Problem 1

$$A = \begin{bmatrix} 0 & 0 & -K \\ 0 & 0 & K \\ 1 & -1 & 0 \end{bmatrix}; \; B = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}; \; C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

$$G(s) = C(sI - A)^{-1}B + \cancel{D} \to 0$$

Plant O.L.T.F.

$$G(s) = \frac{0.5\omega^2}{s^3 + \omega^2 s} = P(s)$$ (Using MATLAB)

$$A(s) = s^3 + \omega^2 s = a_0 s^3 + a_1 s^2 + a_2 s + a_3$$

$$\Rightarrow a_0 = 1,\; a_1 = 0,\; a_2 = \omega^2,\; a_3 = 0$$

$$B(s) = \tfrac{1}{2}\omega^2 = b_0 s^3 + b_1 s^2 + b_2 s + b_3$$

$$\Rightarrow b_0 = b_1 = b_2 = 0,\; b_3 = 0.5\omega^2$$

Observer Poles: $10(-\omega \pm \omega j)$

$$\Omega(s) = (s + 10\omega + 10\omega j)(s + 10\omega + 10\omega j)$$

$$= (s + 10\omega)^2 + 100\omega^2 = s^2 + 20\omega s + 200\omega^2 = 0$$

State Feedback Poles: $-\omega,\; -\omega \pm \omega j$

$$\Phi(s) = s^3 + 3\omega s^2 + 4\omega^2 s + 2\omega^3 = 0$$

$$\alpha(s)A(s) + \beta(s)B(s) = \lambda(s)\Phi(s) = D(s)$$

$$\lambda(s)\Phi(s) = (s^2+20\omega s+200\omega^2)(s^3+3\omega s^2+4\omega^2 s+2\omega^3)$$

$$\lambda(s)\Phi(s) = s^5+23\omega s^4+264\omega^2 s^3+682\omega^3 s^2+840\omega^4 s+400\omega^5$$

$$\Rightarrow d_0 = 1,\ d_1 = 23\omega,\ d_2 = 264\omega^2,\ d_3 = 682\omega^3,\ d_4 = 840\omega^4,\ d_5 = 400\omega^5$$

$$\alpha(s) = \alpha_0 s^2 + \alpha_1 s + \alpha_2\ ;\ \beta(s) = \beta_0 s^2 + \beta_1 s + \beta_2$$

$$\begin{bmatrix} a_0 & 0 & 0 & b_0 & 0 & 0 \\ a_1 & a_0 & 0 & b_1 & b_0 & 0 \\ a_2 & a_1 & a_0 & b_2 & b_1 & b_0 \\ a_3 & a_2 & a_1 & b_3 & b_2 & b_1 \\ 0 & a_3 & a_2 & 0 & b_3 & b_2 \\ 0 & 0 & a_3 & 0 & 0 & b_3 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \omega^2 & 0 & 1 & 0 & 0 & 0 \\ 0 & \omega^2 & 0 & \frac{1}{2}\omega^2 & 0 & 0 \\ 0 & 0 & \omega^2 & 0 & \frac{1}{2}\omega^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2}\omega^2 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 23\omega \\ 264\omega^2 \\ 682\omega^3 \\ 840\omega^4 \\ 400\omega^5 \end{bmatrix}$$

Using MATLAB:

$\alpha_0 = 1$, $\alpha_1 = 23\omega$, $\alpha_2 = 263\omega^2$

$\beta_0 = 1318\omega$, $\beta_2 = 1154\omega^2$, $\beta_3 = 800\omega^3$

$\alpha(s) = s^2 - 23\omega s + 263\omega^2$

$\beta(s) = 1318\omega s^2 + 1154\omega^2 s + 800\omega^3$

$$\boxed{\begin{aligned} F_1(s) &= \frac{\lambda(s)}{\alpha(s)} = \frac{s^2 - 20\omega s + 200\omega^2}{s^2 - 23\omega s + 263\omega^2} \\ F_2(s) &= \frac{\beta(s)}{\lambda(s)} = \frac{1318\omega s^2 + 1154\omega^2 s + 800\omega^3}{s^2 - 20\omega s + 200\omega^2} \end{aligned}}$$

These results are identical to those obtained previously.

Additionally, since $\frac{Y(s)}{R(s)} = \frac{F_1(s)P(s)}{1 + F_1(s)F_2(s)P(s)}$, and since $F_1(s)$ and $F_2(s)$ are identical to those obtained in HW6, it's implied that both methods (the approach used in HW6 and the approach used in this homework) yield the same $\frac{Y(s)}{R(s)}$ transfer function.

$$\boxed{\frac{Y(s)}{R(s)} = \frac{\omega^2}{2(s^3 + 3\omega s^2 + 4\omega^2 s + 2\omega^3)}}$$ (from HW6)

MATLAB Code:

```
close all
clear
clc

syms w s

K = (w.^2)./2;

% Importing State Matrices:
A = [0 0 -K; 0 0 K; 1 -1 0];
B = [0; 1; 0];
C = [1 0 0];

G_s = C * inv(s * eye(3) - A) * B; % Obtains the plant's open-loop transfer
function

% Solving for D(s)
lam_s = s^2 + ((20*w*s) + (200 * (w.^2))); % Desired characteristic equation
for the observer
phi_s = s^3 + ((3 * w) * (s^2)) + ((4 * (w.^2))*s) + (2 * (w.^3)); % Desired
characteristic equation for the state feedback controller
D_s = expand(lam_s * phi_s);

% Solving the Matrix Equation A_mat * x = B_mat for this system of equations:
A_mat = [1 0 0 0 0 0; 0 1 0 0 0 0; (w.^2) 0 1 0 0 0; 0 (w.^2) 0 (0.5*w.^2) 0
0; 0 0 (w.^2) 0 (0.5*w.^2) 0; 0 0 0 0 0 (0.5*w.^2)];
B_mat = [1; (23 * w); (264 * (w.^2)); (682 * (w.^3)); (840 * (w.^4)); (400 *
(w.^5))];
x_mat = inv(A_mat) * B_mat; % Solving for the alpha and beta parameters

% Extracting the alpha and beta coefficients:
alph = x_mat(1:3);
beta = x_mat(4:6);
```

Problem 2

$\dot{x} = ax + u \Rightarrow b = 1$

$J = \int_0^\infty (x^2 + ru^2)\,dt = \int_0^\infty (x^T Q x + u^T R u)\,dt \Rightarrow Q = 1,\ R = r$

ARE: $A^T P + PA + Q - PBR^{-1}B^T P = 0$

Since we are working with scalars:

$aP + Pa + 1 - \frac{1}{r}P^2 = 0 \Rightarrow P^2 - 2arP - r = 0$

$\Rightarrow P = \dfrac{2ar \pm \sqrt{4a^2r^2 + 4r}}{2}$

$P = ar \pm \sqrt{a^2r^2 + r} \Rightarrow \underline{P = ar + \sqrt{a^2r^2 + r}}$

Since P is always positive

$K = R^{-1}B^T P = \left(\frac{1}{r}\right)\left(ar + \sqrt{a^2r^2 + r}\right) \Rightarrow \underline{K = a + \dfrac{\sqrt{a^2r^2 + r}}{r}}$

$$\boxed{u = -Kx = -\left(a + \frac{\sqrt{a^2r^2 + r}}{r}\right)x}$$

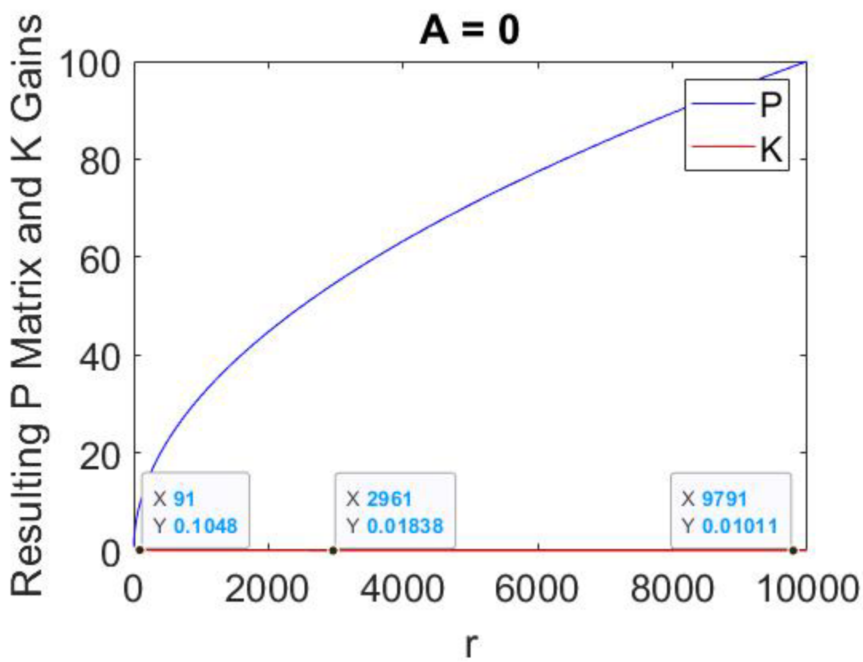
(Steady-state LQR optimal controller)

When $a=0$:

$K = 1/\sqrt{r}$

As $r \to 0$, $K \to +\infty$; As $r \to +\infty$, $K \to 0$

Using MATLAB:



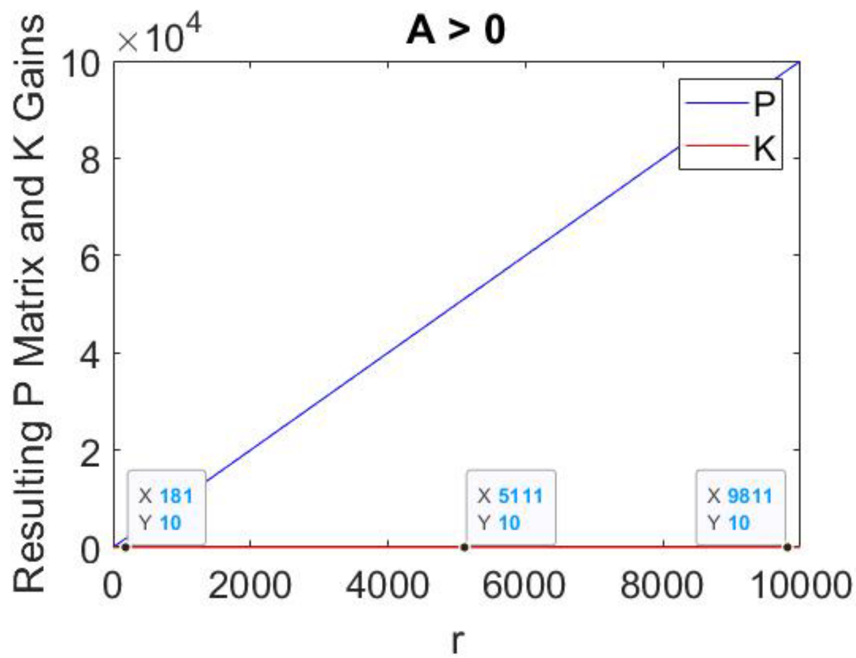
When $a=0$, the P-value rises logarithmically and the gains K gradually approach 0 as r approaches $+\infty$.

When $a > 0$:

$K = a + \frac{\sqrt{r^2a^2+r}}{r}$; Over $0 < r < +\infty$, K is always positive

As $r \to +\infty$, $K \to a + |a|$; As $r \to 0$, $K \to +\infty$

Using MATLAB:



When $a > 0$ ($a = 5$, in this case), the P-values appear to increase linearly and the K gains stay roughly constant around 10 as $r \to +\infty$ (as predicted since $K \to a + |a| = 10$).

When $a < 0$:

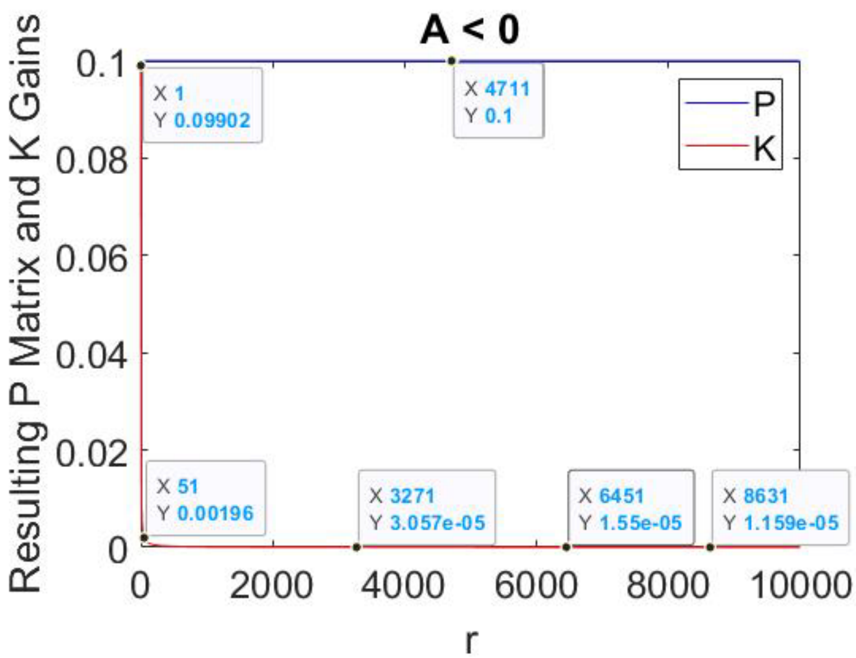
$K = a + \frac{\sqrt{a^2r^2+r}}{r}$

$a = -|a|$ since $a < 0$

As $r \to +\infty$, $K \to |a| + a = |a| - |a| = 0$

As $r \to 0$, $K \to +\infty$

Using MATLAB:



When $a > 0$ ($a = -5$, in this case), the P-values appear to stay roughly constant around 0.1 and the K gains rapidly approach zero as $r \to +\infty$.

MATLAB Code:

```
close all
clear
clc

% Importing state matrices and performance indices:
B = 1;
Q = 1;
R = [1:10:10000]; % Vector of possible R values
syms P;
                    or A<0, A>0
% A = 0:           Can change A value from case to case

A = 0; % State matrix for this case
K1 = []; % Vector of K gains
p = []; % Vector of P matrices (scalars, in this case)

% Solving the P matrix and gains K for each value of r:
for i = 1:length(R)
    % Solving for the P matrix using ARE:
    ARE = (Q - ((P.^2)/R(i))) == 0;
    ARE_soln = double(solve(ARE, P));
    p_cur = ARE_soln(ARE_soln > 0);
    p = [p; p_cur];

    % Solving for the gains:
    K1_cur = inv(R(i)) * B' * p_cur;
    K1 = [K1; K1_cur];
end

% Plotting results:
figure
plot(R, p, 'b')
hold on
plot(R, K1, 'r')
xlabel('r')
ylabel('Resulting P Matrix and K Gains')
title('A = 0') (or A<0, A>0)
legend('P', 'K')
set(gca, 'FontSize', 18)
```

$\boxed{\text{Problem 3}}$

$$\frac{Y(s)}{U(s)} = \frac{s+2}{s^2+2s+5}$$

$$s^2Y + 2sY + 5Y = sU + 2U$$

$$Y = \underbrace{\frac{1}{s}\left(-2Y + U + \underbrace{\frac{1}{s}(-5Y+2U)}_{x_2(s)}\right)}_{x_1(s)}$$

$$\begin{aligned} \dot{x}_2 &= -5y + 2u \\ \dot{x}_1 &= -2y + u + x_2 \\ y &= x_1 \end{aligned} \Rightarrow \left\{ \begin{aligned} \dot{x}_2 &= -5x_1 + 2u \\ \dot{x}_1 &= -2x_1 + x_2 + u \\ y &= x_1 \end{aligned} \right.$$

$$\boxed{\begin{gathered} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} -2 & 1 \\ -5 & 0 \end{bmatrix}}_{A} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 1 \\ 2 \end{bmatrix}}_{B} u \\ y = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} ; D = 0 \end{gathered}}$$

$$J = y^2(S) + \int_0^S r u^2(t)\, dt \; ; \; y = x_1 ; t_0 = 0, t_f = S$$

$$J = x_1(t_f)\, x_1(t_f) + \int_{t_0}^{t_f} r u^2(t)\, dt$$

$$= x^T(t_f)\, P_f\, x(t_f) + \int_{t_0}^{t_f} [x^T(t) Q x(t) + u^T(t) R u(t)]\, dt$$

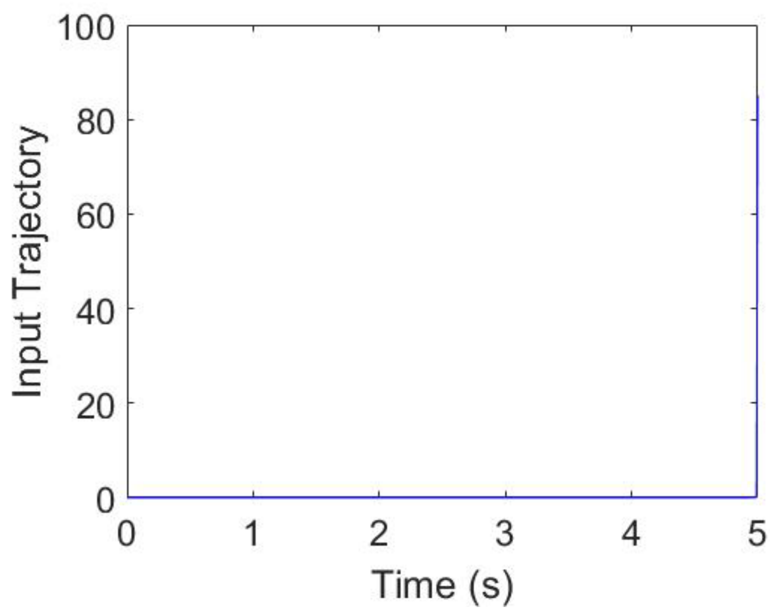
$$\boxed{Q = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} ; \; R = [r]}$$

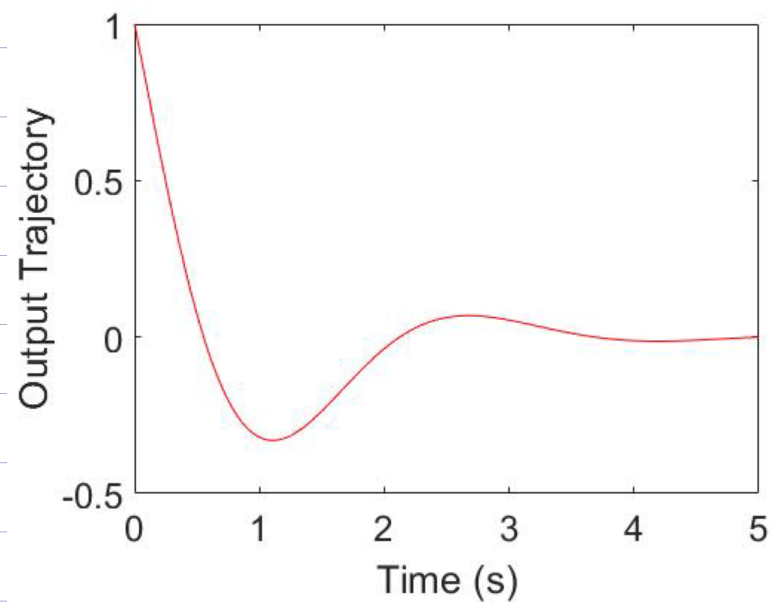
$$x_1^2(t_f) = x^T(t_f) \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} x(t_f) \Rightarrow P(t_f) = P(S) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

Ricatti Equation:

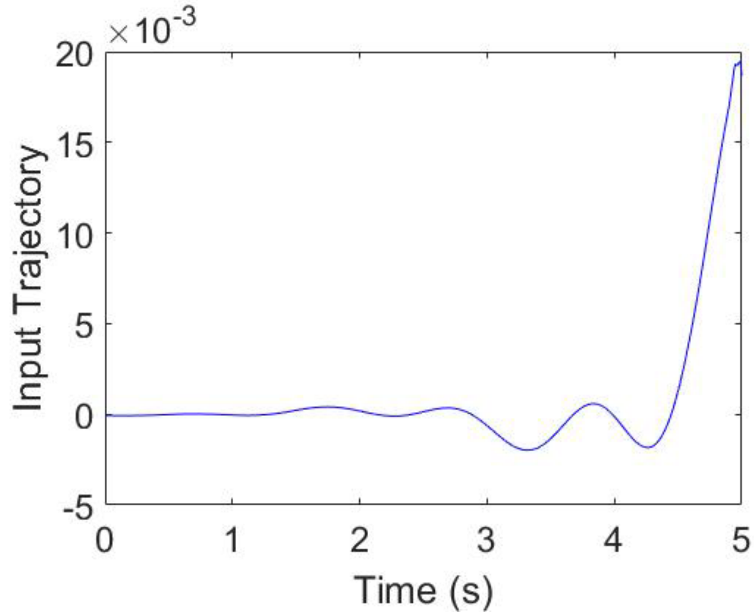
$$\dot{P} + PA + A^T P + Q - PBR^{-1}B^T P = 0, \; \overset{P(S)}{\overset{\shortparallel}{P(t_f)}} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

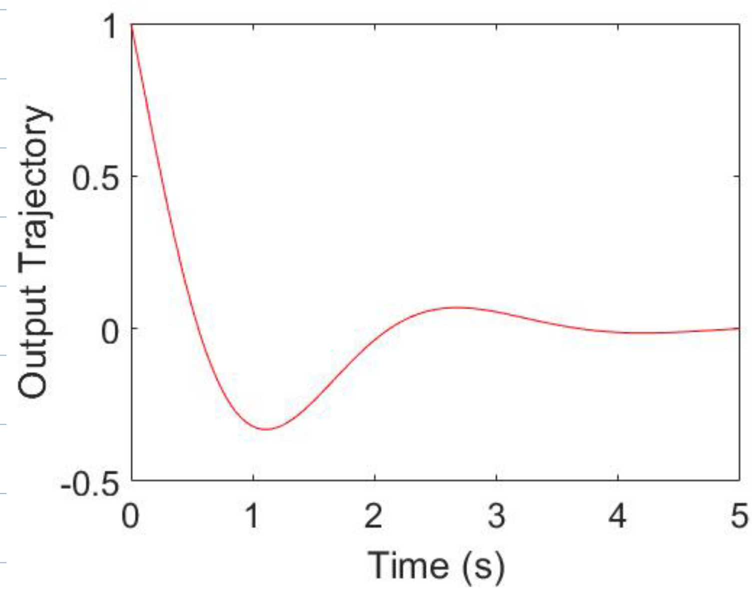
Using MATLAB, for $r = 10^{-6}$:



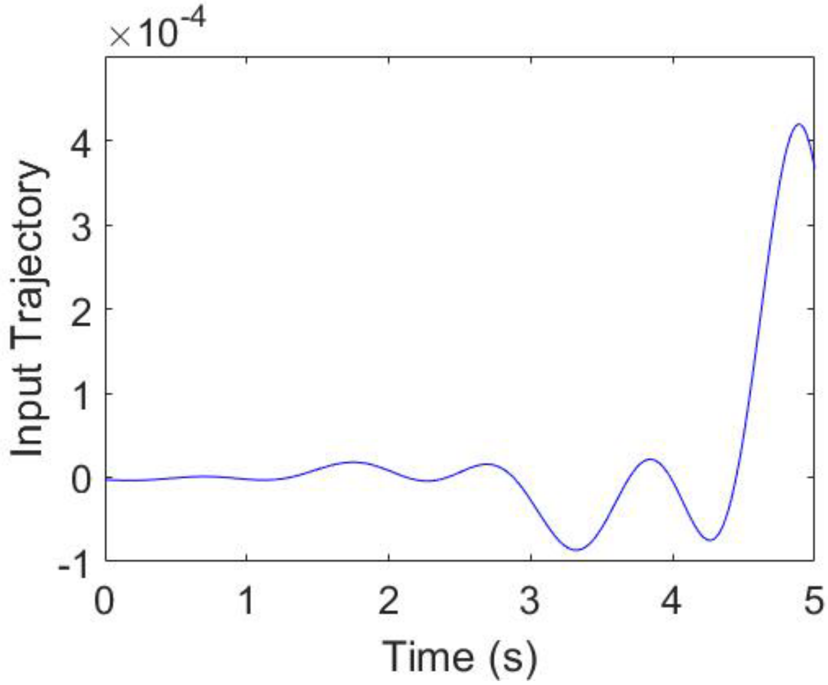


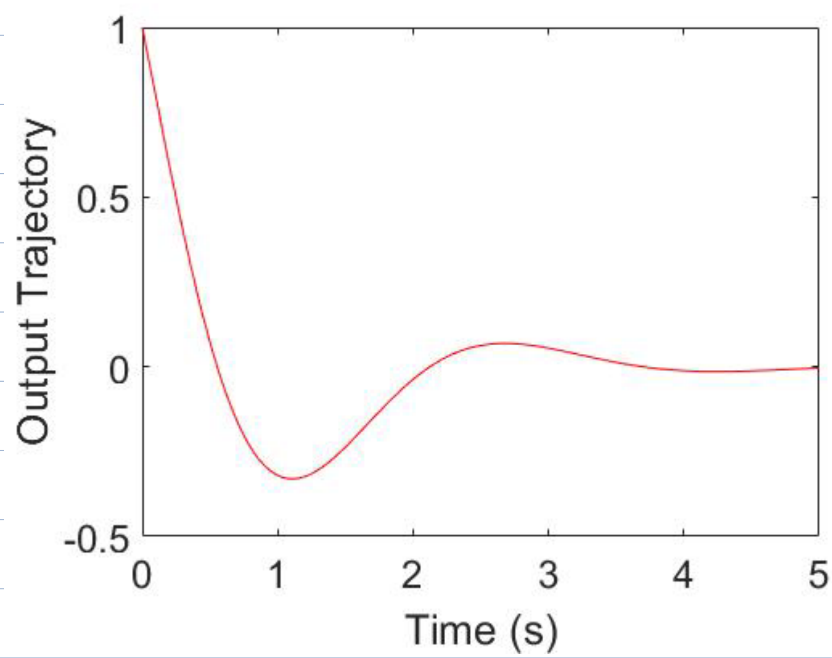
Using MATLAB, for $r = 0.01$:





Using MATLAB, for r = 10 :





By observing the plots of the input and output trajectories for the three different r-values (10^{-6}, 0.01, 10), it's clearly seen that as r increases, the magnitude of the input trajectory decreases, while the output trajectory stays the same (since $Q=0$). In reality, decreasing 'r' increases the actuator efforts applied to control and stabilize the system, thereby decreasing the settling time of the system for systems where $Q \neq 0$.

MATLAB Code:

```matlab
close all
clear
clc

% Importing State Matrices:
A = [-2 1; -5 0];
B = [1; 2];
C = [1 0];
D = 0;

% Performance Index Matrices
Q = zeros(2, 2);
P = [1 0; 0 0];

% Call ode45 to solve for Pb
T=[0:0.01:5]';
XP0=P(:);

% Can change r-value as necessary:
R = 10;

[T, XPb] = ode45('ricfun', T, XP0, [], A, B, Q, R);
XP = flipud(XPb);

for k = 1:length(T)
    Pk = reshape(XP(k,:)',2,2);
    K(k,:) = inv(R)*B'*Pk;
end

% Call ode45 to simulate the system response
[T,X]=ode45('sysfun', T, [1;0], [], A, B, K, T);

inp = -K.*X; % Solving for the input trajectory

% Plotting results:
figure
plot(T, inp(:, 1), 'b')
xlabel('Time (s)')
ylabel('Input Trajectory')
set(gca, 'FontSize', 18)

figure
plot(T, X(:, 1), 'r')
xlabel('Time (s)')
ylabel('Output Trajectory')
set(gca, 'FontSize', 18)
```

```matlab
% Helper Functions:
function dx = ricfun(t,x,flag,A,B,Q,R)
%Ricatti Eq. Function to be called by ode45
Pb=reshape(x,2,2);
dPb=Pb*A+A'*Pb+Q-Pb*B*inv(R)*B'*Pb;
dx=dPb(:);
end

function dx=sysfun(t,x,flag,A,B,K,T)
%System Function to be called by ode45
k=max(find(t>=T)); %determine the index k
Kk=K(k,:);
dx=(A-B*Kk)*x;
end
```

Problem 4

$$J = \int y(t)^2 + r u^2(t)\,dt = \int x_1^2(t) + r u^2(t)\,dt$$
$$= \int (x^T Q x + u^T R u)\,dt$$

$$\Rightarrow R = r, \quad Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

ARE: $A^T P + PA + Q - PBR^{-1}B^T P = 0$

$R = r, \quad 0 < r < +\infty$

Since $R = r =$ scalar, ARE implies:

$$1 + \left(\tfrac{1}{r}\right) G(-s)G(s) = H(-s)H(s)$$

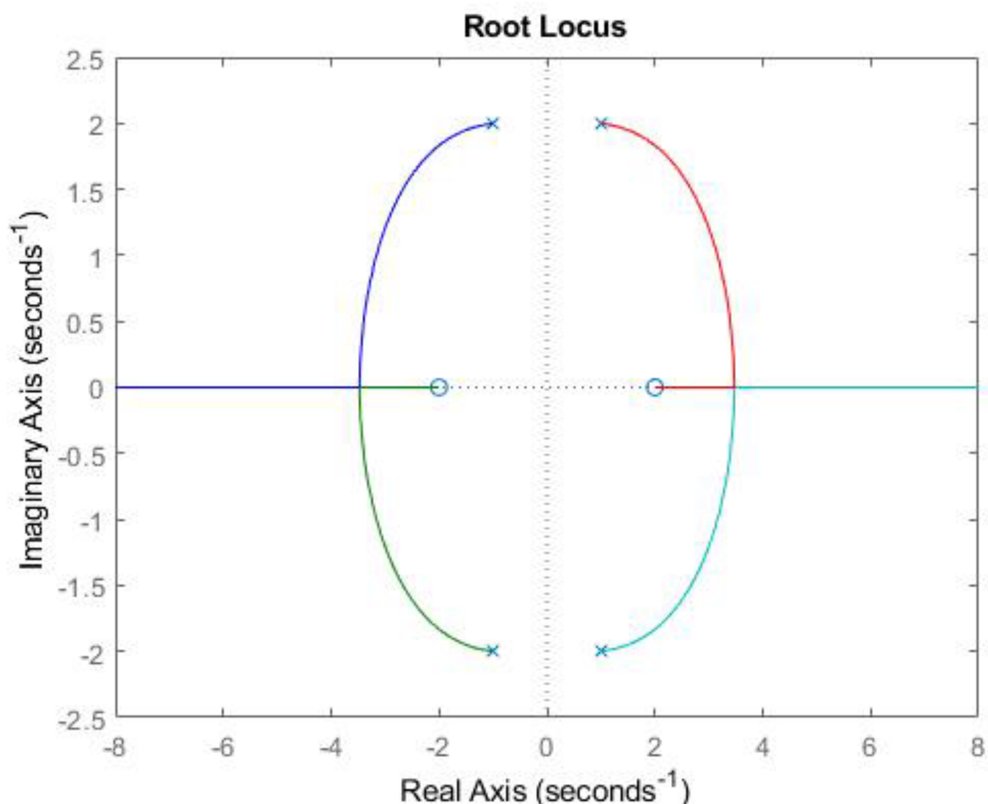
Gain * Open-Loop TF

Closed-Loop CE

$$G(s) = \frac{s+2}{s^2+2s+5} \; ; \; G(-s) = \frac{-s+2}{s^2-2s+5}$$

A root locus plot of $G(s)G(-s)$ would show how the closed-loop poles of the controlled system change as $\frac{1}{r}$ (and therefore r as well) ranges from 0 to $+\infty$.

Root-Locus Plot of $G(s)G(-s)$:



Closed-loop poles of controlled system as r varies from 0 to $+\infty$

MATLAB Code:

```
%% Problem 4

close all
clear
clc

% Importing system transfer functions:
Gs = tf([0 1 2], [1 2 5]);
Gnegs = tf([0 -1 2], [1 -2 5]);

% Plotting Root-Locus of System using "1/r" as the gain:
rlocus(Gs*Gnegs)
```

$\boxed{\text{Problem 5}}$

$$A = \begin{bmatrix} -10 & 0 & -10 & 0 \\ 0 & -0.7 & 9 & 0 \\ 0 & -1 & -0.7 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}; \quad B = \begin{bmatrix} 20 & 2.8 \\ 0 & -3.3 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$J = \int_0^\infty (\phi^2 + \beta^2 + \delta_a^2 + \delta_r^2)\,dt = \int_0^\infty (x^T Q x + u^T R u)\,dt$$

$x^T Q x = \phi^2 + \beta^2$ $\Rightarrow$ Only last two states are taken into account with coefficients of 1

Equal to each other

$$x^T Q x = x^T \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & d \end{bmatrix} x = ap^2 + br^2 + c\beta^2 + d\phi^2$$

$\Rightarrow a=0, b=0, c=1, d=1$

$$\Rightarrow Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$u^T R u = (\delta_a^2 + \delta_r^2) = \begin{bmatrix} \delta_a & \delta_r \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} = u^T u$$

$$u^T R u = u^T u \Rightarrow R = I$$

$$= R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Using the above info to solve for P using ARE:

$$PA + A^T P + Q - PBR^{-1}B^T P = 0$$

Using MATLAB:

P =

0.0046	0.0032	-0.0079	0.0500
0.0032	0.0372	-0.0339	0.0319
-0.0079	-0.0339	0.3692	-0.0821
0.0500	0.0319	-0.0821	0.5912

$K = R^{-1} B^T B$, using MATLAB:

K =

0.0915	0.0641	-0.1584	0.9992
0.0028	-0.1074	0.0838	0.0400

Hence, the LQR state feedback controller is:

$$u = -\underbrace{\begin{bmatrix} 0.0915 & 0.0641 & -0.1584 & 0.9992 \\ 0.0028 & -0.1074 & 0.0838 & 0.0400 \end{bmatrix}}_{K} \underbrace{\begin{bmatrix} p \\ r \\ \beta \\ \phi \end{bmatrix}}_{x}$$

MATLAB Code:

```
%% Problem 5

close all
clear
clc

% Importing State Matrices:
A = [-10 0 -10 0; 0 -0.7 9 0; 0 -1 -0.7 0; 1 0 0 0];
B = [20 2.8; 0 -3.13; 0 0; 0 0];

% Importing Performance Index Matrices:
Q = [0 0 0 0; 0 0 0 0; 0 0 1 0; 0 0 0 1];
R = [1 0; 0 1];

% Calculating the state-feedback gains using LQR Control
[P, K] = icare(A, B, Q, R);
```