8

Robust Control of a Mass-Damper-Spring System

In this chapter we consider the design of a robust control system for a simple, second-order, mechanical system, namely a *mass-damper-spring system.* The mass-damper-spring system is a common control experimental device frequently seen in an undergraduate teaching laboratory. As the first case study considered in this book, we show in detail the design of three different controllers for this system and present robust stability and robust performance analysis of the corresponding closed-loop systems, respectively. In order to keep the designs simple we take into account only the structured (parametric) perturbations in the plant coefficients. In this design example we give some information for several basic commands from the Robust Control Toolbox that are used in the analysis and design in this and subsequent case studies. To illuminate in a clear way the implementation of the most important Robust Control Toolbox commands, we include in the chapter all files used in computations related to the analysis and design of the mass-damper-spring control system. It is hoped that this chapter may serve as a tutorial introduction not only to robust control systems analysis and design but also to the use of the Robust Control Toolbox.

8.1 System Model

The *one-degree-of-freedom(1DOF)* mass-damper-spring system is depicted in Figure 8.1.

The dynamics of such a system can be described by the following 2nd-order differential equation, by Newton's Second Law,

$$m\ddot{x} + c\dot{x} + kx = u$$

where x is the displacement of the mass block from the equilibrium position and $u = F$ is the force acting on the mass, with m the mass, c the damper constant and k the spring constant. A block diagram of such a system is shown in Figure 8.2.

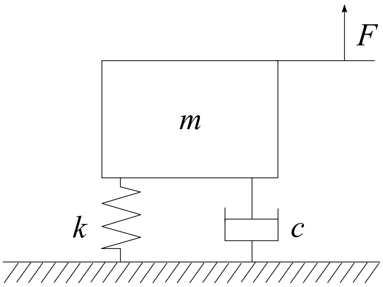


Fig. 8.1. Mass-damper-spring system

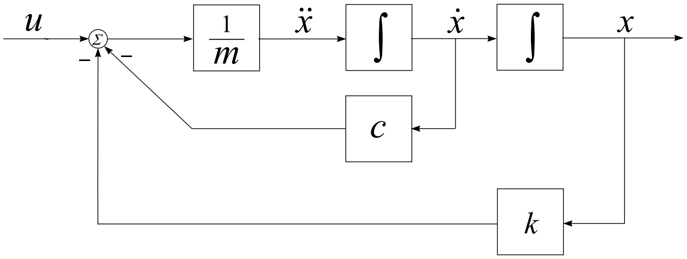


Fig. 8.2. Block diagram of the mass-damper-spring system

In a realistic system, the three physical parameters m, c and k are not known exactly. However, it can be assumed that their values are within certain, known intervals. That is,

$$m = \overline{m}(1 + p_m\delta_m),\ c = \overline{c}(1 + p_c\delta_c),\ k = \overline{k}(1 + p_k\delta_k)$$

where $\overline{m} = 3$, $\overline{c} = 1$, $\overline{k} = 2$ are the so-called nominal values of m, c and k. p_m, p_c and p_k and δ_m, δ_c and δ_k represent the possible (relative) perturbations on these three parameters. In the present study, we let $p_m = 0.4$, $p_c = 0.2$, $p_k = 0.3$ and $-1 \leq \delta_m,\ \delta_c,\ \delta_k \leq 1$. Note that this represents up to 40% uncertainty in the mass, 20% uncertainty in the damping coefficient and 30% uncertainty in the spring stiffness.

The three constant blocks in Figure 8.2 can be replaced by block diagrams in terms of $\overline{m}$, p_m, δ_m, *etc.*, in a unified approach. We note that the quantity $\frac{1}{m}$ may be represented as a linear fractional transformation (LFT) in δ_m

$$\begin{aligned}\frac{1}{m} &= \frac{1}{\overline{m}(1 + p_m\delta_m)} = \frac{1}{\overline{m}} - \frac{p_m}{\overline{m}}\delta_m(1 + p_m\delta_m)^{-1} \\ &= F_U(M_{mi}, \delta_m)\end{aligned}$$

with

$$M_{mi} = \begin{bmatrix} -p_m & \frac{1}{\overline{m}} \\ -p_m & \frac{1}{\overline{m}} \end{bmatrix}$$

Similarly, the parameter $c = \overline{c}(1 + p_c\delta_c)$ may be represented as an upper LFT in δ_c

$$c = F_U(M_c, \delta_c)$$

with

$$M_c = \begin{bmatrix} 0 & \overline{c} \\ p_c & \overline{c} \end{bmatrix}$$

and the parameter $k = \overline{k}(1 + p_k\delta_k)$ may be represented as an upper LFT in δ_k,

$$k = F_U(M_k, \delta_k)$$

with

$$M_k = \begin{bmatrix} 0 & \overline{k} \\ p_k & \overline{k} \end{bmatrix}$$

All these LFTs are depicted by block diagrams in Figure 8.3.

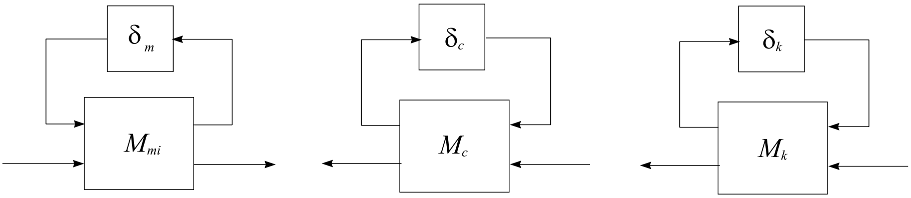


Fig. 8.3. Representation of uncertain parameters as LFTs

To further represent the system model as an LFT of the unknown, real perturbations δ_m, δ_c and δ_k, we use the block diagrams in Figure 8.3 and denote the inputs and outputs of δ_m, δ_c and δ_k as y_m, y_c, y_k and u_m, u_c, u_k, respectively, as shown in Figure 8.4.

With the above substitutions, the equations relating all "inputs"to corresponding "outputs"around these perturbed parameters can now be obtained as

$$\begin{bmatrix} y_m \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} -p_m & \frac{1}{\overline{m}} \\ -p_m & \frac{1}{\overline{m}} \end{bmatrix} \begin{bmatrix} u_m \\ u - v_c - v_k \end{bmatrix}$$

$$\begin{bmatrix} y_c \\ v_c \end{bmatrix} = \begin{bmatrix} 0 & \overline{c} \\ p_c & \overline{c} \end{bmatrix} \begin{bmatrix} u_c \\ \dot{x} \end{bmatrix}$$

$$\begin{bmatrix} y_k \\ v_k \end{bmatrix} = \begin{bmatrix} 0 & \overline{k} \\ p_k & \overline{k} \end{bmatrix} \begin{bmatrix} u_k \\ x \end{bmatrix}$$

$$u_m = \delta_m y_m$$
$$u_c = \delta_c y_c$$
$$u_k = \delta_k y_k$$

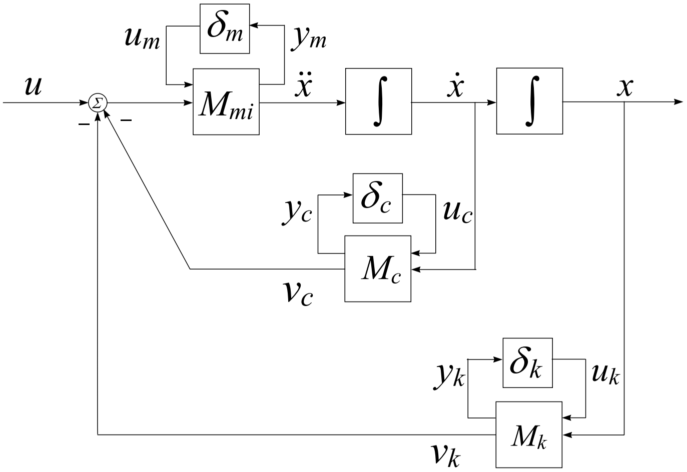


Fig. 8.4. Block diagram of the mass-damper-spring system with uncertain parameters

Let us set

$$x_1 = x,\ x_2 = \dot{x} = \dot{x}_1,\ y = x_1$$

such that

$$\dot{x}_2 = \ddot{x} = \ddot{x}_1$$

As a result, we obtain the following equations

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -p_m u_m + \tfrac{1}{\overline{m}}(u - v_c - v_k) \\
y_m &= -p_m u_m + \tfrac{1}{\overline{m}}(u - v_c - v_k) \\
y_c &= \overline{c} x_2 \\
y_k &= \overline{k} x_1 \\
v_c &= p_c u_c + \overline{c} x_2 \\
v_k &= p_k u_k + \overline{k} x_1 \\
y &= x_1 \\
u_m &= \delta_m y_m \\
u_c &= \delta_c y_c \\
u_k &= \delta_k y_k
\end{aligned}$$

By eliminating the variables v_c and v_k, the equations governing the system dynamic behaviour are given by

$$\left[\begin{array}{c}\dot{x}_1\\ \dot{x}_2\\ \hline y_m\\ y_c\\ y_k\\ \hline y\end{array}\right]=\left[\begin{array}{cc|ccc|c}0 & 1 & 0 & 0 & 0 & 0\\ -\frac{\overline{k}}{\overline{m}} & -\frac{\overline{c}}{\overline{m}} & -p_m & -\frac{p_c}{\overline{m}} & -\frac{p_k}{\overline{m}} & \frac{1}{\overline{m}}\\ \hline -\frac{\overline{k}}{\overline{m}} & -\frac{\overline{c}}{\overline{m}} & -p_m & -\frac{p_c}{\overline{m}} & -\frac{p_k}{\overline{m}} & \frac{1}{\overline{m}}\\ 0 & \overline{c} & 0 & 0 & 0 & 0\\ \overline{k} & 0 & 0 & 0 & 0 & 0\\ \hline 1 & 0 & 0 & 0 & 0 & 0\end{array}\right]\left[\begin{array}{c}x_1\\ x_2\\ \hline u_m\\ u_c\\ u_k\\ \hline u\end{array}\right]$$

$$\begin{bmatrix}u_m\\ u_c\\ u_k\end{bmatrix}=\begin{bmatrix}\delta_m & 0 & 0\\ 0 & \delta_c & 0\\ 0 & 0 & \delta_k\end{bmatrix}\begin{bmatrix}y_m\\ y_c\\ y_k\end{bmatrix}$$

Let $G_{\rm mds}$ denote the input/output dynamics of the mass-damper-spring system, which takes into account the uncertainty of parameters as shown in Figure 8.5. $G_{\rm mds}$ has four inputs $(u_m,\ u_c,\ u_k,\ u)$, four outputs $(y_m,\ y_c,\ y_k,\ y)$ and two states $(x_1,\ x_2)$.

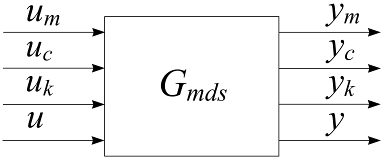


Fig. 8.5. Input/output block diagram of the mass-damper-spring system

The state space representation of $G_{\rm mds}$ is

$$G_{\rm mds}=\left[\begin{array}{c|cc}A & B_1 & B_2\\ \hline C_1 & D_{11} & D_{12}\\ C_2 & D_{21} & D_{22}\end{array}\right]$$

where

$$A=\begin{bmatrix}0 & 1\\ -\frac{\overline{k}}{\overline{m}} & -\frac{\overline{c}}{\overline{m}}\end{bmatrix},\ B_1=\begin{bmatrix}0 & 0 & 0\\ -p_m & -\frac{p_c}{\overline{m}} & -\frac{p_k}{\overline{m}}\end{bmatrix},\ B_2=\begin{bmatrix}0\\ \frac{1}{\overline{m}}\end{bmatrix}$$

$$C_1=\begin{bmatrix}-\frac{\overline{k}}{\overline{m}} & -\frac{\overline{c}}{\overline{m}}\\ 0 & \overline{c}\\ \overline{k} & 0\end{bmatrix},\ D_{11}=\begin{bmatrix}-p_m & -\frac{p_c}{\overline{m}} & -\frac{p_k}{\overline{m}}\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix},\ D_{12}=\begin{bmatrix}\frac{1}{\overline{m}}\\ 0\\ 0\end{bmatrix}$$

$$C_2=\begin{bmatrix}1\ 0\end{bmatrix},\ D_{21}=\begin{bmatrix}0\ 0\ 0\end{bmatrix},\ D_{22}=0$$

Note that $G_{\rm mds}$ depends only on $\overline{m},\ \overline{c},\ \overline{k},\ p_m,\ p_c,\ p_k$ and on the original differential equation connecting y with u. Hence, $G_{\rm mds}$ is known and contains no uncertain parameters.

Below, we give the M-file `mod_mds.m`, which can be used to compute the system matrix G_{mds} and to save it in the MATLAB® variable G.

File mod_mds.m

```
m = 3;
c = 1;
k = 2;
pm = 0.4;
pc = 0.2;
pk = 0.3;
%
A = [  0      1
     -k/m  -c/m];
B1 = [  0     0         0
      -pm  -pc/m   -pk/m];
B2 = [ 0
      1/m];
C1 = [-k/m  -c/m
       0     c
       k     0 ];
C2 = [ 1 0 ];
D11 = [-pm  -pc/m  -pk/m
        0     0      0
        0     0      0 ];
D12  = [1/m
         0
         0 ];
D21 = [0 0 0];
D22 = 0;
G  = pck(A,[B1,B2],[C1;C2],[D11 D12;D21 D22]);
```

The uncertain behaviour of the original system can be described by an upper LFT representation

$$y = F_U(G_{\mathrm{mds}}, \varDelta)u$$

with diagonal uncertainty matrix $\varDelta = \mathrm{diag}(\delta_m, \delta_c, \delta_k)$, as shown in Figure 8.6. It should be noted that the unknown matrix $\varDelta$, which will be called the *uncertainty matrix*, has a fixed *structure*. It is a diagonal matrix. It could, in general, be block diagonal. Such uncertainty is thus called *structured uncertainty.*

Apart from the method presented above in which system equations are derived first followed by explicitly defining all those coefficient matrices, the system matrix G_{mds} may also be obtained by using the MATLAB® command `sysic`. The following file `sys_mds.m` shows how this can be done.

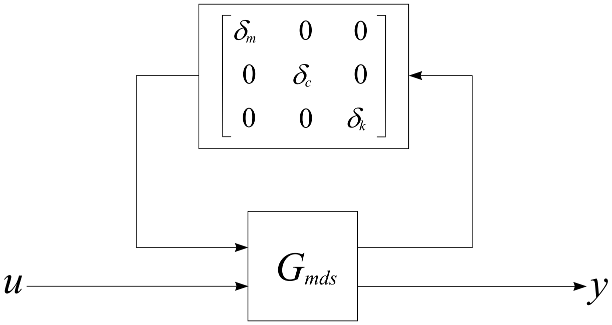

$\begin{bmatrix}\delta_m & 0 & 0\\ 0 & \delta_c & 0\\ 0 & 0 & \delta_k\end{bmatrix}$
G_{mds}
u
y

Fig. 8.6. LFT representation of the mass-damper-spring system with uncertainties

```
File sys_mds.m

  m_nom = 3; c_nom = 1; k_nom = 2;
  p_m = 0.4; p_c = 0.2; p_k = 0.3;
  mat_mi = [-p_m 1/m_nom; -p_m 1/m_nom];
  mat_c = [0 c_nom; p_c c_nom];
  mat_k = [0 k_nom; p_k k_nom];
  int1 = nd2sys([1],[1 0]);
  int2 = nd2sys([1],[1 0]);
  systemnames = 'mat_mi mat_c mat_k int1 int2';
  sysoutname = 'G';
  inputvar = '[um;uc;uk;u]';
  input_to_mat_mi = '[um;u-mat_c(2)-mat_k(2)]';
  input_to_mat_c = '[uc;int1]';
  input_to_mat_k = '[uk;int2]';
  input_to_int1 = '[mat_mi(2)]';
  input_to_int2 = '[int1]';
  outputvar = '[mat_mi(1);mat_c(1);mat_k(1);int2]';
  sysic;
```

8.2 Frequency Analysis of Uncertain System

The frequency responses of the perturbed open-loop system may be computed by using the command `starp` at a few different values of the perturbation parameters δ_m, δ_c, δ_k. In the M-file `pfr_mds` below, three values of each perturbation are chosen, the corresponding open-loop transfer function matrices generated and frequency responses calculated and plotted.

File pfr_mds.m

```
%
% Frequency responses of the perturbed plants
%
mod_mds
omega = logspace(-1,1,100);
[delta1,delta2,delta3] = ndgrid([-1 0 1],[-1 0 1], ...
                                [-1 0 1]);
for j = 1:27
    delta = diag([delta1(j),delta2(j),delta3(j)]);
    olp = starp(delta,G);
    olp_ic = sel(olp,1,1);
    olp_g = frsp(olp_ic,omega);
    figure(1)
    vplot('bode',olp_g,'c-')
    subplot(2,1,1)
    hold on
    subplot(2,1,2)
    hold on
end
subplot(2,1,1)
olp_ic = sel(G,4,4);
olp_g = frsp(olp_ic,omega);
vplot('bode',olp_g,'r--')
subplot(2,1,1)
title('BODE PLOTS OF PERTURBED PLANTS')
hold off
subplot(2,1,2)
hold off
```

The Bode plots of the family of perturbed systems for $-1 \leq \delta_m,\ \delta_c,\ \delta_k \leq 1$ are shown in Figure 8.7.

8.3 Design Requirements of Closed-loop System

The design objective for the mass-damper-spring system in this study is to find a linear, output feedback control $u(s) = K(s)y(s)$, which ensures the following properties of the closed-loop system.

Nominal stability and performance:

The controller designed should make the closed-loop system internally stable. Further, the required closed-loop system performance should be achieved for

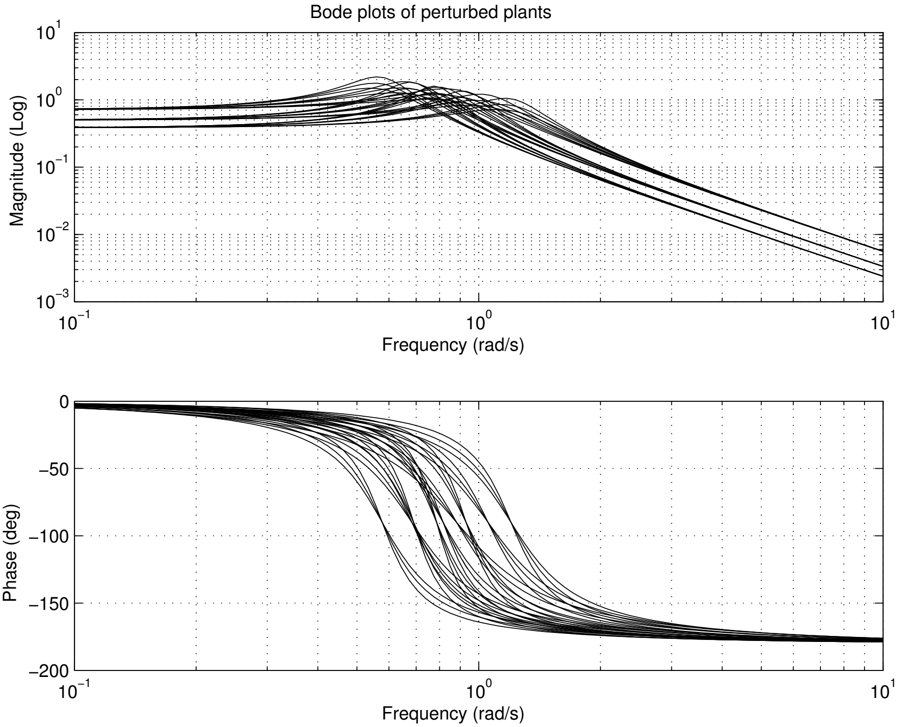


Fig. 8.7. Bode plots of perturbed open-loop systems

the nominal plant model G_{mds}. In this case study, the performance criterion for the closed-loop system is the so-called *S over KS* design and is described by

$$\left\| \begin{bmatrix} W_p S(G_{\mathrm{mds}}) \\ W_u K S(G_{\mathrm{mds}}) \end{bmatrix} \right\|_{\infty} < 1 \tag{8.1}$$

where $S(G_{\mathrm{mds}}) = (I + G_{\mathrm{mds}}K)^{-1}$ is the output sensitivity function of the nominal system, and W_p, W_u are weighting functions chosen to represent the frequency characteristics of some external (output) disturbance d and performance requirement (including consideration of control-effort constraint) level. Satisfaction of the above norm inequality indicates that the closed-loop system successfully reduces the effect of the disturbance to an acceptable level, and achieves the required performance. It should also be noted that the sensitivity function S denotes the transfer function of the reference tracking error.

Robust stability:

The closed-loop system achieves robust stability if the closed-loop system is internally stable for all possible plant models $G = F_U(G_{\mathrm{mds}}, \Delta)$. In the present case this means that the system must remain stable for any $1.8 \leq m \leq 4.2,\ 0.8 \leq c \leq 1.2,\ 1.4 \leq k \leq 2.6$.

Robust performance:

In addition to the robust stability, the closed-loop system, for all $G = F_U(G_{\mathrm{mds}}, \Delta)$, must satisfy the performance criterion

$$\left\| \begin{bmatrix} W_p(I+GK)^{-1} \\ W_uK(I+GK)^{-1} \end{bmatrix} \right\|_\infty < 1$$

Also, it is desirable that the complexity of the controller is acceptable, *i.e.*, it is of sufficiently low order.

The block diagram of the closed-loop system showing the feedback structure and including the elements reflecting model uncertainty and performance requirements, is given in Figure 8.8.

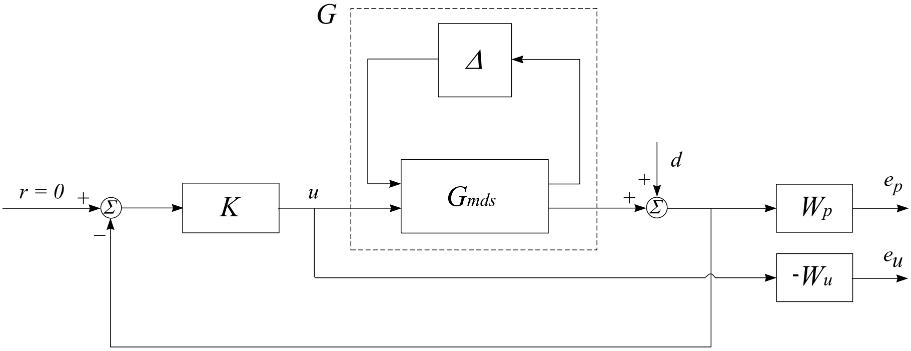


Fig. 8.8. Closed-loop system structure

The rectangle with dashed lines in Figure 8.8 represents the transfer function matrix G. Inside the rectangle is the nominal model G_{mds} of the mass-damper-spring system and the uncertainty matrix Δ that includes the model uncertainties. In general, the matrix Δ could be a transfer function matrix and is assumed to be stable. Δ is unknown but satisfies the norm condition $\|\Delta\|_\infty < 1$. The variable d is the disturbance on the system, at the system output. It is easy to work out that

$$\begin{bmatrix} e_p \\ e_u \end{bmatrix} = \begin{bmatrix} W_p(I+GK)^{-1} \\ W_uK(I+GK)^{-1} \end{bmatrix} d$$

Hence, the performance criterion is that the transfer functions from d to e_p and e_u should be small in the sense of $\|.\|_\infty$, for all possible uncertain transfer matrices Δ. The weighting functions W_p and W_u are used to reflect the relative significance of the performance requirement over different frequency ranges.

In the given case, the performance weighting function is a scalar function $W_p(s) = w_p(s)$ and is chosen as

$$w_p(s) = 0.95\frac{s^2 + 1.8s + 10}{s^2 + 8.0s + 0.01}$$

which ensures, apart from good disturbance attenuation, good transient response (settling time less than 10 s and overshoot less than 20% for the nominal system). The control weighting function W_u is chosen simply as the scalar $w_u = 10^{-2}$. Note that finding appropriate weighting functions is a crucial step in robust controller design and usually needs a few trials. For complex systems, significant efforts may be required.

To define the chosen weighting functions in MATLAB®, the following file `wts_mds.m` is used.

```
File wts_mds.m

  nuWp = [1  1.8  10];
  dnWp = [1  8    0.01];
  gainWp = 0.95;
  Wp = nd2sys(nuWp,dnWp,gainWp);
  nuWu = 1;
  dnWu = 1;
  gainWu = 10^(-2);
  Wu = nd2sys(nuWu,dnWu,gainWu);
```

To achieve the desired performance of disturbance rejection (or, of tracking error) it is necessary to satisfy the inequality $\|W_p(I + GK)^{-1}\|_\infty < 1$. Since W_p is a scalar function in the present case, the singular values of the sensitivity function $(I+GK)^{-1}$ over the frequency range must lie below that of $\frac{1}{w_p}$. This indicates $\|W_p(I + GK)^{-1}\|_\infty < 1$ if and only if for all frequencies $\sigma[(I + GK)^{-1}(j\omega)] < |1/w_p(j\omega)|$. The inverse weighting function $\frac{1}{w_p}$ is calculated by the commands

```
omega = logspace(-4,4,100);
Wp_g = frsp(Wp,omega);
Wpi_g = minv(Wp_g);
vplot('liv,lm',Wpi_g)
title('Inverse of Performance Weighting Function')
xlabel('Frequency (rad/sec)')
ylabel('Magnitude')
```

The singular values of $\frac{1}{w_p}$ over the frequency range $[10^{-4},\ 10^4]$ are shown in Figure 8.9.

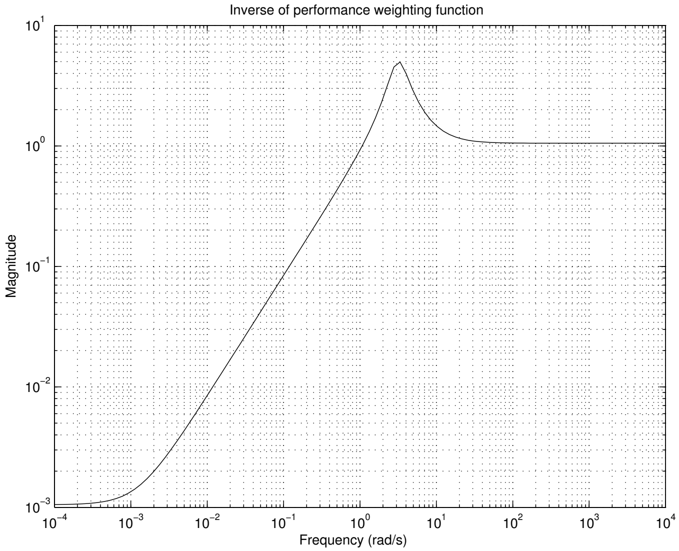


Fig. 8.9. Singular values of $\frac{1}{w_p}$

This weighting function shows that for low frequencies the closed-loop system (the nominal as well as perturbed) must attenuate the output disturbance in the ratio of 10 to 0.01. In other words, The effect of a unit disturbance on the steady-state output should be of the order 10^{-3} or less. (The same is also valid for the reference tracking error since in the given case the corresponding transfer function coincides with the sensitivity transfer function.) This performance requirement becomes less stringent with increasing frequency. It is seen in Figure 8.9 that from the frequency 1 rad/s the disturbance is no longer to be “attenuated”. This shows in the time response the effect of disturbance will not be alleviated until some time later and will then be reduced to a scale of 10^{-3} or even less.

8.4 System Interconnections

The structure of the open-loop system is represented in Figure 8.10. The variable `y_c` (the controller input) is taken with negative sign since the commands intended for design in the Robust Control Toolbox produce controllers with positive feedback.

The variables `pertin` and `pertout` have three elements, and the variables `control`, `dist`, `e_p`, `e_u` and `y_c` have one element.

The command `sysic` can be used to create the structure of open-loop systems in MATLAB®. In the present case study, the open-loop system is

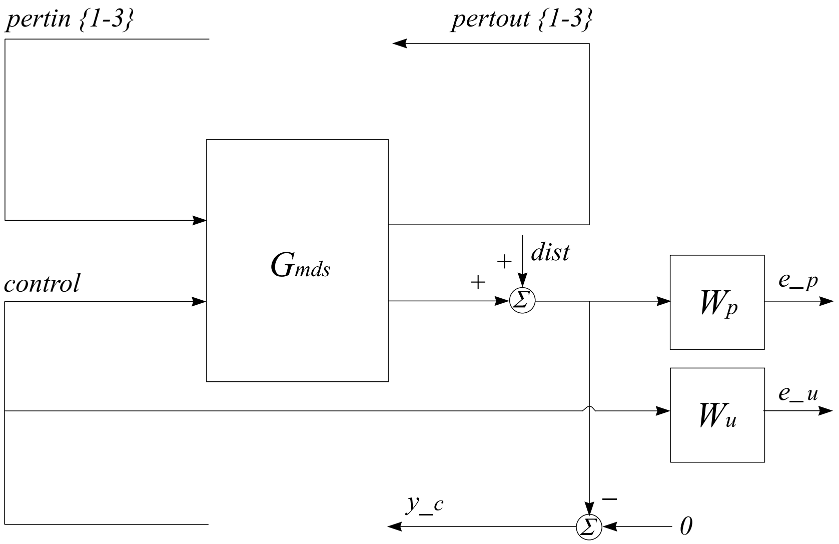


Fig. 8.10. Structure of open-loop system

saved as the variable `sys_ic`. `sys_ic` has five inputs and six outputs, as shown in Figure 8.11, and is a variable of the type SYSTEM (denoted also as P).

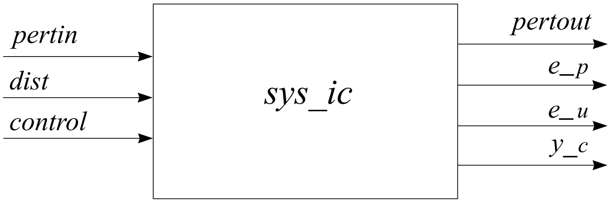


Fig. 8.11. Generalised block diagram of open-loop system

The following M-file `olp_mds.m` is used to create the variable `sys_ic`.

File `olp_mds.m`

```
systemnames = ' G Wp Wu ';
inputvar = '[ pert{3}; dist; control ]';
outputvar = '[ G(1:3); Wp; -Wu; -G(4)-dist ]';
input_to_G = '[ pert; control ]';
input_to_Wp = '[ G(4)+dist ]';
input_to_Wu = '[ control ]';
sysoutname = 'sys_ic';
```

```
cleanupsysic = 'yes';
sysic
```

To analyse the open-loop system, the following commands can be used.

```
minfo(sys_ic)
spoles(sys_ic)
spoles(W_p)
```

To assess the performance of designed systems, a unified simulation file of codes, `clp_mds.m`, is used and will be listed in Section 8.6. This simulation of closed-loop systems with designed controllers is based on the structure shown in Figure 8.12. Note that the weighting functions W_p and W_u are not included in the block-diagram for obvious reasons.

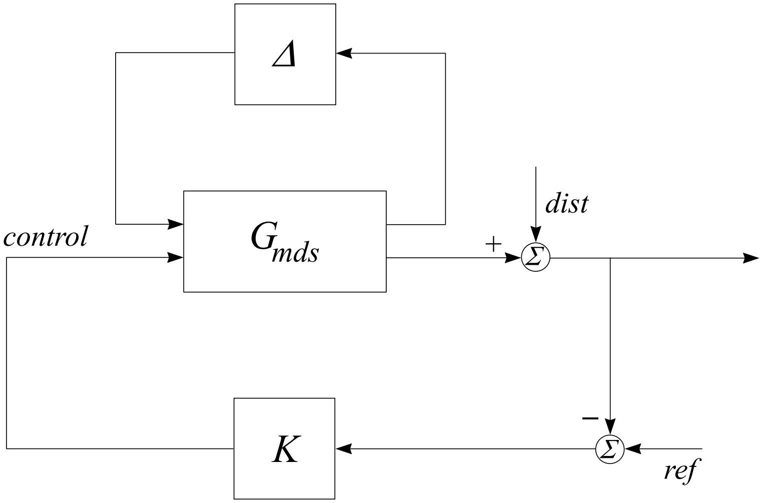


Fig. 8.12. Structure of the closed-loop system

The model of the open-loop system with uncertainties is set by the M-file `sim_mds.m` below.

File `sim_mds.m`

```
systemnames = ' G ';
inputvar = '[ pert{3}; ref; dist; control ]';
outputvar = '[ G(1:3); G(4)+dist; ref - G(4) - dist ]';
input_to_G = '[ pert; control ]';
sysoutname = 'sim_ic';
cleanupsysic = 'yes';
sysic
```

8.5 Suboptimal $\mathcal{H}_\infty$ Controller Design

The first controller to be designed for the connection of type SYSTEM, shown in Figure 8.11, is an $\mathcal{H}_\infty$ (sub)optimal controller. This controller minimizes the infinite-norm of $F_L(P, K)$ over all stabilising controllers K. Remember that $F_L(P, K)$ is the transfer function matrix of the nominal closed-loop system from the disturbance (the variable `dist`) to the errors (`e`), as shown in Figure 8.13, where $e = \begin{bmatrix} e_p \\ e_u \end{bmatrix}$. For this purpose, we first extract from `sys_ic` the corresponding transfer function matrix P using the following command and save it in the variable `hin_ic`.

```
hin_ic = sel(sys_ic,[4:6],[4:5])
```

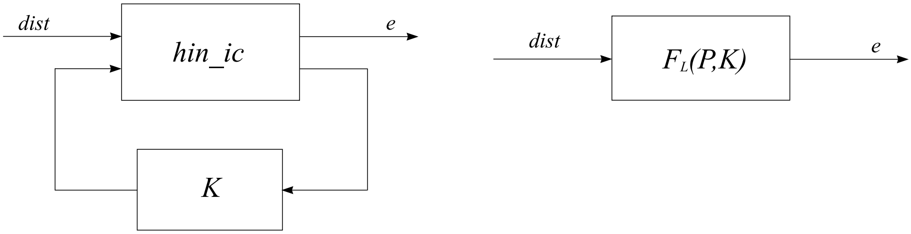


Fig. 8.13. Closed-loop LFTs in $\mathcal{H}_\infty$ design

The design uses the command `hinfsyn` that computes a suboptimal $\mathcal{H}_\infty$ controller, based on the given open-loop structure. The syntax, input and output arguments of `hinfsyn` are

```
[k,clp] = hinfsyn(p,nmeas,ncon,glow,ghigh,tol)
```

The arguments have the following meanings.

Input arguments

open-loop interconnection (matrix of type SYSTEM)	`p`
number of measurements	`nmeas`
number of controls	`ncons`
lower bound of bisection	`glow`
upper bound of bisection	`ghigh`
absolute tolerance for the bisection method	`tol`

Output arguments

controller (matrix of type SYSTEM)	`k`
closed-loop system (matrix of type SYSTEM)	`clp`

In the present exercise, the open-loop interconnection is saved in the variable `hin_ic`. It consists of one measurement (obtained by a sensor), two error signals, one control input, one disturbance and four states (two states of the plant plus two of the weighting function W_p). *Note that for the given structure, the open-loop system whose norm is to be minimised has 1 input and 2 outputs.* The interval for γ iteration is chosen between 0.1 and 10 with tolerance $tol = 0.001$. At each iteration the program displays the current value of γ and the results of five tests for existence of a suboptimal controller. At the end of each iteration the symbol `p` or `f` is displayed, which indicates the value of the current γ is either accepted or rejected. The symbol # is used to denote which of the five conditions for the existence of $\mathcal{H}_\infty$ (sub)optimal controllers is violated for the γ used. When the iteration procedure succeeds, the achievable minimum value of γ is given. The transfer function matrix of the closed-loop system from `dist` to `e` is saved in the variable `clp`. Below, we give the file `hin_mds.m` used to design an $\mathcal{H}_\infty$ (sub)optimal controller K_{hin}, followed by the display of results obtained in the exercise.

File `hin_mds.m`

```
nmeas = 1;
ncon = 1;
gmin = 1;
gmax = 10;
tol = 0.001;
hin_ic = sel(sys_ic,4:6,4:5);
[K_hin,clp] = hinfsyn(hin_ic,nmeas,ncon,gmin,gmax,tol);
```

```
Resetting value of Gamma min based on D_11, D_12, D_21 terms

Test bounds:      0.9500 <  gamma  <=     10.0000

  gamma  hamx_eig  xinf_eig  hamy_eig  yinf_eig   nrho_xy  p/f
 10.000  8.9e-001  6.2e-003  1.3e-003  0.0e+000   0.0000    p
  5.475  8.9e-001  6.2e-003  1.3e-003  0.0e+000   0.0000    p
  3.212  8.9e-001  6.3e-003  1.3e-003  -1.1e-020  0.0000    p
  2.081  8.8e-001  6.3e-003  1.3e-003  0.0e+000   0.0000    p
  1.516  8.8e-001  6.4e-003  1.3e-003  -2.2e-014  0.0000    p
  1.233  8.8e-001  6.5e-003  1.3e-003  -7.9e-021  0.0000    p
  1.091  8.8e-001  6.7e-003  1.3e-003  0.0e+000   0.0000    p
  1.021  8.8e-001  6.8e-003  1.3e-003  0.0e+000   0.0000    p
  0.985  8.8e-001  6.9e-003  1.3e-003  -7.9e-021  0.0000    p
```

```
   0.968  8.8e-001  7.0e-003  1.3e-003  0.0e+000   0.0000    p
   0.959  8.8e-001  7.1e-003  1.3e-003  -7.9e-021  0.0000    p
   0.954  8.8e-001  7.2e-003  1.3e-003  0.0e+000   0.0000    p
   0.952  8.8e-001  7.2e-003  1.3e-003  0.0e+000   0.0000    p
   0.951  8.8e-001  7.2e-003  1.3e-003  -7.9e-021  0.0000    p
   0.951  8.8e-001  7.2e-003  1.3e-003  0.0e+000   0.0000    p

  Gamma value achieved:     0.9506
```

The controller obtained is of 4th order. To check the achieved $\mathcal{H}_\infty$-norm of the closed-loop system that is found to be 0.95 in the iteration, the following command line can be used.

```
hinfnorm(clp)
```

8.6 Analysis of Closed-loop System with K_{hin}

From the last section, it is obvious that the required condition

$$\left\| \begin{bmatrix} W_p(I+G_{\mathrm{mds}}K)^{-1} \\ W_uK(I+G_{\mathrm{mds}}K)^{-1} \end{bmatrix} \right\|_\infty < 1$$

is satisfied and thus the closed-loop system achieves nominal performance requirements. Further, we use the following commands to form the transfer function of Figure 8.13, with the designed (sub)optimal $\mathcal{H}_\infty$ controller K_{hin}, to analyse the behaviour of the closed-loop system.

```
minfo(K_hin)
spoles(K_hin)
omega = logspace(-2,6,100);
clp_g = frsp(clp,omega);
vplot('liv,lm',vsvd(clp_g))
title('Singular Value Plot of clp')
xlabel('Frequency (rad/sec)')
ylabel('Magnitude')
```

Figure 8.14 shows the singular values of the closed-loop system `clp`. Since `clp` is of dimensions 2 × 1, there is just one (nonzero) singular value at each frequency.

Since the $\mathcal{H}_\infty$-norm of the closed-loop system is less than one, the condition $\|W_p(I+G_{\mathrm{mds}}K)^{-1}\|_\infty < 1$ is satisfied in the given case. This may be checked by computing the sensitivity function of the closed-loop system and comparing it with the inverse of the performance weighting function. The following file `sen_mds.m` can be used for this purpose.

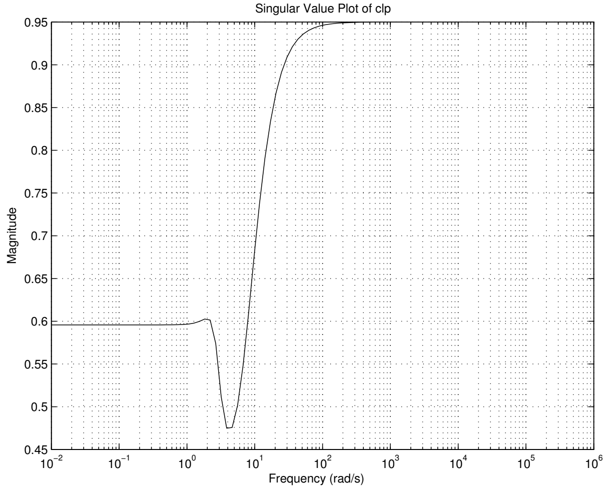


Fig. 8.14. Singular values of the closed-loop system with K_{hin}

File `sen_mds.m`

```
sim_mds
K = K_hin;
clp = starp(sim_ic,K);
%
% inverse performance weighting function
wts_mds
omega = logspace(-4,2,100);
Wp_g = frsp(Wp,omega);
Wpi_g = minv(Wp_g);
%
% sensitivity function
sen_loop = sel(clp,4,5);
sen_g = frsp(sen_loop,omega);
vplot('liv,lm',Wpi_g,'m--',vnorm(sen_g),'y-')
title('CLOSED-LOOP SENSITIVITY FUNCTION')
xlabel('Frequency (rad/sec)')
ylabel('Magnitude')
```

The result of the comparison is shown in Figure 8.15. It is seen that in the low-frequency range the sensitivity function lies below $\frac{1}{w_p}$.

The test for robust stability is conducted on the leading 3 × 3 diagonal block of the transfer function matrix `clp` and the test for nominal performance is tested on its bottom-right (the $(4-4)$ element), 2 × 1 transfer function. These transfer functions may be obtained by using the command `sel` from `clp`, and their frequency responses can be calculated afterwards. Or, it is

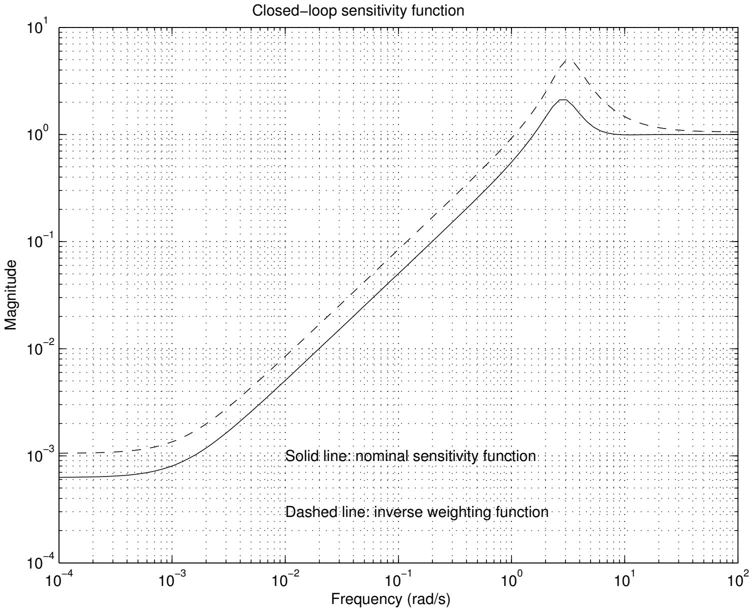


Fig. 8.15. Sensitivity function with $K_{\rm hin}$

possible to calculate the frequency response `clp_g` of the whole system `clp` first, and then use `sel` to obtain corresponding frequency responses. The closed-loop transfer function matrix `clp_ic` is obtainable from the open-loop interconnection `sys_ic` together with the controller `K` by implementing the function `starp`.

Since the uncertainty considered is *structured*, verification of the robust stability and robust performance needs the frequency response in terms of μ values. The syntax of the command `mu` is as the following.

```
[bnds,dvec,sens,pvec] = mu(matin,deltaset)
```

The function `mu` for μ-analysis computes upper and lower bounds for the structured singular value of the matrix `matin` with respect to the block structure `deltaset`. The matrix `matin` may be a matrix of the type CONSTANT or a matrix of the type VARYING, such as the frequency response of the closed-loop system. The command `mu` finds the upper and lower bounds for μ values in the 1×2 matrix `bnds` of the type VARYING, the frequency-dependent D-scaling matrices in `dvec`, the frequency-dependent perturbation, related to the lower bound, in `pvec` and the sensitivity of the upper bound to the D-scalings in `sens`. To achieve robust stability it is necessary that the upper bound of μ values is less than 1 over the frequency range.

In the present example, the frequency response in terms of μ values is denoted by `rob_stab` and the block structure is set by

```
blksR = [-1 1;-1 1;-1 1]
```

This means that in the robust stability analysis the parametric perturbations are assumed to be real. However, for better convergence of the algorithm that computes the lower bound of μ, we include 1% complex perturbations. This may, of course, produce some conservativeness in the design.

The file rob_mds.m below analyses the robust stability of the designed system, in which

```
K = K_hin;
```

has been defined in advance. The same file may be used for robust stability analysis of other controllers K.

File rob_mds.m

```
clp_ic = starp(sys_ic,K);
omega = logspace(-1,2,100);
clp_g = frsp(clp_ic,omega);
blkrsR = [-1 1;-1 1;-1 1];
rob_stab = sel(clp_g,[1:3],[1:3]);
pdim = ynum(rob_stab);
fixl = [eye(pdim); 0.1*eye(pdim)]; % 1% Complex
fixr = fixl';
blkrs = [blkrsR; abs(blkrsR)];
clp_mix = mmult(fixl,rob_stab,fixr);
[rbnds,rowd,sens,rowp,rowg] = mu(clp_mix,blkrs);
disp(' ')
disp(['mu-robust stability: ' ...
      num2str(pkvnorm(sel(rbnds,1,1)))])
disp(' ')
vplot('liv,lm',sel(rbnds,1,1),'y--',sel(rbnds,1,2),'m-', ...
      vnorm(rob_stab),'c-.')
title('ROBUST STABILITY')
xlabel('Frequency (rad/s)')
ylabel('mu')
```

The frequency responses of the upper and lower bounds of μ are shown in Figure 8.16. It is clear from the figure that the closed-loop system with K_{hin} achieves robust stability. The maximum value of μ is 0.764 that shows that structured perturbations with norm less than $\frac{1}{0.764}$ are allowable, *i.e.*, the stability maintains for $\|\Delta\|_\infty < \frac{1}{0.764}$. In the same figure we also plot the frequency response of the maximum singular value of the leading 3×3 transfer function matrix, which characterises the robust stability with respect to unstructured perturbations. It is seen that the latter is greater than 1 over the frequency interval [0.5, 1] rad/s roughly. Thus, the robust stability

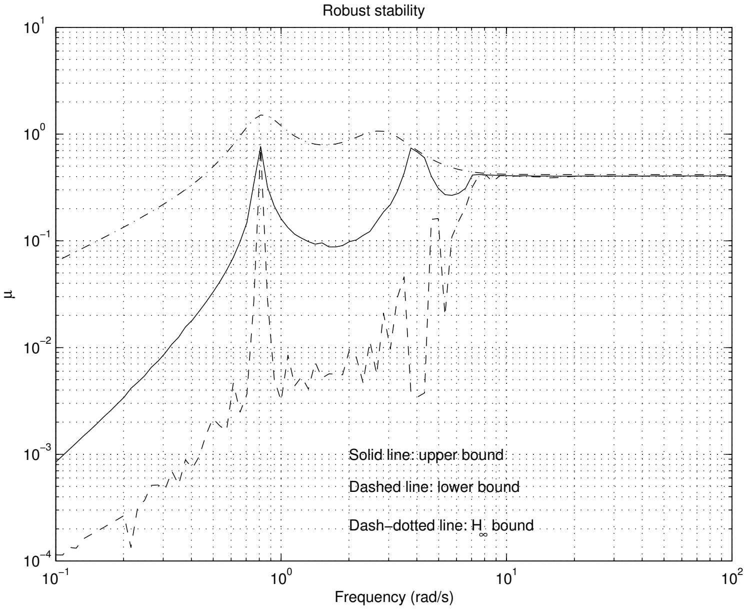


Fig. 8.16. Robust stability analysis of K_{hin}

is not preserved if the uncertainty is unstructured, which shows that the μ values give less conservative results if further information is known about the uncertainty.

The nominal performance of the closed-loop system (with respect to the weighting performance function) is analysed by means of the frequency response of the lower-right 2×1 transfer function block of `clp`. The nominal performance is achieved, if and only if for the frequency range considered the response magnitude is less than 1.

The robust performance of the closed-loop system with K_{hin} may be tested by means of the μ-analysis. The closed-loop transfer function `clp` have four inputs and five outputs. The first three inputs/outputs correspond to the three channels of perturbations Δ, while the 4th input/4th and 5th outputs pair corresponds to the weighted mixed sensitivity function. Hence, for μ-analysis of the robust performance the block structure must consist of a 3×3 uncertainty block and an 1×2 performance block as

$$\Delta_P := \left\{ \begin{bmatrix} \Delta & 0 \\ 0 & \Delta_F \end{bmatrix} : \Delta \in \mathcal{R}^{3\times 3},\ \Delta_F \in \mathcal{C}^{1\times 2} \right\}$$

The robust performance of the designed system is achieved, if and only if $\mu_{\Delta_P}(\cdot)$ is less than 1 for each frequency.

The nominal and robust performance of the closed-loop system with the controller K_{hin} is analysed using the file nrp_mds.m, which may be used for other controllers as well by defining, accordingly, the controller K.

File nrp_mds.m

```
clp_ic = starp(sys_ic,K);;
omega = logspace(-1,2,100);
clp_g = frsp(clp_ic,omega);
blkrsR = [-1 1;-1 1;-1 1];
%
% nominal performance
nom_perf = sel(clp_g,4,4);
%
% robust performance
rob_perf = clp_g;
blkrp = [blkrsR;[1 2]];
bndsrp = mu(rob_perf,blkrp);
vplot('liv,lm',vnorm(nom_perf),'y-',sel(bndsrp,1,1),'m--',...
      sel(bndsrp,1,2),'c--')
tmp1 = 'NOMINAL PERFORMANCE (solid) and';
tmp2 = ' ROBUST PERFORMANCE (dashed)';
title([tmp1 tmp2])
xlabel('Frequency (rad/s)')
disp(' ')
disp(['mu-robust performance: ' ...
     num2str(pkvnorm(sel(bndsrp,1,1)))])
disp(' ')
```

The frequency responses showing the nominal and robust performance are given in Figure 8.17. It is seen from this figure that the system with K_{hin} achieves nominal performance but fails to satisfy the robust performance criterion. This conclusion follows from the fact that the frequency response of the nominal performance has a maximum of 0.95, while the μ curve for the robust performance has a maximum of 1.67.

With respect to the robust performance, this means in the present case that the size of the perturbation matrix Δ must be limited to $\|\Delta\|_\infty \leq \frac{1}{1.67}$, to ensure the (perturbed) performance function satisfying

$$\left\| \begin{bmatrix} W_p(I + F_U(G_{\text{mds}}, \Delta_G)K)^{-1} \\ W_u K(I + F_U(G_{\text{mds}}, \Delta_G)K)^{-1} \end{bmatrix} \right\|_\infty \leq 1$$

The closed-loop simulation with the file clp_mds.m includes determination of the transient responses in respect to the reference and disturbance, in which the command trsp is used.

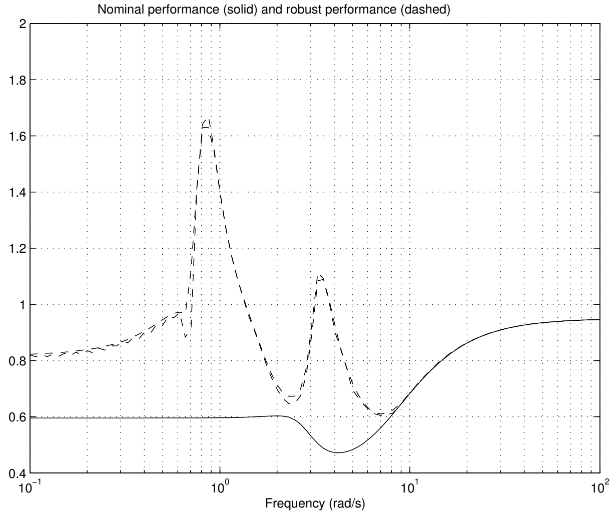


Fig. 8.17. Nominal and robust performance of K_{hin}

File `clp_mds.m`

```
% response to the reference
sim_mds
clp = starp(sim_ic,K);
timedata = [0 20 40];
stepdata = [1 0 1];
dist = 0;
ref = step_tr(timedata,stepdata,0.1,60);
u = abv(0,0,0,ref,dist);
y = trsp(clp,u,60,0.1);
figure(1)
vplot(sel(y,4,1),'y-',ref,'r--')
title('CLOSED-LOOP TRANSIENT RESPONSE')
xlabel('Time (secs)')
ylabel('y (m)')
%
% response to the disturbance
timedata = [0 20 40];
stepdata = [1 0 1];
dist = step_tr(timedata,stepdata,0.1,60);
ref = 0;
u = abv(0,0,0,ref,dist);
y = trsp(clp,u,60,0.1);
figure(2)
```

```
vplot(sel(y,4,1),'y-',dist,'r--')
title('TRANSIENT RESPONSE TO THE DISTURBANCE')
xlabel('Time (secs)')
ylabel('y (m)')
```

The transient responses to the reference input and to the disturbance input are shown in Figures 8.18 and 8.19, respectively. The transient responses are relatively slow and have slight overshoots.

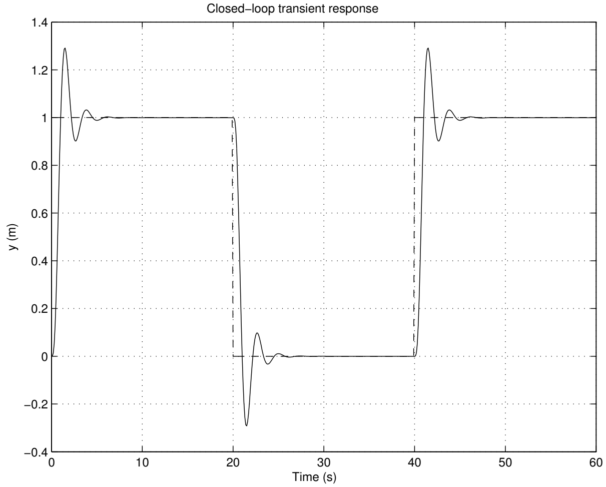


Fig. 8.18. Transient response to reference input ($K_{\rm hin}$)

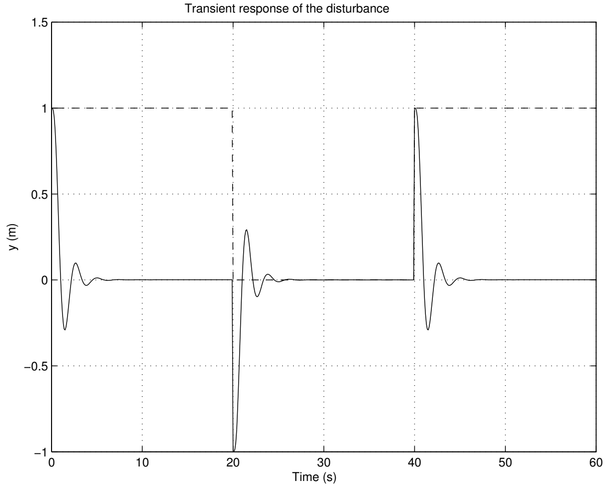


Fig. 8.19. Transient response to disturbance input ($K_{\rm hin}$)

8.7 $\mathcal{H}_\infty$ Loop-shaping Design

Let us consider now using the $\mathcal{H}_\infty$ loop-shaping design procedure (LSDP) for the mass-damper-spring system. For this aim the pre- and postcompensators (see Chapter 5) are taken as

$$W_1(s) = 2\frac{8s+1}{0.9}, \quad W_2(s) = 1$$

The precompensator W_1 is chosen so as to introduce an integrating effect in the low-frequency range that leads to good attenuation of disturbances.

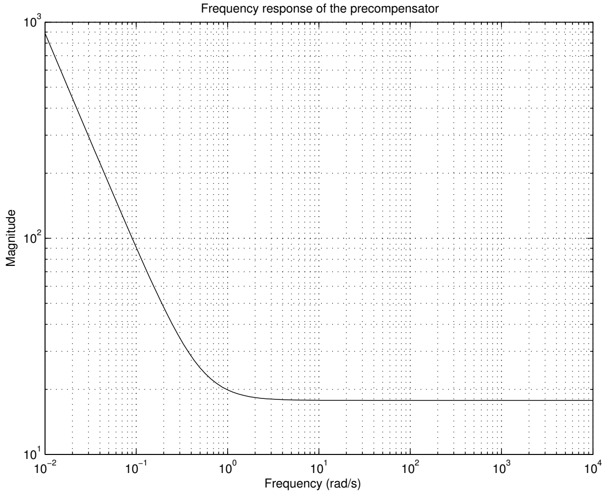


Fig. 8.20. Frequency response of the precompensator

The frequency response of the precompensator W_1 is shown in Figure 8.20, and the frequency responses of the plant and of the shaped plant are given in Figure 8.21.

The $\mathcal{H}_\infty$ LSDP controller is computed by calling the function `ncfsyn` from the Robust Control Toolbox. The syntax, input and output arguments of `ncfsyn` are

```
[sysk,emax,sysobs] = ncfsyn(sysgw,factor,opt)
```

The arguments have the following meanings:

Input arguments

`sysgw` the shaped plant
(matrix of type SYSTEM);

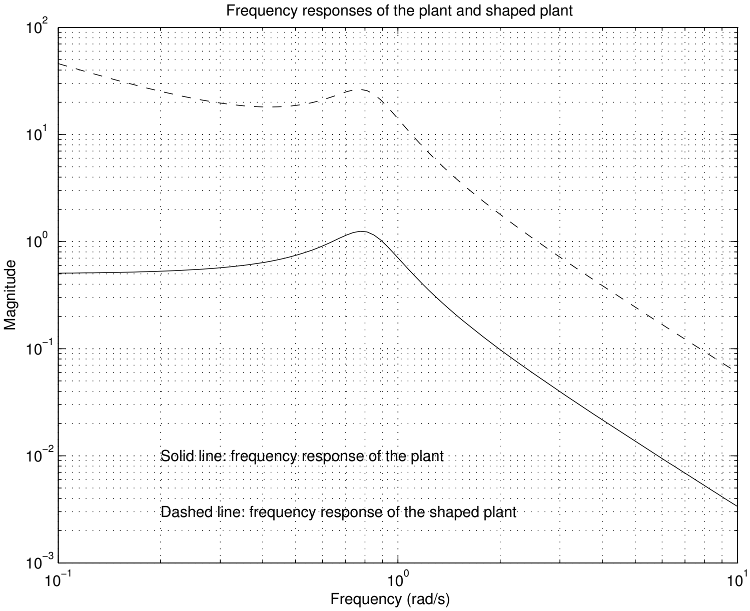


Fig. 8.21. Frequency responses of the plant and shaped plant

`factor` = 1 implies that optimal controller is required.
> 1 implies that suboptimal controller is required achieving a performance that is `FACTOR` times less than the optimal one;

`opt` `'ref'` the controller includes an additional set of reference signals and should be implemented as a two-degree-of-freedom controller (optional).

Output arguments

`sysk` $\mathcal{H}_\infty$ loop-shaping controller;

`emax` Stability margin that shows the robustness to unstructured perturbations. `emax` is always less than 1 and values of `emax` larger than 0.3 generally indicate good robustness margins;

`sysobs` $\mathcal{H}_\infty$ controller with state observer. This variable is created only if `factor>1` and `opt = 'ref'`.

The loop-shaping design of the mass-damper-spring system is executed by the file `lsh_mds.m` below. The parameter `factor` is chosen as 1.1, which means that the resulted suboptimal controller will be close to the optimal one. The designed controller is named K_{lsh} in the program.

File `lsh_mds.m`

```
%
% set the precompensator
nuw1 = [2 1];
dnw1 = [0.9 0];
gainw1 = 8;
w1 = nd2sys(nuw1,dnw1,gainw1);
%
% frequency response of w1
omega = logspace(-2,4,100);
w1g = frsp(w1,omega);
figure(1)
vplot('liv,lm',w1g,'r-')
title('Frequency response of the precompensator')
xlabel('Frequency (rad/sec)')
ylabel('Magnitude')
%
% form the shaped plant
G_l = sel(G,4,4);
sysGW = mmult(G_l,w1);
%
% frequency responses of the plant and shaped plant
omega = logspace(-1,1,100);
G_lg = frsp(G_l,omega);
sysGWg = frsp(sysGW,omega);
figure(2)
vplot('liv,lm',G_lg,'c-',sysGWg,'m--')
title('Frequency responses of the plant and shaped plant')
xlabel('Frequency (rad/sec)')
ylabel('Magnitude')
%
% compute the suboptimal positive feedback controller
[sysK,emax] = ncfsyn(sysGW,1.1);
disp(['Stability margin emax = ' num2str(emax)]);
K = mmult(w1,sysK);
%
% construct the negative feedback controller
[ak,bk,ck,dk] = unpck(K);
```

```
ck = -ck;
dk = -dk;
K_lsh = pck(ak,bk,ck,dk);
```

8.8 Assessment of $\mathcal{H}_\infty$ Loop-shaping Design

The $\mathcal{H}_\infty$ LSDP controller K_{lsh} ensures a stability margin `emax = 0.395`, which is a good indication with respect to the robust stability. As in the case of the previous $\mathcal{H}_\infty$ design, the order of the resulting controller is equal to 4.

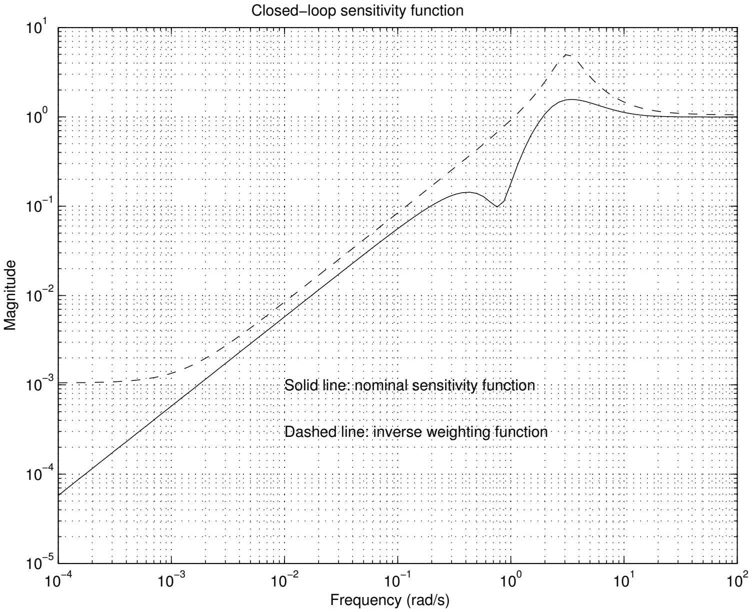


Fig. 8.22. Sensitivity function of K_{lsh}

The sensitivity function of the closed-loop system of K_{lsh} is shown in Figure 8.22. It is clear that the requirement for disturbance attenuation is satisfied.

To check the robust stability of the designed closed-loop system with K_{lsh}, the upper- and low-bounds of μ values of the corresponding transfer function matrix (the leading 3×3 diagonal block of `clp` with K_{lsh}) are shown in Figure 8.23. It is clear from that figure that the closed-loop system achieves robust stability for the parametric perturbations under consideration.

The nominal and robust performance of the closed-loop system are assessed in Figure 8.24. It is seen in the figure that with the designed $\mathcal{H}_\infty$

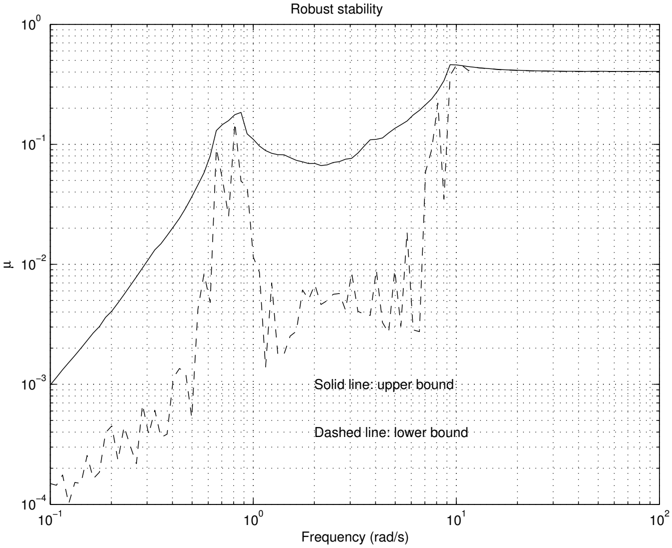


Fig. 8.23. Robust stability of closed-loop system with K_{lsh}

LSDP controller the closed-loop system achieves both nominal and robust performance.

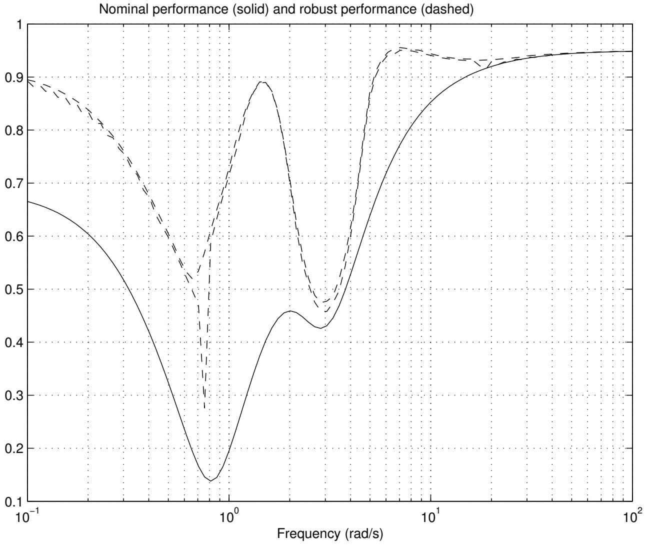


Fig. 8.24. Nominal and robust performance of K_{lsh}

Further, the transient responses of the closed-loop system are obtained, once more using the file `clp_mds.m`, and are shown in Figures 8.25 and 8.26. The time responses are slower than in the case of the $\mathcal{H}_\infty$ controller but with smaller overshoots (less than 25%).

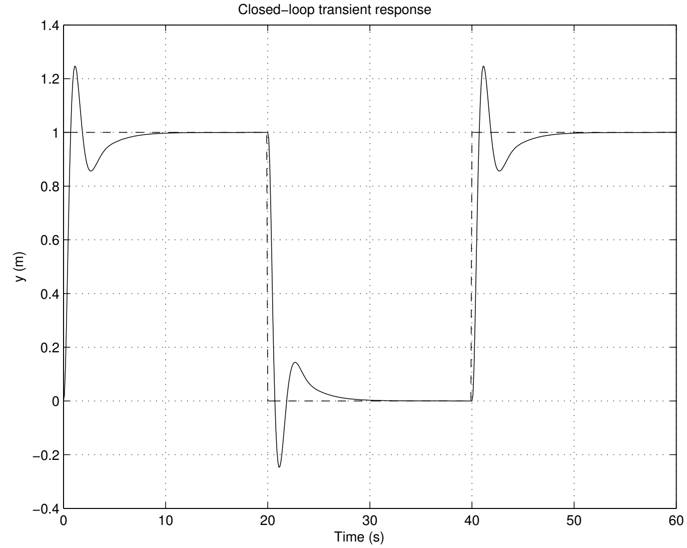


Fig. 8.25. Transient response to reference input (K_{lsh})

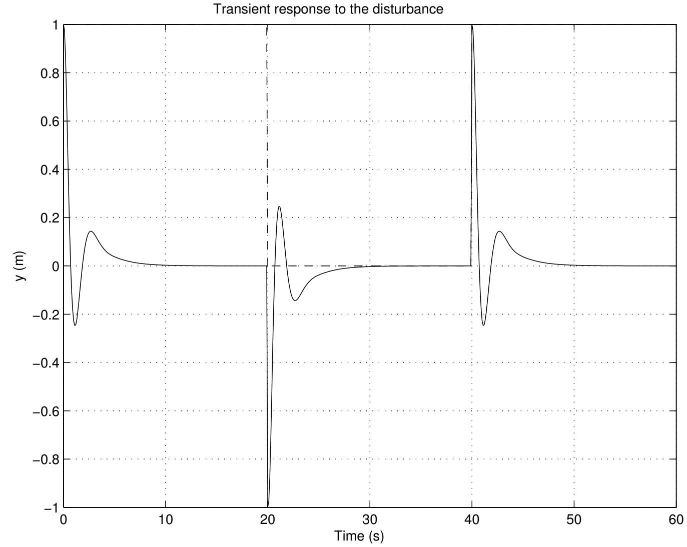


Fig. 8.26. Transient response to disturbance input (K_{lsh})

8.9 μ-Synthesis and D-K Iterations

The block diagram of the closed-loop system used in the μ-synthesis is given in Figure 8.8 and is reproduced for convenience in Figure 8.27.

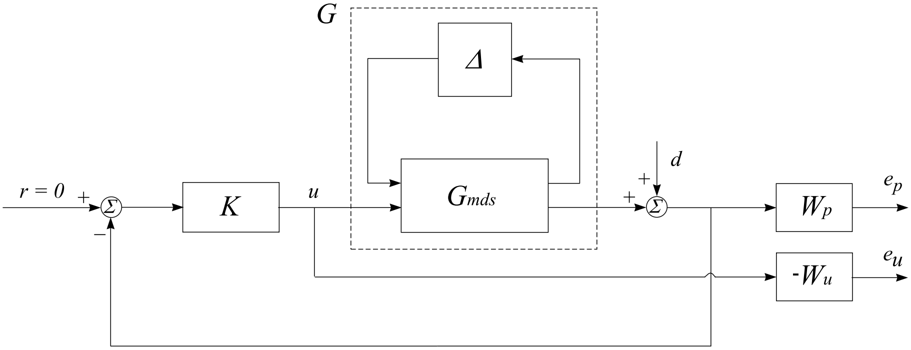


Fig. 8.27. Structure of closed-loop system

Let $P(s)$ denote the transfer function matrix of the five inputs, six outputs open-loop system `sys_ic` in Figure 8.11 and let the block structure Δ_P be defined as in the case of the robust performance analysis in previous sections as

$$\Delta_P := \left\{ \begin{bmatrix} \Delta & 0 \\ 0 & \Delta_F \end{bmatrix} : \Delta \in \mathcal{R}^{3\times 3},\ \Delta_F \in \mathcal{C}^{1\times 2} \right\}$$

The first uncertainty block Δ of this structured matrix is diagonal and corresponds to the uncertainties used in the modelling of the mass-damper-spring system. The second block Δ_F is a fictitious uncertainty block that is introduced to represent the performance requirements in the framework of the μ-approach.

The following optimisation problem is formed to minimize the upper bound of μ values which in turn reduces the maximum value of μ.

$$\begin{array}{ccc} \min & \min & \left\| D_\ell(s) F_L(P,K) D_r^{-1}(s) \right\|_\infty \\ K & D_\ell(s), D_r(s) & \\ \text{stabilizing} & \text{stable,} & \\ & \text{min. phaze} & \end{array}$$

$$D_\ell(s) = \begin{bmatrix} d_1(s) & 0 & 0 & 0 \\ 0 & d_2(s) & 0 & 0 \\ 0 & 0 & d_3(s) & 0 \\ 0 & 0 & 0 & d_4(s) I_2 \end{bmatrix}$$

$$D_r(s) = \begin{bmatrix} d_1(s) & 0 & 0 & 0 \\ 0 & d_2(s) & 0 & 0 \\ 0 & 0 & d_3(s) & 0 \\ 0 & 0 & 0 & d_4(s) \end{bmatrix}$$

where $d_1(s)$, $d_2(s)$, $d_3(s)$, $d_4(s)$ are scaling transfer functions. The finding of a minimum value of the cost function and construction of a controller K, which would achieve performance arbitrary close to the optimal level, is called the μ-synthesis.

In other words, the aim of μ-synthesis is to find a stabilising controller K, such that for each frequency $\omega \in [0, \infty]$ the structured singular value is to satisfy the condition

$$\mu_{\Delta_P}[F_L(P, K)(j\omega)] < 1$$

Fulfilment of the above condition guarantees robust performance of the closed-loop system, *i.e.*,

$$\left\| \begin{bmatrix} W_p(I + F_U(G_{\mathrm{mds}}, \Delta)K)^{-1} \\ W_u K(I + F_U(G_{\mathrm{mds}}, \Delta)K)^{-1} \end{bmatrix} \right\|_\infty < 1$$

The μ-synthesis is executed by the M-file `dkit` from the Robust Control Toolbox, which automates the procedure by using D-K iterations. To implement the function `dkit` it is necessary to prepare a file such as the following `dk_mds.m`, in which the necessary variables are assigned. The file below can be easily modified for the μ-synthesis of other systems.

File `dk_mds.m`

```
% dk_mds
%
%  This script file contains the USER DEFINED VARIABLES
%  for the mutools DKIT script file. The user MUST define
%  the 5 variables below.
%----------------------------------------------%
%       REQUIRED USER DEFINED VARIABLES        %
%----------------------------------------------%
% Nominal plant interconnection structure
NOMINAL_DK = sys_ic;

% Number of measurements
NMEAS_DK = 1;

% Number of control inputs
NCONT_DK = 1;

% Block structure for mu calculation
```

```
BLK_DK = [-1 1;-1 1;-1 1;1 2];

% Frequency response range
OMEGA_DK = logspace(-2,4,100);

AUTOINFO_DK = [1 4 1 4*ones(1,size(BLK_DK,1))];

NAME_DK = 'mds';
%------------------ end of dk_mds -------------------- %
```

The variables, set in the above file, have the following meanings.

NOMINAL_DK A variable containing the nominal open-loop system

NMEAS_DK Number of measurements (number of inputs of K)

NCONT_DK Number of control inputs (number of outputs of K)

BLK_DK Block structure for computation of μ (involves the uncertainty blocks as well as the performance block)

OMEGA_DK Frequency response range

AUTINFO_DK Variable, which is used for full automation of the D-K-iteration.
It has the following components:

AUTOINFO_DK(1) Initial iteration

AUTOINFO_DK(2) Final iteration

AUTOINFO_DK(3) Visualisation flag (1 - the results appear on the display, 2 - the results are not displayed)
The rest elements in AUTOINFO_DK (their number is equal to the number of blocks in BLK_DK) set the maximum dynamic order of the transfer functions during the D-scaling

NAME_DK Suffix to the names of the saved variables

After preparing the file dk_mds, it is necessary to assign the name of the string variable DK_DEF_NAME, in the MATLAB® workspace, to DK_MDS. Now the command dkit can be called, with which begins the D-K-iteration procedure. The whole design is completed by the file ms_mds.m below.

File ms_mds.m

```
DK_DEF_NAME = 'dk_mds';
dkit
```

```
  K_mu = k_dk4mds;
```

Note that the μ-controller named as K_μ is obtained after four iterations in this exercise.

The results from the first iteration are listed below, in which the numbers have been shortened for display.

```
Iteration Number:  1
--------------------


Resetting value of Gamma min based on D_11, D_12, D_21 terms

Test bounds:      0.9500 <  gamma  <=    100.0000

gamma    hamx_eig  xinf_eig  hamy_eig  yinf_eig  nrho_xy p/f
100.000  6.4e-001  2.2e-001  1.3e-003  -1.6e-013  0.0002   p
 50.475  6.3e-001  2.2e-001  1.3e-003  0.0e+000   0.0010   p
 25.713  6.3e-001  2.2e-001  1.3e-003  -2.1e-014  0.0038   p
 13.331  6.3e-001  2.2e-001  1.3e-003  -1.9e-013  0.0143   p
  7.141  6.1e-001  2.2e-001  1.3e-003  0.0e+000   0.0522   p
  4.045  5.5e-001  2.3e-001  1.3e-003  -7.3e-018  0.1924   p
  2.498  3.8e-001  2.4e-001  1.3e-003  2.9e-019   1.6946#  f
  3.097  4.8e-001  2.3e-001  1.3e-003  0.0e+000   0.4322   p

 Gamma value achieved:     3.0966
Calculating MU of closed-loop system:
points completed....
1.2.3.4.5.6.7.8.9.10.11.12.13.14.15.16.17.
18.19.20.21.22.23.24.25.26.27.28.29.30.31.32.33.34.35.
36.37.38.39.40.41.42.43.44.45.46.47.48.49.50.51.52.53.
54.55.56.57.58.59.60.61.62.63.64.65.66.67.68.69.70.71.
72.73.74.75.76.77.78.79.80.81.82.83.84.85.86.87.88.89.
90.91.92.93.94.95.96.97.98.99.100.
points completed....
1.2.3.4.5.6.7.8.9.10.11.12.13.14.15.16.17.
18.19.20.21.22.23.24.25.26.27.28.29.30.31.32.33.34.35.
36.37.38.39.40.41.42.43.44.45.46.47.48.49.50.51.52.53.
54.55.56.57.58.59.60.61.62.63.64.65.66.67.68.69.70.71.
72.73.74.75.76.77.78.79.80.81.82.83.84.85.86.87.88.89.
90.91.92.93.94.95.96.97.98.99.100.
Iteration Summary
-----------------------------
Iteration #                 1
Controller Order            4
Total D-Scale Order         0
Gamma Acheived          3.097
```

```
Peak mu-Value          2.340

Auto Fit in Progress
     Block 1, MaxOrder=4, Order = 0 1 2 3 4
     Block 2, MaxOrder=4, Order = 0 1 2 3
     Block 3, MaxOrder=4, Order = 0 1 2
     Block 4, MaxOrder=4, Order = 0
```

The results obtained may be interpreted in the following way.

First, the design of a fourth-order $\mathcal{H}_\infty$ controller is finished (initially the open-loop system is of fourth order), with the scaling matrix being set equal to the unit matrix. With this controller one achieves a value of γ equal to 3.097 and maximum value of μ equal to 2.34.

Then, an approximation (curve fitting) of the diagonal elements of the scaling matrix, obtained in the computation of μ, takes place. These elements are functions of the frequency and are approximated by stable minimum phase transfer functions whose orders do not exceed 4. It is seen from the results that the first diagonal element is approximated by a fourth-order transfer function, the second one by a third-order transfer function, the third one by a second-order transfer function, and the fourth one by a scalar. These transfer functions are "absorbed" into the transfer function P, which is to be used in the $\mathcal{H}_\infty$ design at the second iteration. Altogether, the order of the "absorbed" transfer functions is $9 + 9 = 18$. This increase of the open-loop system order leads to the increase of the controller order at the second iteration to 22.

The structured singular value μ of the closed-loop system with the $\mathcal{H}_\infty$ controller from the first step and the $\mathcal{H}_\infty$-norm of the scaled system are shown in Figure 8.28. The latter is an upper bound of the former.

The second iteration gives:

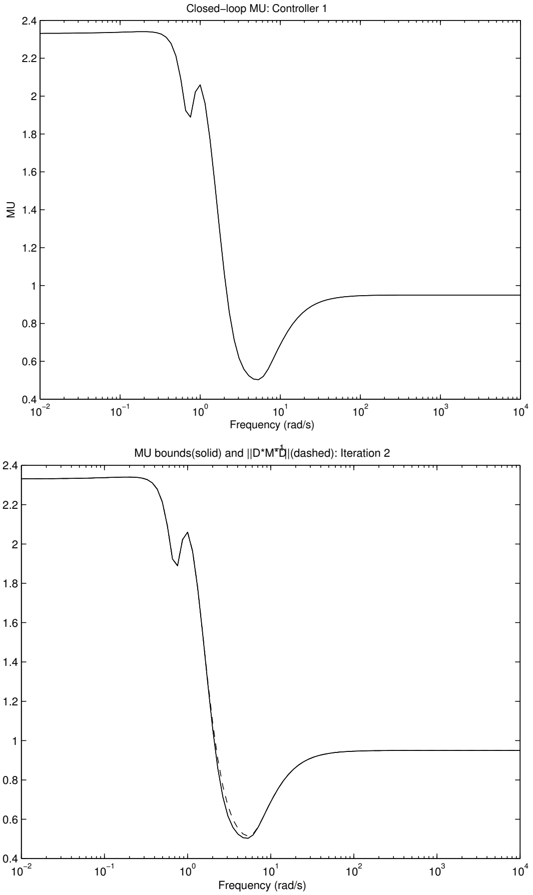
```
Iteration Number:  2
--------------------


Resetting value of Gamma min based on D_11, D_12, D_21 terms

Test bounds:      0.9500 <  gamma  <=      2.3873

 gamma  hamx_eig  xinf_eig  hamy_eig   yinf_eig   nrho_xy  p/f
 2.387  1.1e-002  2.6e-008  1.3e-003   -1.2e-017  0.0671    p
 1.669  1.1e-002  2.6e-008  1.3e-003   -1.1e-014  0.1720    p
 1.309  1.1e-002  2.6e-008  1.3e-003   -1.7e-017  1.5886#   f
 1.433  1.1e-002  2.6e-008  1.3e-003   1.4e-017   0.3344    p
 1.358  1.1e-002  2.6e-008  1.3e-003   -2.9e-020  0.5738    p
 1.333  1.1e-002  2.6e-008  1.3e-003   -1.9e-020  0.8201    p

 Gamma value achieved:     1.3334
Calculating MU of closed-loop system:
```



Closed−loop MU: Controller 1

MU bounds(solid) and ||D*M*D^{-1}||(dashed): Iteration 2

Fig. 8.28. μ values and D–scaling at first iteration

```
points completed....
1.2.3.4.5.6.7.8.9.10.11.12.13.14.15.16.17.
18.19.20.21.22.23.24.25.26.27.28.29.30.31.32.33.34.35.
36.37.38.39.40.41.42.43.44.45.46.47.48.49.50.51.52.53.
54.55.56.57.58.59.60.61.62.63.64.65.66.67.68.69.70.71.
72.73.74.75.76.77.78.79.80.81.82.83.84.85.86.87.88.89.
90.91.92.93.94.95.96.97.98.99.100.
points completed....
1.2.3.4.5.6.7.8.9.10.11.12.13.14.15.16.17.
18.19.20.21.22.23.24.25.26.27.28.29.30.31.32.33.34.35.
36.37.38.39.40.41.42.43.44.45.46.47.48.49.50.51.52.53.
54.55.56.57.58.59.60.61.62.63.64.65.66.67.68.69.70.71.
72.73.74.75.76.77.78.79.80.81.82.83.84.85.86.87.88.89.
```

```
90.91.92.93.94.95.96.97.98.99.100.
Iteration Summary
---------------------------------------
Iteration #                 1         2
Controller Order            4        22
Total D-Scale Order         0        18
Gamma Acheived          3.097     1.333
Peak mu-Value           2.340     1.274

Auto Fit in Progress
     Block 1, MaxOrder=4, Order = 0 1 2 3 4
     Block 2, MaxOrder=4, Order = 0 1 2 3
     Block 3, MaxOrder=4, Order = 0 1 2 3
     Block 4, MaxOrder=4, Order = 0
```

After this iteration, the value of γ decreases to 1.333, and the value of μ reduces to 1.274. This is achieved, however, by a 22nd-order controller. The corresponding closed-loop system obtained is of 26th order. The diagonal elements of the scaling matrix are approximated by transfer functions of orders 4, 3, 3 and 0, respectively, which increases the order of the open-loop system by 20. Thus, the open-loop system is now of 24th order. Note that the increase of the open-loop system order is significant for the controller order only at the next step and there is no “accumulation” of order in the absorbed transfer functions in the sense that at each iteration new scaling functions are generated and old ones are discarded.

The μ values and the $\mathcal{H}_\infty$-norm of DMD^{-1} at Iteration 2 are shown in Figure 8.29. Note the approximation error in the range from 1 rad/s to 1000 rad/s.

The results after the third iteration are:

```
Iteration Number:  3
--------------------

Resetting value of Gamma min based on D_11, D_12, D_21 terms

Test bounds:      0.9500 <  gamma  <=      1.4889

gamma  hamx_eig  xinf_eig  hamy_eig   yinf_eig   nrho_xy  p/f
1.489  1.1e-002  4.4e-011  1.3e-003   -8.9e-019  0.2997    p
1.219  1.1e-002  4.4e-011  1.3e-003   1.8e-019   0.5799    p
1.085  1.1e-002  4.4e-011  1.3e-003   -4.0e-018  0.9993    p
1.017  1.1e-002  4.4e-011  1.3e-003   -1.9e-018  1.5626#   f
1.071  1.1e-002  4.4e-011  1.3e-003   -2.5e-019  1.0760#   f
1.082  1.1e-002  4.4e-011  1.3e-003   -3.4e-020  1.0138#   f

 Gamma value achieved:     1.0847
```

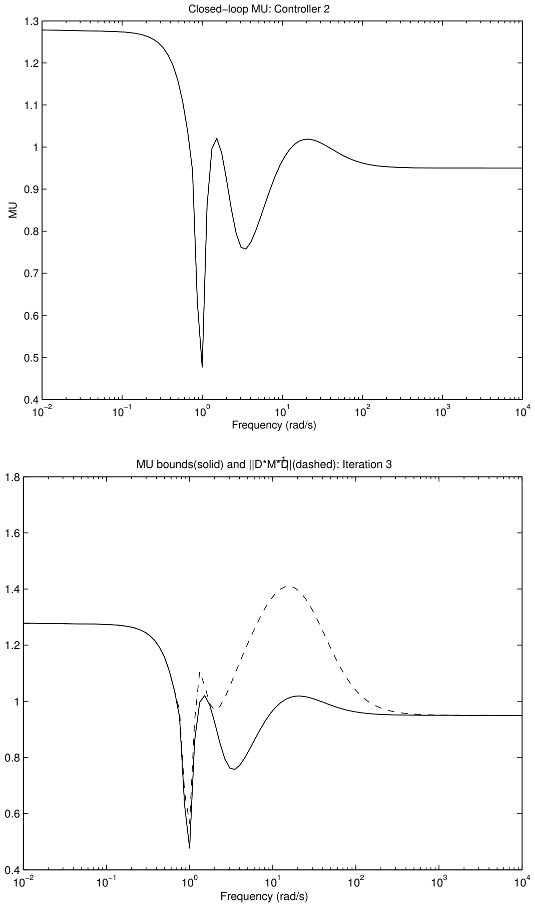


Fig. 8.29. μ and D–scaling at second iteration

```
Calculating MU of closed-loop system:
points completed....
1.2.3.4.5.6.7.8.9.10.11.12.13.14.15.16.17.
18.19.20.21.22.23.24.25.26.27.28.29.30.31.32.33.34.35.
36.37.38.39.40.41.42.43.44.45.46.47.48.49.50.51.52.53.
54.55.56.57.58.59.60.61.62.63.64.65.66.67.68.69.70.71.
72.73.74.75.76.77.78.79.80.81.82.83.84.85.86.87.88.89.
90.91.92.93.94.95.96.97.98.99.100.
points completed....
1.2.3.4.5.6.7.8.9.10.11.12.13.14.15.16.17.
18.19.20.21.22.23.24.25.26.27.28.29.30.31.32.33.34.35.
36.37.38.39.40.41.42.43.44.45.46.47.48.49.50.51.52.53.
```

```
54.55.56.57.58.59.60.61.62.63.64.65.66.67.68.69.70.71.
72.73.74.75.76.77.78.79.80.81.82.83.84.85.86.87.88.89.
90.91.92.93.94.95.96.97.98.99.100.
Iteration Summary
--------------------------------------------------
Iteration #                     1          2          3
Controller Order                4         22         24
Total D-Scale Order             0         18         20
Gamma Acheived              3.097      1.333      1.085
Peak mu-Value               2.340      1.274      1.079

Auto Fit in Progress
     Block 1, MaxOrder=4, Order = 0 1 2 3
     Block 2, MaxOrder=4, Order = 0 1 2 3
     Block 3, MaxOrder=4, Order = 0 1 2
     Block 4, MaxOrder=4, Order = 0
```

The value of γ after this iteration is 1.085, and the maximum value of μ is 1.079. The elements of the scaling matrix are approximated by transfer functions of 3rd, 3rd, 2nd and 0th orders, respectively, so that the order of the open-loop system for computation of a controller at the next iteration becomes 20.

The μ values of M and the $\mathcal{H}_\infty$-norm of DMD^{-1} at the third iteration are shown in Figure 8.30.

After the fourth iteration we obtain:

```
Iteration Number:  4
--------------------

Resetting value of Gamma min based on D_11, D_12, D_21 terms

Test bounds:      0.9500 <  gamma  <=      1.1153

gamma  hamx_eig  xinf_eig  hamy_eig   yinf_eig   nrho_xy  p/f
1.115  7.1e-003  1.3e-009  1.3e-003   -5.3e-021  0.3322    p
1.033  7.1e-003  1.4e-009  1.3e-003   -9.0e-019  0.4737    p
0.991  7.1e-003  1.4e-009  1.3e-003   -2.3e-019  0.6042    p
0.971  7.1e-003  1.4e-009  1.3e-003   7.7e-022   0.7069    p

 Gamma value achieved:     0.9707
Calculating MU of closed-loop system:
points completed....
1.2.3.4.5.6.7.8.9.10.11.12.13.14.15.16.17.
18.19.20.21.22.23.24.25.26.27.28.29.30.31.32.33.34.35.
36.37.38.39.40.41.42.43.44.45.46.47.48.49.50.51.52.53.
54.55.56.57.58.59.60.61.62.63.64.65.66.67.68.69.70.71.
```

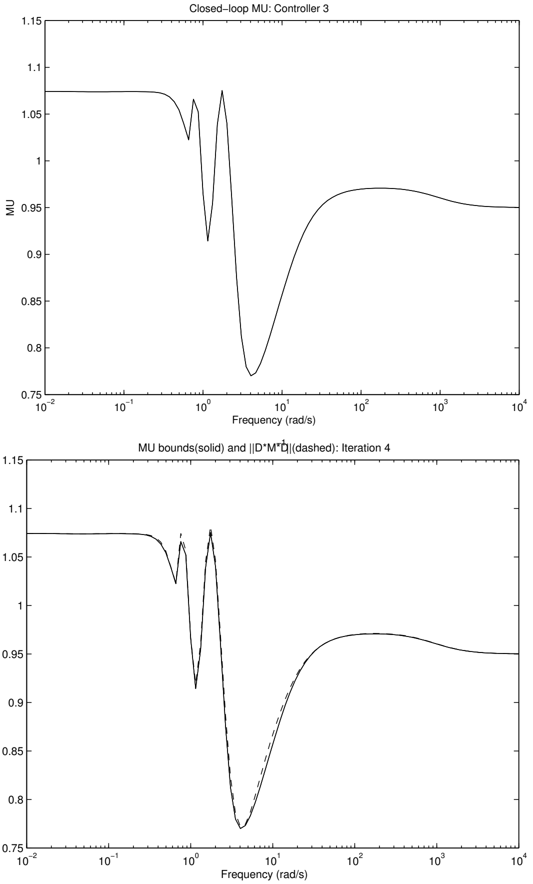


Fig. 8.30. μ and D–scaling at third iteration

```
72.73.74.75.76.77.78.79.80.81.82.83.84.85.86.87.88.89.
90.91.92.93.94.95.96.97.98.99.100.
points completed....
1.2.3.4.5.6.7.8.9.10.11.12.13.14.15.16.17.
18.19.20.21.22.23.24.25.26.27.28.29.30.31.32.33.34.35.
36.37.38.39.40.41.42.43.44.45.46.47.48.49.50.51.52.53.
54.55.56.57.58.59.60.61.62.63.64.65.66.67.68.69.70.71.
72.73.74.75.76.77.78.79.80.81.82.83.84.85.86.87.88.89.
90.91.92.93.94.95.96.97.98.99.100.
Iteration Summary
--------------------------------------------------
Iteration #                1        2        3        4
Controller Order           4       22       24       20
```

```
Total D-Scale Order          0        18        20        16
Gamma Acheived           3.097     1.333     1.085     0.971
Peak mu-Value            2.340     1.274     1.079     0.965

 Next MU iteration number:  5
```

It is seen that at this iteration the value of γ decreases to 0.971 and the value of μ becomes equal to 0.965, which means that robust performance has been achieved.

The μ plot of the closed-loop system with the newly obtained controller $K_{\rm mu}$ at Iteration 4 is shown in Figure 8.31.

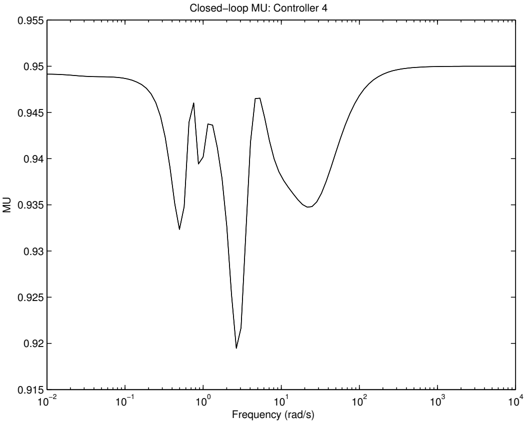


Fig. 8.31. μ plot after fourth iteration

8.10 Robust Stability and Performance of $K_{\rm mu}$

In Figure 8.32 we show the sensitivity function of the closed-loop system with the 20th-order controller $K_{\rm mu}$. It is clear that the sensitivity function is below the inverse of the performance weighting function, which shows that the nominal performance is achieved.

The robust stability of the closed-loop system with μ-controller is analysed by calling the file `rob_mds.m`, with defining in advance

```
K = K_mu;
```

The upper and lower bounds of μ are shown in Figure 8.33. It is seen that in this case the robust stability of the closed-loop system is achieved since the maximum value of μ is equal to 0.468, *i.e.*, the system stability is preserved for $\|\varDelta\|_\infty < \frac{1}{0.468}$.

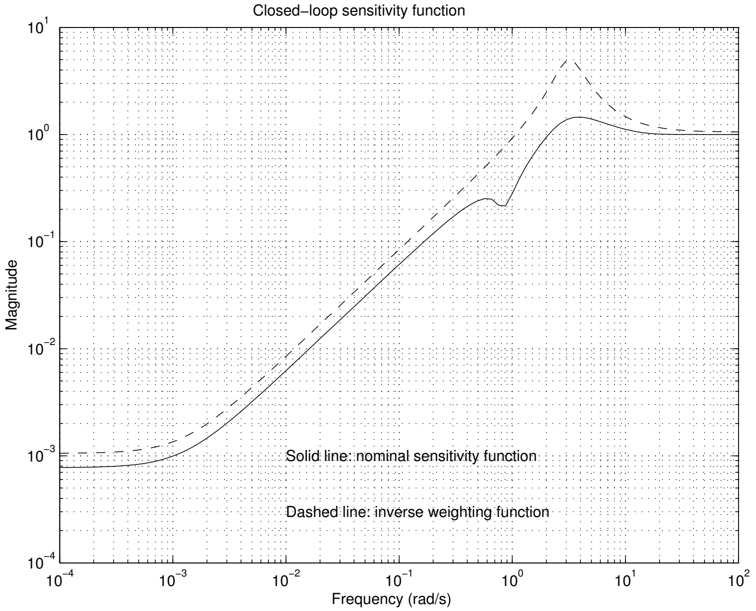


Fig. 8.32. Sensitivity function of K_{mu}

The frequency responses of the nominal and robust performance criteria are obtained by the commands from the file `nrp_mds.m`, also used in the analysis of the $\mathcal{H}_\infty$ controller, and are shown in Figure 8.34. The maximum value of μ in the robust performance analysis is 0.96. This means that the closed-loop system with μ-controller achieves robust performance since

$$\left\| \begin{bmatrix} W_p(I + F_U(G_{\mathrm{mds}}, \Delta)K)^{-1} \\ W_u K(I + F_U(G_{\mathrm{mds}}, \Delta)K)^{-1} \end{bmatrix} \right\|_\infty < 0.96$$

for every diagonal Δ with $\|\Delta\|_\infty < 1$.

To illustrate the robust properties of the system with the μ-controller, we show in Figure 8.35 the frequency response of the sensitivity functions of the perturbed closed-loop systems, obtained by using the file `psf_mds.m` below.

File `psf_mds.m`

```
%
% Sensitivity functions of perturbed systems
%
sim_mds
clp = starp(sim_ic,K);
```

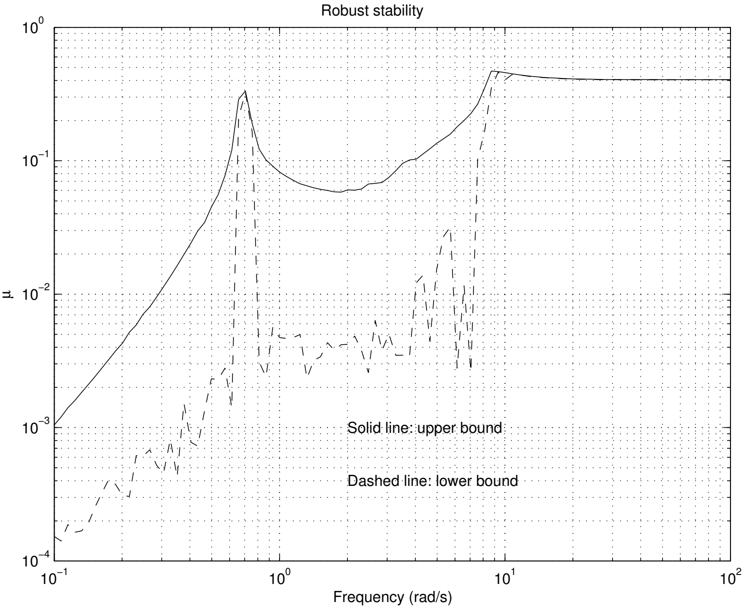


Fig. 8.33. Robust stability of K_{mu}

```
%
% inverse performance weighting function
wts_mds
omega = logspace(-4,2,100);
Wp_g = frsp(Wp,omega);
Wpi_g = minv(Wp_g);
figure(1)
vplot('liv,lm',Wpi_g,'m--')
hold on
[delta1,delta2,delta3] = ndgrid([-1 0 1],[-1 0 1], ...
                                [-1 0 1]);
for j = 1:27
    delta = diag([delta1(j),delta2(j),delta3(j)]);
    clp = starp(delta,starp(sim_ic,K));
    sen_loop = sel(clp,1,2);
    clp_g = frsp(sen_loop,omega);
    figure(1)
    vplot('liv,lm',clp_g,'c-')
    hold on
end
title('PERTURBED SENSITIVITY FUNCTIONS')
xlabel('Frequency (rad/s)')
```

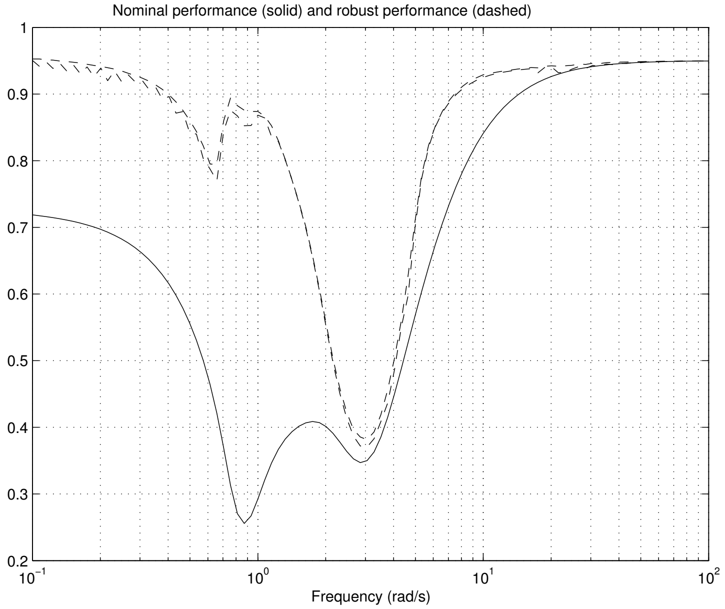


Fig. 8.34. Nominal and robust performance of K_{mu}

```
ylabel('Magnitude')
hold off
```

It is seen that the frequency responses of the perturbed sensitivity functions remain below the frequency response of the inverse of the performance weighting function.

In Figure 8.36 we show the magnitude responses of

$$\begin{bmatrix} W_p(I + F_U(G_{\mathrm{mds}}, \Delta)K)^{-1} \\ W_u K(I + F_U(G_{\mathrm{mds}}, \Delta)K)^{-1} \end{bmatrix}$$

(the weighted, mixed sensitivity function) of the perturbed system, using the following file `ppf_mds.m`. It is clear from Figure 8.36 that for all perturbed systems, the magnitudes over the frequency range are below the criterion for the closed-loop system robust performance.

File `ppf_mds.m`

```
%
% Performance of the perturbed closed-loop systems
%
% perturbed peformance
```

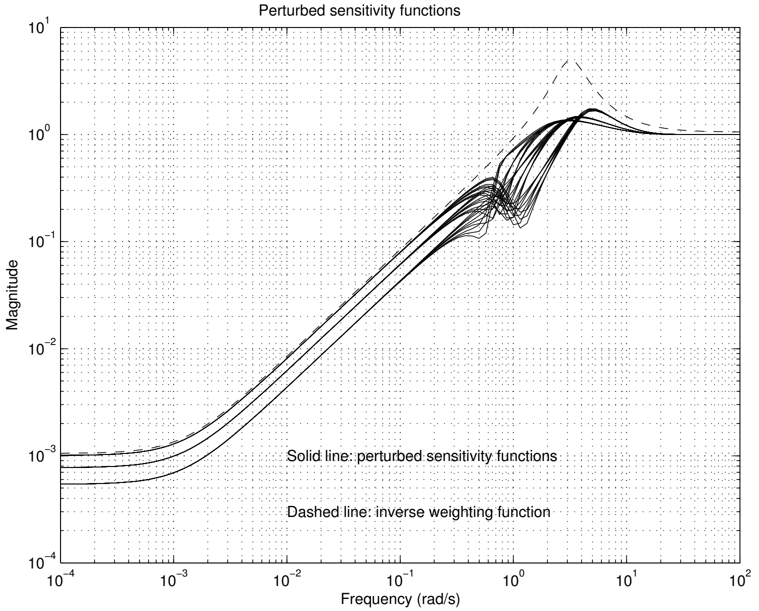


Fig. 8.35. Sensitivity functions of perturbed systems with K_{mu}

```
omega = logspace(-2,2,100);
[delta1,delta2,delta3] = ndgrid([-1 0 1],[-1 0 1], ...
                                [-1 0 1]);
for j = 1:27
    delta = diag([delta1(j),delta2(j),delta3(j)]);
    clp = starp(delta,starp(sys_ic,K));
    clp_g = frsp(clp,omega);
    figure(1)
    vplot('liv,lm',vsvd(sel(clp_g,1:2,1)),'c-')
    hold on
end
%
% robust performance
clp_ic = starp(sys_ic,K);
clp_g = frsp(clp_ic,omega);
% real perturbations
blkrsR = [-1 1;-1 1;-1 1];
rob_perf = clp_g;
blkrp = [blkrsR;[1 2]];
bndsrp = mu(rob_perf,blkrp);
figure(1)
```

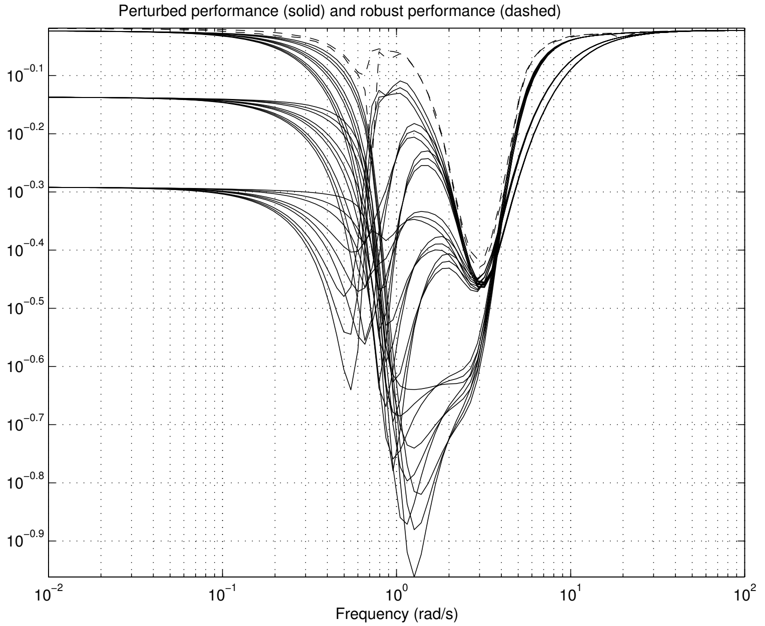


Fig. 8.36. Performance of perturbed systems with K_{mu}

```
    vplot('liv,lm',sel(bndsrp,1,1),'y--',sel(bndsrp,1,2),'m--')
    tmp1 = 'PERTURBED PERFORMANCE (solid) and';
    tmp2 = ' ROBUST PERFORMANCE(dashed)';
    title([tmp1 tmp2])
    xlabel('Frequency (rad/s)')
    hold off
```

In Figure 8.37 we show the frequency responses of the perturbed closed-loop systems. These responses are obtained by the commands included in the file pcf_mds.m.

File pcf_mds.m

```
%
% Frequency responses of the perturbed closed-loop
% systems
%
sim_mds
omega = logspace(-1,2,100);
[delta1,delta2,delta3] = ndgrid([-1 0 1],[-1 0 1], ...
                                [-1 0 1]);
```

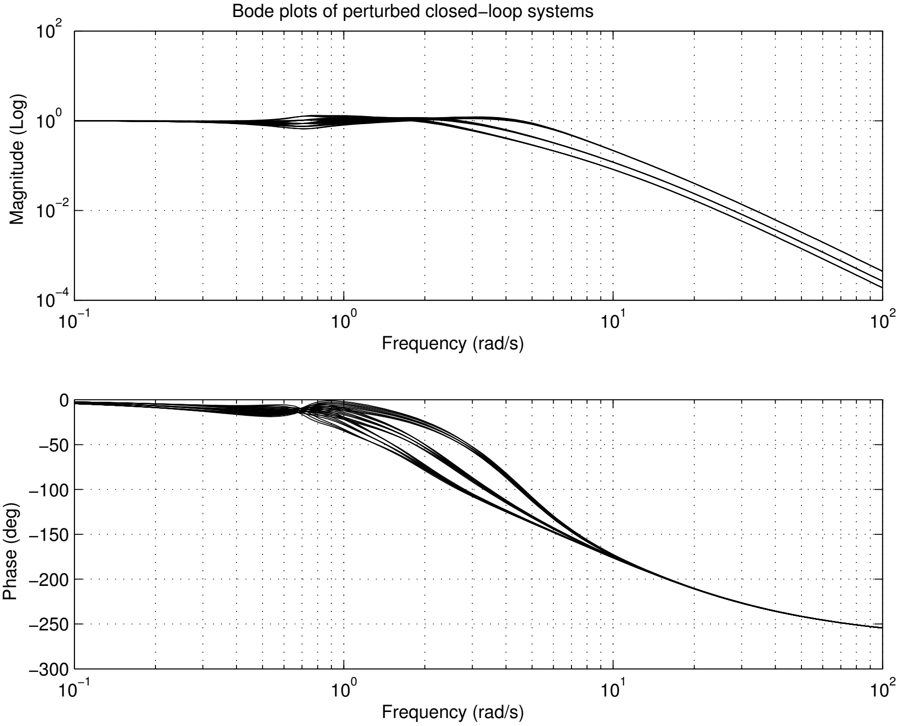


Fig. 8.37. Frequency responses of perturbed systems with K_{mu}

```
for j = 1:27
    delta = diag([delta1(j),delta2(j),delta3(j)]);
    clp = starp(delta,starp(sim_ic,K));
    clp_g = frsp(clp,omega);
    figure(1)
    vplot('bode',sel(clp_g,1,1),'c-')
    subplot(2,1,1)
    hold on
    subplot(2,1,2)
    hold on
end
subplot(2,1,1)
clp = starp(sim_ic,K);
clp_g = frsp(clp,omega);
vplot('bode',sel(clp_g,4,4),'r--')
subplot(2,1,1)
title('BODE PLOTS OF PERTURBED CLOSED-LOOP SYSTEMS')
hold off
```

```
subplot(2,1,2)
hold off
```

We see from Figure 8.37 that the frequency responses of the closed-loop, perturbed systems maintain their magnitude over a wider frequency band, in comparison to that of the open-loop system (Figure 8.7). Hence, faster responses would be expected with the designed closed-loop system.

In Figures 8.38 and 8.39 we show the transient responses of the system with μ-controller. Comparing with the responses in the case of the LSDP controller (Figures 8.25 and 8.26), we see that the μ-controller ensures smaller overshoot (10%) while maintaining the similar settling time.

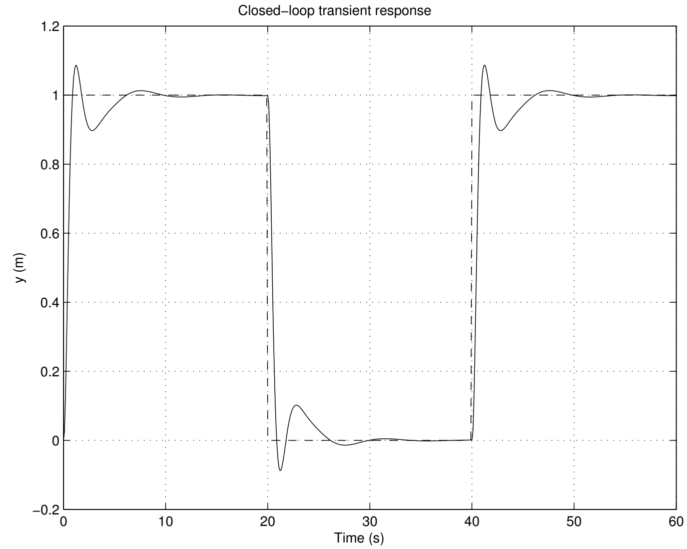


Fig. 8.38. Transient response to reference input (K_{mu})

In Figure 8.40 we show the transient responses (to the reference input) of a family of perturbed closed-loop systems with μ-controller. In all cases the overshoot does not exceed 20% that demonstrates satisfactory performance in the presence of parametric perturbations, *i.e.*, the closed-loop system achieves robust performance.

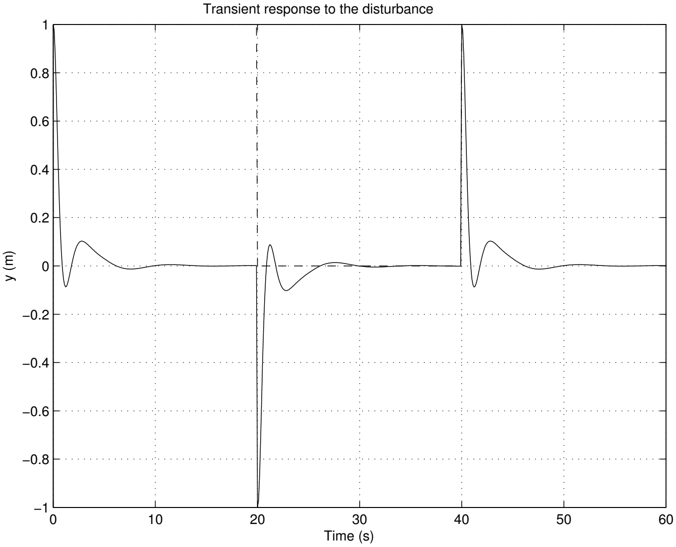


Fig. 8.39. Transient response to disturbance input (K_{mu})

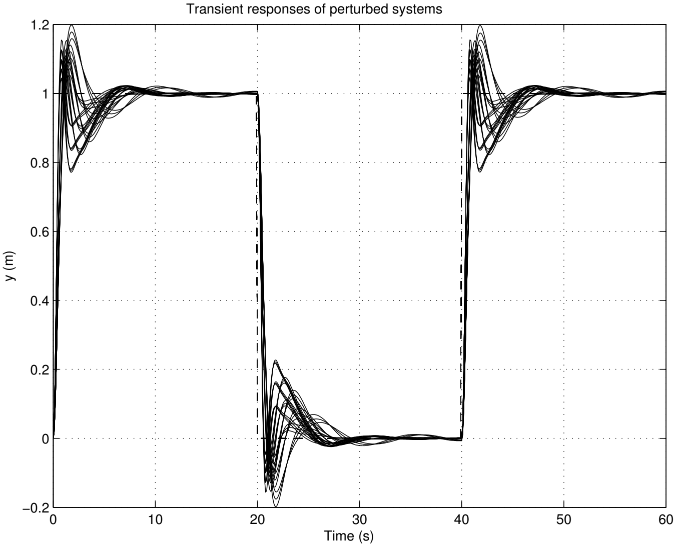


Fig. 8.40. Transient responses of perturbed systems (K_{mu})

8.11 Comparison of $\mathcal{H}_\infty$, $\mathcal{H}_\infty$ LSDP and μ-controllers

The comparison of the designed systems with $\mathcal{H}_\infty$, $\mathcal{H}_\infty$ loop-shaping and μ-controllers begins with the frequency responses of these three controllers. These responses are produced by using the file kf_mds.m below and are shown in Figure 8.41.

File kf_mds.m

```
omega = logspace(-2,3,100);
%
% H_infinity controller
K_hing = frsp(K_hin,omega);
%
% Loop Shaping controller
K_lshg = frsp(K_lsh,omega);
%
% mu-controller
K_mug = frsp(K_mu,omega);
%
figure(1)
vplot('bode',K_hing,'r-',K_lshg,'m--',K_mug,'c-.')
subplot(2,1,1)
title('BODE PLOTS OF ALL CONTROLLERS')
subplot(2,1,2)
```

It can be seen from Figure 8.41 that the $\mathcal{H}_\infty$ loop-shaping controller and μ-controller are characterised by larger gains compared with the $\mathcal{H}_\infty$ controller, in the frequency range above 5 rad/s. All the phase responses are close to each other up to about 3 rad/s and after that frequency the $\mathcal{H}_\infty$ controller continues to introduce a larger phase delay.

The comparison of frequency responses of the nominal closed-loop systems is conducted by the commands included in the file clf_mds.m, and the results are shown in Figure 8.42.

File clf_mds.m

```
sim_mds
omega = logspace(-2,2,100);
%
% H_infinity controller
clp_hin = starp(sim_ic,K_hin);
ref_hin = sel(clp_hin,4,4);
ref_hing = frsp(ref_hin,omega);
```

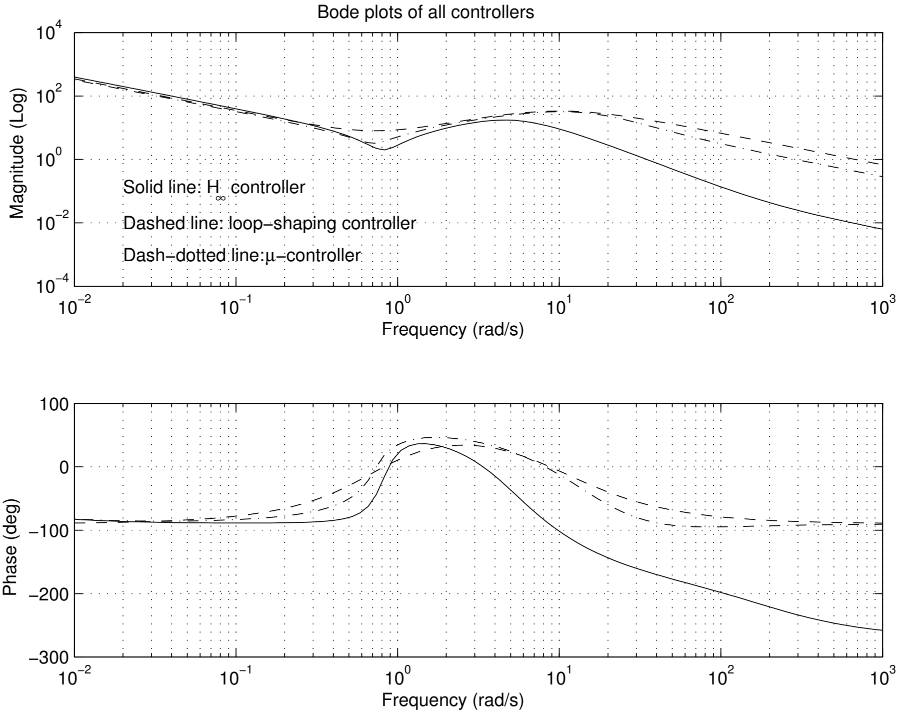


Fig. 8.41. Frequency responses of controllers

```
%
% Loop Shaping controller
clp_lsh = starp(sim_ic,K_lsh);
ref_lsh = sel(clp_lsh,4,4);
ref_lshg = frsp(ref_lsh,omega);
%
% mu-controller
clp_mu = starp(sim_ic,K_mu);
ref_mu = sel(clp_mu,4,4);
ref_mug = frsp(ref_mu,omega);
figure(1)
vplot('bode',ref_hing,'r-',ref_lshg,'m--',ref_mug,'c-.')
subplot(2,1,1)
title('BODE PLOTS OF CLOSED-LOOP SYSTEMS')
subplot(2,1,2)
```

Figure 8.42 shows that the systems with the $\mathcal{H}_\infty$ loop-shaping and μ-controllers are characterised by larger bandwidth that leads to faster dynamics of the corresponding closed-loop systems.

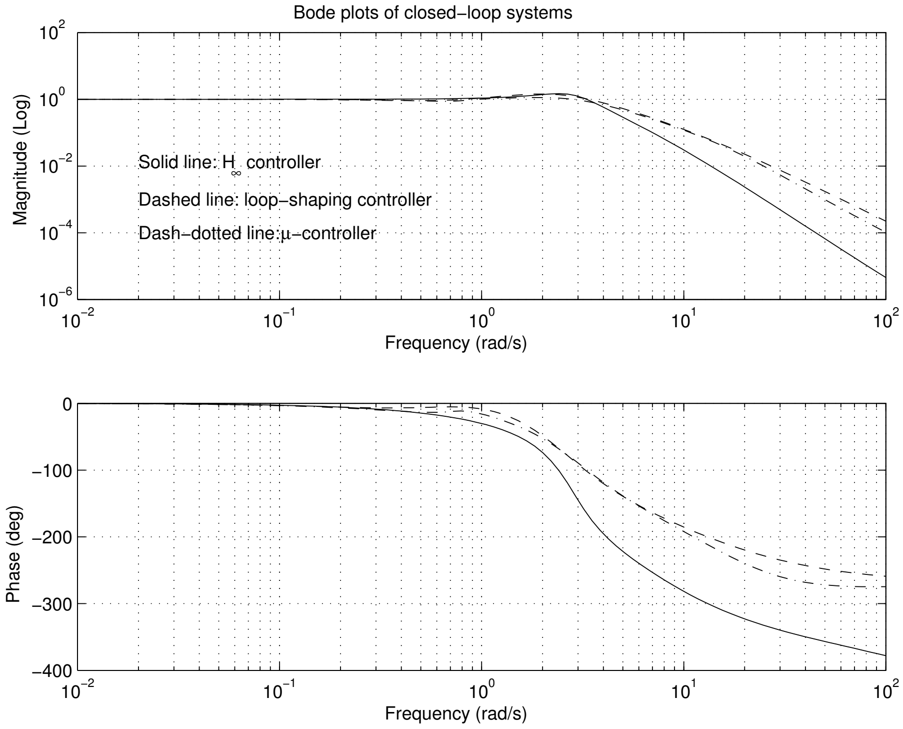


Fig. 8.42. Frequency responses of closed-loop systems

The comparison of the robust stability is conducted by the file `rbs_mds.m`.

File `rbs_mds.m`

```
omega = logspace(-1,2,100);
% Real perturbations
blkrs_R = [-1 1;-1 1;-1 1];
blkrs = [blkrsR; abs(blkrsR)];
pdim = 3;
%
% Hinf controller
clp_h = starp(sys_ic,K_hin);
clp_hg = frsp(clp_h,omega);
rob_stab = sel(clp_hg,[1:3],[1:3]);
fixl = [eye(pdim); 0.1*eye(pdim)]; % 1% Complex
fixr = fixl';
clp_mix = mmult(fixl,rob_stab,fixr);
```

```
    rbnds_h = mu(clp_mix,blkrs);
    %
    % Loop Shaping controller
    clp_lsh = starp(sys_ic,K_lsh);
    clp_lshg = frsp(clp_lsh,omega);
    rob_stab = sel(clp_lshg,[1:3],[1:3]);
    fixl = [eye(pdim); 0.1*eye(pdim)]; % 1% Complex
    fixr = fixl';
    clp_mix = mmult(fixl,rob_stab,fixr);
    rbnds_lsh = mu(clp_mix,blkrs);
    %
    % mu-controller
    clp_mu = starp(sys_ic,K_mu);
    clp_mug = frsp(clp_mu,omega);
    rob_stab = sel(clp_mug,[1:3],[1:3]);
    fixl = [eye(pdim); 0.1*eye(pdim)]; % 1% Complex
    fixr = fixl';
    clp_mix = mmult(fixl,rob_stab,fixr);
    rbnds_mu = mu(clp_mix,blkrs);
    %
    figure(1)
    vplot('liv,lm',sel(rbnds_h,1,1),'r-', ...
                sel(rbnds_lsh,1,1),'m--', ...
                sel(rbnds_mu,1,1),'c-.')
   title('ROBUST STABILITY FOR ALL CONTROLLERS')
   xlabel('Frequency (rad/s)')
   ylabel('Upper bound of \mu')
```

The frequency responses of μ for three controllers are shown in Figure 8.43. The system with the $\mathcal{H}_\infty$ loop-shaping controller is characterized with best robust stability since in this case the destabilising perturbations have the largest norm (note that the norm of these perturbations is inversely proportional to the maximum value of μ).

The nominal performance of the closed-loop systems is obtained by the file prf_mds.m. The frequency responses of the weighted sensitivity functions of the nominal systems are shown in Figure 8.44. Again, the systems with the $\mathcal{H}_\infty$ loop-shaping and μ-controllers achieve better performance.

File prf_mds.m

```
    omega = logspace(-2,2,100);
    %
    % H_infinity controller
    clp_hin = starp(sys_ic,K_hin);
```

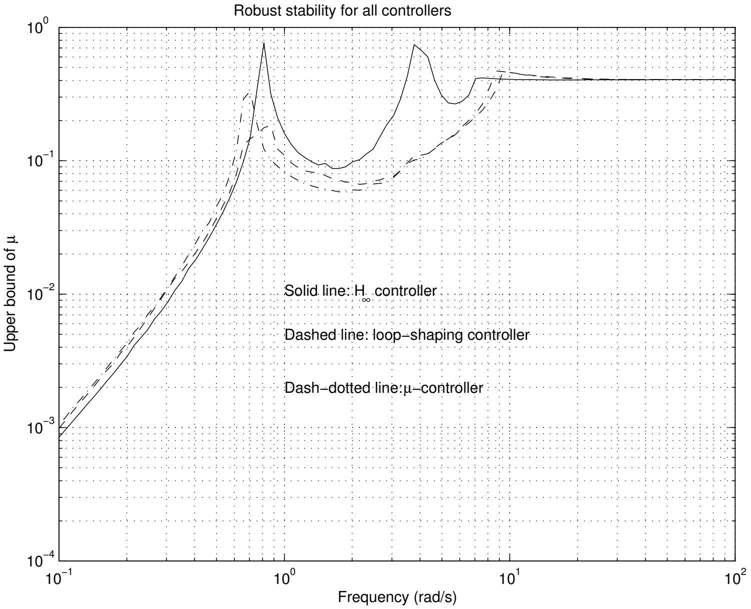


Fig. 8.43. Comparison of robust stability for 3 controllers

```
prf_hin = sel(clp_hin,4,4);
prf_hing = frsp(prf_hin,omega);
%
% Loop Shaping controller
clp_lsh = starp(sys_ic,K_lsh);
prf_lsh = sel(clp_lsh,4,4);
prf_lshg = frsp(prf_lsh,omega);
%
% mu-controller
clp_mu = starp(sys_ic,K_mu);
prf_mu = sel(clp_mu,4,4);
prf_mug = frsp(prf_mu,omega);
figure(1)
vplot('liv,m',vnorm(prf_hing),'r-',vnorm(prf_lshg), ...
      'm--',vnorm(prf_mug),'c-.')
title('NOMINAL PERFORMANCE: ALL CONTROLLERS')
xlabel('Frequency (rad/sec)')
```

The comparison of the robust performance is obtained by the file `rbp_mds.m`.

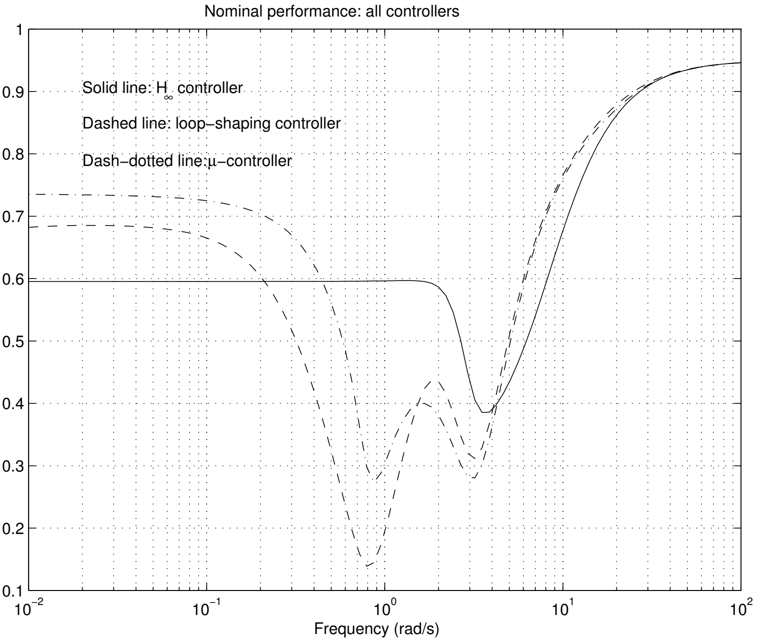


Fig. 8.44. Comparison of nominal performance for 3 controllers

File `rbp_mds.m`

```
blkrsR = [-1 1;-1 1;-1 1];
blkrp = [blkrsR;[1 2]];
omega = logspace(-2,2,100);
%
% H_infinity controller
clp_hin = starp(sys_ic,K_hin);
rbp_hin = frsp(clp_hin,omega);
bnd_hin = mu(rbp_hin,blkrp);
%
% Loop Shaping controller
clp_lsh = starp(sys_ic,K_lsh);
rbp_lsh = frsp(clp_lsh,omega);
bnd_lsh = mu(rbp_lsh,blkrp);
%
% mu-controller
clp_mu = starp(sys_ic,K_mu);
rbp_mu = frsp(clp_mu,omega);
bnd_mu = mu(rbp_mu,blkrp);
figure(1)
```

```
vplot('liv,m',bnd_hin,'r-',bnd_lsh,'m--',bnd_mu,'c-.')
title('ROBUST PERFORMANCE: ALL CONTROLLERS')
xlabel('Frequency (rad/sec)')
ylabel('\mu')
text(2.D+0,1.7D+0,'-   H_\infty controller')
text(2.D+0,1.6D+0,'--  Loop shaping controller')
text(2.D+0,1.5D+0,'-.  \mu-controller')
```

The μ values over the frequency range for all three designed systems are plotted in Figure 8.45. This confirms again that the system with the $\mathcal{H}_\infty$ controller does not achieve the robust performance criterion.

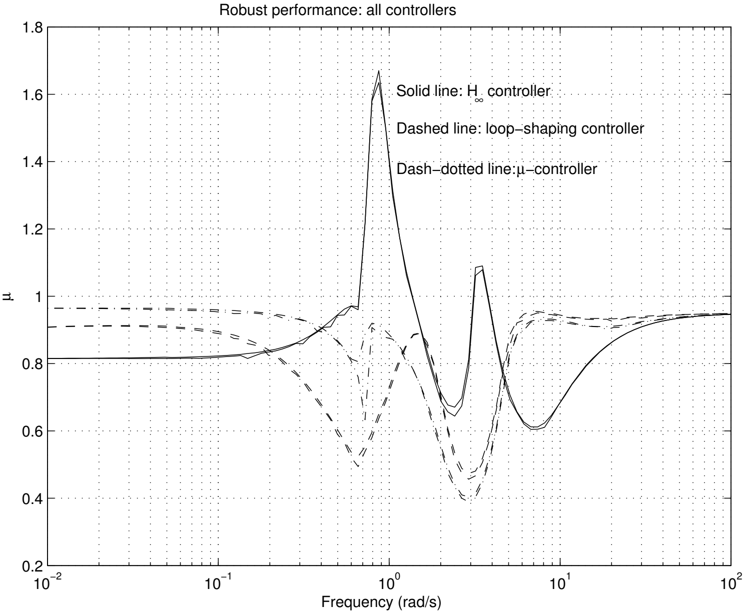


Fig. 8.45. Comparison of robust performance for 3 controllers

In summary, all three controllers ensure robust stability of the closed-loop system with respect to the parametric perturbations that are included in the 3×3 diagonal uncertainty matrix. However, the closed-loop system performance varies in a different way under the action of these diagonal perturbations. In the following file, `pdc_mds.m`, we use the function `wcperf`, which allows us to find the worst-case perturbation with respect to the performance, and to compare the performance of the three systems when the norm of perturbations increases. (The execution of this file may be accompanied by some warnings.)

File `pdc_mds.m`

```
omega = logspace(-2,3,100);
% Real perturbations
blks = [-1 1;-1 1;-1 1];
%
% H_infinity controller
clp_hin = starp(sys_ic,K_hin);
clp_hing = frsp(clp_hin,omega);
%
% Loop Shaping controller
clp_lsh = starp(sys_ic,K_lsh);
clp_lshg = frsp(clp_lsh,omega);
%
% mu-controller
clp_mu = starp(sys_ic,K_mu);
clp_mug = frsp(clp_mu,omega);
%
alpha = 0.1;
npts = 10;
[deltabadh,lowbndh,uppbndh] = wcperf(clp_hing,blks, ...
                                           alpha,npts);
[deltabadlsh,lowbndlsh,uppbndlsh] = wcperf(clp_lshg, ...
                                      blks,alpha,npts);
[deltabadmu,lowbndmu,uppbndmu] = wcperf(clp_mug,blks, ...
                                      alpha,npts);
figure(1)
vplot(lowbndh,'r-',uppbndh,'r-',lowbndlsh,'m--', ...
      uppbndlsh,'m--',lowbndmu,'c-.',uppbndmu,'c-.')
axis([0 1.8 0 2.0])
title('PERFORMANCE DEGRADATION CURVES FOR ALL CONTROLLERS')
xlabel('Size of Uncertainty')
text(1.0,0.6,'-H_\infty controller')
text(1.0,0.4,'-- Loop shaping controller')
text(1.0,0.2,'-. \mu-controller')
```

The results in Figure 8.46 show that the $\mathcal{H}_\infty$ loop-shaping controller and the μ-controller ensure robust performance for larger perturbations, with the former slightly outperforming the latter. The closed-loop performance deteriorates most rapidly with increasing perturbation magnitude in the case of the $\mathcal{H}_\infty$ controller.

The comparison results indicate that in the present exercise the $\mathcal{H}_\infty$ loop-shaping controller might be the best one to choose in terms of system robustness, though the μ-controller is not that far behind.

It is important to stress that in the above comparison the μ-controller is a reduced-order one of order 4. The order of the original μ-controller is actually

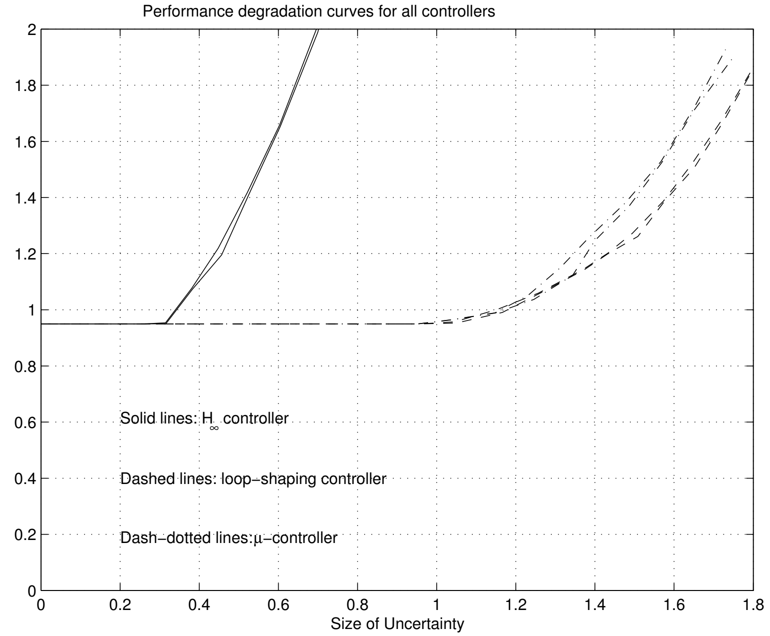


Fig. 8.46. Performance degradation for 3 controllers

20. However, the use of such a reduced-order μ-controller will be justified in the next section, which shows that the reduced-order controller does not seriously deteriorate the closed-loop system performance. It is also important to point out that for different design cases these three design approaches may well produce different results. That is, even the use of the $\mathcal{H}_\infty$ controller may lead to acceptable results in terms of performance and controller complexity. However, in general, in the case of structured uncertainty, the μ-synthesis will always produce more satisfactory and less conservative controllers.

8.12 Order Reduction of μ-controller

As shown in Section 8.9, the order of the μ-controller K_{mu} is 20, which makes it difficult in implementation. A reduced-order controller would be usually preferred. Various methods may be used to reduce the order, as discussed in Chapter 7. In this case study, the Hankel-norm approximation is used and implemented in the file `red_mds.m`. Two important commands in the file are `sysbal` and `hankmr`. The command `sysbal` generates the balanced realisation of a system (input argument) of the type `SYSTEM`. This command removes the unobservable and/or uncontrollable modes, if the system is not minimal. The command `sysbal` also returns the Hankel singular values of the system that can be used to choose the order to be reduced. To use the file `red_mds.m`, first define

```
K = K_mu;
```

File `red_mds.m`

```
omega = logspace(-2,4,100);
K_g = frsp(K,omega);
[Kb,hsig] = sysbal(K);
Kred = hankmr(Kb,hsig,4,'d');
Kred_g = frsp(Kred,omega);
vplot('bode',K_g,'y-',Kred_g,'m--')
subplot(2,1,1)
title('BODE PLOTS OF FULL AND REDUCED ORDER CONTROLLERS')
subplot(2,1,2)
K = Kred;
```

In the present case the balanced realization Kb turns out to be of 20th order as well, which means that the state-space model of the μ-controller is already minimal. The Hankel singular values are

```
hsig =

  1.0e+003 *

   1.28877293261018
   0.01757808424290
   0.01413607705650
   0.00185159191695
   0.00043945590100
   0.00027713314731
   0.00013630163054
   0.00006894336934
   0.00002465391647
   0.00000787583669
   0.00000779094168
   0.00000247258791
   0.00000053897770
   0.00000023805949
   0.00000016178585
   0.00000008448505
   0.00000001681482
   0.00000000066695
   0.00000000021363
   0.00000000000201
```

Some of these values are very small, which suggests that the controller order can be greatly reduced. In this exercise this order is set equal to 4 in the command `hankmr` and the test with this reduced-order controller shows that the closed-loop system obtained is stable. Actually, a 3rd-order controller has been tried as well, but noticeable differences exist in the frequency responses in the range of $10^{-1} - 10^{1}$ rad/s. The system matrices of this 4th order controller are

```
[Ak,Bk,Ck,Dk] = unpck(K)

Ak =

  -5.94029444678383 -12.71890180361086  11.09659323806946
  15.39552256762868 -29.53184484831400 -16.41287136235254
                  0                  0  -6.84730630402175
                  0                  0                  0

                                        -0.04081607183541
                                         0.00507186515891
                                         0.03584524485830
                                        -0.00125108980521
Bk =

 -15.96860717359783
   7.91326562204500
  13.97349755769254
  -1.79576743717989


Ck =

 -12.96086158489860 -14.24741790999237   9.28057043520263
                                        -1.79885756979541


Dk =

   0.54044998732827
```

In Figure 8.47 we display the frequency responses of the full-order and reduced-order controllers. The responses plots practically coincide with each other, for frequencies up to 100 rad/s (up to about 1000 rad/s in magnitude) that ensures almost the same closed-loop performance for both controllers. In particular, the transient responses of the closed-loop systems with full-order and reduced-order controllers are practically indistinguishable (compare Figure 8.48 with Figure 8.38). Clearly, the 4th-order controller is implemented much easier compared to the 20th-order controller.

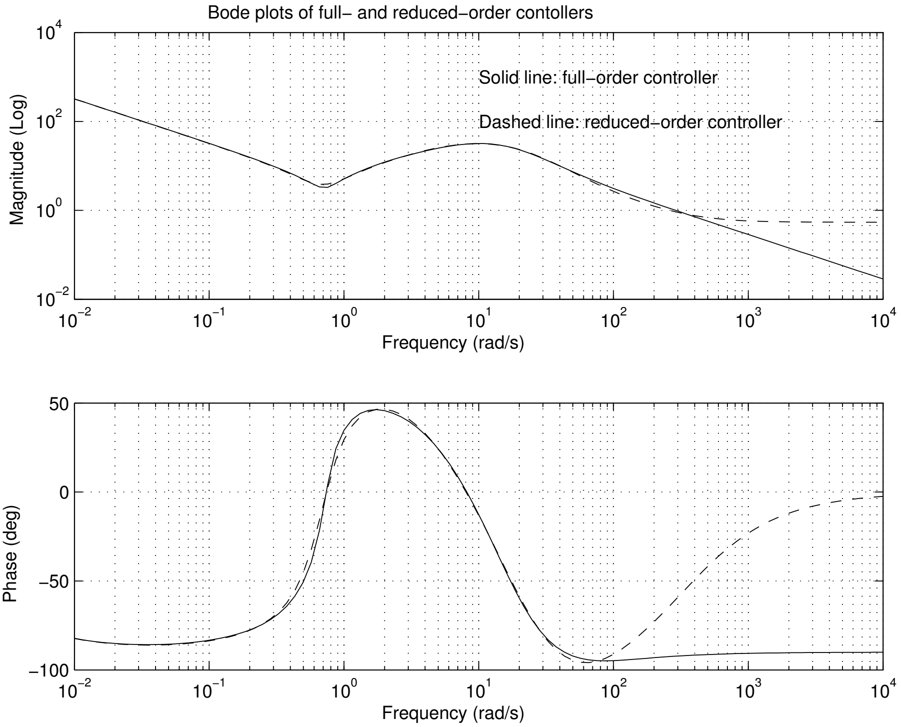


Fig. 8.47. Frequency responses of full- and reduced-order controllers

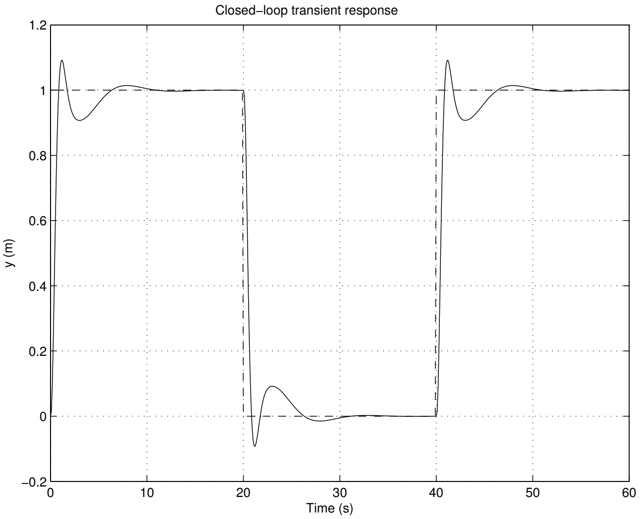


Fig. 8.48. Transient response for reduced-order controller

8.13 Conclusions

The design exercise of the mass-damper-spring system leads to the following conclusions.

- Even for a low-order plant such as the mass-damper-spring system the derivation of the uncertainty model may not be straightforward. For a system with parametric perturbations, it has to form a standard configuration in order to use the analysis and design commands from the Robust Control Toolbox. In this procedure, it may be convenient to consider separately the uncertain parameters, describe these perturbation influences by simple LFTs and construct the whole uncertainty model of the plant by using the function `sysic`.
- Finding appropriate weighting functions is a crucial step in robust control designs. It usually involves trial and error. Design experience and knowledge of the plants will help in choosing weighting functions.
- Best design results for this particular system are obtained by using the $\mathcal{H}_\infty$ loop-shaping design procedure and μ-synthesis. With nearly identical robustness properties, the μ-controller gives better transient responses with smaller overshoot than the $\mathcal{H}_\infty$ LSDP controller.
- Even for a second-order plant like this, the order of the μ-controller may be very high (20 in the given design). This makes it necessary to apply model reductions. Usually, the order may be reduced significantly without serious performance degradation.

Notes and References

The derivation of the uncertainty model of mass-damper-spring system is shown in [175, Chapter 10]. A model, obtained by using the function `sysic` and implemented by the file `sys_mds.m`, is presented in [9, Chapter 4].

Springer

http://www.springer.com/978-1-85233-983-8

Robust Control Design with MATLAB®
Gu, D.-W.; Petkov, P.; Konstantinov, M.M.
2005, XIV, 389 p. 288 illus., Softcover
ISBN: 978-1-85233-983-8