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**

1. State Space System

|  |  |  |  |
| --- | --- | --- | --- |
|  | angle of the sideslip |  | angle of bank |
|  | speed of the roll |  | speed of the yaw |
|  | angle of the wing |  | angle of rudder |
|  | disturbance to system |  | disturbance to output |

1. System Parameters

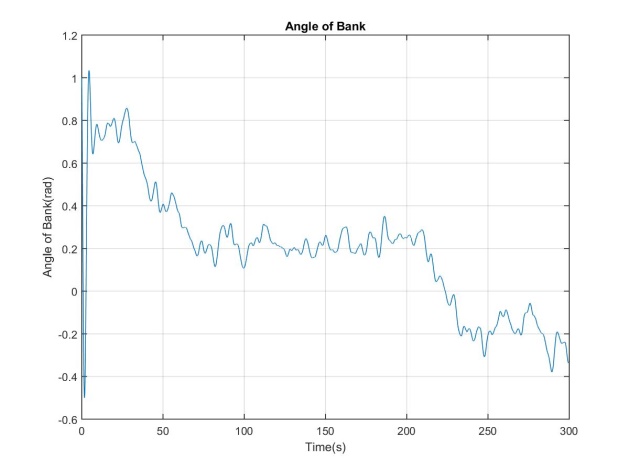
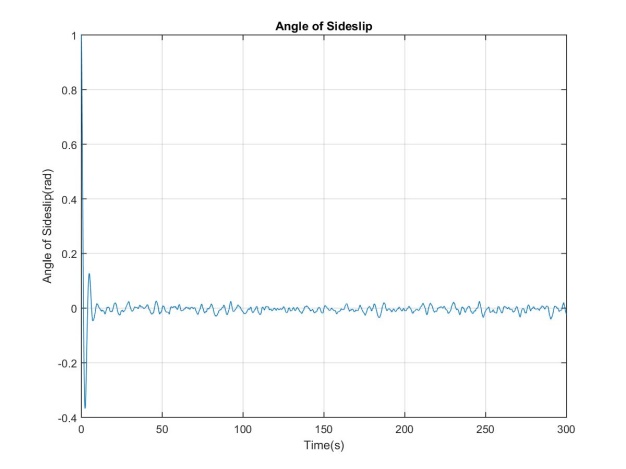
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | |
|  | -0.12 |  | 3.13 |  | -4.12 |  | -0.974 |  | 2.92 |  | 0.310 |
|  | 0.183 |  | 1.62 |  | -0.0157 |  | -0.232 |  | 0.0127 |  | -0.922 |

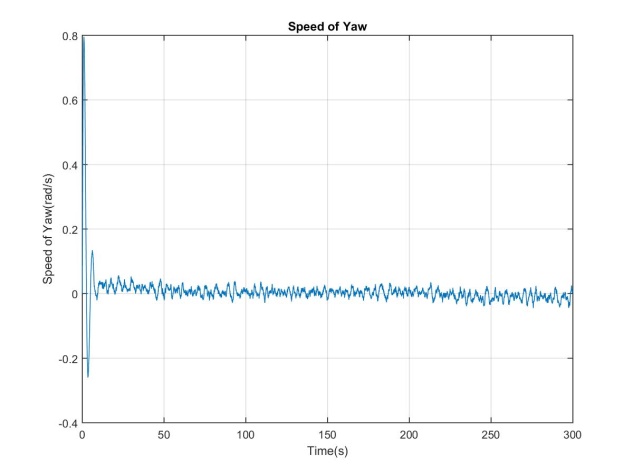
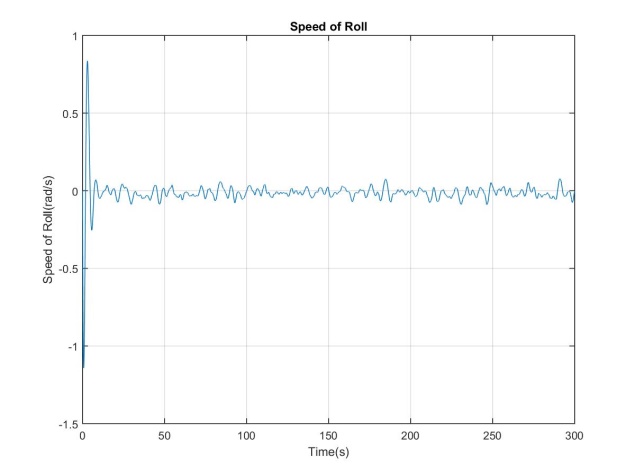
1. System Matrices

**Problem 1. Kinematical Simulation**

1. Controllability and Observability

1. Stability
2. Kinematical Simulation





1. 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.

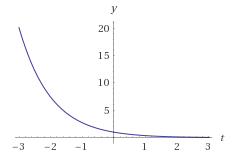
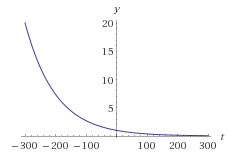
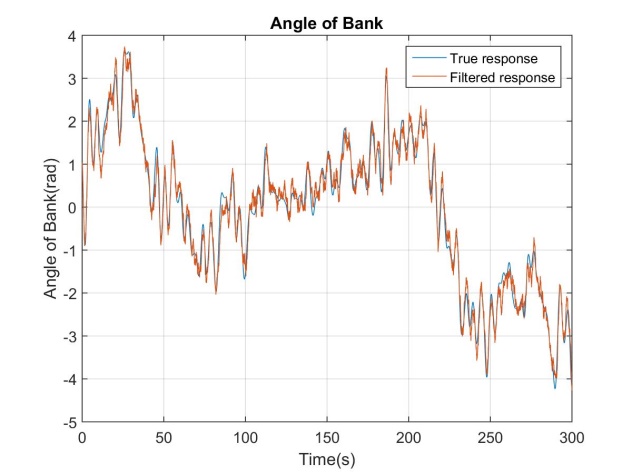
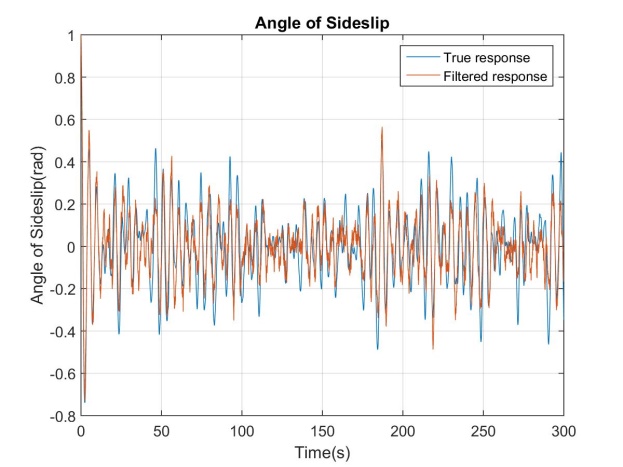
 

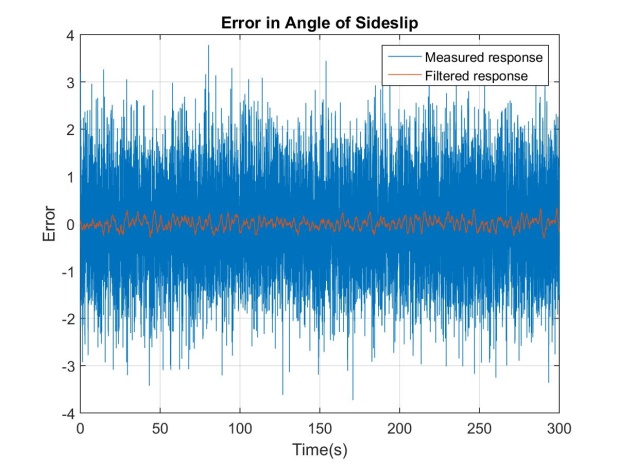
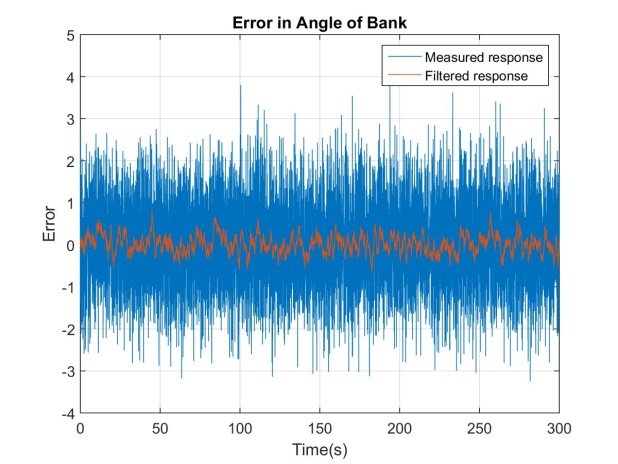
Fig.1 Fig.2 [1]

**Problem 2. Kalman Filter**

1. Kalman Gain and The solution of ARE

1. Kalman Filter



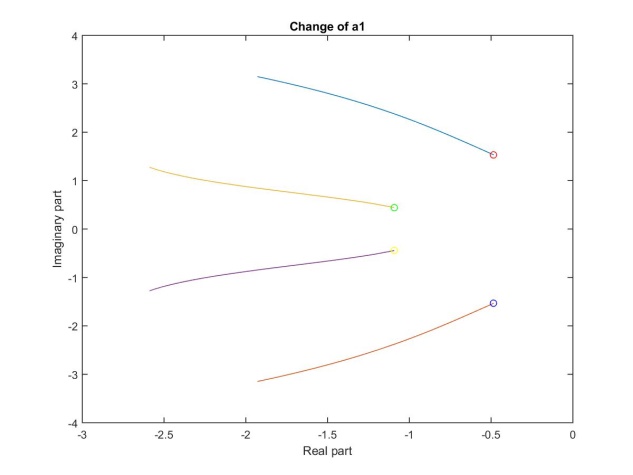
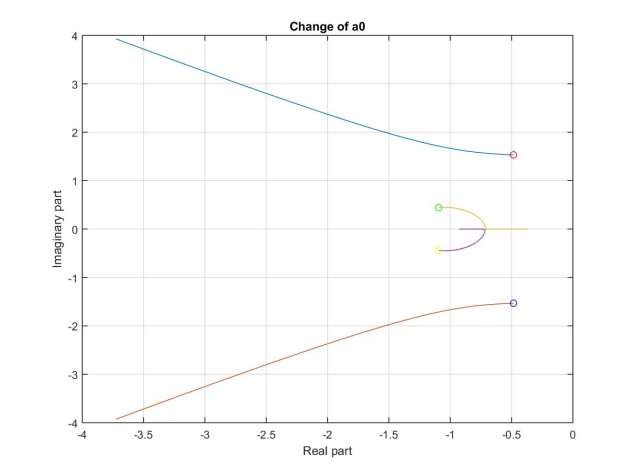
 

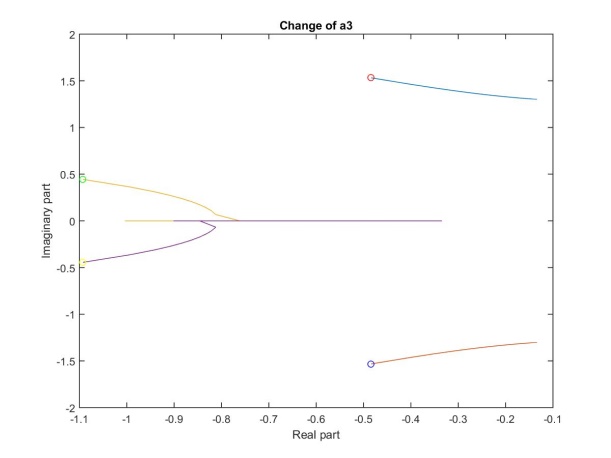
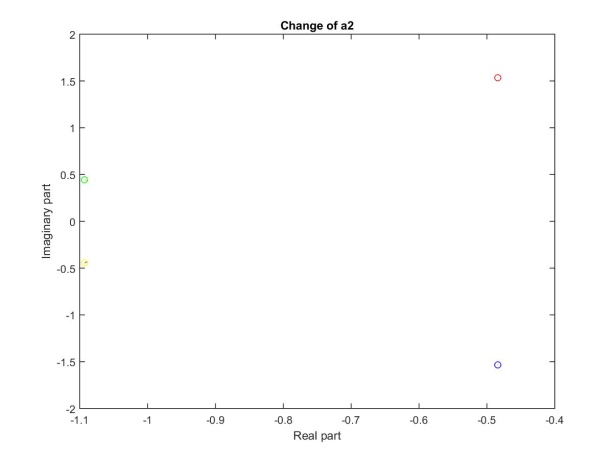
1. Conclusion

|  |  |  |
| --- | --- | --- |
| Variance of Error | | |
|  | Error in measured response | Error in filtered response |
| Angle of the Sideslip |  |  |
| Angle of Bank |  |  |

**Problem 3. Poles**

1. Change of poles





1. Conclusion

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

In which, is Damping Ratio and 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.

As the increasing and being constant, the will be smaller and the 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.

According to the above calculation, 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.

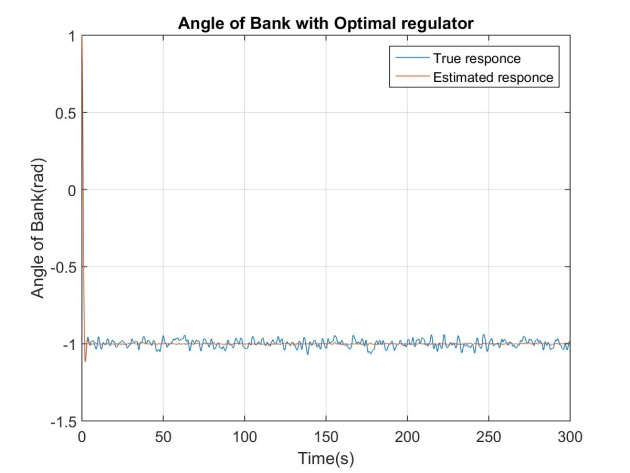
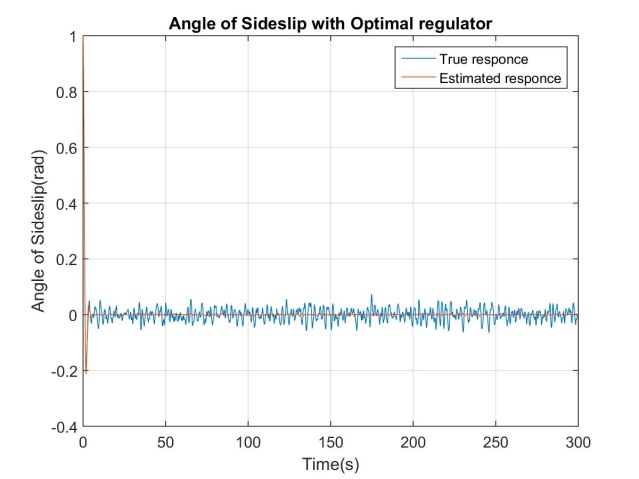
As the stated above, I would like to choose a larger in order to get a faster convergence response of the angle of bank. Here are the selected parameters.

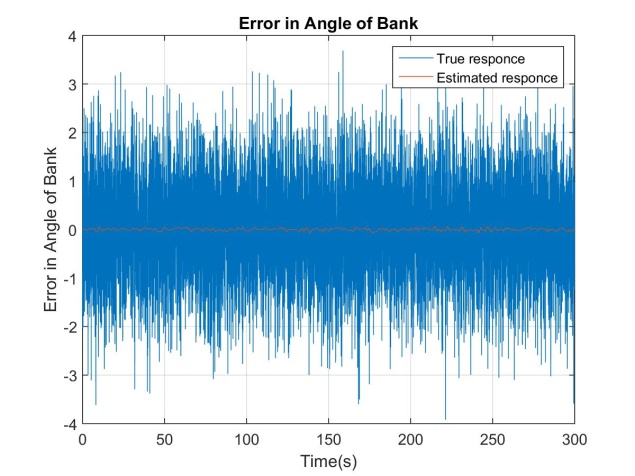
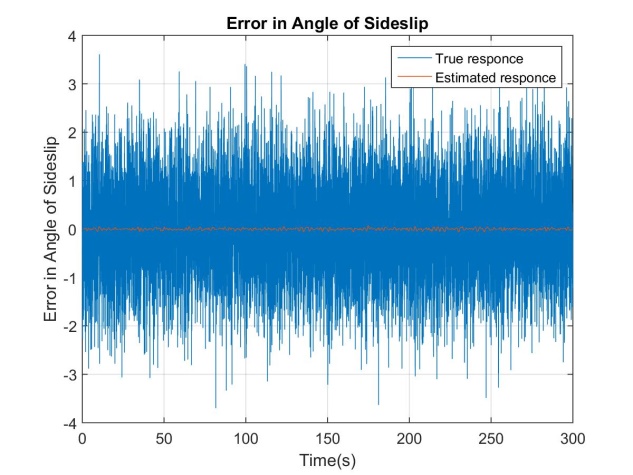
**Problem 4. Optimal regulator**

1. Initial state and final output

1. Weight matrices

1. Optimal regulator





1. Conclusion

|  |  |  |
| --- | --- | --- |
| Variance of Error | | |
|  | Error in measured response | Error in estimated response |
| Angle of the Sideslip |  |  |
| Angle of Bank |  |  |

**Problem 5. Optimal Control**

1. State Space System
2. Initial & final condition
3. Input constraint
4. Solution

Step 1: Define Hamiltonian

Step 2: Minimize the Hamiltonian

Step 3: Define the minimal of Hamiltonian

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

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

= 0

So we can get the optimal as followed.

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.

As for the minimum final time, because the constraints of initial state and final state and the relationship between input and state, we can implied that

|  |
| --- |
| **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,:)),imag(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,:)),imag(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,:)),imag(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/