- Assignment - 9

→ 1) Dead-beat observer + Pole placement observe.

Roll no :- CH16B001 , $p = 0$, $q = 1$

$$\Rightarrow A = \begin{bmatrix} 0 & -0.4 \\ 0.1 & 0.25 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Case 1 :- Dead-beat Observer

Let $L = \begin{bmatrix} a & b \end{bmatrix}^T$

$$\Rightarrow A - LC = \begin{bmatrix} -a & -0.4 \\ (0.1 - b) & 0.25 \end{bmatrix}$$

We want $\lambda_1 = \lambda_2 = 0 = eig(A - LC)$

$\Rightarrow \quad tr(A - LC) = \lambda_1 + \lambda_2 = 0$

$\Rightarrow \quad 0.25 - a = 0 \quad \Rightarrow \quad \boxed{a = 0.25}$

$\Rightarrow \quad det(A - LC) = \lambda_1 \cdot \lambda_2 = 0$

$\Rightarrow \quad -0.25a + 0.4(0.1 - b) = 0$

$\Rightarrow \quad \boxed{b = -0.05625}$

$\Rightarrow \quad \boxed{L_{dead\text{-}beat} = [0.25 \quad -0.05625]^T}$

Case-2 : Pole-placement, $\lambda_1 = 0.25, \lambda_2 = 0.4$

$\Rightarrow tr(A - LC) = 0.25 - a = \lambda_1 + \lambda_2 = 0.65$

$\Rightarrow \quad \boxed{a = -0.4}$

$L_{PP} = \begin{bmatrix} -0.4 \\ 0.1 \end{bmatrix}$

$det(A - LC) = \lambda_1 \cdot \lambda_2 = 0.1 \quad \Rightarrow \quad \boxed{b = 0.1}$

$\Rightarrow \quad -0.25a + 0.4(0.1 - b) = 0.1$

Function

Reset

```
function [Roll_No,K_db,K_pp]=hw9Prob1(Roll_No)
% Please enter your solution below
Roll_No='CH16b001';     % Please replace with your roll number
K_db= [ 0.25 -0.05625]' ;                % Please enter observer gain for deadbeat observer
K_pp= [-0.4 0.1]';                % Please enter observer gain for closed-loop poles at 0.25 and 0.4
end
```

Code to call your function

```
[RNo,L1,L2]=hw9Prob1('Test01')
```

```
function [Roll_No,K_inf,P_inf,Pbar_inf]=hw9Prob2(Roll_No)
% Please enter your solution below
Roll_No='CH16b001';     % Please replace with your roll number

% Please do your calculations below this line
p = 0;
q = 1;
A = [0.1*p -0.4;0.1*q 0.25];
nx = 2;    % number of states
C = [1 0];
M = [0.2;1.2];
cov_eps = 1;
R1 = M*cov_eps*M';
R2 = 0.25;
[X,L,G] = dare(A',C',R1,R2);
Pbar_inf = X;
Kbar_inf = A*Pbar_inf*C'*inv(C*Pbar_inf*C'+R2);
K_inf = inv(A)*Kbar_inf;
P_inf = (eye(nx)-K_inf*C)*Pbar_inf;

end
```

Code to call your function

```
[RNo,L1,L2]=hw9Prob2('Test01')
```

```m
function [Roll_No,K_kf1,Lam_1,K_kf2,Lam_2]=hw9Prob3(Roll_No)
% Please enter your solution below
Roll_No='CH16B001';     % Please replace with your roll number

% ***** Please do your calculations below this line *****

p = 0;
q = 1;
A = [0.1*p -0.4;0.1*q 0.25];
C = [1 0];
M = [0.2 1.2]';


% 1. Calculations with Q=1 and R=2.5e-5
%    Provide Kalman gain in K_kf1 and observer eigenvalues in Lam_1

cov_eps = 1;
R1 = M*cov_eps*M';
R2 = 2.5*10^-5;
[X,L,G] = dare(A',C',R1,R2);
Pbar_inf = X;
Kbar_inf = A*Pbar_inf*C'*inv(C*Pbar_inf*C'+R2);
K_kf1 = inv(A)*Kbar_inf;
Lam_1 = eig(A-Kbar_inf*C);

% 2. Calculations with Q=1e-4 and R=0.25
%    Provide Kalman gain in K_kf2 and observer eigenvalues in Lam_2
cov_eps = 1*10^-4;
R1_new = M*cov_eps*M';
R2_new = 0.25;
[X,L,G] = dare(A',C',R1_new,R2_new)
Pbar_inf_2 = X;
Kbar_inf_2 = A*Pbar_inf_2*C'*inv(C*Pbar_inf_2*C'+R2_new);
K_kf2 = inv(A)*Kbar_inf_2;
Lam_2 = eig(A-Kbar_inf_2*C);


end
```

```
function [XHAT,SSE,Roll_No]=hw9Prob4b(Roll_No)
% Simulations of a Kalman Filter
% Please enter your solution below
Roll_No='CH16b001';     % Please replace with your roll number
[XALL,YALL]=estimation_data(Roll_No);

% Please return estimates in array   XHAT
% Please return sum of square error:  SSE
% ---------------------------------------
% Please start typing from the line below
% ---------------------------------------
A = [0 -0.4;0.1 0.25];
M = [0.2 1.2]';
C = [1 0];
cov_eps = 1;
R1 = M*cov_eps*M';
R2 = 0.25;
[X,L,G] = dare(A',C',R1,R2);
Pbar_inf = X;
Kbar_inf = A*Pbar_inf*C'*inv(C*Pbar_inf*C'+R2);
K_inf = inv(A)*Kbar_inf;
XHAT_init = [0 0]';
XHAT = XHAT_init;
for i = 1:199
    XHAT_pred = A*XHAT(:,i);
    XHAT_corr = XHAT_pred + K_inf*(YALL(i+1) - C*XHAT_pred);  % @ (i+1)
    XHAT = [XHAT,XHAT_corr];
end
%% Plot the results
subplot(2,1,1)
plot(1:200,XALL(1,:), 1:200,XHAT(1,:));
subplot(2,1,2)
plot(1:200,XALL(2,:), 1:200,XHAT(2,:));

%% Compute SSE
sqErr=(XALL-XHAT).^2;
SSE=sum(sum(sqErr));
end
```