

Massachusetts Institute of Technology

Department of Electrical Engineering and Computer Science
6.245: MULTIVARIABLE CONTROL SYSTEMS

by A. Megretski

Problem Set 1 Solutions ¹

Problem 1.1

CONSIDER THE TASK OF FINDING A CONTROLLER $F = F(s)$ (WITH TWO INPUTS r, q AND ONE OUTPUT v) WHICH STABILIZES THE SYSTEM ON FIGURE 1.1 WITH

$$H(s) = \frac{10}{s + 10}, \quad P_0(s) = \frac{s}{s^2 + 1},$$

AND MINIMIZES THE H2 NORM OF THE CLOSED LOOP TRANSFER FUNCTION FROM f TO e (ESSENTIALLY, THIS MEANS MINIMIZING THE TRACKING ERROR AT LOW FREQUENCIES).

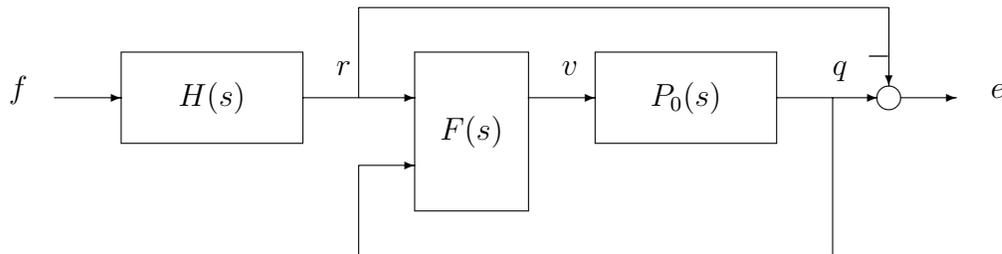


Figure 1.1: Design setup for Problem 1.1

¹Version of February 19, 2004

- (a) THE FEEDBACK OPTIMIZATION PROBLEM FORMULATED ABOVE IS A SPECIAL CASE OF A STANDARD LTI FEEDBACK OPTIMIZATION SETUP. EXPRESS THE CORRESPONDING SIGNALS w, u, z, v IN TERMS OF f, r, v, q, e , AND WRITE DOWN THE RESULTING PLANT TRANSFER MATRIX $P = P(s)$.

Use

$$w = f, \quad u = v, \quad z = e, \quad y = \begin{bmatrix} r \\ q \end{bmatrix},$$

which leads to

$$P(s) = \begin{bmatrix} -\frac{10}{s+10} & \frac{s}{s^2+1} \\ \frac{10}{s+10} & 0 \\ 0 & \frac{s}{s^2+10} \end{bmatrix}.$$

- (b) WRITE DOWN A (MINIMAL) STATE SPACE MODEL FOR P .

One possible choice of the state vector is

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} r \\ q \\ \dot{q} - v \end{bmatrix},$$

which leads to the state space model

$$\begin{aligned} \dot{x}_1 &= -10x_1 + 10w, \\ \dot{x}_2 &= x_3 + u, \\ \dot{x}_3 &= -x_2, \\ z &= x_2 - x_1, \\ y_1 &= x_1, \\ y_2 &= x_2. \end{aligned}$$

The corresponding coefficient matrices are

$$A = \begin{bmatrix} -10 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix},$$

$$C_1 = \begin{bmatrix} -1 & 1 & 0 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad D_{11} = D_{12} = 0, \quad D_{21} = D_{22} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

- (c) FIND ALL FREQUENCIES $\omega \in \mathbf{R}$ AT WHICH THE SETUP HAS CONTROL SINGULARITY OR SENSOR SINGULARITY.

Using lecture notes notation,

$$M_u(s) = \begin{bmatrix} -10 - s & 0 & 0 & 0 \\ 0 & -s & 1 & 1 \\ 0 & -1 & -s & 0 \\ -1 & 1 & 0 & 0 \end{bmatrix}.$$

A control singularity occurs when M_u is not left invertible. Since M_u is a square matrix, this is equivalent to M_u having a zero determinant. Hence, a control singularity occurs at $s = 0$ and $s = \infty$ (the latter just the consequence of having $D_{12} = 0$).

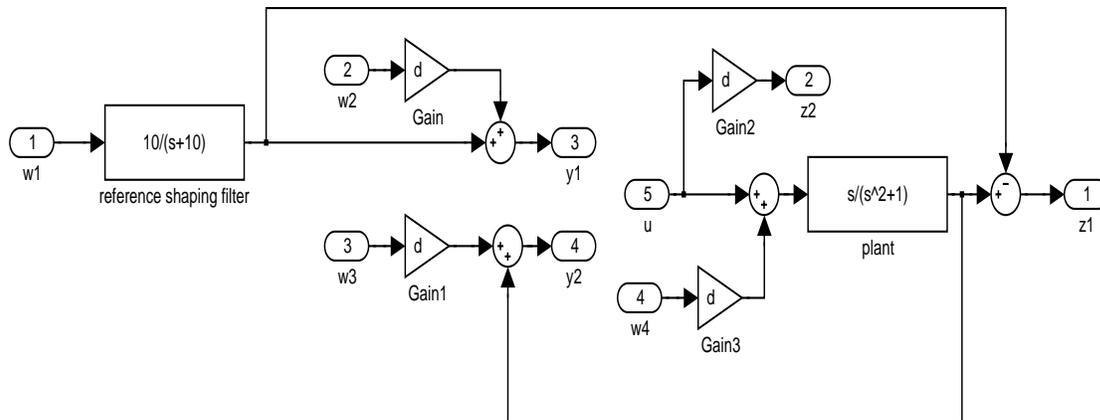
There is not much need to write down M_y : since there are more sensors than noises, the number of rows in M_y is larger than the number of its columns. Hence, the matrix is *never* right invertible, and there is a sensor singularity at *every* point of the imaginary axis.

- (d) SUGGEST A WAY TO MODIFY THE SETUP, BY INTRODUCING EXTRA COST AND DISTURBANCE VARIABLES, SCALED BY A SINGLE REAL PARAMETER $d \in \mathbf{R}$, SO THAT THE PARAMETERIZED PROBLEM BECOMES WELL-POSED FOR $d \neq 0$, AND THE ORIGINAL ILL-POSED PROBLEM IS RECOVERED AT $d = 0$.

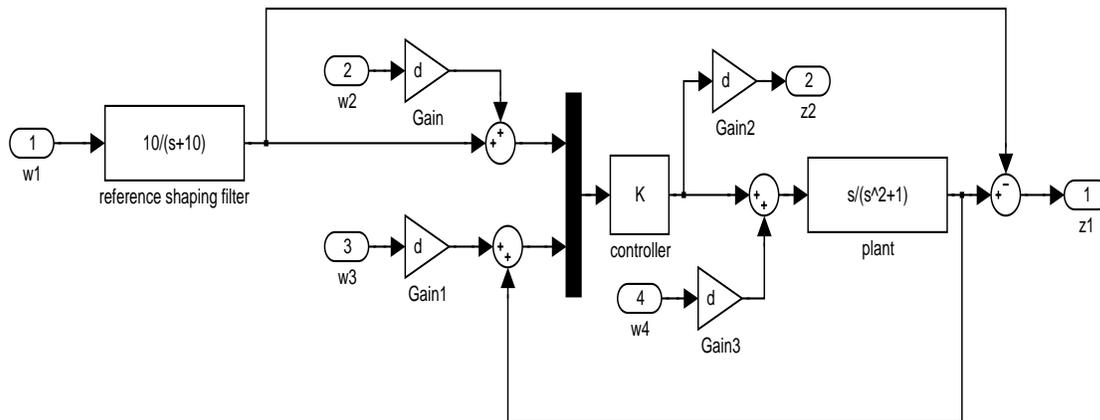
To fix sensor singularities, want to include three extra noise components, scaled by d : one added to y_1 , the other added to y_2 , and the third injected at the plant input (need the last one because the plant has a pole on the imaginary axis). To fix the control singularity, append one extra component (d times u) to the original cost z .

- (e) WRITE AND TEST A MATLAB FUNCTION, UTILIZING `h2syn.m`, WHICH TAKES $d > 0$ AS AN INPUT AND PRODUCES THE H2 OPTIMAL CONTROLLER.

The SIMULINK design diagram describing the setup is defined in `ps1_1a.mdl`:



The SIMULINK diagram for testing the controller is defined in `ps1_1b.mdl`:



The M-file handling the processing is `ps1_1.m`:

```
function ps1_1(d)
% function ps1_1(d)
%
% function for 6.245/Spring 2004 PS Problem 1.1

if nargin<1, d=0.001; end      % the default value of d
```

```

s=tf('s'); % a convenient shortcut
assignin('base','s',s); % export s to the workspace
assignin('base','d',d); % export d to the workspace
load_system('ps1_1a'); % load the design model into workspace
[a,b,c,d]=linmod('ps1_1a'); % extract the LTI model
close_system('ps1_1a'); % close the design model
p=pck(a,b,c,d); % re-write plant model in Mu-Tools format
[k,g]=h2syn(p,2,1,2,0); % design the controller
[ak,bk,ck,dk]=unpck(k); % get a state space model of the controller
K=ss(ak,bk,ck,dk); % define controller as a standard LTI object
assignin('base','K',K); % export controller into workspace
load_system('ps1_1b'); % open the testing model into workspace
[ac,bc,cc,dc]=linmod('ps1_1b'); % extract the testing model coefficients
close_system('ps1_1b'); % close the testing model
G=ss(ac,bc,cc,dc); % calculate the closed loop H2 norm
disp(['True H2 norm: ' num2str(norm(G))]); % the actual H2 norm
disp(['Promised H2 norm: ' num2str(h2norm(g))]) % H2 norm promised by h2syn

```

Running `ps1_1.m` with different values of d (meaningful results achieved for $d < 0.001$) shows that the closed loop H2 norm can be greatly reduced at the expense of spending more power to control and getting higher sensitivity to noises.

Problem 1.2

Consider the feedback design setup from Figure 1.2. IT IS FREQUENTLY CLAIMED THAT

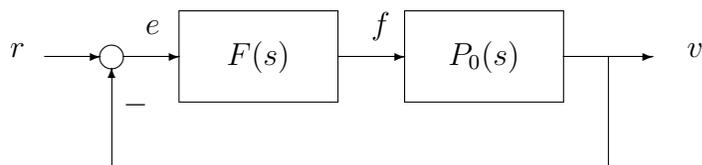


Figure 1.2: Design Setup For Problem 1.2

LOCATION OF UNSTABLE ZEROS OF P_0 LIMITS THE MAXIMAL ACHIEVABLE CLOSED LOOP BANDWIDTH, WHICH CAN BE DEFINED AS THE LARGEST $\omega_0 > 0$ SUCH THAT

$|S(j\omega)| \leq 0.1$ FOR ALL $\omega \in [0, \omega_0]$, WHERE

$$S = \frac{1}{1 + P_0F}$$

IS THE CLOSED LOOP SENSITIVITY FUNCTION. WHILE MATHEMATICALLY THIS IS NOT EXACTLY TRUE, THE ONLY WAY TO ACHIEVE A SUFFICIENTLY LARGE BANDWIDTH IS BY MAKING $|S(j\omega)|$ EXTREMELY LARGE AT OTHER FREQUENCIES.

YOU ARE ASKED TO VERIFY THIS USING H-INFINITY OPTIMIZATION ON THE FOLLOWING SETUP. LET

$$P_0(s) = \frac{s - a}{s + 1},$$

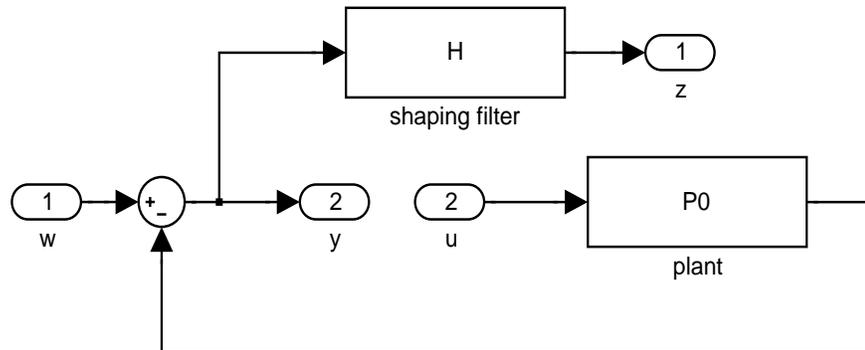
WHERE $a > 0$ IS A POSITIVE PARAMETER (LOCATION OF THE UNSTABLE ZERO). FOR

$$H(s) = 10 \frac{(s/c)^2 + \sqrt{2}s/c + 1}{(s/b)^2 + \sqrt{2}s/b + 1},$$

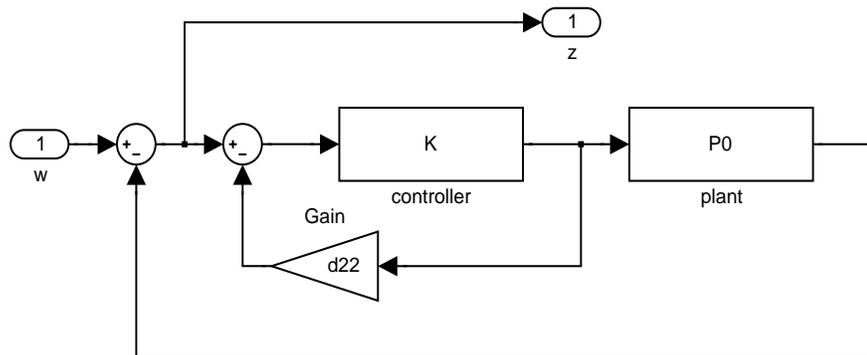
WHERE b, c ARE POSITIVE PARAMETERS, AND $c \gg b$, EXAMINE THE POSSIBILITY OF FINDING A CONTROLLER F WHICH MAKES $|S(j\omega)H(j\omega)| < 1$ FOR *all* $\omega \in \mathbf{R}$. SINCE $|H(j\omega)| \approx 10$ FOR $\omega \ll b$, AND $|H(j\omega)| \approx 10(b/c)^2 \ll 1$ FOR $\omega \gg c$, A CONTROLLER SATISFYING CONDITION $|S(j\omega)H(j\omega)| < 1$ WILL PROVIDE (AT LEAST) THE CLOSED LOOP BANDWIDTH b .

FOR ALL $a \in \{0.1, 1, 10\}$, USE H-INFINITY OPTIMIZATION TO FIND, WITH RELATIVE ACCURACY 20 PERCENT, THE MAXIMAL b SUCH THAT THE OBJECTIVE $|S(j\omega)H(j\omega)| < 1$ CAN BE ACHIEVED WITH $c = 20b$. MAKE A CONCLUSION ABOUT THE RELATION BETWEEN a AND THE ACHIEVABLE CLOSED LOOP BANDWIDTH.

The plant model for the corresponding setup will have $D_{22} \neq 0$, which apparently exposes a bug in `hinfsyn.m`. To avoid this, modify the sensor measurement by replacing y with $y_m = y - D_{22}u$ (not to forget to include this transformation when testing the resulting controller). If SIMULINK complains about algebraic loops, disable the warning in the preferences setting. The design SIMULINK diagram `ps1_2a.mdl`,



and the testing block diagram ps1_2b.mdl:



are used by the M-function ps1_2.m:

```
function ps1_2(a,b)
% function ps1_2(a,b)
%
% Solves 6.245/Spring 2004

s=tf('s');
if nargin<1, a=1; end
if nargin<2, b=0.2*a; end
c=20*b;
assignin('base','a',a);
assignin('base','b',b);
assignin('base','s',s);

% the "Laplace transform" s as a system
% default a
% default b
% extra parameter
% export variables
```

```

P0=(s-a)/(s+1); % plant and shaping filter
H=10*((s/c)^2+sqrt(2)*(s/c)+1)/((s/b)^2+sqrt(2)*(s/b)+1);
assignin('base','P0',P0); % export variables
assignin('base','H',H);
load_system('ps1_2a'); % generate state space plant model
[a,b,c,d]=linmod('ps1_2a');
close_system('ps1_2a');
d22=d(2,2); % remember D22
d(2,2)=0; % zero out D22 in the hinfsyn input
p=pck(a,b,c,d); % H-Infinity optimization
[k,g,gfin]=hinfsyn(p,1,1,0,1,0.1,2,1e-10,1e-6,0);
if ~isempty(k), % if H-Inf norm not exceeding 1 is possible
    [ak,bk,ck,dk]=unpck(k); % get the controller
    K=ss(ak,bk,ck,dk);
    assignin('base','K',K); % export controller and D22
    assignin('base','d22',d22);
    load_system('ps1_2b'); % generate closed loop model
    [ac,bc,cc,dc]=linmod('ps1_2b');
    close_system('ps1_2b');
    disp(['gmin:' num2str(gfin)])
    bode(ss(ac,bc,cc,dc)) % check the Bode plot visually
else
    disp('Infeasible specifications')
end

```

The results of optimization show that the ratio b/a is approximately equal to 0.3.