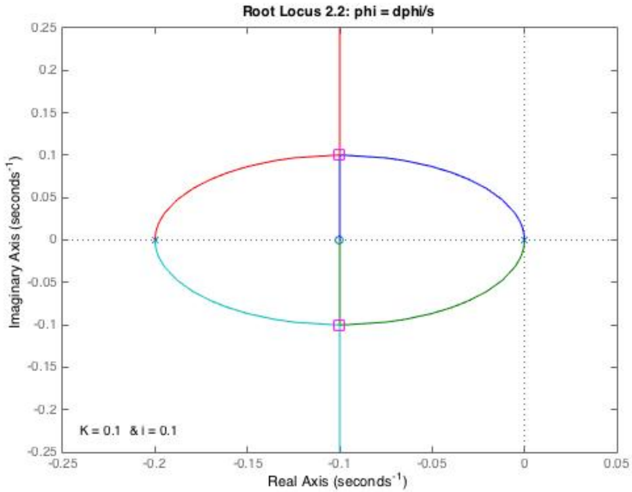


Figure: Root Locus Plot (Roll & Pitch channel)

Appendix C

Root Locus Plot–Roll & Pitch

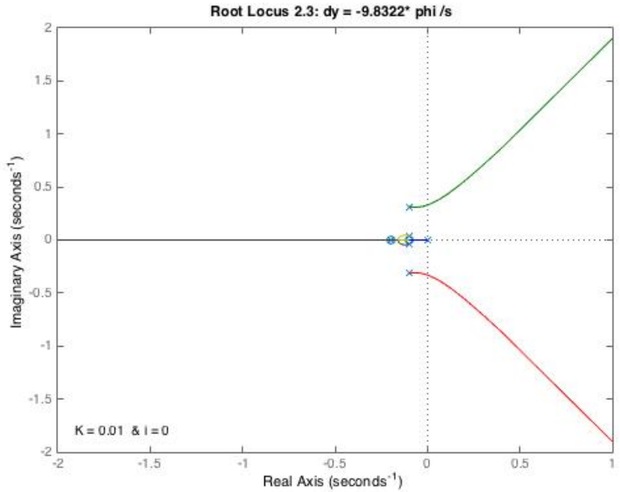


Figure: Root Locus Plot (Roll & Pitch channel)

Appendix C

Root Locus Plot–Roll & Pitch

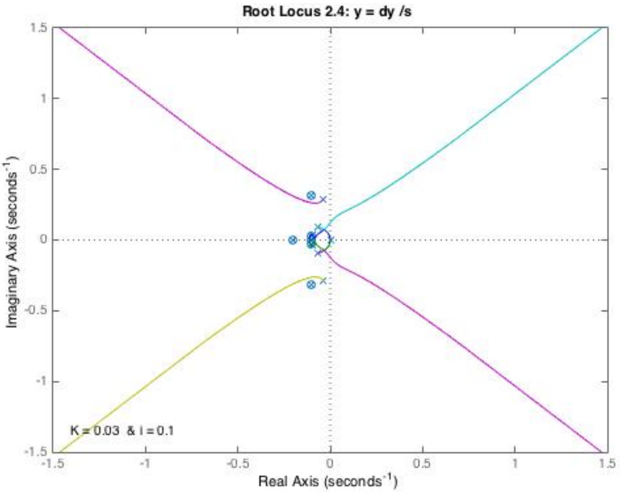


Figure: Root Locus Plot (Roll & Pitch channel)

Appendix D
Time Specification

- Rise Time: time to go from 10% to 90% of final value
- Settling Time: time to get within 1% of final value and stay there