2023 Spring, AE280

Auto-Control System of Aircraft

20210097 김서윤

Abstract

This project designs a control system that automatically controls an aircraft. I wrote MATLAB code that finds out the best controller gain with an arbitrary input that minimizes the peak time, which is an indicator of swiftness of response, while satisfying other stability constraints. It automatically calculates transfer function of the closed loop feedback system. Then, it reduces 3rd order transfer function to 2nd order to utilize performance indices. Plots of step response with K obtained and other Ks shows validity of the result. To compare with PID controller, my code plots root locus of the open loop system. Then, it compares the step response with PID controller.

I. Introduction

Since I am majoring in Electrical Engineering, I wanted to find the overlapping subject between two majors. Consequently, I thought control would be the best topic, in that it requires analysis based on both signal system and aircraft dynamics. The goal is to design a control system such that the dominant closed-loop system poles have satisfactory natural frequency and damping. To be specific, bank angle auto-control system for an aircraft in steady, level flight to achieve fastest peak time, while satisfying percent overshoot and settling time constraints will be designed.

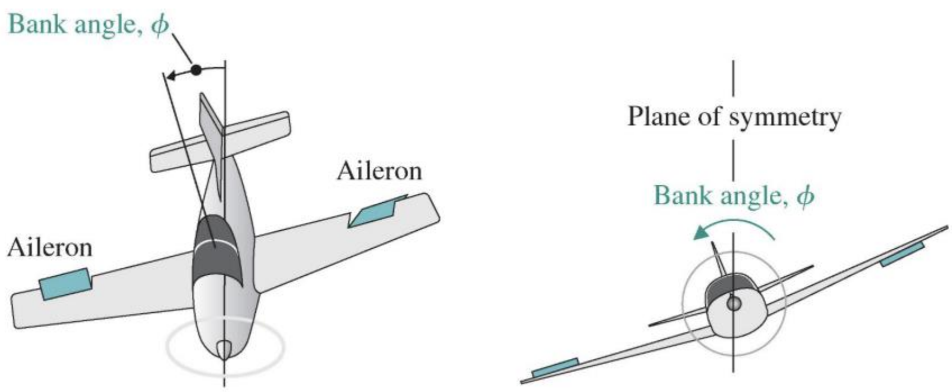

Figure 1. Controlling bank angle with Aileron

II. Theoretical Background

1. Transfer function

Transfer function describing the back angle output $\phi(s)$ to the aileron deflection input $\delta_a(s)$ is

$$\frac{\phi(s)}{\delta_a(s)} = \frac{k(s-c_0)(s^2+b_1 s+b_0)}{s(s+d_0)(s+e_0)(s^2+f_1 s+f_0)}$$

$s+d_0$ is associated with the spiral mode, and $s+e_0$ is associated with the roll subsidence mode. $s^2+f_1 s+f_0$ represent Dutch roll motion. These are three main modes in roll, yaw and lateral motion. In steady, level flight, angle of attack is so small that the Dutch roll mode generally cancels out of the transfer function with the $s^2+b_1 s+b_0$ term. Also, we can ignore the spiral mode since it is essentially a yaw motion. We assume $s-c_0$ is negligible, which represents a gravity effect that causes the aircraft to sideslip as it rolls. Then the transfer function to obtain a single degree of freedom approximation becomes

$$\frac{\phi(s)}{\delta_a(s)} = \frac{k}{s(s+e_0)}$$

Typical aileron actuator is a simple first order system model, where $e(s)$ is an error signal, $\phi_d(s)-\phi(s)$.

$$\frac{\delta_a(s)}{e(s)} = \frac{a}{s+a}$$

2. Performance of Second-Order System

Second order system is represented as equation below, where Y(s) is the output to an input R(s). ξ is the damping ratio, and ω_n is natural frequency.

$$Y(s) = \frac{{\omega_n}^2}{s^2+2\xi\omega_n s+{\omega_n}^2}R(s)$$

ξ is an important indicator of stability of the system because root locus varies with ξ as ω_n is constant as shown in the figure below.

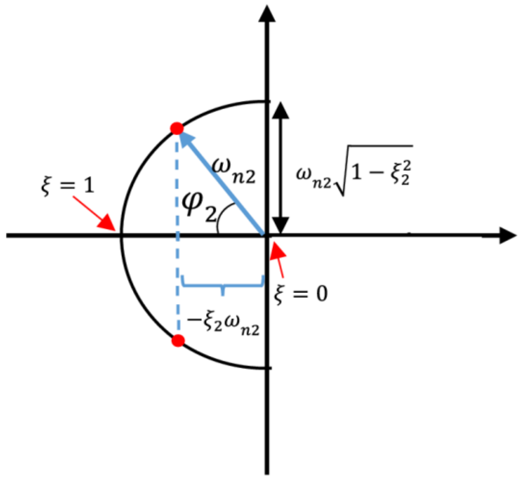

Figure 2. 2nd Order System

In second order system, peak time, percent overshoot, and settling time to a step response are

$$T_p = \frac{\pi}{\omega_n\sqrt{1-\xi^2}}$$

$$P.O. = 100\, e^{\frac{-\pi\xi}{\sqrt{1-\xi^2}}}$$

$$T_s = \frac{4}{\xi\omega_n}$$

Peak time is the indicator of swiftness of the response that represents 0-100% rise time in underdamped system. Percent overshoot is defined as

$$P.O. = \frac{M_{pt} - fv}{fv} \times 100\%$$

where M_{pt} is the peak value and fv is the final value. Settling time is defined as the time required for the system to settle within a certain percentage. We use 2% criterion in this analysis.

3. Reducing 3rd order TF to 2nd order

Since the transfer function of our system is 3rd ordered, we should reduce it to 2nd order system to use performance measuring indicators of 2nd order system.

$$T(s) = \frac{1}{1 + c_1 s + c_2 s^2 + c_3 s^3}$$

$$G_L(s) = \frac{1}{1 + d_1 s + d_2 s^2}$$

Actual transfer function is $T(s)$ and approximated transfer function is $G_L(s)$. If we define $M(s)$ and $\Delta(s)$ as the numerator and denominator of $T(s)/G_L(s)$,

$$M_{2q} = \sum_{k=0}^{2q} \frac{(-1)^{k+q}\, M^{(k)}(0)\, M^{(2q-k)}(0)}{k!\,(2q-k)!}$$

$$\Delta_{2q} = \sum_{k=0}^{2q} \frac{(-1)^{k+q}\, \Delta^{(k)}(0)\, \Delta^{(2q-k)}(0)}{k!\,(2q-k)!}$$

Setting $\Delta = \mathrm{M}$ for q = 1, 2, … we can obtain coefficients of $G_L(s)$.

III. Implementation

1. Modeling

When entering arbitrary input in MATLAB, it automatically obtains transfer function of the closed loop system. Below is the example of inputs.

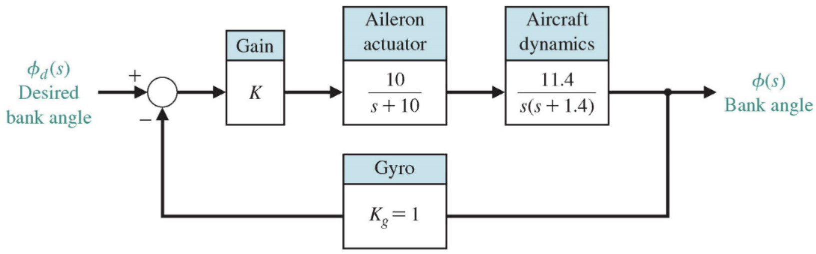

Figure 3. Arbitrary closed loop feedback system

For this case, a = 10, b = 1.4, c = 11.4. To assemble individual elements into an integrated closed loop transfer function, I used three MATLAB commands.

$$tf(num, den)$$

$$series\ (Aileron\_Actuator, Aileron\_Dynamics)$$

$$feedback\ (G, 1)$$

tf command transforms matrix in to transfer function. For example, $tf([a],[1\ a])$ becomes

$$\frac{a}{s+a}$$

$series$ command multiplies two transfer functions. $feedback$ command utilizes open loop transfer function into closed loop transfer function with an input feedback value. In this case, G is the open loop transfer function, and 1 represents unity negative feedback.

```
%% Modeling

num1 = a; den1 = [1 a];                          % Aileron actuator
num2 = c; den2 = [1 b 0];                        % Aircraft dynamics
Kg = 1;                                          % Gyro

%% TF functions

Controller = 0.1563;
Aileron = tf(num1, den1);
Aircraft = tf(num2, den2);

G = Controller * series(Aileron, Aircraft);      % Open-loop TF
T = feedback(G, 1);                              % Closed-loop TF
```

Figure 4. Modeling and TF functions

Since the result was 3rd order system, I reduced it to the 2nd order.

$$T = \frac{17.82}{s^3 + 11.4s^2 + 14s + 17.82}$$

2. Reducing Order

The equation in Figure 5 is the result of a series of calculations mentioned in theoretical background. I used MATLAB

$$df\ = diff(f,\ s)$$

$$subs\ (df,\ s,\ 0)$$

commands to differentiate characteristic equations and substitute 0.

I used MATLAB

$$solve(eqn, d2)$$

command to represent d2 with respect to variable k. Also, I used

$$vpa()$$

to convert complex expression to intended forms in significant digits.

The coefficients of simplified model can be obtained with this code.

```
%% 3rd order to 2nd order
syms d1 d2 k;
eqn = -1/3*(b/(k*c))*(6/(k*a*c))+1/4*(2*(a+b)/(k*a*c))^2 == d2^2;
sold2 = solve(eqn, d2);
d2 = vpa(sold2(2));                                  % d2 = A/k
d1 = sqrt(vpa(2*d2-2*(a+b)/(k*a*c)+(b/(k*c))^2));   % d1 = (B/k^2-C/k)^1/2
```

Figure 5. Reducing order

Actual result with intended form is as follows. For further analysis, I set coefficients of d2 and d1 as A, B, C.

$$d2 = \frac{0.0885747758692234233444437774581 84}{k}$$

$$d1 = \left(\frac{0.015081563558017851646660510926439}{k^2} - \frac{0.022850448261553153311112445083632}{k}\right)^\wedge(1/2)$$

In this case,

$$A = 0.008875, \qquad B = 0.01582, \qquad C = 0.022850$$

3. Optimize to find K

Using methods in II and coefficients A, B, C obtained in process 2, relationships between variables are calculated as below. Since the relation between peak time, percent overshoot, and settling time is nonlinear, I used

$$fmincon()$$

and nonlinear conditions in MATLAB.

In $fmincon()$, $x(1)$ is damping ratio, $x(2)$ is settling time, and $x(3)$ is percent overshoot. fun represents peak time, which has to be minimized. For an underdamped system, damping ratio should be between 0 and 1. So the lower bound was set to $[0\ 0\ 0]$. Also, upper bound was set to $[1\ 10\ 20]$ to give 10 seconds and 20% constraints of settling time and percent overshoot.

```
%% Finding optimized K

% x(1) = zeta, x(2) = settling time, x(3) = P.O.

fun = @(x) pi/sqrt(1-x(1)^2) * sqrt(A/B) * sqrt(4*A*x(1)^2+C);

x0 = [0 0 0];
lb = [0 0 0];
ub = [1 10 20];

res = fmincon(fun, x0, [], [], [], [], lb, ub, @nonlcon);

K = B/(4*A*res(1)^2+C);
```

Figure 6. Finding optimized K

The relationship between peak time, settling time and percent overshoot was calculated like below. The coefficients calculated in 2 were subsituted.

$$K = \frac{B}{4A\zeta^2 + C}, \quad \omega_n = \sqrt{\frac{K}{A}}$$

$$T_p = \frac{\pi}{\sqrt{1-\zeta^2}}\sqrt{\frac{A}{B}}\sqrt{4A\zeta^2 + C}, \quad T_s = \frac{4}{\zeta}\sqrt{\frac{A}{B}}\sqrt{4A\zeta^2 + C}, \quad P.O. = 100\, e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}}$$

This is the code for nonlinear constraints.

```
%% Nonlinear Constraints

function [c, ceq] = nonlcon(x)
    A = 0.088575; B = 0.015082; C = 0.022850;

    % Nonlinear equality constraints
    c = [];
    ceq = [x(2) - 4/x(1) * sqrt(A/B) * sqrt(4*A*x(1)^2 + C);
        x(3) - 100 * exp(-x(1) * pi / sqrt(1 - x(1)^2))];
end
```

Figure 7. Nonlinear Constraints

IV. Result

The result of optimization are as follows.

$$K = 0.156281$$

$$\zeta = 0.455950,\ T_s = 6.604578s,\ P.O. = 19.999994,$$

$$T_P = 2.657415s,$$

The best value of K that makes minimizes the peak time while satisfying the constraints of damping ratio, settling time, and percent overshoot was about 0.1563.

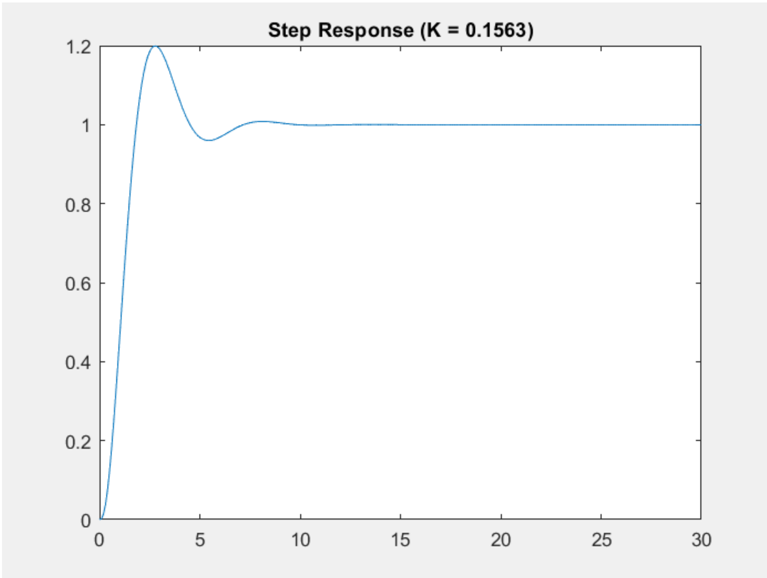

Figure 8. Step Response (K = 0.1563)

This is the step response of the system obtained by using

$$step()$$

command in MATLAB. We can verify that peak time is quite fast, and satisfies all constraints. Also, it converges to the desiring output.

Figure 9 is the graph of various step response with changes in K. We can easily find that as K goes below 0.15, peak time and settling time becomes larger. In contrast, as K goes above 0.15, percent overshoot becomes larger.

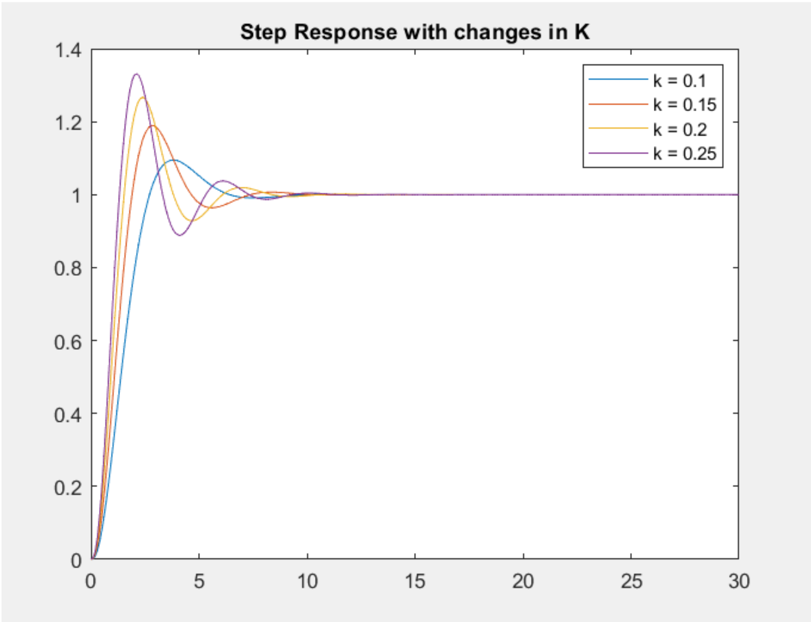

Figure 9. Step Response with changes in K

V. Discussion

1. Comparing with PID controller

This is the root locus of the closed loop system. I used

$$rlocus()$$

command in MATLAB. To find PID controller, I increased K to the border of instability. Then, I computed ultimate gain and ultimate period.

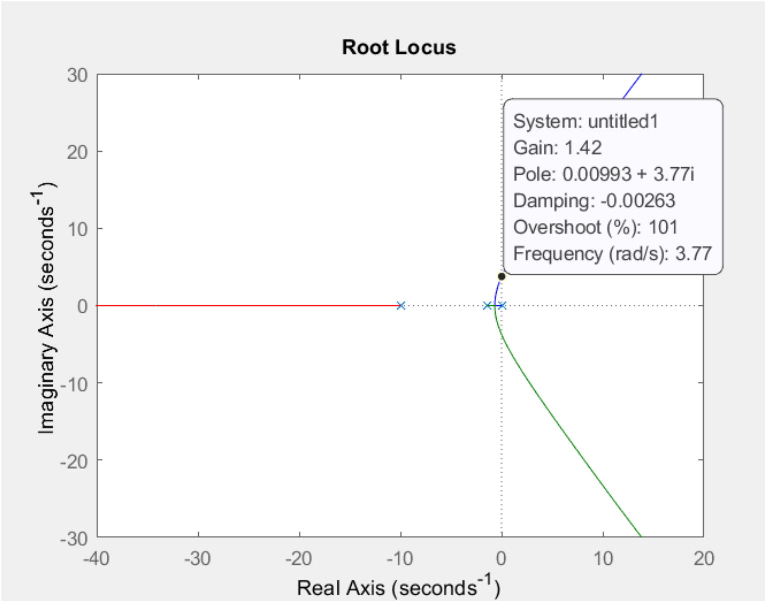

Figure 10. Root Locus of an open loop TF

From the graph, ultimate gain and ultimate period are 1.42 and 1.67s. Using Ziegler–Nichols PID tuning,

$$K_P = 0.6K_U = 0.852, \quad K_I = \frac{1.2K_U}{T_U} = 1.02, \quad K_D = \frac{0.6\,K_U T_U}{8} = 0.178$$

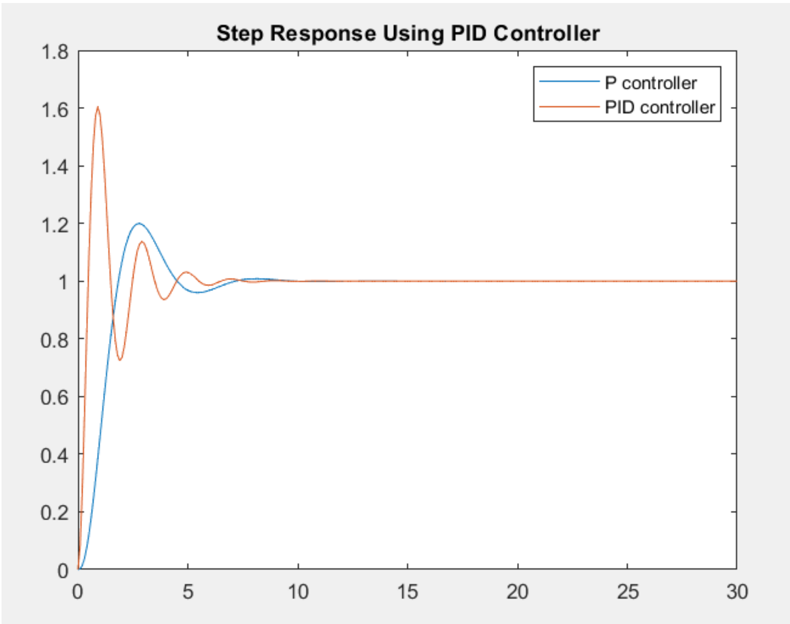

Figure 11. Step response Using PID Controller

With Ziegler-Nichols PID tunning, percent overshoot is almost 60%. So, in this example, my controller is more suitable for stable control of an aircraft.

VI. Conclusion

This MATLAB code outputs proportional controller gain that minimizes the peak time while satisfying all the stability constraints, so that enables auto controlling of an aircraft with arbitrary closed loop system. For example, when entered $a = 10$, $b = 1.4$, $c = 11.4$, the output was 0.1563. Simultaneously, damping ratio was 0.4560, settling time was 6.605s, percent overshoot was 20%, and peak time was 2.657s. The step response verified that it converges to the desiring output. Also, when compared to the result of PID controller selected by using Ziegler-Nichols tunning method, previously obtained controller was more suitable for stability control.

<MATLAB Commands>

Optimization	fmincon, nonlcon
Reducing Order	solve, diff, subs, vpa
Transfer Functions	feedback, tf, series
Root Locus	rlocus
Step Response	step, plot, hold on

VII. Reference

[1] Brian L. Stevens, Frank L. Lewis, Eric N. Johnson - Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems, 3rd Edition

[2] Richard C. Dorf, Robert H. Bishop - Modern Control Systems, 13th Edition