TTK4190 Guidance and Control of Vehicles

Assignment 2

Written Fall 2020 By Magnus Dyre-Moe, Patrick Nitschke and Siawash Naqibi

Problem 1 - Open-loop analysis

a)

With no wind, the ground speed V_g is equal to the air speed V_a. Such that $V_g = V_a = 580$ km/h

b)

$$\begin{aligned} \beta &= \chi - \psi \\ \beta &= \sin^{-1}\left(\frac{v_r}{U_r}\right) = \sin^{-1}\left(\frac{v_r}{\sqrt{u_r^2 + v_r^2}}\right) \end{aligned} \tag{1}$$

c)

Defining the system model and using Matlab function damp(A) of the system matrix yields the following output:

Pole	Damping	Frequency (rad/TimeUnit)	Time Constant (TimeUnit)
-3.16e-01 + 2.83e+00i	1.11e-01	2.85e+00	3.16e+00
-3.16e-01 - 2.83e+00i	1.11e-01	2.85e+00	3.16e+00
-2.88e+00	1.00e+00	2.88e+00	3.47e-01
-5.60e-04	1.00e+00	5.60e-04	1.79e+03
-7.50e+00	1.00e+00	7.50e+00	1.33e-01

Table 1: Result of *damp(A)* in Matlab.

Based on lecture notes from september 14th we assume the pole at $-5.60e-04$ is the integrator in the system equation for lateral movement. The remaining infor can then be extracted from table 1. We then have the fastest 1st order response as spiral mode, the fastest 1st order response as subsidiary roll mode and the 2nd order response as Dutch roll.

The eigenvalues for lateral movement can be written as

$$\lambda(\lambda + e)(\lambda + f)(\lambda^2 + 2\zeta_D\omega_D\lambda + \omega_D^2) = 0 \tag{2}$$

This results in the following values:

$$\begin{aligned} e &= -2.88 \\ f &= -7.50 \\ \omega_D &= 2.85 \\ \zeta_D &= 0.111 \end{aligned} \tag{3}$$

Dutch roll mode can be split into a motion in roll and a motion in yaw.

For a higher relative damping ratio the motion would go towards a more critically damped system. Hence, the oscillations would be damped.

d)

The spiral-divergence mode can also be retrieved from Table 1, where we see that $\lambda = -e = -2.88$. Negative eigenvalues yields a stable response.

e)

Roll or subsidiary roll can also be retrieved from Table 1, where we see that $\lambda = -f = -7.50$. Negative eigenvalues yields a stable response.

Problem 2 - Autopilot for course hold using aileron and successive loop closure

a)

Smallest signed angle , ssa($\cdot$), finds the smallest difference between two angles. This is beneficial as it finds a difference in angles limited to a max of 180. This guarantees that we do not have any unwanted motions, such as a plane doing a barrel roll to reach reference angle.

b)

From the state-space model, we get the equation for the pitch rate, $\dot{p}$:

$$\dot{p} = -10.6\beta - 2.87p + 0.46r - 0.65\delta_a$$

Since we assume an autopilot system, $r = 0$ and $\beta = 0$. With this, we can also take the Laplace transform to achieve:

$$sp = -2.86p - 0.65\delta_a$$
$$(s + 2.87)p = -0.65\delta_a$$
$$\frac{p}{\delta_a} = \frac{-0.65}{s + 2.87}$$

By inspecting Figure 1 in the problem set, we see that:

$$a_{\phi_1} = 2.87$$
$$a_{\phi_2} = -0.65$$

c)

Calculating the transfer function for roll motion we arrive at:

$$H_{\phi/\phi^c}(s) = \frac{sa_{\phi_2}k_{p_\phi} + k_{i_\phi}a_{\phi_2}}{s^3 + s^2(a_{\phi_1} + k_{p_\phi}a_{\phi_2}) + sa_{\phi_2}k_{p_\phi} + a_{\phi_2}k_{i_\phi}} \tag{4}$$

Our model is identical to Beard McLain figure 6.6. In the assignment text we are given $\delta_a^{max} = 30$, $e_\phi^{max} = 15$ and $\zeta_\phi = 0.707$. Using equations (6.7), (6.8) and (6.9) from Beard and McLain we can find numerical values for our parameters.

$$k_{p_\phi} = \frac{\delta_a^{max}}{e_\phi^{max}}\text{sign}(a_{\phi_2}) = \frac{30}{15}(-1) = -2 \tag{5}$$

$$\omega_{n_\phi} = \sqrt{a_{\phi_2}\frac{\delta_a^{max}}{e_\phi^{max}}} = \sqrt{0.65 \cdot \frac{30}{15}} \approx 1.14 \tag{6}$$

$$k_{d_\phi} = \frac{2\zeta_\phi\omega_{n_\phi} - a_{\phi_1}}{a_{\phi_2}} = \frac{2 \cdot 0.707 \cdot 1.14 - 2.87}{-0.65} \approx 1.94 \tag{7}$$

We use our knowledge of successive loop closure to define ω_{n_χ} by the property:

$$\omega_{n_\chi} = \frac{1}{W}\omega_{n_\phi} \tag{8}$$

In order to ensure that our response is stable we use $W = 10$ such that

$$\omega_{n_\chi} = \frac{1}{10}\omega_{n_\phi} \approx 0.114 \tag{9}$$

To find the input range for our gains, we used the Matlab function *rlocus*, which tests different feedback gain values and plots the movement of poles for these values. Our result is shown below:

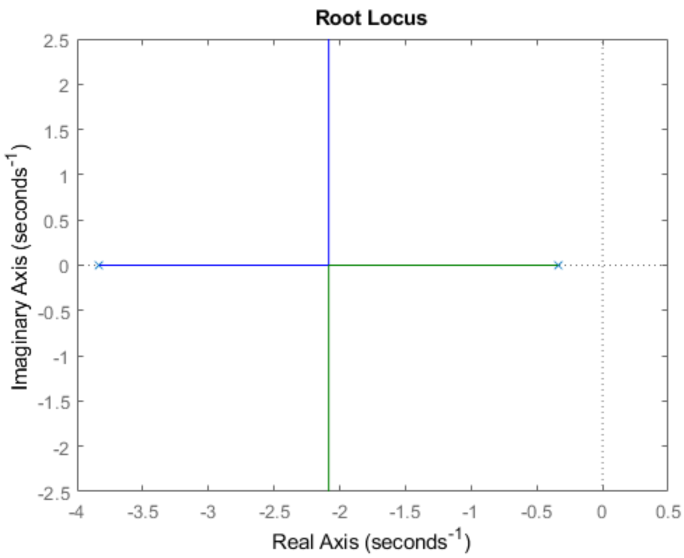

Figure 1: Attempt to use root-locus analysis to find our integral gain. Here, $\boldsymbol{H}_{\phi/\phi^c}$ was the transfer function plotted, with k_{i_ϕ} set to 0.

However, we did not understand which gain value was being varied and we saw that the poles of our system were already on the left half plane. Thus, we do not have a concrete range for our integral gain.

Yet, we can find the equations for our integrator term in equations (6.12) and (6.13) in Beard McLain:

$$k_{p_\chi} = 2\zeta_\chi \omega_{n_\chi} \frac{V_g}{g} = 2 \cdot 1 \cdot 0.114 \cdot \frac{580}{9.81 \cdot 3.6} \approx 3.74 \tag{10}$$

$$k_{i_\chi} = \omega_{n_\chi}^2 \frac{V_g}{g} = 0.114^2 \cdot \frac{580}{9.81 \cdot 3.6} \approx 0.21 \tag{11}$$

By choosing $\zeta_\chi = 1$.

d)

Having multiple integrators can lead to nasty stability properties. Using an integrator gives a phase angle of 90°. Using two integrators leads to a phase angle of 180°. When the face angle is 180 stability becomes a real concern. Sacrificing the integrator in the inner loop will lead to a stationary error in roll. However, the error can be corrected by using an integrator in course angle. Going forward we will only use an integrator for course.

e)

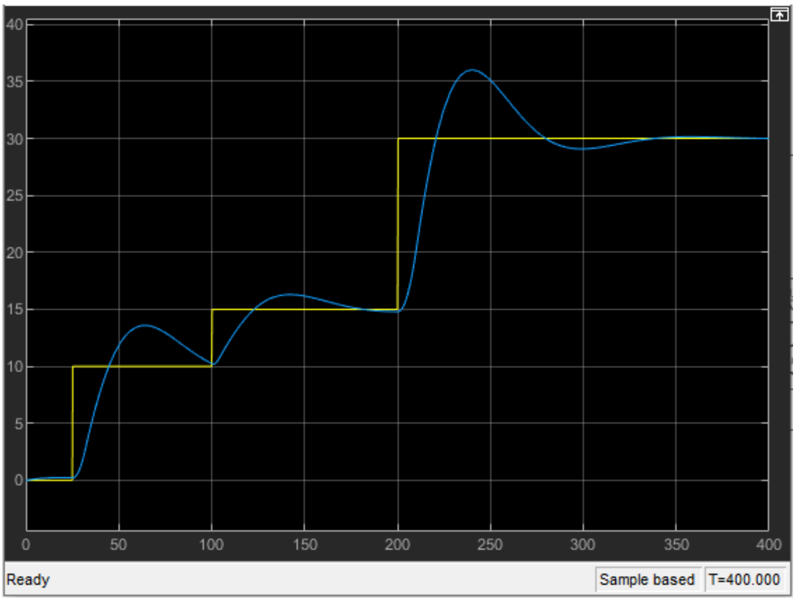

Figure 2: Response - yellow is reference and blue is actual state

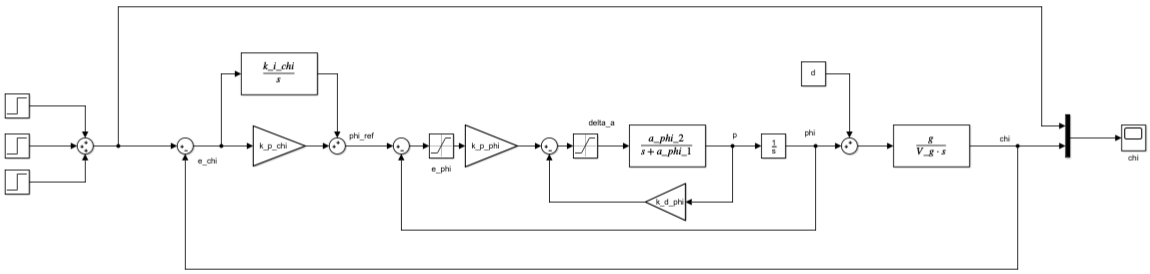

Figure 3: Simulink

f)

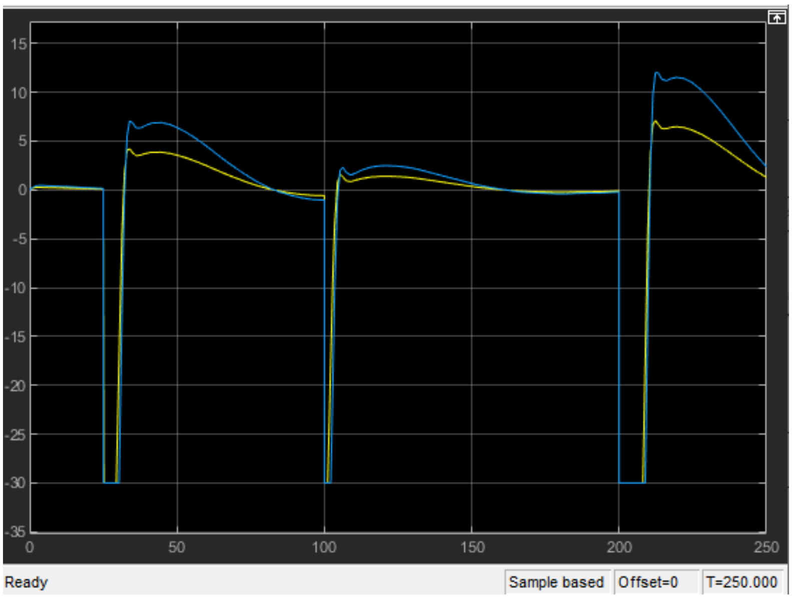

Figure 4: Aileron

g)

If the input is saturated the integrator will accumulate error over a certain amount of time. This may lead to a slower correction. A solution to this is to use an anti-windup scheme.

A Matlab code

A.1 Prob1.m

```matlab
display('Magnus has a')
pp = 'small pp'
display(pp)
```

References