Iftach Naftaly

QFT Controller Design for 2D Quadrotor

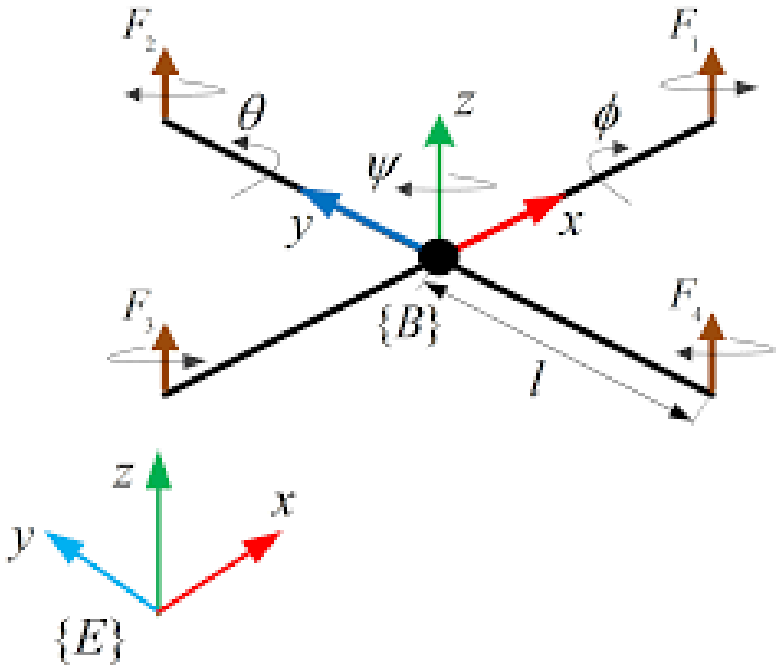


Table of Contents:

1. Problem Definition:

We wish to model a quadrotor movement in a 2D plane, as seen in figure 1.

The quadrotor builds from two motors and a frame. The total mass of the quad is $\mathbf{m}$, and the length between each motor and the C.G of the quad is $\mathbf{L}$. The momentum of inertia with respect to the body frame is $\mathbf{I}$.

Define **e**, the inertial coordinate system in some fixed place and **b**, a coordinate system which its origin located in the quad C.G, $\boldsymbol{x_b}$ pointing along with the quad's frame for motor 1 and $\boldsymbol{y_b}$ perpendicular to the quad's frame. Also, $\mathbf{g}$ is the gravity vector which pointing in $\boldsymbol{y_b}$ direction and its magnitude fixed.

Suppose we can control the quadrotor using two controllers: $\boldsymbol{u_1}$ – supply thrust in the $\boldsymbol{y_b}$ direction and $\boldsymbol{u_2}$ – supply momentum in the $\boldsymbol{z_b}$ direction, where $\boldsymbol{z_b}$ it's the complementary vector from **b** coordinate system, and can be found by the right-hand rule.

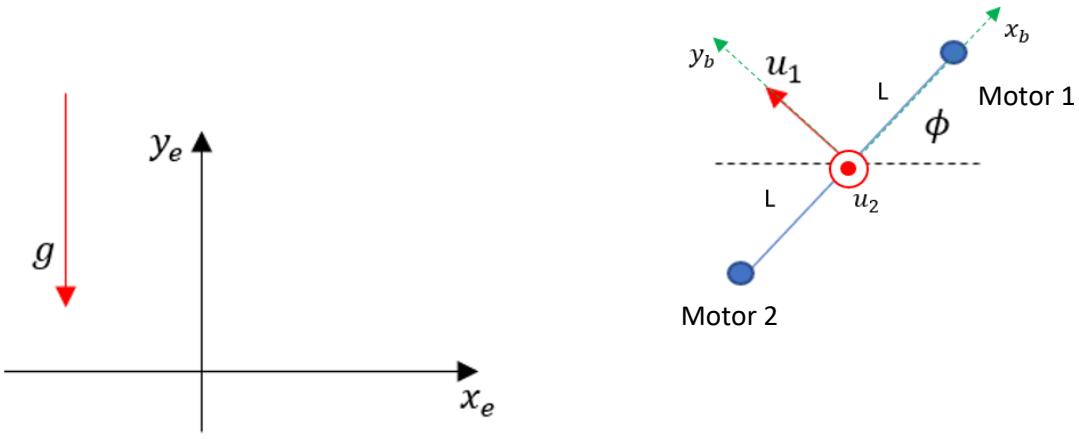


Figure 1 – Definitions

2. Equations of Motion:

The rotation matrix between the two coordinate systems:

(1)
$$ {}^{b}R_{e} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} $$

and using the rotation matrix to convert the controller's forces and momentum to **e** coordinate system:

(2) $$u_{1,b}=\begin{bmatrix}0\\u_1\\0\end{bmatrix}\rightarrow u_{1,e}=^{b} R_e u_{1,b}=\begin{bmatrix}\cos(\phi) & -\sin(\phi) & 0\\ \sin(\phi) & \cos(\phi) & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}0\\u_1\\0\end{bmatrix}=\begin{bmatrix}-\sin(\phi)u_1\\ \cos(\phi)u_1\\ 0\end{bmatrix}$$

$$u_{2,b}=u_2$$

And the equation of motion:

(3) $$\ddot{x}_e=-\sin(\phi)\frac{u_1}{m}$$
$$\ddot{y}_e=-g+\cos(\phi)\frac{u_1}{m}$$
$$\ddot{\phi}=\frac{1}{I}u_2$$

In matrix form:

(4) $$\begin{bmatrix}\ddot{x}_e\\ \ddot{y}_e\\ \ddot{\phi}\end{bmatrix}=\begin{bmatrix}0\\-g\\0\end{bmatrix}+\begin{bmatrix}-\sin(\phi)\frac{1}{m} & 0\\ \cos(\phi)\frac{1}{m} & 0\\ 0 & \frac{1}{I}\end{bmatrix}\begin{bmatrix}u_1\\u_2\end{bmatrix}$$

Defining the state-variable:

(5) $$x=\begin{bmatrix}x_1\\x_2\\x_3\\x_4\\x_5\\x_6\end{bmatrix}=\begin{bmatrix}x_e\\y_e\\\phi\\\dot{x}_e\\\dot{y}_e\\\dot{\phi}\end{bmatrix}$$

3. Transform to State – Space:

Now we can use the state-variable:

(7)
$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ \ddot{x}_e \\ \tilde{\ddot{y}}_e \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ -\sin(\phi)\frac{u_1}{m} \\ \cos(\phi)\frac{u_1}{m} \\ \frac{u_2}{I} \end{bmatrix}$$

And similarly:

(8)
$$\dot{x} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{A} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -\sin(\phi)\frac{1}{m} & 0 \\ \cos(\phi)\frac{1}{m} & 0 \\ 0 & \frac{1}{I} \end{bmatrix}}_{B(x)} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

We can notice that this is a non-linear model. Suppose we want to control the X-Y quadrotor's position:

(9)
$$\dot{x} = Ax + B(x)u$$
$$y = cx$$
$$c = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

4. V – Transformation:

To deal with the non-linearity of the system, we define the following transformation:

$$v_{11} = -\sin(\phi)u_1$$
$$v_{12} = \cos(\phi)u_1 \quad (10)$$
$$u_1 = \frac{-v_{11}}{\sin(\phi)} = \frac{v_{12}}{\cos(\phi)}$$

The equivalent system for the quad position will be:

$$\ddot{x} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 \\ -g \end{bmatrix} + \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} \quad (11)$$

While ϕ can be calculated from:

$$\tan(\phi) = \frac{-v_{11}}{v_{12}} \quad (12)$$

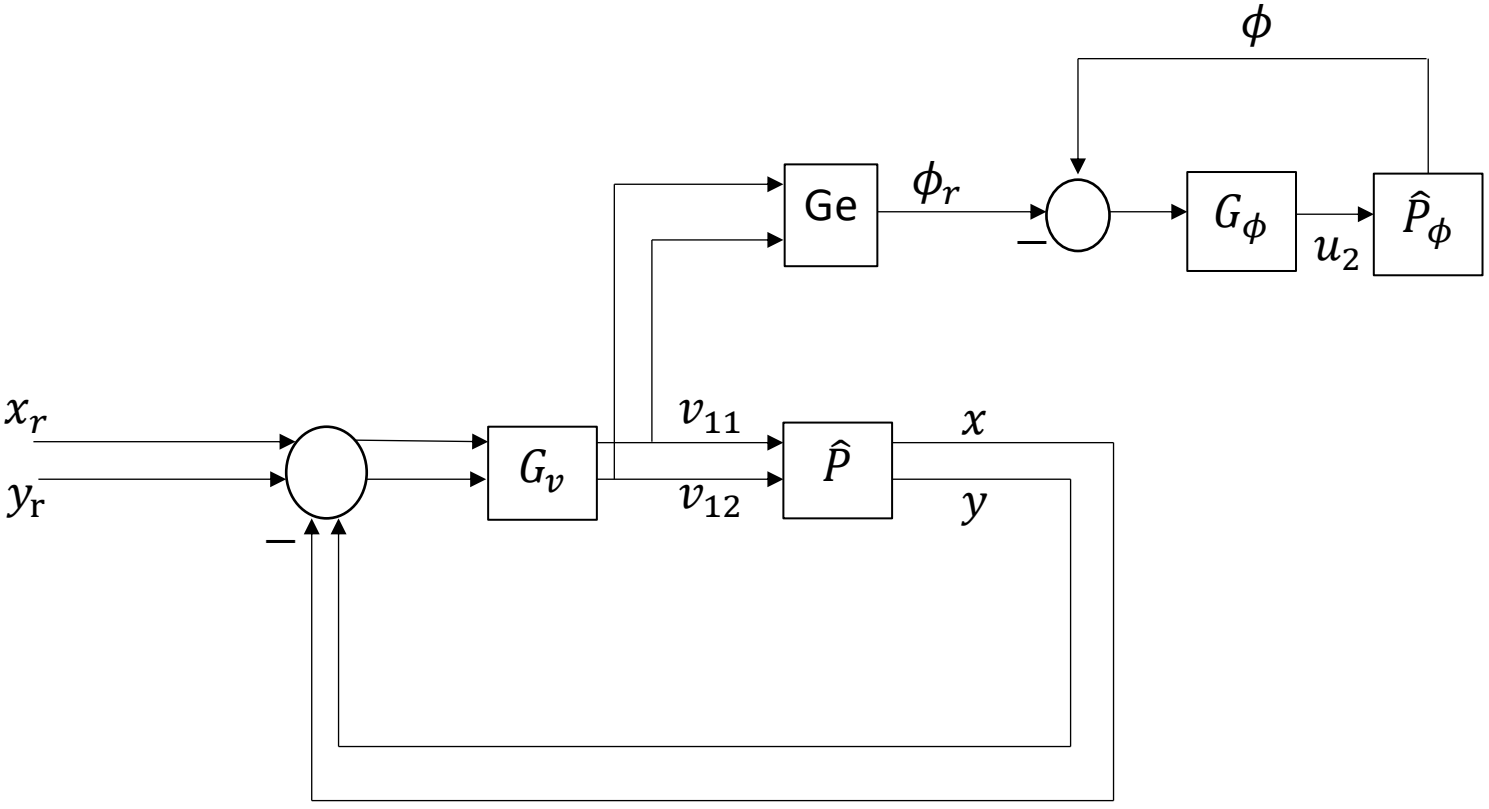


Figure 2 – V Transformation

$\hat{P}$ - Is the plant after the V transformation, G_v is the controller of v_1, v_2. G_ϕ is the controller of ϕ and Ge is a geometric function defined in eq.12.

To reach the new system's equations, we define:

(13)
$$\begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \\ \hat{x}_4 \end{bmatrix} = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix}$$

And it's derivative:

(14)
$$\dot{\hat{x}} = \begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \\ \dot{\hat{x}}_3 \\ \dot{\hat{x}}_4 \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \hat{x}_3 \\ \hat{x}_4 \\ \ddot{x} \\ \ddot{y} \end{bmatrix}$$

So, the state-space implementation will be:

(15)
$$\dot{\hat{x}} = \begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \\ \dot{\hat{x}}_3 \\ \dot{\hat{x}}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \\ \hat{x}_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1/m & 0 \\ 0 & 1/m \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix}$$

5. Motor and Load Model:

The quadrotor model consists of 2 motors, each with its propeller. We can model the motor-propeller system as follow:

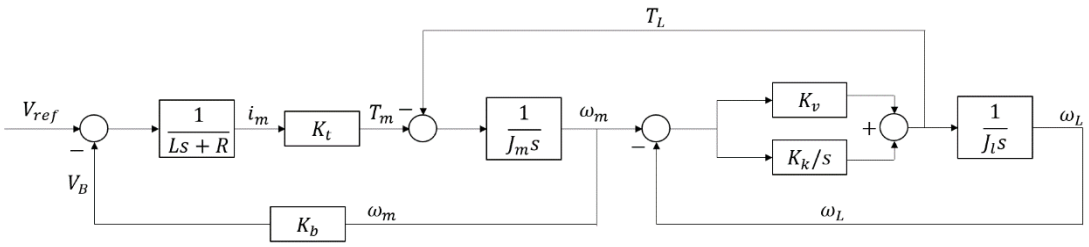


Figure 3 – Motor and Propeller Model

Where R, L – Motor's resistor and inductor, K_t – torque constant, K_b – back-emf constant, J_m - motor's shaft momentum of inertia, K_v – dynamical stiffness of the propeller, K_k static stiffness of the propeller and J_l – prop momentum of inertia.

The input to the system is a PWM signal and the output is the propeller angular speed.

The thrust and the moment applied by each motor is given by:

(16)
$$F = k_f \cdot \omega_l^2$$
$$M = k_m \cdot \omega_l^2$$

k_f and k_m are constants related to the surroundings of the system and the propeller's geometric constants.

Usually, the loop is closed by current controllers called *ESC (Electronic Speed Control)* [1].*ESC*'s are off–the–shelves products with limited configurations option, and they are used to close the current loop in relatively high frequency.

We can approximate the dynamic of this system to 1st-order transfer function as shown in [2], with time constants τ_1 and τ_2 which has some uncertainty caused by non – accurate measurements of the motor's constants:

(17)
$$u_{1,cmd}(s) = \frac{1}{\tau_1 s + 1} u_1$$
$$u_{2,cmd}(s) = \frac{1}{\tau_2 s + 1} u_2$$

6. QFT Controller Design for X – Y Loops:

Suppose we have uncertainty in the mass of the quadrotor, which can occur due to different batteries size used in-flight, and uncertainty in the motor's dynamical model time constant τ_1:

$$m \in (0.85, 1.15)\ [kg]$$
$$\tau_1 \in (0.05, 0.1)\ [s] \quad (18)$$

And we want the X – Y loop to achieve the following servo conditions:

1. $T_r \in [0.15, 0.5]\ [sec]$
2. $M \leq 15\%$
3. $T_s = 0.85\ [sec]$

And the sensitivity condition:

$|S| < 6\ \ [dB]$

Using the QFT method we can find a controller that answers the demands. With openqsyn we can develop the controller:

The conditions:

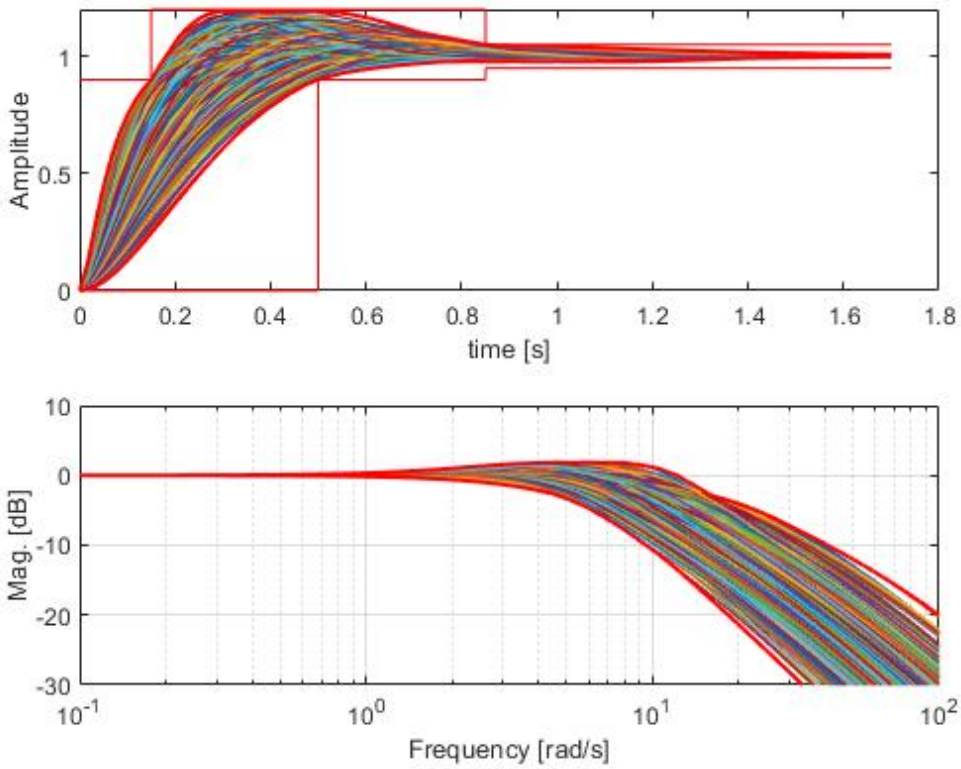


Figure 4 – X-Y Loop Conditions

H-S Bounds:

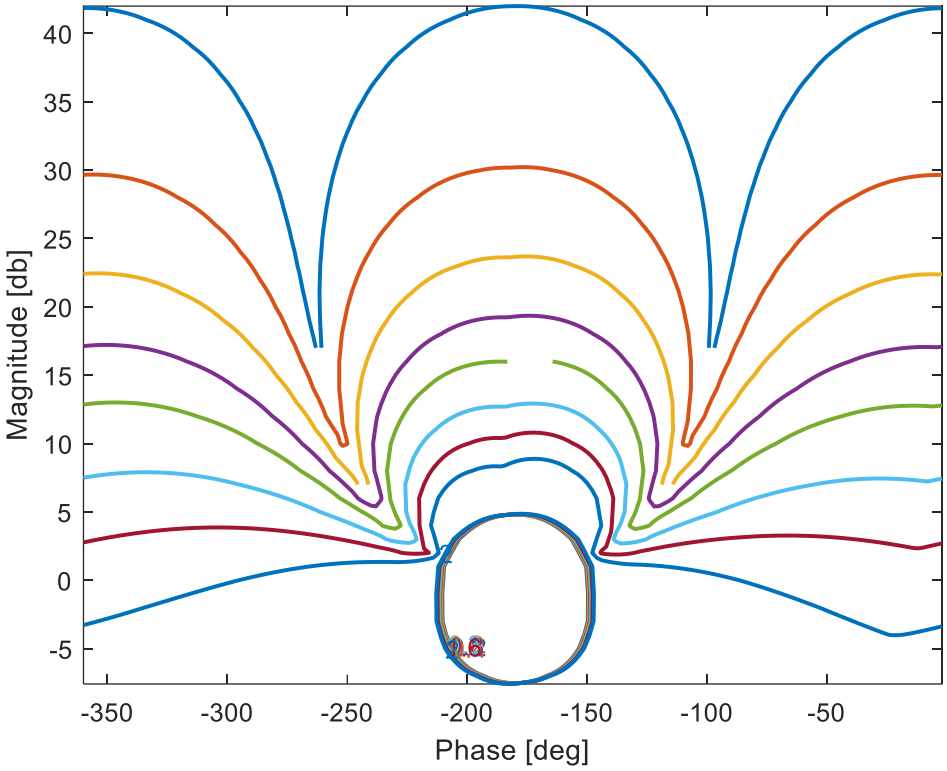


Figure 5 – X-Y H-S Bounds

Using Loop Shaping GUI from openqsyn we can plan the following controller:

(19)
$$\hat{G}_{11,12} = 0.178\frac{s^2\left(s/0.013+1\right)}{s^2\left(s/1e9+1\right)}$$

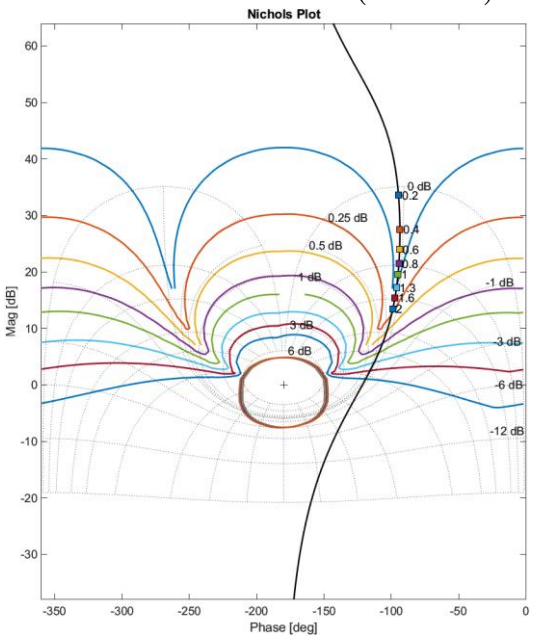


Figure 6 – X-Y Open Loop Nichols Chart

And the close – loop bode diagram with the pre-filter:

(20)
$$F = \frac{1}{0.1s+1}$$

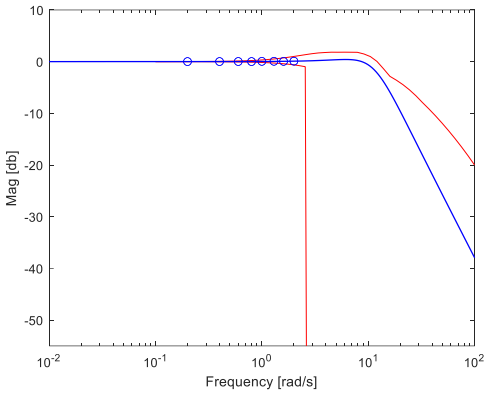


Figure 7 – X-Y Close Loop Bode Chart

We can verify the results with simulation:

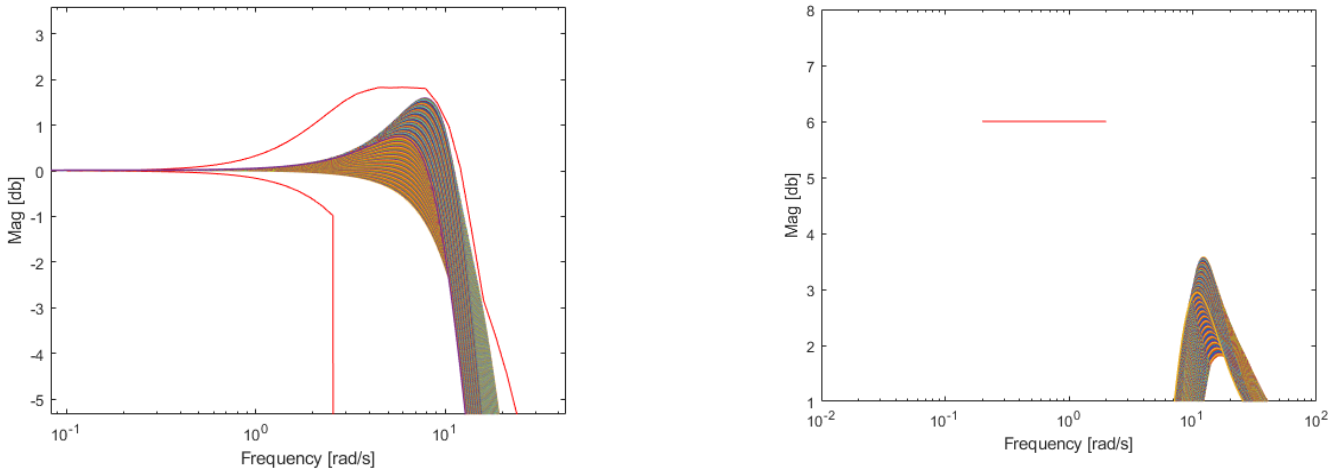


Figure 8 – X-Y Close Loop Simulations

7. QFT Controller Design for ϕ Loop:

To command the right setpoints to the X-Y loop, we need that the ϕ loop will be much faster the X-Y loop because we assume to have the correct ϕ.

ϕ loop has the following uncertainties:

(21)
$$I \in [0.4, 0.6]\left[kg \cdot m^2\right]$$
$$\tau_2 \in [0.05, 0.07][\sec]$$

And the following conditions:

1. $T_r \in [0.04, 0.08]\ [sec]$
2. $M \leq 25\%$
3. $T_s = 0.11\ [sec]$

And the sensitivity condition:

$|S| < 8 \quad [dB]$

Using the same process, we can achieve:

(22)
$$\hat{G}_\phi = 0.5\frac{s^2\left(s/0.0895+1\right)\left(s/57.735+1\right)\left(s/0.077+1\right)}{s^2\left(s/0.182+1\right)\left(s/173.2+1\right)\left(s/100000+1\right)}$$

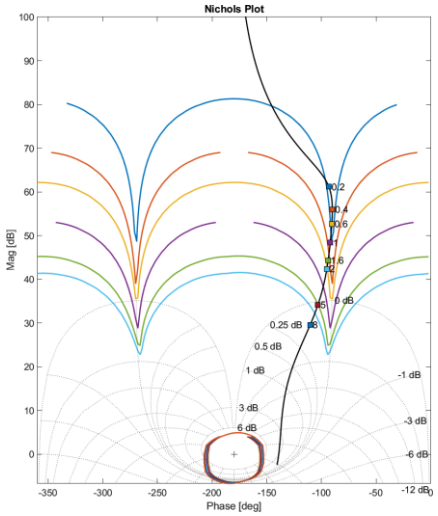


Figure 9 –ϕ Open Loop Nichols

Pre-filter:

$$F = \frac{1}{0.035 \cdot s + 1} \tag{23}$$

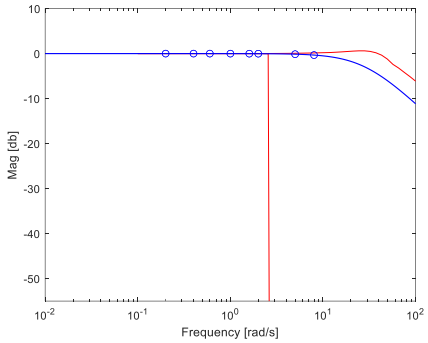


Figure 10 –ϕ Close Loop Bode

We can see that the ϕ loop has ~300 $\left[\frac{rad}{s}\right]$ cut-off frequency, while X-Y cut-off is ~10 $\left[\frac{rad}{s}\right]$. The ϕ process was similar to [3].

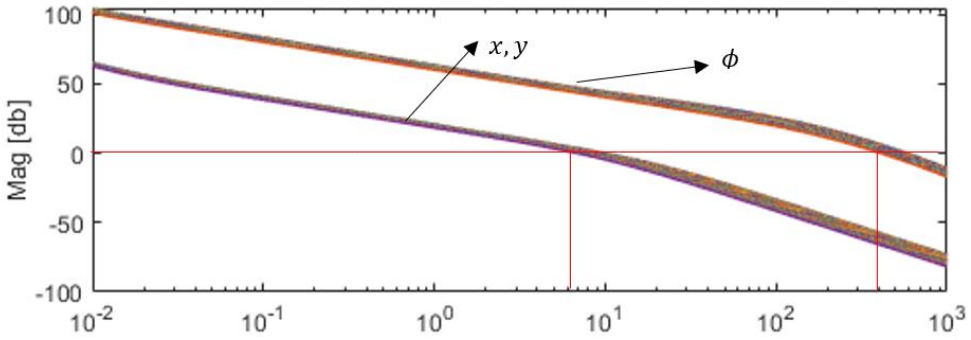


Figure 11 –ϕ vs X-Y Open Loop Bode Magnitude

8. Anti – Reset Windup:

Since the control inputs are bound, to maintain the loop properties with the saturation effects on it, there is a need to use Anti – Reset – Windup. We Chose to use the “Non – Interfering” method [4] because with this method we can still use the planned controller without the need to modify it like [5] – not the best way after the V transformation. While the control signal is not saturating, the system uses the usual structure of the controller.

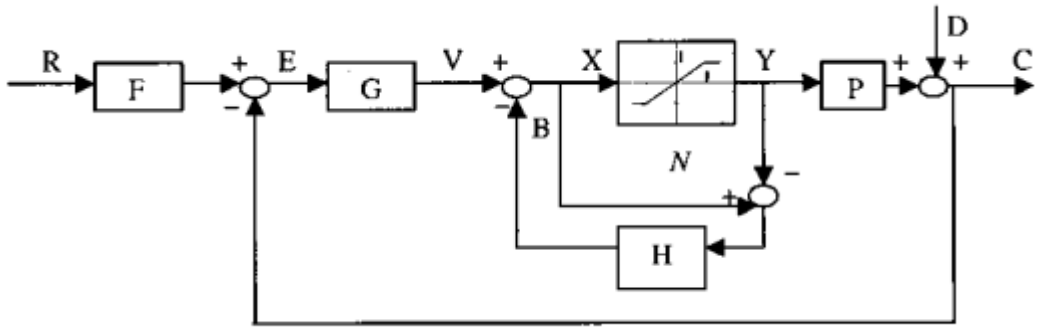


Figure 12 – Non – Interfering ARW Model

H can be found in the following way:

Because the planned controllers are proper and the plants are stable with two integrators, we can choose:

(24) $$L_n(s) = 10/s^2(s+1)$$

Since L_n needs to hold the same number of integrators as the plant. H is given by:

(25) $$H(s) = \frac{-1.34s^3 - 2.7s^2 - 0.3758s + 0.98}{s^2\left(0.001s^5 + 0.1s^4 + 0.2s^3 + 0.1s^2 + 1s + 1\right)}$$

We return the same process for ϕ loop. More information in appendix 2.

The controller’s limits are:

(26) $$u_1 \in [0,25]\left[m/s^2\right]$$
$$u_2 \in [-10,10][m \cdot N]$$

9. Simulink Control Simulation:

Once we finish designing the controllers and plan the ARW, we can build a Simulink model of the system, and simulate the system's step response to step command, under different values of the uncertainties. We will use the following Simulink diagram:

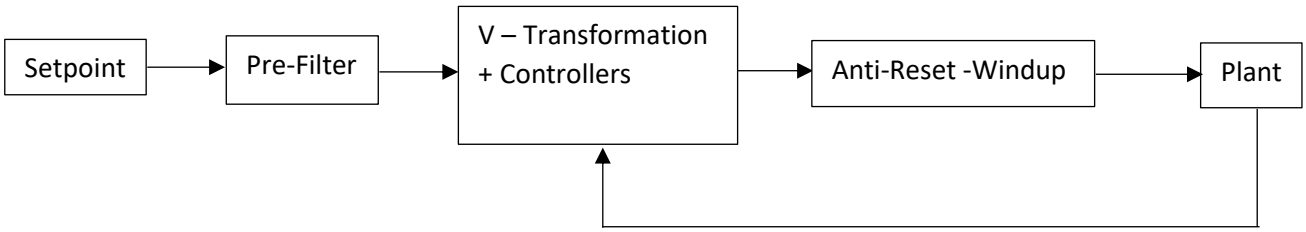


Figure 13 – System's Simulink Mode

Our reference point is [x,y] = [1,1] and the initial conditions are:

$$\dot{x}_0 = 0\left[m/s\right] \quad x_0 = 0\left[m\right]$$
$$\dot{y}_0 = 0\left[m/s\right] \quad y_0 = 0\left[m\right] \tag{27}$$
$$\dot{\phi}_0 = 0\left[rad/s\right] \quad \phi_0 = -\frac{\pi}{4}\left[rad\right]$$

Because we used V – Transformation which contains singular points at a specific angle, we will use the QFT controller only between the two points and not use it to maintain the position near the point. When reaching close enough to the setpoint, we will switch to another position controller.

Results:

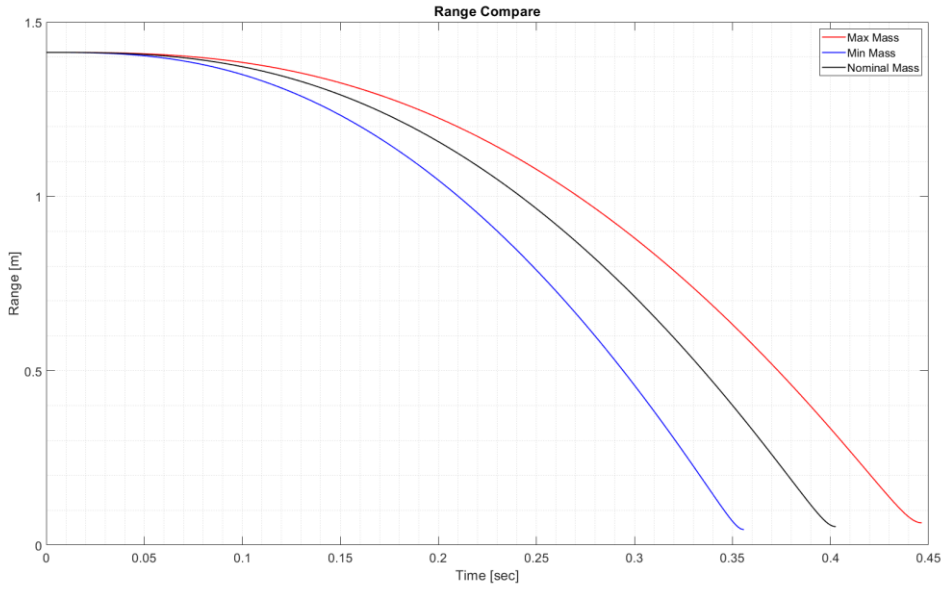


Figure 13 – Quad's Range from terminal location vs Time

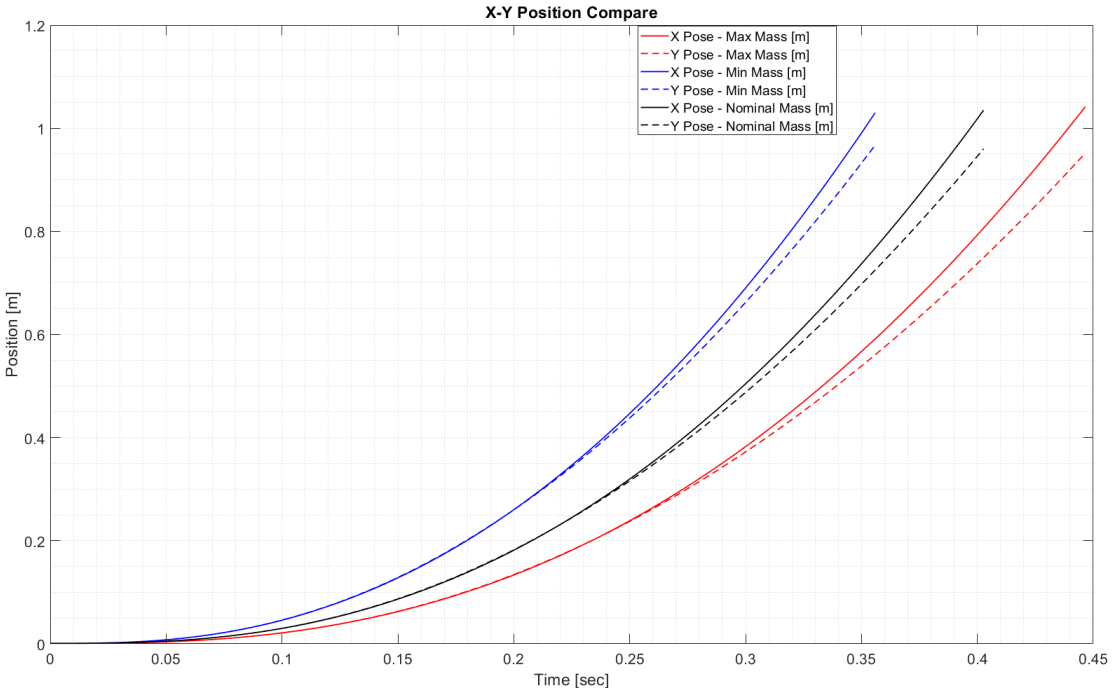


Figure 14 – Quad's Position vs Time

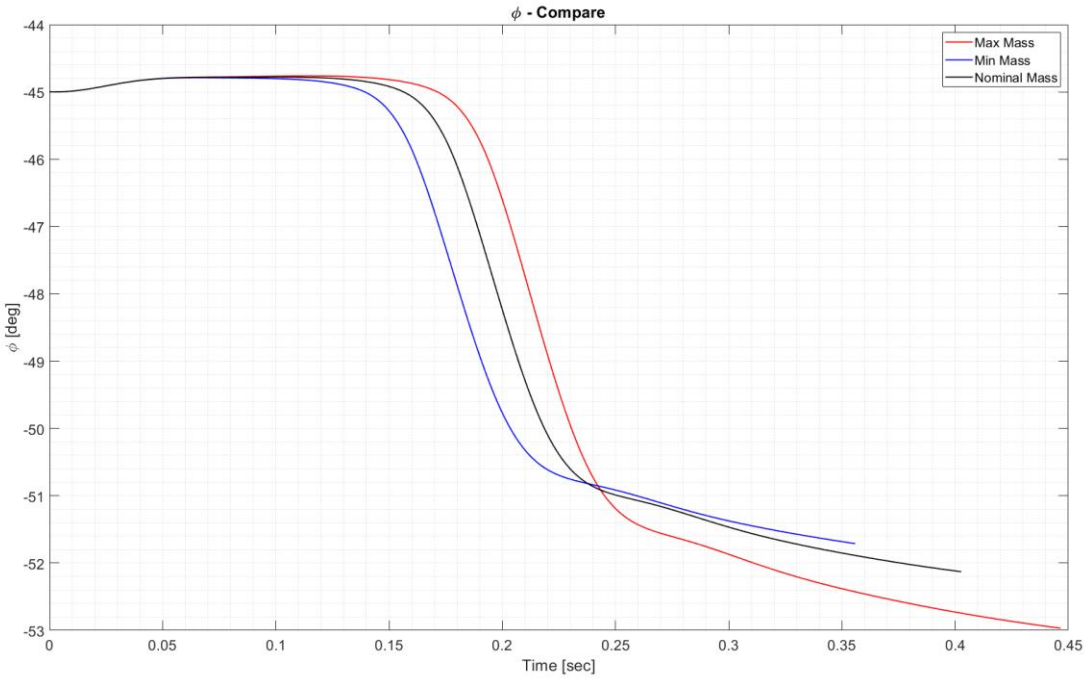


Figure 15 – Quad's ϕ vs Time

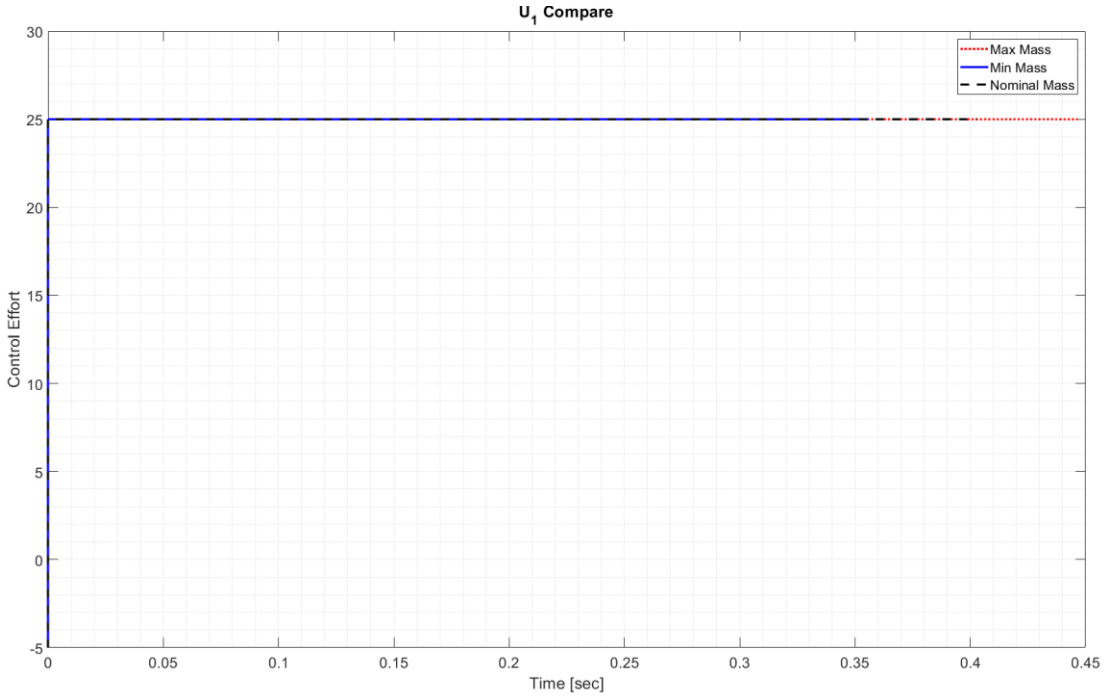


Figure 16 $-u_1$ vs Time

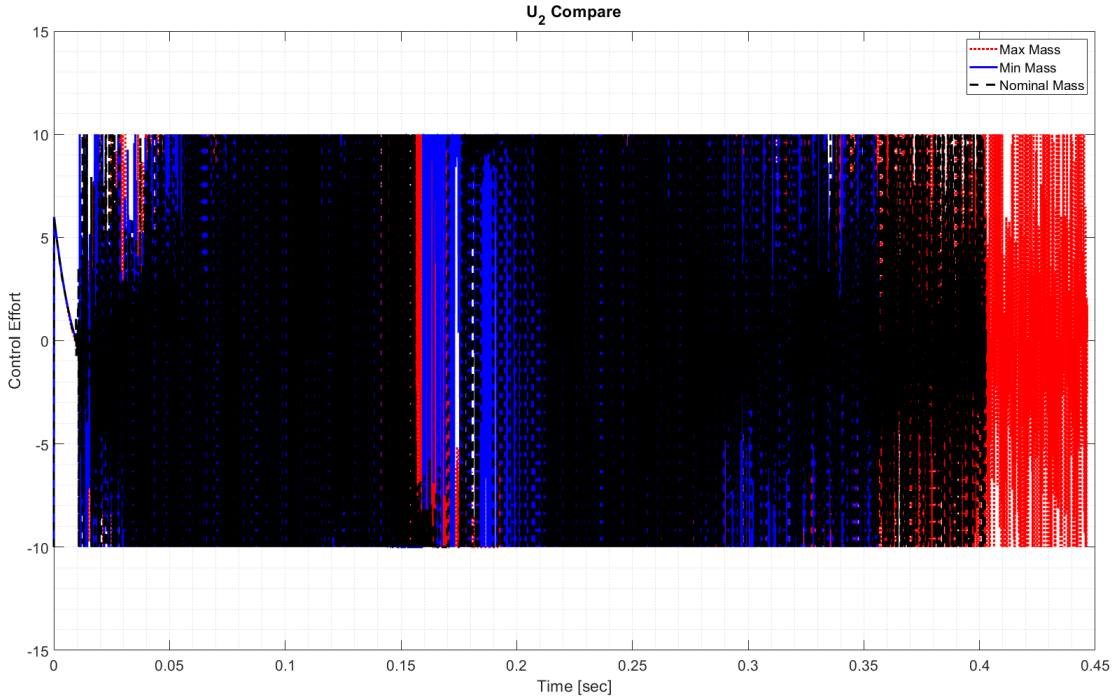


Figure 17 $-u_2$ vs Time

From *Figure 13* we can see that the range is dropping to almost 0 [m] before switching to the position controller. We can also notice that the heaviest configuration takes more time to reach the final position, and the lightest configuration is the fastest and still, for both cases, the system reaches the terminal position in the accepted time. In *Figure 14,* the Same result is shown. From *Figure 15* we notice that ϕ stays almost constant during the flight, with a small shift to higher angles, which results in X channel to be a little bit faster than Y.

During the flight, u_1 uses its full thrust and it constantly on saturation, while u_2 oscillate very aggressively to maintain the desired ϕ.

In conclusion, the designed QFT controller stands for all the conditions with the given uncertainties. There are many gaps between the simulation model and real-time implementation such as the physical model in the simulation does not include drag forces and the aggressive oscillations in u_2 which may damage the motors and I couldn't explain. Nevertheless, I believe that such a model can be implemented on a real quadrotor, with the correct tuning and adjustments.

10. References:

[1] E. R. Magsino, C. M. Dollosa, S. Gavinio, G. Hermoso, N. Laco, and L. A. Roberto, "Stabilizing quadrotor altitude and attitude through speed and torque control of BLDC motors," in *2014 13th International Conference on Control Automation Robotics and Vision, ICARCV 2014*, 2014, pp. 438–443.

[2] Y. Xu and C. Tong, "Quantitative feedback control of a quadrotor," in *IEEE International Symposium on Industrial Electronics*, 2012, pp. 1309–1314.

[3] M. Mardan, M. Esfandiari, and N. Sepehri, "Attitude and position controller design and implementation for a quadrotor," *Int. J. Adv. Robot. Syst.*, vol. 14, no. 3, p. 172988141770924, May 2017.

[4] W. Wu and S. Jayasuriya, "A new QFT design methodology for feedback systems under input saturation," *J. Dyn. Syst. Meas. Control. Trans. ASME*, vol. 123, no. 2, pp. 225–232, Jun. 2001.

[5] A. Berger and P.-O. Gutman, "A new view of anti-windup design for uncertain linear systems in the frequency domain," *Int. J. Robust Nonlinear Control*, vol. 26, no. 10, pp. 2116–2135, Jul. 2016.

Appendix 1:

From [2], We know that the design constraints for $H(s)$ are:

1. $H(s)$ has the same number of integrators as in the controller $G(s)$.
2. $H(s)$ has all LHP poles expect for the integrators.
3. $|L_n(j\omega) - v| > \lambda \; for \; v \in [-a, -1] and \; \forall \; \omega > 0$ for eliminating limit cycle.
4. cutoff frequency of $H(s)$ needs to be higher from the cutoff frequency of the original loop.
5. complementary sensitivity of the close loop with L_n needs to be smaller than λ_1 for all $P(s)$ for some range of ω.
6. $\left|P(j\omega) + \frac{H(j\omega)P(j\omega)}{1+G(j\omega)P(j\omega)}\right| \leq \lambda_2$ for all $P(s)$ for some range of ω.

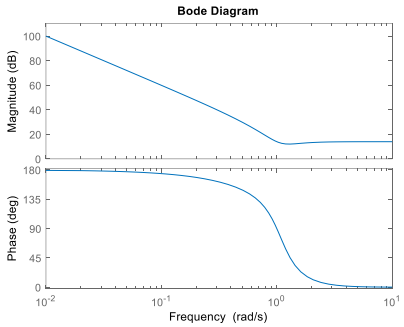
With all these constraints, a normal QFT design can be executed to design $H(s)$.

We've design $G(s)$ to be:

$$\hat{G}_{11,12} = 0.178\frac{s^2\left(s/0.013+1\right)}{s^2\left(s/1e9+1\right)}$$

Meaning 2 integrators, so $H(s)$ must be with 2 integrators also.

For the selected $L_n(s)$, we have $|L_n(j\omega) - v|$ with $v = -5$:



With $\lambda = 6\ [dB]$, the condition is satisfied.

Now we can use openqsyn to QFT design of $H(s)$ under the following conditions:

1. $H(s)$ have two integrators.
2. ω_b of $H(s)$ needs to be higher than 10 [rad/sec].
3. $\left|P(j\omega) + \frac{H(j\omega)P(j\omega)}{1+G(j\omega)P(j\omega)}\right| \leq 8\ [dB]$

The result of the design:

$$H(s) = \frac{-1.34s^3 - 2.7s^2 - 0.3758s + 0.98}{s^2\left(0.001s^5 + 0.1s^4 + 0.2s^3 + 0.1s^2 + 1s + 1\right)}$$ *Not so sure about this result..