Robust Control Systems

Uncertainty and Robust Stability: Some MATLAB Examples

Hassan Bevrani

Professor, University of Kurdistan

Fall 2023

Contents

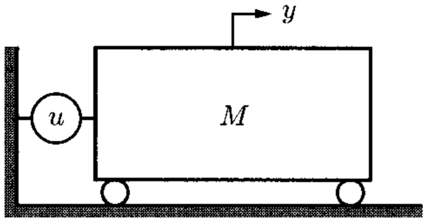
Reference

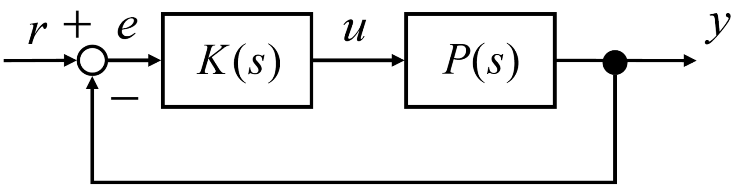
1. M. Hirata, **Practical Robust Control**,
CORONA Press , 2017 (In Japanese).

Mass Control System (Nominal)

One Mass Model:

$$P = \frac{1}{Ms^2}$$





PD Controller:

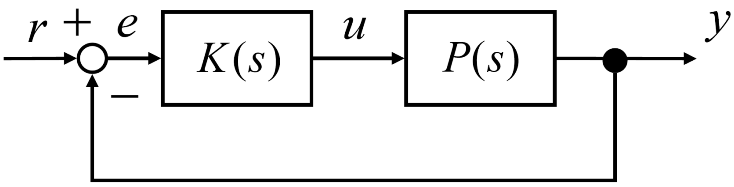
$$K = k_p + sk_d$$

$$1 + PK = 0 \quad \Rightarrow \quad s^2 + \frac{k_d}{M}s + \frac{k_p}{M} = 0$$

Continue

$$P = \frac{1}{Ms^2}$$

$$K = k_p + sk_d$$



$$s^2 + \frac{k_d}{M}s + \frac{k_p}{M} = 0$$

$$s^2 + 2\eta\omega_n s + \omega_n^2$$

$$k_p = \omega_n^2 M \qquad k_d = 2\eta\omega_n M$$

$$K = \omega_n^2 M + 2\eta\omega_n M s$$

MATLAB Example 1

Assuming $M = 1$ and $\eta = 1$, for the following ω_n values, plot the step response and bode diagram of the closed-loop system:

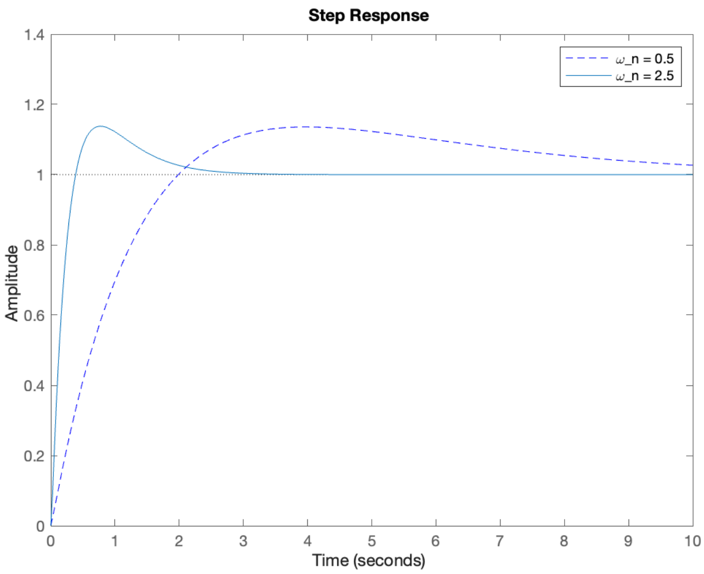
$$\omega_n = 2.5$$

$$\omega_n = 0.5$$

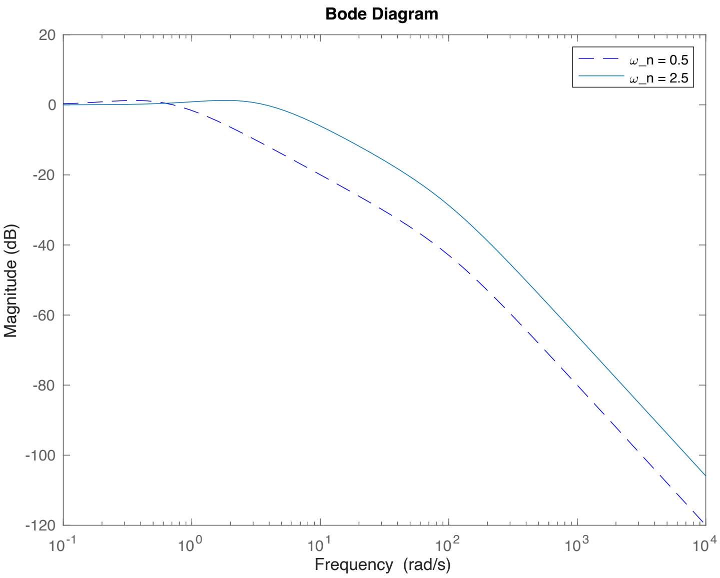
MATLAB Program 1

```
%% Mass System Control
rng('default') % Initializing random numbers
%% Defining the nominal model(one-mass model)
s = tf('s'); % Definition of Laplace operator s
M = 1; % mass
Pn = 1/(M*s^2); % nominal model(1/(Ms^2)
Eta=1;
%% PD controller[approximate differentiator s/(0.01s+1)]
Omega_n = 0.5;
K1 = Omega_n^2*M + 2*Eta*Omega_n*M*s/(0.01*s+1);
Omega_n = 2.5;
K2 = Omega_n^2*M + 2*Eta*Omega_n*M*s/(0.01*s+1);
%% Response calculation for nominal model
Tn1 = feedback(Pn*K1,1);
Tn2 = feedback(Pn*K2,1);
figure(1);
step(Tn1,'--',Tn2,10);
ylim([0 1.4]);
legend('\omega_n = 0.5','\omega_n = 2.5');
figure(2);
bodemag(Tn1,'--',Tn2);
legend('\omega_n = 0.5','\omega_n = 2.5');
```


Results



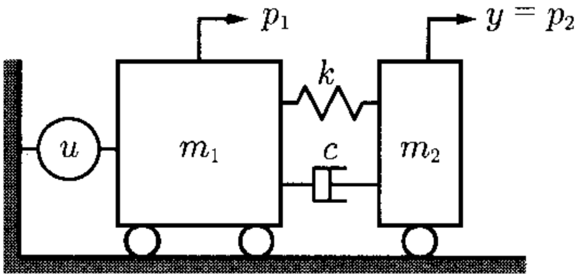
figure(1)



figure(2)

Mass Plant with additional Mass-Spring-Damper System (Uncertainty)

Two-Mass Model:



$$P = \frac{cs + k}{s^2 \left[m_1 m_2 s^2 + (m_1 + m_2)cs + (m_1 + m_2)k\right]}$$

$$M = m_1 + m_2$$

MATLAB Example 2

The nominal parameters and uncertainties are given in the following Table. Plot the bod diagrams of the uncertain plant and compare it with the given plant in Example 1.
Assume $M = 1$.

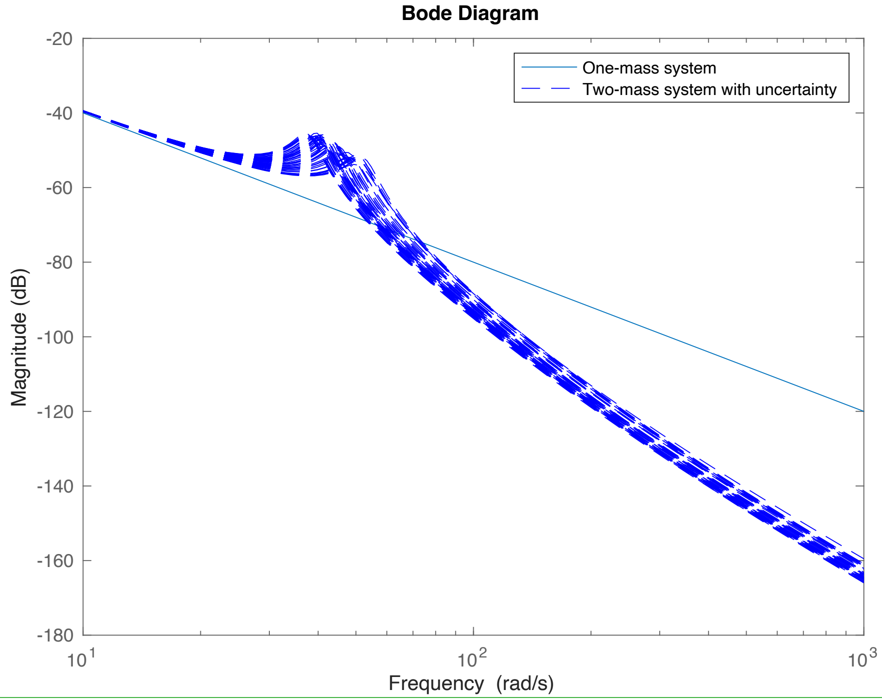
Parameters	m_1	k	c
Nominal	0.8	300	1
Perturbation	$\pm 10\%$	$\pm 10\%$	$\pm 10\%$

MATLAB Program 2

```
%% Definition of perturbation model
% Define the fluctuation parameters using ureal
m1 = ureal('m1',0.8,'percent',10); % 10% perturbation
m2 = M - m1; % m1+m2=M
k = ureal('k',300,'percent',10); % 10% perturbation
c = ureal('c',1,'percent',10); % 10% perturbation
% Two mass model definition
P = (c*s+k)/(s^2*(m1*m2*s^2 + (m1+m2)*c*s + (m1+m2)*k));
% Select 50 models from the model set
P = usample(P,50);
figure(3)
bodemag(Pn,P,'--',{1e1,1e3});
legend('One-mass system','Two-mass system with
uncertainty');
```


Results

`figure(3)`



Uncertain Mass System Control: MATLAB Example 3

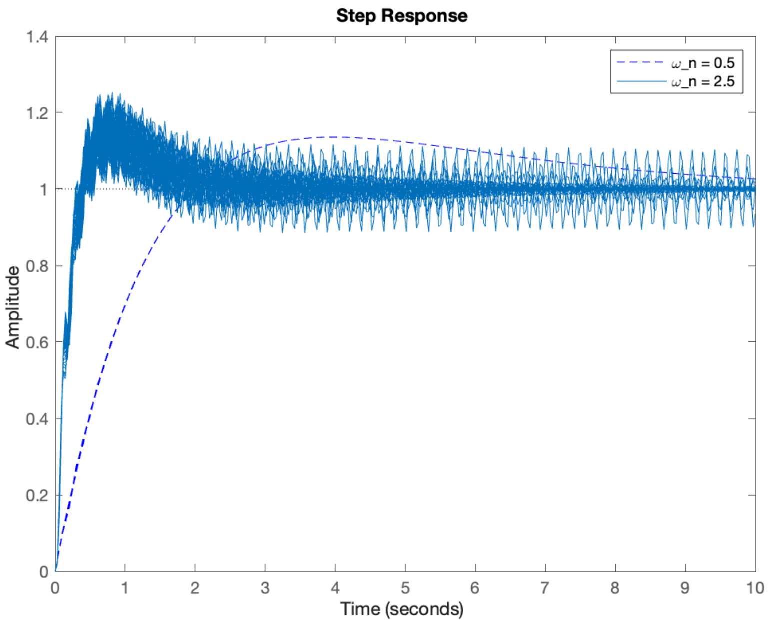
Evaluate the closed-loop system for the designed PD controllers in Example 1, and the uncertain plant of Example 2:

MATLAB Program 3

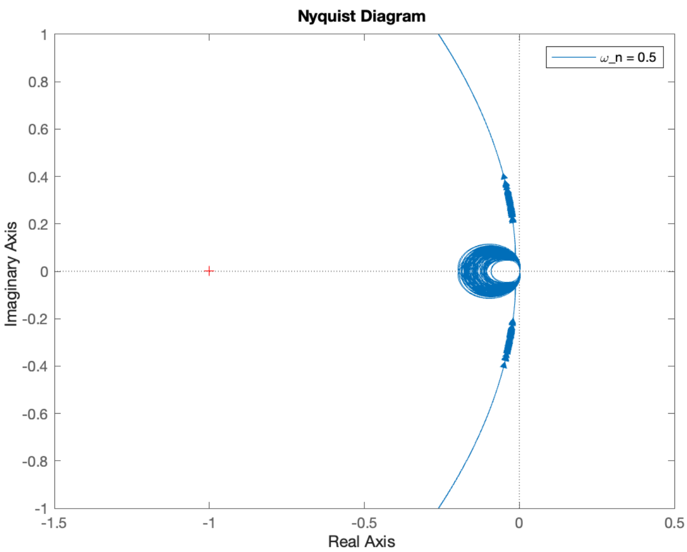
```
%% Response Evaluation for perturbation model
T1 = feedback(P*K1,1);
T2 = feedback(P*K2,1);
figure(4);
step(T1,'--',T2,10);
ylim([0 1.4]);
legend('\omega_n = 0.5','\omega_n = 2.5');
figure(5);
nyquist(P*K1);
axis([-1.5 0.5 -1 1]);
legend('\omega_n = 0.5');
figure(6);
nyquist(P*K2);
axis([-1.5 0.5 -1 1]);
legend('\omega_n = 2.5');
```

Results

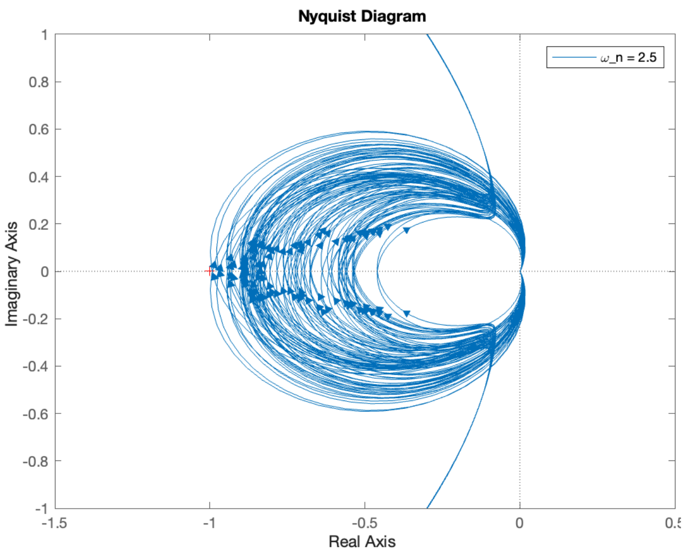
figure(4)



Results



figure(5)



figure(6)

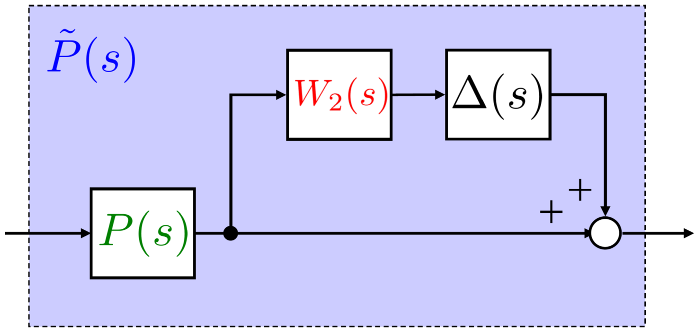
Multiplicative Uncertainty Modeling: MATLAB Example 4

Uncertainty in a system is modeled in form of output multiplicative as follows:

$$P = \frac{1}{s+1}$$

$$W_2 = \frac{2s}{s+10}$$

$$\|\Delta\|_\infty \leq 1$$



Plot the bode diagrams of nominal and uncertain systems.

MATLAB Program 4

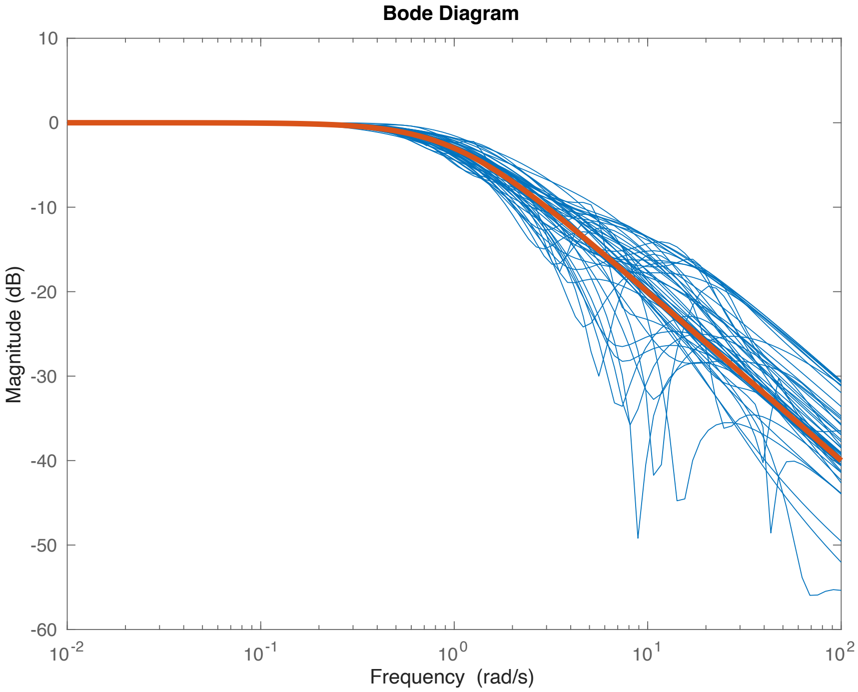
```
%% Multiplicative Uncertainty


close all;
clear all;
rng('default'); % Initializing random numbers


s = tf('s');
Pn = 1/(s+1); % Nominal model
W2 = 2*s/(s+10); % Gain characteristics of multiplicative
uncertainty
delta = ultidyn('delta',[1 1],'SampleStateDim',4);
% meaning of arguments
% 'delta' : perturbation name
% [1 1] : Perturbation size (1 row and 1 column)
% 'SampleStateDim' : perturbation order
P = (1+W2*delta)*Pn;
P = usample(P,50);
w = logspace(-2,2,100);
bodemag(P,Pn,w);
```

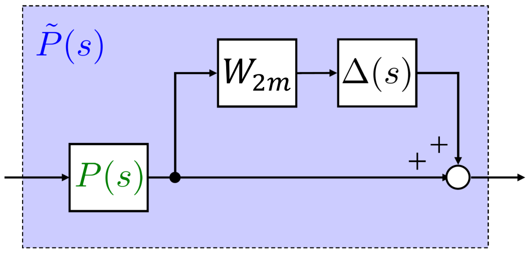
Results

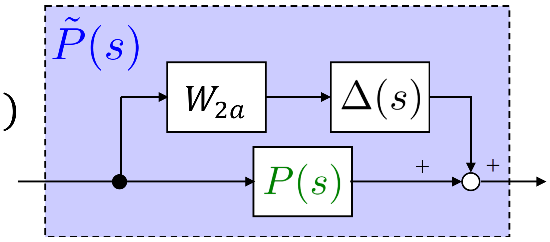
figure(4)



Multiplicative and Additive Uncertainty Modeling: MATLAB Example 5

Uncertainty in a standard 2nd order system is modeled in the following multiplicative and additive forms:





$$P = \frac{{\omega_n}^2}{s^2 + 2\eta\omega_n s + {\omega_n}^2} \qquad \|\Delta\|_\infty \le 1$$

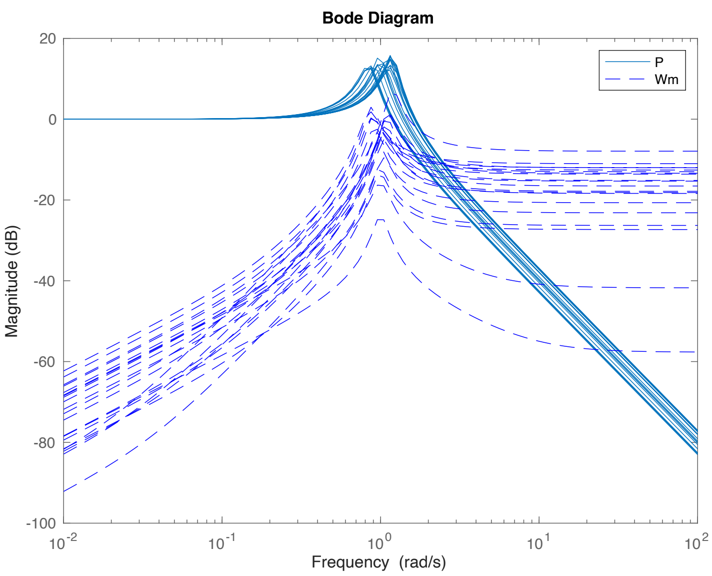
Parameters perturbation: η = 0.1 $\pm$20% (0.1)
ω_n = 1 $\pm$20% (1)

Plot the bode diagrams of uncertain system and uncertainties.

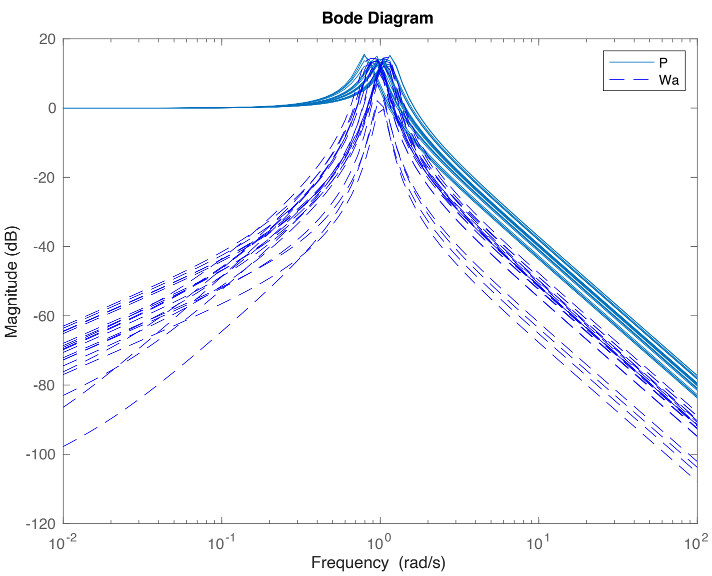
MATLAB Program 5

```
%% Multiplicative and Additive Uncertainty Models
close all; clear all;
rng('default'); % Initializing random numbers
% Define real variation parameters
Eta_n = ureal('omega',1,'percent',20);
zeta = ureal('zeta',0.1,'percent',20);
% Transfer function definition
s = tf('s');
P = Eta_n^2/(s^2+2*zeta*Eta_n*s+Eta_n^2);
% Frequency response calculation
w = logspace(-2,2,100);
P_g = ufrd(P,w); % Use "ufrd" instead of "frd" when including uncertainties
% multiplicative perturbation
Wm = (P_g - P_g.nominal)/P_g.nominal;
% additive Uncertainty
Wa = P_g - P_g.nominal;
% Gain plot
figure(1);
bodemag(P_g,Wm,'--');
legend('P','Wm');
figure(2);
bodemag(P_g,Wa,'--');
legend('P','Wa');
```


Results



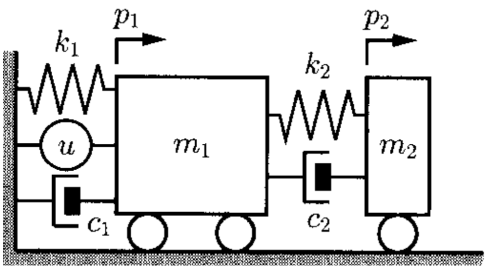
figure(1)



figure(2)

4th-order Mass-Spring-Damper System with Uncertainty

System and Dynamic Model:



$$\begin{cases} m_1\ddot{p}_1 = K_s u - k_1 p_1 - c_1\dot{p}_1 - k_2(p_1 - p_2) - c_2(\dot{p}_1 - \dot{p}_2) \\ m_2\ddot{p}_2 = -k_2(p_2 - p_1) - c_2(\dot{p}_2 - \dot{p}_1) \end{cases}$$

$$\Rightarrow \begin{cases} m_1\ddot{p}_1 + (c_1 + c_2)\dot{p}_1 - c_2\dot{p}_2 + (k_1 + k_2)p_1 - k_2 p_2 = K_s u \\ m_2\ddot{p}_2 - c_2\dot{p}_1 + c_2\dot{p}_2 - k_2 p_1 + k_2 p_2 = 0 \end{cases}$$

$$p = [p_1, p_2]^T \Rightarrow \boxed{M\ddot{p} + C\dot{p} + Kp = Fu}$$

$$M = \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix}, \quad C = \begin{bmatrix} c_1 + c_2 & -c_2 \\ -c_2 & c_2 \end{bmatrix}$$

$$K = \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix}, \quad F = \begin{bmatrix} K_s \\ 0 \end{bmatrix}$$

Continue

$$M\ddot{p} + C\dot{p} + Kp = Fu$$

$$M = \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix}, \quad C = \begin{bmatrix} c_1 + c_2 & -c_2 \\ -c_2 & c_2 \end{bmatrix}$$

$$K = \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix}, \quad F = \begin{bmatrix} K_s \\ 0 \end{bmatrix}$$

$$\Rightarrow \begin{cases} x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} p \\ \dot{p} \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ \hdashline \dot{p}_1 \\ \dot{p}_2 \end{bmatrix} \\ \dot{x}_1 = \dot{p} = x_2 \\ \dot{x}_2 = \ddot{p} = M^{-1}(-Kp - C\dot{p} + Fu) \\ \quad = -M^{-1}Kx_1 - M^{-1}Cx_2 + M^{-1}Fu \end{cases}$$

$$\Rightarrow \boxed{\dot{x} = A_p x + B_p u}$$

$$A_p = \begin{bmatrix} O & I \\ -M^{-1}K & -M^{-1}C \end{bmatrix}, \quad B_p = \begin{bmatrix} O \\ M^{-1}F \end{bmatrix}$$

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

$$\Rightarrow \boxed{P = (A_p, B_p, C_p, 0)}$$

MATLAB Example 6

The nominal parameters and uncertainty are given bellow. Plot the bod diagrams of the uncertain plant.

Uncertainty:

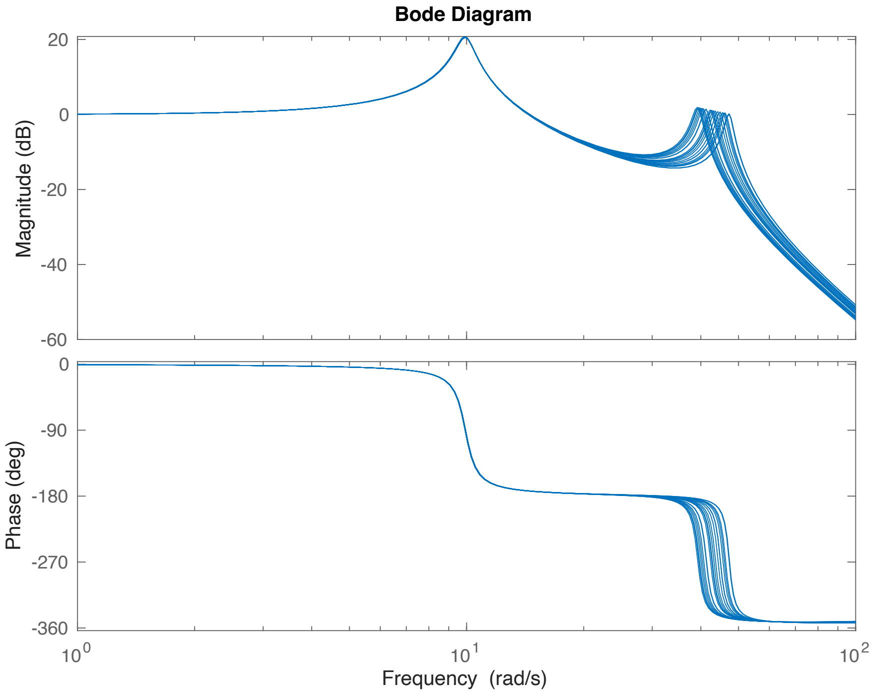
20% perturbation in k_2

Parameters	Value
m_1	$0.8\,\mathrm{kg}$
m_2	$0.2\,\mathrm{kg}$
k_1	$100\,\mathrm{N/m}$
k_2	$300\,\mathrm{N/m}$
c_1	$1\,\mathrm{Ns/m}$
c_2	$0.3\,\mathrm{Ns/m}$
K_s	$100\,\mathrm{N/V}$

MATLAB Program 6

```
%% 4th order Mass-Spring-Damper System
%% Parameter definition
m1 = 0.8; m2 = 0.2; k1 = 100;
k2 = ureal('k2',300,'percent',20);
c1 = 1; c2 = 0.3; Ks = 100;
%% Define the M, K, C matrices of the motion equation
M = [ m1, 0 ; 0, m2 ];
C = [ c1+c2, -c2 ; -c2, c2 ];
K = [ k1+k2, -k2 ; -k2, k2 ];
F = [ Ks ; 0 ];
%% State space realization
iM = inv(M);
Ap = [ zeros(2,2), eye(2,2) ; -iM*K, -iM*C ];
Bp = [ zeros(2,1) ; iM*F ];
Cp = [ 0 1 0 0 ];
Dp = 0;
%% Bode plot of the plant
P = ss(Ap,Bp,Cp,Dp);
figure(1);
bode(P,{1e0,1e2}); % Bode plot
```

Results



Uncertainties and its Cover (MATLAB Program 7)

If for the 4th order plant given in example 6, the uncertainty transfer function (cover) is determined as follows. Plot multiplicative uncertainties.

MATLAB Program 7

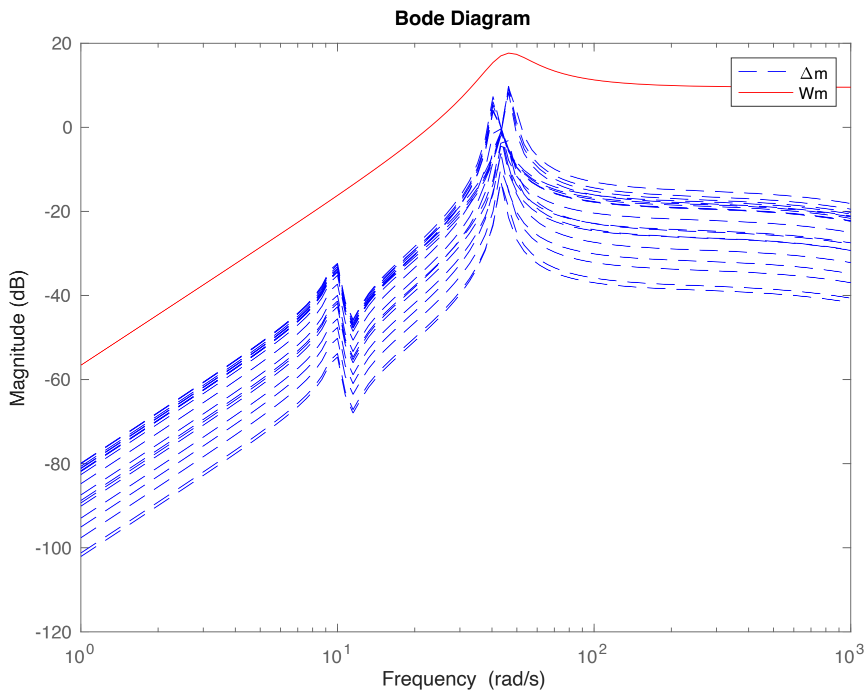
```
%% Uncertainties (Deltam) and its Cover (Wm)

%% Multiplicative uncertainty model
w = logspace(0,3,100); % Definition of frequency vector
P_g = ufrd(P,w); % Frequency response calculation
Dm_g = (P_g - P_g.nominal)/P_g.nominal; % Computing multiplicative uncertainty

%% Definition of weight Wm
s = tf('s');
Wm = 3*s^2/(s^2+18*s+45^2);

%% Gain plot
bodemag(Dm_g,'--',Wm,'r-',w);
legend('\Deltam','Wm');
```


Results



Thank You!