ESE 547 Homework 7 /

Seif Elkhashab / 04-05-2024 /

Lyapunov Stability and OBLTR Design

1. Consider the following linear time varying system

$$\ddot{y}(t) + p(t)\dot{y}(t) + e^{-t}y(t) = 0$$

which can be written in state space form as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -e^{-t} & -p(t) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

using as a Lyapunov function $v(t, x_1, x_2) = x_1^2 + e^t x_2^2$ determine conditions on $p(t)$ that guarantees the origin is stable.

For the origin to be AS, we need $\dot{V} < 0$:

$$\dot{V} = 2x_1\dot{x_1} + 2x_2e^t\dot{x_2} + e^tx_2^2$$

$$= 2x_1x_2 + 2x_2e^t(-e^{-t}x_1 - p(t)x_2) + e^tx_2^2$$

$$= 2x_1x_2 - 2x_1x_2 - 2e^tp(t)x_2^2 + e^tx_2^2$$

$$= -2e^tp(t)x_2^2 + e^tx_2^2$$

$$= e^tx_2^2(-2p(t) + 1)$$

Since $e^tx_2^2$ is always positive (except at the origin), the expression for $\dot{V}$ will always be negative if the term $(-2p(t) + 1)$ is negative. So long as $p(t) > 1/2$, the system will be stable at the origin.

2) For each of the following systems, use a quadratic Lyapunov function candidate to show that the origin is asymptotically stable:

$$i)\ \dot{x}_1 = x_1 + x_1 x_2 \qquad ii)\ \dot{x}_1 = -x_2 - x_1(1 - x_1^2 - x_2^2)$$

$$\dot{x}_2 = -x_2 \qquad \dot{x}_2 = x_1 - x_2(1 - x_1^2 - x_2^2)$$

$$iii)\ \dot{x}_1 = x_2(1 - x_1^2) \qquad iiii)\ \dot{x}_1 = -x_1 - x_2$$

$$\dot{x}_2 = -(x_1 + x_2)(1 - x_1^2) \qquad \dot{x}_2 = 2x_1 - x_2^3$$

$$i)\ \dot{x}_1 = -x_1 + x_1 x_2$$

$$\dot{x}_2 = -x_2$$

$$V = x_1^2 + x_2^2 + x_1 x_2$$

$$\dot{V} = 2x_1\dot{x_1} + 2x_2\dot{x_2} + \dot{x_1}x_2 + x_1\dot{x_2}$$

$$\dot{V} = 2x_1(-x_1 + x_1x_2) - 2x_2^2 + x_2(-x_1 + x_1x_2) - x_1x_2$$

$$\dot{V} = -2x_1^2 + 2x_1^2x_2 - 2x_2^2 - x_1x_2 + x_1x_2^2 - x_1x_2$$

$$\dot{V} = -2x_1^2 - 2x_2^2 - 2x_1x_2 + x_1x_2^2 + 2x_1^2x_2$$

Since all these terms are guarunteed to be negative, with the exception of the last two terms we can say the the system is AS for the region where $x_1 < 2 \cup x_2 < 1$

This is because the coefficients on the last two terms are dominated by the coefficients of the first two terms.

$$ii)\ \dot{x}_1 = -x_2 - x_1(1 - x_1^2 - x_2^2)$$

$$\dot{x}_2 = x_1 - x_2(1 - x_1^2 - x_2^2)$$

$$V = x_1^2 + x_2^2$$

$$\dot{V} = 2x_1(-x_2 - x_1(1 - x_1^2 - x_2^2)) + 2x_2(x_1 - x_2(1 - x_1^2 - x_2^2))$$

$$\dot{V} = -2x_1x_2 - 2x_1^2 + 2x_1^4 + 2x_1^2x_2^2 + 2x_1x_2 - 2x_2^2 + 2x_1^2x_2^2 + 2x_2^4$$

$$\dot{V} = -2x_1^2 + 2x_1^4 + 2x_1^2x_2^2 - 2x_2^2 + 2x_1^2x_2^2 + 2x_2^4$$

$$\dot{V} = -2x_1^2 - 2x_2^2 + 2x_1^4 + 4x_1^2x_2^2 + 2x_2^4$$

$$\dot{V} = -2x_1^2 - 2x_2^2 + 2(x_1^2 + x_2^2)^2$$

$$\dot{V} = -2(x_1^2 + x_2^2) + 2(x_1^2 + x_2^2)^2$$

For this expression to be negative, $(x_1^2 + x_2^2)^2 < (x_1^2 + x_2^2)$**, which means:** $(x_1^2 + x_2^2) < 1$**. The the region of attraction where the system is AS is:** $B(0, 1)$ **where** B **is open.**

$$iii)\ \dot{x}_1 = x_2(1 - x_1^2)$$
$$\dot{x}_2 = -(x_1 + x_2)(1 - x_1^2)$$

$$V = x_1^2 + x_2^2$$

$$\dot{V} = 2x_1\dot{x_1} + 2x_2\dot{x_2}$$

$$\dot{V} = 2x_1(x_2(1 - x_1^2)) + 2x_2(-(x_1 + x_2)(1 - x_1^2))$$

$$\dot{V} = 2x_1x_2 - 2x_1^3x_2 - 2x_2(x_1 + x_2)(1 - x_1^2)$$

$$\dot{V} = 2x_1x_2 - 2x_1^3x_2 - 2x_2(x_1 - x_1^3 + x_2 - x_1^2x_2)$$

$$\dot{V} = 2x_1x_2 - 2x_1^3x_2 - 2x_1x_2 + 2x_1^3x_2 - 2x_2^2 + 2x_1^2x_2^2$$

$$\dot{V} = -2x_2^2(1 - x_1^2)$$

Since we need $(1 - x_1^2) > 0$**, our region where the system is AS is all** x **where** $|x_1| < 1$

$$iiii)\ \dot{x}_1 = -x_1 - x_2$$
$$\dot{x}_2 = 2x_1 - x_2^3$$

$$V = x_1^2 + x_2^2$$

$$\dot{V} = 2x_1\dot{x_1} + 2x_2\dot{x_2}$$

$$\dot{V} = 2x_1(-x_1 - x_2) + 2x_2(2x_1 - x_2^3)$$

$$\dot{V} = -2x_1^2 - 2x_1x_2 + 4x_1x_2 - 2x_2^4$$

$$\dot{V} = 2(-x_1^2 + x_1x_2 - x_2^4)$$

Since this expression will always be negative / negative definite (with the exception of the origin) the system is GAS.

3. This problem is very similar to Homework 6. The primary difference is in the Q and R matrices used in forming the observer gain matrix.

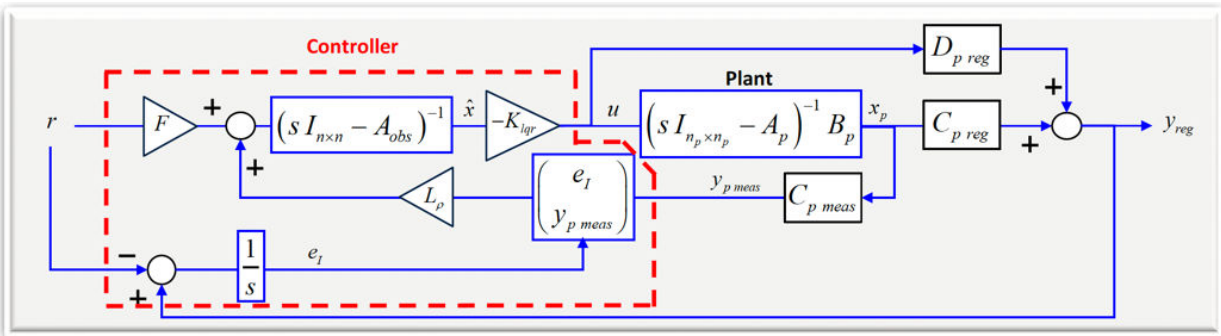
Design an OBLTR controller implementing the RSLQR controller from homework 3. The control law from homework 3 will be implemented with estimated state feedback: $u = -K_{lqr}\hat{x}$, where $\hat{x} = [\hat{e}_I \quad \hat{\alpha} \quad \hat{q}]^T$. Analyze your design by including 2^{nd} order $11Hz$ actuator model $(\omega_n = 2\pi * 11\,, \zeta = 0.707)$ in the plant analysis model. Do not use the actuator model in the design of the RSLQR or the observer.

```
load('HW5data.mat')

% constants
V = 886.78;
Md = -104.8346;
Ma = 47.7109;
Za = -1.3046*V;
Zd = -0.2142*V;

omega = 11*2*pi;
zeta = .707;

dd=0.:.001:2*pi;
xx1=cos(dd)-1;yy1=sin(dd);

% Plant Matricies without the actuator
Ap = [Za/V 1;
       Ma 0];
Bp = [Zd/V;
       Md];
Cp_reg = [Za 0];
Cp_meas = [0 1];
Dp_reg = Zd;

% Wiggle System
Aw = [zeros(1,1) Cp_reg;
     zeros(2,1) Ap];
```

```
Bw = [Dp_reg;
      Bp];
```

The observer is: $\dot{\hat{x}} = \left(\widetilde{A} - \widetilde{B}K_{lqr} - L_v C\right)\hat{x} + L_v y_{meas} + Fr$, where $r = A_{z_{cmd}}$ and $y_{meas} = [e_I \quad q]^T$.

This requires us to form the integral error $e_I = \int (y_{reg} - r) = \int (A_z - A_{zcmd})$ and make it available to the observer as a measurement. This method adds an additional integrator into the overall control architecture, and is shown in the following block diagram:



Where $A_{obs} = \widetilde{A} - \widetilde{B}K_{lqr} - L_v C$, $y_{reg} = A_z$, and $y_{p_{meas}} = q$.

The observer gain matrix L_v is designed using

$Q_v = Q_0 + c_0 \dfrac{(v+1)}{v} \bar{B}\bar{B}^T, R = \dfrac{v}{(v+1)} R_0$, and letting $v \to 0$ The constant c_0 is used to scale $\bar{B}\bar{B}^T$. The matrix $\bar{B}_{3\times 2} = \left[\widetilde{B}_{3\times 1} \quad B_{2_{3\times 1}}\right]$ above is to be designed to place a left-half-plane transmission zero in the observer state space model at $z_i = -10$. The column $B2$ is added with the $\widetilde{B}$ column as shown in the lecture notes. Use the Matlab script `square_system_inputs_pp` (provided on Blackboard) to compute $\bar{B}$ as outlined below. The observer design matrices are:

```
Aw_lqg = Aw;
Bw_lqg = Bw;
Cw_lqg = [ 1 0 0;
           0 0 1];
Dw_lqg = [ 0; 0];
```

and are placed in a state space object (sys_1) as follows:

```
sys_1 = ss(Aw_lqg, Bw_lqg, Cw_lqg, Dw_lqg);
```

The function `square_system_inputs_pp` takes the state space structure sys_1 and creates a new state space structure, sys_2, that has a transmission zero at $z_i = -10$.

The B-matrix in the structure sys_2 is $\overline{B}$ to be added into Q_v.

```
zz = [ -10 ];
sys_2 = square_system_inputs_pp(sys_1,zz);
disp(['Bbar Matrix z = ' num2str(zz) ])
```

```
Bbar Matrix z = -10
```

```
sys_2.b
```

```
ans = 3×2
 -189.9483  104.8211
   -0.2142   -3.4926
 -104.8346 -189.9238
```

```
tzero(sys_2)
```

```
ans = -10.0000
```

The tzero command above in Matlab will compute the transmission zeros showing that the function `square_system_inputs_pp` worked properly.

The $Q_v = Q_0 + c_0 \dfrac{(v+1)}{v} \overline{B}\overline{B}^T$ is then formed as

When $v \to 0$, $L_v \to \infty$. Care must be used to prevent the observer gain matrix L_v from becoming too large or sensor noise will be amplified to undesirable levels. Use

Design observers for $v = [100 \quad 0.3162 \quad 0.001] = logspace(2, -3, 3)$.

```
Q0 = diag([ 0.001 0.0014 0.005 ]);
R0 = 1000*diag([ 0.025^2 0.001^2]);
C0 = 0.00001
```

```
C0 = 1.0000e-05
```

```
vee = logspace(2,-3,3)
```

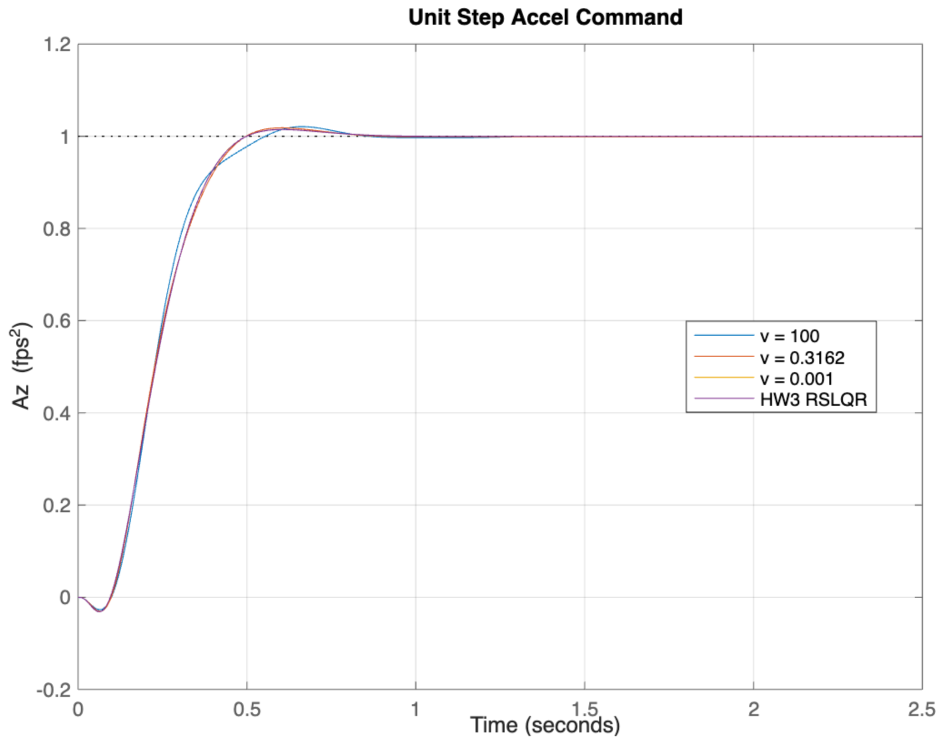
```
vee = 1×3
  100.0000    0.3162    0.0010
```

```
sys_u_cell = cell(1, 3); % Preallocate cell array for sys_u
sys_cl_cell = cell(1, 3); % Preallocate cell array for sys_cl
sys_controls_cell = cell(3,6);

for i = 1:3

    Qv = Q0 + C0*((vee(i)+1)/vee(i))*(sys_2.b*sys_2.B');
    Rv = (vee(i)/(vee(i)+1))*R0;
    Lv = transpose(lqr(Aw_lqg',Cw_lqg',Qv,Rv));

    % Plant Matricies with the actuator
    Ap = [Za/V 1    Zd/V        0;
           Ma  0     Md         0;
           0   0     0          1;
           0   0 -omega^2 -2*zeta*omega];
    Bp = [0;
          0;
          0
          omega^2];
    Cp = [Za  0 Zd  0;
           1  0  0  0;
           0  1  0  0;
           0  0  1  0;
           0  0  0  1];
    Dp = [0;
          0;
          0;
          0;
          0;];

    % populate the controller matrices
    Ac = [0               zeros(1,3);
         Lv(:,1)  Aw - Bw*rslqr.Kx - Lv*[1 0 0; 0 Cp_meas]];
    Bc1 = [1 0 0 0 0; zeros(3,2) Lv(:,2) zeros(3,2)];
    Bc2 = [-1; -1;zeros(2,1)];
    Cc = [0 -rslqr.Kx];
    Dc1 = [0 0 0 0 0];
    Dc2 = 0;

    Zi = inv(eye(1) - Dc1*Dp);
    Acl = [     Ap + Bp*Zi*Dc1*Cp                Bp*Zi*Cc;
             Bc1*(eye(5) + Dp*Zi*Dc1)*Cp    Ac + Bc1*Dp*Zi*Cc];
    Bcl = [    Bp*Zi*Dc2;
           Bc2 + Bc1*Dp*Zi*Dc2];
    Ccl = [(eye(5) + Dp*Zi*Dc1)*Cp Dp*Zi*Cc];
    Dcl = Dp*Zi*Dc2;
    sys_cl = ss(Acl,Bcl,Ccl,Dcl);
```

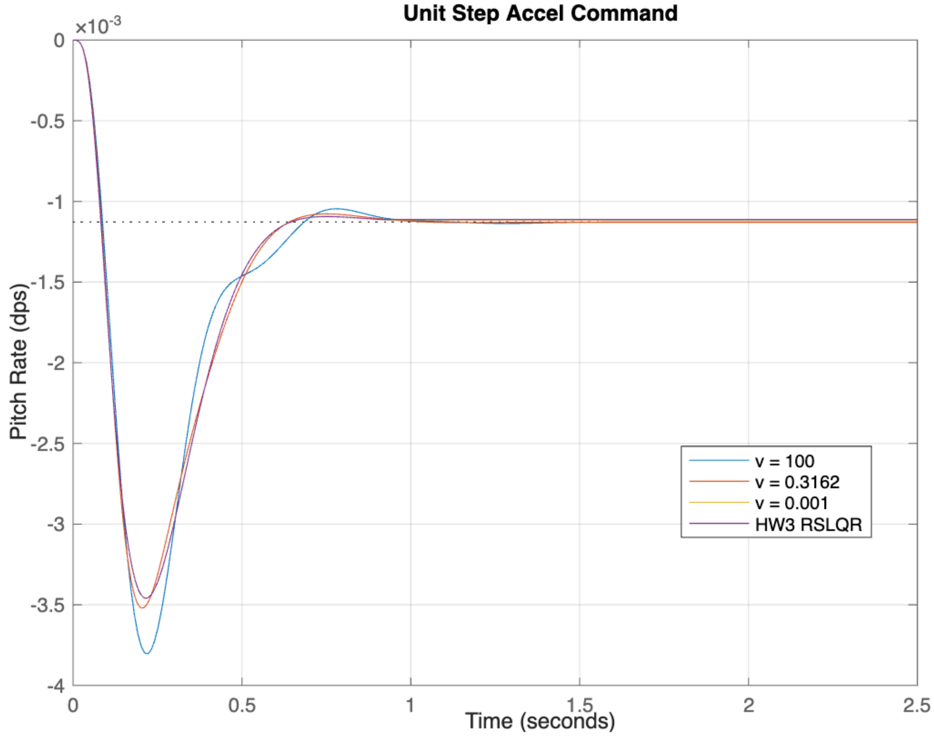
```
    % at plant input
    Alu = [Ap     zeros(4,4);
           Bc1*Cp     Ac];
    Blu = [  Bp;
           zeros(4,1)];
    Clu = -[Dc1*Cp Cc];
    Dlu = 0;
    sys_u = ss(Alu,Blu,Clu,Dlu);

    sys_cl_cell{i} = sys_cl;
    sys_u_cell{i} = sys_u;
    sys_controls_cell{i,1} = Ac;
    sys_controls_cell{i,2} = Bc1;
    sys_controls_cell{i,3} = Bc2;
    sys_controls_cell{i,4} = Cc;
    sys_controls_cell{i,5} = Dc1;
    sys_controls_cell{i,6} = Dc2;
end
```

Simulate using a constant unit step command. Compare these LQG/LTR designs with the RSLQR state feedback controller. (Include the RSLQR and all three designs on each plot).

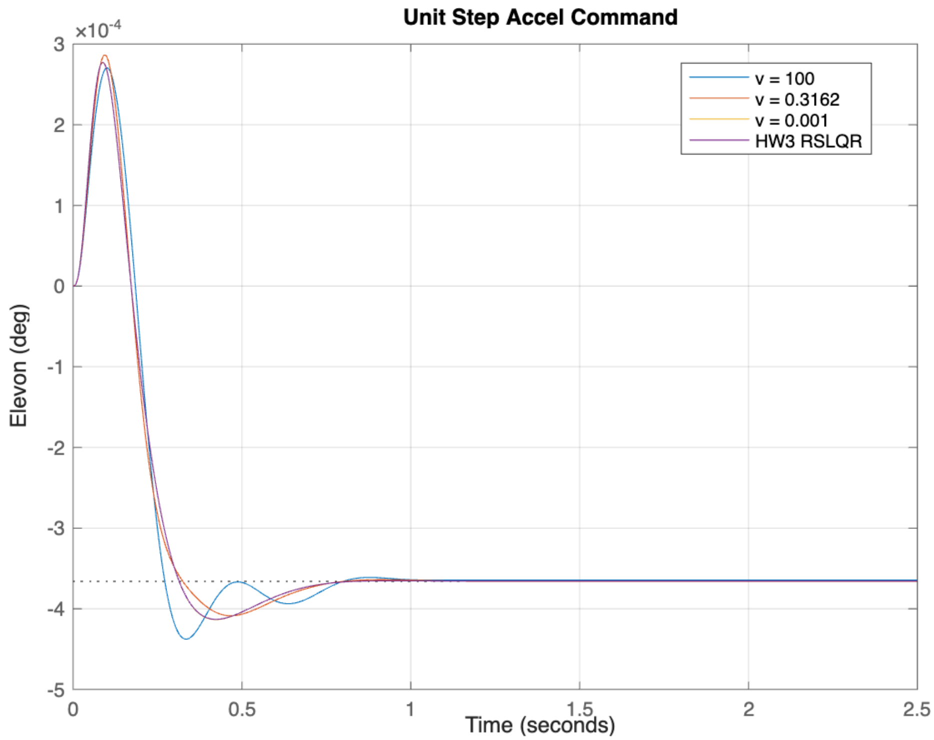
```
% from HW3
sys_rslqr_cl = ss(rslqr.Acl,rslqr.Bcl,rslqr.Ccl,rslqr.Dcl);
sys_rslqr_u = ss(rslqr.Alu,rslqr.Blu,rslqr.Clu,rslqr.Dlu);


figure;
step(sys_cl_cell{1}(1,1)); hold on,
step(sys_cl_cell{2}(1,1));
step(sys_cl_cell{3}(1,1));
step(sys_rslqr_cl(1,1)); hold off,
xlim([0 2.5]), grid on;
legend('v = 100','v = 0.3162','v = 0.001','HW3 RSLQR', 'Location', 'best')
ylabel('Az (fps^2)'), title('Unit Step Accel Command')
```



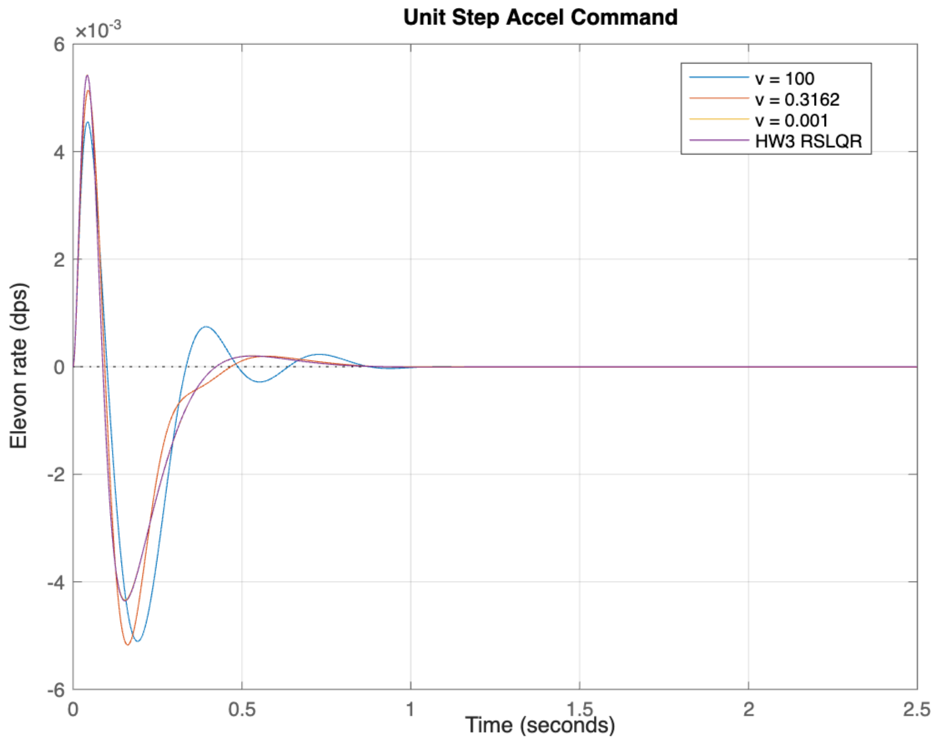

```
figure;
step(sys_cl_cell{1}(3,1)); hold on,
step(sys_cl_cell{2}(3,1));
step(sys_cl_cell{3}(3,1));
step(sys_rslqr_cl(3,1)); hold off,
xlim([0 2.5]), grid on;
legend('v = 100','v = 0.3162','v = 0.001','HW3 RSLQR', 'Location', 'best')
ylabel('Pitch Rate (dps)'), title('Unit Step Accel Command')
```




```
figure;
step(sys_cl_cell{1}(4,1)); hold on,
step(sys_cl_cell{2}(4,1));
step(sys_cl_cell{3}(4,1));
step(sys_rslqr_cl(4,1)); hold off,
xlim([0 2.5]), grid on;
legend('v = 100','v = 0.3162','v = 0.001','HW3 RSLQR', 'Location', 'best')
ylabel('Elevon (deg)'), title('Unit Step Accel Command')
```



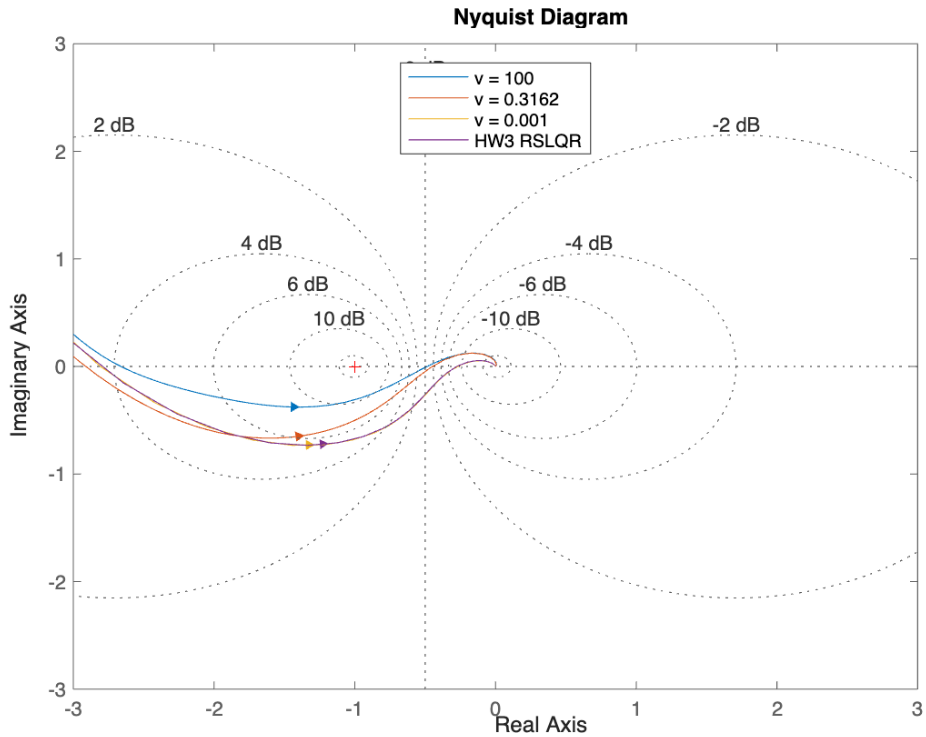

```
figure;
step(sys_cl_cell{1}(5,1)); hold on,
step(sys_cl_cell{2}(5,1));
step(sys_cl_cell{3}(5,1));
step(sys_rslqr_cl(5,1)); hold off,
xlim([0 2.5]), grid on;
legend('v = 100','v = 0.3162','v = 0.001','HW3 RSLQR', 'Location', 'best')
ylabel('Elevon rate (dps)'), title('Unit Step Accel Command')
```



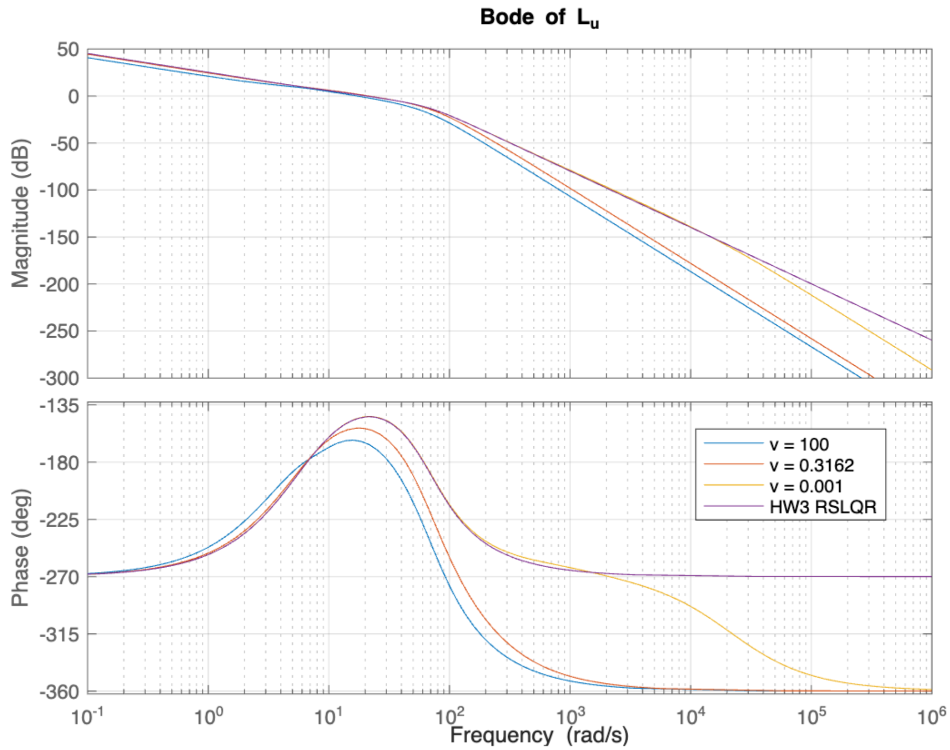
Analyze each controller in the frequency domain using a plant model that contains the actuator.

Plot Nyquist and Bode for L_u, $\underline{\sigma}(I + L_u)$, $\underline{\sigma}\left(I + L_u^{-1}\right)$, $e_{A_z}/r = S$, $A_z/r = T$ **and compute singular value margins at the plant input for each. Plot the noise-to-control and noise-to-control rate frequency responses.**

```
figure;
n = nyquistplot(sys_u_cell{1,1});
hold on;
n = nyquistplot(sys_u_cell{1,2});
n = nyquistplot(sys_u_cell{1,3});
n = nyquistplot(sys_rslqr_u);
hold off;
setoptions(n,'ShowfullContour','off'), grid on
legend('v = 100','v = 0.3162','v = 0.001','HW3 RSLQR', 'Location', 'best')
xlim([-3 3]),ylim([-3 3]),
```

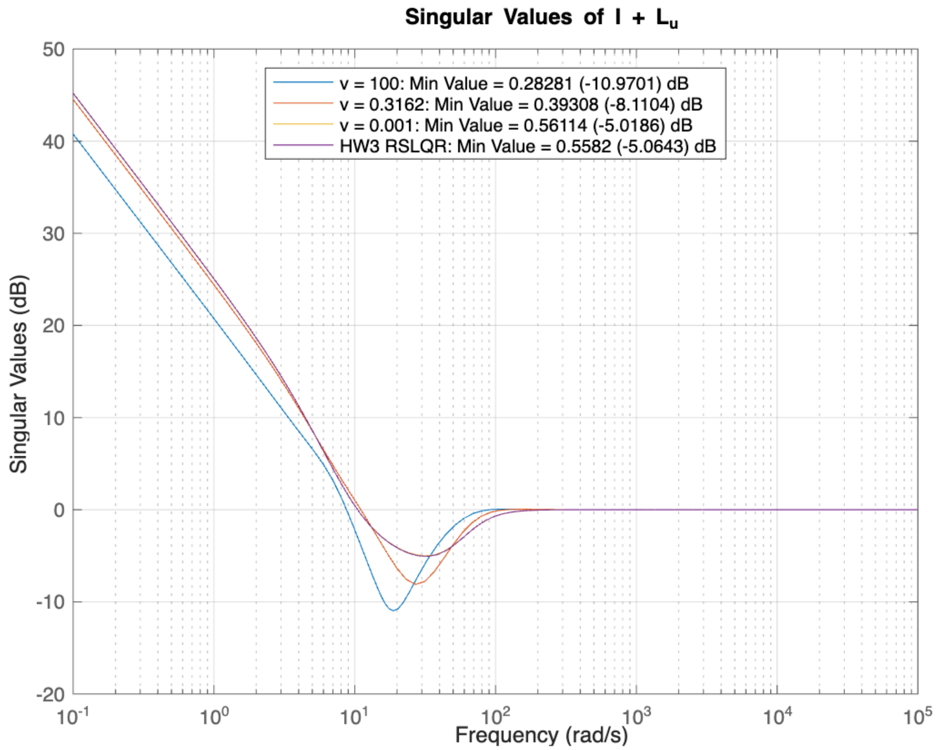



```
figure;
bode(sys_u_cell{1,1});
hold on;
bode(sys_u_cell{1,2});
bode(sys_u_cell{1,3});
bode(sys_rslqr_u);
hold off; grid on; title('Bode of L_{u}')
legend('v = 100','v = 0.3162','v = 0.001','HW3 RSLQR', 'Location', 'best')
```

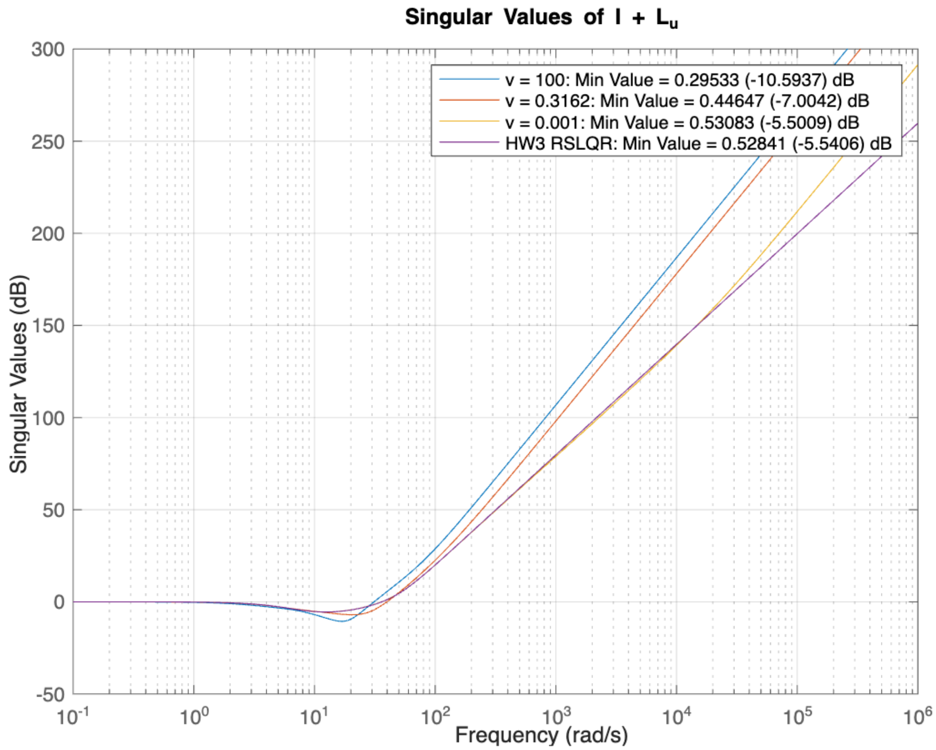



```
[a,b] =
ss2tf(sys_u_cell{1,1}.A,sys_u_cell{1,1}.B,sys_u_cell{1,1}.C,sys_u_cell{1,1}.
D);
c = b + a;
sys_IL_10 = tf(c,b);
alpha_0_10 = min(sigma(sys_IL_10));

sys_ILi_10 = tf(c,a);
beta_0_10 = (min(sigma(sys_ILi_10)));

[a,b] =
ss2tf(sys_u_cell{1,2}.A,sys_u_cell{1,2}.B,sys_u_cell{1,2}.C,sys_u_cell{1,2}.
D);
c = b + a;
sys_IL_4 = tf(c,b);
alpha_0_4 = min(sigma(sys_IL_4));

sys_ILi_4 = tf(c,a);
beta_0_4 = (min(sigma(sys_ILi_4)));

[a,b] =
ss2tf(sys_u_cell{1,3}.A,sys_u_cell{1,3}.B,sys_u_cell{1,3}.C,sys_u_cell{1,3}.
D);
c = b + a;
sys_IL_2 = tf(c,b);
```

```
alpha_0_2 = min(sigma(sys_IL_2));

sys_ILi_2 = tf(c,a);
beta_0_2 = (min(sigma(sys_ILi_2)));

[a,b] = ss2tf(rslqr.Alu,rslqr.Blu,rslqr.Clu,rslqr.Dlu);
c = b + a;
sys_IL_rslqr = tf([c 0],[b 0]);
alpha_0_rslqr = min(sigma(sys_IL_rslqr));

sys_ILi_rslqr = tf(c,a);
beta_0_rslqr = (min(sigma(sys_ILi_rslqr)));

figure;
sigma(sys_IL_10);
hold on;
sigma(sys_IL_4);
sigma(sys_IL_2);
sigma(sys_IL_rslqr);
legend([ 'v = 100: Min Value = ',num2str(alpha_0_10), ' (',
num2str(20*log10(alpha_0_10)),') dB'],...
        ['v = 0.3162: Min Value = ',num2str(alpha_0_4), ' (',
num2str(20*log10(alpha_0_4)),') dB'],...
        ['v = 0.001: Min Value = ',num2str(alpha_0_2), ' (',
num2str(20*log10(alpha_0_2)),') dB'],...
       ['HW3 RSLQR: Min Value = ',num2str(alpha_0_rslqr), ' (',
num2str(20*log10(alpha_0_rslqr)),') dB'],'Location','best')
grid on, title('Singular Values of I + L_u')
hold off;
```




```
figure;
sigma(sys_ILi_10);
hold on
sigma(sys_ILi_4);
sigma(sys_ILi_2);
sigma(sys_ILi_rslqr);
legend([ 'v = 100: Min Value = ',num2str(beta_0_10), ' (',
num2str(20*log10(beta_0_10)),') dB'],...
        ['v = 0.3162: Min Value = ',num2str(beta_0_4), ' (',
num2str(20*log10(beta_0_4)),') dB'],...
        ['v = 0.001: Min Value = ',num2str(beta_0_2), ' (',
num2str(20*log10(beta_0_2)),') dB'],...
       ['HW3 RSLQR: Min Value = ',num2str(beta_0_rslqr), ' (',
num2str(20*log10(beta_0_rslqr)),') dB'])
grid on, title('Singular Values of I + L_u')
hold off;
```



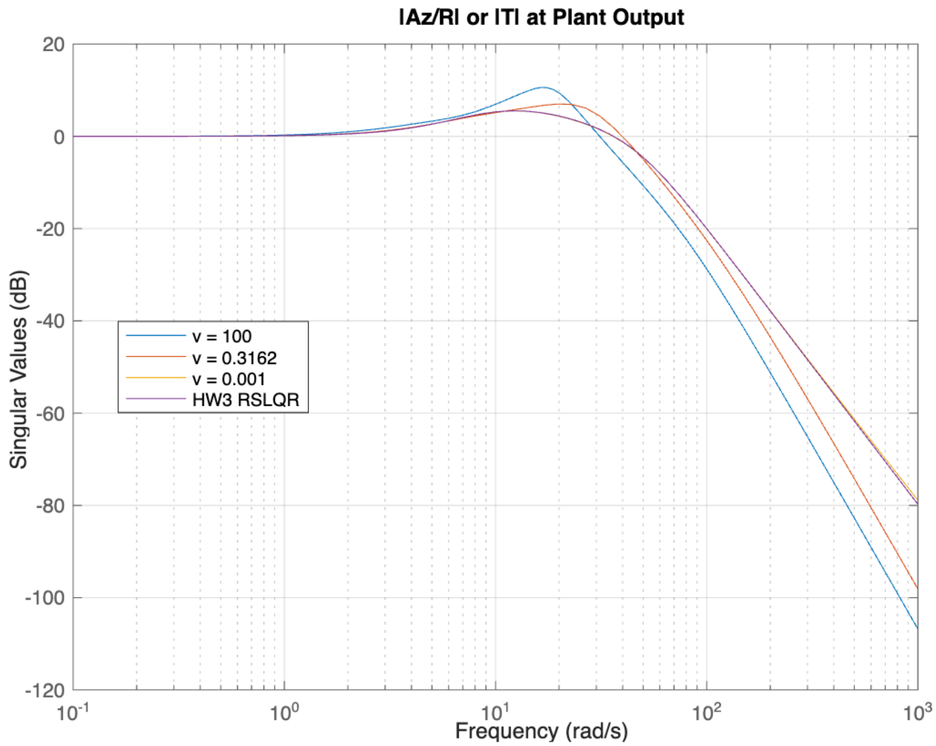

```
pm = 2*asin(alpha_0_2/2)*180/pi;
GM_l = 20*log10(1/(1 + alpha_0_2));
GM_u = 20*log10(1/(1 - alpha_0_2));
Gm_IL_2 = [GM_l GM_u];
pm_IL_2 = [-pm pm];


pm = 2*asin(alpha_0_rslqr/2)*180/pi;
GM_l = 20*log10(1/(1 + alpha_0_rslqr));
GM_u = 20*log10(1/(1 - alpha_0_rslqr));
Gm_IL_rslqr = [GM_l GM_u];
pm_IL_rslqr = [-pm pm];

GM_u = 20*log10(1 + beta_0_2);
GM_l = 20*log10(1 - beta_0_2);
pm = 2*asin(beta_0_2/2)*180/pi;
Gm_ILi_2 = [GM_l GM_u];
pm_ILi_2 = [-pm pm];

GM_u = 20*log10(1 + beta_0_rslqr);
GM_l = 20*log10(1 - beta_0_rslqr);
pm = 2*asin(beta_0_rslqr/2)*180/pi;
Gm_ILi_rslqr = [GM_l GM_u];
pm_ILi_rslqr = [-pm pm];
```

```
pm = 2*asin(alpha_0_4/2)*180/pi;
GM_l = 20*log10(1/(1 + alpha_0_4));
GM_u = 20*log10(1/(1 - alpha_0_4));
Gm_IL_4 = [GM_l GM_u];
pm_IL_4 = [-pm pm];


pm = 2*asin(alpha_0_10/2)*180/pi;
GM_l = 20*log10(1/(1 + alpha_0_10));
GM_u = 20*log10(1/(1 - alpha_0_10));
Gm_IL_10 = [GM_l GM_u];
pm_IL_10 = [-pm pm];

GM_u = 20*log10(1 + beta_0_10);
GM_l = 20*log10(1 - beta_0_10);
pm = 2*asin(beta_0_10/2)*180/pi;
Gm_ILi_10 = [GM_l GM_u];
pm_ILi_10 = [-pm pm];

GM_u = 20*log10(1 + beta_0_4);
GM_l = 20*log10(1 - beta_0_4);
pm = 2*asin(beta_0_4/2)*180/pi;
Gm_ILi_4 = [GM_l GM_u];
pm_ILi_4 = [-pm pm];


Gm_sv_10 = [min(Gm_ILi_10(1),Gm_IL_10(1)),max(Gm_ILi_10(2),Gm_IL_10(2))];
pm_sv_10 = [min(pm_ILi_10(1),pm_IL_10(1)),max(pm_ILi_10(2),pm_IL_10(2))];
Gm_sv_4 = [min(Gm_ILi_4(1),Gm_IL_4(1)),max(Gm_ILi_4(2),Gm_IL_4(2))];
pm_sv_4 = [min(pm_ILi_4(1),pm_IL_4(1)),max(pm_ILi_4(2),pm_IL_4(2))];


Gm_sv_2 = [min(Gm_ILi_2(1),Gm_IL_2(1)),max(Gm_ILi_2(2),Gm_IL_2(2))];
pm_sv_2 = [min(pm_ILi_2(1),pm_IL_2(1)),max(pm_ILi_2(2),pm_IL_2(2))];

Gm_sv_rslqr =
[min(Gm_ILi_rslqr(1),Gm_IL_rslqr(1)),max(Gm_ILi_rslqr(2),Gm_IL_rslqr(2))];
pm_sv_rslqr =
[min(pm_ILi_rslqr(1),pm_IL_rslqr(1)),max(pm_ILi_rslqr(2),pm_IL_rslqr(2))];

text = ['v = 100$ SV Gain Margin:  [',num2str(Gm_sv_10(1)),',
',num2str(Gm_sv_10(2)),'] dB',...
         newline 'v = 100 SV Phase Margin: [',num2str(pm_sv_10(1)),',
',num2str(pm_sv_10(2)),'] degrees',...
         newline 'v = 0.3162 SV Gain Margin:  [',num2str(Gm_sv_4(1)),',
',num2str(Gm_sv_4(2)),' ] dB',...
         newline 'v = 0.3162 SV Phase Margin: [',num2str(pm_sv_4(1)),',
',num2str(pm_sv_4(2)),'] degrees',...
         newline 'v = 0.001 SV Gain Margin:  [',num2str(Gm_sv_2(1)),',
',num2str(Gm_sv_2(2)),'] dB',...
```

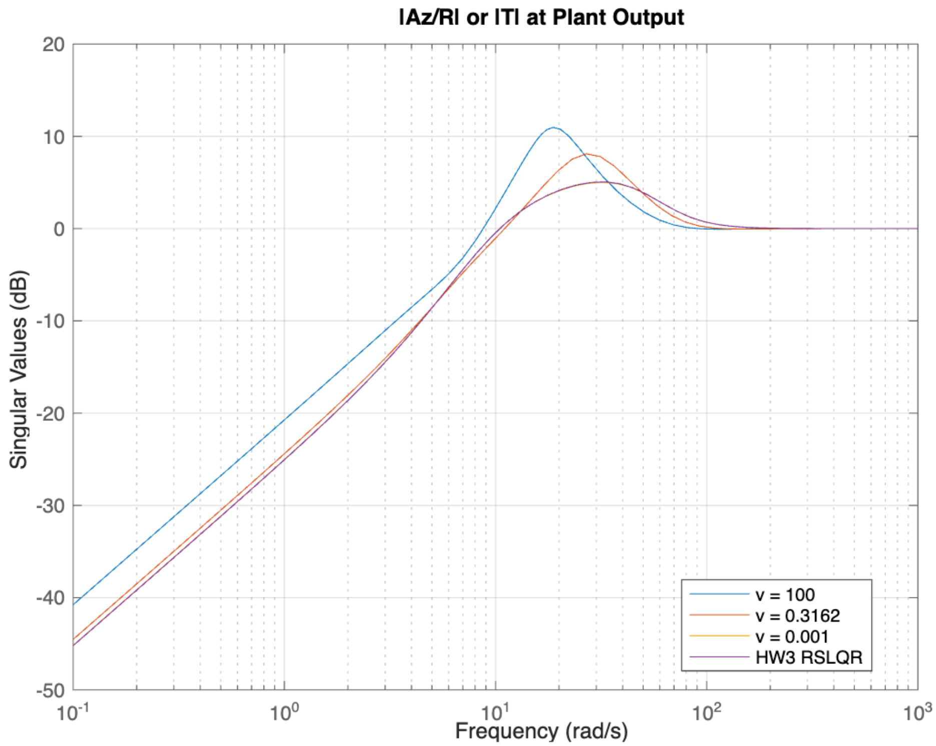
```
        newline 'v = 0.001 SV Phase Margin: [',num2str(pm_sv_2(1)),',
',num2str(pm_sv_2(2)),'] degrees',...
        newline 'HW3 RSLQR SV Gain Margin:  [',num2str(Gm_sv_rslqr(1)),',
',num2str(Gm_ILi_rslqr(2)),'] dB',...
        newline 'HW3 RSLQR SV Phase Margin: [',num2str(pm_sv_rslqr(1)),',
',num2str(pm_sv_rslqr(2)),'] degrees'];
disp(text);
```

```
v = 100$ SV Gain Margin:  [-3.0403, 2.8873] dB
v = 100 SV Phase Margin: [-16.9835, 16.9835] degrees
v = 0.3162 SV Gain Margin:  [-5.1371, 4.3374 ] dB
v = 0.3162 SV Phase Margin: [-25.798, 25.798] degrees
v = 0.001 SV Gain Margin:  [-6.5734, 7.1534] dB
v = 0.001 SV Phase Margin: [-32.5883, 32.5883] degrees
HW3 RSLQR SV Gain Margin:  [-6.5287, 3.6848] dB
HW3 RSLQR SV Phase Margin: [-32.4127, 32.4127] degrees
```

```
T_y_10 = sys_u_cell{1,1}(1,1)/(1+sys_u_cell{1,1}(1,1));
T_y_4 = sys_u_cell{1,2}(1,1)/(1+sys_u_cell{1,2}(1,1));
T_y_2 = sys_u_cell{1,3}(1,1)/(1+sys_u_cell{1,3}(1,1));
T_y_rslqr = sys_rslqr_u(1,1)/(1+sys_rslqr_u(1,1));
sigma(T_y_10), hold on,
sigma(T_y_4)
sigma(T_y_2),
sigma(T_y_rslqr), hold off
grid on, title('|Az/R| or |T| at Plant Output')
xlim([.1 1000]),legend('v = 100','v = 0.3162','v = 0.001','HW3 RSLQR',
'Location', 'best')
```

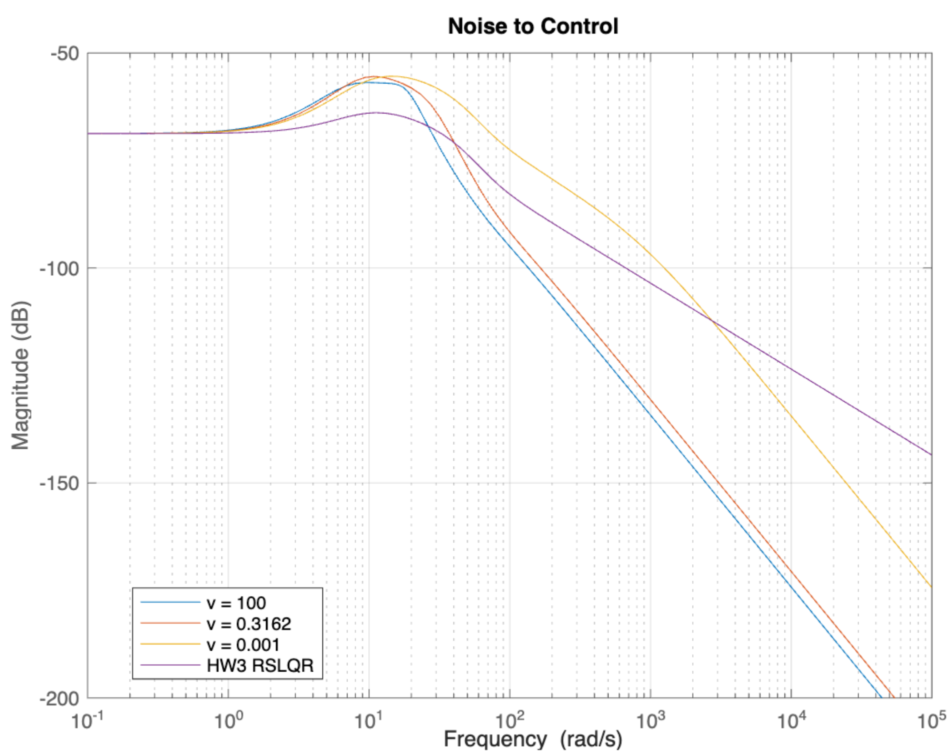



```
S_y_10 = 1/(1+sys_u_cell{1,1}(1,1));
S_y_4 = 1/(1+sys_u_cell{1,2}(1,1));
S_y_2 = 1/(1+sys_u_cell{1,3}(1,1));
S_y_rslqr = 1/(1+sys_rslqr_u(1,1));

figure;
sigma(S_y_10), hold on,
sigma(S_y_4)
sigma(S_y_2),
sigma(S_y_rslqr), hold off
grid on, title('|Az/R| or |T| at Plant Output')
xlim([.1 1000]), legend('v = 100','v = 0.3162','v = 0.001','HW3 RSLQR',
'Location', 'best')
```

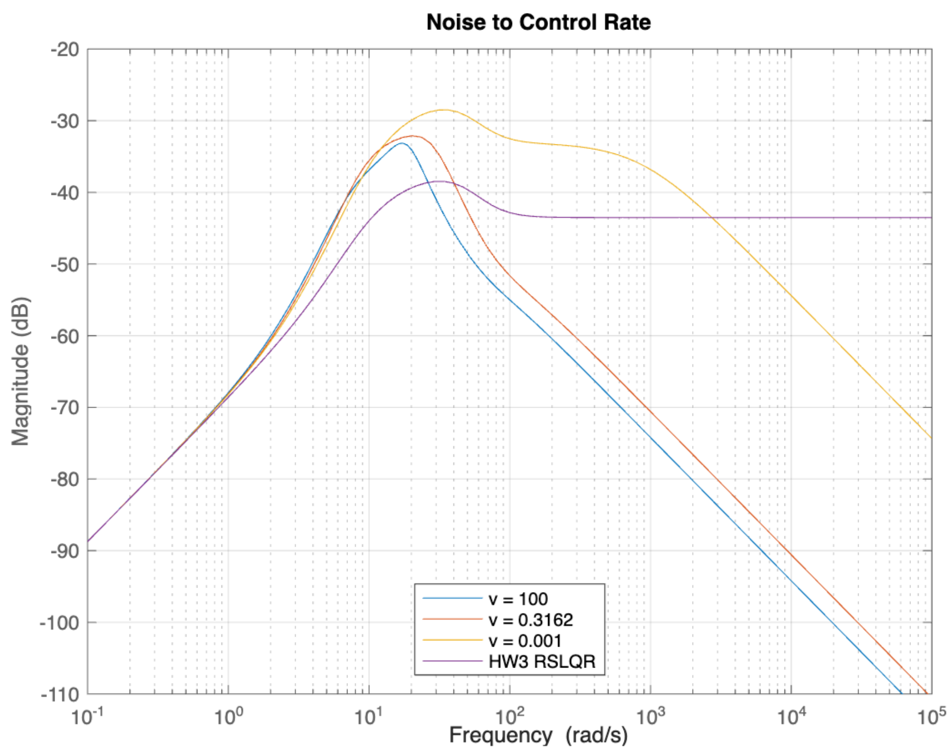



```
% Compute the noise-to-control TF
sys10.Bv = [ Bp*Zi*sys_controls_cell{1,5};

(sys_controls_cell{1,2}+sys_controls_cell{1,2}*Dp*Zi*sys_controls_cell{1,5})
];
sys10.Cv = [ Zi*sys_controls_cell{1,5}*Cp Zi*sys_controls_cell{1,4}];
sys10.Cvv = [ sys10.Cv ; sys10.Cv*sys_cl_cell{1,1}.A ];
sys10.Dv = Zi*sys_controls_cell{1,5};
sys10.Dvv = [ sys10.Dv; sys10.Cv*sys10.Bv];
sys_noise_10 = ss(sys_cl_cell{1,1}.A,sys10.Bv,sys10.Cvv,sys10.Dvv);


% Compute the noise-to-control TF
sys4.Bv = [ Bp*Zi*sys_controls_cell{2,5};

(sys_controls_cell{2,2}+sys_controls_cell{2,2}*Dp*Zi*sys_controls_cell{2,5})
];
sys4.Cv = [ Zi*sys_controls_cell{2,5}*Cp Zi*sys_controls_cell{2,4}];
sys4.Cvv = [ sys4.Cv ; sys4.Cv*sys_cl_cell{1,2}.A ];
sys4.Dv = Zi*sys_controls_cell{2,5};
sys4.Dvv = [ sys4.Dv; sys4.Cv*sys4.Bv];
sys_noise_4 = ss(sys_cl_cell{1,2}.A,sys4.Bv,sys4.Cvv,sys4.Dvv);

% Compute the noise-to-control TF
```

```
sys2.Bv = [ Bp*Zi*sys_controls_cell{3,5};

(sys_controls_cell{3,2}+sys_controls_cell{3,2}*Dp*Zi*sys_controls_cell{3,5})
];
sys2.Cv = [ Zi*sys_controls_cell{3,5}*Cp Zi*sys_controls_cell{3,4}];
sys2.Cvv = [ sys2.Cv ; sys2.Cv*sys_cl_cell{1,3}.A ];
sys2.Dv = Zi*sys_controls_cell{3,5};
sys2.Dvv = [ sys2.Dv; sys2.Cv*sys2.Bv];
sys_noise_2 = ss(sys_cl_cell{1,3}.A,sys2.Bv,sys2.Cvv,sys2.Dvv);


% Compute the noise-to-control TF
sys_noise_rslqr = ss(rslqr.Acl,rslqr.Bv,rslqr.Cvv,rslqr.Dvv);


figure;
bodemag(sys_noise_10(1,1))
hold on;
bodemag(sys_noise_4(1,1))
bodemag(sys_noise_2(1,1))
bodemag(sys_noise_rslqr(1,1))
grid on;
legend('v = 100','v = 0.3162','v = 0.001','HW3 RSLQR', 'Location', 'best')
title('Noise to Control')
hold off
```



```
figure;
bodemag(sys_noise_10(2,1))
hold on;
bodemag(sys_noise_4(2,1))
bodemag(sys_noise_2(2,1))
bodemag(sys_noise_rslqr(2,1))
grid on;
legend('v = 100','v = 0.3162','v = 0.001','HW3 RSLQR', 'Location', 'best')
title('Noise to Control Rate')
hold off
```



For $v = 10^{-3}$, List out your observer design matrices $(\widetilde{A}, C, Q, R, L_v)$. List the controller matrices $(A_c, B_{c1}, B_{c2}, C_c, D_{c1}, D_{c2})$ implementing the controller

$$\widetilde{A} = \begin{pmatrix} 0 & -1156.89 & 0 \\ 0 & -1.30 & 1 \\ 0 & 47.71 & 0 \end{pmatrix} \quad C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad Q = \begin{pmatrix} 471.14 & -3.25 & 0.05 \\ -3.25 & 0.12 & 6.86 \\ 0.05 & 6.86 & 471.09 \end{pmatrix}$$

$$R = \begin{pmatrix} 0.0006 & 0 \\ 0 & 0 \end{pmatrix} \quad L_v = \begin{pmatrix} 876.81 & -22.08 \\ -6.14 & 317.62 \\ -0.03 & 21716.12 \end{pmatrix}$$

$$A_c = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 876.81 & -875.54 & -1567.15 & -18.26 \\ -6.14 & 6.14 & -1.76 & -316.66 \\ -0.03 & 0.73 & -178.71 & -21738.40 \end{pmatrix} \qquad B_{c1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -22.09 & 0 & 0 \\ 0 & 0 & 317.62 & 0 & 0 \\ 0 & 0 & 21716.12 & 0 & 0 \end{pmatrix}$$

$$B_{c2} = \begin{pmatrix} -1 \\ -1 \\ 0 \\ 0 \end{pmatrix}$$

$$Cc = (0 - 0.006 \quad 2.15 \quad 0.21) \qquad D_{c1} = (0 \quad 0 \quad 0 \quad 0 \quad 0) \qquad D_{c2} = (0)$$