# ESE 547 Homework 6 /

# Seif Elkhashab / 03-22-2024 /

# Output Feedback Doyle and Stein LQG/LTR Design

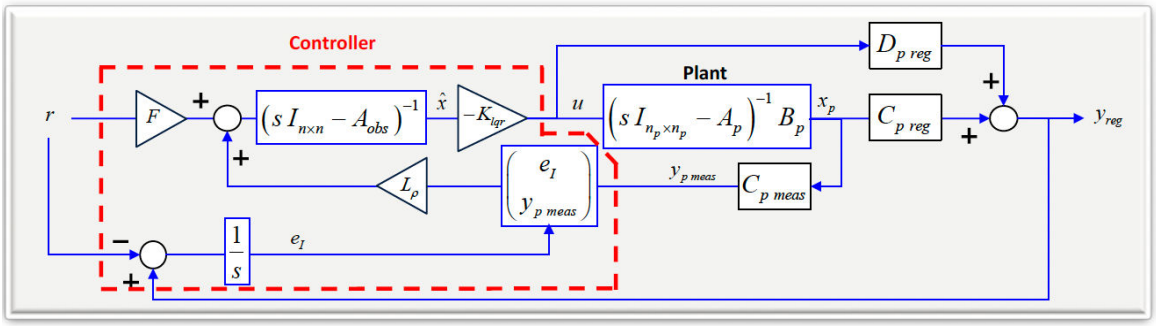
**Design a Doyle and Stein based LQG/LTR controller implementing the RSLQR controller from homework 3. The control law from homework 3 will be implemented with estimated state feedback: $\mathbf{u} = -\mathbf{K}_{\mathbf{lqr}}\hat{\mathbf{x}}$ , where $\hat{x} = [\hat{e_I} \quad \hat{\alpha} \quad \hat{q}]^T$**

**Analyze your design by including a $2^{nd}$ 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:** $\dot{\hat{x}} = \left(\tilde{A} - \tilde{B}K_{lqr} - L_pC\right)\hat{x} + L_\rho y_{meas} + Fr$ **, 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** $A_{obs} = \tilde{A} - \tilde{B}K_{lqr} - L_pC$ , $y_{reg} = A_z$, **and** $y_{p_{meas}} = q$ .

**The observer gain matrix** $L_p$ **is designed using** $Q = Q_0 + \frac{1}{\rho}\tilde{B}\tilde{B}^T, R = R_0$ **and letting** $\rho \to 0$. **When** $\rho \to 0, L_p \to \infty$ . **Care must be used to prevent the observer gain matrix** $L_p$ **from becoming too large or sensor noise will be amplified to undesirable levels. Use the following:**

$$Q_0 = diag([0.001 \quad 0.0014 \quad 0.005]);$$

$$R_0 = 1000 * diag([0.025^2 \quad 0.001^2]);$$

**Design observers for** $\rho = [10^{10} \quad 10^4 \quad 10^2]$.

```
rhos  = [10^10 10^4 10^2];
Q0 = diag([ 0.001 0.0014 0.005 ]);
R0 = 1000*diag([ 0.025^2 0.001^2]);

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
```

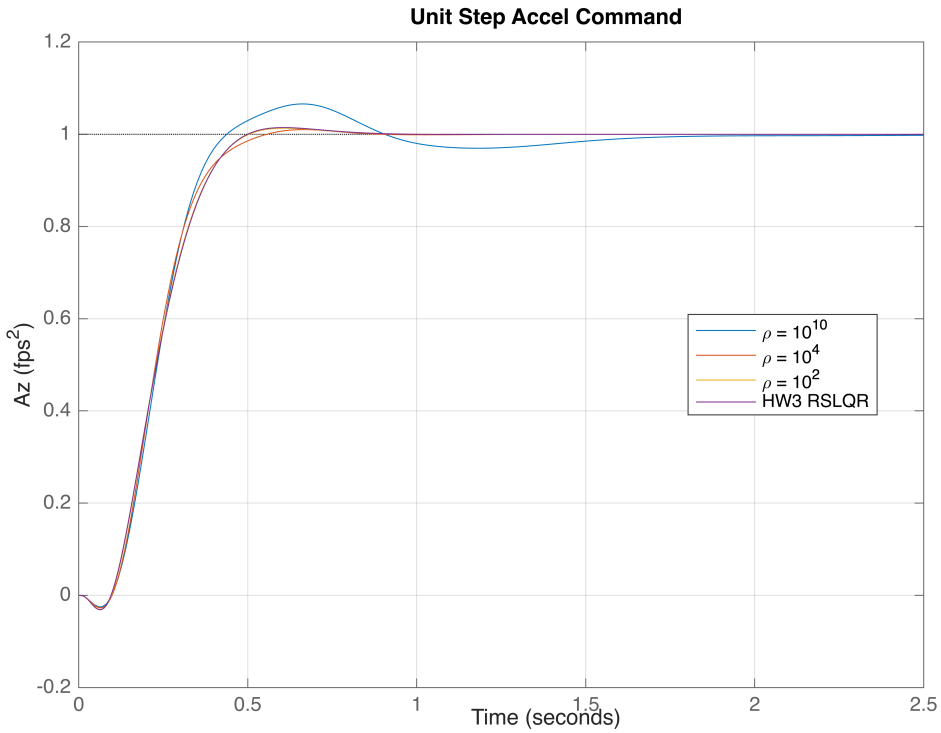
```
    Q = Q0 + (1/rhos(i))*(Bw*Bw');
    R = R0;
    Lp = transpose(lqr(Aw',[1 0 0; 0 Cp_meas]',Q,R));

    % 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);
         Lp(:,1)  Aw - Bw*rslqr.Kx - Lp*[1 0 0; 0 Cp_meas]];
    Bc1 = [1 0 0 0 0; zeros(3,2) Lp(:,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
```

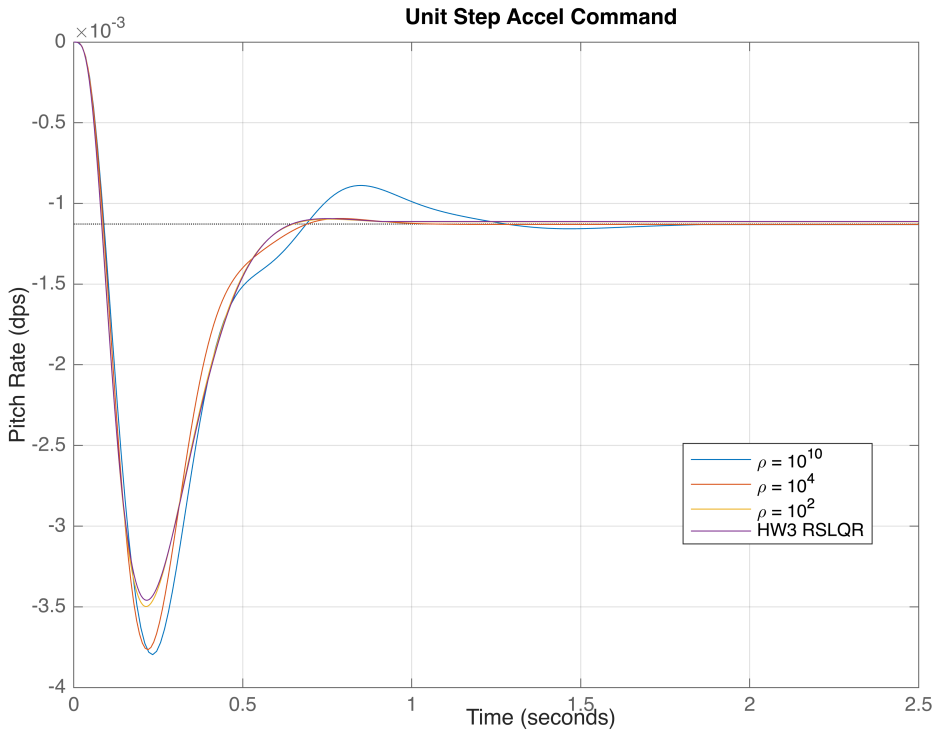
## Simulate using a constant unit step command. Compare these LQG/LTR designs with the RSLQR state feedback controller. (Include the RSLQR and all three designs on each plot).

## Turn in plots of the accel command and response, pitch rate, and elevon deflection and rate.

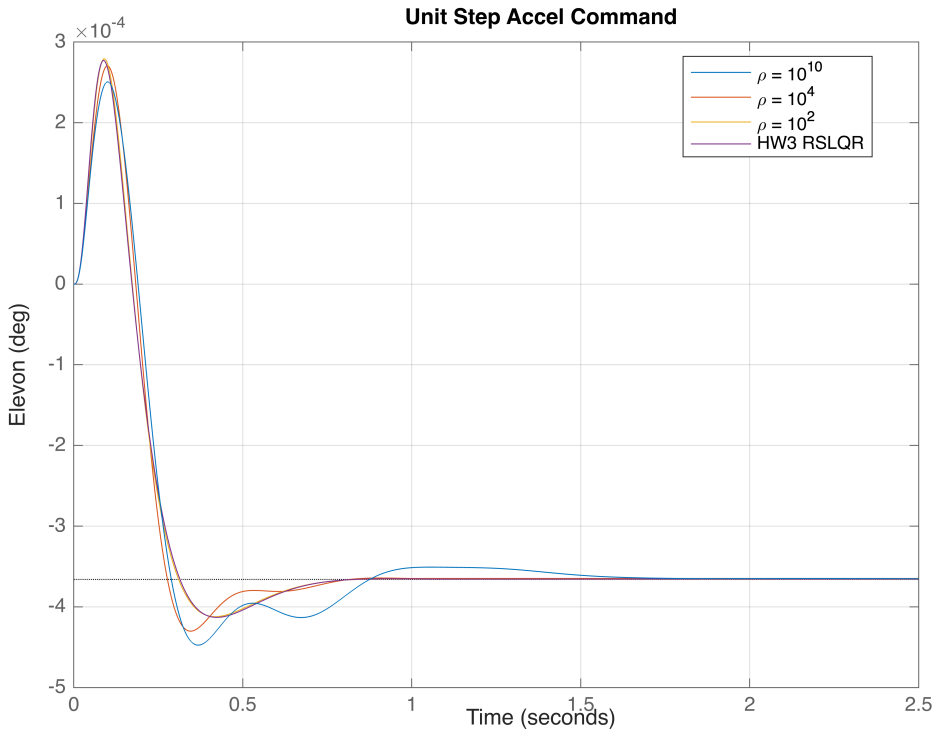
```
% from HW3
sys_rslqr_cl = ss(rslqr.Acl,rslqr.Bcl,rslqr.Ccl,rslqr.Dcl);
sys_rslqr_u = ss(rslqr.Alu,rslqr.Blu,rslqr.Clu,rslqr.Dlu);


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('\rho = 10^{10}','\rho = 10^{4}','\rho = 10^{2}','HW3 RSLQR',
'Location', 'best')
ylabel('Az (fps^2)'), title('Unit Step Accel Command')
```



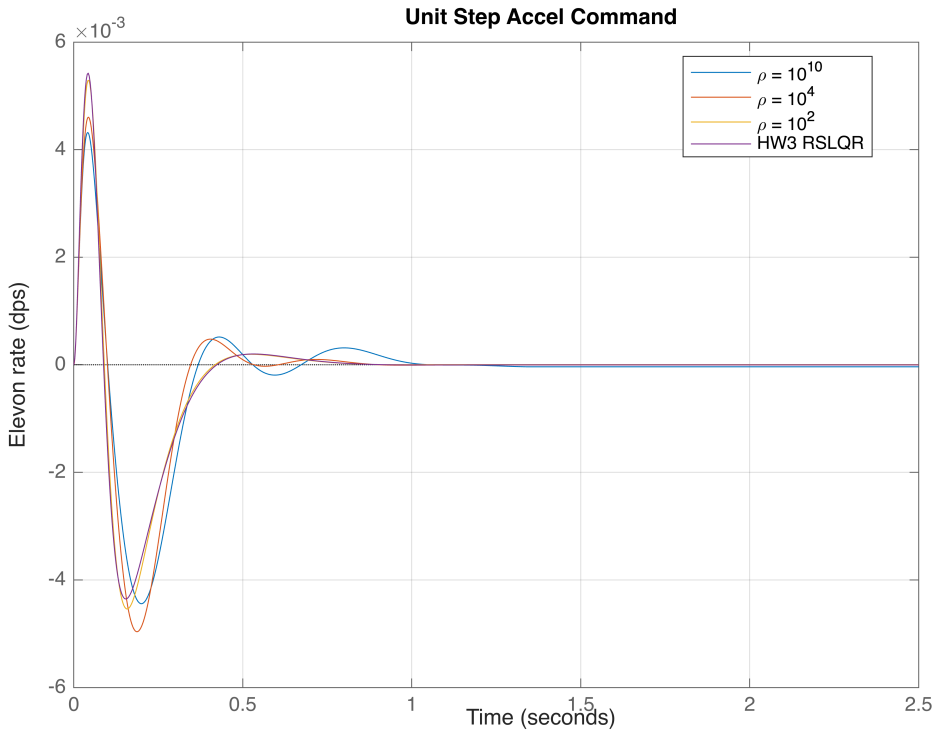

```
figure;
step(sys_cl_cell{1}(3,1)); hold on,
step(sys_cl_cell{2}(3,1));
step(sys_cl_cell{3}(3,1));
step(sys_rslqr_cl(3,1)); hold off,
xlim([0 2.5]), grid on;
legend('\rho = 10^{10}','\rho = 10^{4}','\rho = 10^{2}','HW3 RSLQR',
'Location', 'best')
ylabel('Pitch Rate (dps)'), title('Unit Step Accel 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('\rho = 10^{10}','\rho = 10^{4}','\rho = 10^{2}','HW3 RSLQR',
'Location', 'best')
ylabel('Elevon (deg)'), title('Unit Step Accel Command')
```



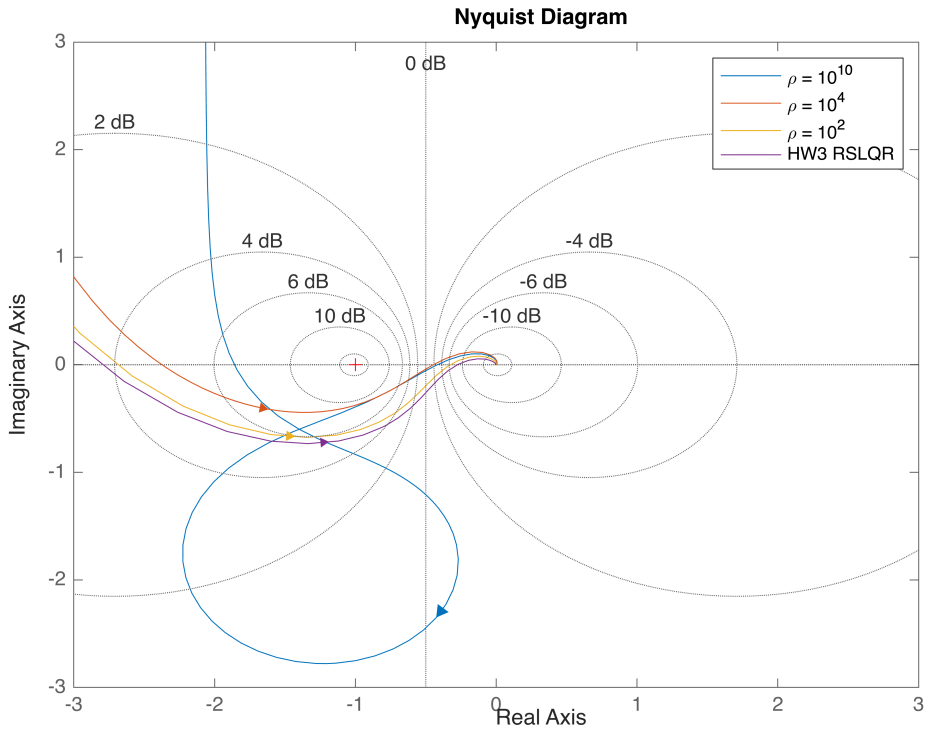

```
figure;
step(sys_cl_cell{1}(5,1)); hold on,
step(sys_cl_cell{2}(5,1));
step(sys_cl_cell{3}(5,1));
step(sys_rslqr_cl(5,1)); hold off,
xlim([0 2.5]), grid on;
legend('\rho = 10^{10}','\rho = 10^{4}','\rho = 10^{2}','HW3 RSLQR',
'Location', 'best')
ylabel('Elevon rate (dps)'), title('Unit Step Accel Command')
```



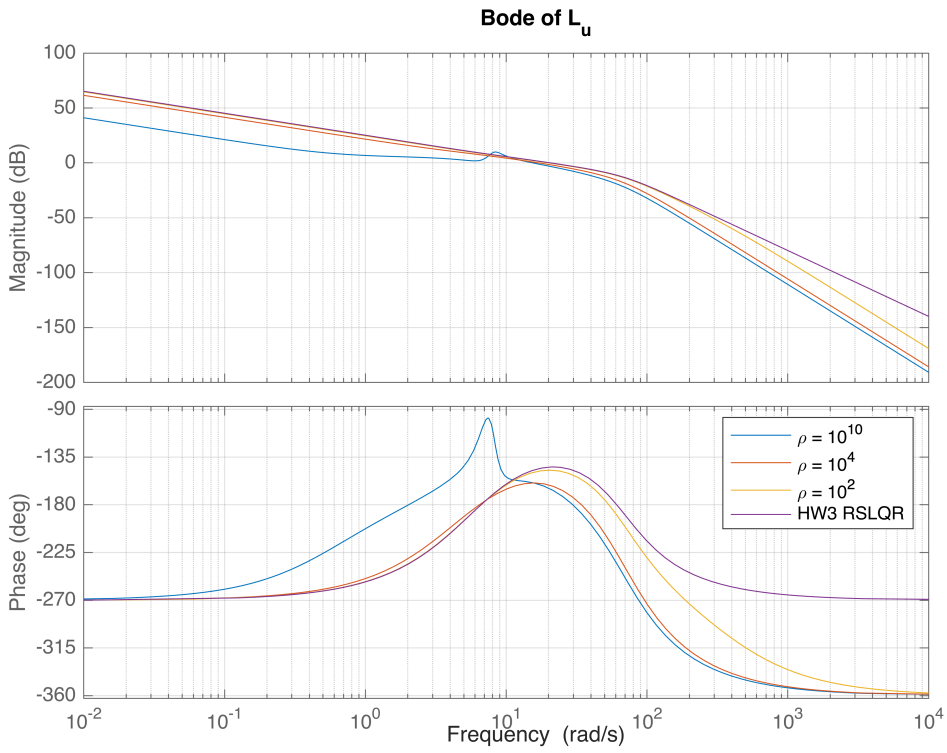

**Analyze each controller in the frequency domain using a plant model that contains the actuator.**

**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('\rho = 10^{10}','\rho = 10^{4}','\rho = 10^{2}','HW3 RSLQR')
xlim([-3 3]),ylim([-3 3]),
```

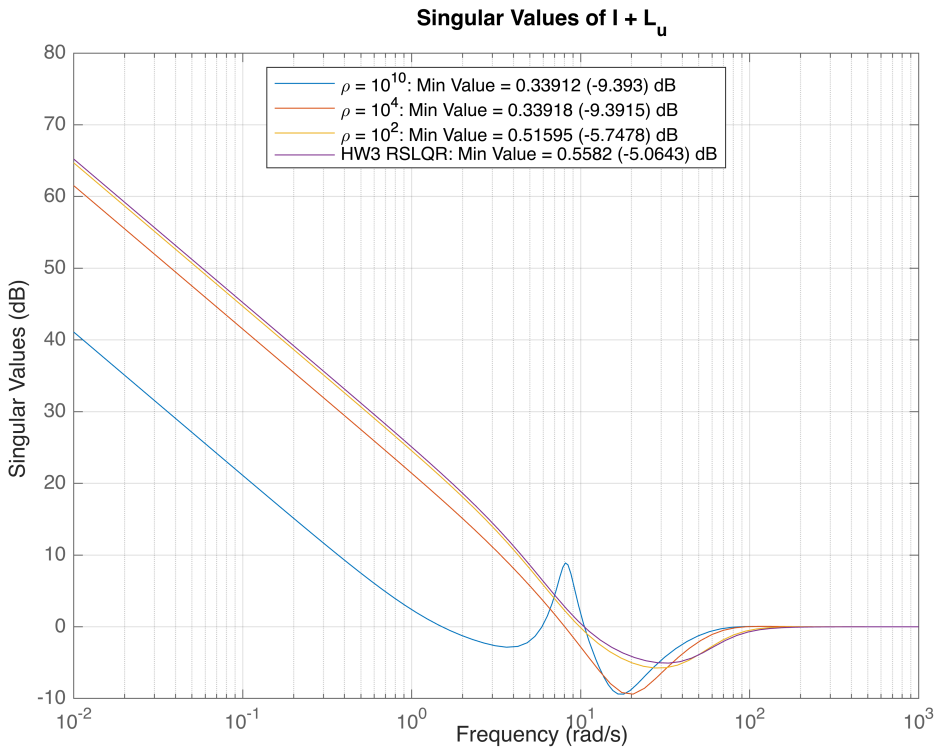



```
figure;
bode(sys_u_cell{1,1});
hold on;
bode(sys_u_cell{1,2});
bode(sys_u_cell{1,3});
bode(sys_rslqr_u);
hold off; grid on; title('Bode of L_{u}')
legend('\rho = 10^{10}','\rho = 10^{4}','\rho = 10^{2}','HW3 RSLQR')
```

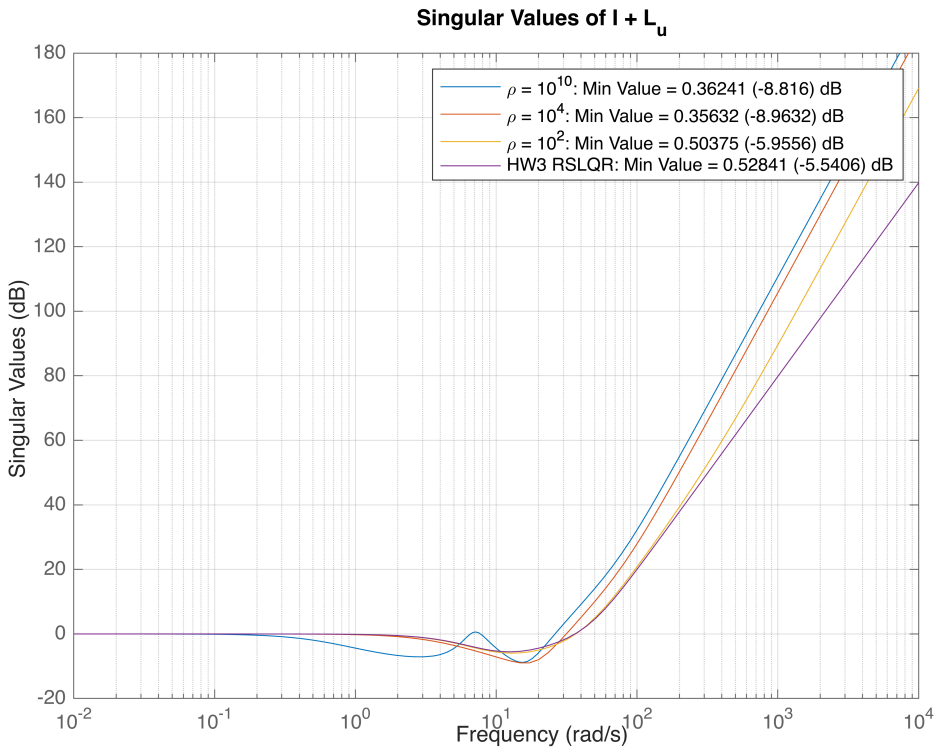



```
[a,b] =
ss2tf(sys_u_cell{1,1}.A,sys_u_cell{1,1}.B,sys_u_cell{1,1}.C,sys_u_cell{1,1}.
D);
c = b + a;
sys_IL_10 = tf(c,b);
alpha_0_10 = min(sigma(sys_IL_10));

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(['\rho = 10^{10}: Min Value = ',num2str(alpha_0_10), ' (',
num2str(20*log10(alpha_0_10)),') dB'],...
        ['\rho = 10^{4}: Min Value = ',num2str(alpha_0_4), ' (',
num2str(20*log10(alpha_0_4)),') dB'],...
        ['\rho = 10^{2}: 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(['\rho = 10^{10}: Min Value = ',num2str(beta_0_10), ' (',
num2str(20*log10(beta_0_10)),') dB'],...
        ['\rho = 10^{4}: Min Value = ',num2str(beta_0_4), ' (',
num2str(20*log10(beta_0_4)),') dB'],...
        ['\rho = 10^{2}: 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;
```



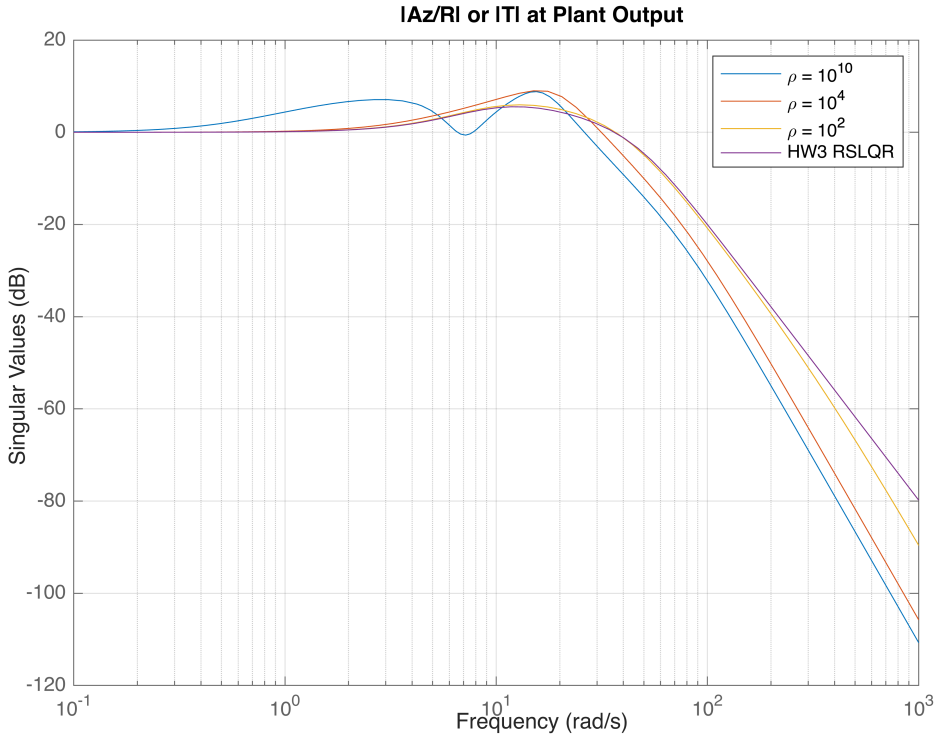

```
pm = 2*asin(alpha_0_2/2)*180/pi;
GM_l = 20*log10(1/(1 + alpha_0_2));
GM_u = 20*log10(1/(1 - alpha_0_2));
Gm_IL_2 = [GM_l GM_u];
pm_IL_2 = [-pm pm];


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 = ['p = 10^{10}$ SV Gain Margin:  [',num2str(Gm_sv_10(1)),',
',num2str(Gm_sv_10(2)),'] dB',...
        newline 'p = 10^{10} SV Phase Margin: [',num2str(pm_sv_10(1)),',
',num2str(pm_sv_10(2)),'] degrees',...
        newline 'p = 10^{4} SV Gain Margin:  [',num2str(Gm_sv_4(1)),',
',num2str(Gm_sv_4(2)),' ] dB',...
        newline 'p = 10^{4} SV Phase Margin: [',num2str(pm_sv_4(1)),',
',num2str(pm_sv_4(2)),'] degrees',...
        newline 'p = 10^{2} SV Gain Margin:  [',num2str(Gm_sv_2(1)),',
',num2str(Gm_sv_2(2)),'] dB',...
```

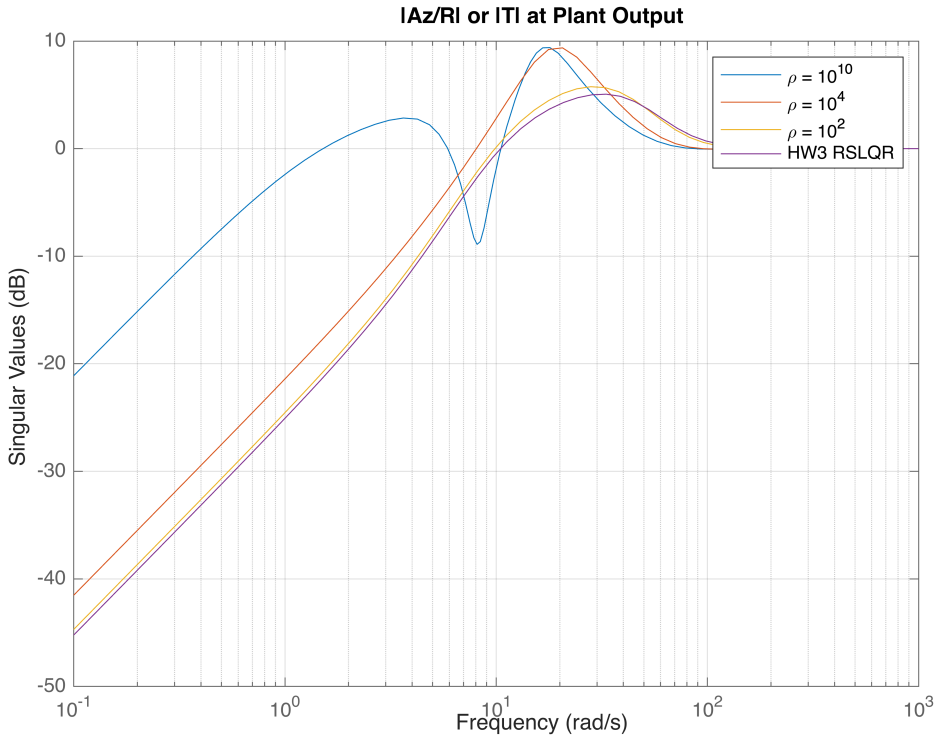
```
        newline 'p = 10^{2} SV Phase Margin: [',num2str(pm_sv_2(1)),',
',num2str(pm_sv_2(2)),'] degrees',...
        newline 'HW3 RSLQR SV Gain Margin:  [',num2str(Gm_sv_rslqr(1)),',
',num2str(Gm_ILi_rslqr(2)),'] dB',...
        newline 'HW3 RSLQR SV Phase Margin: [',num2str(pm_sv_rslqr(1)),',
',num2str(pm_sv_rslqr(2)),'] degrees'];
disp(text);
```

```
p = 10^{10}$ SV Gain Margin:  [-3.9091, 3.5975] dB
p = 10^{10} SV Phase Margin: [-20.8798, 20.8798] degrees
p = 10^{4} SV Gain Margin:  [-3.8266, 3.5983 ] dB
p = 10^{4} SV Phase Margin: [-20.5251, 20.5251] degrees
p = 10^{2} SV Gain Margin:  [-6.086, 6.3022] dB
p = 10^{2} SV Phase Margin: [-29.8999, 29.8999] 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('\rho = 10^{10}','\rho = 10^{4}','\rho =
10^{2}','HW3 RSLQR')
```

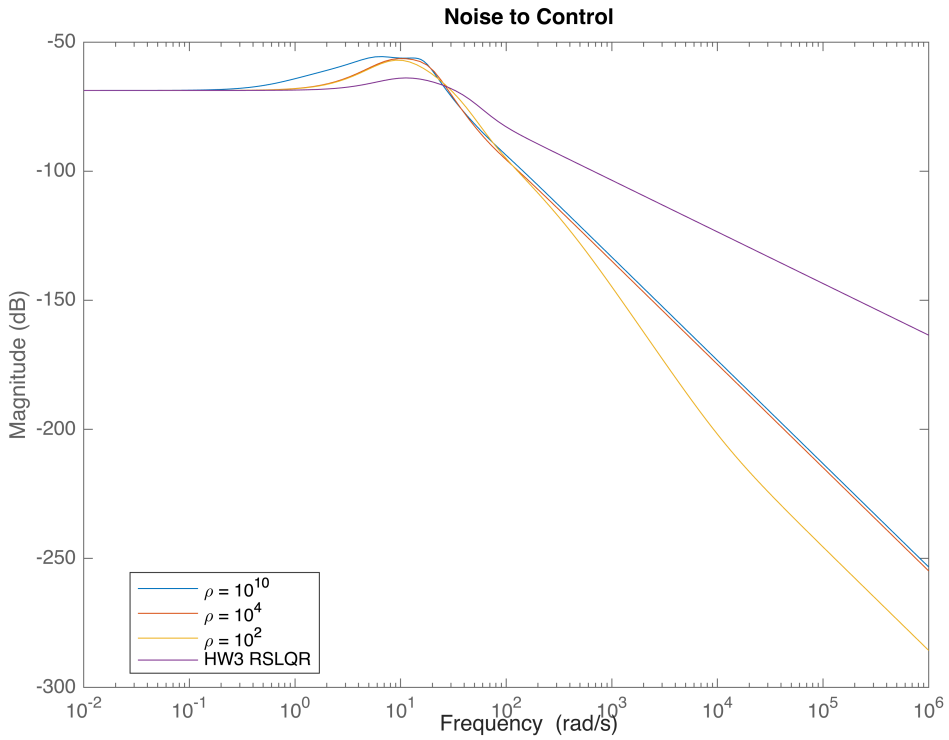


```
S_y_10 = 1/(1+sys_u_cell{1,1}(1,1));
S_y_4 = 1/(1+sys_u_cell{1,2}(1,1));
S_y_2 = 1/(1+sys_u_cell{1,3}(1,1));
S_y_rslqr = 1/(1+sys_rslqr_u(1,1));

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('\rho = 10^{10}','\rho = 10^{4}','\rho =
10^{2}','HW3 RSLQR')
```

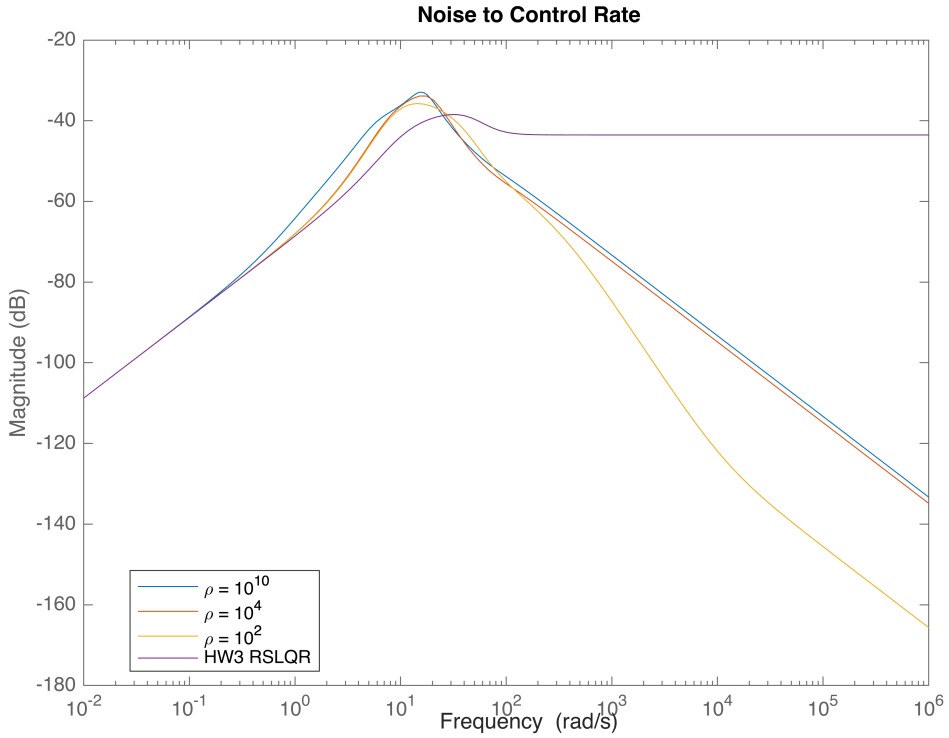



```
% Compute the noise-to-control TF
sys10.Bv = [ Bp*Zi*sys_controls_cell{1,5};

(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))
legend('\rho = 10^{10}','\rho = 10^{4}','\rho = 10^{2}','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))
legend('\rho = 10^{10}','\rho = 10^{4}','\rho = 10^{2}','HW3
RSLQR','Location','best')
title('Noise to Control Rate')
hold off
```




**For $\rho = 10^2$, List out your observer design matrices $(\tilde{A}, C, Q, R, L_p)$. List the controller matrices $(A_c, B_{c1}, B_{c2}, C_c, D_{c1}, D_{c2})$ implementing the controller**

$$A_w = \begin{pmatrix} 0 & -1156.89 & 0 \\ 0 & -1.30 & 1 \\ 0 & 47.71 & 0 \end{pmatrix} \quad C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad Q = \begin{pmatrix} 360.80 & 0.40 & 199.13 \\ 0.40 & 0.001 & 0.22 \\ 199.13 & 0.22 & 109.90 \end{pmatrix}$$

$$R = \begin{pmatrix} 0.625 & 0 \\ 0 & 0.001 \end{pmatrix} \quad L_p = \begin{pmatrix} 12.97 & 568.77 \\ -0.04 & 1.79 \\ 0.91 & 331.00 \end{pmatrix}$$

$$A_c = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 12.97 & -11.711 & -1567.15 & -609.13 \\ -0.047 & 0.048 & -1.76 & -0.84 \\ 0.91 & -0.21 & -178.71 & -353.27 \end{pmatrix} \qquad B_{c1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 568.77 & 0 & 0 \\ 0 & 0 & 1.79 & 0 & 0 \\ 0 & 0 & 331.00 & 0 & 0 \end{pmatrix}$$

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

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