Question:1

Monday, October 10, 2022 1:10 PM

Given -

$$\dot{x} = -\underset{\substack{\uparrow \\ \text{Terminal penalty}}}{\alpha} x + \underset{\substack{\uparrow \\ \text{Control i/p penalty}}}{K} u(t)$$

$x \rightarrow$ motor speed

$u \rightarrow$ input voltage

$$J = \underbrace{\beta x^2(t_f)}_{\text{Terminal cost}} + \underbrace{\int_0^{t_f} (x^2(t) + \rho u^2(t))\, dt}_{\text{Stage cost}}$$

From Hamilton :

$$\mathcal{H}(x(t), u(t), J_x^*, t) = g(x, u, t) + J_x^{*T} f(x, u, t)$$

$$= x^2 + \rho u^2 + J_x^{*T}[-\alpha x + Ku]$$

Finding the 'u' that minimizes the Hamilton :

$$\frac{\partial \mathcal{H}}{\partial u} = 0 \Rightarrow \frac{\partial \mathcal{H}}{\partial u} = 0 + 2\rho u + K J_x^* = 0$$

$$\Rightarrow 2\rho u + K J_x^* = 0$$

$$\Rightarrow \boxed{u^*(t) = -\frac{K J_x^*}{2\rho}}$$

Checking if the 'u' above is the minimizing solution -

$$\Rightarrow \frac{\partial^2 H}{\partial u^2} = 2\rho > 0$$

(as ρ is positive definite)

HJB Equation:

$$0 = J_t^* + H(x, u^*, J_x^*, t)$$

$$0 = J_t^* + x^2 + \rho\left(\frac{-K J_x^*}{2\rho}\right)^2$$

$$+ J_x^*\left[-\alpha x + K\left(\frac{-K J_x^*}{2\rho}\right)\right] \text{ — } 1$$

Boundary condition:

$$J^*(x(t_f), t_f) = \beta x^2(t_f)$$

Taking a guess -

$$J^*(x(t), t) = \frac{1}{2} P(t) x^2(t)$$

$$\Rightarrow J_x^* = P(t) x(t)$$

$$\Rightarrow J_t^* = \frac{1}{2} \dot{P}(t) x^2(t)$$

$$\Rightarrow u^* = \frac{-K J_x^*}{2\rho} = \frac{-K}{2\rho}\left(P(t) x(t)\right)$$

At $t = t_f$;

$$\frac{1}{2} P(t_f) \cancel{x^2(t_f)} = \beta \cancel{x^2(t_f)}$$

$$P(t_f) = 2\beta$$

Substituting the guess in HJB (equation ①) –

$$0 = \frac{1}{2}\dot{P}(t)x^2(t) + x^2(t) + \rho\left(\frac{-K}{2\rho}P(t)x(t)\right)^2$$

$$+ P(t)x(t)\left[-\alpha x(t) + K\left(\frac{-K}{2\rho}P(t)x(t)\right)\right]$$

The above eqn needs to be satisfied by all $x(t)$:

$$\frac{1}{2}\dot{P} + 1 + \frac{K^2P^2}{4\rho} - P\alpha - \frac{K^2P^2}{2\rho} = 0$$

$$\dot{P} = \frac{K^2P^2}{\rho} + 2P\alpha - \frac{K^2P^2}{2\rho} - 2$$

$$\frac{dP}{dt} = \frac{2K^2P^2 + 4P\rho\alpha - K^2P^2 - 4\rho}{2\rho}$$

$$2\rho\int_{P(t)}^{P(t_f)} \frac{1}{(K^2P^2 + 4P\rho\alpha - 4\rho)}\,dP = \int_{t}^{t_f} dt$$

This can be written as:
$(P-a)(P-b)$
Where 'a' and 'b' are the roots of the equation.

$P(t_f)$

$$2\rho \int_{P(t)}^{P(t_f)} \frac{1}{(P-a)(P-b)} dP = t_f - 0$$

$$= 2\rho \int_{P(t)}^{P(t_f)} \frac{A}{(P-a)} + \frac{B}{(P-b)}$$

$$= 2\rho \int_{P(t)}^{P(t_f)} \frac{A(P-b)+B(P-a)}{(P-a)(P-b)}$$

$$= 2\rho \int_{P(t)}^{P(t_f)} \frac{(A+B)P - Ab - Ba}{(P-a)(P-b)}$$

$$\left(\begin{array}{l} A+B=0 \Rightarrow A=-B \\ Bb - Ba = 1 \\ B(b-a)=1 \\ \qquad \Rightarrow B = \frac{1}{b-a}, A = \frac{1}{a-b} \end{array} \right)$$

$$\Rightarrow 2\rho \int_{P(t)}^{P(t_f)} \left(\frac{1}{a-b} \cdot \frac{1}{(P-a)} + \frac{1}{b-a} \frac{1}{(P-b)} \right) dP = t_f$$

$$\Rightarrow \frac{2\rho}{(a-b)} \int_{P(t)}^{P(t_f)} \frac{1}{(P-a)} dP + \frac{2\rho}{(b-a)} \int_{P(t)}^{P(t_f)} \frac{1}{(P-b)} dP = t_f$$

$$\Rightarrow \frac{2\rho}{(a-b)} \ln(P-a) \Big|_{P(t)}^{P(t_f)} + \frac{2\rho}{(b-a)} \ln(P-b) \Big|_{P(t)}^{P(t_f)} = t_f$$

$$\Rightarrow \frac{2\rho}{(a-b)} \ln(P-a)\Big|_{P(t)} + \frac{2\rho}{(b-a)} \ln(P-b)\Big|_{P(t)} = t_f$$

$$\Rightarrow \frac{2\rho}{(a-b)} \ln \frac{P_{t_f}-a}{P-a} + \frac{2\rho}{(b-a)} \ln \frac{P_{t_f}-b}{P-b} = t_f$$

We Know, $P(t_f) = 2\beta$

$$\Rightarrow \frac{2\rho}{a-b} \ln \frac{2\beta-a}{P-a} + \frac{2\rho}{(b-a)} \ln \frac{2\beta-b}{P-b} = t_f$$

$$\Rightarrow \frac{2\rho}{a-b}\left(\ln \frac{2\beta-a}{P-a} - \ln \frac{2\beta-b}{P-b}\right) = t_f$$

$$\Rightarrow \ln\left(\frac{2\beta-a}{P-a} \times \frac{P-b}{2\beta-b}\right) = \frac{t_f(a-b)}{2\rho}$$

$$\Rightarrow \boxed{\frac{(2\beta-a)(P-b)}{(P-a)(2\beta-b)} = e^{\frac{t_f(a-b)}{2\rho}}}$$

Solving this for $P(t)$ will give -

$$\Rightarrow J^* = \frac{1}{2} P(t)\, x^2(t)$$

$$\Rightarrow u^* = -\frac{K J_x^*}{2\rho}$$

Closed-loop System -

$$\dot{x} = x + u$$

Question:2

Wednesday, October 5, 2022 5:40 PM

Given –

States –

β (sideslip angle)
ψ (yaw angle)
r (yaw angular velocity)
ϕ (roll angle)
p (roll angular velocity)

Control inputs –

δ_r (rudder deflection)
δ_a (aileron deflection)

Cost function –

$$J = \int_0^\infty \left[\frac{\delta_a^2}{\delta_{a_0}^2} + \frac{\delta_r^2}{\delta_{r_0}^2} + \frac{(\beta+\psi)^2}{\epsilon_0^2} + \frac{\phi^2}{\phi_0^2} \right] dt$$

Linearized dynamics of lateral motion (a fifth order system) are –

(a) $\dot{\beta} + r = \frac{Y_\beta}{mV}\beta + \frac{g}{V}\phi$

(b) $\dot{r} + \frac{I_{xz}}{I_{zz}}\dot{p} = \frac{n_\beta}{I_{zz}}\beta + \frac{n_r}{I_{zz}}r + \frac{n_p}{I_{zz}}p + \frac{n_{\delta_r}}{I_{zz}}\delta_r$

(c) $\dot{p} + \frac{I_{zz}}{I_{xx}}\dot{r} = \frac{l_p}{I_{xx}} + \frac{l_\beta}{I_{xx}}\beta + \frac{l_r}{I_{xx}}r + \frac{I_{\delta a}}{I_{xx}}\delta_a$

(d) $\dot{\phi} = p$

(e) $\dot{\psi} = r$

Assigning numerical values to constants equation ①,②,③,④,⑤ becomes-

$$\dot{\beta} = -0.279\beta + 0.0438\phi - r \quad \text{—— ①}$$

$$\dot{r} - 0.0423\dot{p} = 0.379\beta - 0.0096r - 0.0125p + 0.379\delta_r \quad \text{—— ②}$$

$$\dot{p} - 0.0438\dot{r} = -0.790p - 1.17\beta + 0.129r + 1.580\delta_a \quad \text{—— ③}$$

$$\dot{\phi} = p \quad \text{—— ④}$$

$$\dot{\psi} = r \quad \text{—— ⑤}$$

Solving for $\dot{p}$ ⇒ Multiplying ② × 0.0438 and adding with ③ we get-

$$0.9982\dot{p} = -1.1534\beta + 0.1286r - 0.7905p + 0.3956\delta_r$$

or

$$\dot{p} = -1.1554\beta + 0.1288r - 0.7919p + 0.3963\delta_r \quad \text{—— ⑥}$$

Solving for $\dot{r}$ ⇒ Multiplying ③ × 0.0423 and adding with ② we get-

$$0.9982\dot{r} = 0.3256\beta - 0.0042r - 0.0459p + 0.379\delta_r + 0.0668\delta_a$$

or

$$\dot{r} = 0.3301\beta - 0.0042r - 0.0459p + 0.3796\delta_r + 0.0669\delta_a \quad \text{—— ⑦}$$

Writing difference equations ①,④,⑤,⑥,⑦ in state space form -

$$\underbrace{\begin{bmatrix} \dot{\beta} \\ \dot{\psi} \\ \dot{r} \\ \dot{\phi} \\ \dot{p} \end{bmatrix}}_{\dot{X}} = \underbrace{\begin{bmatrix} -0.297 & 0 & -1 & 0.0438 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0.3301 & 0 & -0.0042 & 0 & -0.0459 \\ 0 & 0 & 0 & 0 & 1 \\ -1.1554 & 0 & 0.1288 & 0 & -0.7919 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \beta \\ \psi \\ r \\ \phi \\ p \end{bmatrix}}_{X} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.3796 & 0.0669 \\ 0 & 0 \\ 0.3963 & 0 \end{bmatrix} \underbrace{\begin{bmatrix} \delta_r \\ \delta_a \end{bmatrix}}_{U}$$

Assigning the values of constants in the cost function -

Assigning the values of constants in the cost function -

$$J = \int_0^\infty \left[\frac{\delta a^2}{1} + \frac{\delta r^2}{1} + \frac{(\beta+\psi)^2}{1} + \frac{\phi^2}{1} \right] dt$$

or

$$J = \int_0^\infty \left[(\delta a^2 + \delta x^2) + (\beta^2 + \psi^2 + 2\beta\psi + \phi^2) \right] dt$$

Finding 'Q' & 'R' matrices -

$$\underset{X^T}{\begin{bmatrix} \beta & \psi & r & \phi & p \end{bmatrix}_{1\times5}} \underset{Q}{\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{5\times5}} \underset{X}{\begin{bmatrix} \beta \\ \psi \\ r \\ \phi \\ p \end{bmatrix}_{5\times1}}$$

$$\underset{U}{\begin{bmatrix} \delta r & \delta a \end{bmatrix}_{1\times2}} \underset{R}{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}_{2\times2}} \underset{U^T}{\begin{bmatrix} \delta r \\ \delta a \end{bmatrix}_{2\times1}}$$

- - - - - Continued in MATLAB - - - - -

Putting together the the matrices, `A`, `B`, `Q`, `R` from the state space equations -

```
clc;
clear;

% Matrix A
A = [-0.297, 0, -1, 0.0438, 0;
     0, 0, 1, 0, 0;
     0.3301, 0, -0.0042, 0, -0.0459;
     0, 0, 0, 0, 1;
     -1.1554, 0 0.1288, 0, -0.7919;];

% Matrix B
B = [0, 0;
     0, 0;
     0.3796, 0.0669;
     0, 0;
     0.3963, 0;];

% matrix Q
Q = [1, 1, 0, 0, 0;
     1, 1, 0, 0, 0;
     0, 0, 0, 0, 0;
     0, 0, 0, 1, 0;
     0, 0, 0, 0, 0;];

% Matrix R
R = [1, 0;
     0, 1;];
```

```
% Checking for controllability
iscontrol = rank(ctrb(A,B))
```

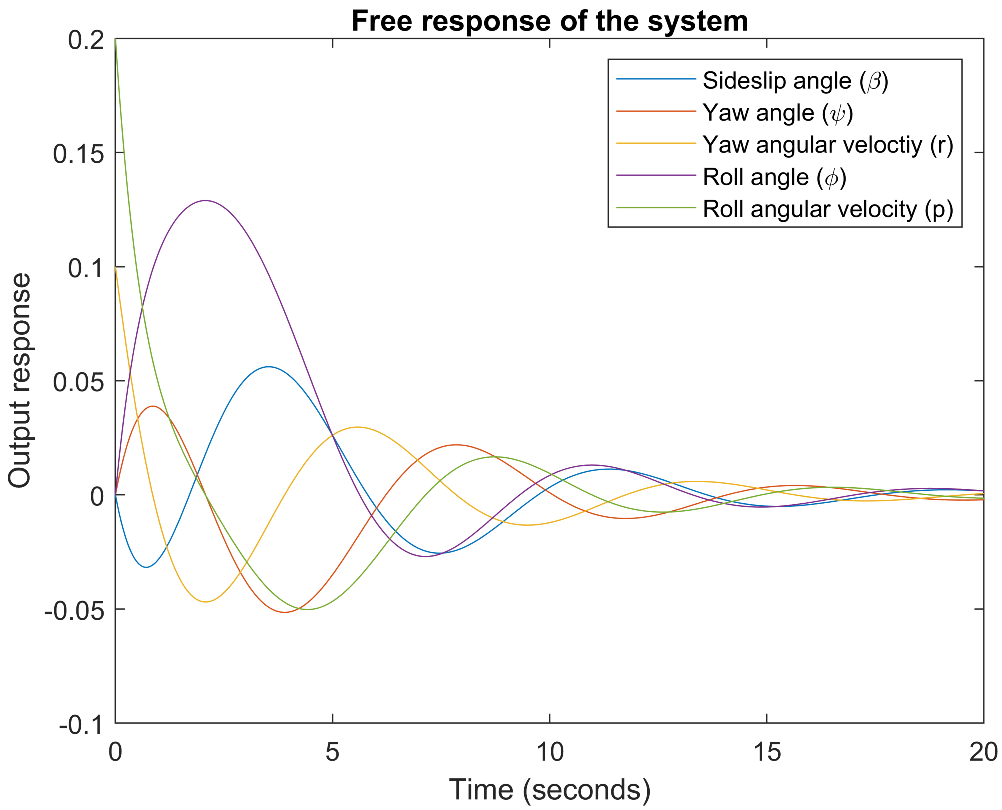
```
iscontrol = 5
```

Since the rank of controllability matrix equals number of states; the system is controllable.

```
% Calculating the full-state feedback gain matrix `K` using the `lqr` command
K = lqr(A, B, Q, R);
```

```
n = length(A);
C = eye(n);
D = zeros([n,2]);
tspan = 0:0.005:20; % time period is 20 seconds
x0 = [0; 0; 0.1; 0; 0.2]';
sys = ss(A-B*K,B,C,D);
[y, tspan, x] = initial(sys, x0, tspan);
figure
plot(tspan,y)
title('Free response of the system')
xlabel('Time (seconds)')
```

```
ylabel('Output response');
legend('Sideslip angle (\beta)', 'Yaw angle (\psi)', 'Yaw angular veloctiy (r)', 'Roll angle (\
```



Question-3)

```
clc;
clear;
```

Defining system dynamics

```
A = [0, 1, 0;
     8.75*10^4, 0, 7*10^4;
     0, -1.25, -597.82];
B = [0;
     0;
     271.74];
n = length(A);
C = eye(n);
D = zeros([n,1]);
```

(a) Stability check; when position is penalized along with the control input -

```
Q = [1, 0, 0; % Assuming a numerical value for q11(=1)
     0, 0, 0;
     0, 0, 0];
R = [1];

fprintf('Rank of observability matrix = %d',rank(obsv(A,Q)))
```

```
Rank of observability matrix = 3
```

```
fprintf('Rank of controllability matrix = %d',rank(ctrb(A,B))) ;
```

```
Rank of controllability matrix = 3
```

Rank of the observability matrix is equal to the number of states, which means the system is *observable* and hence *detectable*.

Rank of the controllability matrix is equal to the number of states, which means the system is *controllable* and hence *stabilizable*.

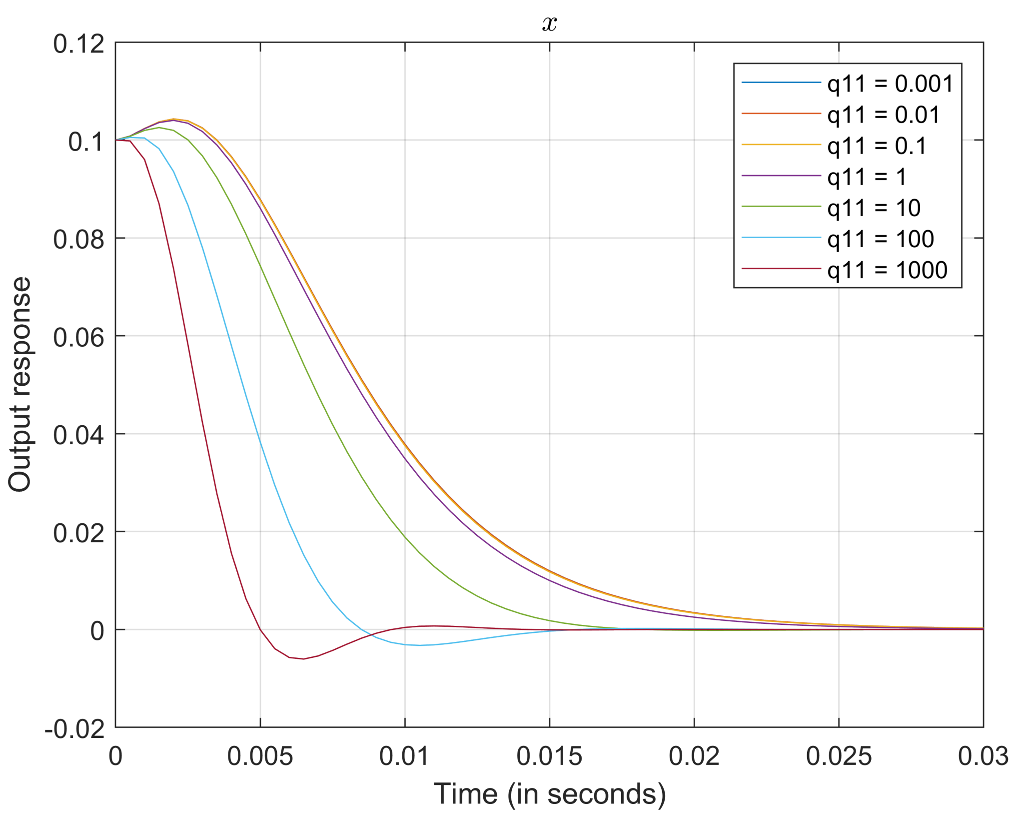
(b) Comparision between different values of q11 in {0.001,0.01,0.1,1,10,100,1000} -

```
tspan = 0:0.0005:0.03; % time period is 0.03 seconds
x0 = [0.1; 0; 0]';
Q = [0, 0, 0;
     0, 0, 0;
     0, 0, 0];
q11 = [0.001,0.01,0.1,1,10,100,1000];
```

X plot

```
for i = 1:length(q11)
    Q(1,1) = q11(i);
    K = lqr(A, B, Q, R);
```

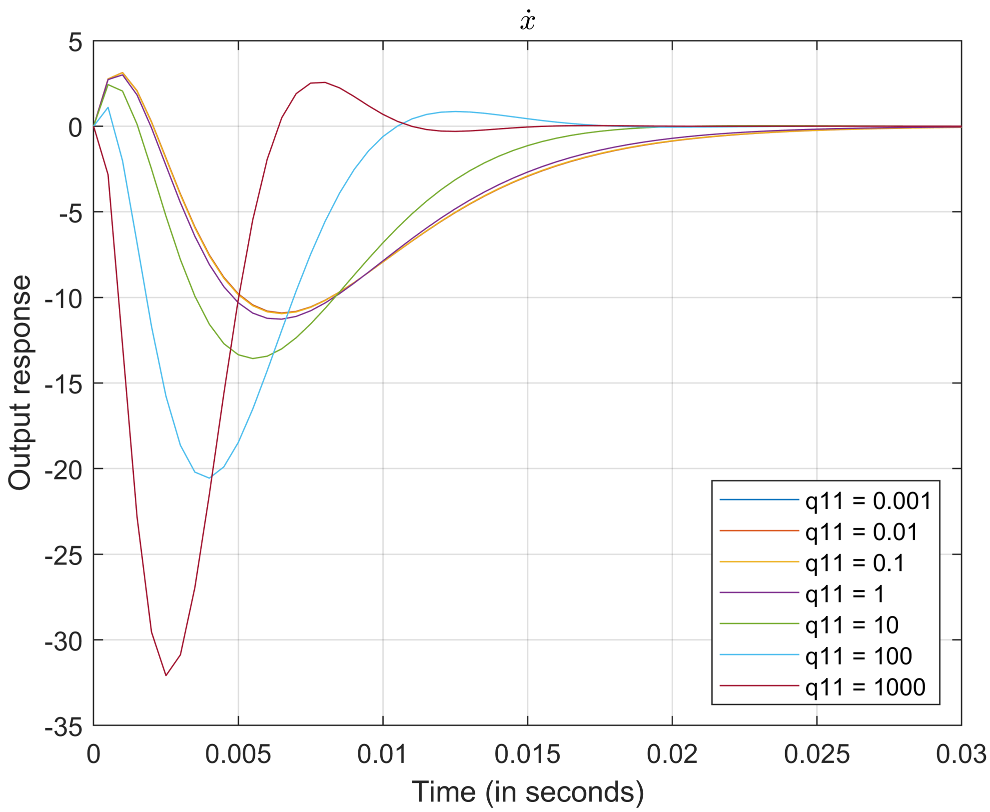
```
    sys = ss(A-B*K,B,C,D); % closed loop-system
    [y, tspan, x] = initial(sys, x0, tspan);
    figure(1)
    plot(tspan,y(:,1))
    title('\textbf{${x}$}','Interpreter','latex');
    xlabel('Time (in seconds)');
    ylabel('Output response');
    hold on
    grid
end
hold off
legend('q11 = 0.001','q11 = 0.01','q11 = 0.1','q11 = 1','q11 = 10','q11 = 100','q11 = 1000', lo
```



Xdot plot

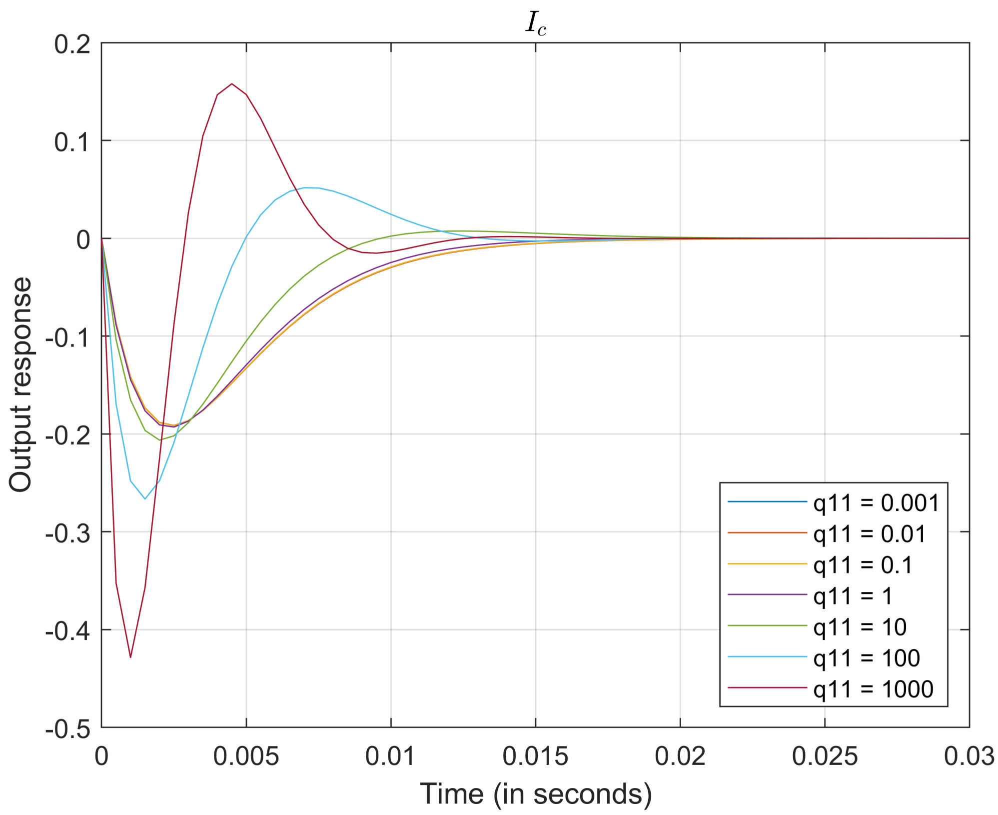
```
for i = 1:length(q11)
    Q(1,1) = q11(i);
    K = lqr(A, B, Q, R);
    sys = ss(A-B*K,B,C,D); % closed loop-system
    [y, tspan, x] = initial(sys, x0, tspan);
    figure(2)
    plot(tspan,y(:,2))
    title('\textbf{$\dot{x}$}','Interpreter','latex');
    xlabel('Time (in seconds)');
    ylabel('Output response');
    hold on
    grid
end
```

```
hold off
legend('q11 = 0.001','q11 = 0.01','q11 = 0.1','q11 = 1','q11 = 10','q11 = 100','q11 = 1000', lo
```



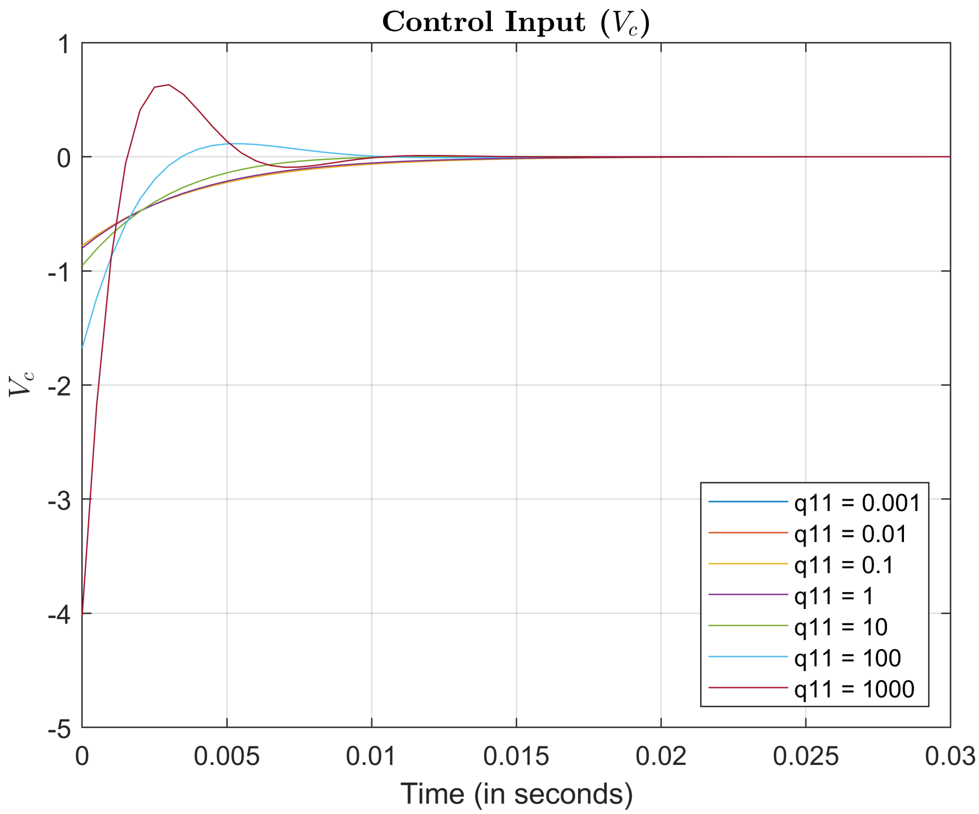
Ic plot

```
for i = 1:length(q11)
    Q(1,1) = q11(i);
    K = lqr(A, B, Q, R);
    sys = ss(A-B*K,B,C,D); % closed loop-system
    [y, tspan, x] = initial(sys, x0, tspan);
    figure(3)
    plot(tspan,y(:,3))
    title('\textbf{${I_c}$}','Interpreter','latex');
    xlabel('Time (in seconds)');
    ylabel('Output response');
    hold on
    grid
end
hold off
legend('q11 = 0.001','q11 = 0.01','q11 = 0.1','q11 = 1','q11 = 10','q11 = 100','q11 = 1000', lo
```



Control input (Vc) plot

```
for i = 1:length(q11)
    Q(1,1) = q11(i);
    K = lqr(A, B, Q, R);
    sys = ss(A-B*K,B,C,D); % closed loop-system
    [y, tspan, x] = initial(sys, x0, tspan);
    u = - (K*x')';
    figure(4)
    plot(tspan,u)
    title('\textbf{Control Input (${V_c}$)}', 'Interpreter','latex');
    xlabel('Time (in seconds)');
    ylabel('${V_c}$', 'Interpreter','latex');
    hold on
    grid
end
hold off
legend('q11 = 0.001','q11 = 0.01','q11 = 0.1','q11 = 1','q11 = 10','q11 = 100','q11 = 1000', lo
```



Recommended choice of q11 (part-b)

As per the graphs above, it can be seen that as the value of *q11* increases, the states are stabilized quicker with more changes in less time. For instance, for the value of *q11 equal to 1000,* the states are stabilized faster compared to when the value of *q11 is equal to 0.001.* Also, since the value of the control penalty matrix *R is fixed,* it can be seen that increasing the value of *q11* leads to higher control input.

Therefore, if control input is not a concern, then *q11=1000* would be the recommended value in the state penalty matrix. However, if control input is a concern, then choosing *q11=0.001* would be optimal, considering the trade-off would be the time required by the system to stabilize.

For this problem, assuming both control input and change in states are of interest, I would choose *q11=10,* which gives a good balance between the control input and change in states.

(c) RMS Values over a period of 0.1 second for Position and Control Input -

```
tspan = 0:0.0005:0.1; % time period is 0.1 seconds
x0 = [0.1; 0; 0]';
Q = [0, 0, 0;
     0, 0, 0;
     0, 0, 0];
q11 = [0.001,0.01,0.1,1,10,100,1000];
```

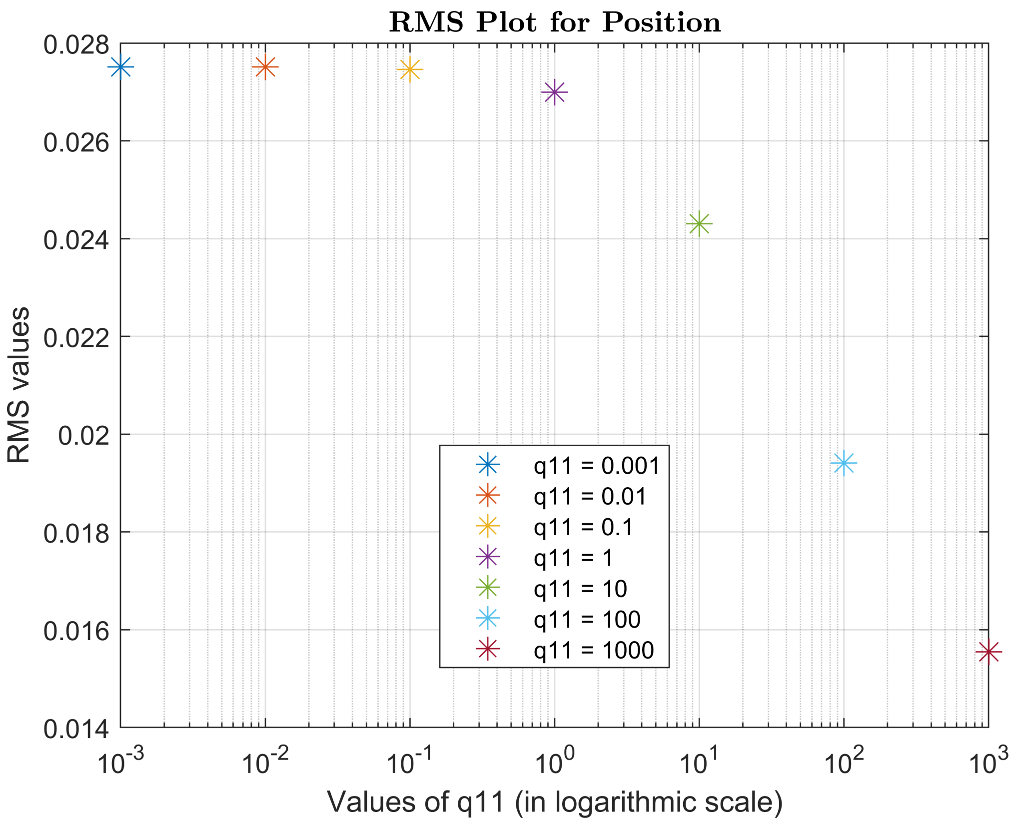
RMS plot for position

```
for i = 1:length(q11)
    Q(1,1) = q11(i);
```

```
    K = lqr(A, B, Q, R);
    sys = ss(A-B*K,B,C,D); % closed loop-system
    [y, tspan, x] = initial(sys, x0, tspan);
    figure(5)
    rms_pos = rms(y(:,1));
    semilogx(q11(i),rms_pos,'*', 'MarkerSize',10);
    title('\textbf{RMS Plot for Position}', 'Interpreter','latex');
    xlabel('Values of q11 (in logarithmic scale)');
    ylabel('RMS values');
    fprintf('RMS value of position when q11 is %0.2f = %0.2f \n',q11(i),rms_pos);
    hold on
    grid
end
```

```
RMS value of position when q11 is 0.00 = 0.03
RMS value of position when q11 is 0.01 = 0.03
RMS value of position when q11 is 0.10 = 0.03
RMS value of position when q11 is 1.00 = 0.03
RMS value of position when q11 is 10.00 = 0.02
RMS value of position when q11 is 100.00 = 0.02
RMS value of position when q11 is 1000.00 = 0.02
```

```
hold off
legend('q11 = 0.001','q11 = 0.01','q11 = 0.1','q11 = 1','q11 = 10','q11 = 100','q11 = 1000', lo
```



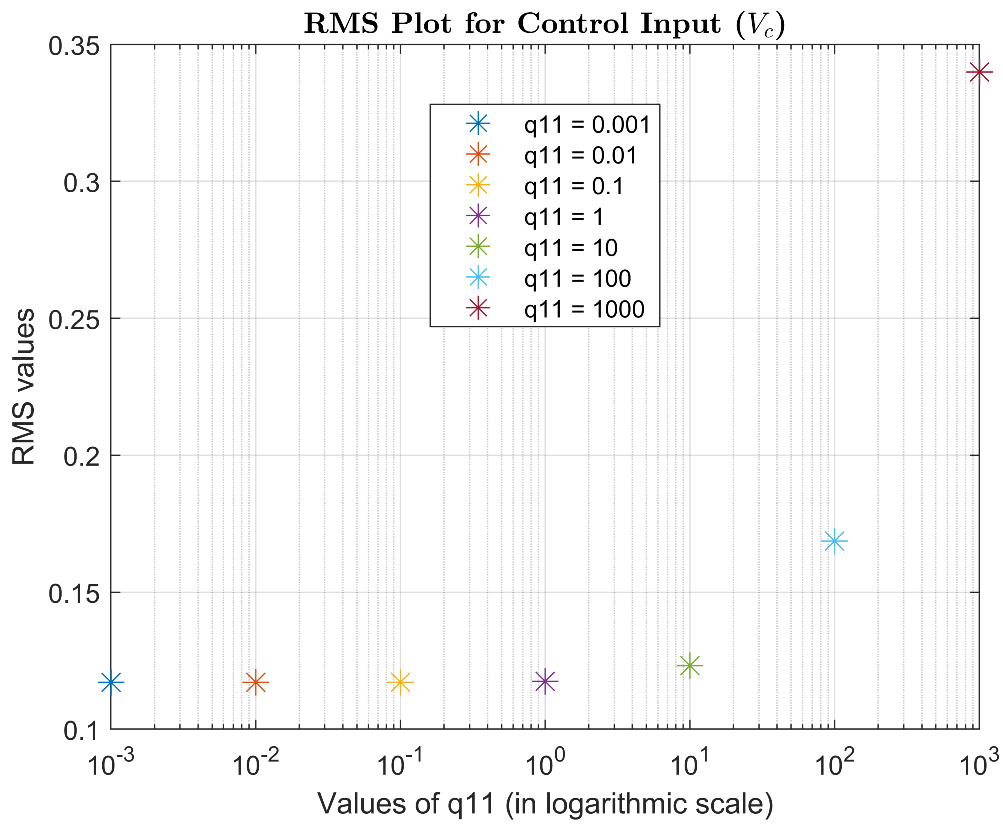
RMS plot for control input (Vc)

```
for i = 1:length(q11)
    Q(1,1) = q11(i);
```

```
    K = lqr(A, B, Q, R);
    sys = ss(A-B*K,B,C,D); % closed loop-system
    [y, tspan, x] = initial(sys, x0, tspan);
    u = - (K*x')';
    figure(6)
    rms_u = rms(u);
    semilogx(q11(i),rms_u,'*', 'MarkerSize',10);
    title('\textbf{RMS Plot for Control Input (${V_c}$)}', 'Interpreter','latex');
    xlabel('Values of q11 (in logarithmic scale)');
    ylabel('RMS values');
    fprintf('RMS value of control input when q11 is %0.2f = %0.2f \n',q11(i),rms_u);
    hold on
    grid
end
```

```
RMS value of control input when q11 is 0.00 = 0.12
RMS value of control input when q11 is 0.01 = 0.12
RMS value of control input when q11 is 0.10 = 0.12
RMS value of control input when q11 is 1.00 = 0.12
RMS value of control input when q11 is 10.00 = 0.12
RMS value of control input when q11 is 100.00 = 0.17
RMS value of control input when q11 is 1000.00 = 0.34
```

```
hold off
legend('q11 = 0.001','q11 = 0.01','q11 = 0.1','q11 = 1','q11 = 10','q11 = 100','q11 = 1000', lo
```



Choice of q11 and the respective eigenvalues of the closed-loop system

Looking at the RMS plots above, choosing q11=100 seems to be the right choice as it shows a good trade-off between the transitions of position and the required control input.

```
% Eigen values of the closed-loop system when q11=100
x0 = [0.1; 0; 0]';
Q = [100, 0, 0;
     0, 0, 0;
     0, 0, 0];
K = lqr(A, B, Q, R);
display('Eigen values of closed-loop system when q11=100')
```

```
Eigen values of closed-loop system when q11=100
```

```
eig(A-B*K) % for q11 = 100
```

```
ans = 3×1 complex
10^2 ×
  -3.6637 + 4.1017i
  -3.6637 - 4.1017i
  -6.5224 + 0.0000i
```

(d) Comparision between RMS plots when q11=100 and penalizing velocity(xdot)

```
tspan = 0:0.0005:0.1; % time period is 0.1 seconds
x0 = [0.1; 0; 0]';
Q = [100, 0, 0;
     0, 0, 0;
     0, 0, 0];
q22 = [0.01,0.1,1,10,100];
```

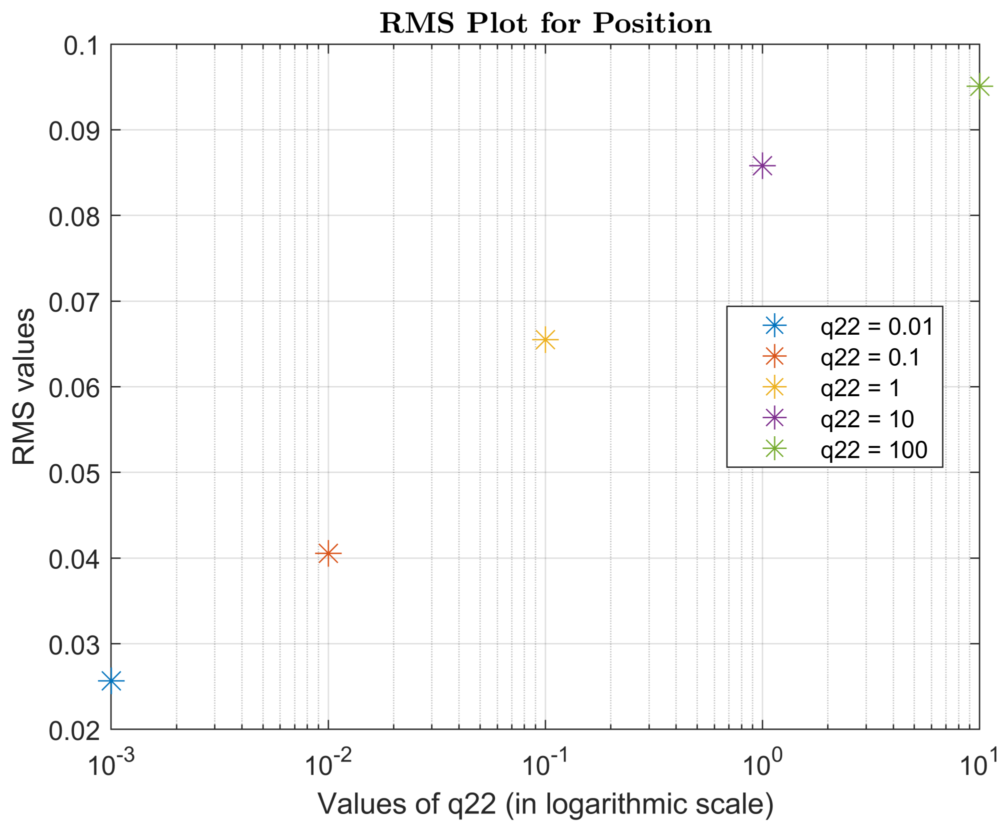
RMS plot for position

```
for i = 1:length(q22)
    Q(2,2) = q22(i);
    K = lqr(A, B, Q, R); % Control gain for different penalties on velocity
    sys = ss(A-B*K,B,C,D); % closed loop-system
    [y, tspan, x] = initial(sys, x0, tspan);
    figure(7)
    rms_pos = rms(y(:,1));
    semilogx(q11(i),rms_pos,'*', 'MarkerSize',10);
    title('\textbf{RMS Plot for Position}', 'Interpreter','latex');
    xlabel('Values of q22 (in logarithmic scale)');
    ylabel('RMS values');
    fprintf('RMS value for position when q22 is %0.2f = %0.2f \n',q22(i),rms_pos);
    hold on
    grid
end
```

```
RMS value for position when q22 is 0.01 = 0.03
RMS value for position when q22 is 0.10 = 0.04
RMS value for position when q22 is 1.00 = 0.07
RMS value for position when q22 is 10.00 = 0.09
RMS value for position when q22 is 100.00 = 0.10
```

```
hold off
```

```
legend('q22 = 0.01','q22 = 0.1','q22 = 1','q22 = 10','q22 = 100', location='best')
```

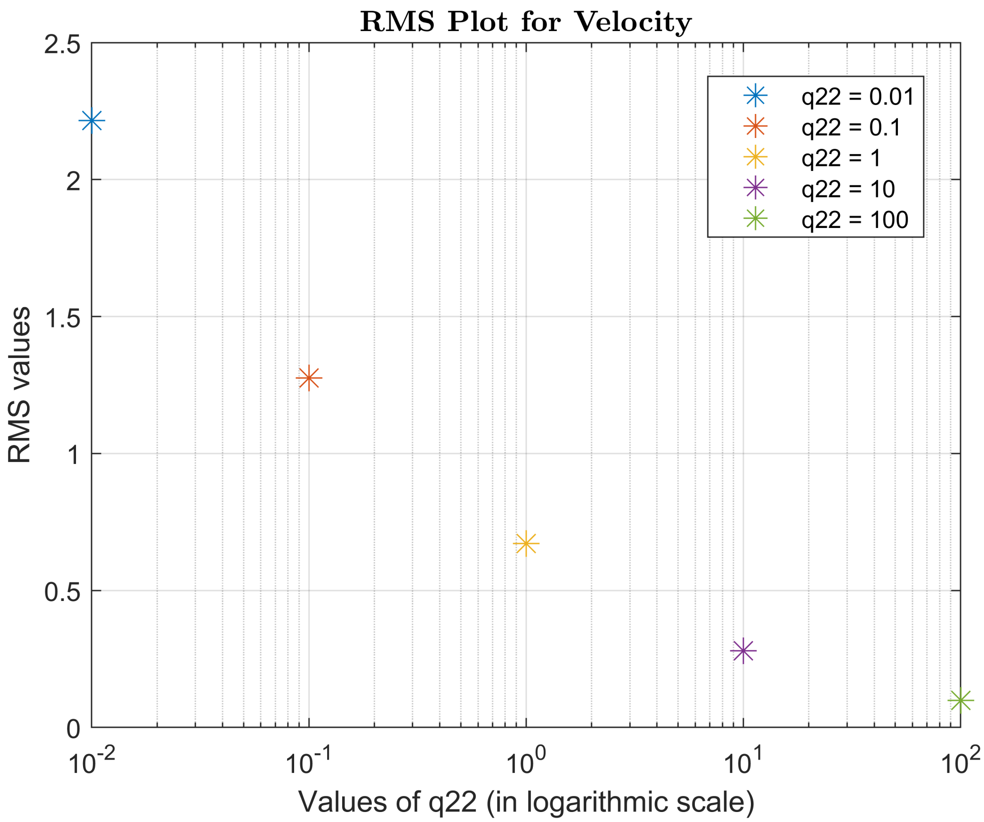


RMS plot for velocity (xdot)

```
for i = 1:length(q22)
    Q(2,2) = q22(i);
    K = lqr(A, B, Q, R); % Control gain for different penalties on velocity
    sys = ss(A-B*K,B,C,D); % closed loop-system
    [y, tspan, x] = initial(sys, x0, tspan);
    figure(8)
    rms_vel = rms(y(:,2));
    semilogx(q22(i),rms_vel,'*', 'MarkerSize',10);
    title('\textbf{RMS Plot for Velocity}', 'Interpreter','latex');
    xlabel('Values of q22 (in logarithmic scale)');
    ylabel('RMS values');
    fprintf('RMS value for velocity when q22 is %0.2f = %0.2f \n',q22(i),rms_vel);
    hold on
    grid
end
```

```
RMS value for velocity when q22 is 0.01 = 2.21
RMS value for velocity when q22 is 0.10 = 1.28
RMS value for velocity when q22 is 1.00 = 0.67
RMS value for velocity when q22 is 10.00 = 0.28
RMS value for velocity when q22 is 100.00 = 0.10
```

```
hold off
legend('q22 = 0.01','q22 = 0.1','q22 = 1','q22 = 10','q22 = 100', location='best')
```

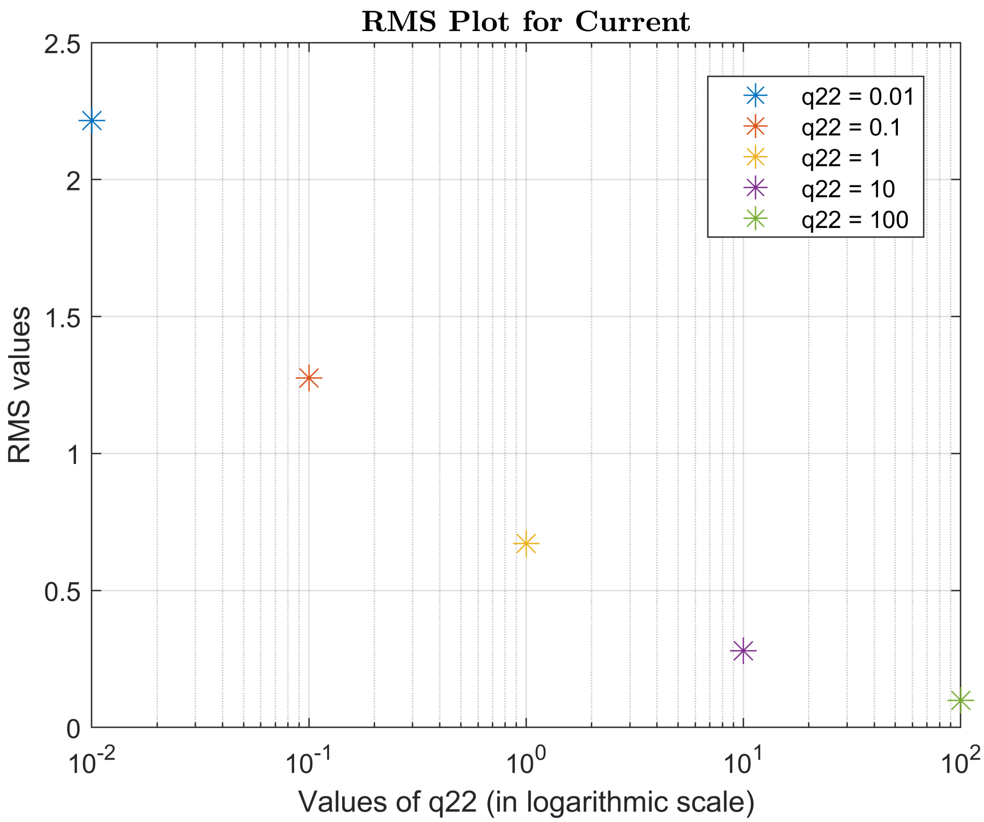


RMS plot for current (Ic)

```
for i = 1:length(q22)
    Q(2,2) = q22(i);
    K = lqr(A, B, Q, R); % Control gain for different penalties on velocity
    sys = ss(A-B*K,B,C,D); % closed loop-system
    [y, tspan, x] = initial(sys, x0, tspan);
    figure(9)
    rms_cur = rms(y(:,2));
    semilogx(q22(i),rms_cur,'*', 'MarkerSize',10);
    title('\textbf{RMS Plot for Current}', 'Interpreter','latex');
    xlabel('Values of q22 (in logarithmic scale)');
    ylabel('RMS values');
    fprintf('RMS value for current when q22 is %0.2f = %0.2f \n',q22(i),rms_cur);
    hold on
    grid
end
```

```
RMS value for current when q22 is 0.01 = 2.21
RMS value for current when q22 is 0.10 = 1.28
RMS value for current when q22 is 1.00 = 0.67
RMS value for current when q22 is 10.00 = 0.28
RMS value for current when q22 is 100.00 = 0.10
```

```
hold off
legend('q22 = 0.01','q22 = 0.1','q22 = 1','q22 = 10','q22 = 100', location='best')
```

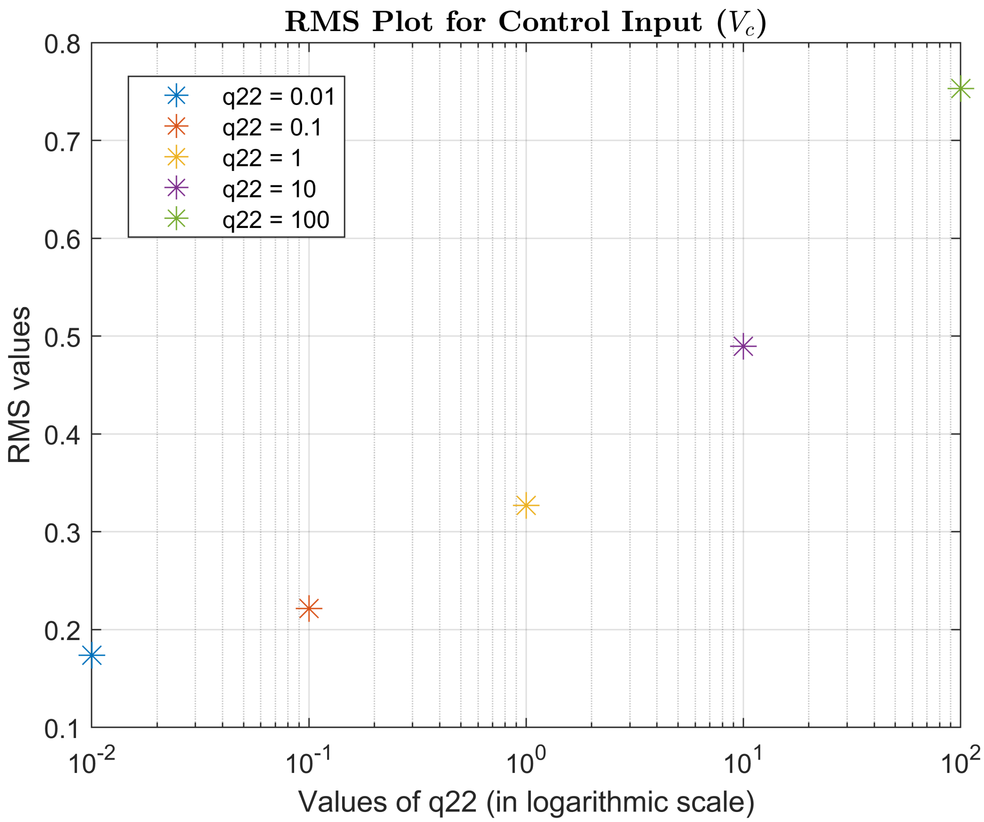


RMS plot for control input (Vc)

```
for i = 1:length(q22)
    Q(2,2) = q22(i);
    K = lqr(A, B, Q, R);
    sys = ss(A-B*K,B,C,D); % closed loop-system
    [y, tspan, x] = initial(sys, x0, tspan);
    u = - (K*x')';
    figure(10)
    rms_u = rms(u);
    semilogx(q22(i),rms_u,'*', 'MarkerSize',10);
    title('\textbf{RMS Plot for Control Input (${V_c}$)}', 'Interpreter','latex');
    xlabel('Values of q22 (in logarithmic scale)');
    ylabel('RMS values');
    fprintf('RMS value of control input when q22 is %0.2f = %0.2f \n',q22(i),rms_u);
    hold on
    grid
end
```

```
RMS value of control input when q22 is 0.01 = 0.17
RMS value of control input when q22 is 0.10 = 0.22
RMS value of control input when q22 is 1.00 = 0.33
RMS value of control input when q22 is 10.00 = 0.49
RMS value of control input when q22 is 100.00 = 0.75
```

```
hold off
legend('q22 = 0.01','q22 = 0.1','q22 = 1','q22 = 10','q22 = 100', location='best')
```



Comments on effects of changing q22

From the RMS plots above it can be inferred that as the penalty on velocity (i.e. value of q22) increases the rms value for velocity decreases implying that the state decays faster towards zero.

(e) Optimal feedback controller for system with augmented integrator

```
clc;
clear;
A = [0, 1, 0;
     8.75*10^4, 0, 7*10^4;
     0, -1.25, -597.82];
B = [0;
     0;
     271.74];
n = length(A);
C = [1 0 0];
D = 0;
Q = eye(3);
r = [1];

% Augmented matrices
Aa = [A, zeros(3,1); C, 0];
Ba = [B ; 0];
Ca = [C, 0];
```

```
sys = ss(Aa, Ba, Ca, D);

% Penalty matrix for states
Qa = [Q, zeros(3,1); zeros(1,3), 8];

% Checking for controllability
rank(ctrb(Aa,Ba)) % the rank is equal to the number of states, hence controllable
```

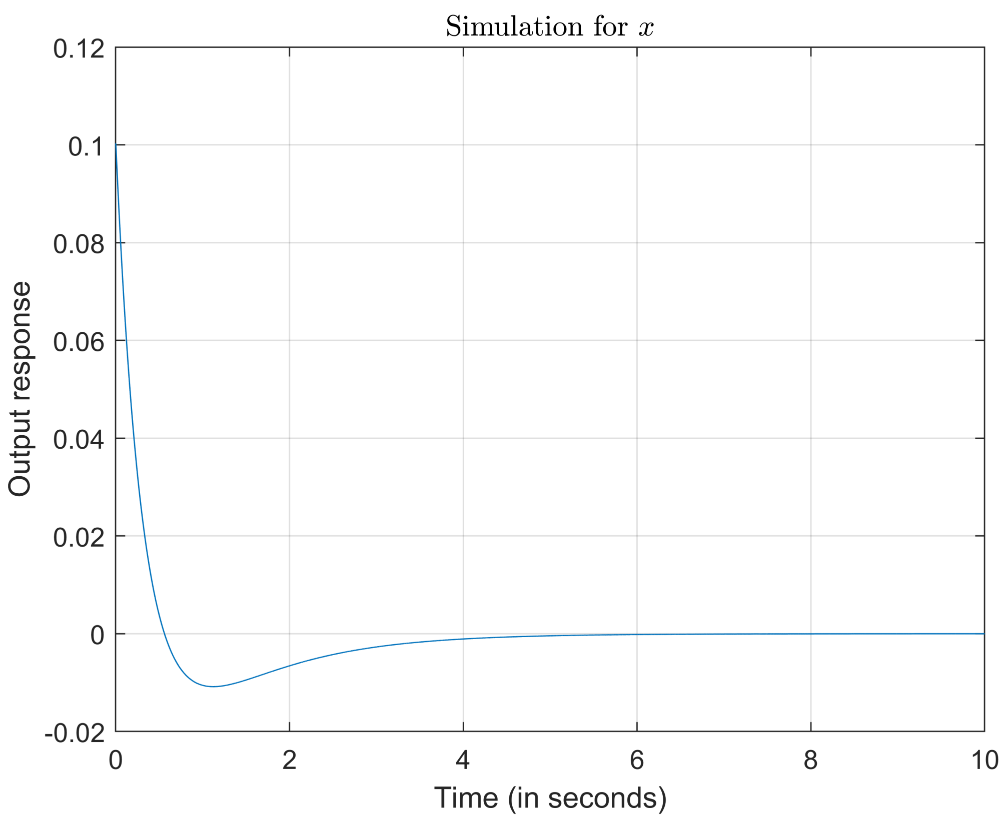
```
ans = 4
```

```
% Calculating gain matrix using lqr
[K,S,e] = lqr(Aa,Ba,Qa,r);

% Adding 0.05 constant drift to the input matrix
sys = ss(Aa-Ba*K,Ba*0.05,Ca,D); % closed-loop system

tspan = 0:0.0005:10; % time period is 10 seconds
x0 = [0.1; 0; 0; 0]';
[y, tspan, x] = initial(sys, x0, tspan);

plot(tspan,y(:,1))
title('Simulation for \textbf{${x}$}','Interpreter','latex');
xlabel('Time (in seconds)');
ylabel('Output response');
grid
```



In the plot, we can see that position state converges to zero after around 5 seconds.