Roderick Landreth
May 22th, 2019
MER-322: Dynamics of Physical Systems
Professor Ramasubramanian

Modelling Frequency Response of a Lowpass Filter

Introduction

The intent of this exercise is to accurately model the effect of input voltage frequency on a low pass filter's output. This will then be compared to data taken from the circuit itself to confirm its accuracy. The input signal was provided by a function generator, and an oscilloscope was used to measure the amplitude and phase shift of the input and output signals.

Methods

The Low-Pass Filter circuit was constructed on a breadboard with the input being a sinusoidal signal with peak to peak amplitude 7.32 volts, and a frequency that was adjusted between 100Hz and 100kHz. Over this range, data was taken at 35 different frequencies of input peak to peak voltage, output peak to peak voltage, and time shift from the zero point of the input to the zero point of the output. The time shift was then converted into the phase shift between the output and input. The values of resistance in the low pass filter (Figure 1) were confirmed with a multimeter.

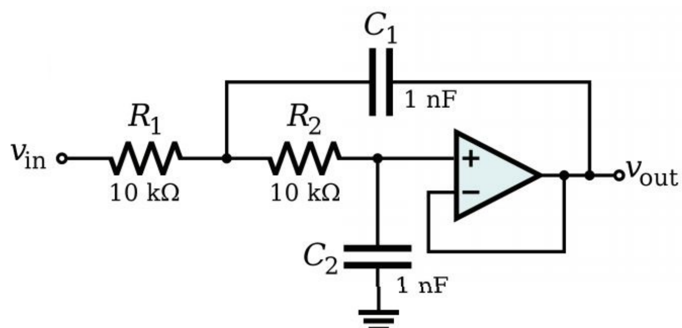

Figure 1: low pass filter, 353 operational amplifier, measured with values $R_1 = 9.85k\Omega$, $R_2 = 9.87k\Omega$

The analytical model for this system's frequency response was based on the transfer function, derived using the impedance method in tandem with nodal analysis on two nodes along the non-inverting input. In this case, the voltage at the non-inverting input should be approximately equal to the inverting input, which should be equal to the output voltage. Taking the Laplace transformation of the resulting equation and evaluating at $s = jw$ produces the sinusoidal transfer function of equation 1. The magnitude of the output sinusoide is directly related to that of the input multiplied by the absolute value of $H(jw)$, and the output phase shift is the complex angle of $H(jw)$, as used in the model.

$$H(jw) = \frac{1}{1 - C^2R^2w^2 + CRwj} \tag{1}$$

Results

Figure 2 shows the agreement between the modeled frequency response of the system and the measured output amplitude and phase shift over 35 different frequencies. As the frequency of the input signal increased, the amplitude of the output signal decreased, and was delayed more, shown by the negative phase shift.

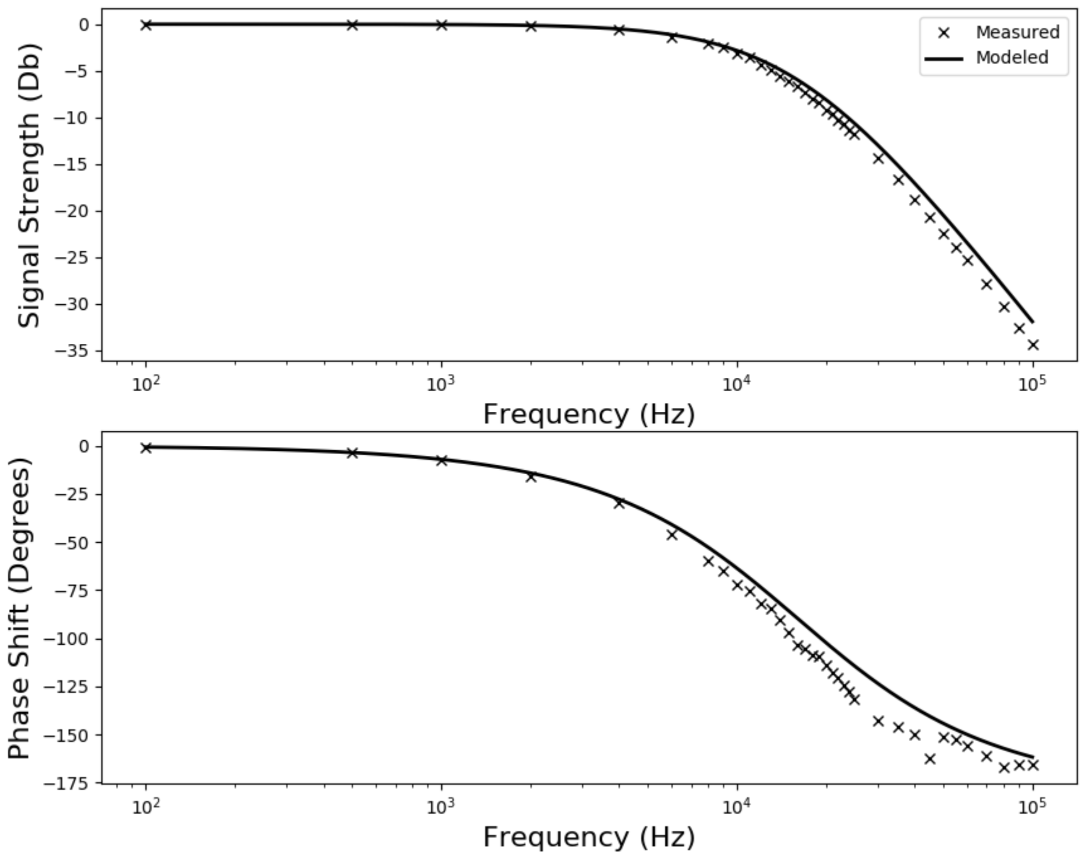

Figure 2: Python Plot of Model Data vs. Measured Data. Top: The Amplitude frequency response shows agreement between the measured data points and the model's analytical approximation. Bottom: The Phase Frequency Response in degrees from the input of the output signal. The increasingly more negative angle indicated the output signal was increasingly more delayed as the frequency increased. Code found in Appendix A.

Discussion

This low lass filter deviates a small amount from the model, potentially caused by any internal impedance not accounted for by the resistors and capacitors. If the resistance used in the transfer function is increased, the model more closely matches the experimental results. This may indicate a bad connection or a small internal impedance within the 353 op-amp. In the phase response the data lines up with the model less because it was more difficult to measure the time shift with smaller values at higher frequencies. Each time the scale on the horizontal axis was decreased, the accuracy of the time value improved. This is displayed in Figure 2 (bottom) as the data dips away from, and then towards the model's approximation between 10^4 and 10^5 Hz.

This model is critically damped, meaning that there are real, equal roots of the polynomial in the transfer function's denominator, and that this circuit configuration has the fastest change of its frequency and phase without oscillation. Within this critically damped system, the cutoff frequency can be altered by changing the resistance or capacitance values. The cutoff frequency is inversely related to the resistance and capacitance. This can be seen in the frequency response (equation 1) as R (or C) increases on the denominator, in order for $|H(jw)|$ to remain the same, w (and therefore f) will have to decrease, shifting the frequency response curve of Figure 2 to the left.

Appendix A: Python Code

```
# Frequency Response Simulation using Python
# Author: Roderick Landreth
# Date: May 22, 2019

import matplotlib.pyplot as plt
from io import BytesIO
from PIL import Image
import math
import cmath

#Constants
radToDeg = 180/(math.pi)
Res=9.86*10**3
Cap=10**-9

""" Plot two subplots each with two datasets"""
def sol_plot(x1,y1,label1,label2,param_dict1,x_axis,y_axis,x2,y2,param_dict2,x21,y21,
            param_dict21,x22,y22,param_dict22,x_axis2,y_axis2):
    fig, ax = plt.subplots(2,1,figsize=(10, 8))

    axis_font = {'size':16}
    ax[0].set_xscale('log')
    ax[0].plot(x1,y1,**param_dict1)
    ax[0].plot(x2,y2,**param_dict2)
    ax[0].set_xlabel("{}".format(x_axis),**axis_font)
    ax[0].set_ylabel("{}".format(y_axis),**axis_font)
    ax[0].legend([label1,label2])

    ax[1].set_xscale('log')
    ax[1].plot(x21, y21, **param_dict21)
    ax[1].plot(x22, y22, **param_dict22)
    ax[1].set_xlabel("{}".format(x_axis2), **axis_font)
    ax[1].set_ylabel("{}".format(y_axis2), **axis_font)

    plt.show()

    png1 = BytesIO()
    fig.savefig(png1, format='png')
    png2 = Image.open(png1)
    png2.save('Frequency Response.tiff')
    png1.close()

# Convert the txt file of measured experimental results from a ascii file to a float array
get_Data = open("Data","r")
expData=[]
for line in get_Data:
    expData.append([float(x.strip()) for x in line.split('\t')])
get_Data.close()
```

```
MeasuredHjw=[-20*math.log10(float(expData[2][x])/expData[3][x]) for x in range(0,35)]
PhaseShift=[-360*expData[0][x]*expData[1][x] for x in range(0,35)]

# Model of System, including 9990 datapoints using the Sinusoidal Transfer Function (Hjw)
modelFreq=range(100,100000,10)
omega=[2*math.pi*x for x in modelFreq]
Hjw=[1/((1-(Cap**2*Res**2*x**2)+Cap*Res*x*complex(0,2))) for x in omega]
MagDec=[20*math.log10(abs(x)) for x in Hjw]
Phase=[cmath.phase(x)*radToDeg for x in Hjw]


sol_plot(expData[1],MeasuredHjw,
         "Measured","Modeled", {'marker':'x','color' : 'black', 'linestyle' : 'none'},
         "Frequency (Hz)","Signal Strength (Db)",
         modelFreq,MagDec,{'color' : 'black', 'linestyle' : '-', 'linewidth':2},
         expData[1],PhaseShift,{'marker':'x','color' : 'black', 'linestyle' : 'none'},
         modelFreq,Phase,{'color' : 'black', 'linestyle' : '-', 'linewidth':2},
         "Frequency (Hz)","Phase Shift (Degrees)")
```