

1. Modeling and System Identification

1.1. System Introduction and Basic Knowledge

Nominal Transfer Functions

$$G_{\theta,0} = \frac{1547.4}{(s^2 + 10.12s + 390.4)(s + 5.373)}$$

$$G_{\phi,0} = \frac{2049.5}{(s^2 + 19.03s + 426.2)(s + 6.764)}$$

Unstructured Uncertainty -> Multiplicative Perturbation

$$W_{\theta}(s) = \frac{1659.6(s^2 + 2.868s + 60.44)}{(s + 9.678)(s + 24770)}$$

$$W_{\phi}(s) = \frac{1.9017(s^2 + 3.813s + 91.61)}{(s^2 + 43.53s + 545.3)}$$

Why Not MIMO?

Practical Constraints: Open-loop instability made MIMO identification challenging.

Pilot difficulty during experiments (Section 2.1: "experimental data is better obtained for each channel separately").

Simplification: The authors prioritized feasibility over theoretical MIMO rigor, noting that coupling effects were small near hover conditions.

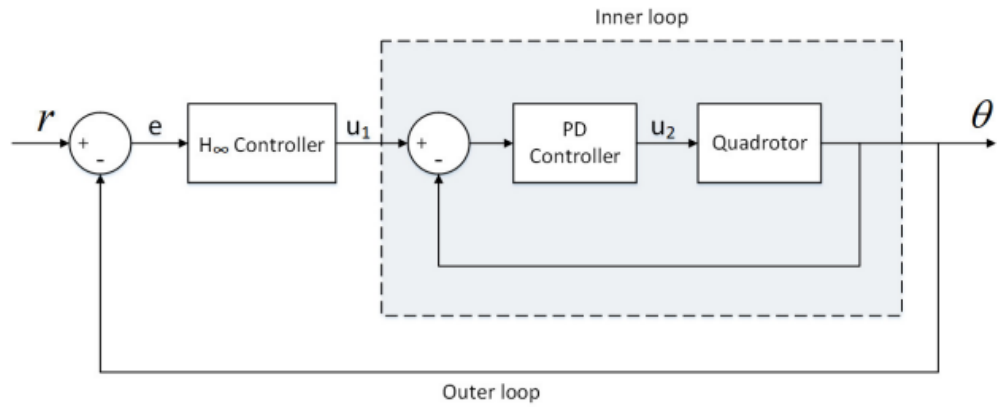
Two-Stage Control Design (Casscade Architecture) -> Used for MIMO Systems, Uncertainty Reduction and Design Improvements

1. Inner Loop (Stabilization):

A basic PID is used during system identification to stabilize the quadrotor and collect experimental data. This PID is crude but sufficient for safe hovering during data collection.

2. Outer Loop (Performance/Robustness):

The H_{∞} controller is designed based on the identified closed-loop model (which includes the PID's effects) and it augments it in a cascade architecture.



```
In [2]: s = tf('s');

% Nominal Transfer functions
G_theta = 1547.4 / ((s^2 + 10.12*s + 390.4)*(s + 5.373))
G_phi = 2049.5 / ((s^2 + 19.03*s + 426.2)*(s + 6.764))

% Weighting Functions
W_theta = 1659.6*(s^2 + 2.868*s + 60.44) / ((s + 9.678)*(s + 24770))
W_phi = 1.9017*(s^2 + 3.813*s + 91.61) / (s^2 + 43.53*s + 545.3)
```

G_theta =

```

              1547
-----
s^3 + 15.49 s^2 + 444.8 s + 2098
```

Continuous-time transfer function.

G_phi =

```

              2050
-----
s^3 + 25.79 s^2 + 554.9 s + 2883
```

Continuous-time transfer function.

W_theta =

```

1660 s^2 + 4760 s + 1.003e05
-----
s^2 + 2.478e04 s + 2.397e05
```

Continuous-time transfer function.

W_phi =

```

1.902 s^2 + 7.251 s + 174.2
-----
s^2 + 43.53 s + 545.3
```

Continuous-time transfer function.

1.2. Modeling Uncertain SISO Systems in Matlab

```
In [3]: % Create uncertain systems
delta_theta = ultidyn('delta_theta', [1 1]); % [1 1] -> SISO Approach
delta_phi = ultidyn('delta_phi', [1 1]);
```

```
% Perturbed Transfer Functions
```

```
G_theta_unc = G_theta * (1 + W_theta * delta_theta)
```

```
G_phi_unc = G_phi * (1 + W_phi * delta_phi)
```

G_theta_unc =

Uncertain continuous-time state-space model with 1 outputs, 1 inputs, 5 states.
The model uncertainty consists of the following blocks:

delta_theta: Uncertain 1x1 LTI, peak gain = 1, 1 occurrences

Type "G_theta_unc.NominalValue" to see the nominal value, "get(G_theta_unc)" to see all properties, and "G_theta_unc.Uncertainty" to interact with the uncertain elements.

G_phi_unc =

Uncertain continuous-time state-space model with 1 outputs, 1 inputs, 5 states.
The model uncertainty consists of the following blocks:

delta_phi: Uncertain 1x1 LTI, peak gain = 1, 1 occurrences

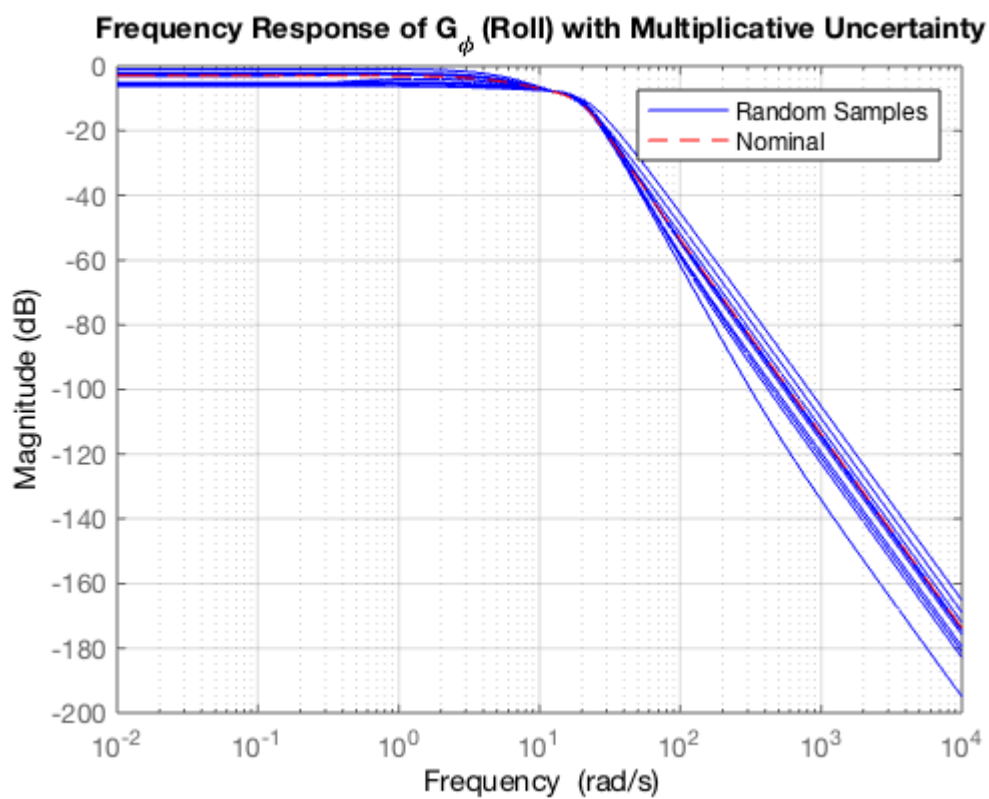
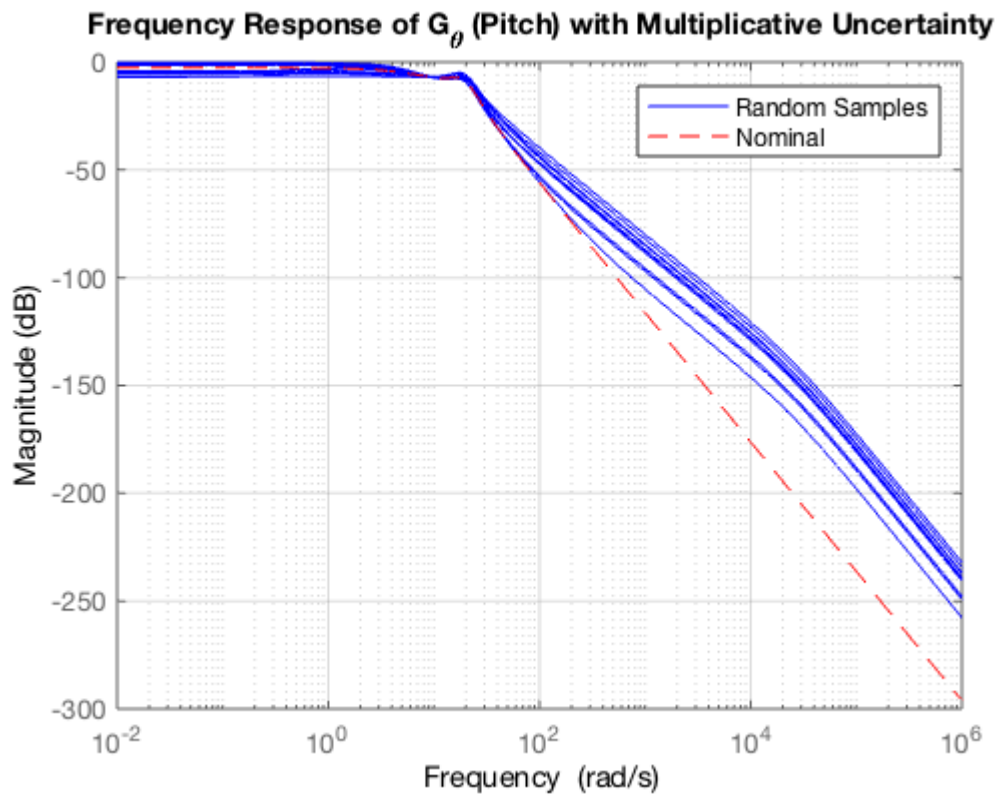
Type "G_phi_unc.NominalValue" to see the nominal value, "get(G_phi_unc)" to see all properties, and "G_phi_unc.Uncertainty" to interact with the uncertain elements.

1.3. Uncertainty Visualization

```
In [4]: % Generate custom number of samples (The default number is 20)
num_samples = 10;
G_theta_samples = usample(G_theta_unc, num_samples);
G_phi_samples = usample(G_phi_unc, num_samples);

% Bode Plots of both Nominal and Uncertain Systems
figure;
bodemag(G_theta_samples, 'b', G_theta, 'r--');
title('Frequency Response of G_{\theta} (Pitch) with Multiplicative Uncertainty');
legend('Random Samples', 'Nominal');
grid on;

figure;
bodemag(G_phi_samples, 'b', G_phi, 'r--');
title('Frequency Response of G_{\phi} (Roll) with Multiplicative Uncertainty');
legend('Random Samples', 'Nominal');
grid on;
```



```
In [5]: omega = logspace(-1, 4, 500);

% Frequency Responses
[mag_theta, phase_theta] = bode(G_theta, omega);
mag_theta = squeeze(mag_theta); % Remove singleton dimensions

[mag_W_theta, ~] = bode(W_theta, omega);
mag_W_theta = squeeze(mag_W_theta);
```

```

[mag_phi, phase_phi] = bode(G_phi, omega);
mag_phi = squeeze(mag_phi);

[mag_W_phi, ~] = bode(W_phi, omega);
mag_W_phi = squeeze(mag_W_phi);

% Uncertainty Bounds
upper_theta = mag_theta .* (1 + mag_W_theta);
lower_theta = mag_theta .* (1 - mag_W_theta);

upper_phi = mag_phi .* (1 + mag_W_phi);
lower_phi = mag_phi .* (1 - mag_W_phi);

% Plotting
figure;

% --- Pitch (G_theta) ---
semilogx(omega, 20*log10(mag_theta), 'b', 'LineWidth', 2); hold on;
semilogx(omega, 20*log10(upper_theta), 'r--', 'LineWidth', 1.5);
semilogx(omega, 20*log10(lower_theta), 'r--', 'LineWidth', 1.5);
title('Nominal Response with Multiplicative Uncertainty Bounds for G_\theta (Pitch)');
xlabel('Frequency (rad/s)');
ylabel('Magnitude (dB)');
legend('Nominal', 'Upper Bound (1 + |W_\theta|)', 'Lower Bound (1 - |W_\theta|)');
grid on;

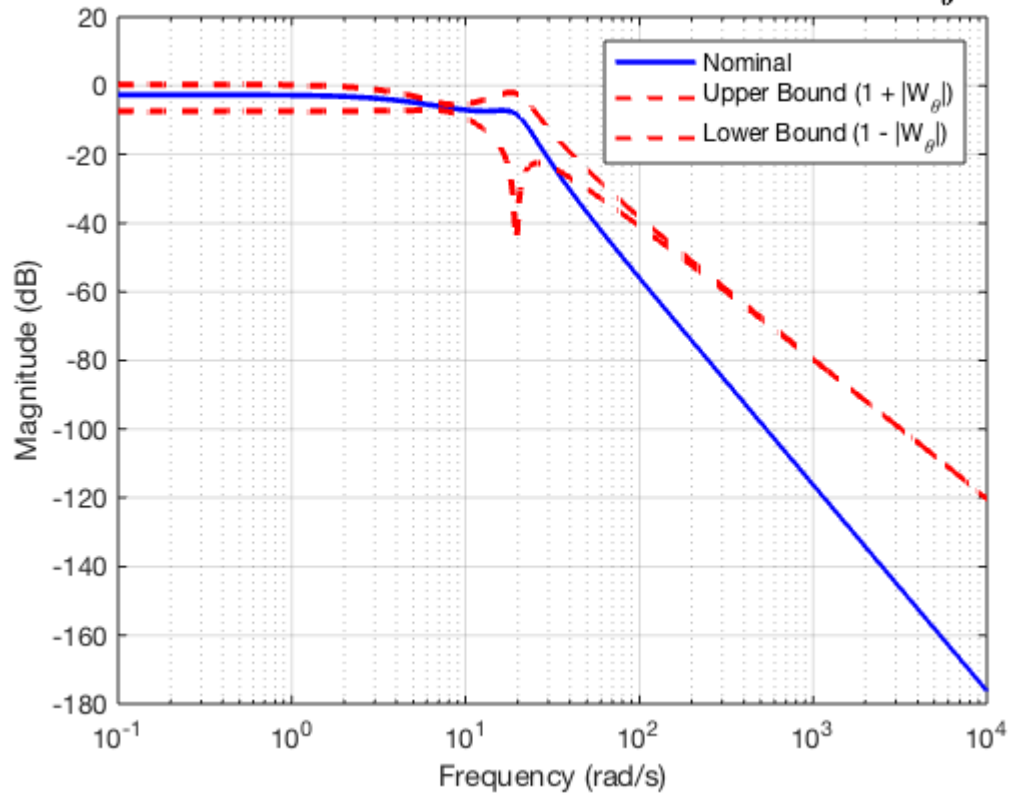
% --- Roll (G_phi) ---
figure;
semilogx(omega, 20*log10(mag_phi), 'b', 'LineWidth', 2); hold on;
semilogx(omega, 20*log10(upper_phi), 'r--', 'LineWidth', 1.5);
semilogx(omega, 20*log10(lower_phi), 'r--', 'LineWidth', 1.5);
title('Nominal Response with Multiplicative Uncertainty Bounds for G_\phi (Roll)');
xlabel('Frequency (rad/s)');
ylabel('Magnitude (dB)');
legend('Nominal', 'Upper Bound (1 + |W_\phi|)', 'Lower Bound (1 - |W_\phi|)');
grid on;

```

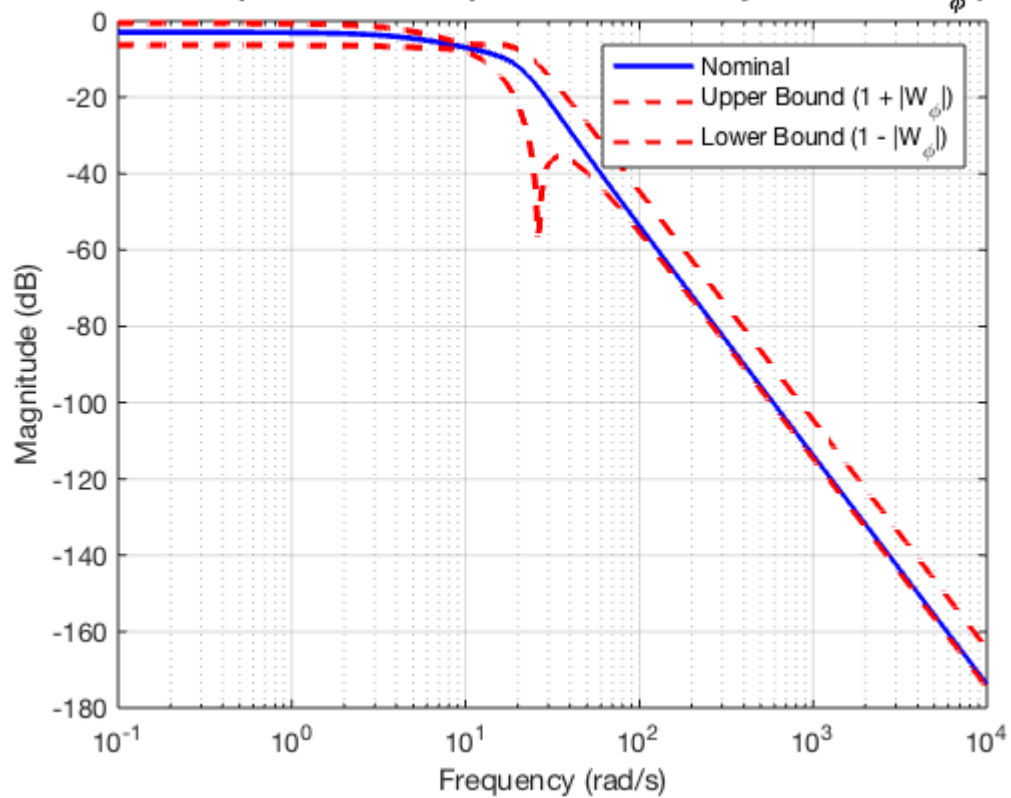
Warning: Imaginary parts of complex X and/or Y arguments ignored

Warning: Imaginary parts of complex X and/or Y arguments ignored

Nominal Response with Multiplicative Uncertainty Bounds for G_θ (Pitch)



Nominal Response with Multiplicative Uncertainty Bounds for G_ϕ (Roll)



1.4. Control Objectives and Performance Specifications

Settling Time $\approx 0.3s$

Overshoot ≈ 0

Steady State Error ≈ 0

2. Robust H_∞ Control Synthesis

2.1. Calculations of Weighting Functions

A) Sensitivity Weighting Function

Assumption: Second Order Closed-Loop System

Closed-Loop Transfer Function (Ideal Function):

$$T_{cl}(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

Peak Overshoot: (Step Input)

$$M_p = \exp\left(-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}\right) \approx 10^{-6}$$

$$\zeta = \frac{-\ln(M_p)}{\sqrt{\pi^2 + (\ln(M_p))^2}}$$

Settling Time:

$$t_s \approx \frac{4}{\zeta\omega_n} \approx 0.3s$$

Solving peak overshoot criterion leads to:

```
In [6]: Mp_desired = 1e-6; % Desired overshoot (e.g., 0.0001%)
zeta_solved = (-log(Mp_desired)) / sqrt(pi^2 + (log(Mp_desired))^2);
disp(['Required Damping Ratio: ', num2str(zeta_solved)]);
```

Required Damping Ratio: 0.97511

Also the natural frequency is obtained from settling time formula:

$$t_s \approx \frac{4}{\zeta\omega_n} \approx 0.3s \Rightarrow \omega_n = \frac{4}{0.3\zeta} = 13.61 \text{ rad. } s^{-1}$$

In the next step, the ideal closed-loop transfer function is calculated:

$$T_{id}(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{185.23}{s^2 + 26.68s + 185.23}$$

Finally, the sensitivity weighting function is as follows:

$$W_s = a \frac{1}{S_{id}(s)} = a \frac{1}{1 - T_{id}(s)} = a \frac{s^2 + 26.68s + 185.23}{s(s + 26.68)}$$

In order to be strictly proper and stable:

$$W_s = a \frac{s^2 + 26.68s + 185.23}{(s + 0.001)(s + 26.68)(s + 1000)}$$

B) Control Effort Weighting Function

$$\|W_u U\|_\infty \leq 1; \|U\|_\infty \leq 20$$

$$\Rightarrow W_u = 0.05$$

$$\text{First Trial} \Rightarrow W_u = 1$$

```
In [7]: % Initial Weights
W_u = 1;
a = 1;
W_s = a * (s^2 + 26.68*s + 185.23) / ((s + 0.001)*(s + 26.68)*(s + 1000));

% Build Generalized Plants
G_theta_p = augw(G_theta , W_s , W_u , W_theta);
G_phi_p = augw(G_phi , W_s , W_u , W_phi);
```

2.2. H_∞ Control Synthesis (Riccati Method)

```
In [8]: % Synthesize  $H_\infty$  Controller
[C_theta , CL1 , gamma1 , info1] = hinfsyn(G_theta_p , 1 , 1 , 'method' , 'ric'
[C_phi , CL2 , gamma2 , info2] = hinfsyn(G_phi_p , 1 , 1 , 'method' , 'ric' , 'D
```

Test bounds: 1.39 <= gamma <= 1.79

gamma	X>=0	Y>=0	rho(XY)<1	p/f
1.58e+00	1.3e-05	0.0e+00	1.208e-18	p
1.48e+00	1.3e-05	0.0e+00	7.284e-17	p
1.44e+00	1.3e-05	0.0e+00	1.020e-16	p
1.41e+00	1.3e-05	0.0e+00	4.871e-17	p
1.40e+00	1.3e-05	0.0e+00	9.412e-17	p

Best performance (actual): 1.4

Test bounds: 1.41 <= gamma <= 1.82

gamma	X>=0	Y>=0	rho(XY)<1	p/f
1.60e+00	1.3e-05	0.0e+00	2.054e-16	p
1.50e+00	1.3e-05	0.0e+00	3.962e-19	p
1.46e+00	1.3e-05	0.0e+00	1.922e-14	p
1.43e+00	1.3e-05	0.0e+00	6.784e-16	p
1.42e+00	1.3e-05	0.0e+00	6.733e-18	p

Best performance (actual): 1.42

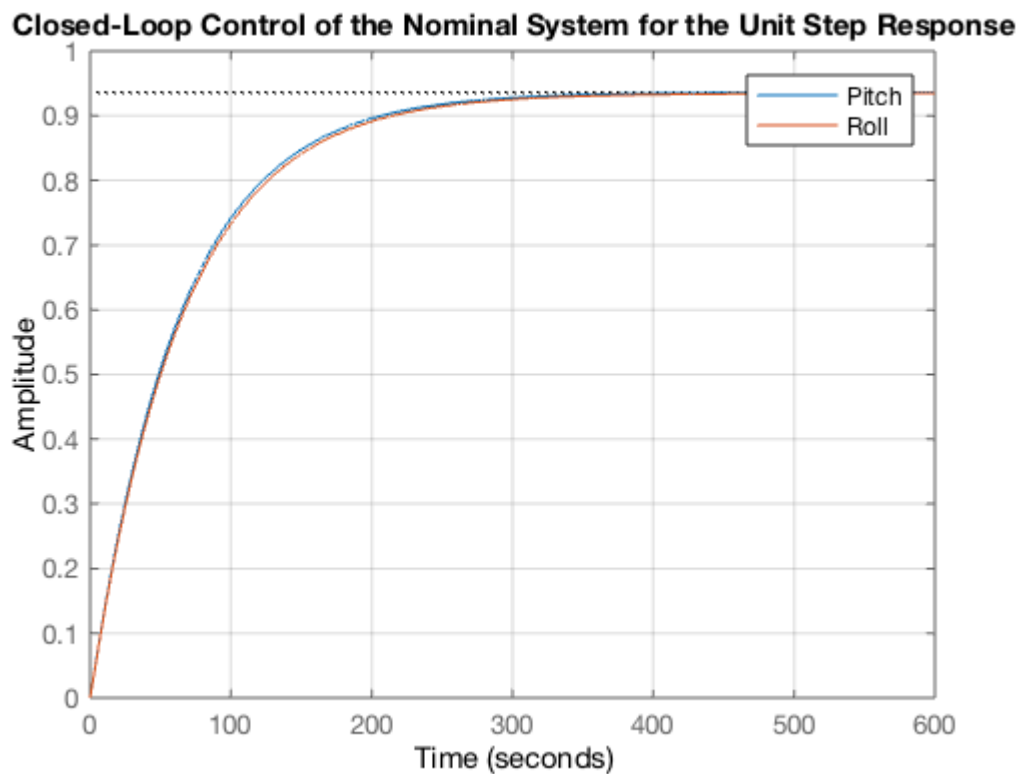
2.3. Nominal Performance Analysis

A) Closed-Loop Response to Step Input

The figures show that the nominal system doesn't reach the final value of the unit step input. It can also be seen that the system response is too slow and also the overshoot is not acceptable for the control effort signal.


```
In [9]: T_theta = feedback(G_theta*C_theta, 1); % Unity feedback
T_phi = feedback(G_phi*C_phi, 1);

% Plot Step Response
hold on;
step(T_theta);
step(T_phi);
title('Closed-Loop Control of the Nominal System for the Unit Step Response');
legend('Pitch' , 'Roll');
grid on;
```

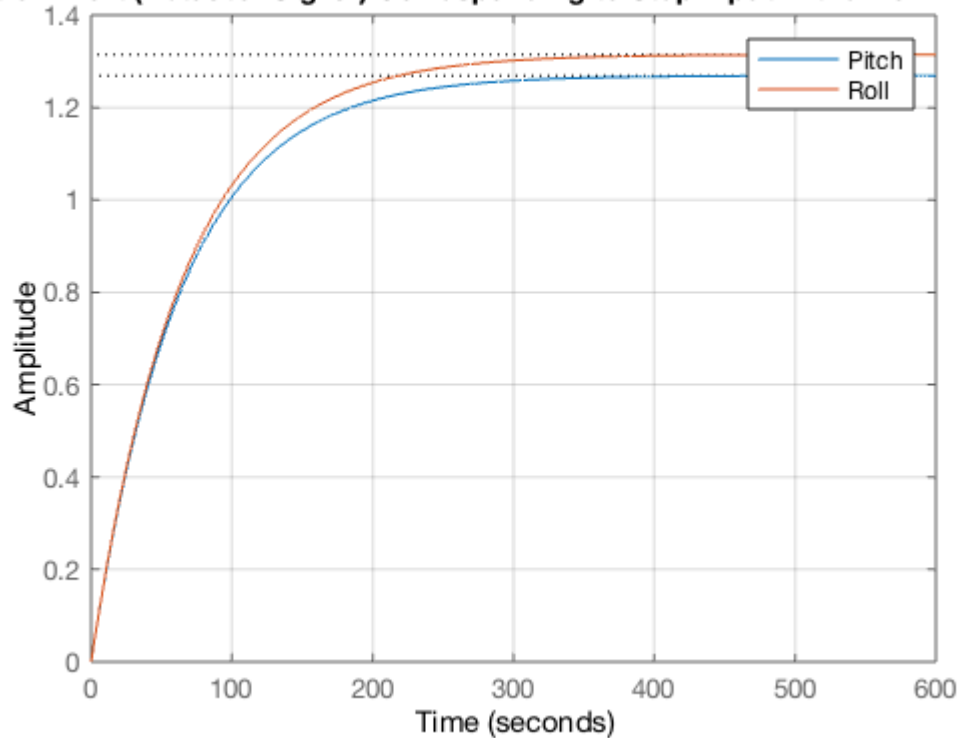


B) Corresponding Control Effort

```
In [10]: U_theta = C_theta / (1 + G_theta * C_theta); % Equivalent to feedback(K, G)
U_phi = C_phi / (1 + G_phi * C_phi);

% Plot Step Response
hold on;
step(U_theta);
step(U_phi);
title('Control Effort (Actuator Signal) Corresponding to Step Input in the Nomin');
legend('Pitch' , 'Roll');
grid on;
```

Control Effort (Actuator Signal) Corresponding to Step Input in the Nominal Sys



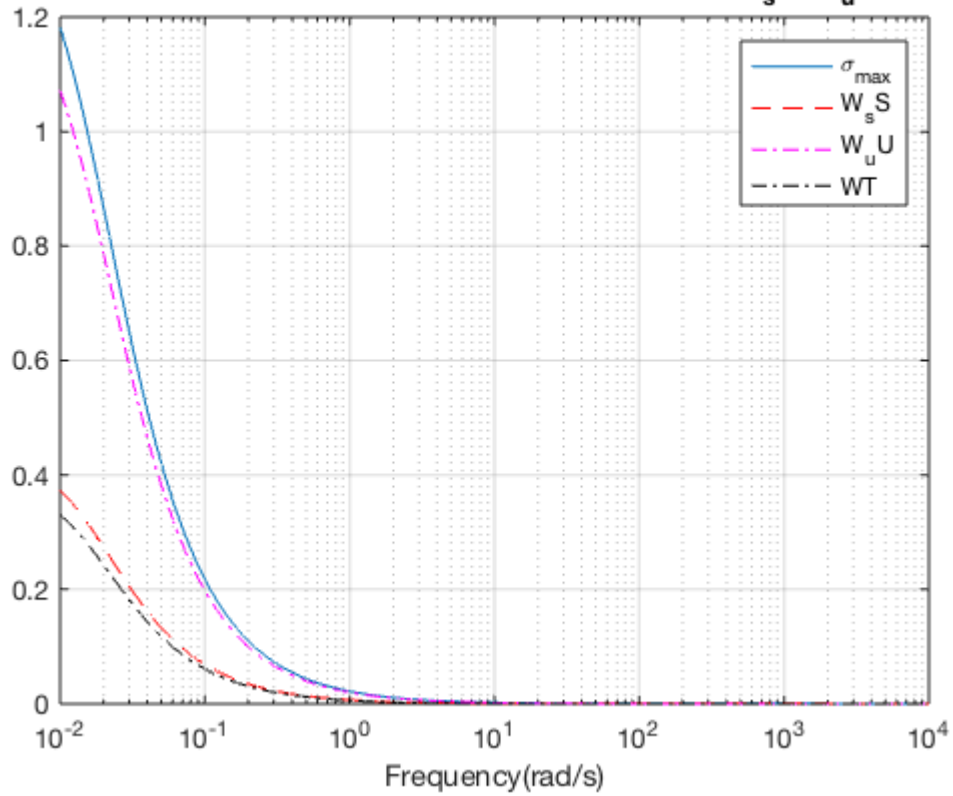
C) Singular Values and Bode Plots

```
In [11]: omega = logspace(-2,4,300);

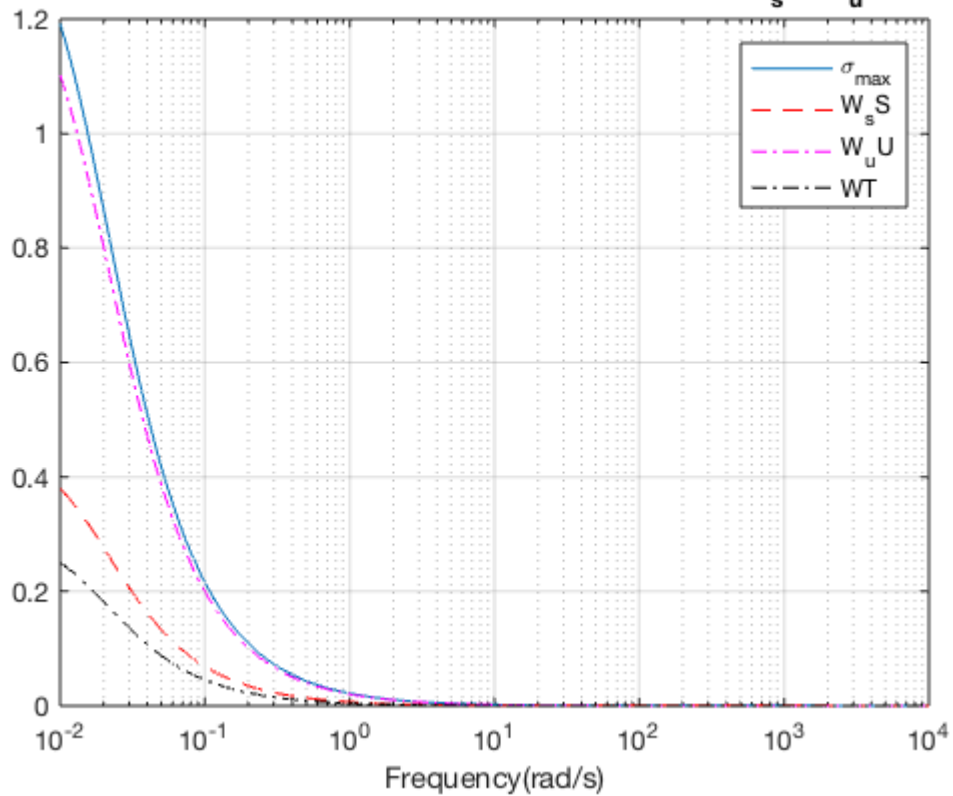
figure;
[mag_g1,phase_g1]=bode(CL1,omega);
sig_g1=sigma(CL1,omega);
semilogx(omega,sig_g1,omega,mag_g1(1,:), '--r',omega,mag_g1(2,:), '-.m',...
omega,mag_g1(3,:), '-.k');
title("Closed-Loop System Singular Values and Bode Plots of W_sS, W_uU, WT (Pitch)");
xlabel("Frequency(rad/s)");
legend('\sigma_{max}', 'W_sS', 'W_uU', 'WT');
grid on;

figure;
[mag_g2,phase_g2]=bode(CL2,omega);
sig_g2=sigma(CL2,omega);
semilogx(omega,sig_g2,omega,mag_g2(1,:), '--r',omega,mag_g2(2,:), '-.m',...
omega,mag_g2(3,:), '-.k');
title("Closed-Loop System Singular Values and Bode Plots of W_sS, W_uU, WT (Roll)");
xlabel("Frequency(rad/s)");
legend('\sigma_{max}', 'W_sS', 'W_uU', 'WT');
grid on;
```

Closed-Loop System Singular Values and Bode Plots of $W_s S$, $W_u U$, WT (Pitch)



Closed-Loop System Singular Values and Bode Plots of $W_s S$, $W_u U$, WT (Roll)



2.4. Improvement Cycle (Pitch)

Trial 2: Increase coefficient of **a** in W_s to decrease settling time

It can be seen that the system response is faster and the max singular value is getting flatter.

```

In [12]: % Modified Weights
W_u = 1;
a = 10;
W_s = a * (s^2 + 26.68*s + 185.23) / ((s + 0.001)*(s + 26.68)*(s + 1000));

% Build Generalized Plant
G_theta_p = augw(G_theta , W_s , W_u , W_theta);

% Synthesize H $\infty$  Controller
[C_theta , CL , gamma , info] = hinfsyn(G_theta_p , 1 , 1 , 'method' , 'ric' , ' ');

T = feedback(G_theta*C_theta, 1);
U = C_theta / (1 + G_theta * C_theta);

figure;
step(T);
title('Closed-Loop Step Response of the Nominal System (Pitch)');
grid on;

figure;
step(U);
title('Control Effort (Actuator Signal) (Pitch)');
grid on;

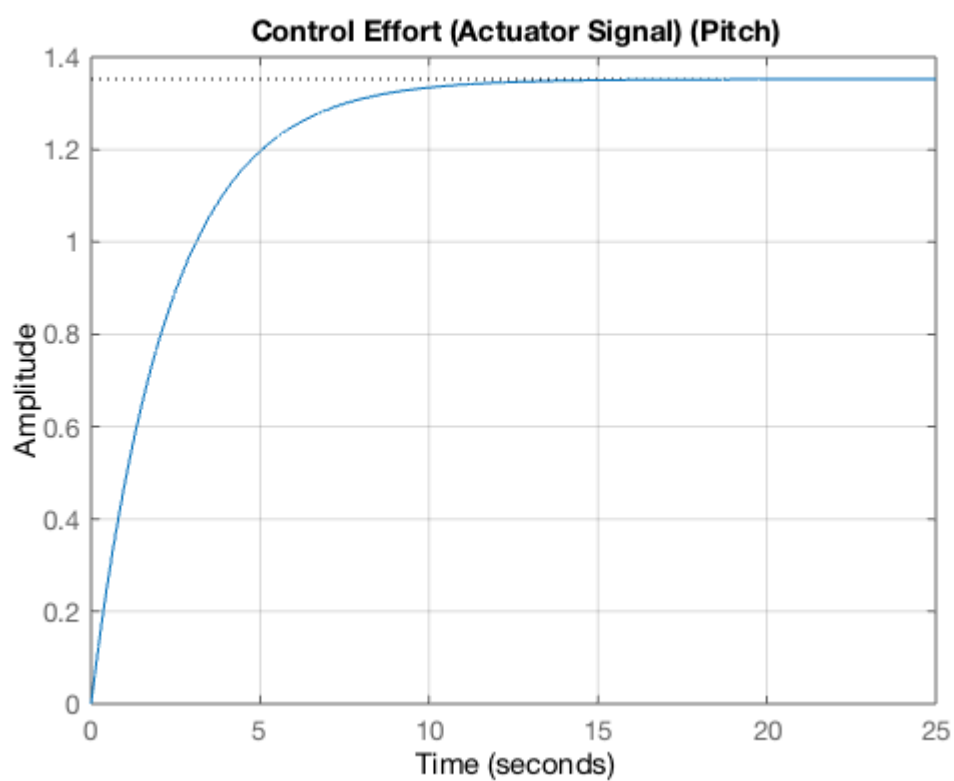
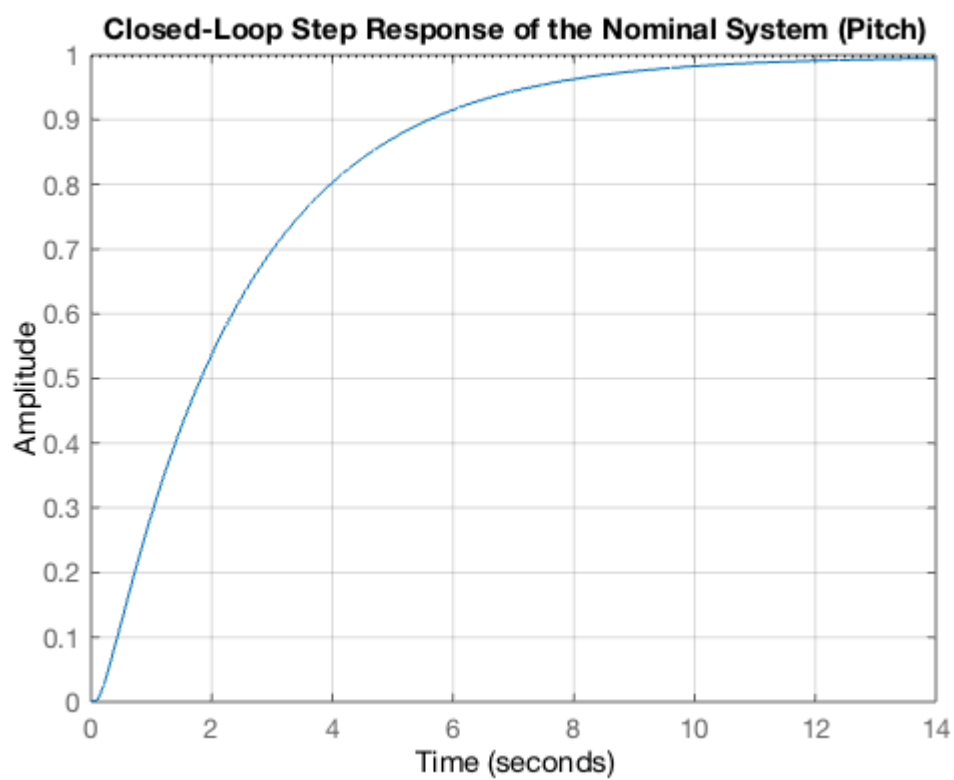
figure;
omega = logspace(-2,4,300);
[mag_g,phase_g]=bode(CL,omega);
sig_g=sigma(CL,omega);
semilogx(omega,sig_g,omega,mag_g(1,:), '--r',omega,mag_g(2,:), '-.m',...
omega,mag_g(3,:), '-.k');
title("Closed-Loop System Singular Values and Bode Plots (Pitch)");
xlabel("Frequency(rad/s)");
legend('\sigma_{max}', 'W_sS', 'W_uU', 'WT');
grid on;

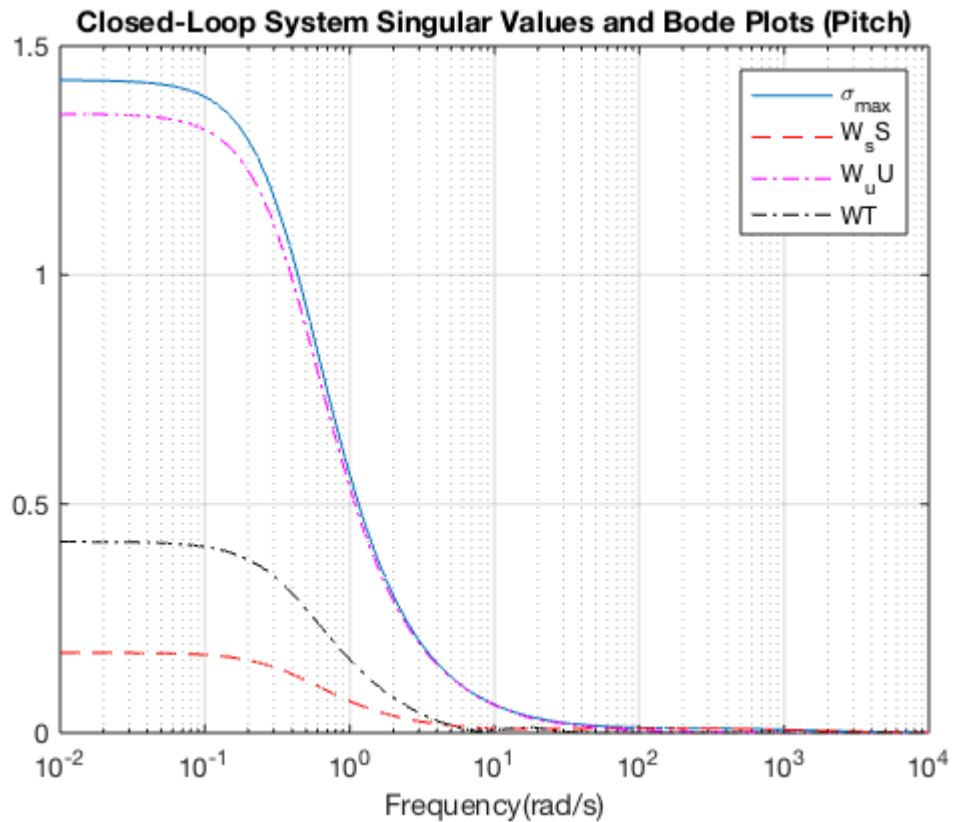
```

Test bounds: 1.42 <= gamma <= 2

gamma	X>=0	Y>=0	rho(XY)<1	p/f
1.68e+00	4.5e-05	0.0e+00	4.359e-18	p
1.54e+00	4.5e-05	0.0e+00	2.604e-16	p
1.48e+00	4.5e-05	0.0e+00	7.971e-17	p
1.45e+00	4.5e-05	0.0e+00	6.079e-18	p
1.43e+00	4.5e-05	0.0e+00	9.899e-18	p
1.43e+00	4.5e-05	0.0e+00	4.704e-18	p

Best performance (actual): 1.43





Trial 3 -> Iterative: Increasing coefficient **a** and decreasing W_u to release limitation on control effort. W_u is fixed with the minimum value and a loop is applied on **a** to reach the settling time of 0.3 seconds and insignificant overshoot.

```
In [13]: % Modified Weights
W_u = 0.05;

a = 0;
settling_time = 100;
while settling_time > 0.35
    W_s = a * (s^2 + 26.68*s + 185.23) / ((s + 0.001)*(s + 26.68)*(s + 1000));

    % Build Generalized Plant
    G_theta_p = augw(G_theta , W_s , W_u , W_theta);

    % Synthesize H∞ Controller
    [C_theta , CL , gamma , info] = hinfsyn(G_theta_p , 1 , 1 , 'method' , 'ric'

    T = feedback(G_theta*C_theta, 1);
    settling_time = stepinfo(T).SettlingTime
    a = a + 50
end
disp("Iteration Completed.")
```

Test bounds: $0 \leq \gamma \leq 2e-06$

γ	$X \geq 0$	$Y \geq 0$	$\rho(XY) < 1$	p/f
$2.00e-07$	$5.4e-06$	$0.0e+00$	$1.410e-12$	p

Best performance (actual): $2.46e-29$

settling_time =

2.0803

a =

50

Test bounds: $0.424 \leq \gamma \leq 0.602$

γ	$X \geq 0$	$Y \geq 0$	$\rho(XY) < 1$	p/f
$5.05e-01$	$6.2e-06$	$0.0e+00$	$5.469e-17$	p
$4.63e-01$	$6.2e-06$	$0.0e+00$	$1.525e-15$	p
$4.43e-01$	$6.2e-06$	$0.0e+00$	$1.113e-17$	p
$4.33e-01$	$6.2e-06$	$0.0e+00$	$1.172e-15$	p
$4.29e-01$	$6.2e-06$	$0.0e+00$	$2.458e-19$	p
$4.26e-01$	$6.2e-06$	$0.0e+00$	$4.355e-16$	p

Best performance (actual): 0.426

settling_time =

1.8228

a =

100

Test bounds: $0.424 \leq \gamma \leq 0.616$

γ	$X \geq 0$	$Y \geq 0$	$\rho(XY) < 1$	p/f
$5.11e-01$	$8.4e-06$	$0.0e+00$	$1.047e-19$	p
$4.65e-01$	$8.4e-06$	$0.0e+00$	$9.874e-18$	p
$4.44e-01$	$8.4e-06$	$0.0e+00$	$3.491e-16$	p
$4.34e-01$	$8.4e-06$	$0.0e+00$	$1.246e-15$	p
$4.29e-01$	$8.4e-06$	$0.0e+00$	$9.109e-16$	p
$4.26e-01$	$8.4e-06$	$0.0e+00$	$5.226e-19$	p

Best performance (actual): 0.426

settling_time =

1.7465

a =

150

Test bounds: 0.424 <= gamma <= 0.64

gamma	X>=0	Y>=0	rho(XY)<1	p/f
5.21e-01	1.1e-05	0.0e+00	2.205e-18	p
4.70e-01	1.1e-05	0.0e+00	6.207e-18	p
4.46e-01	1.1e-05	0.0e+00	3.239e-16	p
4.35e-01	1.1e-05	0.0e+00	2.921e-17	p
4.29e-01	1.1e-05	0.0e+00	9.075e-17	p
4.27e-01	1.1e-05	0.0e+00	7.735e-17	p

Best performance (actual): 0.427

settling_time =

1.4183

a =

200

Test bounds: 0.424 <= gamma <= 0.671

gamma	X>=0	Y>=0	rho(XY)<1	p/f
5.33e-01	1.4e-05	0.0e+00	9.857e-17	p
4.75e-01	1.4e-05	0.0e+00	5.479e-16	p
4.49e-01	1.4e-05	0.0e+00	6.745e-18	p
4.36e-01	1.5e-05	0.0e+00	1.082e-15	p
4.30e-01	1.5e-05	0.0e+00	1.378e-18	p
4.27e-01	1.5e-05	0.0e+00	2.151e-19	p

Best performance (actual): 0.427

settling_time =

1.3634

a =

250

Test bounds: 0.424 <= gamma <= 0.708

gamma	X>=0	Y>=0	rho(XY)<1	p/f
5.48e-01	1.7e-05	0.0e+00	5.877e-16	p
4.82e-01	1.8e-05	0.0e+00	3.350e-16	p
4.52e-01	1.8e-05	0.0e+00	1.284e-16	p
4.38e-01	1.8e-05	0.0e+00	6.879e-17	p
4.31e-01	-3.3e+03 #	-3.6e-20	9.919e-17	f
4.34e-01	-3.4e+04 #	-4.3e-13	5.148e-14	f

Best performance (actual): 0.438

settling_time =

1.0648

a =

300

Test bounds: 0.424 <= gamma <= 0.748

gamma	X>=0	Y>=0	rho(XY)<1	p/f
5.63e-01	2.0e-05	0.0e+00	5.052e-16	p
4.89e-01	2.1e-05	0.0e+00	7.304e-18	p
4.55e-01	2.2e-05	0.0e+00	5.723e-16	p
4.39e-01	-1.6e+03 #	-5.7e-17	1.741e-19	f
4.47e-01	-3.7e+03 #	-1.2e-17	2.969e-19	f
4.51e-01	-9.9e+03 #	-2.2e-16	3.564e-15	f

Best performance (actual): 0.455

settling_time =

1.0137

a =

350

Test bounds: 0.424 <= gamma <= 0.791

gamma	X>=0	Y>=0	rho(XY)<1	p/f
5.79e-01	2.3e-05	0.0e+00	4.049e-16	p
4.95e-01	2.4e-05	0.0e+00	1.545e-17	p
4.58e-01	-1.4e+03 #	-1.8e-17	2.975e-15	f
4.76e-01	-5.4e+03 #	-1.4e-13	5.087e-20	f
4.86e-01	2.4e-05	0.0e+00	1.032e-16	p
4.81e-01	-1.7e+04 #	-7.0e-16	2.710e-14	f

Best performance (actual): 0.486

settling_time =

0.7221

a =

400

Test bounds: 0.424 <= gamma <= 0.836

gamma	X>=0	Y>=0	rho(XY)<1	p/f
5.95e-01	2.5e-05	0.0e+00	6.116e-20	p
5.02e-01	-2.6e+03 #	-1.7e-18	1.104e-19	f
5.47e-01	2.6e-05	0.0e+00	3.243e-18	p
5.24e-01	2.7e-05	0.0e+00	1.240e-15	p
5.13e-01	-6.9e+03 #	-9.5e-15	2.260e-16	f
5.19e-01	-3.1e+04 #	-1.8e-18	1.234e-19	f

5.21e-01 2.7e-05 0.0e+00 3.958e-16 p

Best performance (actual): 0.521

settling_time =

0.7241

a =

450

Test bounds: 0.442 <= gamma <= 0.883

gamma	X>=0	Y>=0	rho(XY)<1	p/f
6.24e-01	2.7e-05	0.0e+00	7.326e-14	p
5.25e-01	-1.6e+03 #	-6.3e-17	1.741e-15	f
5.73e-01	2.8e-05	0.0e+00	1.194e-18	p
5.48e-01	-4.7e+03 #	-2.5e-18	9.556e-22	f
5.60e-01	-6.2e+04 #	-5.2e-19	2.490e-19	f
5.66e-01	2.9e-05	0.0e+00	8.331e-17	p
5.63e-01	2.9e-05	0.0e+00	9.717e-17	p

Best performance (actual): 0.563

settling_time =

0.7158

a =

500

Test bounds: 0.491 <= gamma <= 0.93

gamma	X>=0	Y>=0	rho(XY)<1	p/f
6.76e-01	2.8e-05	0.0e+00	4.668e-16	p
5.76e-01	-6.2e+02 #	-1.3e-17	6.383e-16	f
6.24e-01	2.9e-05	0.0e+00	2.920e-42	p
5.99e-01	-3.5e+03 #	-1.4e-15	1.232e-15	f
6.11e-01	3.0e-05	0.0e+00	1.037e-15	p
6.05e-01	3.0e-05	0.0e+00	3.182e-16	p
6.02e-01	-7.7e+03 #	-2.3e-18	4.275e-17	f

Best performance (actual): 0.605

settling_time =

0.7008

a =

550

Test bounds: 0.54 <= gamma <= 0.978

gamma	X>=0	Y>=0	rho(XY)<1	p/f
7.27e-01	2.9e-05	0.0e+00	1.035e-17	p
6.26e-01	-9.5e+02 #	-8.8e-18	1.380e-16	f
6.75e-01	3.0e-05	0.0e+00	3.665e-16	p
6.50e-01	3.1e-05	0.0e+00	1.096e-14	p
6.38e-01	-2.0e+03 #	-2.3e-17	1.170e-13	f
6.44e-01	-4.1e+03 #	-9.0e-18	1.032e-18	f

Best performance (actual): 0.65

settling_time =

0.6812

a =

600

Test bounds: 0.589 <= gamma <= 1.03

gamma	X>=0	Y>=0	rho(XY)<1	p/f
7.78e-01	3.0e-05	0.0e+00	1.203e-17	p
6.77e-01	-1.4e+03 #	-1.1e-14	1.814e-17	f
7.25e-01	3.1e-05	0.0e+00	1.595e-14	p
7.01e-01	3.2e-05	0.0e+00	1.130e-15	p
6.89e-01	-4.0e+03 #	-1.6e-16	6.136e-19	f
6.95e-01	-3.1e+04 #	-1.0e-17	1.968e-15	f

Best performance (actual): 0.701

settling_time =

0.6571

a =

650

Test bounds: 0.638 <= gamma <= 1.08

gamma	X>=0	Y>=0	rho(XY)<1	p/f
8.29e-01	3.1e-05	0.0e+00	6.213e-15	p
7.27e-01	-2.1e+03 #	-6.5e-16	4.916e-13	f
7.76e-01	3.2e-05	0.0e+00	3.975e-16	p
7.51e-01	3.2e-05	0.0e+00	6.573e-16	p
7.39e-01	-1.1e+04 #	-8.7e-16	5.421e-16	f
7.45e-01	3.3e-05	0.0e+00	6.788e-16	p

Best performance (actual): 0.745

settling_time =

0.6319

a =

700

Test bounds: 0.687 <= gamma <= 1.13

gamma	X>=0	Y>=0	rho(XY)<1	p/f
8.80e-01	3.2e-05	0.0e+00	7.276e-17	p
7.77e-01	-3.2e+03 #	-3.1e-15	6.025e-16	f
8.27e-01	3.2e-05	0.0e+00	7.122e-18	p
8.02e-01	3.3e-05	0.0e+00	1.716e-15	p
7.90e-01	3.3e-05	0.0e+00	2.041e-15	p
7.83e-01	-6.8e+03 #	-7.1e-15	1.032e-13	f

Best performance (actual): 0.79

settling_time =

0.6035

a =

750

Test bounds: 0.736 <= gamma <= 1.18

gamma	X>=0	Y>=0	rho(XY)<1	p/f
9.31e-01	3.2e-05	0.0e+00	4.898e-17	p
8.28e-01	-5.0e+03 #	-2.4e-18	9.513e-15	f
8.78e-01	3.3e-05	0.0e+00	1.318e-17	p
8.52e-01	3.3e-05	0.0e+00	5.150e-16	p
8.40e-01	3.3e-05	0.0e+00	8.806e-19	p
8.34e-01	-1.7e+04 #	-7.2e-21	4.777e-17	f

Best performance (actual): 0.84

settling_time =

0.5656

a =

800

Test bounds: 0.785 <= gamma <= 1.23

gamma	X>=0	Y>=0	rho(XY)<1	p/f
9.82e-01	3.3e-05	0.0e+00	2.992e-19	p

8.78e-01	-8.1e+03 #	-6.5e-17	9.017e-16	f
9.28e-01	3.3e-05	0.0e+00	3.141e-17	p
9.03e-01	3.4e-05	0.0e+00	1.193e-17	p
8.90e-01	3.4e-05	0.0e+00	6.389e-17	p
8.84e-01	-9.6e+05 #	-1.0e-21	1.868e-15	f

Best performance (actual): 0.89

settling_time =

0.3477

a =

850

Iteration Completed.

```
In [14]: a = 850;
W_s = a * (s^2 + 26.68*s + 185.23) / ((s + 0.001)*(s + 26.68)*(s + 1000));
% Build Generalized Plant
G_theta_p = augw(G_theta , W_s , W_u , W_theta);

% Synthesize H∞ Controller
[C_theta , CL , gamma , info] = hinfsyn(G_theta_p , 1 , 1 , 'method' , 'ric' , '

T = feedback(G_theta*C_theta, 1);
U = C_theta / (1 + G_theta * C_theta);

figure;
step(T);
title('Closed-Loop Step Response of the Nominal System (Pitch)');
grid on;

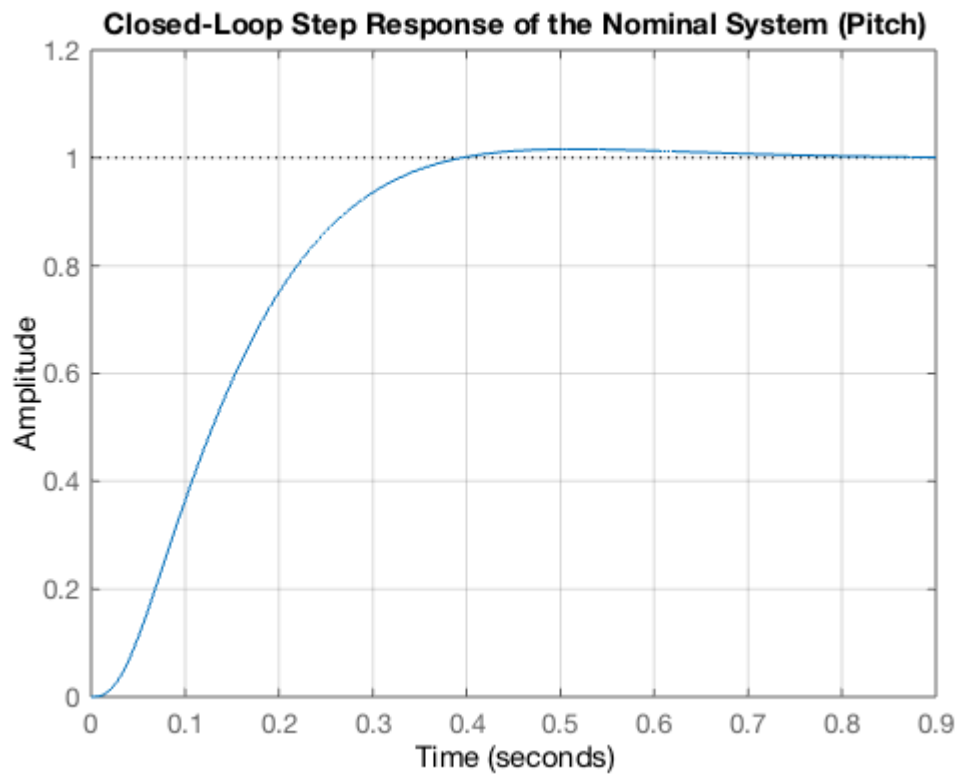
figure;
step(U);
title('Control Effort (Actuator Signal) (Pitch)');
grid on

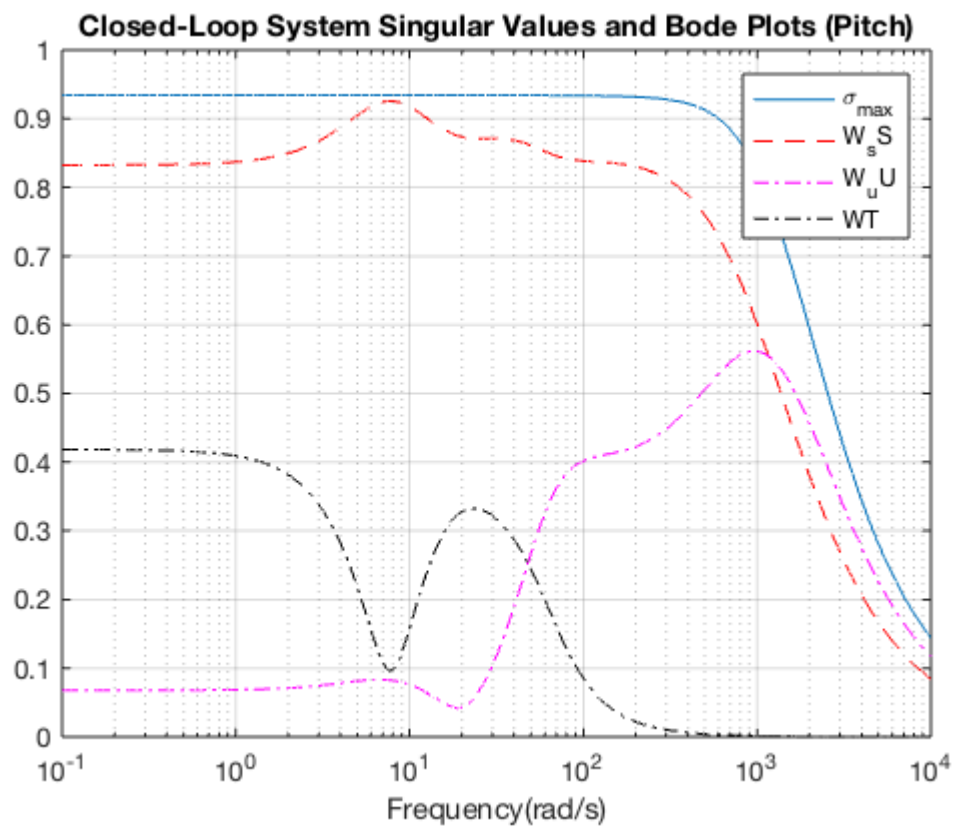
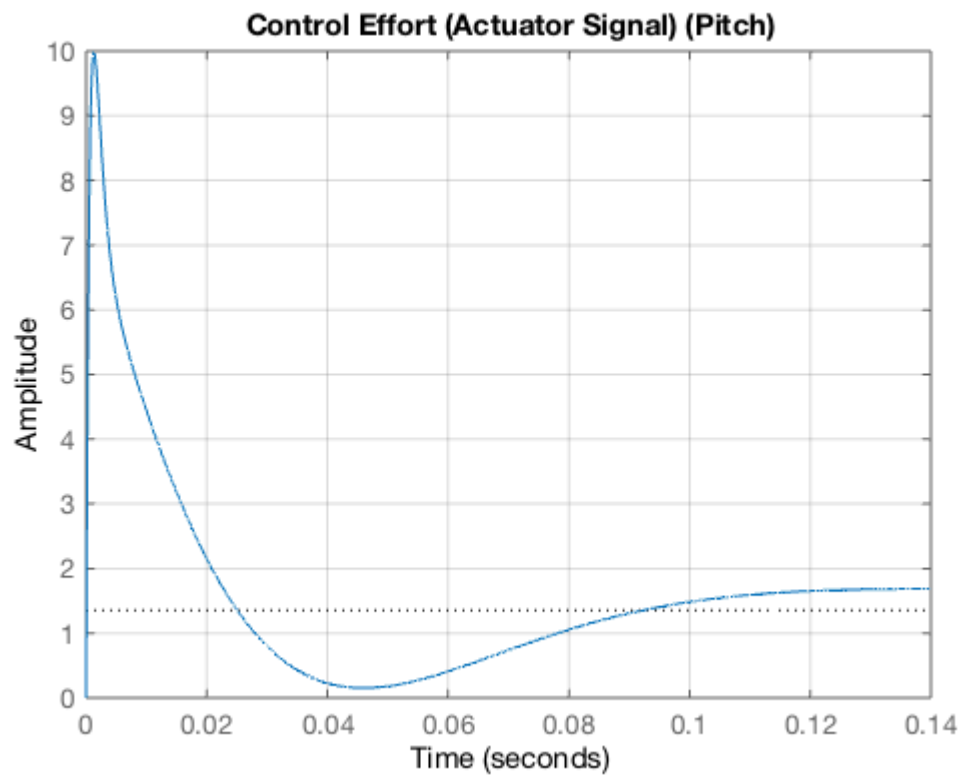
figure;
omega = logspace(-1,4,300);
[mag_g,phase_g]=bode(CL,omega);
sig_g=sigma(CL,omega);
semilogx(omega,sig_g,omega,mag_g(1,:), '--r',omega,mag_g(2,:), '-.m',...
omega,mag_g(3,:), '-.k');
title("Closed-Loop System Singular Values and Bode Plots (Pitch)");
xlabel("Frequency(rad/s)");
legend('\sigma_{max}', 'W_sS', 'W_uU', 'WT');
grid on;
```

Test bounds: $0.834 \leq \gamma \leq 1.28$

γ	$X \geq 0$	$Y \geq 0$	$\rho(XY) < 1$	p/f
1.03e+00	3.3e-05	0.0e+00	3.194e-17	p
9.28e-01	-1.5e+04 #	-1.9e-14	1.330e-14	f
9.79e-01	3.3e-05	0.0e+00	1.133e-17	p
9.53e-01	3.4e-05	0.0e+00	6.035e-17	p
9.40e-01	3.4e-05	0.0e+00	3.826e-16	p
9.34e-01	3.4e-05	0.0e+00	1.695e-16	p

Best performance (actual): 0.934





2.4. Improvement Cycle (Roll)

The same steps are applied on the Roll associated objectives.

```
In [15]: % Modified Weights
w_u = 0.05;

a = 0;
```

```

settling_time = 100;
while settling_time > 0.35
    W_s = a * (s^2 + 26.68*s + 185.23) / ((s + 0.001)*(s + 26.68)*(s + 1000));

    % Build Generalized Plant
    G_phi_p = augw(G_phi , W_s , W_u , W_phi);

    % Synthesize H $\infty$  Controller
    [C_phi , CL , gamma , info] = hinfsyn(G_phi_p , 1 , 1 , 'method' , 'ric' , '

    T = feedback(G_phi*C_phi, 1);
    settling_time = stepinfo(T).SettlingTime
    a = a + 50
end
disp("Iteration Completed.")

```


Test bounds: $0 \leq \gamma \leq 2e-06$

γ	$X \geq 0$	$Y \geq 0$	$\rho(XY) < 1$	p/f
$2.00e-07$	$6.3e-04$	$0.0e+00$	$3.644e-18$	p

Best performance (actual): $6.85e-30$

settling_time =

1.7988

a =

50

Test bounds: $0.327 \leq \gamma \leq 0.473$

γ	$X \geq 0$	$Y \geq 0$	$\rho(XY) < 1$	p/f
$3.93e-01$	$3.8e-04$	$0.0e+00$	$3.011e-15$	p
$3.59e-01$	$3.9e-04$	$0.0e+00$	$2.817e-17$	p
$3.43e-01$	$3.9e-04$	$0.0e+00$	$2.190e-16$	p
$3.35e-01$	$3.9e-04$	$0.0e+00$	$4.301e-17$	p
$3.31e-01$	$3.9e-04$	$0.0e+00$	$4.412e-15$	p
$3.29e-01$	$4.0e-04$	$0.0e+00$	$2.918e-15$	p

Best performance (actual): 0.329

settling_time =

1.1940

a =

100

Test bounds: $0.327 \leq \gamma \leq 0.493$

γ	$X \geq 0$	$Y \geq 0$	$\rho(XY) < 1$	p/f
$4.01e-01$	$6.4e-04$	$0.0e+00$	$3.923e-17$	p
$3.62e-01$	$6.4e-04$	$0.0e+00$	$9.130e-15$	p
$3.44e-01$	$6.4e-04$	$0.0e+00$	$1.848e-17$	p
$3.36e-01$	$6.4e-04$	$0.0e+00$	$1.237e-14$	p
$3.31e-01$	$6.4e-04$	$0.0e+00$	$4.014e-14$	p
$3.29e-01$	$6.4e-04$	$0.0e+00$	$1.640e-17$	p

Best performance (actual): 0.329

settling_time =

1.1552

a =

150

Test bounds: 0.327 <= gamma <= 0.521

gamma	X>=0	Y>=0	rho(XY)<1	p/f
4.13e-01	5.0e-04	0.0e+00	8.843e-16	p
3.67e-01	5.1e-04	0.0e+00	4.099e-18	p
3.47e-01	5.3e-04	0.0e+00	1.246e-17	p
3.37e-01	5.3e-04	0.0e+00	8.242e-17	p
3.32e-01	5.4e-04	0.0e+00	9.827e-18	p
3.30e-01	-1.6e+03 #	-9.2e-16	8.014e-16	f

Best performance (actual): 0.332

settling_time =

1.1173

a =

200

Test bounds: 0.327 <= gamma <= 0.555

gamma	X>=0	Y>=0	rho(XY)<1	p/f
4.26e-01	6.5e-04	0.0e+00	5.110e-17	p
3.73e-01	6.6e-04	0.0e+00	2.041e-13	p
3.49e-01	6.7e-04	0.0e+00	9.896e-17	p
3.38e-01	-2.6e+03 #	-5.1e-17	1.993e-14	f
3.44e-01	6.7e-04	0.0e+00	2.283e-13	p
3.41e-01	-5.3e+04 #	-1.2e-16	1.944e-12	f

Best performance (actual): 0.344

settling_time =

0.8734

a =

250

Test bounds: 0.327 <= gamma <= 0.593

gamma	X>=0	Y>=0	rho(XY)<1	p/f
4.41e-01	6.9e-04	0.0e+00	1.358e-17	p
3.80e-01	6.9e-04	0.0e+00	4.896e-17	p
3.52e-01	-1.4e+03 #	-7.3e-16	1.262e-15	f
3.66e-01	6.9e-04	0.0e+00	1.183e-15	p
3.59e-01	-4.7e+03 #	-2.4e-17	7.385e-17	f
3.62e-01	6.9e-04	0.0e+00	9.757e-18	p

Best performance (actual): 0.362

settling_time =

0.8128

a =

300

Test bounds: 0.327 <= gamma <= 0.635

gamma	X>=0	Y>=0	rho(XY)<1	p/f
4.56e-01	7.0e-04	0.0e+00	8.826e-17	p
3.86e-01	-2.7e+03 #	-3.7e-17	8.226e-17	f
4.20e-01	7.0e-04	0.0e+00	1.198e-14	p
4.03e-01	7.0e-04	0.0e+00	5.012e-17	p
3.94e-01	7.0e-04	0.0e+00	1.039e-15	p
3.90e-01	-5.7e+03 #	-5.5e-17	9.506e-17	f
3.92e-01	-1.3e+04 #	-5.6e-17	2.421e-16	f
Limiting gains...				
3.96e-01	7.0e-04	0.0e+00	8.708e-14	p
3.96e-01	7.0e-04	0.0e+00	8.709e-14	p

Best performance (actual): 0.396

settling_time =

0.6054

a =

350

Test bounds: 0.343 <= gamma <= 0.679

gamma	X>=0	Y>=0	rho(XY)<1	p/f
4.83e-01	7.1e-04	0.0e+00	3.351e-16	p
4.07e-01	-1.0e+03 #	-9.4e-16	2.305e-15	f
4.44e-01	7.1e-04	0.0e+00	8.095e-17	p
4.25e-01	-3.7e+03 #	-2.9e-16	1.130e-15	f
4.34e-01	7.1e-04	0.0e+00	5.366e-16	p
4.30e-01	-9.4e+03 #	-6.2e-17	1.105e-16	f
4.32e-01	-3.9e+04 #	-3.4e-17	6.374e-17	f

Best performance (actual): 0.434

settling_time =

0.6025

a =

400

Test bounds: 0.393 <= gamma <= 0.725

gamma	X>=0	Y>=0	rho(XY)<1	p/f
5.34e-01	7.2e-04	0.0e+00	3.004e-18	p

4.58e-01	-2.0e+03 #	-8.7e-15	2.960e-16	f
4.94e-01	7.2e-04	0.0e+00	1.695e-14	p
4.76e-01	7.2e-04	0.0e+00	3.591e-15	p
4.67e-01	-4.1e+03 #	-4.8e-15	1.874e-17	f
4.71e-01	-8.6e+03 #	-3.6e-17	1.443e-18	f
Limiting gains...				
4.76e-01	7.2e-04	0.0e+00	1.667e-16	p
4.76e-01	7.2e-04	0.0e+00	1.668e-16	p

Best performance (actual): 0.476

settling_time =

0.5885

a =

450

Test bounds: 0.442 <= gamma <= 0.772

gamma	X>=0	Y>=0	rho(XY)<1	p/f
5.84e-01	7.2e-04	0.0e+00	5.388e-16	p
5.08e-01	-3.7e+03 #	-5.1e-15	7.323e-15	f
5.45e-01	7.3e-04	0.0e+00	3.816e-16	p
5.26e-01	7.3e-04	0.0e+00	9.664e-17	p
5.17e-01	-1.5e+04 #	-1.1e-15	1.297e-14	f
5.21e-01	7.4e-04	0.0e+00	4.012e-16	p
Limiting gains...				
5.22e-01	7.4e-04	0.0e+00	1.750e-16	p
5.22e-01	7.4e-04	0.0e+00	1.751e-16	p

Best performance (actual): 0.522

settling_time =

0.5656

a =

500

Test bounds: 0.491 <= gamma <= 0.82

gamma	X>=0	Y>=0	rho(XY)<1	p/f
6.34e-01	7.0e-04	0.0e+00	5.953e-18	p
5.58e-01	-1.8e+03 #	-2.3e-17	1.782e-17	f
5.95e-01	7.1e-04	0.0e+00	1.515e-16	p
5.76e-01	7.1e-04	0.0e+00	1.732e-18	p
5.67e-01	7.1e-04	0.0e+00	1.056e-15	p
5.62e-01	-4.1e+03 #	-2.9e-17	4.142e-16	f

Best performance (actual): 0.567

settling_time =

0.5349

a =

550

Test bounds: 0.54 <= gamma <= 0.869

gamma	X>=0	Y>=0	rho(XY)<1	p/f
6.85e-01	7.1e-04	0.0e+00	7.332e-18	p
6.08e-01	-3.6e+03 #	-8.4e-17	3.022e-15	f
6.45e-01	7.2e-04	0.0e+00	1.536e-17	p
6.26e-01	7.2e-04	0.0e+00	1.570e-18	p
6.17e-01	7.2e-04	0.0e+00	2.132e-17	p
6.13e-01	-7.4e+04 #	-4.3e-17	1.945e-17	f

Best performance (actual): 0.617

settling_time =

0.4688

a =

600

Test bounds: 0.589 <= gamma <= 0.918

gamma	X>=0	Y>=0	rho(XY)<1	p/f
7.35e-01	7.2e-04	0.0e+00	3.160e-17	p
6.58e-01	-9.6e+03 #	-5.8e-17	1.743e-17	f
6.96e-01	7.2e-04	0.0e+00	5.336e-17	p
6.77e-01	7.3e-04	0.0e+00	7.867e-16	p
6.67e-01	7.3e-04	0.0e+00	3.718e-16	p
6.63e-01	7.3e-04	0.0e+00	1.995e-17	p

Best performance (actual): 0.663

settling_time =

0.3329

a =

650

Iteration Completed.

```
In [16]: a = 650;
W_s = a * (s^2 + 26.68*s + 185.23) / ((s + 0.001)*(s + 26.68)*(s + 1000));
% Build Generalized Plant
G_phi_p = augw(G_phi , W_s , W_u , W_phi);

% Synthesize H∞ Controller
[C_phi , CL , gamma , info] = hinfsyn(G_phi_p , 1 , 1 , 'method' , 'ric' , 'Disp
```

```

T = feedback(G_phi*C_phi, 1);
U = C_theta / (1 + G_phi * C_phi);

figure;
step(T);
title('Closed-Loop Step Response of the Nominal System (Roll)');
grid on;

figure;
step(U);
title('Control Effort (Actuator Signal) (Roll)');
grid on

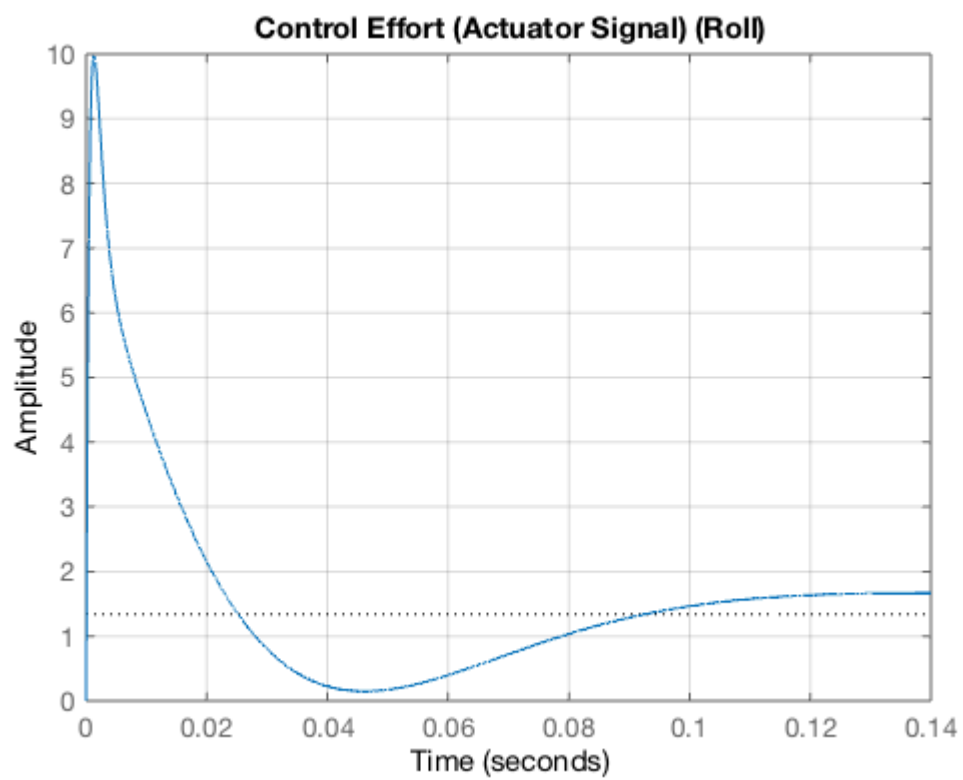
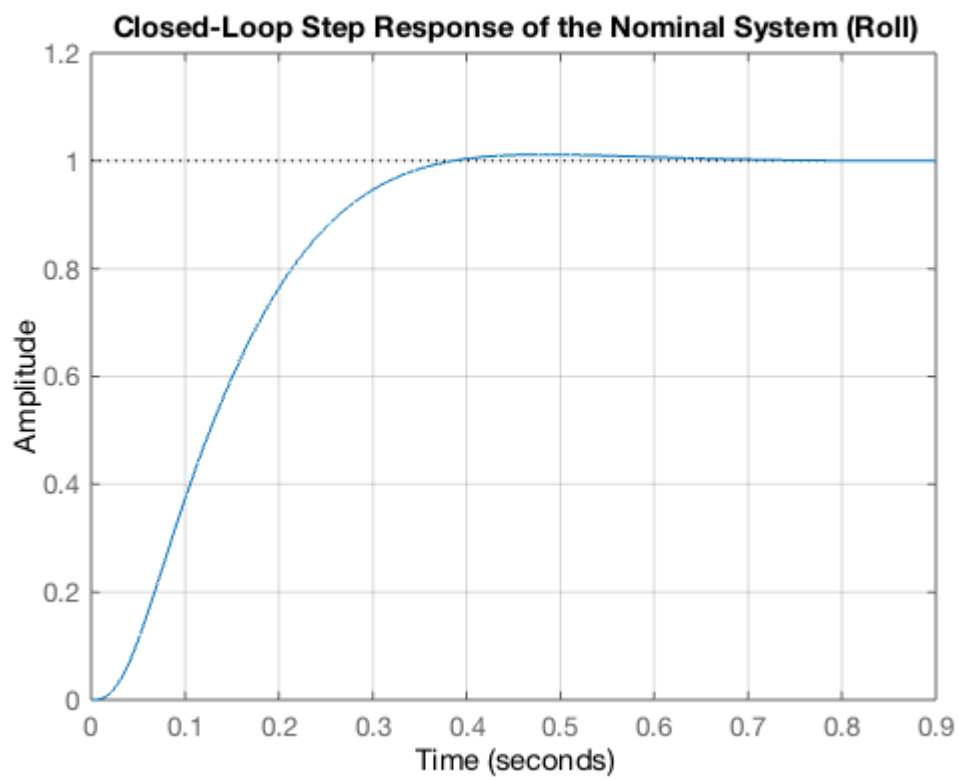
figure;
omega = logspace(-1,4,300);
[mag_g,phase_g]=bode(CL,omega);
sig_g=sigma(CL,omega);
semilogx(omega,sig_g,omega,mag_g(1,:), '--r',omega,mag_g(2,:), '-.m',...
omega,mag_g(3,:), '-.k');
title("Closed-Loop System Singular Values and Bode Plots (Roll)");
xlabel("Frequency(rad/s)");
legend('\sigma_{max}', 'W_sS', 'W_uU', 'WT');
grid on;

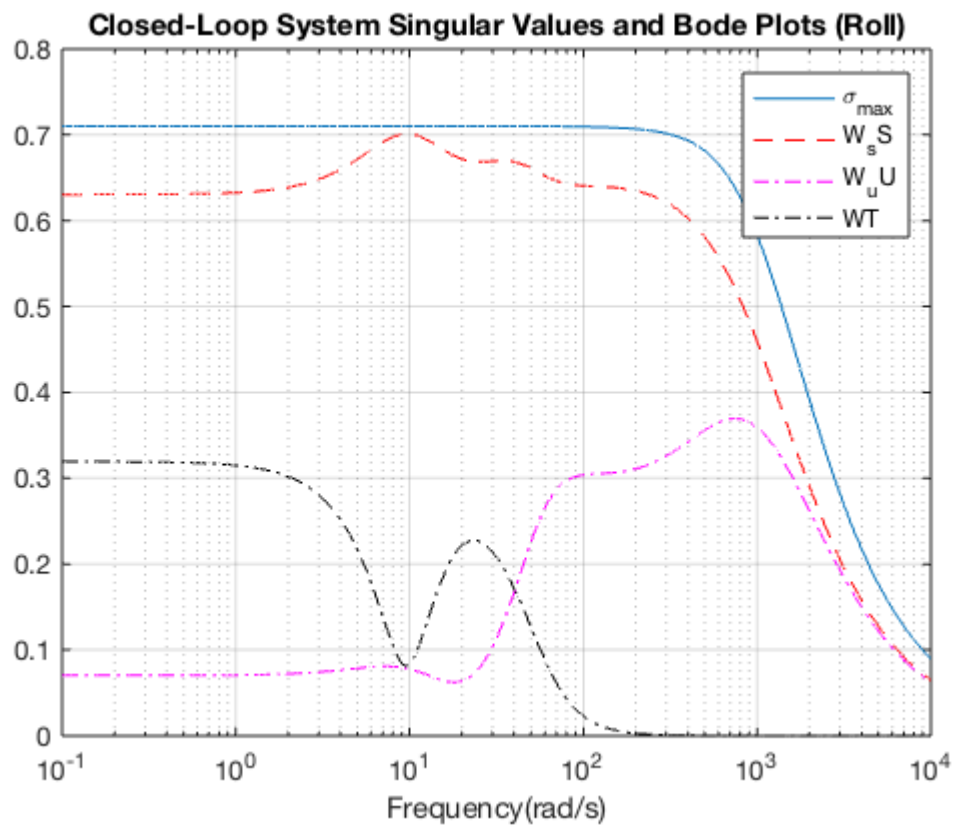
```

Test bounds: 0.638 <= gamma <= 0.968

gamma	X>=0	Y>=0	rho(XY)<1	p/f
7.86e-01	7.2e-04	0.0e+00	2.199e-17	p
7.08e-01	7.4e-04	0.0e+00	1.320e-14	p
6.72e-01	-6.1e+02 #	-1.0e-15	2.487e-15	f
6.90e-01	-1.3e+03 #	-7.1e-17	5.423e-17	f
6.99e-01	-2.6e+03 #	-2.6e-17	7.476e-17	f
7.03e-01	-5.3e+03 #	-3.2e-17	5.085e-18	f
Limiting gains...				
7.10e-01	7.4e-04	0.0e+00	1.194e-16	p
7.10e-01	7.4e-04	0.0e+00	1.194e-16	p

Best performance (actual): 0.71





2.5. Testing on Uncertain Systems

The Optimal Robust H_{∞} Controllers for Pitch and Roll angles are as follows.

```
In [17]: disp("Controller for Pitch Angle: ");
         tf(C_theta)

         disp("Controller for Roll Angle: ");
         tf(C_phi)
```


Controller for Pitch Angle:

ans =

$$\frac{2.369e04 s^7 + 5.982e08 s^6 + 2.858e11 s^5 + 1.872e13 s^4 + 4.554e14 s^3 + 8.612e15 s^2 + 7.941e16 s + 2.344e17}{s^8 + 2.715e04 s^7 + 6.056e07 s^6 + 3.819e10 s^5 + 3.997e12 s^4 + 2.165e14 s^3 + 4.539e15 s^2 + 2.44e16 s + 2.44e13}$$

Continuous-time transfer function.

Controller for Roll Angle:

ans =

$$\frac{1.258e04 s^7 + 6.534e06 s^6 + 6.665e08 s^5 + 3.016e10 s^4 + 7.851e11 s^3 + 1.256e13 s^2 + 1.142e14 s + 3.868e14}{s^8 + 2032 s^7 + 1.172e06 s^6 + 1.496e08 s^5 + 9.86e09 s^4 + 3.484e11 s^3 + 5.991e12 s^2 + 3.843e13 s + 3.843e10}$$

Continuous-time transfer function.

```
In [18]: % Simulate step response for each sample (Pitch)
figure;
hold on;
T = feedback(G_theta*C_theta, 1);
step(T);
for i = 1:num_samples
    T = feedback(G_theta_samples(:, :, i)*C_theta, 1); % Closed-loop
    step(T);
end
title('Closed-loop Step Response (Pitch)');
legend('Nominal System' , 'Uncertain Systems')
grid on;
hold off;

figure;
hold on;
U = C_theta / (1 + G_theta * C_theta);
step(U);
for i = 1:num_samples
    U = C_theta / (1 + G_theta_samples(:, :, i)*C_theta);
    step(U);
end
title('Control Effort (Pitch)');
```

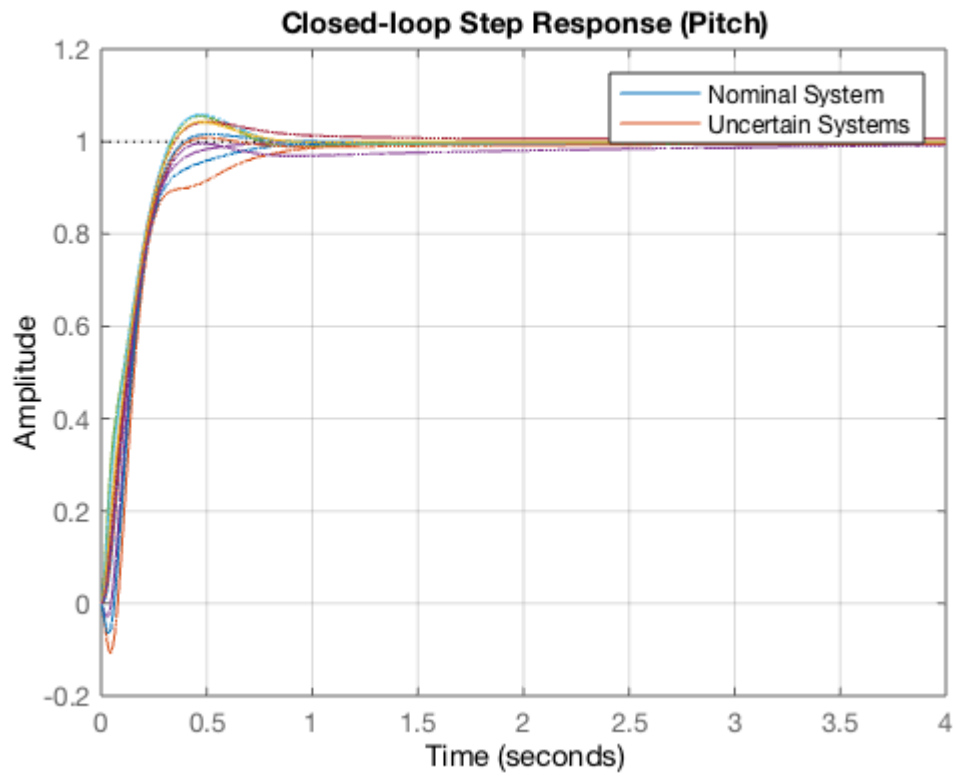
```

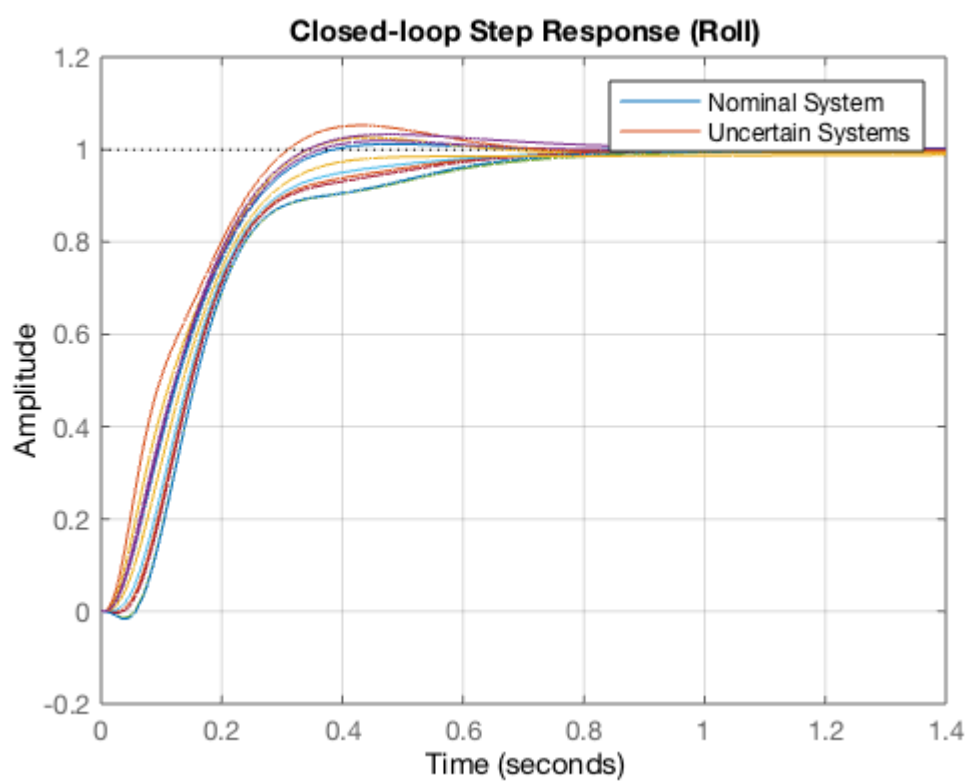
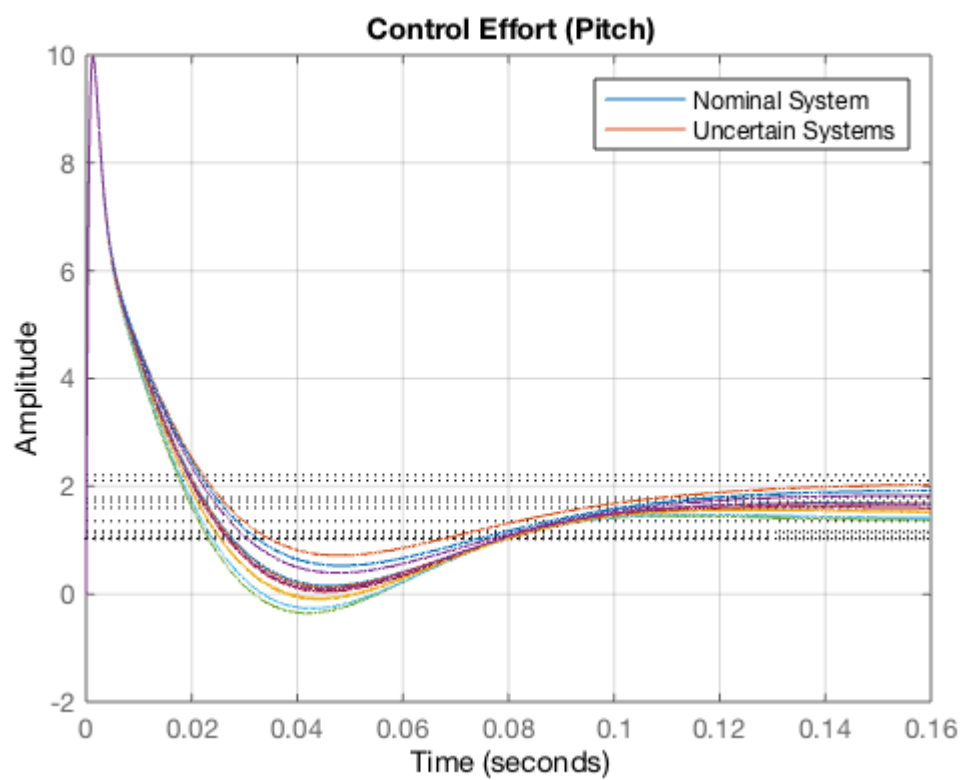
legend('Nominal System' , 'Uncertain Systems')
grid on;

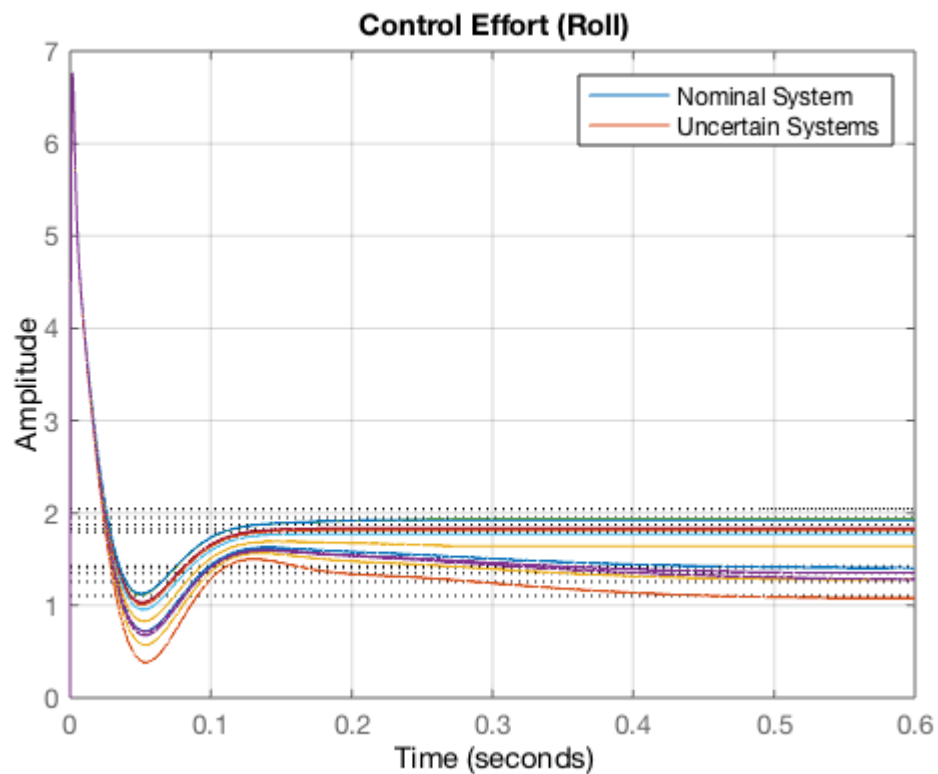
% Simulate step response for each sample (Roll)
figure;
hold on;
T = feedback(G_phi*C_phi, 1);
step(T);
for i = 1:num_samples
    T = feedback(G_phi_samples(:,:,i)*C_phi, 1);
    step(T);
end
title('Closed-loop Step Response (Roll)');
legend('Nominal System' , 'Uncertain Systems')
grid on;
hold off;

figure;
hold on;
U = C_phi / (1 + G_phi * C_phi);
step(U);
for i = 1:num_samples
    U = C_phi / (1 + G_phi_samples(:,:,i)*C_phi);
    step(U);
end
title('Control Effort (Roll)');
legend('Nominal System' , 'Uncertain Systems')
grid on;

```







2.6. Dealing with Reality: Saturation!

Actuator saturation refers to the physical limits of the actuators of the system (motors, valves, etc.). Every real-world actuator has:

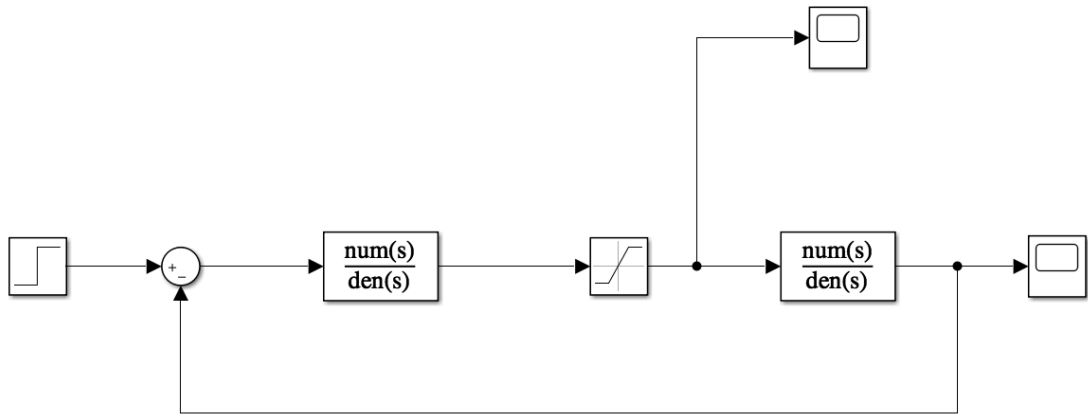
Upper limit: Maximum force/voltage/speed it can provide

Lower limit: Minimum value (often zero, but could be negative for bidirectional actuators)

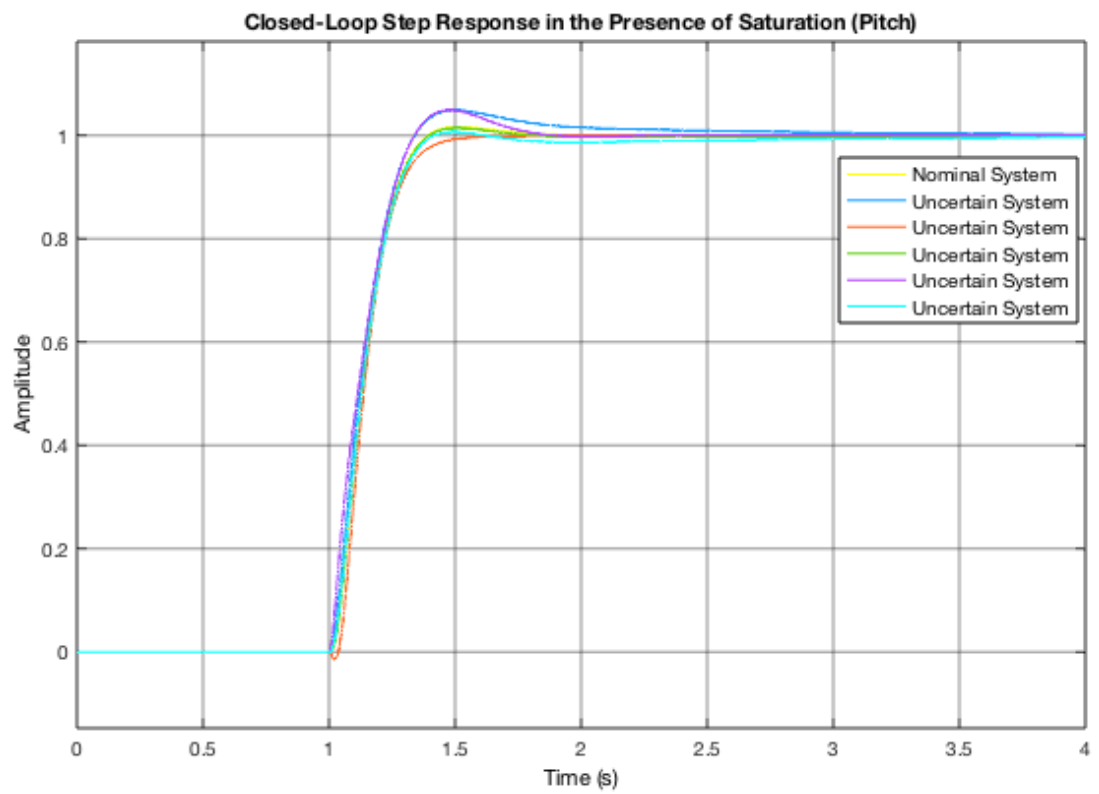
When the controller demands values beyond these limits, the actuator "saturates" (can't deliver more).

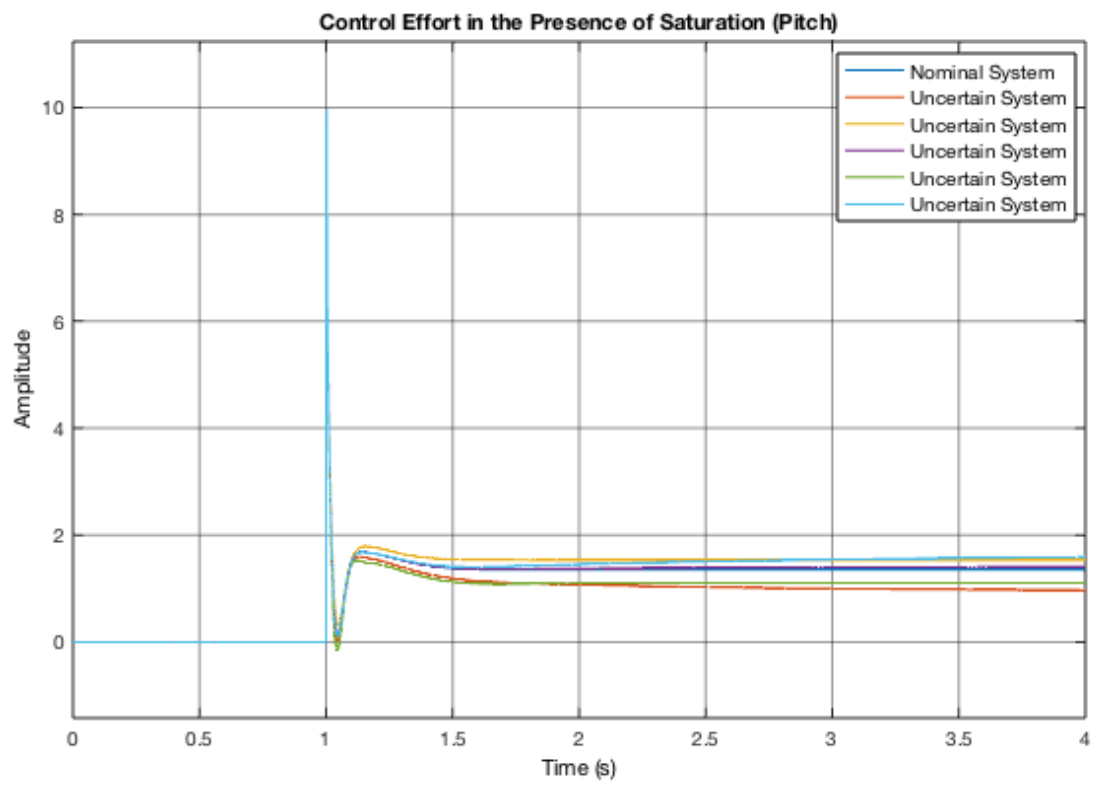
But the fact is that the control effort under review is actually the reference input of the inner loop in the cascade architecture. Therefore, it's physically the desired angle (Pitch or Roll) and considered to be less than 20 degrees.

Simulation is done using Simulink with the presence of saturation block and the resulting curves are as follows.

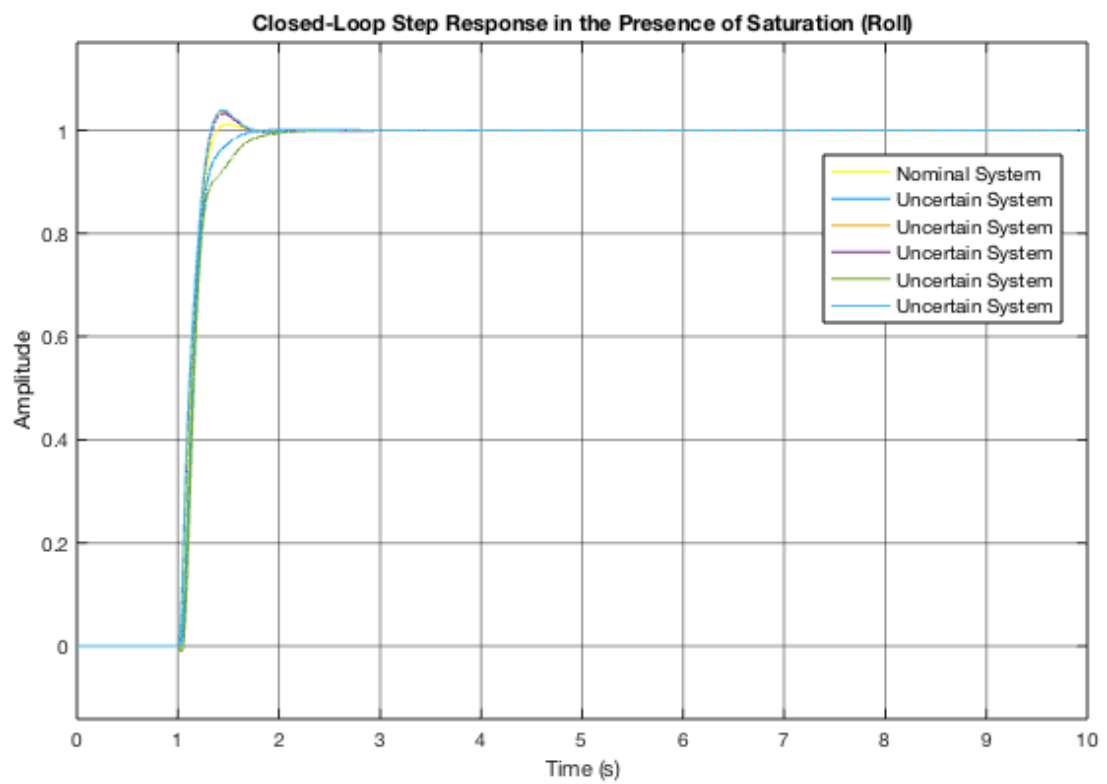


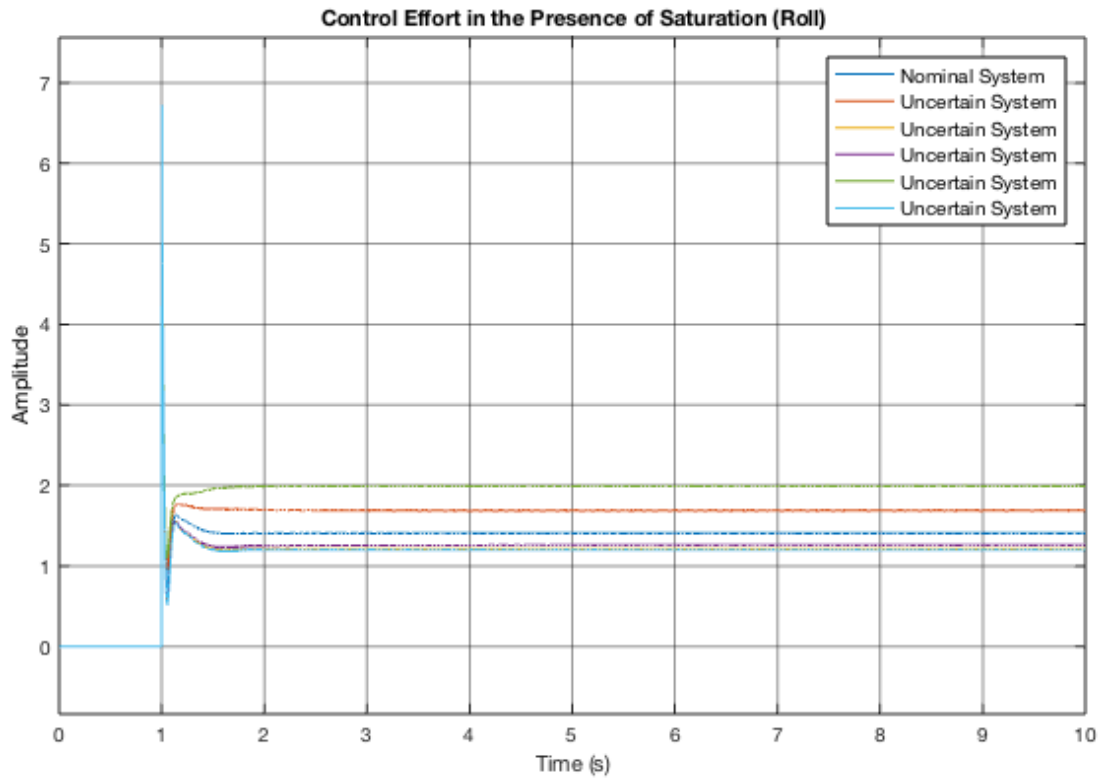
```
In [ ]: load('simulink_files/workspace.mat'); % Load workspace
sim('simulink_files/saturated_pitch_system.slx'); % Run Simulink
```





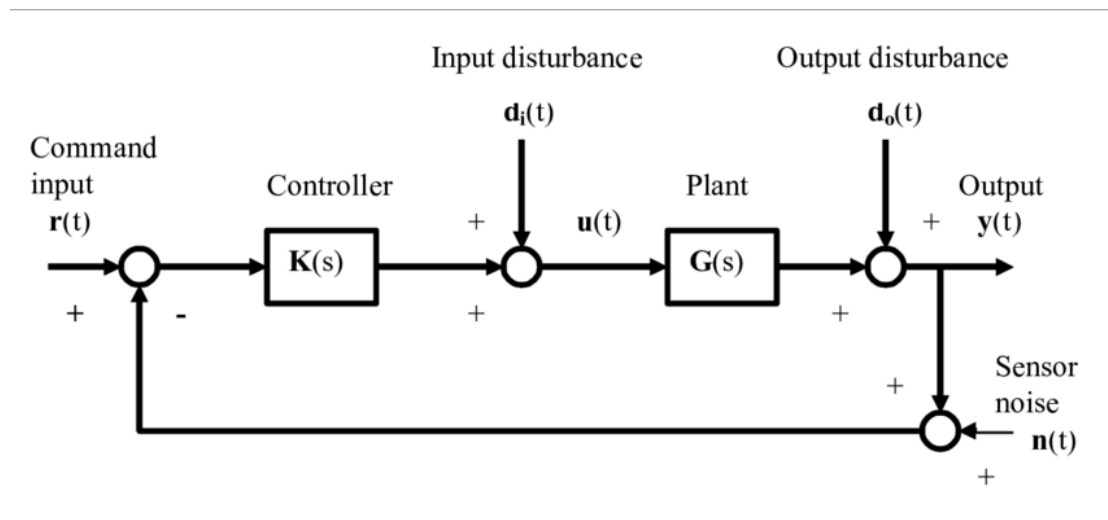
```
In [ ]: sim('simulink_files/saturated_roll_system.slx');
```





2.7. Testing in the Presence of Disturbance and Noise

It's essential to test the system's robustness against common disturbances and noise. Below are standard inputs used to model control effort disturbances, output disturbances, and feedback sensor noise.

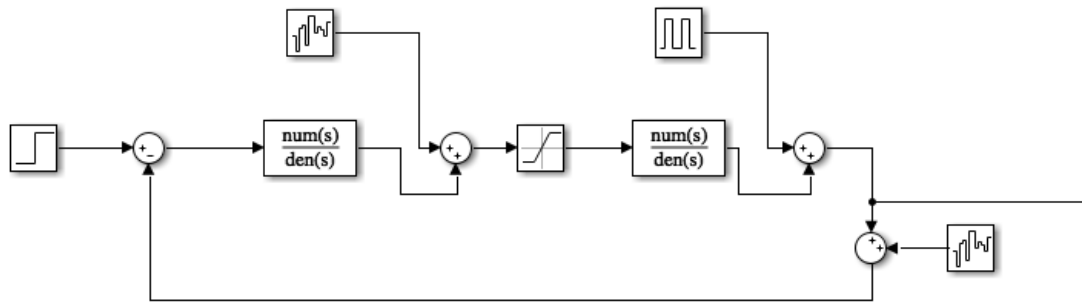


Control Effort (Input) Disturbance: Band-Limited White Noise with the Noise Power of 0.001

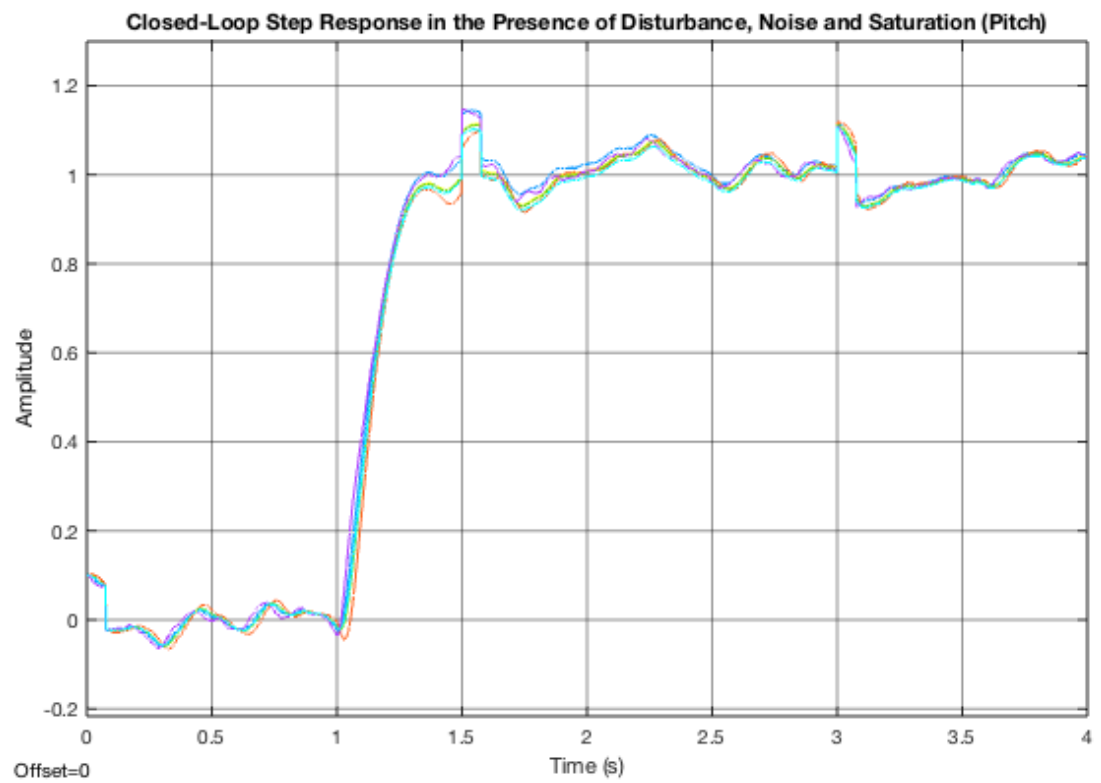
Output Disturbance: Pulse Input with the amplitude of 0.1 and period time of 1.5 seconds

Feedback Noise: The same Band-Limited White Noise to the feedback path

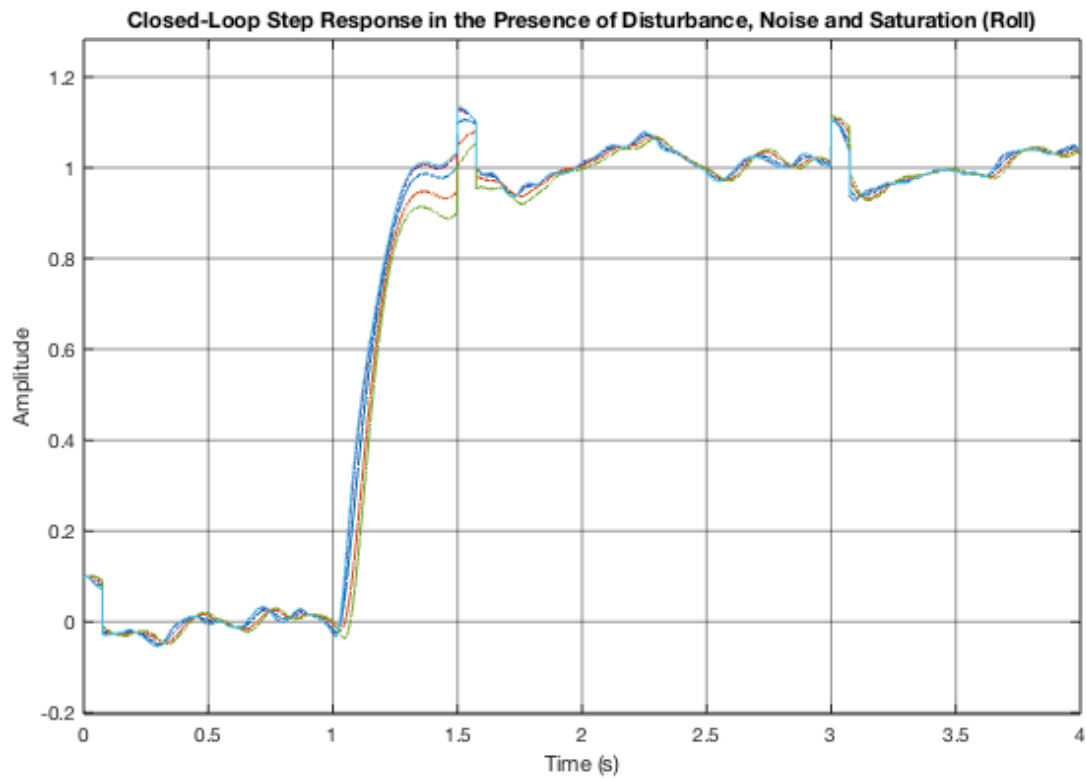
The disturbed systems are also saturated and simulated as shown below.



```
In [ ]: sim('simulink_files/disturbed_pitch_system.slx');
```



```
In [ ]: sim('simulink_files/disturbed_roll_system.slx');
```

3.1 μ -Analysis of H_∞ Controller

To determine whether the H_∞ controller guarantees robust stability (RS) and robust performance (RP) against structured uncertainties.

First Building interconnection structures (cloop_theta and cloop_phi) using 'sysic' for the uncertain models of pitch (theta) and roll (phi) axes.

We create two interconnected systems using the 'sysic' command, one for theta and one for phi, to prepare the augmented plant

```
In [19]: P_theta = G_theta_unc;
P_phi= G_phi_unc;

systemnames = 'P_theta C_theta W_s W_u W_theta';
inputvar = '[yd]';

input_to_C_theta = '[yd -P_theta]';
input_to_P_theta = '[C_theta]';
input_to_W_s = '[yd -P_theta]';
input_to_W_u = '[C_theta]';
input_to_W_theta = '[P_theta]';

outputvar = '[W_s; W_u; W_theta]';
cleanupysic = 'yes';
cloop_theta = sysic;

systemnames = 'P_phi C_phi W_s W_u W_phi';
inputvar = '[yd]';

input_to_C_phi = '[yd -P_phi]';
```

```

input_to_P_phi = '[C_phi]';
input_to_W_s = '[yd -P_phi]';
input_to_W_u = '[C_phi]';
input_to_W_phi = '[P_phi]';

outputvar = '[W_s; W_u; W_phi]';
cleanup_sysic = 'yes';
cloop_phi = sysic;

```

Robustness Analysis using Mu (μ) Analysis

This step evaluates:

1. Robust Stability (RS): Ensures system remains stable under all modeled uncertainties.
2. Robust Performance (RP): Ensures both stability and performance objectives are met despite uncertainties.

```

In [20]: [stabmarg_theta, ~, ~, info_theta] = robuststab(cloop_theta);
[M_theta, ~, blkstruct_theta] = lftdata(cloop_theta);
M11_theta = M_theta(1,1);
omega_theta = info_theta.Frequency;
M11_g_theta = frd(M11_theta, omega_theta);
mu_bounds_rs_theta = mussv(M11_g_theta, blkstruct_theta, 's');

[perf_marg_theta, ~, ~, info_perf_theta] = robustperf(cloop_theta);
mu_bounds_rp_theta = info_perf_theta.MussvBnds;

figure;
semilogx(mu_bounds_rs_theta, 'b');
xlabel('Frequency (rad/sec)');
ylabel('Mu upper/lower bounds');
title('μ Plot - Robust Stability (θ)');

figure;
semilogx(mu_bounds_rp_theta, 'b');
xlabel('Frequency (rad/sec)');
ylabel('Mu upper/lower bounds');
title('μ Plot - Robust Performance (θ)');

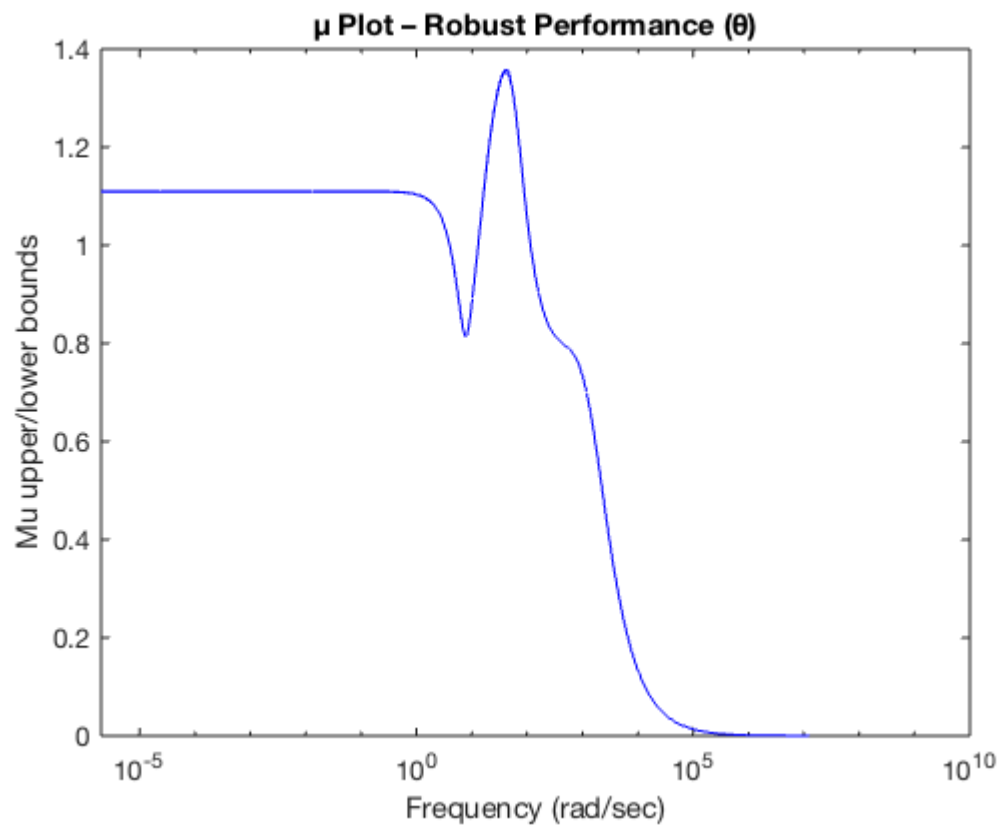
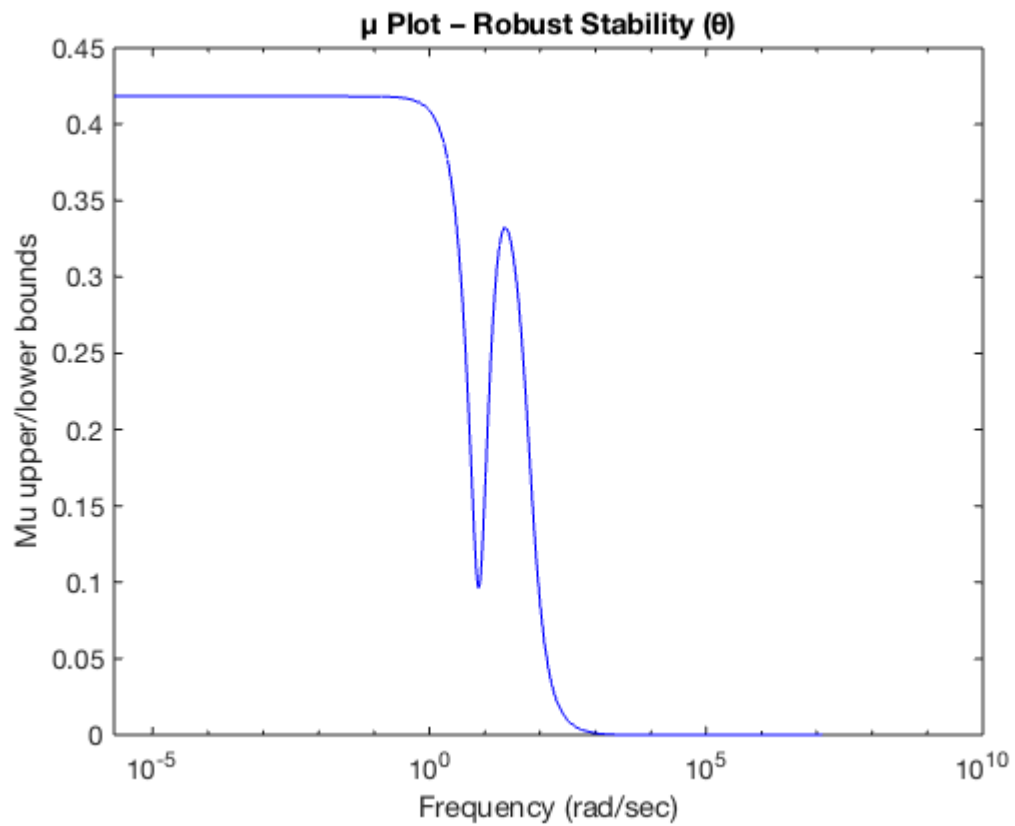
[stabmarg_phi, ~, ~, info_phi] = robuststab(cloop_phi);
[M_phi, ~, blkstruct_phi] = lftdata(cloop_phi);
M11_phi = M_phi(1,1);
omega_phi = info_phi.Frequency;
M11_g_phi = frd(M11_phi, omega_phi);
mu_bounds_rs_phi = mussv(M11_g_phi, blkstruct_phi, 's');

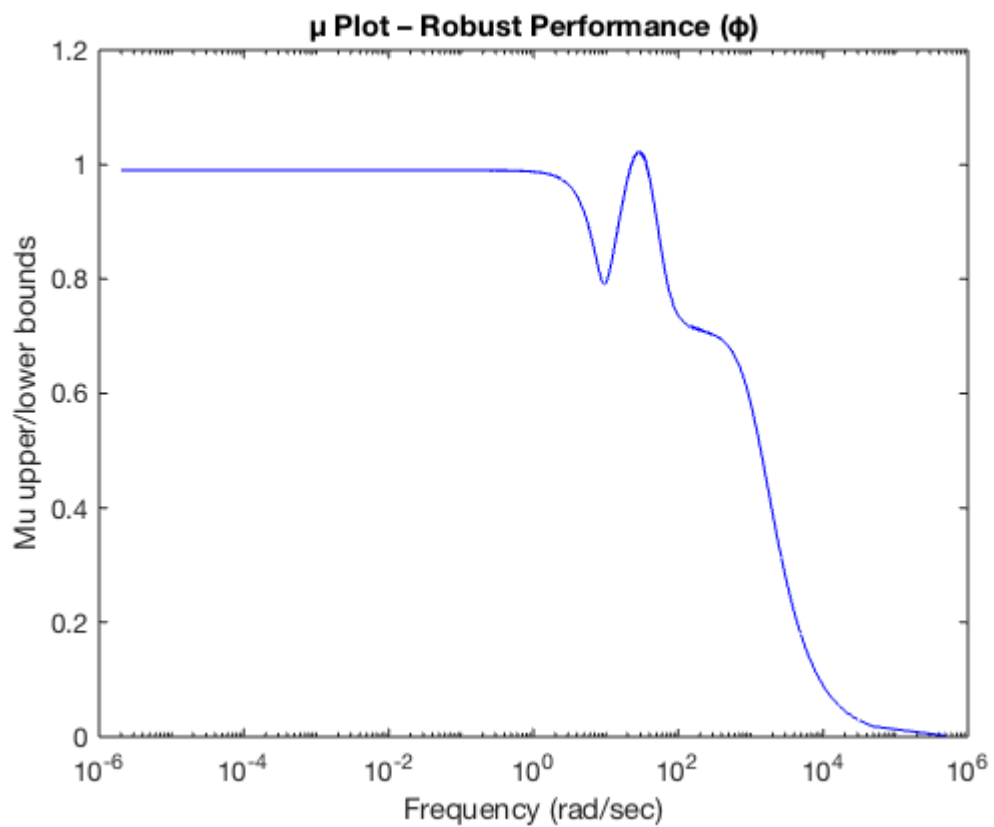
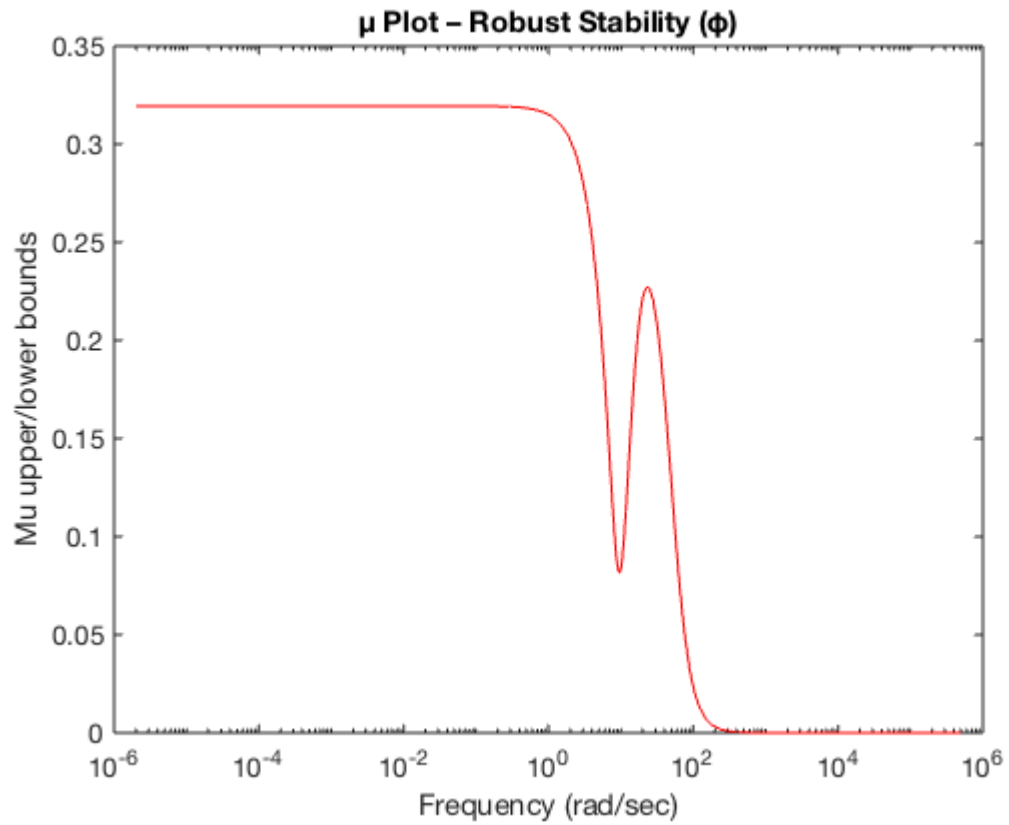
[perf_marg_phi, ~, ~, info_perf_phi] = robustperf(cloop_phi);
mu_bounds_rp_phi = info_perf_phi.MussvBnds;

figure;
semilogx(mu_bounds_rs_phi, 'r');
xlabel('Frequency (rad/sec)');
ylabel('Mu upper/lower bounds');
title('μ Plot - Robust Stability (φ)');

```

```
figure;
semilogx(mu_bounds_rp_phi, 'b');
xlabel('Frequency (rad/sec)');
ylabel('Mu upper/lower bounds');
title('μ Plot - Robust Performance (φ)');
```





3.2 D-K Iteration: Tuning W_s to Reduce μ -Peak

Since the μ -performance exceeds 1 at certain frequencies for both channels, tuning is required. For the θ channel, MUSYN alone was not sufficient, so the performance weight scaling was adjusted before reapplying MUSYN.

```

In [21]: %Decreasing the gain of Ws
W_s = 1 * (s^2 + 26.68*s + 185.23) / ((s + 0.001)*(s + 26.68)*(s + 1000));

% Applying MUSYN to reduce the  $\mu$ -peak and improve robust performance
opt = musynOptions('MaxIter', 5, 'TargetPerf', 0.3, 'TolPerf', 0, 'FitOrder', 50);
P_for_synthesis_tetha = augw(G_theta_unc, W_s, W_u, []);
[C_musyn_tetha] = musyn(P_for_synthesis_tetha, 1, 1, opt);

C_theta_musyn = C_musyn_tetha;

systemnames = 'P_theta C_theta_musyn W_s W_u W_theta';
inputvar = '[yd]';
input_to_C_theta_musyn = '[yd -P_theta]';
input_to_P_theta = '[C_theta_musyn]';
input_to_W_s = '[yd -P_theta]';
input_to_W_u = '[C_theta_musyn]';
input_to_W_theta = '[P_theta]';
outputvar = '[W_s; W_u; W_theta]';
cleanupysic = 'yes';
cloop_theta_musyn = sysic;

[stabmarg_theta_musyn, ~, ~, info_theta_musyn] = robuststab(cloop_theta_musyn);
[M_theta_musyn, ~, blkstruct_theta_musyn] = lfdata(cloop_theta_musyn);
M11_theta_musyn = M_theta_musyn(1,1);
omega_theta_musyn = info_theta_musyn.Frequency;
M11_g_theta_musyn = frd(M11_theta_musyn, omega_theta_musyn);
mu_bounds_rs_theta_musyn = mussv(M11_g_theta_musyn, blkstruct_theta_musyn, 's');

[perf_marg_theta_musyn, ~, ~, info_perf_theta_musyn] = robustperf(cloop_theta_mu
mu_bounds_rp_theta_musyn = info_perf_theta_musyn.MussvBnds;

figure;
semilogx(mu_bounds_rs_theta_musyn, 'b');
xlabel('Frequency (rad/sec)');
ylabel('Mu upper/lower bounds');
title('μ Plot - Robust Stability (θ) with MUSYN Controller');

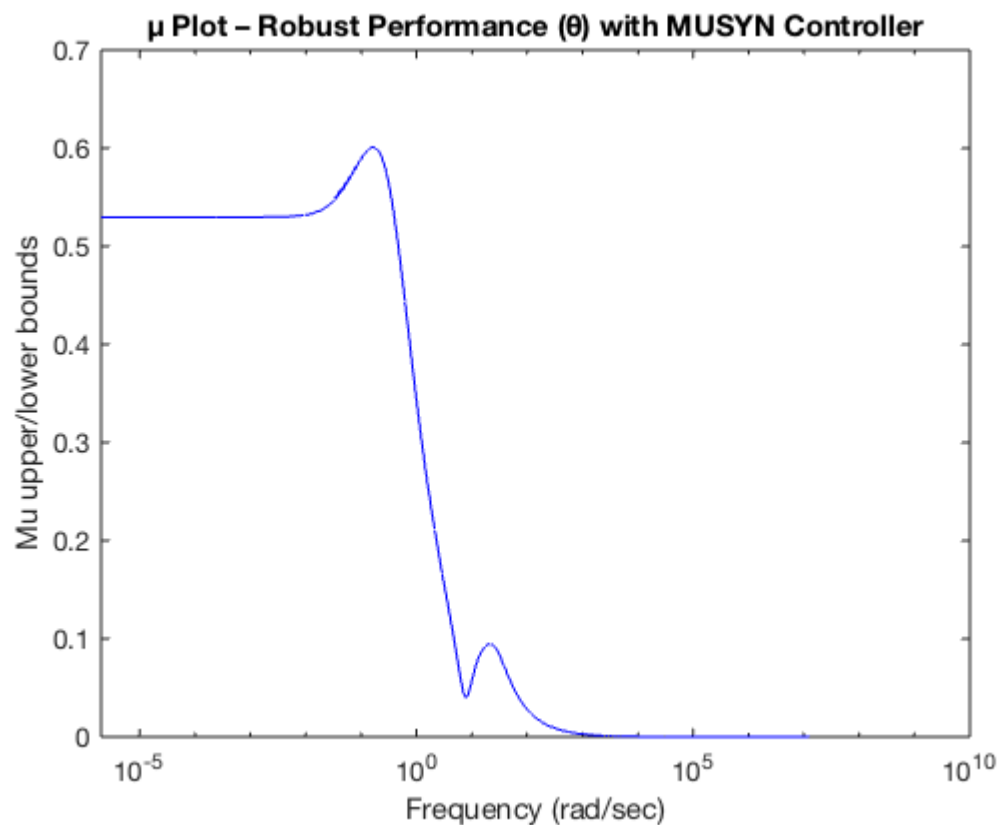
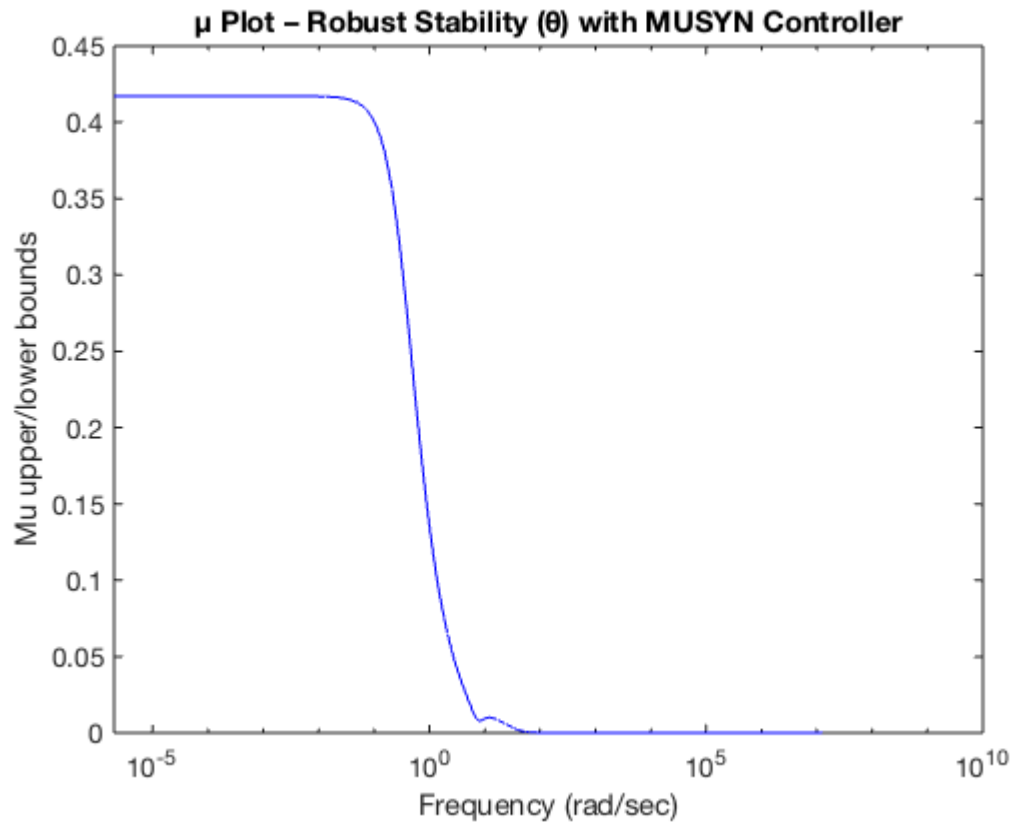
figure;
semilogx(mu_bounds_rp_theta_musyn, 'b');
xlabel('Frequency (rad/sec)');
ylabel('Mu upper/lower bounds');
title('μ Plot - Robust Performance (θ) with MUSYN Controller');

```

D-K ITERATION SUMMARY:

Robust performance				Fit order
Iter	K Step	Peak MU	D Fit	D
1	1.399	0.7706	0.7783	0
2	0.6362	0.5168	0.5206	0
3	0.5033	0.4896	0.4916	0
4	0.4904	0.488	0.4893	0
5	0.4892	0.4878	0.493	2

Best achieved robust performance: 0.488



Now, we focus on the φ channel. We use the same W_s weighting function scaled by a factor of 650, and apply MUSYN directly without further weight adjustments.

```
In [22]: W_s = 650 * (s^2 + 26.68*s + 185.23) / ((s + 0.001)*(s + 26.68)*(s + 1000));

opt = musynOptions('MaxIter', 5, 'TargetPerf', 0.3, 'TolPerf', 0, 'FitOrder', 50
P_for_synthesis_phi = augw(G_phi_unc, W_s, W_u, []);
[C_musyn_phi] = musyn(P_for_synthesis_phi, 1, 1, opt);
```

```

C_phi_musyn = C_musyn_phi;

systemnames = 'P_phi C_phi_musyn W_s W_u W_phi';
inputvar = '[yd]';

input_to_C_phi_musyn = '[yd -P_phi]';
input_to_P_phi = '[C_phi_musyn]';
input_to_W_s = '[yd -P_phi]';
input_to_W_u = '[C_phi_musyn]';
input_to_W_phi = '[P_phi]';

outputvar = '[W_s; W_u; W_phi]';
cleanup_sysic = 'yes';
cloop_phi_musyn = sysic;

[stabmarg_phi_musyn, ~, ~, info_phi_musyn] = robuststab(cloop_phi_musyn);
[M_phi_musyn, ~, blkstruct_phi_musyn] = lftdata(cloop_phi_musyn);
M11_phi_musyn = M_phi_musyn(1,1);
omega_phi_musyn = info_phi_musyn.Frequency;
M11_g_phi_musyn = frd(M11_phi_musyn, omega_phi_musyn);
mu_bounds_rs_phi_musyn = mussv(M11_g_phi_musyn, blkstruct_phi_musyn, 's');

[perf_marg_phi_musyn, ~, ~, info_perf_phi_musyn] = robustperf(cloop_phi_musyn);
mu_bounds_rp_phi_musyn = info_perf_phi_musyn.MussvBnds;

figure;
semilogx(mu_bounds_rs_phi_musyn, 'b');
xlabel('Frequency (rad/sec)');
ylabel('Mu upper/lower bounds');
title('μ Plot - Robust Stability (φ) with MUSYN Controller');

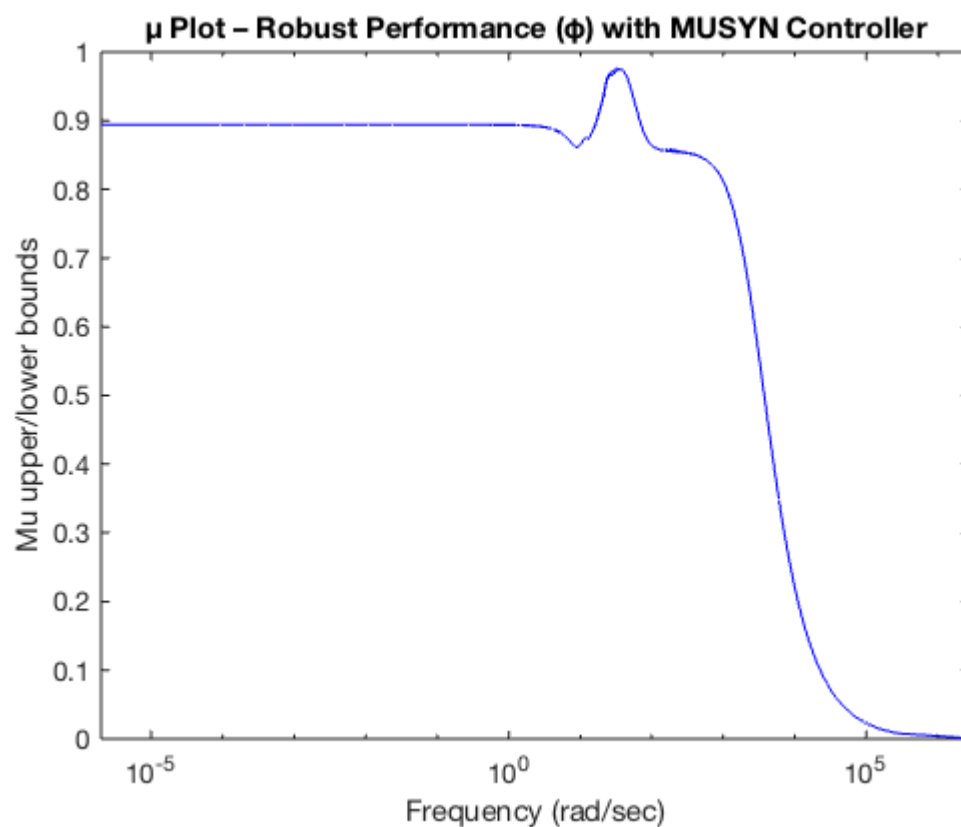
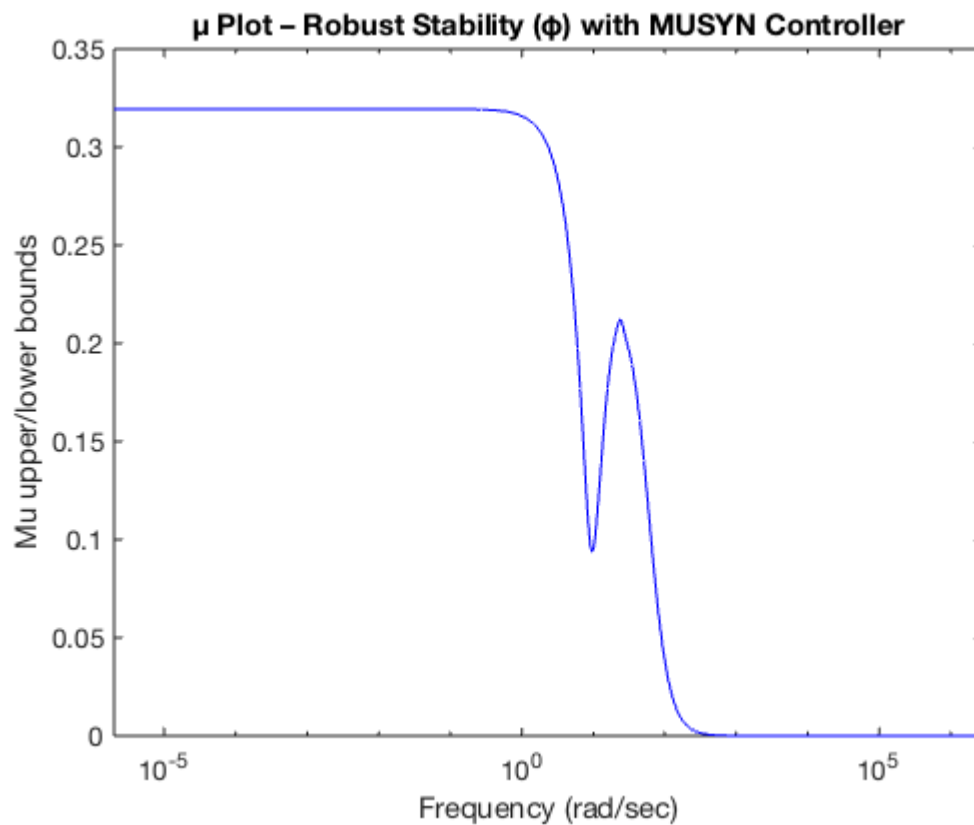
figure;
semilogx(mu_bounds_rp_phi_musyn, 'b');
xlabel('Frequency (rad/sec)');
ylabel('Mu upper/lower bounds');
title('μ Plot - Robust Performance (φ) with MUSYN Controller');

```

D-K ITERATION SUMMARY:

Robust performance				Fit order
Iter	K Step	Peak MU	D Fit	D
1	1.67	1.141	1.153	8
2	0.914	0.9135	0.9228	18
3	0.8688	0.8688	0.8726	18
4	0.8663	0.8663	0.867	18
5	0.8649	0.8647	0.8652	18

Best achieved robust performance: 0.865



3.3 Comparing the H-infinity controller with the μ -synthesis controllers

```
In [23]: C_musyn_tetha=tf(C_musyn_tetha)
C_theta=tf(C_theta)

C_musyn_phi=tf(C_musyn_phi)
C_phi=tf(C_phi)
```


C_musyn_tetha =

$$\begin{aligned} &1.12 \, s^7 + 2.891e04 \, s^6 + 2.923e07 \, s^5 + 1.475e09 \, s^4 + 3.577e10 \, s^3 \\ &\quad + 6.232e11 \, s^2 + 5.317e12 \, s + 1.503e13 \end{aligned}$$

$$\begin{aligned} &s^8 + 2.584e04 \, s^7 + 2.649e07 \, s^6 + 1.704e09 \, s^5 + 5.392e10 \, s^4 \\ &\quad + 9.169e11 \, s^3 + 6.609e12 \, s^2 + 3.287e13 \, s + 3.286e10 \end{aligned}$$

Continuous-time transfer function.

C_theta =

$$\begin{aligned} &2.369e04 \, s^7 + 5.982e08 \, s^6 + 2.858e11 \, s^5 + 1.872e13 \, s^4 + 4.554e14 \, s^3 \\ &\quad + 8.612e15 \, s^2 + 7.941e16 \, s + 2.344e17 \end{aligned}$$

$$\begin{aligned} &s^8 + 2.715e04 \, s^7 + 6.056e07 \, s^6 + 3.819e10 \, s^5 + 3.997e12 \, s^4 \\ &\quad + 2.165e14 \, s^3 + 4.539e15 \, s^2 + 2.44e16 \, s + 2.44e13 \end{aligned}$$

Continuous-time transfer function.

C_musyn_phi =

$$\begin{aligned} &4.341e04 \, s^{25} + 5.287e08 \, s^{24} + 1.868e12 \, s^{23} + 1.588e15 \, s^{22} \\ &\quad + 4.729e17 \, s^{21} + 7.149e19 \, s^{20} + 6.722e21 \, s^{19} + 4.557e23 \, s^{18} \\ &\quad + 2.395e25 \, s^{17} + 1.02e27 \, s^{16} + 3.62e28 \, s^{15} + 1.089e30 \, s^{14} \\ &\quad + 2.813e31 \, s^{13} + 6.288e32 \, s^{12} + 1.222e34 \, s^{11} + 2.071e35 \, s^{10} \\ &\quad + 3.061e36 \, s^9 + 3.936e37 \, s^8 + 4.385e38 \, s^7 + 4.196e39 \, s^6 \\ &\quad + 3.405e40 \, s^5 + 2.295e41 \, s^4 + 1.244e42 \, s^3 + 5.125e42 \, s^2 \\ &\quad + 1.439e43 \, s + 2.071e43 \end{aligned}$$

$$\begin{aligned} &s^{26} + 1.514e04 \, s^{25} + 8.002e07 \, s^{24} + 1.743e11 \, s^{23} + 1.517e14 \, s^{22} \\ &\quad + 5.174e16 \, s^{21} + 9.478e18 \, s^{20} + 1.103e21 \, s^{19} + 8.997e22 \, s^{18} \\ &\quad + 5.489e24 \, s^{17} + 2.634e26 \, s^{16} + 1.028e28 \, s^{15} + 3.334e29 \, s^{14} \\ &\quad + 9.127e30 \, s^{13} + 2.129e32 \, s^{12} + 4.257e33 \, s^{11} + 7.325e34 \, s^{10} \end{aligned}$$

$$\begin{aligned}
& + 1.084e36 \, s^9 + 1.378e37 \, s^8 + 1.496e38 \, s^7 + 1.371e39 \, s^6 \\
& + 1.044e40 \, s^5 + 6.399e40 \, s^4 + 3.012e41 \, s^3 + 9.805e41 \, s^2 \\
& + 1.737e42 \, s + 1.736e39
\end{aligned}$$

Continuous-time transfer function.

C_phi =

$$\begin{aligned}
& 1.258e04 \, s^7 + 6.534e06 \, s^6 + 6.665e08 \, s^5 + 3.016e10 \, s^4 + 7.851e11 \, s^3 \\
& + 1.256e13 \, s^2 + 1.142e14 \, s + 3.868e14 \\
& \hline
& s^8 + 2032 \, s^7 + 1.172e06 \, s^6 + 1.496e08 \, s^5 + 9.86e09 \, s^4 + 3.484e11 \, s^3 \\
& + 5.991e12 \, s^2 + 3.843e13 \, s + 3.843e10
\end{aligned}$$

Continuous-time transfer function.

Use Hankel-based reduction to lower the controller order if necessary.

The controller order for θ is acceptable, but the φ controller requires model reduction using Hankel-based methods (e.g., 'reduce'). Two reduced orders were tested for the φ controller. Order 5 was found to be the most suitable, as the frequency responses of the original and reduced controllers closely match.

```
In [24]: hankelsv(C_musyn_phi);
C_musyn_phi_reduced1=reduce(C_musyn_phi,3);
bodemag(C_musyn_phi,'b',C_musyn_phi_reduced1,'r--');
C_musyn_phi_reduced1=tf(C_musyn_phi_reduced1)

C_musyn_phi_reduced=reduce(C_musyn_phi,5);
bodemag(C_musyn_phi,'b',C_musyn_phi_reduced,'r--');
C_musyn_phi_reduced=tf(C_musyn_phi_reduced)
```

C_musyn_phi_reduced1 =

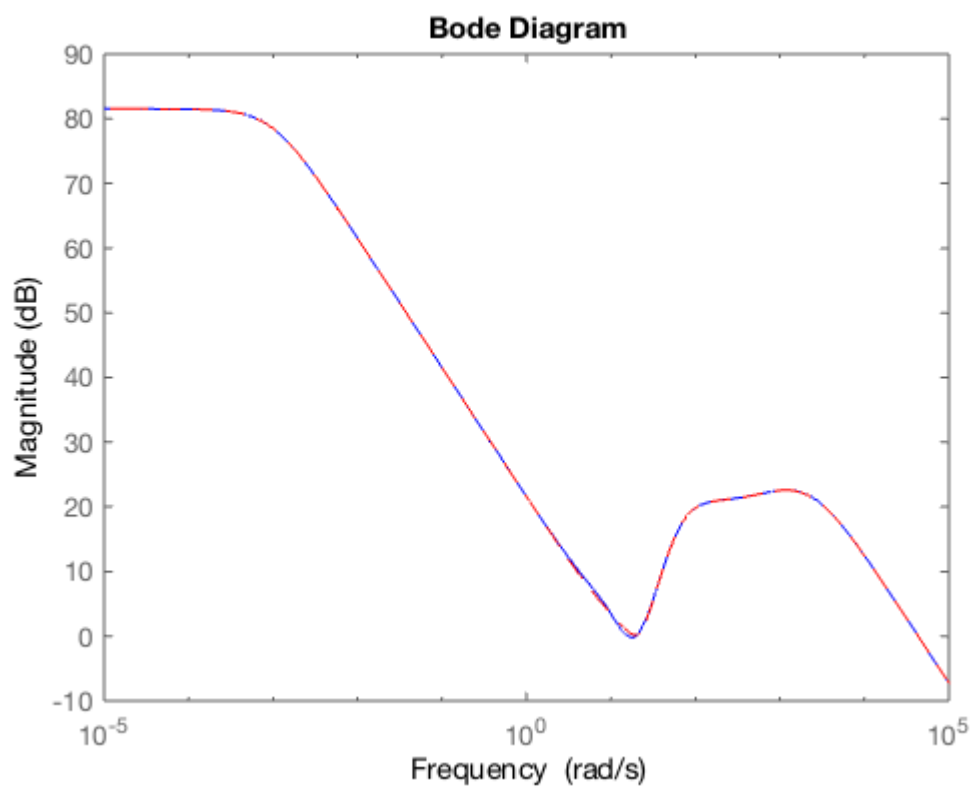
$$\frac{4.422e04 s^2 + 3.551e05 s + 5.251e06}{s^3 + 3197 s^2 + 4.401e05 s + 440.2}$$

Continuous-time transfer function.

C_musyn_phi_reduced =

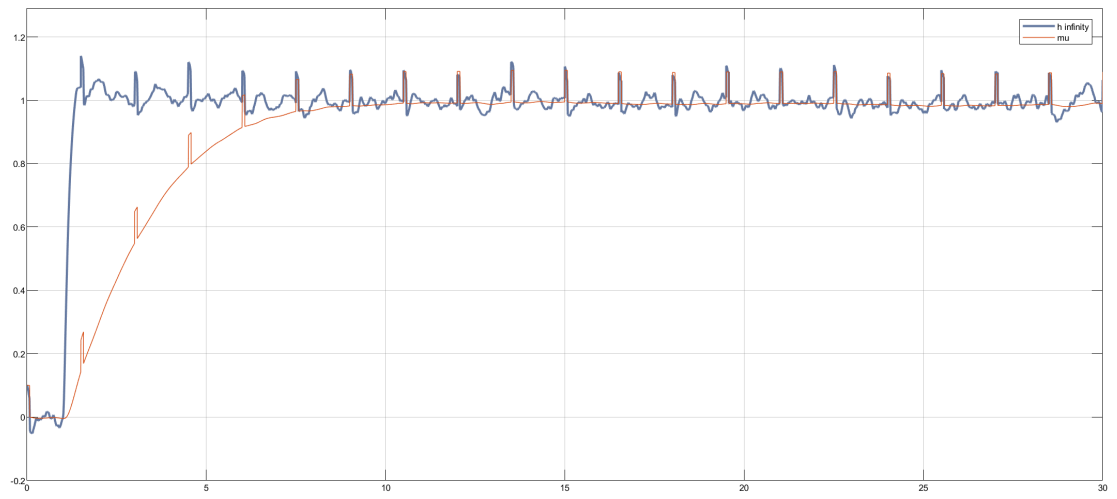
$$\frac{4.341e04 s^4 + 3.008e07 s^3 + 8.491e08 s^2 + 1.979e10 s + 1.42e11}{s^5 + 3660 s^4 + 2.926e06 s^3 + 2.682e08 s^2 + 1.191e10 s + 1.191e07}$$

Continuous-time transfer function.

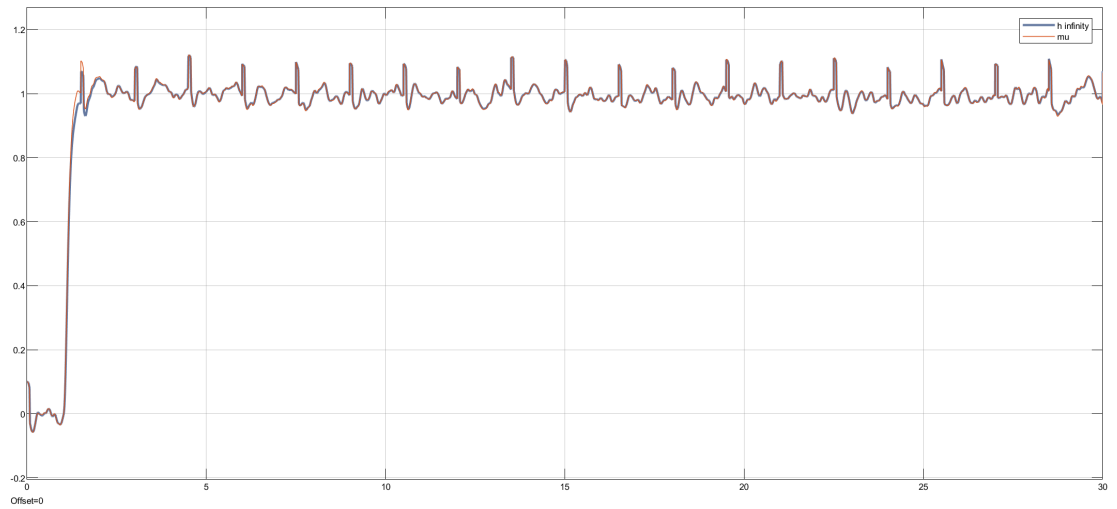


Simulation

The next step is to simulate the closed-loop system in Simulink and compare the performance of the controllers designed using μ -synthesis and H-infinity approaches for theta angle



for ϕ angle



In []: