Dynamic Systems Control Theory

Tues. 10:30-12:00

Mr. Nakanishi

Graduate School of Engineering
Mechanical Engineering
April, 2017
Student Number：1030-29-9698
Lecture Number: 37
Mei Jie

Topic. System and Parameters

a) State Space System

$$\frac{d}{dt}x = Ax + Bu + Gd \qquad y = Cx + v$$

$$x = \begin{bmatrix} \beta \\ p \\ r \\ \emptyset \end{bmatrix} \left(x_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right) \qquad \mathrm{u} = \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad \mathrm{d} = \text{ random} \quad \mathrm{v} = \text{random}$$

β	angle of the sideslip	$\emptyset$	angle of bank
p	speed of the roll	r	speed of the yaw
δ_a	angle of the wing	δ_r	angle of rudder
d	disturbance to system	v	disturbance to output

b) System Parameters

$\mathrm{g} = 9.8\ \mathrm{m}/s^2 \quad \mathrm{v_{sound}} = 340.29\ m/s \quad \mathrm{U} = 0.8 \times \mathrm{v_{sound}}$											
Y_v	-0.12	Y_{δ_r}	3.13	L_β	-4.12	L_p	-0.974	L_r	2.92	L_{δ_a}	0.310
L_{δ_r}	0.183	N_β	1.62	N_p	-0.0157	N_r	-0.232	N_{δ_a}	0.0127	N_{δ_r}	-0.922

c) System Matrices

$$\mathrm{A} = \begin{bmatrix} Y_v & 0 & -1 & g/U \\ L_\beta & L_p & L_r & 0 \\ N_\beta & N_p & N_r & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad \mathrm{B} = \begin{bmatrix} 0 & Y_{\delta_r}/U \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \\ 0 & 0 \end{bmatrix} \quad \mathrm{G} = \begin{bmatrix} Y_{\delta_r}/U \\ L_{\delta_r} \\ N_{\delta_r} \\ 0 \end{bmatrix} \quad \mathrm{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Problem 1. Kinematical Simulation

1) Controllability and Observability

$$U_c = \begin{bmatrix} 0.0000 & 0.0115 & -0.0127 & 0.9206 & 0.0205 & -0.3335 & 0.0009 & -1.4237 \\ 0.3100 & 0.18300 & -0.2982 & -0.4948 & 0.3405 & -3.2439 & -0.4202 & 4.9560 \\ 0.0127 & -0.9220 & -0.0078 & 0.2296 & -0.0141 & 1.4459 & 0.0311 & -0.8248 \\ 0.00000 & 0.0000 & 0.3100 & 0.18300 & -0.2982 & -0.4948 & 0.3405 & -3.2440 \end{bmatrix}$$

$$U_o = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 \\ -0.1200 & 0.0000 & -1.0000 & 0.0360 \\ 0.0000 & 1.0000 & 0.0000 & 0.0000 \\ -1.6056 & 0.0517 & 0.3520 & -0.0043 \\ -4.1200 & -0.9740 & 0.2920 & 0.0000 \\ 0.5499 & 0.0602 & 1.5390 & -0.0578 \\ 4.9803 & 0.9440 & 3.7678 & -0.1483 \end{bmatrix}^T$$

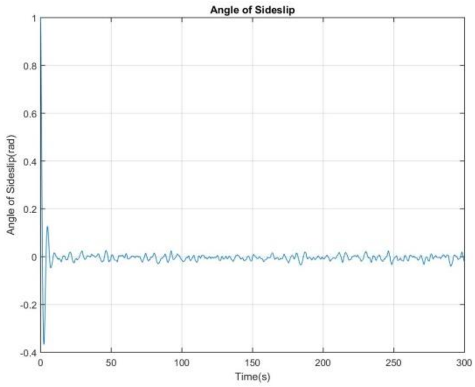
$$\mathrm{Rank}(U_c) = \mathrm{n} = 4 \qquad \mathrm{Rank}(U_o) = \mathrm{n} = 4$$

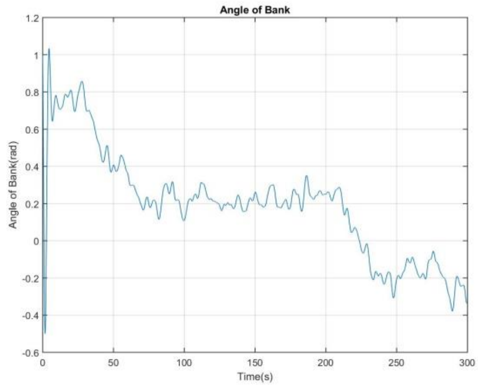
2) Stability

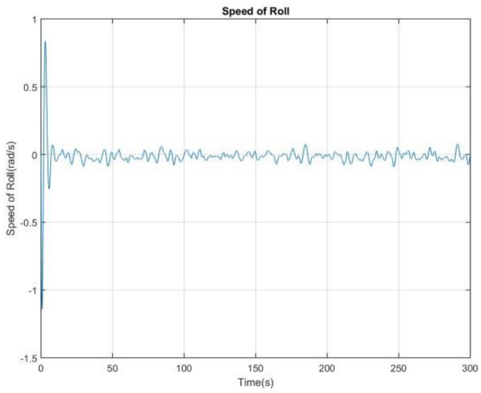
$$\lambda = \begin{bmatrix} -0.1309 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & -0.1309 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & -1.0544 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & -0.0010 \end{bmatrix}$$

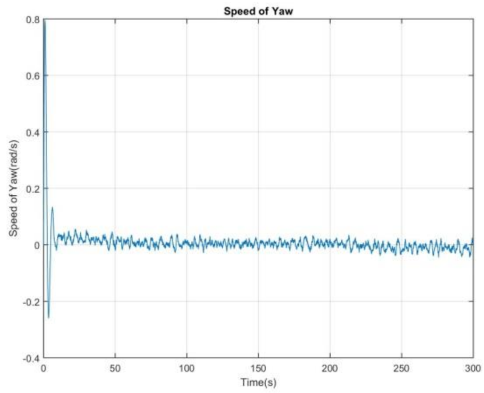
$$\upsilon = \begin{bmatrix} -0.1662 & -0.1662 & -0.0133 & 0.0048 \\ 0.7154 & 0.7154 & -0.7255 & -0.0096 \\ -0.2793 & 0.0000 & -0.0123 & 0.0354 \\ -0.0549 & -0.0549 & 0.6880 & 0.9993 \end{bmatrix}$$

3) Kinematical Simulation









4) Conclusion

Given that the controllability matrix and observability matrix are full of rank, it can be stated that this system is controllable and observable. Besides, the stability can be claimed in accord with that all the real part of eigenvalue are negative, which can be proved by the sequential kinematical simulation as well. As for the angle of the bank, the reason why it seems not convergent into zero locates at the limited range of time.

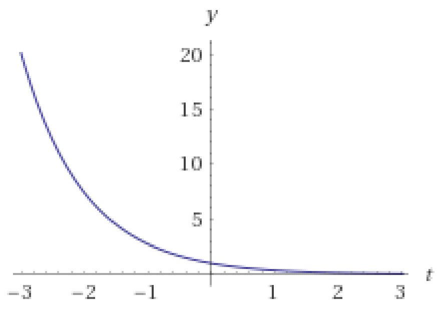


Fig.1 $y = e^{-x}$

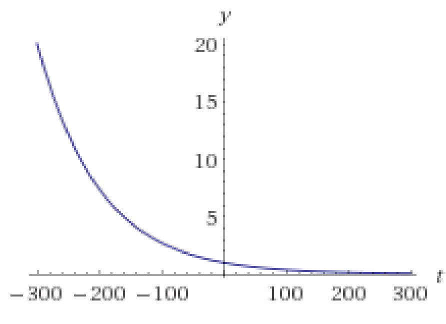


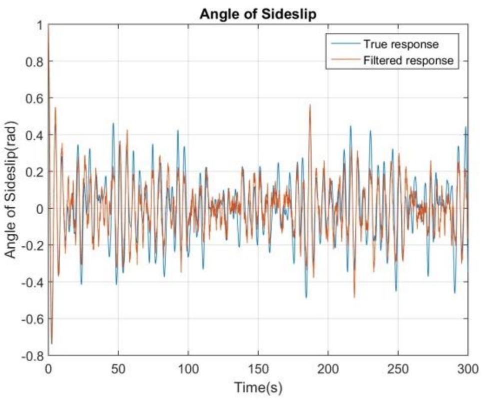
Fig.2 $y = e^{-0.01x}$ [1]

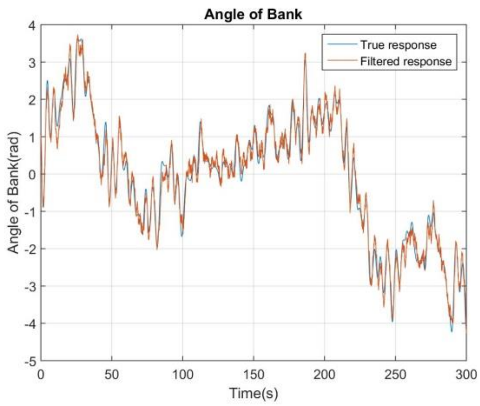
Problem 2. Kalman Filter

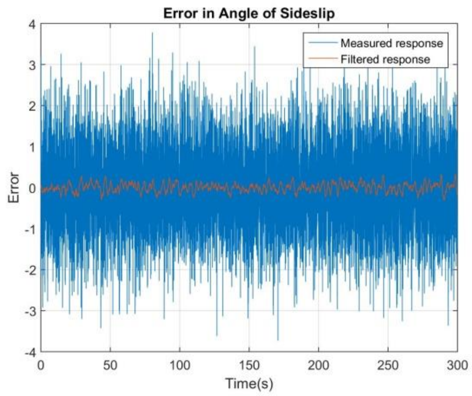
1) Kalman Gain and The solution of ARE

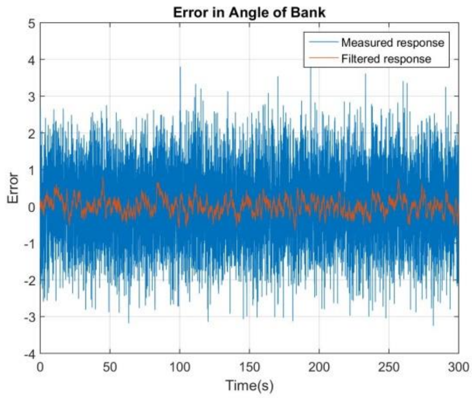
$$L = \begin{bmatrix} 0.3699 & -0.1608 \\ -0.7191 & 1.0609 \\ -0.1314 & -0.3555 \\ -0.1608 & 1.4477 \end{bmatrix} \qquad P = \begin{bmatrix} 0.3699 & -0.7191 & -0.1314 & -0.1608 \\ -0.7191 & 2.1261 & -0.2989 & 1.0609 \\ -0.1314 & -0.0290 & 0.6249 & -0.3555 \\ -0.1608 & 1.0609 & -0.3558 & 1.4477 \end{bmatrix}$$

2) Kalman Filter







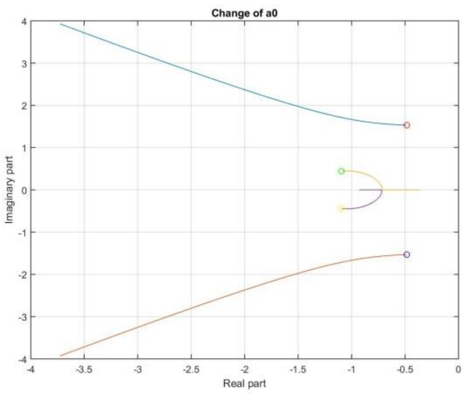


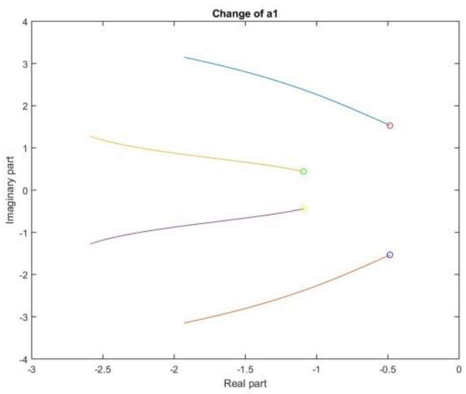
3) Conclusion

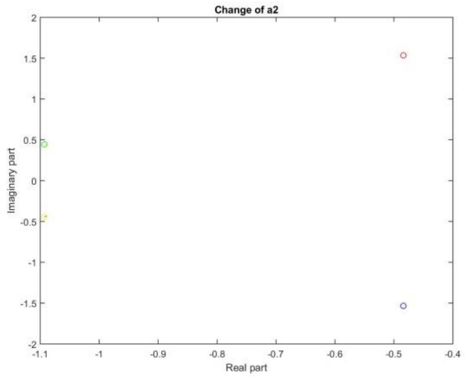
Variance of Error		
	Error in measured response	Error in filtered response
Angle of the Sideslip	1.0292×10^{0}	9.7910×10^{-1}
Angle of Bank	1.1448×10^{-2}	4.7648×10^{-2}

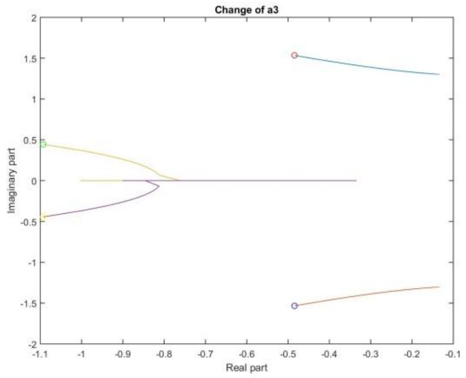
Problem 3. Poles

1) Change of poles









2) Conclusion

With below equations shown, it can be stated that poles of a system will affect its time response.

$$\text{Poles} = \sigma \pm \mathrm{j}\omega_d = -\xi\omega_n \pm j\omega_n\sqrt{1-\xi^2}$$

In which, ξ is Damping Ratio and ω_n is the Natural Frequency. So according to the knowledge of classical control theory, those response parameters such as settling time and maximum overshoot will be determined by the value of the poles.

$$T_s = \frac{4}{\sigma} \qquad\qquad M_p = e^{-\xi\pi/\sqrt{1-\xi^2}}$$

As the σ increasing and ω_n being constant, the T_s will be smaller and the M_p will decreases. However, it has to be mentioned that the instability will be reached with a too large value. By the way, it also can be quantitatively analyzed by selecting the response parameters and calculated the value of poles reversely.

$$\mathrm{J} = \int_0^\infty \left(x^T C^T \begin{bmatrix} a_0 & 0 \\ 0 & a_1 \end{bmatrix} Cx + u^T \begin{bmatrix} a_2 & 0 \\ 0 & a_3 \end{bmatrix} u \right) dx$$

$$\mathrm{J} = \int_0^\infty \left(a_0\beta^2 + a_1\emptyset^2 + a_2{\delta_a}^2 + a_3{\delta_r}^2 \right) dx$$

According to the above calculation, $\mathrm{a}_0 \dots \mathrm{a}_3$ have a correspondence relationship with the angle states and inputs, which means the time response of each parameters will change as giving different values of a.

As the stated above, I would like to choose a larger a_1 in order to get a faster convergence response of the angle of bank. Here are the selected parameters.

$$\mathrm{Q} = C^T \begin{bmatrix} 10 & 0 \\ 0 & 20 \end{bmatrix} \mathrm{C} \qquad\qquad \mathrm{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Problem 4. Optimal regulator

1) Initial state and final output

$$x_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad y_t = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

2) Weight matrices

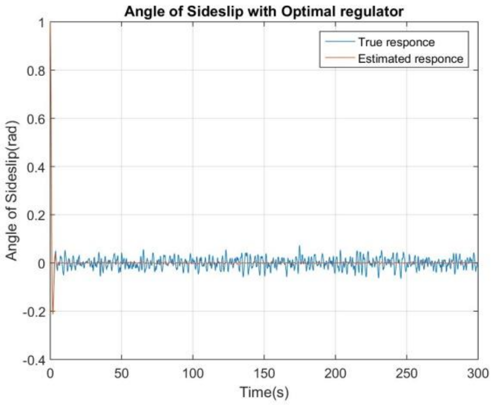
$$\mathrm{J} = \int_0^\infty (x^T Q x + u^T R u)\, dx$$

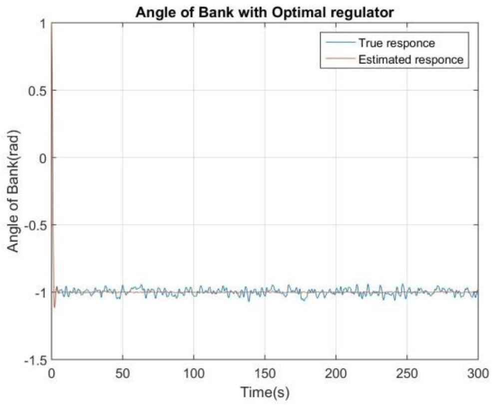
$$\mathrm{Q} = C^T \begin{bmatrix} 10 & 0 \\ 0 & 20 \end{bmatrix} \mathrm{C} \qquad \mathrm{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

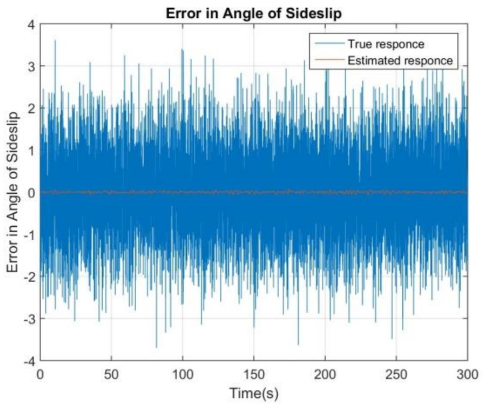
3) Optimal regulator

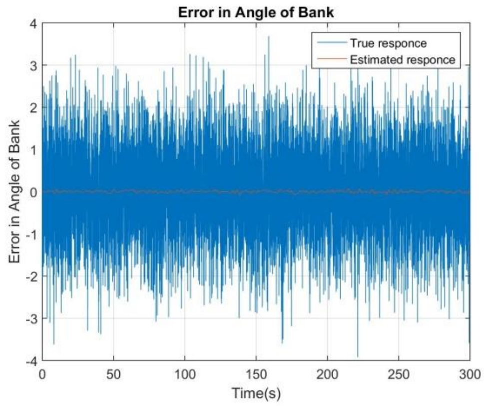
$$K = \begin{bmatrix} -2.9786 & 1.7429 & 1.1678 & 3.0930 \\ 6.5996 & -2.3920 & -3.9803 & -3.0648 \end{bmatrix}$$

$$A_{cl} = A - BK$$









4) Conclusion

Variance of Error		
	Error in measured response	Error in estimated response
Angle of the Sideslip	9.9119×10^{-1}	4.3716×10^{-4}
Angle of Bank	9.9241×10^{-1}	5.3901×10^{-4}

Problem 5. Optimal Control

1) State Space System

$$\ddot{x} = u$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ u \end{bmatrix}$$

2) Initial & final condition

$$x(0) = 0 \qquad \dot{x}\,(0) = 0$$
$$x(t_f) = 1 \qquad \dot{x}\,(t_f) = 0$$

3) Input constraint

$$|u| \le 1$$

4) Solution

Step 1: Define Hamiltonian

$$J = \emptyset\left(x(t_f)\right) + \int_{t_0}^{t_f} L(x(t), u(t), t)dt$$

$$J = t_f = \int_0^{t_f} 1dt$$

$$\emptyset\left(x(t_f)\right) = 0 \qquad L(x(t), u(t), t) = 1 \qquad t_0 = 0$$

$$\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} \qquad f = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ u \end{bmatrix}$$

$$H(x, u, \lambda, t) = L(x(t), u(t), t) + \lambda^T f = 1 + \lambda_1 x_2 + \lambda_2 u$$

Step 2: Minimize the Hamiltonian

$$u^*(t) = \underset{u(t)}{\operatorname{argmin}}[H(x, u, \lambda, t)] = \begin{cases} +1 & if\ \lambda_2 < 0 \\ -1 & if\ \lambda_2 > 0 \end{cases}$$

Step 3: Define the minimal of Hamiltonian

$$H^* = \min_{u(t)}[H(x, u^*, \lambda, t)] = 1 + \lambda_1 x_2 + \lambda_2 u^*$$

Step 4: Solve the state and co-state equations under the boundary condition

$$\dot{x} = \frac{\partial H}{\partial \lambda} = \begin{bmatrix} x_2 \\ u \end{bmatrix} \qquad \dot{\lambda} = -\frac{\partial H}{\partial x} = \begin{bmatrix} 0 \\ -\lambda_1 \end{bmatrix}$$

$$\lambda(t_f) = (\frac{\partial \phi}{\partial x} + \left(\frac{\partial \psi}{\partial x}\right)^T v)_{t=t_f}$$

$$H(t_f) = (\frac{\partial \phi}{\partial t} + \left(\frac{\partial \psi}{\partial t}\right)^T v)_{t=t_f} = 1 + \lambda_1(t_f) x_2(t_f) + \lambda_2(t_f) u^*(t_f)$$

Because of the final condition as followed, we can furtherly simplify the Hamiltonian in final time.

$$\begin{bmatrix} x_1(t_f) \\ x_2(t_f) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$H(t_f) = 1 + \lambda_2(t_f) u^*(t_f) = 0$$

So we can get the optimal $\lambda_2(t_f)$ as followed.

$$\lambda_2(t_f) = -\frac{1}{u^*(t_f)} = \begin{cases} -1 & if\ u^*(t_f) = +1 \\ +1 & if\ u^*(t_f) = -1 \end{cases}$$

Step 5: Substitute the results at step 4 and obtain the optimal control

Given the input constraint, it is clear that the optimal strategy is selecting the maximum of the absolute value of input.

$$|u| \leqq 1$$

$$u^* = \begin{cases} +1 & when\ t \in [0, \frac{1}{2} t_f] \\ -1 & when\ t \in (\frac{1}{2} t_f, t_f] \end{cases}$$

As for the minimum final time t_f, because the constraints of initial state and final state and the relationship between input and state, we can implied that $t_f = 2s$.

$$\mathrm{x}(0) = 0 \qquad \mathrm{x}(t_f) = 1 \qquad \ddot{x} = u^*$$

The code attached

```
% Problem 1
clear; % clear memory

n=4; % dim(state)
m=2; % dim(input)
N = 10000; % sampling number

x = zeros(N,n); % state vector
u = zeros(N,m); % input vector
y = zeros(N,m); % output vector

d = 0.1 * randn(1,N); % disturbance to system
v = 0.1 * randn(m,N); % disturbance to output

% Time parameter
initial_time = 0;
final_time = 300;
t = zeros(N,1); % time
dt = (final_time - initial_time)/N; % delta time

for i=0:N-1
	t(i+1,1)=i*dt;
end

%Global parameters
global A
global B
global G
global C
global U

% System parameters
Y_v = -0.12;
Y_deta_r = 3.13;
L_beta = -4.12;
L_p = -0.974;
L_r = 0.292;
L_deta_a = 0.31;
L_deta_r = 0.183;
N_beta = 1.62;
N_p = -0.0157;
N_r = -0.232;
N_deta_a = 0.0127;
```

```
N_deta_r = -0.922;

g = 9.8;
v_sound = 340.29;
U = 0.8 * v_sound; % speed of the plane

% System matrices
A = [Y_v        0         -1         g/U    ;
     L_beta    L_p        L_r         0     ;
     N_beta    N_p        N_r         0     ;
       0        1          0          0     ];

B = [0                 Y_deta_r/U    ;
     L_deta_a    L_deta_r            ;
     N_deta_a    N_deta_r            ;
       0              0              ];

G = [ Y_deta_r/U    ;
     L_deta_r       ;
     N_deta_r       ;
        0           ];

C = [1 0 0 0;
     0 0 0 1];

D = 0;

% State System
system = ss(A,[B G],C,D);

% Controllability and Observability
Uc = ctrb(A,B);
Uo = obsv(A,C);
Kc = rank(Uc);
Ko = rank(Uo);
Xco = [Kc Ko];
save -ascii controllability.dat Uc
save -ascii observabillity.dat Uo
save -ascii rank.dat Xco

% Eigen vector and Eigen value
[vec, val] = eig(A);
save -ascii vector.dat vec
save -ascii value.dat val
```

```
% Simulation
x(1, :) = [1 0 0 1]; % initial state
y = lsim(system,[u d'],t,x(1,:));

for i=1:N-1

%runge-kutta 4th order
p1 = xdt(x(i,:)',u(i,:)',d(:,i));
p2 = xdt(x(i,:)'+dt/2*p1,u(i,:)',d(:,i));
p3 = xdt(x(i,:)'+dt/2*p2,u(i,:)',d(:,i));
p4 = xdt(x(i,:)'+dt+p3,u(i,:)',d(:,i));

x(i+1,:) = x(i,:)' + dt/6 * (p1 + p2*2 + p3*2 + p4);

end

% Figure
figure(1)
plot(t, x(:, 1))
xlabel('Time(s)');
ylabel('Angle of Sideslip(rad)');
title('Angle of Sideslip');
grid on;
saveas(figure(1), 'Angle of Sideslip.jpg');

figure(2)
plot(t, x(:, 2))
xlabel('Time(s)');
ylabel('Speed of Roll(rad/s)');
title('Speed of Roll');
grid on;
saveas(figure(2), 'Speed of Roll.jpg');

figure(3)
plot(t, x(:, 3))
xlabel('Time(s)');
ylabel('Speed of Yaw(rad/s)');
title('Speed of Yaw');
grid on;
saveas(figure(3), 'Speed of Yaw.jpg');

figure(4)
plot(t, x(:, 4))
```

```
xlabel('Time(s)');
ylabel('Angle of Bank(rad)');
title('Angle of Bank');
grid on;
saveas(figure(4), 'Angle of Bank.jpg');

X = [t x];
save -ascii Problem1.dat X
```

```
function x = xdt(x,u,d)

%Global parameters
global A
global B
global G

x = A*x + B*u + G*d;
```

```
% Problem 2
clear; % clear memory

n=4; % dim(state)
m=2; % dim(input)
N = 10000; % sampling number

x = zeros(N,n); % state vector
u = zeros(N,m); % input vector
y = zeros(N,m); % output vector
x_estimation = zeros(N,n); %estimated state
y_filtered = zeros(N,m); %estimated output
y_measured = zeros(N,m); %observed output

Q = 1; % Var. of process niose
R = eye(2); % Var. of measurement niose

d = sqrt(Q)*randn(1,N); % disturbance to system
v = sqrt(R)*randn(m,N); % disturbance to output

% Time parameter
initial_time = 0;
final_time = 300;
t = zeros(N,1); % time
dt = (final_time - initial_time)/N; % delta time
```

```
for i=0:N-1
        t(i+1,1)=i*dt;
end

% System parameters
Y_v = -0.12;
Y_deta_r = 3.13;
L_beta = -4.12;
L_p = -0.974;
L_r = 0.292;
L_deta_a = 0.31;
L_deta_r = 0.183;
N_beta = 1.62;
N_p = -0.0157;
N_r = -0.232;
N_deta_a = 0.0127;
N_deta_r = -0.922;

g = 9.8;
v_sound = 340.29;
U = 0.8 * v_sound; % speed of the plane

% System matrice
A = [Y_v        0         -1         g/U    ;
        L_beta    L_p       L_r        0      ;
        N_beta    N_p       N_r        0      ;
          0          1          0        0      ];

B = [0                Y_deta_r/U    ;
        L_deta_a    L_deta_r       ;
        N_deta_a    N_deta_r       ;
          0               0             ];

G = [ Y_deta_r/U    ;
        L_deta_r        ;
        N_deta_r       ;
           0             ];

C = [1 0 0 0;
        0 0 0 1];

D = 0;

% State System
```

```
system = ss(A,[B G],C,D);

%initial value
x(1,:) = [1 0 0 1];

% Kalman Gain and The solution of ARE
[L, P] = lqe(A, G, C, Q, R);
save -ascii L.dat L
save -ascii P.dat P

%Kalman Filter
[kalman,L,P,M] = kalman(system,Q,R);
kalman =(kalman(1:2,:));
y = lsim(system,[u d'],t,x(1,:)); % true response
y_measured = y+v'; % measured response
y_filtered = lsim(kalman,[u y_measured],t,x(1,:)); % filtered response

% Error
MeasErr = y-y_measured;
MeasErrCov = sum(MeasErr.*MeasErr)/length(MeasErr);
EstErr = y-y_filtered ;
EstErrCov = sum(EstErr.*EstErr)/length(EstErr);
Err = [MeasErrCov EstErrCov];
save -ascii Error.dat Err

% Figure
figure(1)
plot(t, y(:, 1), t, y_filtered(:, 1))
legend('True response','Filtered response')
xlabel('Time(s)');
ylabel('Angle of Sideslip(rad)');
title('Angle of Sideslip');
grid on;
saveas(figure(1), 'Angle of Sideslip with Kalman filter.jpg');

figure(2)
plot(t, y(:, 2), t, y_filtered(:, 2))
legend('True response','Filtered response')
xlabel('Time(s)');
ylabel('Angle of Bank(rad)');
title('Angle of Bank');
grid on;
saveas(figure(2), 'Angle of Bank with Kalman filter.jpg');
```

```
figure(3)
plot(t, y(:, 1)-y_measured(:, 1), t, y(:, 1)-y_filtered(:, 1))
legend('Measured response','Filtered response')
xlabel('Time(s)');
ylabel('Error');
title('Error in Angle of Sideslip');
grid on;
saveas(figure(3), 'Error in Angle of Sideslip.jpg');

figure(4)
plot(t, y(:, 2)-y_measured(:, 2), t, y(:, 2)-y_filtered(:, 2))
legend('Measured response','Filtered response')
xlabel('Time(s)');
ylabel('Error');
title('Error in Angle of Bank');
grid on;
saveas(figure(4), 'Error in Angle of Bank.jpg');

X = [t y y_measured y_filtered];
save -ascii Problem2.dat X
```

```
% Problem 3
clear; % clear memory

% Time parameter
N = 1000; % sampling number
initial_time = 0;
final_time = 300;
t = zeros(N,1); % time
dt = (final_time - initial_time)/N; % delta time

for i=0:N-1
    t(i+1,1)=i*dt;
end

% System parameters
Y_v = -0.12;
Y_deta_r = 3.13;
L_beta = -4.12;
L_p = -0.974;
L_r = 0.292;
L_deta_a = 0.31;
L_deta_r = 0.183;
N_beta = 1.62;
```

```
N_p = -0.0157;
N_r = -0.232;
N_deta_a = 0.0127;
N_deta_r = -0.922;

g = 9.8;
v_sound = 340.29;
U = 0.8 * v_sound; % speed of the plane



% System matrice
A = [Y_v        0         -1         g/U    ;
        L_beta    L_p       L_r        0      ;
        N_beta    N_p       N_r        0      ;
         0           1          0        0      ];

B = [0                 Y_deta_r/U    ;
        L_deta_a    L_deta_r       ;
        N_deta_a    N_deta_r       ;
         0               0              ];

G = [ Y_deta_r/U    ;
        L_deta_r        ;
        N_deta_r        ;
          0             ];

C = [1 0 0 0;
        0 0 0 1];

D = 0;

% Evaluation function
P1 = zeros(4,N);
P2 = zeros(4,N);
P3 = zeros(4,N);
P4 = zeros(4,N);

figure
for i = 1:4
        a = [1 1 1 1];
        for j = 1:N
                a(1,i) = j;
                Q = C'*diag([a(:,1),a(:,2)])*C;
                R = diag([a(:,3),a(:,4)]);
```

```
        % optimal regulator
        [K,S,E] = lqr(A,B,Q,R);
        % system
        system = ss(A-B*K, B, C, D);
        p = pole(system);
        P1(i,j) = p(1);
        P2(i,j) = p(2);
        P3(i,j) = p(3);
        P4(i,j) = p(4);
    end
end


% Figure
figure(1)
plot(real(P1(1,:)),imag(P1(1,:)), real(P2(1,:)),imag(P2(1,:)), real(P3(1,:)),imag(P3(1,:)),
real(P4(1,:)),imag(P4(1,:)))
hold on
plot(real(P1(1,1)),imag(P1(1,1)),'ro',real(P2(1,1)),imag(P2(1,1)),'bo',real(P3(1,1)),imag(P3(1,1)),'go
',real(P4(1,1)),imag(P4(1,1)),'yo')
xlabel('Real part')
ylabel('Imaginary part')
title('Change of a0')
grid on;
saveas(figure(1), 'Change of a0.jpg');

figure(2)
plot(real(P1(2,:)),imag(P1(2,:)),real(P2(2,:)),imag(P2(2,:)),real(P3(2,:)),imag(P3(2,:)),real(P4(2,:)),i
mag(P4(2,:)))
hold on
plot(real(P1(2,1)),imag(P1(2,1)),'ro',real(P2(2,1)),imag(P2(2,1)),'bo',real(P3(2,1)),imag(P3(2,1)),'go
',real(P4(2,1)),imag(P4(2,1)),'yo')
xlabel('Real part')
ylabel('Imaginary part')
title('Change of a1')
saveas(figure(2), 'Change of a1.jpg');

figure(3)
plot(real(P1(3,:)),imag(P1(3,:)),real(P2(3,:)),imag(P2(3,:)),real(P3(3,:)),imag(P3(3,:)),real(P4(3,:)),i
mag(P4(3,:)))
hold on
plot(real(P1(3,1)),imag(P1(3,1)),'ro',real(P2(3,1)),imag(P2(3,1)),'bo',real(P3(3,1)),imag(P3(3,1)),'go
',real(P4(3,1)),imag(P4(3,1)),'yo')
xlabel('Real part')
```

```
ylabel('Imaginary part')
title('Change of a2')
saveas(figure(3), 'Change of a2.jpg');

figure(4)
plot(real(P1(4,:)),imag(P1(4,:)),real(P2(4,:)),imag(P2(4,:)),real(P3(4,:)),imag(P3(4,:)),real(P4(4,:)),i
mag(P4(4,:)))
hold on
plot(real(P1(4,1)),imag(P1(4,1)),'ro',real(P2(4,1)),imag(P2(4,1)),'bo',real(P3(4,1)),imag(P3(4,1)),'go
',real(P4(4,1)),imag(P4(4,1)),'yo')
xlabel('Real part')
ylabel('Imaginary part')
title('Change of a3')
saveas(figure(4), 'Change of a3.jpg');

Pole = [P1 P2 P3 P4];
save -ascii Problem3.dat Pole
```

```
% Problem 4
clear; % clear memory

n=4; % dim(state)
m=2; % dim(input)
N = 10000; % sampling number

x = zeros(N,n); % state vector
u = zeros(N,m); % input vector
y = zeros(N,m); % output vector
x_estimation = zeros(N,n); %estimated state
y_estimated = zeros(N,m); %estimated output
y_measured = zeros(N,m); %observed output

Q = 1; % Var. of process niose
R = eye(2); % Var. of measurement niose

d = sqrt(Q)*randn(1,N); % disturbance to system
v = sqrt(R)*randn(m,N); % disturbance to output

% Time parameter
initial_time = 0;
final_time = 300;
t = zeros(N,1); % time
dt = (final_time - initial_time)/N; % delta time
```

```
for i=0:N-1
        t(i+1,1)=i*dt;
end

% System parameters
Y_v = -0.12;
Y_deta_r = 3.13;
L_beta = -4.12;
L_p = -0.974;
L_r = 0.292;
L_deta_a = 0.31;
L_deta_r = 0.183;
N_beta = 1.62;
N_p = -0.0157;
N_r = -0.232;
N_deta_a = 0.0127;
N_deta_r = -0.922;

g = 9.8;
v_sound = 340.29;
U = 0.8 * v_sound; % speed of the plane

% System matrice
A = [Y_v        0        -1        g/U;
                 L_beta    L_p       L_r      0     ;
                 N_beta    N_p       N_r      0     ;
         0           1          0         0     ];

B = [0                Y_deta_r/U;
                 L_deta_a    L_deta_r     ;
                 N_deta_a    N_deta_r     ;
         0               0                 ];

G = [ Y_deta_r/U;
                   L_deta_r      ;
                   N_deta_r      ;
           0                 ];

C = [1 0 0 0;
         0 0 0 1];

D = 0;

% Initial value
```

```
x(1,:) = [1 0 0 1];
% Convergence value
y_final = [0;-1];

% Weighting matrix
Q_K = C'*diag([10,20])*C;
R_K = diag([1,1]);

[K,S,E] = lqr(A,B,Q_K,R_K);
save -ascii optimal_gain.dat K
save -ascii closedloop_eigenvalue.dat E

% System
A_cl = A-B*K;
system = ss(A_cl,[B G],C,D);

u = -inv((C/A_cl)*B)*y_final;
u1 = zeros(N,1);
u2 = zeros(N,1);
u1(:,1) = u(1,1);
u2(:,1) = u(2,1);

% Kalman filter
[kest,L,P] = kalman(system,Q,R,0);
kest = kest(1:2,:);

% Observation and Estimation
y = lsim(system,[u1 u2 d'],t,x(1,:));
y_measured = y+v';
y_estimated = lsim(kest,[u1 u2 y_measured],t,x(1,:));

% Error
MeasErr = y-y_measured;
MeasErrCov = sum(MeasErr.*MeasErr)/length(MeasErr);
EstErr = y-y_estimated;
EstErrCov = sum(EstErr.*EstErr)/length(EstErr);
Err = [MeasErrCov EstErrCov];
save -ascii Error.dat Err

% Figure
figure(1)
plot(t, y(:, 1), t, y_estimated(:, 1))
legend('True responce','Estimated responce')
xlabel('Time(s)');
```

```
ylabel('Angle of Sideslip(rad)');
title('Angle of Sideslip with Optimal regulator');
grid on;
saveas(figure(1), 'Angle of Sideslip with Optimal regulator.jpg');

figure(2)
plot(t, y(:, 2), t, y_estimated(:, 2))
legend('True responce','Estimated responce')
xlabel('Time(s)');
ylabel('Angle of Bank(rad)');
title('Angle of Bank with Optimal regulator');
grid on;
saveas(figure(2), 'Angle of Bank with Optimal regulator.jpg');

figure(3)
plot(t, y(:, 1)-y_measured(:, 1), t, y(:, 1)-y_estimated(:, 1))
legend('True responce','Estimated responce')
xlabel('Time(s)');
ylabel('Error in Angle of Sideslip');
title('Error in Angle of Sideslip');
grid on;
saveas(figure(3), 'Error in Angle of Sideslip.jpg');

figure(4)
plot(t, y(:, 2)-y_measured(:, 2), t, y(:, 2)-y_estimated(:, 2))
legend('True responce','Estimated responce')
xlabel('Time(s)');
ylabel('Error in Angle of Bank');
title('Error in Angle of Bank');
grid on;
saveas(figure(4), 'Error in Angle of Bank.jpg');

X = [t y y_measured y_estimated];
save -ascii Problem4.dat X
```

Conference:
[1]http://www.wolframalpha.com
[2]https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-30-feedback-control-systems-fall-2010/
[3]https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-323-principles-of-optimal-control-spring-2008/