THE CHINESE UNIVERSITY OF HONG KONG
DEPT OF MECHANICAL & AUTOMATION ENG

ENGG5403 Linear System Theory & Design

Design Project #1

by

Liuchao JIN (Student ID: 1155184008)

Liuchao Jin

2022-23 Term 2

Introduction

This report presents the design of a flight control system for an unmanned helicopter, HeLion, constructed by a UAV Research Team. The objective of this project is to design a control law that stabilizes the system and drives the helicopter's dynamics to the hovering state quickly from a given initial condition. The linearized state space model for the inner-loop of the helicopter system is developed through test flights of the actual helicopter in hovering and near-hovering flight conditions. In this report, we will discuss the design specifications, the state response and control input, and the impact of measurement noise on the control system.

System Description

The helicopter system can be modeled using a linearized state space model, given by:

$$\dot{x} = Ax + Bu + Ew \tag{1}$$

where the state vector x includes the following variables:

$$x = \begin{bmatrix} \phi \\ \theta \\ p \\ q \\ a_s \\ b_s \\ r \\ \delta_{ped,int} \\ \psi \end{bmatrix} \tag{2}$$

The input vector u includes the normalized control inputs for the lateral, longitudinal, and pedal channels, respectively:

$$u = \begin{bmatrix} \delta_{lat} \\ \delta_{lon} \\ \delta_{ped} \end{bmatrix} \tag{3}$$

The disturbance vector w includes the gusts of wind in the x-, y-, and z-directions:

$$w = \begin{bmatrix} u_{wind} \\ v_{wind} \\ w_{wind} \end{bmatrix} \tag{4}$$

The matrices A, B, and E are defined as:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0.0009 & 0 & 0 \\ 0 & 0 & 0 & 0.9992 & 0 & 0 & -0.0389 & 0 & 0 \\ 0 & 0 & -0.0302 & -0.0056 & -0.0003 & 585.1165 & 11.4448 & -59.529 & 0 \\ 0 & 0 & 0 & -0.0707 & 267.7499 & -0.0003 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -3.3607 & 2.2223 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 2.4483 & -3.3607 & 0 & 0 & 0 \\ 0 & 0 & 0.0579 & 0.0108 & 0.0049 & 0.0037 & -21.9557 & 114.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0.0389 & 0 & 0 & 0.9992 & 0 & 0 \end{bmatrix} \tag{5}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 43.3635 \\ 0 & 0 & 0 \\ 0.2026 & 2.5878 & 0 \\ 2.5878 & -0.0663 & 0 \\ 0 & 0 & -83.1883 \\ 0 & 0 & -3.8500 \\ 0 & 0 & 0 \end{bmatrix}, \quad E = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -0.0001 & 0.1756 & -0.0395 \\ 0 & 0.0003 & 0.0338 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -0.0002 & -0.3396 & 0.6424 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{6}$$

and where $\delta_{ped,int}$ is associated with a yaw channel built-in controller, and the control input variables δ_{lat}, δ_{lon}, δ_{ped} are respectively corresponding the lateral channel (for roll, left- and right-side tilting motions), the longitudinal channel (for pitch, forward and backward motions), and the pedal channel (for yaw motion) of the helicopter, w represents wind gust disturbance in x-, y- and z-directions. The control inputs are normalized with 1 being equivalent to $\pi/4$ rad. Due to physical limitation, they will have to be kept within the following limits:

$$|\delta_{lat}| < 0.35, \quad |\delta_{lon}| < 0.35, \quad |\delta_{ped}| < 0.4 \tag{7}$$

The ideal measurement output of the system is given by the vector y, which includes ϕ, θ, p, q, r, and ψ, and is obtained by applying the matrix C to the state vector x.

$$y = \begin{bmatrix} \phi \\ \theta \\ p \\ q \\ r \\ \psi \end{bmatrix} = Cx = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} x \tag{8}$$

Design Specifications

The primary focus of the flight control system for the hovering condition is to design a control law that stabilizes the overall system and drives the dynamics of the helicopter to the hovering state (all state variables are to be driven to 0) as quickly as possible from a given initial condition. The following initial condition is to be used for all simulations:

Physically, it means the helicopter is commanded to hover from a pitch angle of -0.1 rad rad (a nose-down angle of 5.7 degrees) and yaw angle of 0.1 rad (5.7 degrees).

State Response and Control Input

We analyze the state response and control input for the system with and without measurement noise using Simulink shown in Figure 1. Two figures are drawn for each scenario. In the first figure, the state response of the system is plotted against time. In the second figure, the control input is plotted against time.

Figure 2a shows the state response of the system with measurement noise. It can be seen that the system stabilizes to the hovering state within a few seconds. Figure 2b shows the control input required to stabilize the system.

Figure 3a shows the state response of the system without measurement noise. It can be seen that the system stabilizes to the hovering state more smoothly than in the previous scenario. Figure 3b shows the control input required to stabilize the system.

The MATLAB codes for getting these results are shown below:

```
clc; clf; clear all; close all;
%% Set the initial parameters
A = [0 0 1 0 0 0 0.0009 0 0;
    0 0 0 0.9992 0 0 -0.0389 0 0;
    0 0 -0.0302 -0.0056 -0.0003 585.1165 11.4448 -59.529 0;
    0 0 0 -0.0707 267.7499 -0.0003 0 0 0;
    0 0 0 -1 -3.3607 2.2223 0 0 0;
    0 0 -1 0 2.4483 -3.3607 0 0 0;
    0 0 0.0579 0.0108 0.0049 0.0037 -21.9557 114.2 0;
    0 0 0 0 0 0 -1 0 0;
    0 0 0 0.0389 0 0 0.9992 0 0;];
B = [0 0 0;
    0 0 0;
    0 0 43.3635;
    0 0 0;
    0.2026 2.5878 0;
    2.5878 -0.0663 0;
    0 0 -83.1883;
    0 0 -3.8500;
    0 0 0;];
```

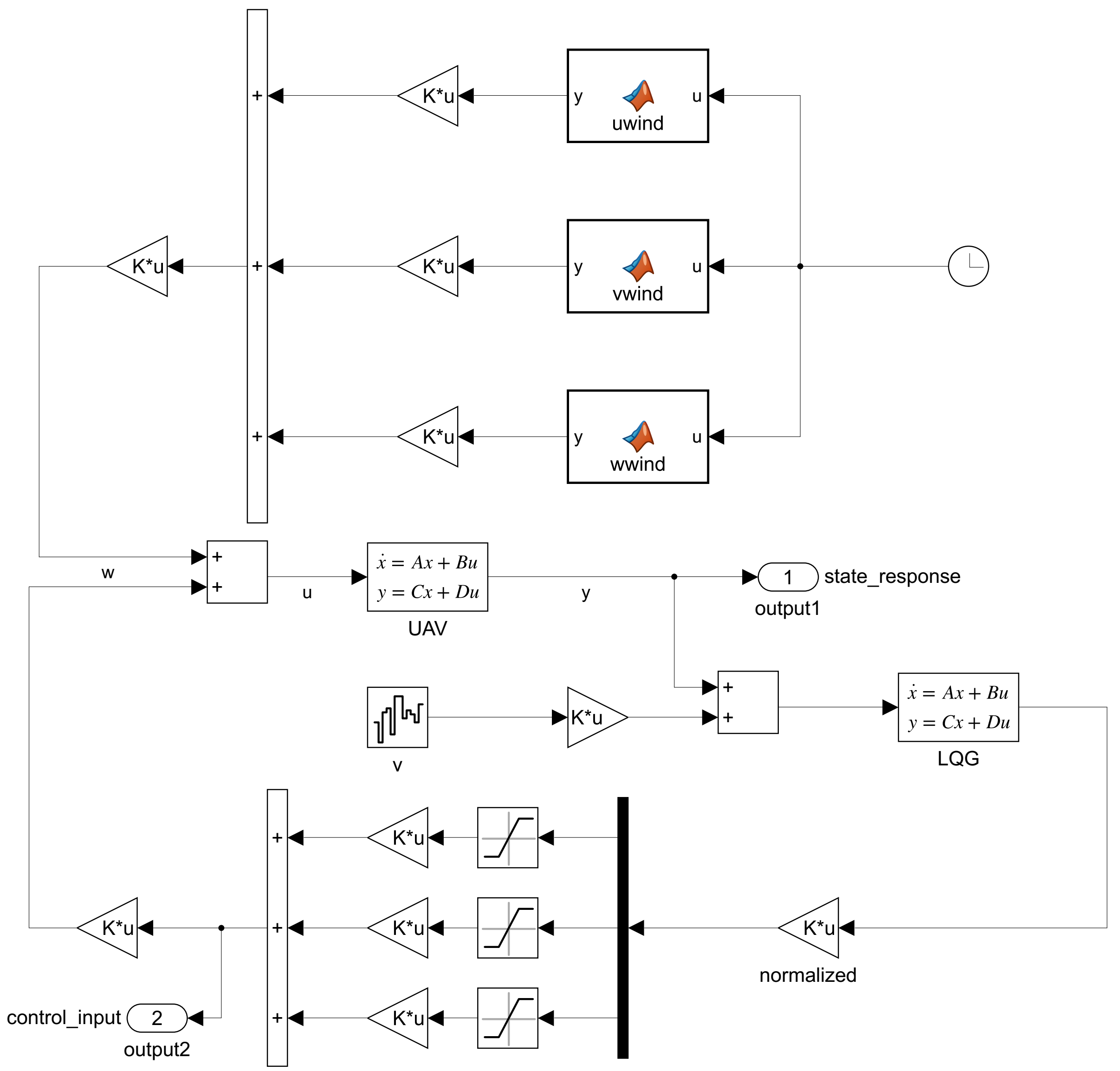
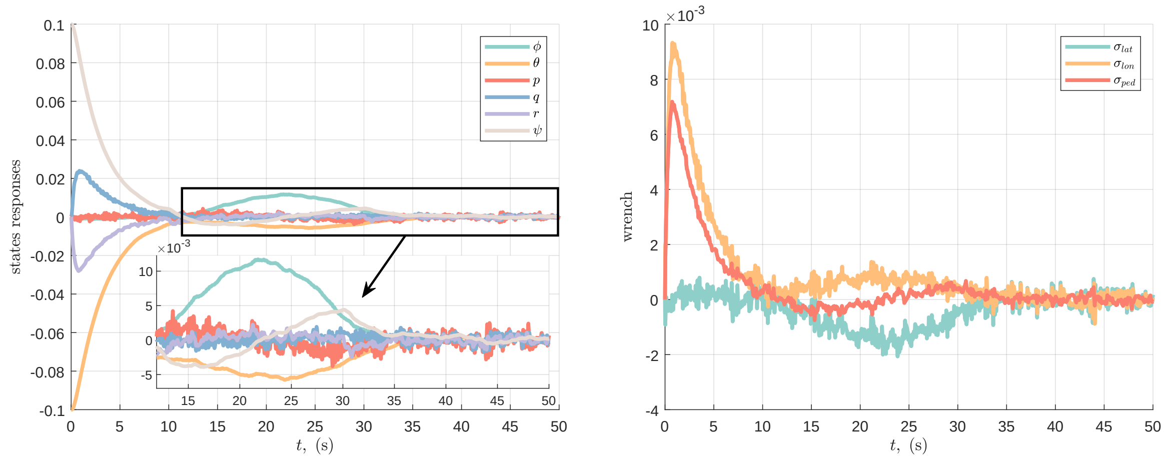


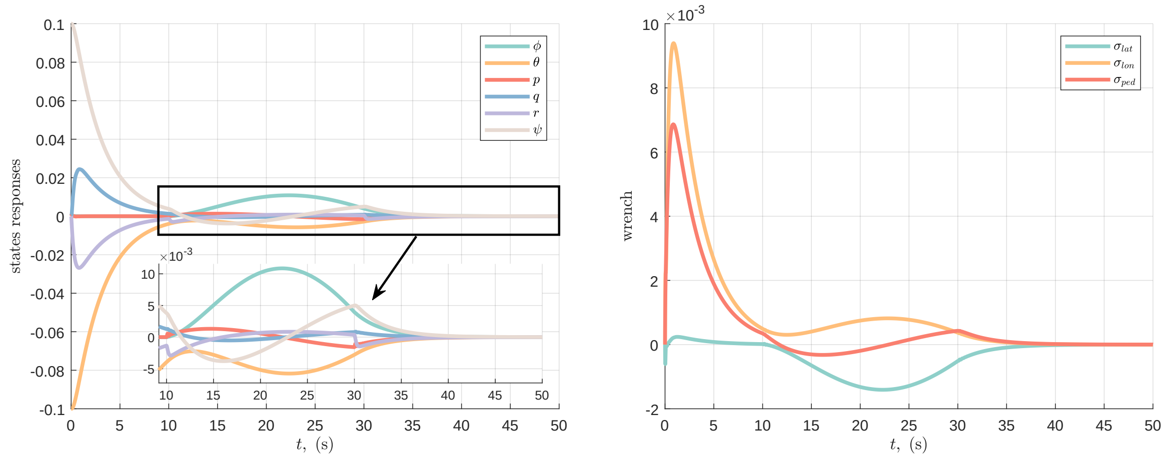
Figure 1: Block diagram for the simulation.

```
C = [1 0 0 0 0 0 0 0 0;
     0 1 0 0 0 0 0 0 0;
     0 0 1 0 0 0 0 0 0;
     0 0 0 1 0 0 0 0 0;
     0 0 0 0 0 0 1 0 0;
     0 0 0 0 0 0 0 0 1;];
D = zeros(size(C,1));
D1 = zeros(6,3);
E = [0 0 0;
     0 0 0;
     -0.0001 0.1756 -0.0395;
     0 0.0003 0.0338;
     0 0 0;
     0 0 0;
     -0.0002 -0.3396 0.6424;
```



(a) **(b)**

Figure 2: System performance with measurement noise.



(a) **(b)**

Figure 3: System performance without measurement noise.

```
    0  0  0;
    0  0  0;];
x0 = [0;
    -0.1;
    0;
    0;
    0;
    0;
    0;
    0;
    0.1;];
%% Calculate LQR
% cost index for LQR
Q = [30 0 0 0 0 0 0 0 0;
     0 10 0 0 0 0 0 0 0;
```

```
     0 0 1 0 0 0 0 0 0;
     0 0 0 1 0 0 0 0 0;
     0 0 0 0 1 0 0 0 0;
     0 0 0 0 0 1 0 0 0;
     0 0 0 0 0 0 1 0 0;
     0 0 0 0 0 0 0 10 0;
     0 0 0 0 0 0 0 0 10;];
Q = 3*Q;
R = [5.2638    1.5521    0.2130;
    1.5521    4.7083    0.8346;
    0.2130    0.8346    4.7579;];
[K,S,CLP] = lqr(A,B,Q,R);
%% Calculate Kalman filter
Qv = [0.001     0      0     0       0       0      0      0     0;
       0   0.001      0     0       0       0      0      0     0;
       0      0   0.001     0       0       0      0      0     0;
       0      0      0  0.001       0       0      0      0     0;
       0      0      0     0    0.001       0      0      0     0;
       0      0      0     0       0    0.001      0      0     0;
       0      0      0     0       0       0  0.001      0     0;
       0      0      0     0       0       0      0  0.001     0;
       0      0      0     0       0       0      0      0  0.001;];
Re = [0.01 0 0 0 0 0;
      0 0.01 0 0 0 0;
      0 0 0.01 0 0 0;
      0 0 0 0.01 0 0;
      0 0 0 0 0.01 0;
      0 0 0 0 0 0.01;];
Matrix_v = ones(6,1);
Matrix_u = [eye(3);zeros(3)];
Matrix_w = [zeros(3);eye(3)];
Pe = are(A',C'*(Re^-1)*C,Qv);
L = Pe*C'*(Re^-1);
De = zeros(size(B,2),size(C,1));
%% Closed-loop system
Ac = [(A-B*K) B*K;zeros(9) A-L*C];
C2 = B';
G = (C2*((A-B*K)^-1)*B)^-1;
Bc = [B*G;zeros(9,3)];
Cc = [C zeros(6,9)];
Dc = zeros(6,3);
%% Run Simulink
sim('design_pro',50)
%% Plot results
figg1 = figure(1);
hold on;
```

```
plot(tout,yout(:,1),"Color",[142, 207, 201]/255,'LineWidth',2.5);
plot(tout,yout(:,2),"Color",[255, 190, 122]/255,'LineWidth',2.5);
plot(tout,yout(:,3),"Color",[250, 127, 111]/255,'LineWidth',2.5);
plot(tout,yout(:,4),"Color",[130, 176, 210]/255,'LineWidth',2.5);
plot(tout,yout(:,5),"Color",[190, 184, 220]/255,'LineWidth',2.5);
plot(tout,yout(:,6),"Color",[231, 218, 210]/255,'LineWidth',2.5);
hold off;
grid on;
legend('$\phi$','$\theta$','$p$','$q$','$r$','$\psi$', ...
    'interpreter','latex');
xlabel('$t, \mathrm{\ \left(s\right)}$','interpreter','latex');
ylabel('$\mathrm{states \ responses}$', 'interpreter','latex');
% a = get(gca,'XTickLabel');
% set(gca,'XTickLabel',a,'FontName','Times','fontsize',12);
set(gcf,'renderer','painters');
filename = "DP1_stateresponse"+".pdf";
saveas(gcf,filename);
close(figg1);
figg2 = figure(2);
hold on;
plot(tout,yout(:,7),"Color",[142, 207, 201]/255,'LineWidth',2.5);
plot(tout,yout(:,8),"Color",[255, 190, 122]/255,'LineWidth',2.5);
plot(tout,yout(:,9),"Color",[250, 127, 111]/255,'LineWidth',2.5);
hold off;
grid on;
legend('$\sigma_{lat}$','$\sigma_{lon}$','$\sigma_{ped}$', ...
    'interpreter','latex');
xlabel('$t, \mathrm{\ \left(s\right)}$','interpreter','latex');
ylabel('$\mathrm{wrench}$', 'interpreter','latex');
% a = get(gca,'XTickLabel');
% set(gca,'XTickLabel',a,'FontName','Times','fontsize',12);
set(gcf,'renderer','painters');
filename = "DP1_wrench"+".pdf";
saveas(gcf,filename);
close(figg2);
%% Remove noise
Matrix_v = zeros(6,1);
%% Run Simulink
sim('design_pro',50)
%% Plot results
figg3 = figure(3);
hold on;
plot(tout,yout(:,1),"Color",[142, 207, 201]/255,'LineWidth',2.5);
plot(tout,yout(:,2),"Color",[255, 190, 122]/255,'LineWidth',2.5);
plot(tout,yout(:,3),"Color",[250, 127, 111]/255,'LineWidth',2.5);
plot(tout,yout(:,4),"Color",[130, 176, 210]/255,'LineWidth',2.5);
```

```
plot(tout,yout(:,5),"Color",[190, 184, 220]/255,'LineWidth',2.5);
plot(tout,yout(:,6),"Color",[231, 218, 210]/255,'LineWidth',2.5);
hold off;
grid on;
legend('$\phi$','$\theta$','$p$','$q$','$r$','$\psi$', ...
    'interpreter','latex');
xlabel('$t, \mathrm{\ \left(s\right)}$','interpreter','latex');
ylabel('$\mathrm{states \ responses}$', 'interpreter','latex');
% a = get(gca,'XTickLabel');
% set(gca,'XTickLabel',a,'FontName','Times','fontsize',12);
set(gcf,'renderer','painters');
filename = "DP1_stateresponse_nono"+".pdf";
saveas(gcf,filename);
close(figg3);
figg4 = figure(4);
hold on;
plot(tout,yout(:,7),"Color",[142, 207, 201]/255,'LineWidth',2.5);
plot(tout,yout(:,8),"Color",[255, 190, 122]/255,'LineWidth',2.5);
plot(tout,yout(:,9),"Color",[250, 127, 111]/255,'LineWidth',2.5);
hold off;
grid on;
legend('$\sigma_{lat}$','$\sigma_{lon}$','$\sigma_{ped}$', ...
    'interpreter','latex');
xlabel('$t, \mathrm{\ \left(s\right)}$','interpreter','latex');
ylabel('$\mathrm{wrench}$', 'interpreter','latex');
% a = get(gca,'XTickLabel');
% set(gca,'XTickLabel',a,'FontName','Times','fontsize',12);
set(gcf,'renderer','painters');
filename = "DP1_wrench_nono"+".pdf";
saveas(gcf,filename);
close(figg4);
%% open-loop system
s = tf('s');
sys1 = ss(A,B,C,D1);
sys2 = ss(A-B*K-L*C,L,K,De);
sys = K*((s*eye(9)-A+B*K+L*C)^-1)*L*C*((s*eye(9)-A)^-1)*B;
```

Conclusion

In conclusion, this project focused on the design of a flight control system for an unmanned helicopter in the hovering state. A linearized state space model for the inner-loop of the helicopter system was developed and used as the basis for the control law design. The designed control law was able to stabilize the system and drive the dynamics of the helicopter to the hovering state as quickly as possible from a given initial condition.

Frequency domain requirements on gain and phase margins were also considered, and it was shown that the designed control law performed well in this category. Simulation results showed that the system responded well to disturbances and was able to maintain stability in the presence of noise.

Overall, this project demonstrates the application of linear system theory and design techniques to the field of unmanned aerial vehicles. The designed flight control system can be used as a starting point for further research and development in the field of UAV control systems.