PURDUE
UNIVERSITY

College of Engineering
School of Aeronautical and Astronautical Engineering

AAE 567: Introduction To Applied Stochastic Processes

Final Exam

Professor:
A. E. Frazho
Purdue AAE Professor

Student:
Tomoki Koike
Purdue AAE Senior

May 4, 2021

Table of Contents

Problem 1

Consider the strictly positive matrix

$$T = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 3 \end{bmatrix}.$$

Using this matrix consider the inner product on $\mathbb{C}^3$ defined by

$$(x, y)_T = (Tx, y) = y'Tx \qquad (x \in \mathbb{C}^3 \text{ and } y \in \mathbb{C}^3)$$

where $'$ denotes the complex conjugate transpose. The norm of this inner product is given by $\|x\|_T = \sqrt{(Tx, x)}$. Let

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \text{and} \quad e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Solve the following optimization problem

$$\begin{aligned} \delta &= min\{\|e_1 - \alpha e_2 - \beta e_3\|_T^2 : \alpha \in \mathbb{C} \text{ and } \beta \in \mathbb{C}\} \\ &= min\{(T(e_1 - \alpha e_2 - \beta e_3), (e_1 - \alpha e_2 - \beta e_3)) : \alpha \in \mathbb{C} \text{ and } \beta \in \mathbb{C}\}. \end{aligned}$$

In other words, find δ and scalars α and β such that

$$\|e_1 - \alpha e_2 - \beta e_3\|_T^2 = \delta = min\{\|e_1 - \alpha e_2 - \beta e_3\|_T^2 : \alpha \in \mathbb{C} \text{ and } \beta \in \mathbb{C}\}.$$

Solution:
From the principle of projection theorem we know that

$$e_1 - \alpha e_2 - \beta e_3 \perp e_2, e_3.$$

Then using the inner product we have

$$\begin{aligned} (e_1 - \alpha e_2 - \beta e_3, e_2)_T = 0 \\ (e_1 - \alpha e_2 - \beta e_3, e_3)_T = 0 \end{aligned}$$

which becomes

$$\begin{aligned} (e_1, e_2)_T = \alpha(e_2, e_2)_T + \beta(e_3, e_2)_T \\ (e_1, e_3)_T = \alpha(e_2, e_3)_T + \beta(e_3, e_3)_T \end{aligned}$$

$$\begin{bmatrix} (e_1, e_2)_T \\ (e_1, e_3)_T \end{bmatrix} = \underbrace{\begin{bmatrix} (e_2, e_2)_T & (e_3, e_2)_T \\ (e_2, e_3)_T & (e_3, e_3)_T \end{bmatrix}}_{G} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.$$

Then we can find α and β by

$$\begin{bmatrix}\alpha\\ \beta\end{bmatrix} = G^{-1}\begin{bmatrix}(e_1, e_2)_T\\ (e_1, e_3)_T\end{bmatrix}.$$

Then,

$$\begin{aligned} G_{11} &= (e_2, e_2)_T = e2'Te_2 \\ &= \begin{bmatrix}0 & 1 & 0\end{bmatrix}\begin{bmatrix}3 & 2 & 1\\ 2 & 3 & 2\\ 1 & 2 & 3\end{bmatrix}\begin{bmatrix}0\\ 1\\ 0\end{bmatrix} \\ &= \begin{bmatrix}2 & 3 & 2\end{bmatrix}\begin{bmatrix}0\\ 1\\ 0\end{bmatrix} \\ &= 3. \end{aligned}$$

Similarly,

$$G_{12} = 2 \qquad G_{21} = 2 \qquad G_{22} = 3$$

and therefore,

$$G = \begin{bmatrix}3 & 2\\ 2 & 3\end{bmatrix}.$$

Also,

$$\begin{aligned} (e_1, e_2)_T &= e_2'Te_1 = 2 \\ (e_1, e_3)_T &= e_3'Te_1 = 1 \end{aligned}$$

and now we can compute the coefficients as

$$\begin{aligned} \begin{bmatrix}\alpha\\ \beta\end{bmatrix} &= \begin{bmatrix}3 & 2\\ 2 & 3\end{bmatrix}^{-1}\begin{bmatrix}2\\ 1\end{bmatrix} \\ &= \begin{bmatrix}0.8\\ -0.2\end{bmatrix}. \end{aligned}$$

Thus,

$$\alpha = 0.8 \qquad \beta = -0.2.$$

Next, we calculate

$$e_1 - \alpha e_2 - \beta e_3 = \begin{bmatrix}1\\ -0.8\\ 0.2\end{bmatrix}.$$

Hence,

$$
\begin{aligned}
\delta &= min\{(T(e_1 - \alpha e_2 - \beta e_3), (e_1 - \alpha e_2 - \beta e_3)) : \alpha \in \mathbb{C} \text{ and } \beta \in \mathbb{C}\} \\
&= \begin{bmatrix} 1 & -0.8 & 0.2 \end{bmatrix} \begin{bmatrix} 3 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ -0.8 \\ 0.2 \end{bmatrix}
\end{aligned}
$$

and we obtain

$$\delta = 1.6.$$

Problem 2

Consider the discrete time system

$$\begin{bmatrix} x_1(n+1) \\ x_2(n+1) \end{bmatrix} = \begin{bmatrix} e^{-\frac{n}{50}} & 1 \\ 2 & \cos(\frac{n}{50}) \end{bmatrix} \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} + \begin{bmatrix} e^{-\frac{n}{50}} \cos(\frac{n}{50}) \\ 1 \end{bmatrix} u(n)$$

$$y(n) = \begin{bmatrix} 1 + e^{-\frac{n}{50}} & 2 + \sin(\frac{n}{50}) \end{bmatrix} \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} + (1 + \frac{1}{2}\sin(\frac{n}{50}))v(n)$$

where u and v are independent Gaussian white noise processes. The initial conditions $x(0) = 0$ and $\hat{x}(0) = 0$. To generate u and v in MATLAB, set

$$\begin{aligned} &rng(1000); \qquad u = randn(1,20); \\ &rng(2000); \qquad v = randn(1,20); \end{aligned}$$

Let $\mathcal{M}_n = span\{y(j)\}_0^n$. Find the following

(i) $P_{\mathcal{M}_{n-1}} x_1(n)$ for n = 8, 9, 10.

(ii) $P_{\mathcal{M}_n} x_2(n)$ for n = 8, 9, 10.

(Note the indices on the state $x_1(n)$ in Part (i), and $x_2(n)$ in Part (ii).) Be careful MATLAB does not have a zero index. So for example, in MATLAB

$$A(0) = A\{1\} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}$$

$$A(1) = A\{2\} = \begin{bmatrix} e^{-\frac{n}{50}} & 1 \\ 2 & \cos(\frac{n}{50}) \end{bmatrix}$$

etc.

Solution:
The time varying discrete system matrices are

$$A(n) = \begin{bmatrix} e^{-\frac{n}{50}} & 1 \\ 2 & \cos(\frac{n}{50}) \end{bmatrix} \qquad B(n) = \begin{bmatrix} e^{-\frac{n}{50}} \cos(\frac{n}{50}) \\ 1 \end{bmatrix}$$

$$C(n) = \begin{bmatrix} 1 + e^{-\frac{n}{50}} & 2 + \sin(\frac{n}{50}) \end{bmatrix} \quad D(n) = 1 + \frac{1}{2}\sin(\frac{n}{50}).$$

and the system state is

$$x(n) = \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix}.$$

From discrete time Kalman filter we know that if

$$\begin{aligned} \phi(n) &= y(n) - P_{\mathcal{M}_{n-1}} y(n) \\ &= y(n) - C(n)\hat{x}(n) \\ &= C(n)\tilde{x}(n) + D(n)v(n) \end{aligned}$$

$$\begin{aligned} \hat{x}(n+1) &= P_{\mathcal{M}_n} x(n+1) \\ &= P_{\mathcal{M}_n}(A(n)x(n) + B(n)u(n)) \\ &= A(n)P_{\mathcal{M}_n} x(n) \\ &= A(n)\hat{x}(n) + AR_{x(n)\phi(n)} R_{\phi(n)}^{-1}(y(n) - C(n)\hat{x}(n)) \end{aligned}$$

where

$$\begin{aligned} R_{x(n)\phi(n)} &= Q(n)C(n)^* \\ R_{\phi(n)} &= C(n)Q(n)C(n)^* + D(n)D(n)^*. \end{aligned}$$

Hence, if $\Delta(n) = A(n)Q(n)C(n)^* \Big(C(n)Q(n)C(n)^* + D(n)D(n)^* \Big)^{-1}$

$$\begin{aligned} \hat{x}(n+1) &= P_{\mathcal{M}_n} x(n+1) \\ &= A(n)\hat{x}(n) + \Delta(n)\Big(y(n) - C(n)\hat{x}(n) \Big) \end{aligned}$$

and

$$Q(n+1) = A(n)Q(n)A(n)^* + B(n)B(n)^* - \Delta(n)C(n)Q(n)A(n)^*$$

(i) For the first part of the problem, we know from the theory above that we must compute $\hat{x}_1(8)$, $\hat{x}_1(9)$, and $\hat{x}_1(10)$. We are able to do this with the following MATLAB code.

Listing 1: Problem 2 part (i) MATLAB code

```
% AAE 567 Final Exam Spring 2021 Problem 2 Part (i)
% Tomoki Koike

% Housekeeping commands
clear all; close all; clc;
%%
getA = @(n) [exp(-n/50), 1; 2, cos(n/50)];
```

```
getB = @(n) [exp(-n/50)*cos(n/50); 1];
getC = @(n) [1+exp(-n/50), 2+sin(n/50)];
getD = @(n) 1 + 0.5*sin(n/50);

% Initialize cells to store matrices
A = {}; B = {}; C = {}; D = {};

% Initialize the covariance matrix Q
Q = {}; Q{1} = zeros(2,2);

% Initialize the x-states and xhat-states
x = {}; x{1} = zeros(2,1);
xhat = {}; xhat{1} = zeros(2,1);

% Initialize the u(n) and v(n) white noise
rng(1000); u = randn(1,20);
rng(2000); v = randn(1,20);

% Initialize the output states y(n)
y = {};

for n = 0:10
    A{n+1} = getA(n);
    B{n+1} = getB(n);
    C{n+1} = getC(n);
    D{n+1} = getD(n);
    % Compute x(n+2) which is actually x(n+1)
    x{n+2} = A{n+1}*x{n+1} + B{n+1}*u(n+1);
    % Compute y(n)
    y{n+1} = C{n+1} * x{n+1} + D{n+1}*v(n+1);
    [xhat{n+2}, Q{n+2}] = dkf(A{n+1}, B{n+1}, C{n+1}, D{n+1}, ...
                                Q{n+1}, xhat{n+1}, y{n+1});
end
%%
function [Xnew, Qnew] = dkf(Ad, Bd, Cd, Dd, Qd, xhat, y)
    % Discrete time Kalman filter
    Del = Ad*Qd*Cd' * inv(Cd*Qd*Cd' + Dd*Dd');
    Xnew = Ad*xhat + Del*(y - Cd*xhat);
    Qnew = Ad*Qd*Ad' + Bd*Bd' - Del*Cd*Qd*Ad';
end
```

This gives us the following results

$$\hat{x}_1(8) = -31.5388 \qquad \hat{x}_1(9) = -75.9468 \qquad \hat{x}_1(10) = -174.4413$$

(ii) For the second part of the problem we modify part (i) so that we obtain

$$\begin{aligned} P_{\mathcal{M}_n} x(n) &= \hat{x}(n) + R_{x(n)\phi(n)} R_{\phi(n)}^{-1} \phi(n) \\ &= \hat{x}(n) + Q(n)C(n)^* \Big(C(n)Q(n)C(n)^* + D(n)D(n)^* \Big)^{-1} \Big(y(n) - C(n)\hat{x}(n) \Big) \end{aligned}$$

and the MATLAB code in Listing 2 will give us these values.

Listing 2: Problem 2 part (ii) MATLAB code

```
% AAE 567 Final Exam Spring 2021 Problem 2 Part (ii)
% Tomoki Koike

% Housekeeping commands
clear all; close all; clc;
%%
getA = @(n) [exp(-n/50), 1; 2, cos(n/50)];
getB = @(n) [exp(-n/50)*cos(n/50); 1];
getC = @(n) [1+exp(-n/50), 2+sin(n/50)];
getD = @(n) 1 + 0.5*sin(n/50);

% Initialize cells to store matrices
A = {}; B = {}; C = {}; D = {};

% Initialize the covariance matrix Q
Q = {}; Q{1} = zeros(2,2);

% Initialize the x-states and xhat-states
x = {}; x{1} = zeros(2,1);
xhat = {}; xhat{1} = zeros(2,1);

% Initialize the u(n) and v(n) white noise
rng(1000); u = randn(1,20);
rng(2000); v = randn(1,20);

% Initialize the output states y(n)
y = {};

% Initialize the value we are looking for
res = {};
```

```

for n = 0:10
    A{n+1} = getA(n);
    B{n+1} = getB(n);
    C{n+1} = getC(n);
    D{n+1} = getD(n);
    % Compute x(n+2) which is actually x(n+1)
    x{n+2} = A{n+1}*x{n+1} + B{n+1}*u(n+1);
    % Compute y(n)
    y{n+1} = C{n+1} * x{n+1} + D{n+1}*v(n+1);
    [xhat{n+2}, Q{n+2}, res{n+1}] = dkf(A{n+1}, B{n+1}, C{n+1}, D{n+1}, ...
                                        Q{n+1}, xhat{n+1}, y{n+1});
end
%%
function [Xnew, Qnew, XX] = dkf(Ad, Bd, Cd, Dd, Qd, xhat, y)
    % Discrete time Kalman filter
    Del = Ad*Qd*Cd' * inv(Cd*Qd*Cd' + Dd*Dd');
    Xnew = Ad*xhat + Del*(y - Cd*xhat);
    Qnew = Ad*Qd*Ad' + Bd*Bd' - Del*Cd*Qd*Ad';

    % What we are looking for
    XX = xhat + Qd*Cd' * inv(Cd*Qd*Cd' + Dd*Dd')*(y - Cd*xhat);
end
```

This gives us the following results

$$P_{\mathcal{M}_n} x_2(8) = -48.0371 \qquad P_{\mathcal{M}_n} x_2(9) = -111.7608 \qquad P_{\mathcal{M}_n} x_2(10) = -260.9925$$

Problem 3

Let x be a mean zero, variance one, Gaussian random variable, and $\{v_n\}_0^\infty$ be a mean zero, variance one, Gaussian white noise process, which is independent of x. Consider the discrete time random process y_n defined by

$$y_n = x + v_n$$

where $n \geq 0$ is a positive integer.

(i) Find best estimate for x given $\{y_j\}_{j=0}^{n-1}$, that is, find

$$\hat{x}_n = E(x|y_0, y_1, \cdots, y_{n-1}) = P_{\mathcal{M}_{n-1}} x$$

where $\mathcal{M}_{n-1} = span\{y_j\}_{j=0}^{n-1}$.

(ii) Find the error σ_n in your estimate, that is

$$\sigma_n^2 = E(x - \hat{x}_n)^2$$

Solution:
(i) We are given that

$$Ex = Ev_n = 0 \qquad Ex^2 = Ev_n^2 = 1.$$

If we let

$$g = \begin{bmatrix} 1 \\ y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix}$$

then,

$$\hat{x}_n = R_{xg} R_g^{-1} g.$$

Now,

$$\begin{aligned} R_{xg} &= Ex \begin{bmatrix} 1 & y_0 & y_1 & \cdots & y_{n-1} \end{bmatrix} \\ &= \begin{bmatrix} Ex & Exy_0 & Exy_1 & \cdots & Exy_{n-1} \end{bmatrix}. \end{aligned}$$

Since,

$$Exy_m = Ex(x+v_m) = Ex^2 + \cancelto{0}{ExEv_m} = 1.$$

Thus,

$$R_{xg} = \begin{bmatrix} 0 & 1 & 1 & \cdots & 1 \end{bmatrix}_{1\times(n+1)}.$$

Then we consider,

$$\begin{aligned} Ey_ny_m &= E(x+v_n)(x+v_m) \\ &= Ex^2 + \cancelto{0}{Ev_nEx} + \cancelto{0}{Ev_mEx} + Ev_nv_m \end{aligned}$$

and here

$$Ev_nv_m = \begin{cases} 1 & \text{if} \quad n = m \\ 0 & \text{if} \quad n \neq m \end{cases}.$$

Therefore,

$$Ey_ny_m = \begin{cases} 2 & \text{if} \quad n = m \\ 1 & \text{if} \quad n \neq m \end{cases}.$$

Then,

$$\begin{aligned} R_g = Eg &= \begin{bmatrix} 1 \\ y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix} \begin{bmatrix} 1 & y_0 & y_1 & \cdots & y_{n-1} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 2 & 1 & \cdots & 1 \\ 0 & 1 & 2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 1 \\ 0 & 1 & \cdots & 1 & 2 \end{bmatrix}_{(n+1)\times(n+1)}. \end{aligned}$$

Listing 3: Problem 3 MATLAB code

```
% AAE 567 Final Exam Spring 2021 Problem 3
% Tomoki Koike

% Housekeeping commands
clear all; close all; clc;
%%
% Problem 3
Rg = [];
sz = 4;
for i = 1:sz
    for j = 1:sz
        if (i==1) && (j==1)
            Rg(i,j) = 1;
        elseif (i==1) || (j==1)
            Rg(i,j) = 0;
        elseif i==j
            Rg(i,j) = 2;
        else
            Rg(i,j) = 1;
        end
    end
end
Rxg = ones(1,sz);
Rxg(1) = 0;
inv(Rg)
coefs = Rxg * inv(Rg)
```

Now running the MATLAB code shown in Listing 3, we can numerically deduce that

$$R_g^{-1} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & \frac{n}{n+1} & -\frac{1}{n+1} & \cdots & -\frac{1}{n+1} \\ 0 & -\frac{1}{n+1} & \frac{n}{n+1} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -\frac{1}{n+1} \\ 0 & -\frac{1}{n+1} & \cdots & -\frac{1}{n+1} & \frac{n}{n+1} \end{bmatrix}_{(n+1)\times(n+1)} .$$

Then,

$$R_{xg}R_g^{-1} = \begin{bmatrix}0 & 1 & 1 & \cdots & 1\end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & \frac{n}{n+1} & -\frac{1}{n+1} & \cdots & -\frac{1}{n+1} \\ 0 & -\frac{1}{n+1} & \frac{n}{n+1} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -\frac{1}{n+1} \\ 0 & -\frac{1}{n+1} & \cdots & -\frac{1}{n+1} & \frac{n}{n+1} \end{bmatrix}$$

$$= \begin{bmatrix}0 & h & h & \cdots & h\end{bmatrix}$$

where

$$\begin{aligned} h &= \frac{n}{n+1} + \left(-\frac{1}{n+1}\right)(n-1) \\ &= \frac{n}{n+1} - \frac{n-1}{n+1} \\ &= \frac{1}{n+1}. \end{aligned}$$

Therefore,

$$\hat{x}_n = \begin{bmatrix}0 & \frac{1}{n+1} & \frac{1}{n+1} & \cdots & \frac{1}{n+1}\end{bmatrix} \begin{bmatrix} 1 \\ y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix}$$

and

$$\hat{x}_n = \frac{1}{n+1}(y_0 + y_1 + \cdots + y_{n-1}).$$

(ii) Next we will find the error.

$$
\begin{aligned}
E\hat{x}_n^2 &= \frac{1}{(n+1)^2} E(y_0 + y_1 + \cdots + y_{n-1})^2 \\
&= \frac{1}{(n+1)^2}\left[\sum_{i=0}^{n-1} Ey_i^2 + \sum_{i=0,i>j}^{n-1}\sum_{j=0}^{n-1} 2Ey_iy_j\right] \\
&= \frac{1}{(n+1)^2}\left[\sum_{i=0}^{n-1} 2 + 2\binom{n}{2}\right] \\
&= \frac{2}{(n+1)^2}\left(n + \frac{n!}{(n-2)!2!}\right) \\
&= \frac{2}{(n+1)^2}\left(n + \frac{n(n-1)}{2}\right) \\
&= \frac{1}{(n+1)^2}\Big[n(n+1)\Big] \\
&= \frac{n}{n+1}.
\end{aligned}
$$

Finally,

$$
E(x-\hat{x})^2 = Ex^2 - E\hat{x}^2 = 1 - \frac{n}{n+1} = \frac{1}{n+1}.
$$

Hence,

$$
\sigma_n^2 = \frac{1}{n+1}.
$$

Problem 4

Consider the unstable state space system given by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.2 & 3 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.5 & 30 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 5 \end{bmatrix} w + \begin{bmatrix} 0 & 0 \\ 0.25 & 0 \\ 0 & 0 \\ 0 & 0.1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\begin{aligned} y_1 &= x_1 + 0.25 v_1 \\ y_2 &= x_3 + 0.05 v_2 \end{aligned}$$

where u_1, u_2, v_1 and v_2 are all independent white noise processes. Moreover, w is the input. Assume that all the initial conditions are zero. Design a feedback controller $w = -K\hat{x}$ based on the steady state Kalman filter such that $|x_1(t)| \leq 1$ and $|x_3(t)| \leq 0.35$. Your state feedback gain $K = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix}$ must statisfy $|k_j| \leq 25$ for $j = 1, 2, 3, 4$. Simulate you controller in SIMULINK for 30 seconds. Hand the graphs from your SIMULINK program for:

(i) The state x_1 and its estimate $\hat{x}_1$ on the same graph.

(ii) The state x_2 and its estimate $\hat{x}_2$ on the same graph.

(iii) The state x_3 and its estimate $\hat{x}_3$ on the same graph.

(iv) The state x_4 and its estimate $\hat{x}_4$ on the same graph.

(v) Hand in your gain K.

On the bank limited white noise generators, set the seed for u_1, u_2, v_1, and v_2 respectively to, 22341, 23342, 23343, and 23344.

Solution:
The system matrices are defined to be

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.2 & 3 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.5 & 30 & 0 \end{bmatrix} \qquad B_0 = \begin{bmatrix} 0 & 0 \\ 0.25 & 0 \\ 0 & 0 \\ 0 & 0.1 \end{bmatrix} \qquad B_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 5 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad D = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}.$$

Now in MATLAB using the `lqr()` command we obtain the following gain matrix K

$$K = \begin{bmatrix} -3.1623 & -7.3031 & 24.8433 & 6.9672 \end{bmatrix}.$$

This gain satisfies the condition of having magnitudes smaller than 25. Then since the system is observable and controllable we can use the `are()` to find a positive definite matrix P which is a solution to an algebraic Ricatti equation.

$$P = \begin{bmatrix} 0.0792 & 0.0505 & 0.0013 & 0.0025 \\ 0.0505 & 0.0771 & 0.0133 & 0.0668 \\ 0.0013 & 0.0133 & 0.0273 & 0.1490 \\ 0.0025 & 0.0668 & 0.1490 & 0.8152 \end{bmatrix}.$$

Then the observer gain L becomes $L = PC^*(DD^*)^{-1}$ which is as follows.

$$L = \begin{bmatrix} 1.2671 & 0.5179 \\ 0.8081 & 5.3104 \\ 0.0207 & 10.9191 \\ 0.0400 & 59.6184 \end{bmatrix}$$

Below we can see that the negative real eigenvalues indicate a stable system with the controller.

$$eig(A - B_1K) \bigcup eig(A - LC) = \begin{bmatrix} -23.9934 \\ -0.8102 \\ -1.4647 + 0.5430\,\mathrm{i} \\ -1.4647 - 0.5430\,\mathrm{i} \\ -0.7126 + 0.7022\,\mathrm{i} \\ -0.7126 - 0.7022\,\mathrm{i} \\ -5.4805 + 0.1759\,\mathrm{i} \\ -5.4805 - 0.1759\,\mathrm{i} \end{bmatrix}$$

Next we define new system matrices for the states and the estimations. The matrices for the actual states become

$$\begin{aligned} A_s &= A & B_s &= [B_0, B_1] \\ C_s &= eye(4) & D_s &= zeros(size(C_s, 1), size(B_s, 2)) \end{aligned}$$

and for the estimation

$$\begin{aligned} A_{kf} &= A - LC & B_{kf} &= [L, B_1] \\ C_{kf} &= eye(4) & D_{kf} &= zeros(size(C_{kf}, 1), size(B_{kf}, 2)). \end{aligned}$$

Based on this we create the following SIMULINK model in Fig 1 for the simulation

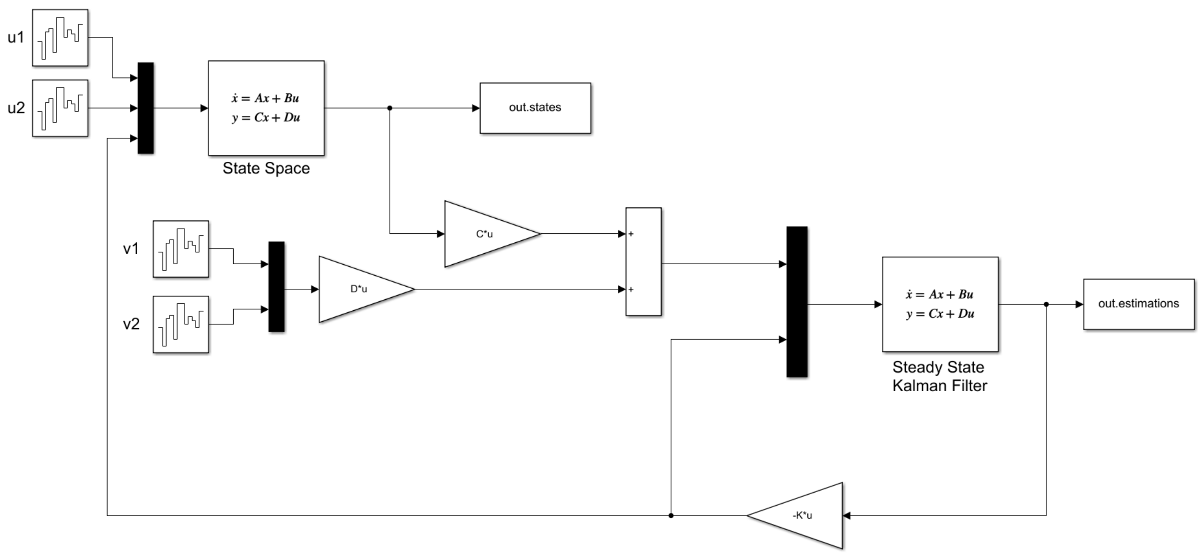

Figure 1: Simulink model for problem 4

Then we run the following MATLAB code.

Listing 4: Problem 4 MATLAB code

```
% AAE 567 Final Exam Spring 2021 Problem 4
% Tomoki Koike

% Housekeeping commands
clear all; close all; clc;
%%
% Define system matrices
A = [0 1 0 0; 0 -0.2 3 0; 0 0 0 1; 0 -0.5 30 0];
B0 = [0 0; 0.25 0; 0 0; 0 0.1];
B1 = [0; 1; 0; 5];
C = [1 0 0 0; 0 0 1 0];
D = [0.25 0; 0 0.05];

% Obtain the K gain with LQR
K = lqr(A, B1, diag([1,2,4,2]), diag([0.1]));
sym(K)

% Find L gain
rank(obsv(A,C))
rank(ctrb(A,B0))
P = are(A', C'*inv(D*D')*C, B0*B0');
```

```
sym(P)
L = P*C'*inv(D*D');
sym(L)

% Check the eigenvalues
sym([eig(A-B1*K); eig(A-L*C)])

% System matrices for the actual states
As = A;
Bs = [B0, B1];
Cs = eye(4);
Ds = zeros(size(Cs,1), size(Bs,2));

% System matrices for the steady state Kalman filter states
Akf = A - L*C;
Bkf = [L, B1];
Ckf = eye(4);
Dkf = zeros(size(Ckf,1), size(Bkf,2));
%%
% Plotting results
set(groot, 'defaulttextinterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');

% Simulate
simout = sim('final_p4_ssKF.slx');

% Data rendering
xs = simout.states.signals.values;
xkf = simout.estimations.signals.values;
t  = simout.tout;

% Plot
fig = figure("Renderer","painters","Position",[60 60 900 1000]);
    subplot(4,1,1)
    plot(t, xs(:,1))
    grid on; grid minor; box on; hold on;
    plot(t, xkf(:,1))
    hold off;
    ylabel('$x_1$')
    legend('states', 'estimate')

    subplot(4,1,2)
    plot(t, xs(:,2))
    grid on; grid minor; box on; hold on;
```

```
    plot(t, xkf(:,2))
    hold off;
    ylabel('$x_2$')

    subplot(4,1,3)
    plot(t, xs(:,3))
    grid on; grid minor; box on; hold on;
    plot(t, xkf(:,3))
    hold off;
    ylabel('$x_3$')

    subplot(4,1,4)
    plot(t, xs(:,4))
    grid on; grid minor; box on; hold on;
    plot(t, xkf(:,4))
    hold off;
    ylabel('$x_4$')
    xlabel('time [sec]')
saveas(fig, 'final_p4_plot.png')
```

And this yields the results on the next page.

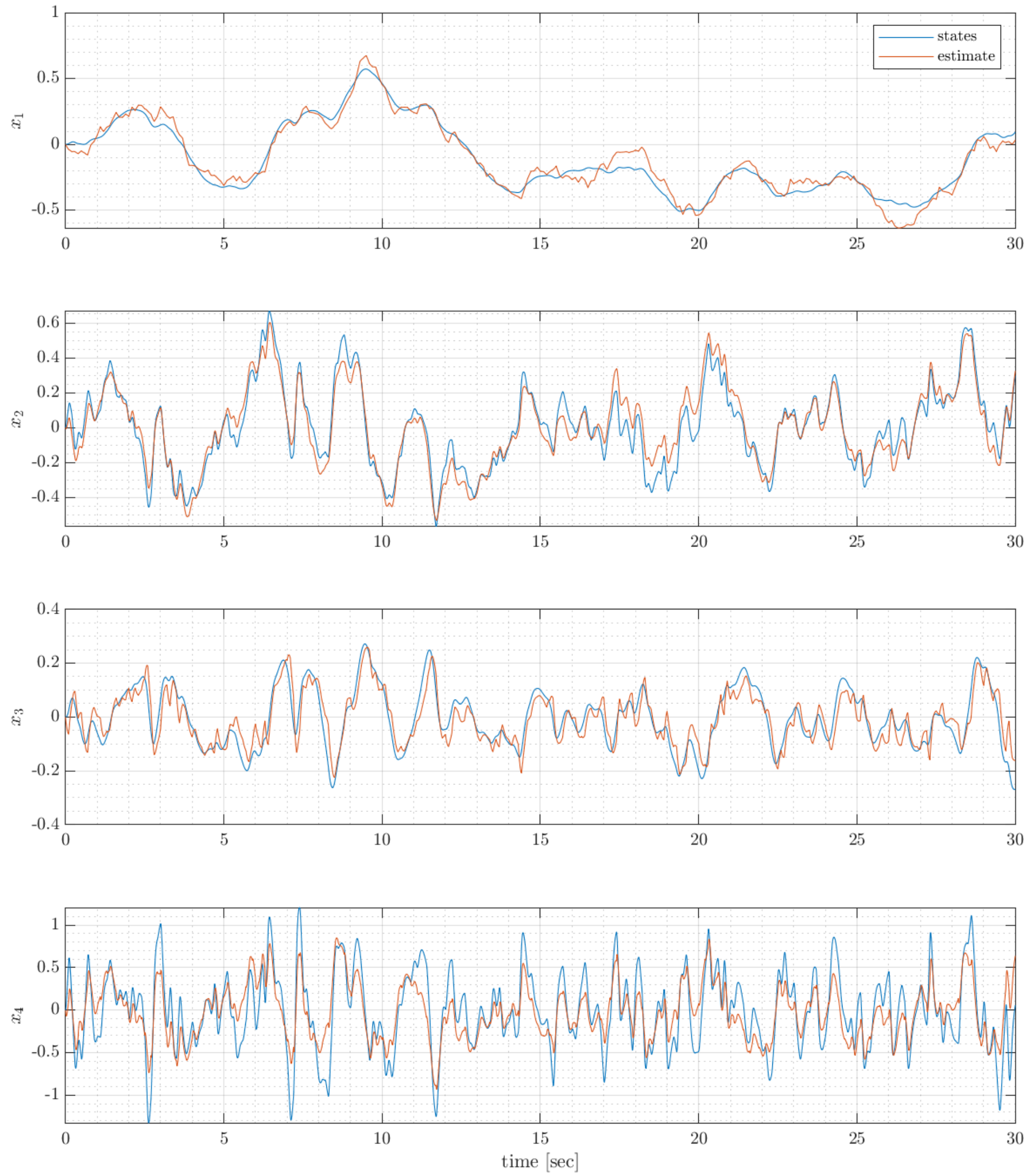

Figure 2: States and Kalman filter estimations for all 4 states