ECE 1659H Assignment 3 Solutions

Winter 2023
Instructor: J. W. Simpson-Porco

Department of Electrical and Computer Engineering
University of Toronto

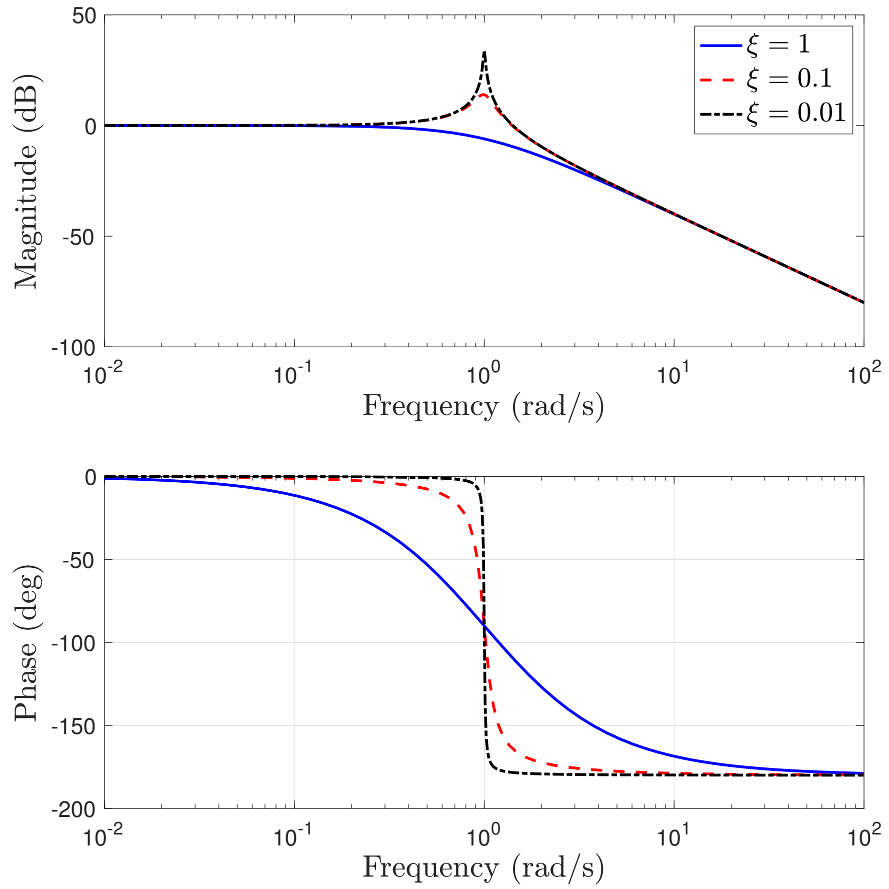
Problem 1 ($\mathcal{H}_2$ vs. $\mathcal{H}_\infty$)

The $\mathcal{H}_\infty$ norm is typically interpreted as the "worst case" gain of the system, while the $\mathcal{H}_2$ norm is interpreted as an "average" gain. Our goal in this problem is just to get a feel for this idea.

(i) Consider the stable strictly proper transfer function $\hat{G}(s) = \frac{1}{s^2+2\xi s+1}$ where $\xi \in (0,1)$. Either analytically or numerically, determine how the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms of this system change as a function of ξ.

(ii) Consider the stable strictly proper transfer function $\hat{G}(s) = 1/(s+a)$ where $a > 0$. Analytically compute the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms.

Solution: (i): The plot is shown below.



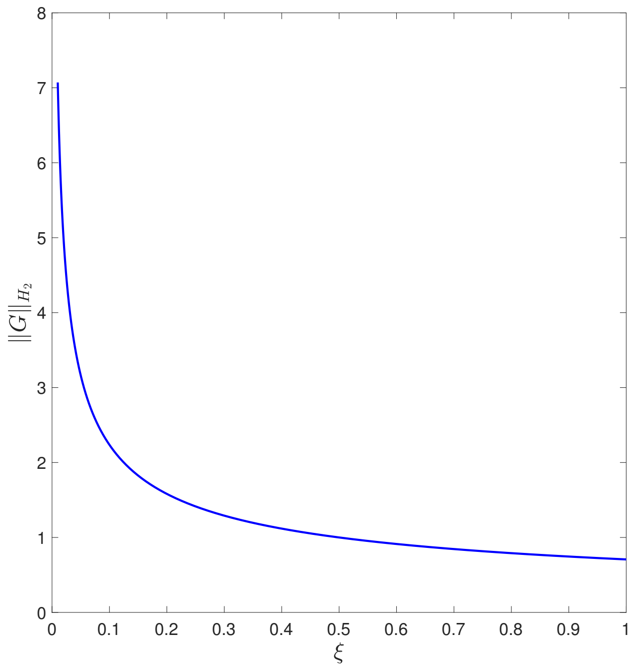
We know that

$$\|\hat{G}\|_{\mathcal{H}_\infty} = \sup_{\omega\in\mathbb{R}} \sigma_{\max}(\hat{G}(\mathbf{j}\omega)) = \sup_{\omega\in\mathbb{R}} |\hat{G}(\mathbf{j}\omega)|.$$

This is simply the maximum value on the Bode magnitude plot over frequency. We see that as the damping ratio ξ decreases, the peak on the magnitude plot increases. Therefore, the $\mathcal{H}_\infty$ norm of this operator is a decreasing function of $\xi \in (0,1)$, and satisfies

$$\lim_{\xi\to 1} \|\hat{G}\|_{\mathcal{H}_\infty} = 1.$$

As a function of $\xi \in (0,1)$, the $\mathcal{H}_2$-norm is plotted below.



Qualitatively then the situation is similar; the $\mathcal{H}_2$ norm is a decreasing function of ξ.

(ii): The magnitude of the frequency response is computed to be

$$|\hat{G}(\mathbf{j}\omega)| = \frac{1}{\sqrt{\omega^2 + a^2}}$$

Since this is a decreasing function of $\omega \geq 0$, we immediately conclude that $\|\hat{G}\|_{\mathcal{H}_\infty} = |\hat{G}(0)| = 1/a$. For the $\mathcal{H}_2$ norm, one could directly evaluate the integral

$$\|\hat{G}\|_{\mathcal{H}_2}^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{1}{\omega^2 + a^2}\, \mathrm{d}\omega = \frac{1}{2a}$$

to obtain the value $1/\sqrt{2a}$. As an alternative, consider the state-space realization

$$\dot{x} = -ax + u, \qquad y = x$$

We can apply Proposition 6.4 and solve

$$-2aY + 1 = 0 \quad \Longrightarrow \quad Y = 1/(2a)$$

yielding $\|\hat{G}\|_{\mathcal{H}_2} = \sqrt{\mathrm{trace}(B^\mathsf{T} Y B)} = \sqrt{Y} = 1/\sqrt{2a}$, so we obtain the same result. Note that the $\mathcal{H}_\infty$ norm decreases as $1/a$, while the $\mathcal{H}_2$ norm decreases as $1/\sqrt{a}$.

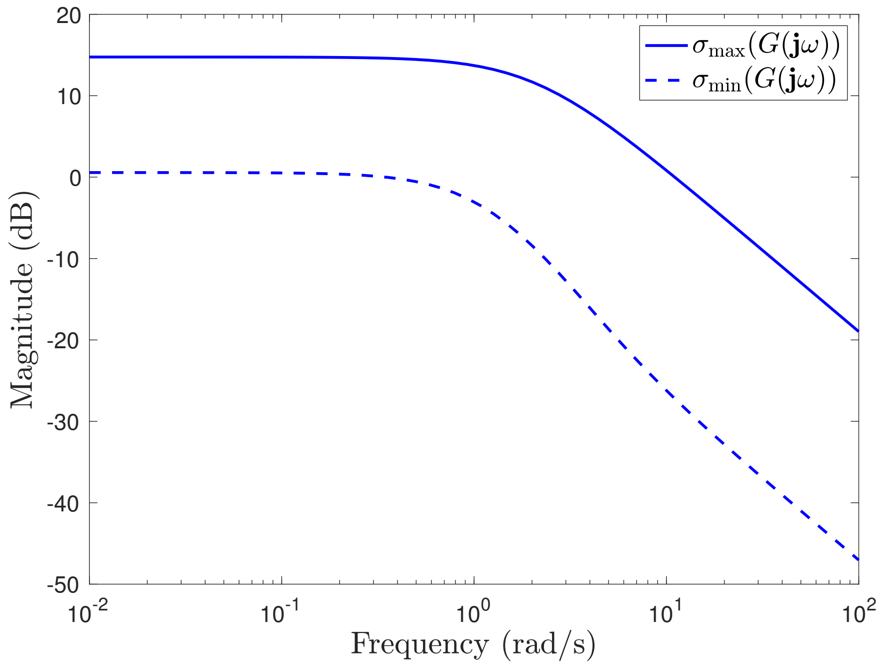
Problem 2 (Two-by-two transfer matrix)

Consider the stable transfer matrix

$$\hat{G}(s) = \begin{bmatrix} \dfrac{1}{s+1} & \dfrac{s+3}{(s+1)(s+2)} \\ \dfrac{10}{s+2} & \dfrac{5}{s+3} \end{bmatrix}$$

Plot the singular values of G as a function of frequency. Determine $\|\hat{G}\|_{\mathcal{H}_\infty}$, the frequency at which the maximum gain is achieved, and the worst-case input direction associated with the maximum gain.

Solution: The singular value plot is shown below



We can see that the maximum gain is achieved at $\omega = 0$, with a magnitude of approximately 15 dB. More precisely, we have that

$$\hat{G}(0) = \begin{bmatrix} 1 & 3/2 \\ 5 & 5/3 \end{bmatrix}.$$

with singular value decomposition

$$\hat{G}(0) = U\Sigma V^\mathsf{T} = \begin{bmatrix} -0.2710 & -0.9626 \\ -0.9626 & 0.2710 \end{bmatrix} \begin{bmatrix} 5.4671 & 0 \\ 0 & 1.0670 \end{bmatrix} \begin{bmatrix} -0.9299 & 0.3678 \\ -0.3678 & -0.9299 \end{bmatrix}^\mathsf{T}.$$

We therefore can see that

$$\|\hat{G}\|_{\mathcal{H}_\infty} = 5.46$$

with associated worst-case input direction $(u_1, u_2) = (9.299, 0.3678)$.

Problem 3 (Discrete-Time ℓ_2-Performance)

Preface to the question: Consider the vector space of right-sided discrete-time signals $\ell^m[0,\infty) = \{f : \mathbb{Z} \to \mathbb{C}^m \mid f(k) = 0 \text{ for all } k < 0\}$ and the subspace

$$\ell_2^m[0,\infty) \triangleq \{f \in \mathsf{Sig}^m[0,\infty) \mid \|f\|_{\ell_2} < +\infty\}$$

of square-summable signals where

$$\|f\|_{\ell_2} \triangleq \left(\sum_{k=-\infty}^{\infty} \|f(k)\|_2^2\right)^{1/2}.$$

This is of course analogous to the space $\mathcal{L}_2^m[0,\infty)$ for continuous-time signals. Just like for continuous-time signals, we can consider the truncation f_T of $f \in \ell^m[0,\infty)$, defined by

$$f_T(k) = \begin{cases} f(k) & \text{if } k \in \{\dots,0,\dots,T-1\} \\ 0 & \text{else} \end{cases}$$

where we set the signal equal to zero *at and after* time $T \in \mathbb{Z}_{\geq 0}$. This allows us to define the extended ℓ_2 space $\ell_{2e}[0,\infty)$ as

$$\ell_{2e}[0,\infty) \triangleq \{f \in \ell^m[0,\infty) \mid f_T \in \ell_2^m[0,\infty) \text{ for all } T \in \mathbb{Z}_{\geq 0}\}.$$

Now consider the (causal) finite-dimensional discrete-time LTI system

$$\begin{aligned} x(k+1) &= Ax(k) + Bw(k), \qquad x(0) = 0, \\ z(k) &= Cx(k) + Dw(k). \end{aligned}$$

The system is said to be ℓ_2-stable with finite ℓ_2-gain less than $\gamma > 0$ if $\|z_T\|_{\ell_2} < \gamma\|w_T\|_{\ell_2}$ for all $T \in \mathbb{Z}_{\geq 0}$ and all $w \in \ell_{2e}[0,\infty)$. The least upper bound on γ for which this inequality holds is called the ℓ_2-gain of the system.

The actual question: Show that if there exists $P \succ 0$ satisfying the strict LMI

$$\begin{bmatrix} I & 0 \\ A & B \end{bmatrix}^\mathsf{T} \begin{bmatrix} -P & 0 \\ 0 & P \end{bmatrix} \begin{bmatrix} I & 0 \\ A & B \end{bmatrix} - \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}^\mathsf{T} \begin{bmatrix} -I & 0 \\ 0 & \gamma^2 I \end{bmatrix} \begin{bmatrix} C & D \\ 0 & I \end{bmatrix} \prec 0,$$

then the system is ℓ_2-stable with finite ℓ_2-gain less than γ. Use this to numerically evaluate the ℓ_2-gain of the system defined by

$$A = \begin{bmatrix} 1/3 & 1/2 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 1/3 & 1/3 & 1/3 \\ 0 & 1/3 & 1/3 & 1/3 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}, \quad D = 1/3.$$

Solution: Let $w \in \ell_{2e}^m[0,\infty)$ with corresponding sequences x and z defined through the dynamics. Left and right multiplying the LMI by $(x(k), w(k))$ we obtain

$$\begin{bmatrix} x(k) \\ Ax(k)+Bu(k) \end{bmatrix}^\mathsf{T} \begin{bmatrix} -P & 0 \\ 0 & P \end{bmatrix} \begin{bmatrix} x(k) \\ Ax(k)+Bu(k) \end{bmatrix} - \begin{bmatrix} Cx(k)+Dw(k) \\ w(k) \end{bmatrix}^\mathsf{T} \begin{bmatrix} -I & 0 \\ 0 & \gamma^2 I \end{bmatrix} \begin{bmatrix} Cx(k)+Dw(k) \\ w(k) \end{bmatrix} \prec 0$$

Using the dynamics, we obtain

$$x(k+1)^{\mathsf{T}}Px(k+1) - x(k)^{\mathsf{T}}Px(k) + z(k)^{\mathsf{T}}z(k) - \gamma^2 w(k)^{\mathsf{T}}w(k) < 0, \qquad k \in \mathbb{Z}_{\geq 0}.$$

Letting $T \in \mathbb{Z}_{\geq 1}$ and summing these equations from $k = 0$ to $k = T-1$ we obtain

$$\sum_{k=0}^{T-1}[x(k+1)^{\mathsf{T}}Px(k+1) - x(k)^{\mathsf{T}}Px(k)] + \sum_{k=0}^{T-1}\|z(k)\|_2^2 < \gamma^2\sum_{k=0}^{T-1}\|w(k)\|_2^2$$

The first sum telescopes, yielding

$$x(T)^{\mathsf{T}}Px(T) - x(0)^{\mathsf{T}}Px(0) + \sum_{k=0}^{T-1}\|z(k)\|_2^2 < \gamma^2\sum_{k=0}^{T-1}\|w(k)\|_2^2$$

Since $P \succ 0$, the first term is nonnegative, and since $x(0) = 0$, the second term is zero. We therefore find that

$$\sum_{k=0}^{T-1}\|z(k)\|_2^2 < \gamma^2\sum_{k=0}^{T-1}\|w(k)\|_2^2, \qquad T \in \mathbb{Z}_{\geq 1}.$$

Note however that

$$\sum_{k=0}^{T-1}\|z(k)\|_2^2 = \sum_{k=0}^{\infty}\|z_T(k)\|_2^2 = \|z_T\|_{\ell_2}^2$$

and we therefore find that

$$\|z_T\|_{\ell_2} < \gamma\|w_T\|_{\ell_2}$$

which shows the desired result. For the example, the following code returns an ℓ_2-gain of $\gamma = 6.333$.

```
%% Define System
A = [1/3,1/2,0,0;
    1/4,1/4,1/4,1/4;
    0,1/3,1/3,1/3;
    0,1/3,1/3,1/3];
B = [1;0;0;0];
C = [1,0,0,0];
D = 1/3;

n = size(A,1); m = size(B,2);
p = size(C,1);

%% Solve LMI Problem
P = sdpvar(n,n); rho = sdpvar(1,1);
small = 1e-8;
KYP = [eye(n),zeros(n,m);A,B]'*blkdiag(-P,P)*[eye(n),zeros(n,m);A,B] - ...
    [C,D;zeros(m,n),eye(m)]'*blkdiag(-eye(p),rho*eye(m))*[C,D;zeros(m,n),eye(m)];
Constraints = [P ≥ small*eye(n), KYP ≤ -small*eye(n+m)];
Cost = rho;
options = sdpsettings('solver','sdpt3','verbose',1);
sol = optimize(Constraints,Cost,options);
gamma = sqrt(value(Cost))
```

Problem 4 ($\mathcal{H}_2$-Controller Design)

Using the MATLAB command rss, generate a random continuous-time LTI system with 20 states, 5 control inputs, 5 disturbance inputs, and 6 measured outputs. As your performance output, use the LQR cost

$$z = \begin{bmatrix} x \\ \sqrt{\kappa} u \end{bmatrix}$$

for some $\kappa > 0$ that you are free to tune. For the following problems, please submit your code, SDP solver output, and associated time-domain plots.

(i) Write your own program to solve the state-feedback $\mathcal{H}_2$ controller synthesis problem. Simulate and plot the response of your closed-loop system when an impulse is applied to the first disturbance channel. On top, plot the response obtained if you instead synthesize the controller using the command h2syn. If the response is the same, great; if not, comment on why you believe the responses are different.

(ii) Write your own program to solve the output-feedback $\mathcal{H}_2$ controller synthesis problem. Simulate and plot the response of your closed-loop system when an impulse is applied to the first disturbance channel. On top, plot the response obtained if you instead synthesize the controller using the command h2syn. If the response is the same, great; if not, comment on why you believe the responses are different.

Solution: (i):

```
clc
clear all
close all

%% Define Plant

n = 20;
m = 5;
n_w = 5;
p = 6;
n_z = n+m;

sys = rss(n,n_z+p,n_w+m);

A = sys.A;
B = sys.B; Bw = B(:,1:n_w); Bu = B(:,n_w+1:n_w+m);
C = sys.C; Cz = C(1:n_z,:); Cy = C(n_z+1:n_z+p,:);
D = sys.D; Dzw = D(1:n_z,1:n_w);       Dzu = D(1:n_z,n_w+1:n_w+m);
           Dyw = D(n_z+1:n_z+p,1:n_w); Dyu = D(n_z+1:n_z+p,n_w+1:n_w+m);

%Now I will redefine the performance outputs
kappa = 10;
Qoh = eye(n);
Roh = sqrt(kappa)*eye(m);
Cz = [Qoh;zeros(m,n)];
Dzu = [zeros(n,m);Roh];
Dzw = 0*Dzw;
```

```

%Now we just rebuild the model
B = [Bw,Bu];
C = [Cz;Cy];
D = [Dzw,Dzu;Dyw,Dyu];
sys = ss(A,B,C,D);

%% Compute H2 State Feedback Controller via LMIs

X = sdpvar(n,n); Z = sdpvar(m,n,'full'); W = sdpvar(n_z,n_z);
small = 1e-6;
Constraints = [X ≥ small*eye(n), ...
               [A,Bu]*[X;Z] + ([A,Bu]*[X;Z])' + Bw*Bw'  ≤ -small*eye(n), ...
               [X,(Cz*X+Dzu*Z)';(Cz*X+Dzu*Z),W] ≥ small*eye(n+n_z)];
Cost = trace(W);
options = sdpsettings('solver','sdpt3','verbose',1);
sol = optimize(Constraints,Cost,options);
F_LMI = value(Z)*inv(value(X));
gamma_LMI = sqrt(value(Cost));

%% Compute H2 State Feedback Controller via MATLAB

[¬,¬,gamma,info] = h2syn(sys,p,m);
F_MATLAB = info.Ku;

%% Simulate H2 State Feedback Controller

T_sim = 20;

A_cl = A+Bu*F_LMI;
B_cl = Bw;
C_cl = Cz+Dzu*F_LMI;
D_cl = zeros(n_z,n_w);
sys_cl_LMI = ss(A_cl,B_cl,C_cl,D_cl);
[response_LMI,T_LMI,¬] = impulse(sys_cl_LMI,T_sim); response_LMI = ...
    response_LMI(:,:,1);

A_cl = A+Bu*F_MATLAB;
B_cl = Bw;
C_cl = Cz+Dzu*F_MATLAB;
D_cl = zeros(n_z,n_w);
sys_cl_MATLAB = ss(A_cl,B_cl,C_cl,D_cl);
[response_MATLAB,T_MATLAB,¬] = impulse(sys_cl_MATLAB,T_sim); response_MATLAB = ...
    response_MATLAB(:,:,1);

figure('Position', [300, 200, 900, 500]);
subplot(2,2,1)
plot(T_LMI,response_LMI(:,1),'b','linewidth',2)
hold on
plot(T_MATLAB,response_MATLAB(:,1),'r--','linewidth',2)
hold off
set(gca, 'FontSize', 16,'XMinorTick','on');
grid on;
ylabel('$z_1$','interpreter','latex','FontSize',20);
legend({'LMI','h2syn'},'interpreter','latex');

subplot(2,2,2)
plot(T_LMI,response_LMI(:,2),'b','linewidth',2)
```

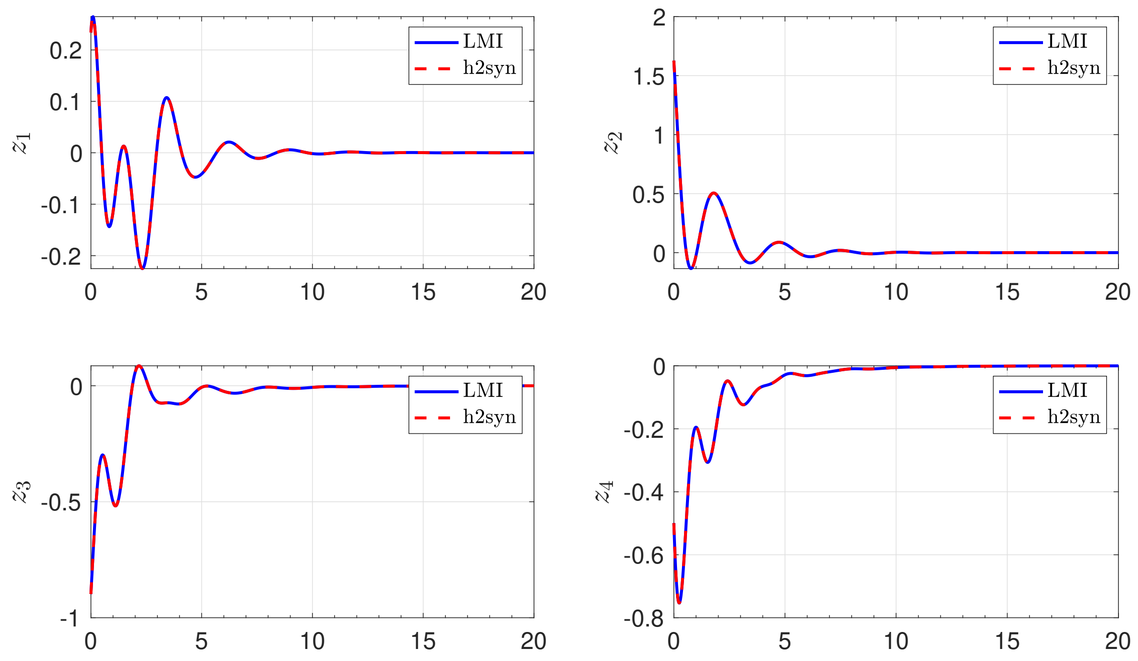
```
hold on
plot(T_MATLAB,response_MATLAB(:,2),'r--','linewidth',2)
hold off
set(gca, 'FontSize', 16,'XMinorTick','on');
grid on;
ylabel('$z_2$','interpreter','latex','FontSize',20);
legend({'LMI','h2syn'},'interpreter','latex');

subplot(2,2,3)
plot(T_LMI,response_LMI(:,3),'b','linewidth',2)
hold on
plot(T_MATLAB,response_MATLAB(:,3),'r--','linewidth',2)
hold off
set(gca, 'FontSize', 16,'XMinorTick','on');
grid on;
ylabel('$z_3$','interpreter','latex','FontSize',20);
legend({'LMI','h2syn'},'interpreter','latex');

subplot(2,2,4)
plot(T_LMI,response_LMI(:,4),'b','linewidth',2)
hold on
plot(T_MATLAB,response_MATLAB(:,4),'r--','linewidth',2)
hold off
set(gca, 'FontSize', 16,'XMinorTick','on');
grid on;
ylabel('$z_4$','interpreter','latex','FontSize',20);
legend({'LMI','h2syn'},'interpreter','latex');

set(gcf, 'color', 'none');
export_fig a3_h2sfcompare.pdf
```

The comparison between the LMI and MATLAB-generated SF controllers is shown below; they agree perfectly.



(ii):

```matlab
%% Compute H2 State Output Feedback Controller via LMIs

% Synthesis Settings

alpha_relax_H2 = 2;
gamma_relax_H2 = 1.05;

alpha_relax_HInf = 5;
gamma_relax_HInf = 1.5;

% H2 Synthesis

% Synthesis Step 1 -- Compute Optimal Gamma

X = sdpvar(n,n); Y = sdpvar(n,n); W = sdpvar(n_z,n_z);
K = sdpvar(n,n,'full');
L = sdpvar(n,p,'full');
M = sdpvar(m,n,'full');
N = sdpvar(m,p,'full');
gam = sdpvar(1,1);

Pv = [Y,eye(n);eye(n),X];
Av = [A*Y,A; zeros(n,n), X*A] + ...
    [zeros(n,n),Bu;eye(n),zeros(n,m)]*[K,L;M,N]*blkdiag(eye(n),Cy);
Bv = [Bw; X*Bw] + [zeros(n,n),Bu;eye(n),zeros(n,m)]*[K,L;M,N]*[zeros(n,n_w);Dyw];
Cv = [Cz*Y,Cz] + [zeros(n_z,n),Dzu]*[K,L;M,N]*blkdiag(eye(n),Cy);

KYP = [eye(2*n),zeros(2*n,n_w); Av, Bv; zeros(n_w,2*n),eye(n_w)];
H2lmi = KYP'*blkdiag([zeros(2*n),eye(2*n);eye(2*n),zeros(2*n)],-gam*eye(n_w))*KYP;

small = 1e-6;
Prob_Constraints = [Pv ≥ small*eye(2*n), ...
                    [Pv,Cv';Cv,W] ≥ small*eye(2*n+n_z), ...
                    Dzw + Dzu*N*Dyw == 0, ...
                    trace(W) ≤ gam, ...
                    H2lmi ≤ -small*eye(2*n+n_w)];

Constraints = [Prob_Constraints];
Cost = gam;
options = sdpsettings('solver','sdpt3','verbose',1);
sol = optimize(Constraints,Cost,options);

gamma_star = value(gam);

% Synthesis Step 2 -- Minimize Size of Decision Variables

gamma_star = gamma_relax_H2*gamma_star; %Back away a bit from optimal gamma

X = sdpvar(n,n); Y = sdpvar(n,n); W = sdpvar(n_z,n_z);
K = sdpvar(n,n,'full');
L = sdpvar(n,p,'full');
M = sdpvar(m,n,'full');
N = sdpvar(m,p,'full');
alpha = sdpvar(1,1);

```

```
Pv = [Y,eye(n);eye(n),X];
Av = [A*Y,A; zeros(n,n), X*A] + ... [zeros(n,n),Bu;eye(n),zeros(n,m)]*[K,L;M,N]*blkdiag(eye(n),Cy);
Bv = [Bw; X*Bw] + [zeros(n,n),Bu;eye(n),zeros(n,m)]*[K,L;M,N]*[zeros(n,n_w);Dyw];
Cv = [Cz*Y,Cz] + [zeros(n_z,n),Dzu]*[K,L;M,N]*blkdiag(eye(n),Cy);

KYP = [eye(2*n),zeros(2*n,n_w); Av, Bv; zeros(n_w,2*n),eye(n_w)];
H2lmi = ... KYP'*blkdiag([zeros(2*n),eye(2*n);eye(2*n),zeros(2*n)],-gamma_star*eye(n_w))*KYP;

small = 1e-6;
Prob_Constraints = [Pv ≥ small*eye(2*n), ...
              [Pv,Cv';Cv,W] ≥ small*eye(2*n+n_z), ...
              Dzw + Dzu*N*Dyw == 0, ...
              trace(W) ≤ gamma_star, ...
              H2lmi ≤ -small*eye(2*n+n_w)];

Reg_Constraints = [X ≤ alpha*eye(n), ...
                   Y ≤ alpha*eye(n), ...
                   [alpha*eye(n+m),[K,L;M,N];[K,L;M,N]',alpha*eye(n+p)] ≥ ... small*eye(n+m+n+p)];

Constraints = [Prob_Constraints, Reg_Constraints];
Cost = alpha;
options = sdpsettings('solver','sdpt3','verbose',1);
sol = optimize(Constraints,Cost,options);

alpha_star = value(alpha);

% Synthesis Step 3 -- Improve X,Y Conditioning

alpha_star = alpha_relax_H2*alpha_star;

X = sdpvar(n,n); Y = sdpvar(n,n); W = sdpvar(n_z,n_z);
K = sdpvar(n,n,'full');
L = sdpvar(n,p,'full');
M = sdpvar(m,n,'full');
N = sdpvar(m,p,'full');
beta = sdpvar(1,1);

Pv = [Y,eye(n);eye(n),X];
Av = [A*Y,A; zeros(n,n), X*A] + ... [zeros(n,n),Bu;eye(n),zeros(n,m)]*[K,L;M,N]*blkdiag(eye(n),Cy);
Bv = [Bw; X*Bw] + [zeros(n,n),Bu;eye(n),zeros(n,m)]*[K,L;M,N]*[zeros(n,n_w);Dyw];
Cv = [Cz*Y,Cz] + [zeros(n_z,n),Dzu]*[K,L;M,N]*blkdiag(eye(n),Cy);

KYP = [eye(2*n),zeros(2*n,n_w); Av, Bv; zeros(n_w,2*n),eye(n_w)];
H2lmi = ... KYP'*blkdiag([zeros(2*n),eye(2*n);eye(2*n),zeros(2*n)],-gamma_star*eye(n_w))*KYP;

small = 1e-6;
Prob_Constraints = [Pv ≥ small*eye(2*n), ...
              [Pv,Cv';Cv,W] ≥ small*eye(2*n+n_z), ...
              Dzw + Dzu*N*Dyw == 0, ...
              trace(W) ≤ gamma_star, ...
              H2lmi ≤ -small*eye(2*n+n_w)];

Reg_Constraints = [X ≤ alpha_star*eye(n), ...
```

```
                    Y ≤ alpha_star*eye(n), ...
                    [alpha_star*eye(n+m),[K,L;M,N];[K,L;M,N]',alpha_star*eye(n+p)] ...
                        ≥ small*eye(n+m+n+p), ...
                    [Y,beta*eye(n);beta*eye(n),X] ≥ small*eye(2*n)];

Constraints = [Prob_Constraints, Reg_Constraints];
Cost = -beta;
options = sdpsettings('solver','sdpt3','verbose',1);
sol = optimize(Constraints,Cost,options);

beta_val_H2 = value(beta);

% Reconstruct Controller

U = eye(n) - value(X)*value(Y);
V = eye(n);
calY = value([Y, eye(n); V',zeros(n)]);
calZ = value([eye(n), zeros(n); X, U]);
P = inv(calY')*calZ;
Controller = inv(value([U,X*Bu;zeros(m,n),eye(m)]))*value([K-X*A*Y,L;M,N])* ...
            inv(value([V',zeros(n,p);Cy*Y,eye(p)]));
Ac_H2 = Controller(1:n,1:n);
Bc_H2 = Controller(1:n,n+1:n+p);
Cc_H2 = Controller(n+1:n+m,1:n);
Dc_H2 = Controller(n+1:n+m,n+1:n+p);


%% Simulate H2 Output Feedback Controller

Con_H2 = ss(Ac_H2,Bc_H2,Cc_H2,Dc_H2);
G_cl_H2 = lft(sys,Con_H2);
G_cl_H2_MATLAB = lft(sys,Con_MATLAB);

%% Simulate H2 State Feedback Controller

T_sim = 20;
[response_OF_LMI,T_OF_LMI,¬] = impulse(G_cl_H2,T_sim); response_OF_LMI = ...
    response_OF_LMI(:,:,1);
[response_OF_MATLAB,T_OF_MATLAB,¬] = impulse(G_cl_H2_MATLAB,T_sim); ...
    response_OF_MATLAB = response_OF_MATLAB(:,:,1);

%% Plotting

figure('Position', [300, 200, 900, 500]);
subplot(2,2,1)
plot(T_OF_LMI,response_OF_LMI(:,1),'b','linewidth',2)
hold on
plot(T_OF_MATLAB,response_OF_MATLAB(:,1),'r--','linewidth',2)
hold off
set(gca, 'FontSize', 16,'XMinorTick','on');
grid on;
ylabel('$z_1$','interpreter','latex','FontSize',20);
legend({'LMI','h2syn'},'interpreter','latex');

subplot(2,2,2)
plot(T_OF_LMI,response_OF_LMI(:,2),'b','linewidth',2)
hold on
plot(T_OF_MATLAB,response_OF_MATLAB(:,2),'r--','linewidth',2)
```

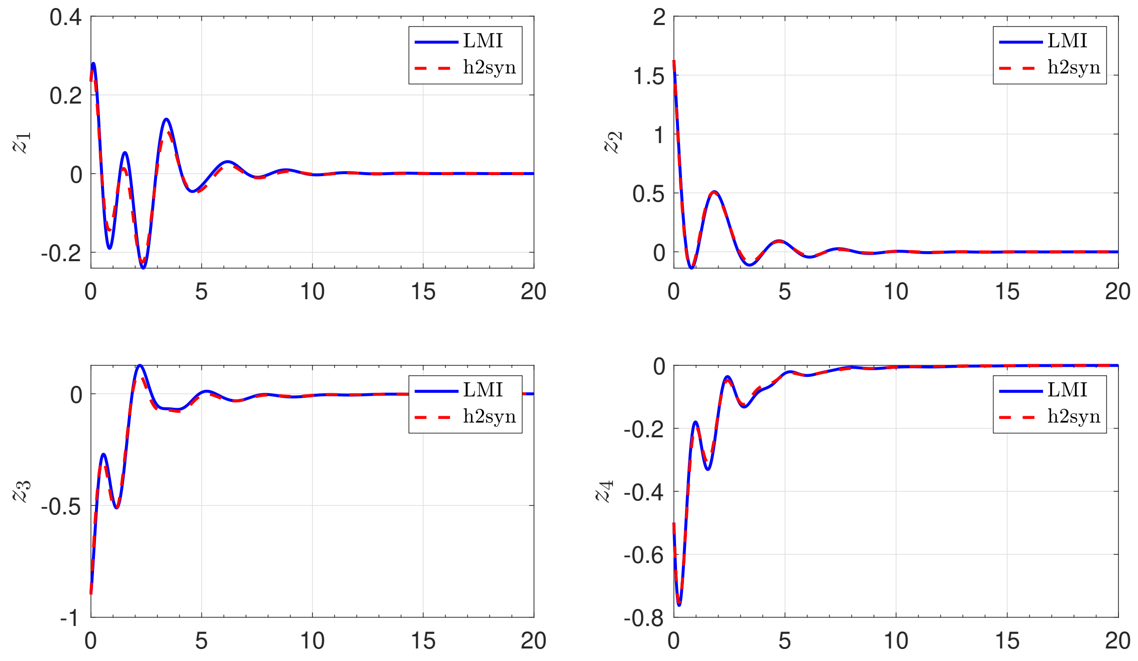
```
hold off
set(gca, 'FontSize', 16,'XMinorTick','on');
grid on;
ylabel('$z_2$','interpreter','latex','FontSize',20);
legend({'LMI','h2syn'},'interpreter','latex');

subplot(2,2,3)
plot(T_OF_LMI,response_OF_LMI(:,3),'b','linewidth',2)
hold on
plot(T_OF_MATLAB,response_OF_MATLAB(:,3),'r--','linewidth',2)
hold off
set(gca, 'FontSize', 16,'XMinorTick','on');
grid on;
ylabel('$z_3$','interpreter','latex','FontSize',20);
legend({'LMI','h2syn'},'interpreter','latex');

subplot(2,2,4)
plot(T_OF_LMI,response_OF_LMI(:,4),'b','linewidth',2)
hold on
plot(T_OF_MATLAB,response_OF_MATLAB(:,4),'r--','linewidth',2)
hold off
set(gca, 'FontSize', 16,'XMinorTick','on');
grid on;
ylabel('$z_4$','interpreter','latex','FontSize',20);
legend({'LMI','h2syn'},'interpreter','latex');

set(gcf, 'color', 'none');
export_fig a3_h2ofcompare.pdf
```

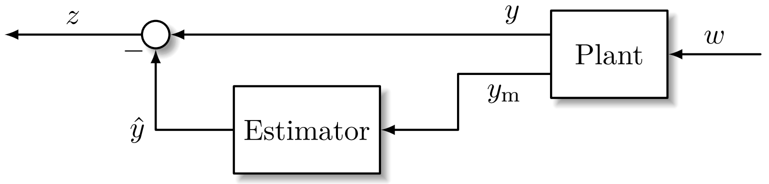
The comparison between the LMI and MATLAB-generated output feedback controllers is shown below; they are qualitatively close, but not exactly the same.



One possible source of difference is the relaxation parameters used in the above LMI computation in order to improve numerical conditioning of the LMIs. Another possible source of difference is plant scaling; MATLAB's algorithm will automatically scale plant variables for improved numerical conditioning, while we are not doing any such scaling. For this example, the LMI-based procedure produced an optimal $\gamma = 6.8$, while MATLAB produced $\gamma = 5.9$. I would probably take MATLAB's answer as being more reliable here; the lesson is that the numerics can be non-trivial, and that LMI-based synthesis requires some care.

Problem 5 ($\mathcal{H}_2$ Estimator Design (Steady-State Kalman Filter))

The machinery we have been developing can be applied not only for the design of optimal controllers, but also for the design of optimal estimators (i.e., observers).



In the diagram above, we have a continuous-time LTI plant described by the equations

$$\begin{aligned} \dot{x} &= Ax + B_w w \\ y_{\mathrm{m}} &= C_{\mathrm{m}} x + D_{\mathrm{m}} w \\ y &= Cx \end{aligned}$$

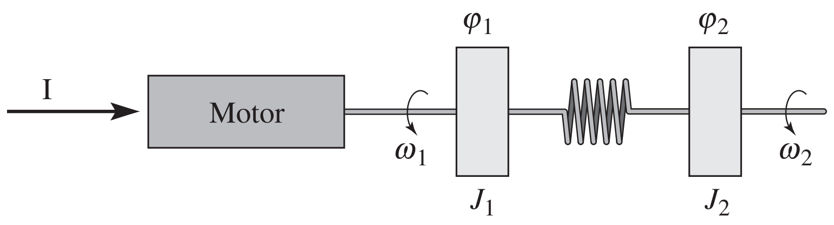
The input w is white noise which disturbs the system. The output y_{m} is a measurable output, while the output y is an unmeasured output that we would like to estimate (it could be, for instance, the entire state of the plant, but it does not need to be). The measurement y_{m} is fed to an estimator, which we assume has the form

$$\begin{aligned} \dot{\xi} &= A\xi + L(\hat{y}_{\mathrm{m}} - y_{\mathrm{m}}) \\ \hat{y}_{\mathrm{m}} &= C_{\mathrm{m}}\xi \\ \hat{y} &= C\xi \end{aligned}$$

where L is the estimator gain to be designed. The error variable $z = y - \hat{y}$ obviously captures the quality of our estimate.

(i) Formulate an LMI problem for computing an estimator gain L such that (1) $A + LC_{\mathrm{m}}$ is Hurwitz, and (2) the $\mathcal{H}_2$ norm from w to z is less than or equal to some number $\gamma > 0$.

(ii) Consider the two-inertia positioning system



described by the equations

$$\begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ J_1\dot{\omega}_1 \\ J_2\dot{\omega}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k & k & -b & b \\ k & -k & b & -b \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \omega_1 \\ \omega_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} w$$

$$\begin{aligned} y_{\mathrm{m}} &= \varphi_1 + 0.1w \\ y &= \varphi_2 \end{aligned}$$

where w is models a white noise signal which influences both the first inertia and the measurement of φ_1; the parameters are found on Slide 4-82 of the notes. Apply your result from (i) to design an *optimal* $\mathcal{H}_2$ estimator for this system, and report the value of the minimally achievable norm.

Hint: Begin by showing that the LTI system mapping w to z admits the realization

$$\begin{aligned} \dot{e} &= (A + LC_{\mathrm{m}})e + (B_w + LD_{\mathrm{m}})w \\ z &= Ce \end{aligned}$$

for some appropriately defined state e.

Solution: (i): Being by introducing the new state variable $e = x - \xi$. We compute that

$$\begin{aligned} \dot{e} &= \dot{x} - \dot{\xi} \\ &= [Ax + B_w w] - [A\xi + L(\hat{y}_{\mathrm{m}} - y_{\mathrm{m}})] \\ &= [Ax + B_w w] - [A\xi + L(C_{\mathrm{m}}\xi - C_{\mathrm{m}}x - D_{\mathrm{m}}w)] \\ &= (A + LC_{\mathrm{m}})(x - \xi) + (B_w + LD_{\mathrm{m}})w \\ &= (A + LC_{\mathrm{m}})e + (B_w + LD_{\mathrm{m}})w. \end{aligned}$$

The estimation error can then be written as

$$\begin{aligned} z &= y - \hat{y} \\ &= Cx - C\xi \\ &= Ce. \end{aligned}$$

We therefore consider the LTI system M_{set} described by

$$M_{\mathrm{est}} : \quad \begin{aligned} \dot{e} &= (A + LC_{\mathrm{m}})e + (B_w + LD_{\mathrm{m}})w \\ z &= Ce \end{aligned} \qquad (1)$$

From Theorem 6.1 in the notes, the matrix $A + LC_{\mathrm{m}}$ is Hurwitz and the system (1) meets the $\mathcal{H}_2$-norm constraint $\|M_{\mathrm{rest}}\|_{\mathcal{H}_2} < \gamma$ if there exists $Y \succ 0$ such that

$$\begin{aligned} (A + LC_{\mathrm{m}})^{\mathsf{T}}Y + Y(A + LC_{\mathrm{m}}) + C^{\mathsf{T}}C &\prec 0 \\ \mathrm{trace}(B_w + LD_{\mathrm{m}})^{\mathsf{T}}Y(B_w + LD_{\mathrm{m}}) &< \gamma^2 \end{aligned}$$

Introducing the new variable $Z = YL$, the first inequality becomes

$$YA + ZC_{\mathrm{m}} + (YA + ZC_{\mathrm{m}})^{\mathsf{T}} + C^{\mathsf{T}}C \prec 0.$$

The second inequality can be rewritten as

$$\begin{aligned} & \operatorname{trace}(B_w + LD_\mathrm{m})^\mathsf{T} Y Y^{-1} Y (B_w + LD_\mathrm{m}) < \gamma^2 \\ \iff \quad & \operatorname{trace}(YB_w + ZD_\mathrm{m})^\mathsf{T} Y^{-1} (YB_w + ZD_\mathrm{m}) < \gamma^2. \end{aligned}$$

This latter inequality is equivalent to the existence of a matrix $W \succ 0$ such that

$$(YB_w + ZD_\mathrm{m})^\mathsf{T} Y^{-1} (YB_w + ZD_\mathrm{m}) \prec W, \qquad \operatorname{trace}(W) < \gamma^2$$

which by Schur complements is equivalent to

$$\begin{bmatrix} Y & (YB_w + ZD_\mathrm{m}) \\ (YB_w + ZD_\mathrm{m})^\mathsf{T} & W \end{bmatrix} \succ 0, \qquad \operatorname{trace}(W) < \gamma^2.$$

We conclude that there exists L meeting the specifications if and only if there exists $X, W \succ 0$ and Z satisfying the linear matrix inequalities

$$\begin{aligned} YA + ZC_\mathrm{m} + (YA + ZC_\mathrm{m})^\mathsf{T} + C^\mathsf{T} C \prec 0 \\ \begin{bmatrix} Y & (YB_w + ZD_\mathrm{m}) \\ (YB_w + ZD_\mathrm{m})^\mathsf{T} & W \end{bmatrix} \succ 0 \\ \operatorname{trace}(W) < \gamma^2 \end{aligned}$$

with L being recovered as $L = ZY^{-1}$.

(ii):

```matlab
%% Define Two-Mass Positioning System
k = 5; b = 5.82e-3;
J1 = 1e-3; J2 = 2e-4;
Jtot = J1+J2; Jred = J1*J2/(J1+J2);

A = [0,0,1,0;
    0,0,0,1;
    -k/J1,k/J1,-b/J1,b/J1;
    k/J2,-k/J2,b/J2,-b/J2];
Bw = [0;0;1/J1;0];
Cm = [1,0,0,0];
Dm = [0.1];
C = [0,1,0,0];

n = size(A,1); n_w = size(Bw,2);
p_m = size(Cm,1);
p = size(C,1);

%% Solve LMI Problem
Y = sdpvar(n,n); Z = sdpvar(n,p_m,'full'); W = sdpvar(n_w,n_w);
small = 1e-8;
Constraints = [Y ≥ small*eye(n), ...
              (Y*A+Z*Cm) + (Y*A+Z*Cm)' + C'*C  ≤ -small*eye(n), ...
              [Y,Y*Bw+Z*Dm;(Y*Bw+Z*Dm)',W] ≥ small*eye(n+n_w)];
Cost = trace(W);
options = sdpsettings('solver','sdpt3','verbose',1);
sol = optimize(Constraints,Cost,options);
L = inv(value(Y))*value(Z);
gamma = sqrt(value(Cost));
gamma =

   1.1025e-04
```