Frequency Domain Analysis of Roll Dynamics

Due May 4 by 11:59pm **Points** 100 **Submitting** a text entry box

The goal of this assignment is to do a frequency domain analysis of the roll dynamics using data from NED_TheC++FlightSimulator. You will create a Bode plot from roll data, and you will identify a transfer function that behaves like the roll dynamics of the airplane. The properties of this transfer function can be useful for successive-loop closure PID control.

System Identification

System identification is a process in which mathematical models are derived from experimental data. You will apply system identification techniques to find a simplified model of the roll dynamics of the airplane. This assignment uses system identification techniques based on Bode plots. To learn more about this topic, study Sections 7.1 and 7.2 of the **ME310_Textbook (https://drive.google.com/file/d/1KuNKXQUv4sNAF6F4pHpH5LmBbVQqZ0O8/view?usp=sharing)**.

The equations of motion from the aileron command δ_a input to the roll rate ω_x of the plane are complicated and difficult to determine based solely on physical principles. System identification allows us to determine simpler models directly from experimental data. For example, it may be possible to determine from experimental data a first-order transfer function

$$\frac{\omega_x}{\delta_a} = \frac{b_0 s + b_1}{s + a_0}$$

or a second order transfer function

$$\frac{\omega_x}{\delta_a} = \frac{b_2 s^2 + b_1 s + b_0}{s^2 + a_1 s + a_0}$$

that sufficiently captures the roll dynamics of the plane. System identification consists of experimentally finding values for the coefficients $a_0, a_1, \ldots$ and $b_0, b_1, \ldots$ that make the transfer functions best fit the experimental data.

Bode Plots From Experimental Data

Section 10.4: Bode Plots from Experimental Data of the AutonomousFlight.pdf book explains how to create Bode plots using experimental data. The process begins by taking the discrete Fourier transform of the input and output data:

$$U = \mathbf{FFT}(u)$$

and

$$Y = \mathbf{FFT}(y)$$

The FFT function is the Fast-Fourier Transform algorithm. It applies the discrete Fourier transform to the input u and output y time-signals. The results, U and Y respectively, are the corresponding Fourier transforms of the data. They are related to the frequency response of the data. A third frequency-domain signal, Z, is defined by the equation

$$Z = \frac{Y}{U}$$

Because U and Y are arrays of data, division is an element-wise operation performed element by element. The frequency response magnitude Mag can be calculated from Z as follows:

$$\mathrm{Mag} = \sqrt{\mathrm{Real}(Z)^2 + \mathrm{Imag}(Z)^2}$$

The Mag calculation is an element-wise calculation. The frequency response phase angle Phase can also be calculated from Z as follows:

$$\mathrm{Phase} = \mathrm{atan2}\left(\mathrm{Imag}(Z), \mathrm{Real}(Z)\right)$$

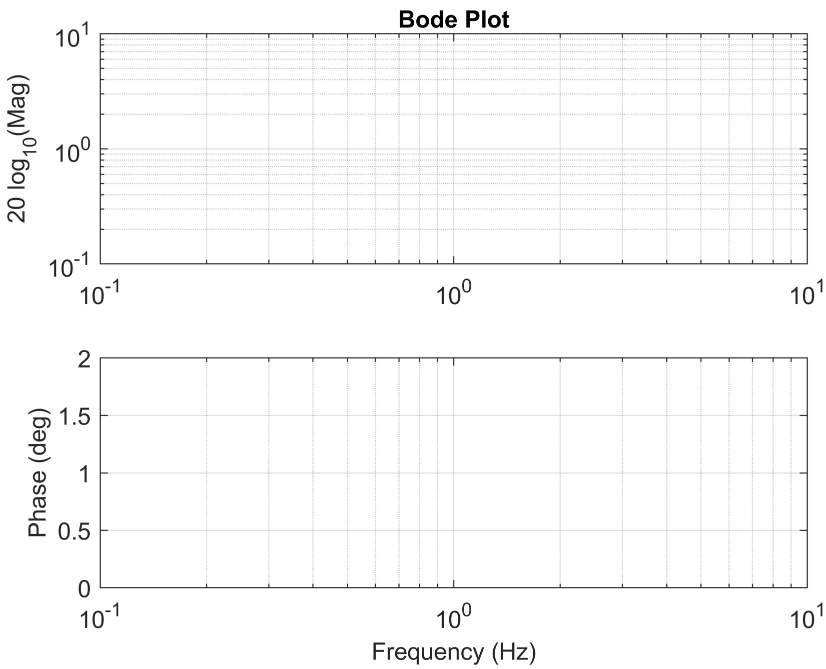
The frequency step size Δf (Hz) can be determined. If the total length of the time-signal u is N (which is also the length of the signal y), and the time-step between samples is Δt (s), then the frequency step size is

$$\Delta f = \frac{1}{N\Delta t}$$

An array of frequencies f between 0 and $\frac{1}{2\Delta t}$ Hz is found from the following equation where f_n is the n^{th} data point in the array.

$$f_n = \frac{n}{\Delta t N},\ n = 0, \ldots, \frac{N}{2} - 1$$

This array of frequencies corresponds to the first $\frac{N}{2} - 1$ points in the Mag and Phase arrays. A frequency response plot is found by creating two graphs. The first plots the Mag array on the y-axis versus the f array on the x-axis. The second plots the Phase array on the y-axis and the f array on the x-axis. A Bode plot is similar, except that the x-axis is in log scale, and the magnitude plot has $20\log_{10}(\mathrm{Mag})$ on the y-axis instead of Mag.



Sample MATLAB Program

Complete the following MATLAB script to fit a transfer function model to the Bode plot created by the data in the following file:

[RollDataOnly.xlsx (https://byui.instructure.com/courses/232306/files/105806578?wrap=1)](https://byui.instructure.com/courses/232306/files/105806578?wrap=1) [(https://byui.instructure.com/courses/232306/files/105806578/download?download_frd=1)](https://byui.instructure.com/courses/232306/files/105806578/download?download_frd=1)

```
clear all
close all
clc

%% import the data
[matlabFile,path] = uigetfile({'*.xlsx;*.csv'}, ...
    'Select The Flight Simulator data');
sim = readtable([path,matlabFile]);

% extract the data
t = sim.Time_s;%Time (s)
dt_avg = mean(t(2:end)-t(1:end-1)); %(s) average time-step
N = length(t); %Number of data points
xG = sim.xRate_rad_s; %(rad/s) x-axis gyrometer angular velocity
delta_a = sim.delta_a; %[-1,1] aileron command

%% Get the frequency domain information for the Bode Plot From Data
U = fft(delta_a); %Discrete Fourier Transform of Input
Y = fft(xG); %Discrete Fourier Transform of Output
Z = Y./U; %Frequency domain transfer function
%ZMag = sqrt(real(Z).^2+imag(Z).^2); %Transfer function magnitude
ZMag = abs(Z); %Transfer function magnitude: ZMag = sqrt(real(Z).^2+imag(Z).^2);
ZPhase = angle(Z); %Transfer function Phase angle: ZPhase = atan2(imag(Z),real(Z));
```

```
Nby2 = floor(N/2); %Get half the length of the data
Mag = ZMag(1:Nby2); %Get the Bode plot Mag data
Phase = ZPhase(1:Nby2); %Get the Bode plot Phase data
df = 1/(N*dt_avg); %Get the frequency step-size
f = 0:df:df*(Nby2-1); %(Hz) get the frequency array

%% Get the frequency domain information for the Bode plot from Sys Id
s = 1i*2*pi*f; %s = j*w
b0 = ; %Guess the numerator coefficient (Your assignment is to find this!)
a0 = ; %Guess the denominator coefficient (Your assignment is to find this!)
TFsys = b0 ./ (s+a0); %Get the transfer function
Magsys = abs(TFsys); %Get the transfer function Mag
Phasesys = angle(TFsys); %Get the transfer function Phase

%% Plot the Bode plot from data versus the Bode plot from Sys Id
figure
subplot(211)
semilogx(f,20*log10(Mag), f, 20*log10(Magsys),'--')
title('Bode Plot')
ylabel('20log_1_0(Mag)')
xlimits = [0.1,10];
xlim(xlimits)
legend('Data','Model')
grid on
subplot(212)
semilogx(f,Phase*180/pi, f, Phasesys*180/pi,'--')
ylabel('Phase (deg)')
xlabel('Frequency (Hz)')
grid on
xlim(xlimits)
legend('Data','Model')

%% Run a time-domain simulation to compare the model with the data
A = -a0;
B = 1;
C = b0;
Fd = expm([A*dt_avg,B*dt_avg;0,0]);
Ad = Fd(1,1);
Bd = Fd(1,2);
y = zeros(N,1);
x = 0;
for ii = 1:N
    y(ii) = C*x;
    x = Ad*x+Bd*delta_a(ii);
end

figure
plot(t,xG, t, y,'--')
legend('Data','Model')
xlabel('Time (s)')
ylabel('Roll Rate (rad/s)')
title('Model Prediction of Roll Rate')
xlim([0,20])
```

What to submit

In the text entry box, submit your model transfer function in the following form:

TF = b0/(s+a0)

For example, you might submit something like this:

TF = 8/(s+12)