Robust Control Systems

H∞ Robust Control Design Example

Hassan Bevrani

Professor, University of Kurdistan

Fall 2023

Contents

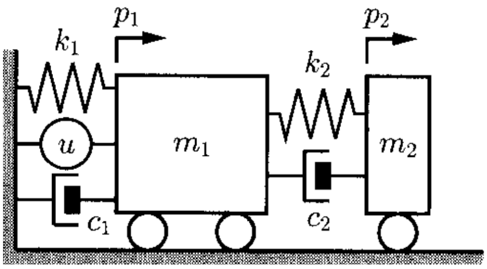
1. Mass-Spring-Damper Control System
2. State-Space Model
3. Nominal Parameters and Uncertainties
4. Mixed-Sensitivity Problem
5. H∞ Control Design
6. Simulation Results

Reference

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

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

System and Dynamic Model:



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

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

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

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

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

Continue

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

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

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

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} p \\ \dot{p} \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ \dot{p}_1 \\ \dot{p}_2 \end{bmatrix}$$

$$\dot{x}_1 = \dot{p} = x_2$$

$$\dot{x}_2 = \ddot{p} = M^{-1}(-Kp - C\dot{p} + Fu)$$

$$= -M^{-1}Kx_1 - M^{-1}Cx_2 + M^{-1}Fu$$

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

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

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

$$P = (A_p, B_p, C_p, 0)$$

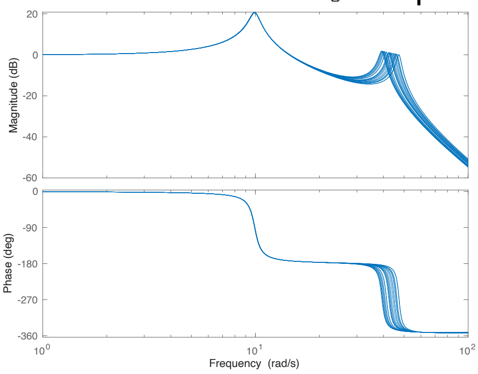
Modeling Using MATLAB Codes (from MATLAB Program 6)

The nominal parameters and uncertainty are assumed as:

Uncertainty: 20% perturbation in k_2

Parameters	Nominal Value
m_1	0.8 kg
m_2	0.2 kg
k_1	100 N/m
k_2	300 N/m
c_1	1 Ns/m
c_2	0.3 Ns/m
K_s	100 N/V

```
%% 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);
bode(P,{1e0,1e2}); % Bode plot
```

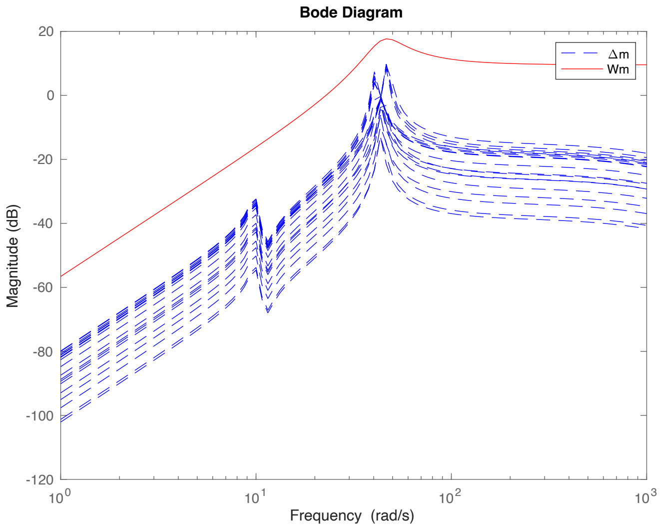


Uncertainties and its Cover (from MATLAB Program 7)

Considering the multiplicative uncertainties, one can give the following weighting transfer function to cover all uncertainties.

$$W_m = \frac{3\,s^2}{s^2 + 2\times 0.2\times 45s + 45^2}$$

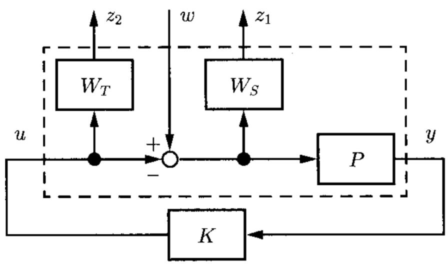
```
%% Uncertainties (Deltam) and its Cover function (Wm)

%% Multiplicative uncertainty model
w = logspace(0,3,100); % Definition of frequency vector
P_g = ufrd(P,w); % Frequency response calculation

%% Computing multiplicative uncertainty:
Dm_g = (P_g - P_g.nominal)/P_g.nominal;

%% 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');
```



H∞ Control Design Using Mixed-Sensitivity Problem

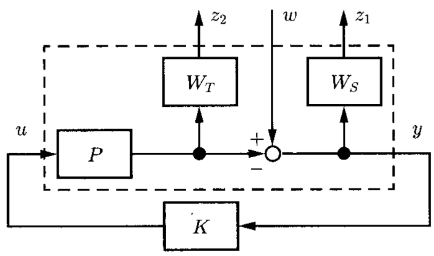
Performance Requirement: To have a small enough sensitivity function at low frequency, we can select the following performance weighting function:

$$W_S = \frac{15}{s + 0.015}$$



Input-side mixed-sensitivity problem

Mixed-Sensitivity Problem: Now, we can formulate the control design problem via a mixed-sensitivity problem (MSP), which can be considered in form of input-side MSP or output-side MSP.



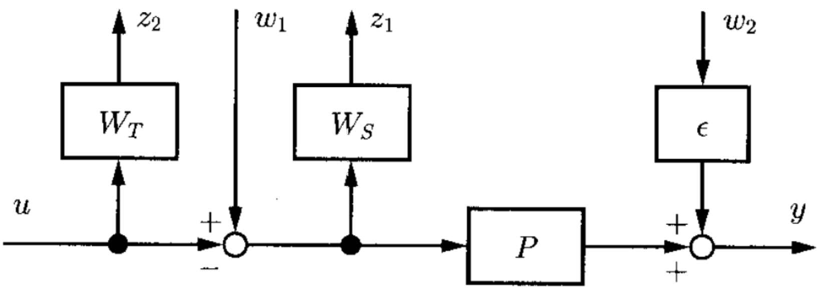
Output-side mixed-sensitivity problem

$$\| W_S S \|_\infty < 1, \quad \| W_T T \|_\infty < 1 \;\Rightarrow\; \left\| \begin{bmatrix} W_S S \\ W_T T \end{bmatrix} \right\|_\infty < 1$$

$$G_{zw} = \begin{bmatrix} W_S S \\ W_T T \end{bmatrix}$$

Framework Preparation

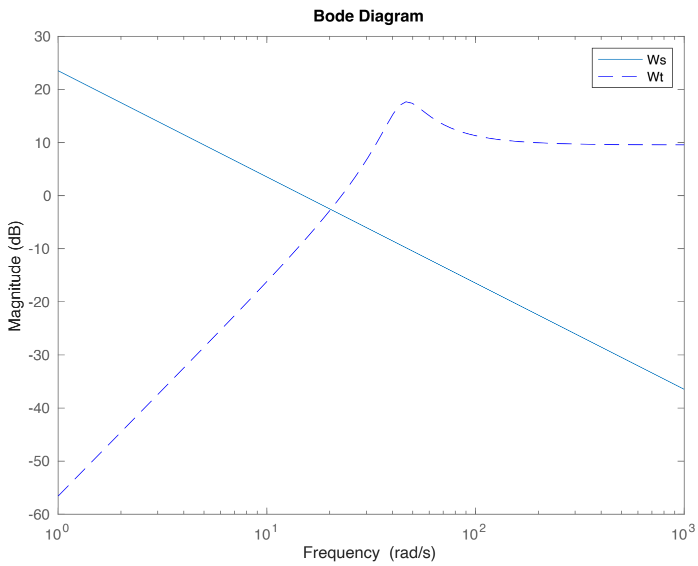
Using the input-side MSP the following configuration can be reached. To satisfy the full rankness of the D_{12} in the given generalized plant, a new input with gain of $\epsilon = 5\times10^{-4}$ is added.



Building the Generalized Plant

MATLAB codes for plotting the uncertainty and performance weighting functions and building the generalized plant.

```
%% Determine of weighting functions
s = tf('s');
Ws = 15/(s + 1.5e−2); % Ws
Wm = 3*s^2/(s^2+18*s+45^2);
Wt = Wm; % Wt
Weps = 5e−4; % Weps
figure(3);
w=logspace(0,3,100);
bodemag(Ws,Wt,'−−',w);
legend('Ws','Wt');
%% Building of generalized plant
Pn = P.nominal;
systemnames = 'Pn Ws Wt Weps';
inputvar = '[w1; w2; u]';
outputvar = '[Ws; Wt; Pn+Weps]';
input_to_Pn = '[w1 − u]';
input_to_Ws = '[w1 − u]';
input_to_Wt = '[ u ]';
input_to_Weps = '[ w2 ]';
G = sysic;
```



H∞ Control Design (MATLAB Codes)

```
%% Hinf Control Design using Mixed sensitivity problem
clear all; close all;
rng('default'); % Initializing random numbers
%% Parameter definition
m1 = 0.8; m2 = 0.2; k1 = 100; c1 = 1; c2 = 0.3; Ks = 100; k2 = ureal('k2',300,'percent',20);
%% 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;
P = ss(Ap,Bp,Cp,Dp);
%% Multiplicative uncertainty model
w = logspace(0,3,100);
P_g = ufrd(P,w); % Frequency response calculation
Dm_g = (P_g - P_g.nominal)/P_g.nominal;
s = tf('s');
Wt = 3*s^2/(s^2+18*s+45^2); %Uncertainty weighting function Wt
Ws = 15/(s + 1.5e-2); % Performance weighting function
Weps = 5e-4; % Weps
%% Building of generalized plant
Pn = P.nominal;
systemnames = 'Pn Ws Wt Weps';
inputvar = '[w1; w2; u]';
outputvar = '[Ws; Wt; Pn+Weps]';
input_to_Pn = '[w1 - u]';
input_to_Ws = '[w1 - u]';
input_to_Wt = '[ u ]';
input_to_Weps = '[ w2 ]';
G = sysic;
%% H-inf Controller Design
[K,clp,gamma_min,hinf_info] = hinfsyn(G,1,1,'display','on')
```

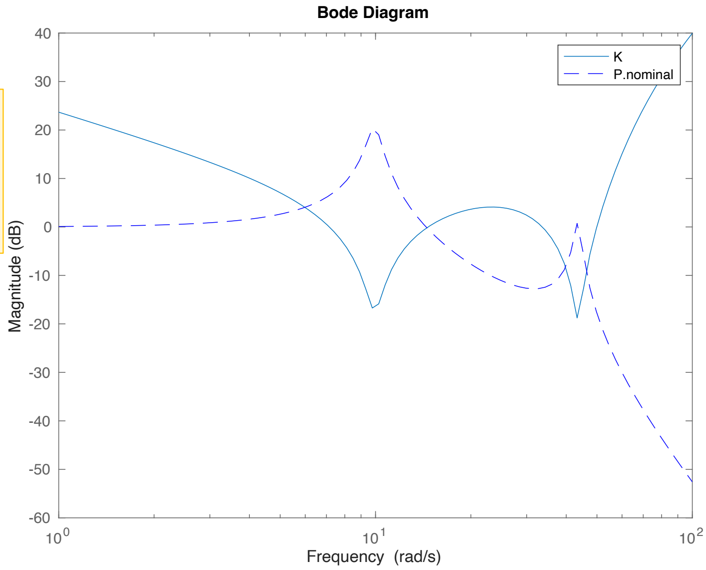

Synthesis Results

```
%% H-inf Controller Design
[K,clp,gamma_min,hinf_info] = hinfsyn(G,1,1,'display','on')
```

```
 Test bounds:  0.55 <=  gamma  <=  1.51

   gamma        X>=0        Y>=0       rho(XY)<1    p/f
  9.10e-01    -6.6e-16    -1.2e-13     1.708e+00 #   f
  1.17e+00     0.0e+00     1.5e-16     3.938e-01     p
  1.03e+00     0.0e+00     1.2e-15     6.800e-01     p
  9.69e-01     0.0e+00     1.0e-15     9.902e-01     p
  9.39e-01     1.6e-19    -2.1e-14     1.260e+00 #   f
  9.54e-01    -8.6e-16    -9.6e-14     1.110e+00 #   f
  9.61e-01    -3.3e-16    -1.6e-14     1.047e+00 #   f
  Limiting gains...
  9.71e-01     0.0e+00     2.4e-15     9.760e-01     p
  9.71e-01     0.0e+00     0.0e+00     9.761e-01     p

  Best performance (actual): 0.971
```

Controller Characteristics

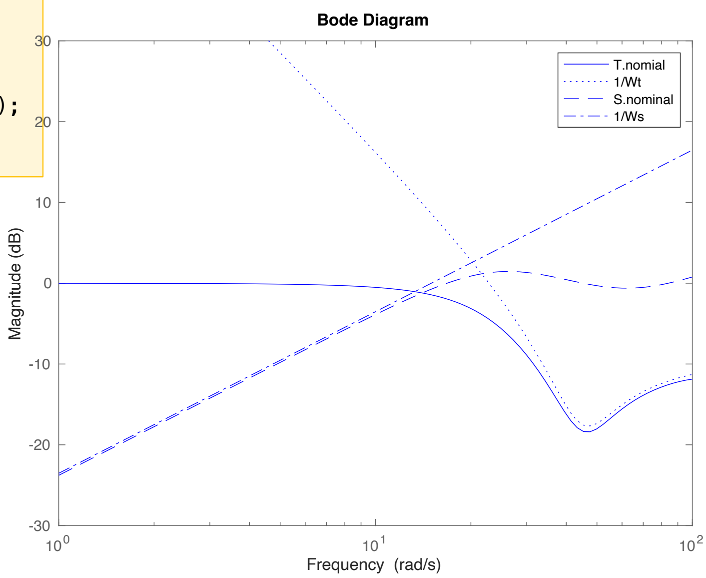
```
%% Controller gain plot
figure(1);
w = logspace(0,2,100); % Definition of frequency vector
bodemag(K,P.nominal,'--',w);
legend('K','P.nominal');
```



Frequency Response Evaluation

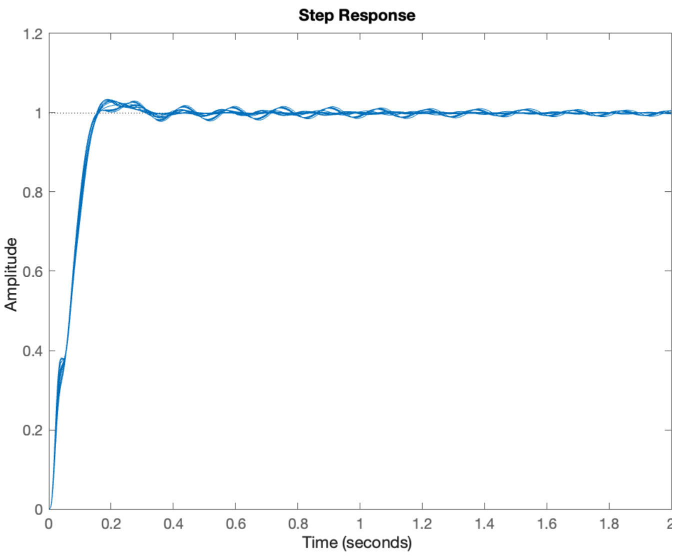
```
%% Evaluation of closed-loop characteristics
L = P*K;
T = feedback(L,1); % T = L/(1+L)
S = feedback(1,L); % S = 1/(1+L)
M = feedback(P,K); % M = P/(1+L)
figure(2);
bodemag(T.nominal,'-',1/Wt,':',S.nominal,'--',1/Ws,'-.',w);
legend('T.nomial','1/Wt','S.nominal','1/Ws');
ylim([-30 30]);
```

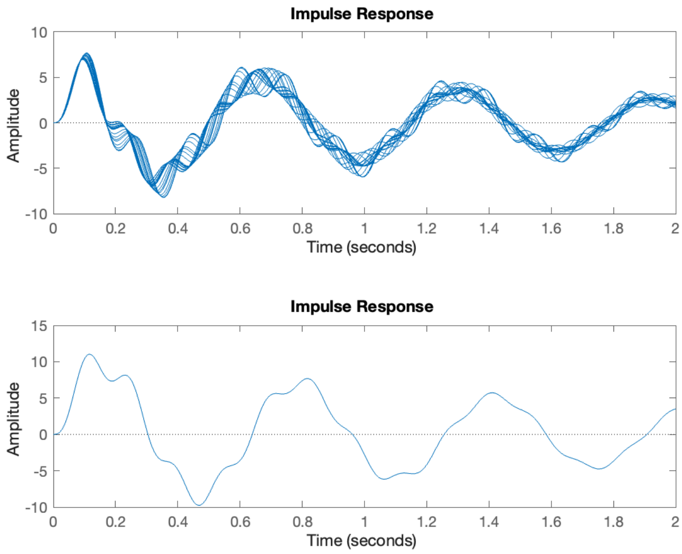
```
>> pole(T)

ans =
  1.0e+03 *
 -2.4272 + 0.0000i
 -0.0522 + 0.1324i
 -0.0522 - 0.1324i
 -0.1257 + 0.0545i
 -0.1257 - 0.0545i
 -0.0011 + 0.0436i
 -0.0011 - 0.0436i
 -0.0180 + 0.0180i
 -0.0180 - 0.0180i
 -0.0005 + 0.0099i
 -0.0005 - 0.0099i
```



Closed-Loop Time Response Evaluation

```
figure(3); step(T,2);
figure(4); subplot(211),impulse(M,2); subplot(212),impulse(Pn,2);
```





Project: Report 7

Consider your dynamic system :

Using uncertainty and performance weighing functions (W2 and W1) in the previous project steps, Design an H∞ controller.

Deadline: The day before next Meeting

Please only use this email address: bevranih18@gmail.com

Thank You!