



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

© Copyright in this work rests with the authors. Please ensure that any reproduction or re-use is done in accordance with the relevant national copyright legislation.

## 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.

## 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 (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.

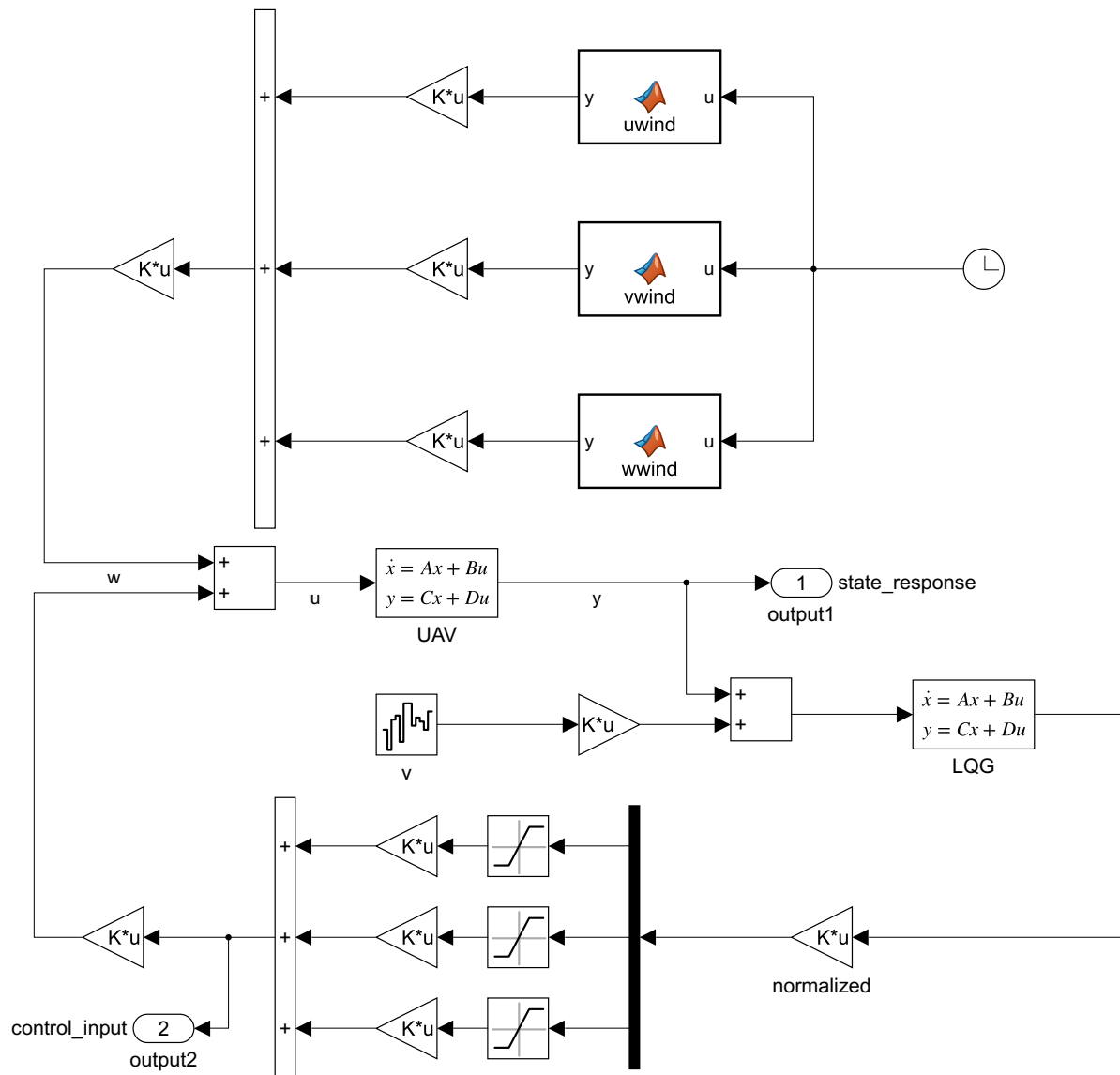
Figure 2a shows the state response of the system with measurement noise. It can be seen that the system stabilizes to the hovering state within a few seconds. Figure 2b shows the control input required to stabilize the system.

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:

---

```
1 clc; clf; clear all; close all;
2 %% Set the initial parameters
3 A = [0 0 1 0 0 0 0.0009 0 0;
4       0 0 0 0.9992 0 0 -0.0389 0 0;
5       0 0 -0.0302 -0.0056 -0.0003 585.1165 11.4448 -59.529 0;
6       0 0 0 -0.0707 267.7499 -0.0003 0 0 0;
7       0 0 0 -1 -3.3607 2.2223 0 0 0;
8       0 0 -1 0 2.4483 -3.3607 0 0 0;
```

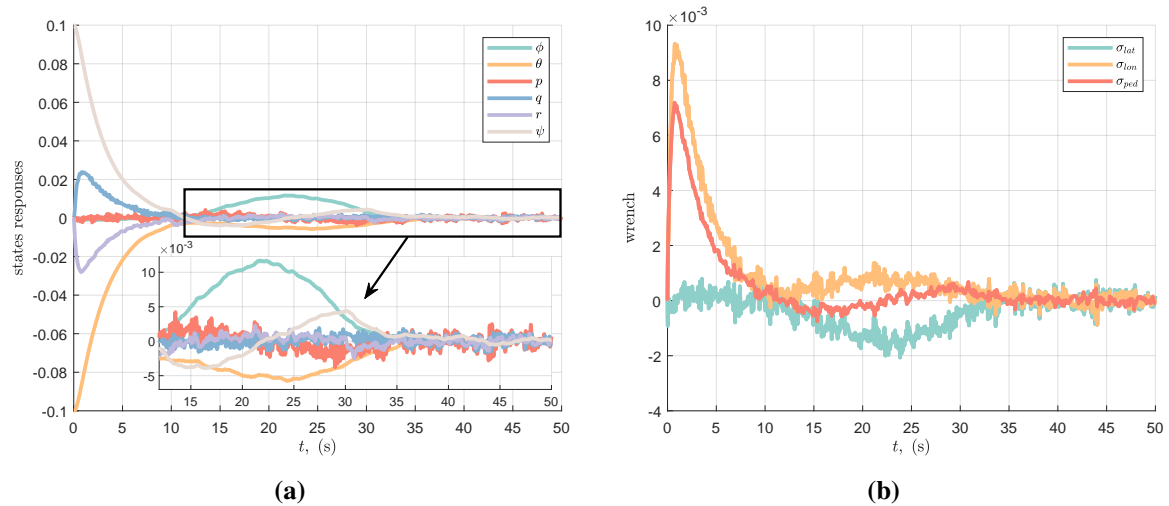


**Figure 1:** Block diagram for the simulation.

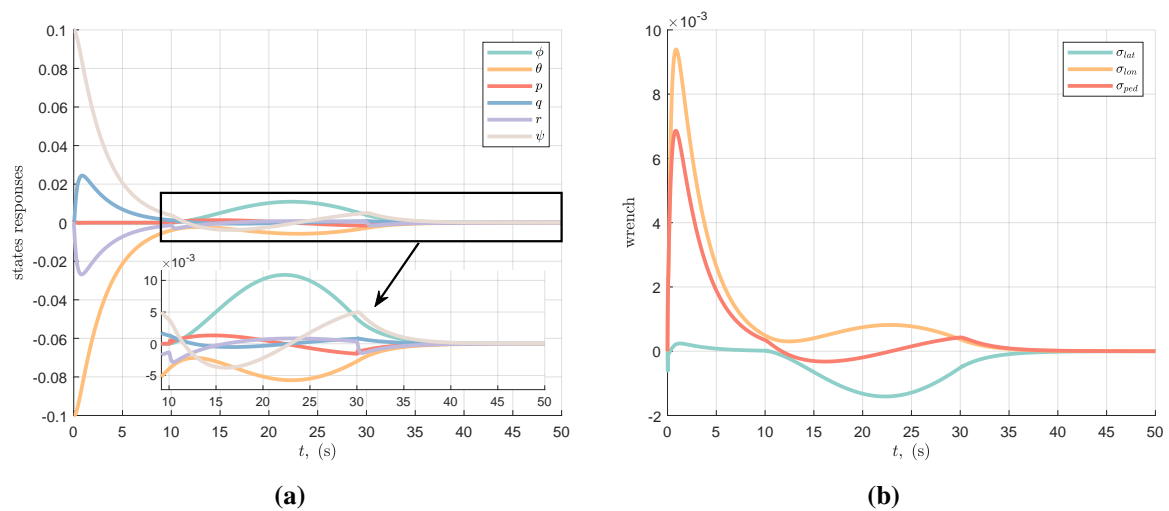
```

9      0 0 0.0579 0.0108 0.0049 0.0037 -21.9557 114.2 0;
10     0 0 0 0 0 0 -1 0 0;
11     0 0 0 0.0389 0 0 0.9992 0 0;];
12 B = [0 0 0;
13       0 0 0;
14       0 0 43.3635;
15       0 0 0;
16       0.2026 2.5878 0;
17       2.5878 -0.0663 0;
18       0 0 -83.1883;
19       0 0 -3.8500;
20       0 0 0;];
21 C = [1 0 0 0 0 0 0 0 0;
22       0 1 0 0 0 0 0 0 0;
23       0 0 1 0 0 0 0 0 0;

```



**Figure 2:** System performance with measurement noise.



**Figure 3:** System performance without measurement noise.

```

24     0 0 0 1 0 0 0 0 0;
25     0 0 0 0 0 0 1 0 0;
26     0 0 0 0 0 0 0 0 1;];
27 D = zeros(size(C,1));
28 D1 = zeros(6,3);
29 E = [0 0 0;
30      0 0 0;
31      -0.0001 0.1756 -0.0395;
32      0 0.0003 0.0338;
33      0 0 0;
34      0 0 0;
35      -0.0002 -0.3396 0.6424;
36      0 0 0;
37      0 0 0;];
38 x0 = [0;

```

```

39     -0.1;
40     0;
41     0;
42     0;
43     0;
44     0;
45     0;
46     0.1;];
47 %% Calculate LQR
48 % cost index for LQR
49 Q = [30 0 0 0 0 0 0 0 0 0;
50      0 10 0 0 0 0 0 0 0 0;
51      0 0 1 0 0 0 0 0 0 0;
52      0 0 0 1 0 0 0 0 0 0;
53      0 0 0 0 1 0 0 0 0 0;
54      0 0 0 0 0 1 0 0 0 0;
55      0 0 0 0 0 0 1 0 0 0;
56      0 0 0 0 0 0 0 10 0 0;
57      0 0 0 0 0 0 0 0 10 0];
58 Q = 3*Q;
59 R = [5.2638    1.5521    0.2130;
60      1.5521    4.7083    0.8346;
61      0.2130    0.8346    4.7579];
62 [K,S,CLP] = lqr(A,B,Q,R);
63 %% Calculate Kalman filter
64 Qv = [0.001    0    0    0    0    0    0    0    0;
65      0 0.001    0    0    0    0    0    0    0;
66      0    0 0.001    0    0    0    0    0    0;
67      0    0    0 0.001    0    0    0    0    0;
68      0    0    0    0 0.001    0    0    0    0;
69      0    0    0    0    0 0.001    0    0    0;
70      0    0    0    0    0    0 0.001    0    0;
71      0    0    0    0    0    0    0 0.001    0;
72      0    0    0    0    0    0    0    0 0.001];
73 Re = [0.01 0 0 0 0 0;
74      0 0.01 0 0 0 0;
75      0 0 0.01 0 0 0;
76      0 0 0 0.01 0 0;
77      0 0 0 0 0.01 0;
78      0 0 0 0 0 0.01];
79 Matrix_v = ones(6,1);
80 Matrix_u = [eye(3);zeros(3)];
81 Matrix_w = [zeros(3);eye(3)];
82 Pe = are(A',C'*(Re^-1)*C,Qv);
83 L = Pe*C'*(Re^-1);
84 De = zeros(size(B,2),size(C,1));

```

```

85 %% Closed-loop system
86 Ac = [(A-B*K) B*K; zeros(9) A-L*C];
87 C2 = B';
88 G = (C2*((A-B*K)^-1)*B)^-1;
89 Bc = [B*G; zeros(9,3)];
90 Cc = [C zeros(6,9)];
91 Dc = zeros(6,3);
92 %% Run Simulink
93 sim('design_pro',50)
94 %% Plot results
95 figg1 = figure(1);
96 hold on;
97 plot(tout,yout(:,1),"Color",[142, 207, 201]/255,'LineWidth',2.5);
98 plot(tout,yout(:,2),"Color",[255, 190, 122]/255,'LineWidth',2.5);
99 plot(tout,yout(:,3),"Color",[250, 127, 111]/255,'LineWidth',2.5);
100 plot(tout,yout(:,4),"Color",[130, 176, 210]/255,'LineWidth',2.5);
101 plot(tout,yout(:,5),"Color",[190, 184, 220]/255,'LineWidth',2.5);
102 plot(tout,yout(:,6),"Color",[231, 218, 210]/255,'LineWidth',2.5);
103 hold off;
104 grid on;
105 legend('$\phi$', '$\theta$', '$p$', '$q$', '$r$', '$\psi$', ...
106         'interpreter','latex');
107 xlabel('$t$, \mathrm{\left(s\right)}$', 'interpreter','latex');
108 ylabel('$\mathrm{states \ responses}$ ', 'interpreter','latex');
109 % a = get(gca,'XTickLabel');
110 % set(gca,'XTickLabel',a,'FontName','Times','fontsize',12);
111 set(gcf,'renderer','painters');
112 filename = "DP1_stateresponse"+"pdf";
113 saveas(gcf,filename);
114 close(figg1);
115 figg2 = figure(2);
116 hold on;
117 plot(tout,yout(:,7),"Color",[142, 207, 201]/255,'LineWidth',2.5);
118 plot(tout,yout(:,8),"Color",[255, 190, 122]/255,'LineWidth',2.5);
119 plot(tout,yout(:,9),"Color",[250, 127, 111]/255,'LineWidth',2.5);
120 hold off;
121 grid on;
122 legend('$\sigma_{lat}$ ', '$\sigma_{lon}$ ', '$\sigma_{ped}$ ', ...
123         'interpreter','latex');
124 xlabel('$t$, \mathrm{\left(s\right)}$', 'interpreter','latex');
125 ylabel('$\mathrm{wrench}$ ', 'interpreter','latex');
126 % a = get(gca,'XTickLabel');
127 % set(gca,'XTickLabel',a,'FontName','Times','fontsize',12);
128 set(gcf,'renderer','painters');
129 filename = "DP1_wrench"+"pdf";
130 saveas(gcf,filename);

```

```

131 close(figg2);
132 %% Remove noise
133 Matrix_v = zeros(6,1);
134 %% Run Simulink
135 sim('design_pro',50)
136 %% Plot results
137 figg3 = figure(3);
138 hold on;
139 plot(tout,yout(:,1),"Color",[142, 207, 201]/255,'LineWidth',2.5);
140 plot(tout,yout(:,2),"Color",[255, 190, 122]/255,'LineWidth',2.5);
141 plot(tout,yout(:,3),"Color",[250, 127, 111]/255,'LineWidth',2.5);
142 plot(tout,yout(:,4),"Color",[130, 176, 210]/255,'LineWidth',2.5);
143 plot(tout,yout(:,5),"Color",[190, 184, 220]/255,'LineWidth',2.5);
144 plot(tout,yout(:,6),"Color",[231, 218, 210]/255,'LineWidth',2.5);
145 hold off;
146 grid on;
147 legend('$\phi$', '$\theta$', '$p$', '$q$', '$r$', '$\psi$', ...
148         'interpreter','latex');
149 xlabel('$t$, \mathrm{\left(s\right)}$', 'interpreter','latex');
150 ylabel('$\mathrm{states \ responses}$$', 'interpreter','latex');
151 % a = get(gca,'XTickLabel');
152 % set(gca,'XTickLabel',a,'FontName','Times','fontsize',12);
153 set(gcf,'renderer','painters');
154 filename = "DP1_stateresponse_nono"+"pdf";
155 saveas(gcf,filename);
156 close(figg3);
157 figg4 = figure(4);
158 hold on;
159 plot(tout,yout(:,7),"Color",[142, 207, 201]/255,'LineWidth',2.5);
160 plot(tout,yout(:,8),"Color",[255, 190, 122]/255,'LineWidth',2.5);
161 plot(tout,yout(:,9),"Color",[250, 127, 111]/255,'LineWidth',2.5);
162 hold off;
163 grid on;
164 legend('$\sigma_{lat}$$', '$\sigma_{lon}$$', '$\sigma_{ped}$$', ...
165         'interpreter','latex');
166 xlabel('$t$, \mathrm{\left(s\right)}$', 'interpreter','latex');
167 ylabel('$\mathrm{wrench}$$', 'interpreter','latex');
168 % a = get(gca,'XTickLabel');
169 % set(gca,'XTickLabel',a,'FontName','Times','fontsize',12);
170 set(gcf,'renderer','painters');
171 filename = "DP1_wrench_nono"+"pdf";
172 saveas(gcf,filename);
173 close(figg4);
174 %% open-loop system
175 s = tf('s');
176 sys1 = ss(A,B,C,D1);

```

```
177 sys2 = ss(A-B*K-L*C, L, K, De);  
178 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.