Project 2

Frequency Response and Experimental Transfer Function of a System

A Project by,
Sagar Jagtap (903935334)
Arun Ramachandran (903928488)

MEE 5318 Instrumentation and Measurement Systems
Florida Institute of Technology, Melbourne, FL
Spring 2020

1) Hardware Connections

Figure 1 shows the circuit connections that were made. To construct a bandpass filter, resistor and capacitor values were taken to be R_1= 20 kΩ, R_2= 5.6 kΩ, C_1= C_2= 27e-9. This combination of components in out circuit results in a bandpass filter with a low cut-off frequency of f_L = 294.73 Hz, and a high cut-off frequency of f_H = 1052.6 Hz. The center frequency for this setup is calculated to be f_C = 556.9 Hz.

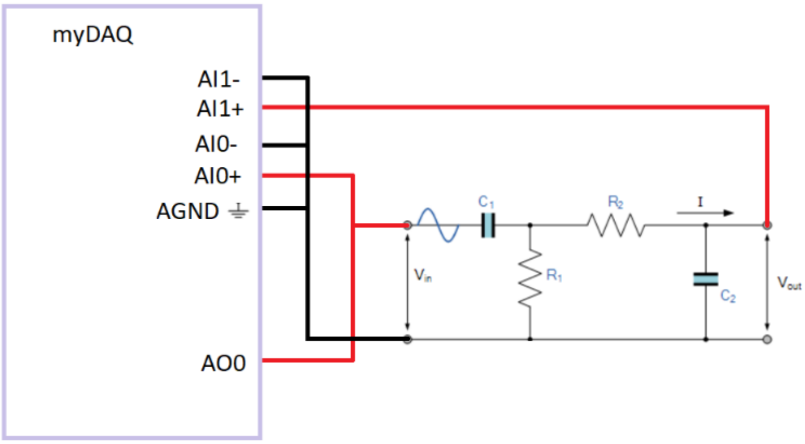

Figure 1: Circuit diagram

Before we designed the VI, a preliminary test was done to check whether the filter functions as intended using the **NI ELVIS II Bode Analyzer** tool. Figure 2 shows the resulting frequency plots generated by the tool. At the center frequency, the magnitude plot is the highest, and the phase plot is at zero degrees.

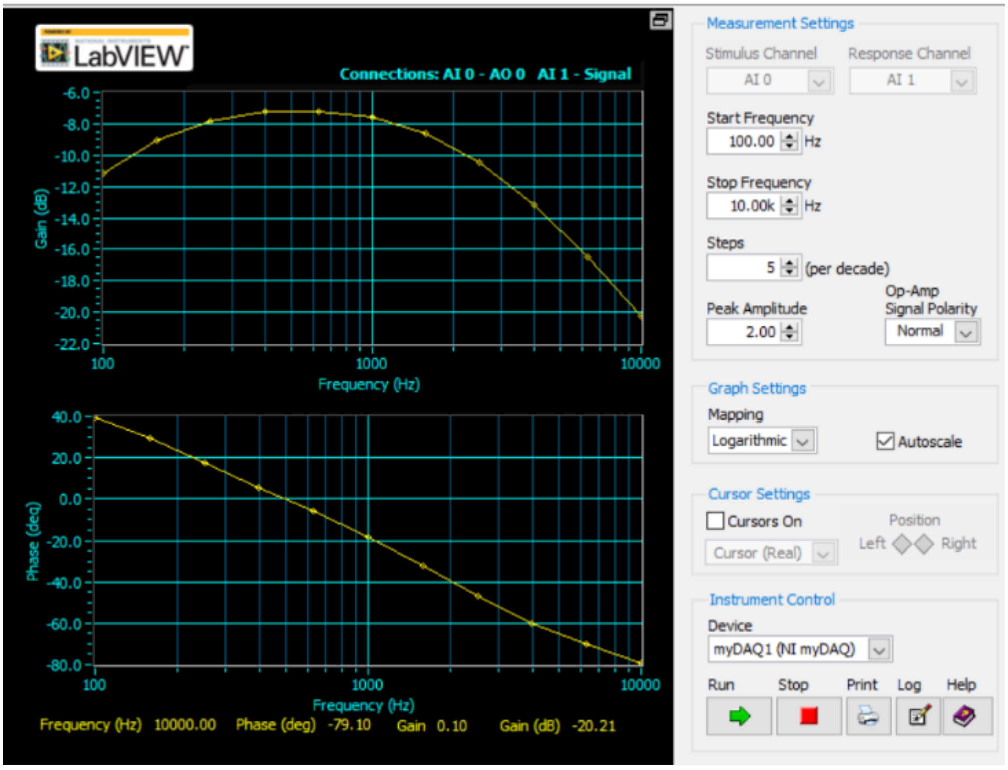

Figure 2: NI ELVIS II Bode Analyzer result

A LabView Virtual Instrument was then created to analyze the frequency spectrum and store the data in a spreadsheet file. Since this project requires the analog output and the analog input to be in sync, the bandpass filter was connected to the NI PCI6024e DAQ card due to its superior support for triggering options over myDAQ. The VI generates a Chirp signal through AO0 starting from 100Hz up to 10kHz, and the signal is sent to the filter input as well as AI6. The filtered signal is read at AI13, and the filter is ground at one of the AIGND pins. Figure 3 below illustrates the output of the VI. The graph on the top displays both filtered and unfiltered waveforms in the time domain, while the middle and the bottom graphs in the image show the magnitude and phase plots in the frequency domain.

A sampling rate of 20 kHz was used as at least two samples per period are necessary at a minimum to alias a signal by the Nyquist frequency criterion. Since the highest frequency that our chirp signal produces is 10kHz, a frequency of 20kHz was used for sampling. We tried higher sampling rates as well, but the results did not seem to vary significantly. Different number of samples were tested, but the plot remains the same throughout, so a total of 2000 number of samples were taken. During testing, it was found that a Rectangle window with no averaging consistently gave us the most desirable plots.

The frequency-domain data was then stored in a spreadsheet file for analysis. The spreadsheet columns are in order: frequency, the magnitude of the unfiltered signal, the magnitude of the filtered signal, phase of the unfiltered signal, and the phase of the filtered signal.

By testing it was found that the best data was achieved by using a signal generation with an amplitude of 2 and an averaging method that included using a Rectangle averaging window and Exponential as the weighting mode.

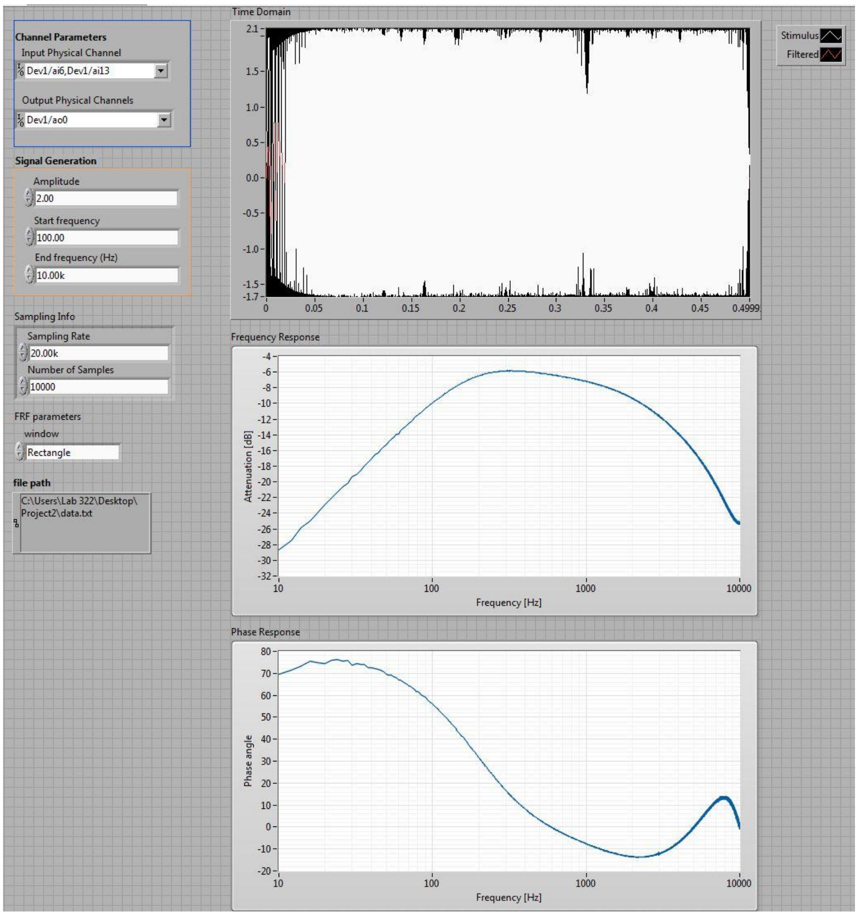

Figure 3: Project 2 VI output

2) Data processing and Model Validation

For data analysis, the recorded response from the VI that was saved in a spreadsheet file was loaded into the MATLAB workspace. This processing was split into three sections. In the first section, an empirical transfer function was derived based on the data using the etfe function, and then the bode function in MATLAB was used to get the frequency response for it. Later in the second section, direct processing of data was done the way it was done in the VI before plotting the frequency plots. And finally, a theoretical transfer function was derived by substituting the values of the resistors and the capacitors in the s-domain dynamic equation of the transfer function. Then data for frequency plot was generated for it using the bode function.

Equation 1 below, is the theoretical transfer function that was derived.

$$Theoretical\ TF = \frac{0.00054\, s}{(8.165e-08)\, s^2 + 0.0006912\, s + 1}$$

Equation 1

4: MATLAB frequency plot below illustrates the comparison between the frequency response plots of the theoretical transfer function, the empirical transfer function, and the actual data. It can be observed that very close to the center frequency all the plots in the magnitude plot have their highest magnitude, and exactly at the center frequency all the plots in the phase plot are at zero degrees. Also, the plots of the empirical transfer function and the experimental data perfectly overlap and closely follow the theoretical frequency response plot.

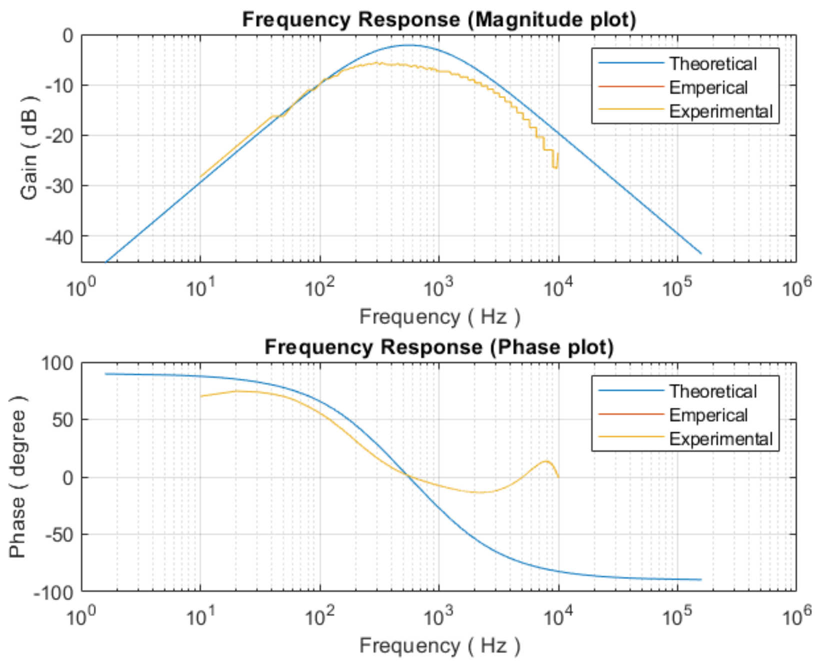

Figure 4: MATLAB frequency plot

From Figure 4, we can see that the empirical and experimental graph overshadow one another. It is interesting to note that the highest magnitude achieved by the theoretical transfer function is up to 4 or 5 dB higher and that its peak is very much parabolic and well defined than that of the empirical or experimental plots. Another thing to note is that although all the three phase plots show zero phase angle at the center frequency and that the theoretical curve closely matches the others for a large set of frequencies, the phase plots of the recorded data begin to deviate from the ideal behavior after the center frequency as illustrated by the theoretical phase plot.

3) MATLAB Code

```
% Project 2
% Sagar Jagtap, Arun Ramachandran

clear, clc;
load('Response_2000.txt');
Response = Response_2000;
Ts = 1/20000;

%% Empirical TF
freq = Response(:, 1);  % Hz
u = Response(:,2).*exp(1i.*Response(:,4));  % Input
y = Response(:,3).*exp(1i.*Response(:,5));  % Output
freq_rad = freq*2.*pi;    % rad/s
data = iddata(y, u, Ts, 'Frequency', freq_rad); % Ts = 0
G_etfe = etfe(data);
[mag_etfe,phase_etfe,w_etfe] = bode(G_etfe);

%% For Experimental plot
mag_ex = Response(:,3)./Response(:,2);  % Magnitude Op/In
phase_in = wrapToPi(Response(:,4));
phase_out = wrapToPi(Response(:,5));
phasedif = phase_out - phase_in;
phase_ex = rad2deg(wrapToPi(phasedif));

%% Theoretical TF
C1 = 27e-9; C2 = 27e-9;
R1 = 20000; R2 = 5600;
G_theoretical = tf([C1*R1  0],[C1*R1*C2*R2  (C1*R1+C2*R2)  1])
[mag_th,phase_th,w_th] = bode(G_theoretical);

%% Frequency Response
subplot(211)
semilogx(w_th/(2*pi),20*log10(squeeze(mag_th)),w_etfe/(2*pi),20*log10(squee
ze(mag_etfe)),freq,20*log10(squeeze(mag_ex)));
title('Frequency Response (Magnitude plot)')
xlabel('Frequency ( Hz )');
ylabel('Gain ( dB )');
legend('Theoretical','Emperical','Experimental');
grid;

subplot(212)
semilogx(w_th/(2*pi),squeeze(phase_th),
w_etfe/(2*pi),squeeze(phase_etfe+360), freq, squeeze(phase_ex));
title('Frequency Response (Phase plot)')
xlabel('Frequency ( Hz )');
ylabel('Phase ( degree )');
legend('Theoretical','Emperical','Experimental');
grid;
```