EE 113D: Digital Signal Processing Design

Lab #4
External Input and Voice Transformations

Names: Erik Hodges & Ivan Manan

Due Date: October 31, 2018

Step 1A: Measuring the Frequency Response of the LCDK

Objective

In this section we measured the frequency response of the LCDK's output in order to account for this in our later measurements.

Procedure

We connected the output of the LCDK and the function generator to the oscilloscope and programmed the LCDK to output a waveform identical to the function generator output. We compared the gain of the LCDK's output with respect to the output of the function generator.

Observations

The Bode plot in Figure 1.1 shows that the LCDK's output is significantly attenuated at very low frequencies. This is because of a series capacitor that exists on the output line of the LCDK, which is meant to block DC signals. The output also begins to attenuate at high frequencies because there is an RC filter with a cutoff frequency of 24 kHz that cuts out any frequencies above the Nyquist frequency of the LCDK. This RC filter should have a gain of -1.5 dB at 24 kHz and this seems to agree with the Bode plot below.

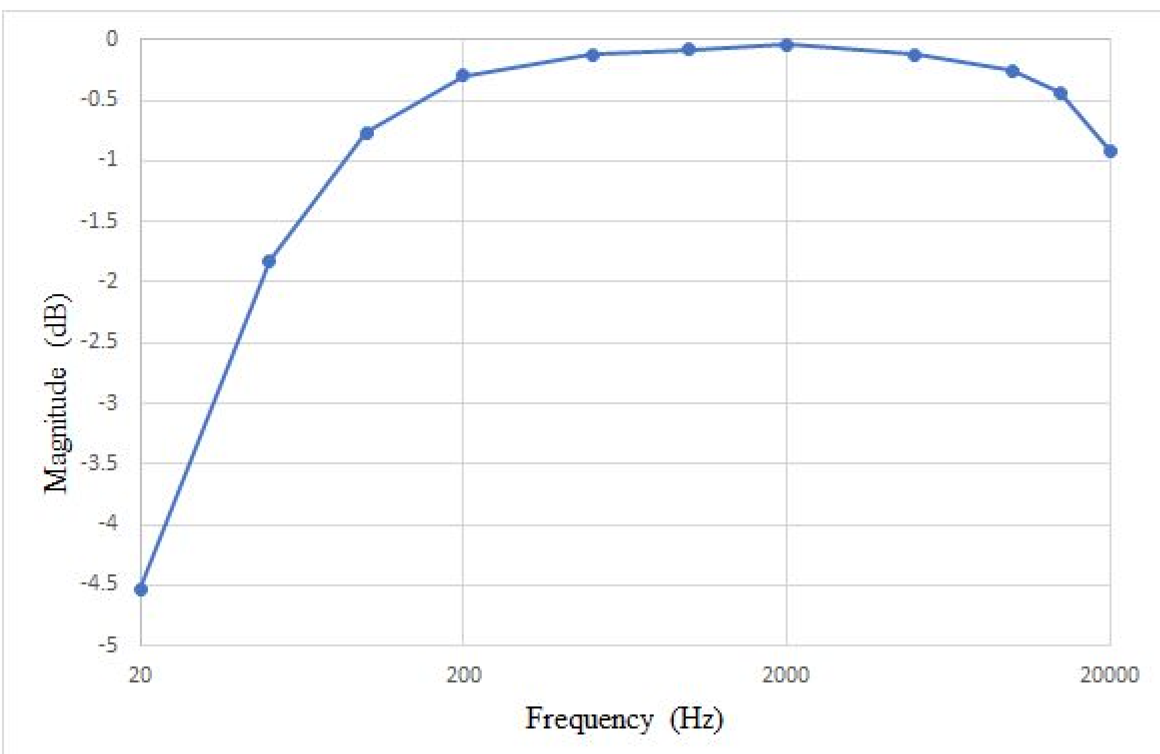

Figure 1.1 Bode Plot for the Frequency Response of the LCDK

Step 1B: Single Pole LPF with Different Cutoffs

Objective

In this section, we used the bilinear transform to approximate first-order RC low pass filters using the LCDK.

Procedure

We used the bilinear transform to obtain a difference equation for first-order RC low pass filters with cutoff frequencies of 1 kHz and 10 kHz. We implemented the difference equations on the LCDK and measured the gain at various frequencies using the oscilloscope and function generator.

Theory

The continuous time transfer function of a first-order RC low pass filter is given by
$H_a(s) = \frac{1}{1+RCs}$.

We can approximate the associated discrete time transfer function by taking the bilinear transform:
$H_d(s) \ = \ H_a(\frac{2}{T}\frac{z-1}{z+1}) = \frac{1+z^{-1}}{(1+2RC/T)+(1-2RC/T)z^{-1}}$.

Since the sample rate is 48 kHz, T = 1/48000.

For simplicity, we define:
$C_1 = 1 + 2RC/T$
$C_2 = 1 - 2RC/T$

We have:
$\frac{Y(z)}{X(z)} = \frac{1+z^{-1}}{C_1+C_2z^{-1}}$

Taking the inverse Z-transform, we obtain:
$y(n) = \frac{1}{C_1}\left[- C_2\, y(n-1) + x(n) + x(n-1)\right]$

We can use this difference equation to implement any arbitrary first-order RC low pass filter on the LCDK.

1 kHz Cutoff Calculations

The cutoff frequency of an RC filter is given by:

$f_c = \frac{1}{2\pi RC}$

For 1 kHz cutoff we have:

$1000 = \frac{1}{2\pi RC} \rightarrow RC = 1.59 \times 10^{-4}$

Using this, we obtain

$C_1 = 16.26$

$C_2 = -14.26$

10 kHz Cutoff Calculations

The cutoff frequency of an RC filter is given by:

$f_c = \frac{1}{2\pi RC}$

For 1 kHz cutoff we have:

$1000 = \frac{1}{2\pi RC} \rightarrow RC = 1.59 \times 10^{-5}$

Using this, we obtain

$C_1 = 2.53$

$C_2 = -0.526$

Observations

We expect to obtain a gain of -1.5 dB at the cutoff frequency. In Table 1.1, we have a gain of -1.495 dB at 1 kHz, which is about -1.5. However, in Table 1.2, we have a gain of -1.827 dB at 10 kHz, which is lower than expected. This is due to frequency warping, which is a property of the bilinear transform that results in the cutoff frequency of the digital filter being lower than the cutoff frequency of the associated analog filter. Frequency warping is the result of the bilinear transform approximating the basis of the Z-transform using the first 2 terms of its Taylor expansion. As a result, the discrete time cutoff frequency becomes a nonlinear function of the continuous time cutoff frequency. Additionally, the error is much larger for high frequencies, which is why the 1 kHz filter has a cutoff frequency at 1 kHz but the 10 kHz filter has a cutoff frequency significantly below 10 kHz (see Figures 1.2 and 1.3). Note that the Bode plots in Figures 1.2 and 1.3 are calibrated based on the frequency response of the LCDK’s output filter. Hence, there is no drop off at low frequencies.

	Test Frequencies for 1 kHz LPF						
Frequency (Hz)	100	200	500	1000	2000	5000	10000
Part 1a Gain (dB)	-0.767	-0.300	-0.126	-0.083	-0.042	-0.126	-0.256
1 kHz LPF V_0/V_i (dB)	-0.767	-0.389	-0.574	-1.58	-3.400	-6.595	-9.420
1 kHz LPF Gain	0.000	-0.089	-0.448	-1.495	-3.358	-6.469	-9.165

Table 1.1 Data Set Displaying the Frequency Response of a 1 kHz Low Pass Filter

	Test Frequencies for 20 kHz LPF							
Frequency (Hz)	200	500	1000	2000	5000	10000	14000	20000
Part 1a Gain (dB)	-0.300	-0.126	-0.084	-0.042	-0.126	-0.256	-0.435	-0.912
20 kHz LPF V_0/V_i (dB)	-0.300	-0.126	-0.126	-0.169	-0.621	-2.083	-3.680	-7.660
20 kHz LPF Gain	0	0	-0.042	-0.127	-0.496	-1.827	-3.245	-6.741

Table 1.2 Data Set Displaying the Frequency Response of a 20 kHz Low Pass Filter

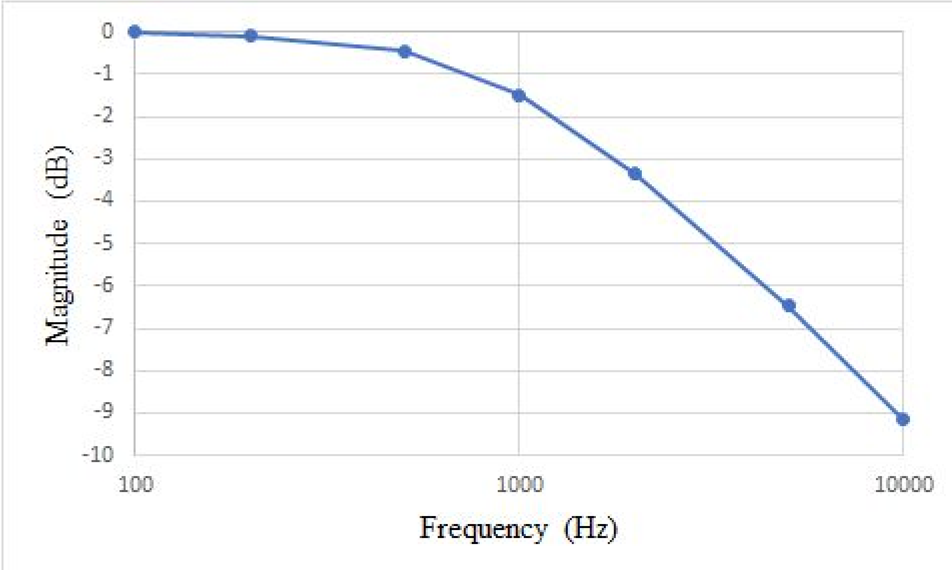

Figure 1.2 Bode Plot for the 1 kHz Low Pass Filter

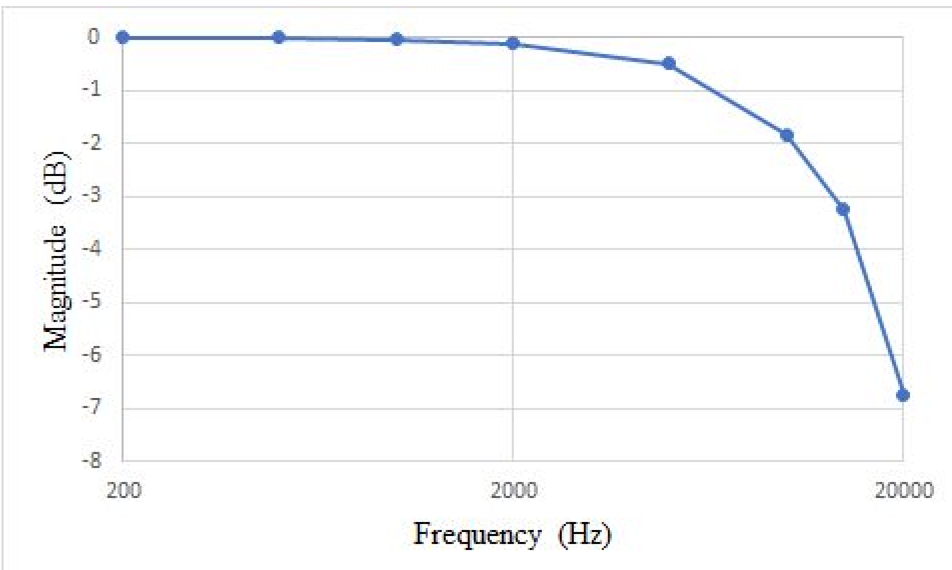

Figure 1.3 Bode Plot for the 20 kHz Low Pass Filter

Code for 1 kHz Cutoff LPF

```
#include "L138_LCDK_aic3106_init.h"
#include "evmomapl138_gpio.h"

float xn = 0;
float xn_1 = 0;
float yn = 0;
float yn_1 = 0;
float sample_out = 0;
```

```
interrupt void interrupt4(void) // interrupt service routine
{
	xn = (float) input_left_sample();
	yn = (14.26 * yn_1 + xn + xn_1) / 16.26;
	yn_1 = yn;
	xn_1 = xn;
	sample_out = (int16_t) yn;
	output_left_sample(sample_out);

	return;
}

int main(void)
{
L138_initialise_intr(FS_48000_HZ,ADC_GAIN_0DB,DAC_ATTEN_0DB,LCDK_LINE_INPUT);
while (1);
}
```

Code for 10 kHz cutoff LPF

```
#include "L138_LCDK_aic3106_init.h"
#include "evmomapl138_gpio.h"

float xn = 0;
float xn_1 = 0;
float yn = 0;
float yn_1 = 0;
float sample_out = 0;

interrupt void interrupt4(void) // interrupt service routine
{
	xn = (float) input_left_sample();
	yn = (0.526 * yn_1 + xn + xn_1) / 2.53;
	yn_1 = yn;
	xn_1 = xn;
	sample_out = (int16_t) yn;
	output_left_sample(sample_out);

	return;
}

int main(void)
{
L138_initialise_intr(FS_48000_HZ,ADC_GAIN_0DB,DAC_ATTEN_0DB,LCDK_LINE_INPUT);
while (1);
}
```

Step 2: IIR LPF

Objective

The purpose of this portion of the lab was to design and analyze a minimum-order Butterworth IIR low-pass filter. This lab also introduced the concept of filter design in Matlab.

Procedure

The Matlab *fdatool* was used to design a minimum-order Butterworth IIR low-pass filter with Fpass of 500 Hz, Fstop of 5 KHz, Apass = 1 dB, and Astop = 80 dB. The sampling frequency was taken at 48 ksps.

The magnitude response of the system was generated below in Figures 2.1. Additionally, Figure 2.2 displays a zoomed-in version of Figure 2.1, but between 0 kHz and 3 kHz. The purpose of having Figure 2.2 display this zoomed-in portion was to display the -3 dB magnitude response.

The *fdatool* also computed the second-order sections and the gains of the system, which are listed in Figure 2.3. This dataset was consequently used to modify the iirsos.c source code that would be exported to the LCDK. The LCDK then outputted a gain value, which was recorded and displayed in Figure 2.4 as a Bode plot. Matlab was used to generate the correct bode plot of the system with the given second-order sections and gains, as demonstrated in Figure 2.5.

Theory

Infinite impulse response filters are distinct from finite impulse response filters in that IIR filters have an impulse response that is non-zero for all time, while FIR filters have a time limited impulse response.

The -3 dB point is known as the half-power point. This measurement is a common benchmark for filters of all types. The -3 dB point indicates that the filter cuts off half of the power at that frequency. The -3 dB magnitude response occurred when the frequency was 0.821 kHz. Hence, the designed Butterworth IIR low-pass filter would cut off half of the power at 0.821 kHz.

Observations

Notice in Figure 2.4, at approximately 821 Hz, there exists a shift into a downward trend. This makes sense because the -3 dB point is the half-power point. Moreover, since the system is a low-pass filter, the gain in Figure 2.4 dropped substantially after 821 Hz.

Figure 2.5 was the expected, Matlab generated bode plot given the second-order section and gain values. At first glance, Figures 2.4 and 2.5 appear differently from each other. However, when zoomed-in to Figure 2.5 between 20 Hz and 20 kHz, the graphs have the same plot characteristics as a low-pass filter. Moreover, the frequency response of the LCDK matches that of the Matlab generated bode plot because there is a substantial drop in gain after 821 Hz. Thus, the Bode plot in Figure 2.4 was expected.

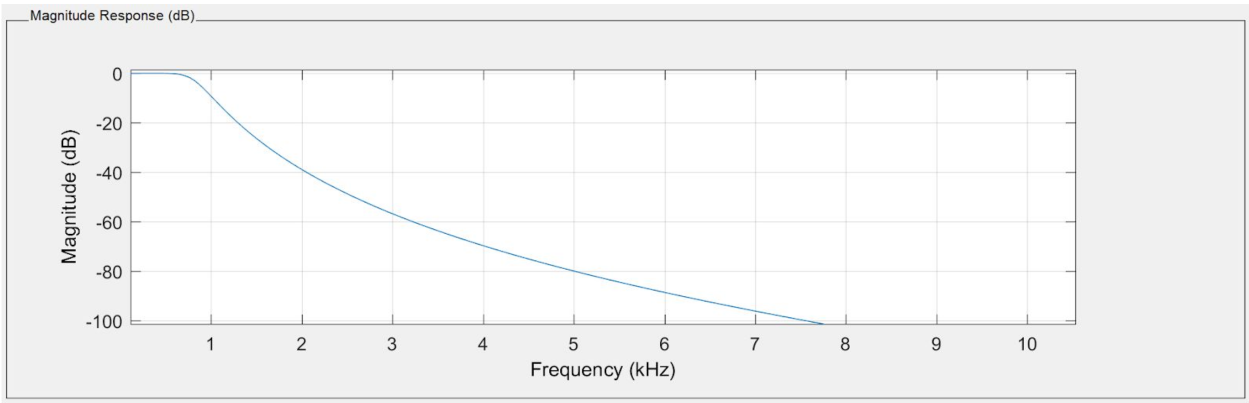

Figure 2.1 Magnitude Response of the Butterworth IIR Low-Pass Filter

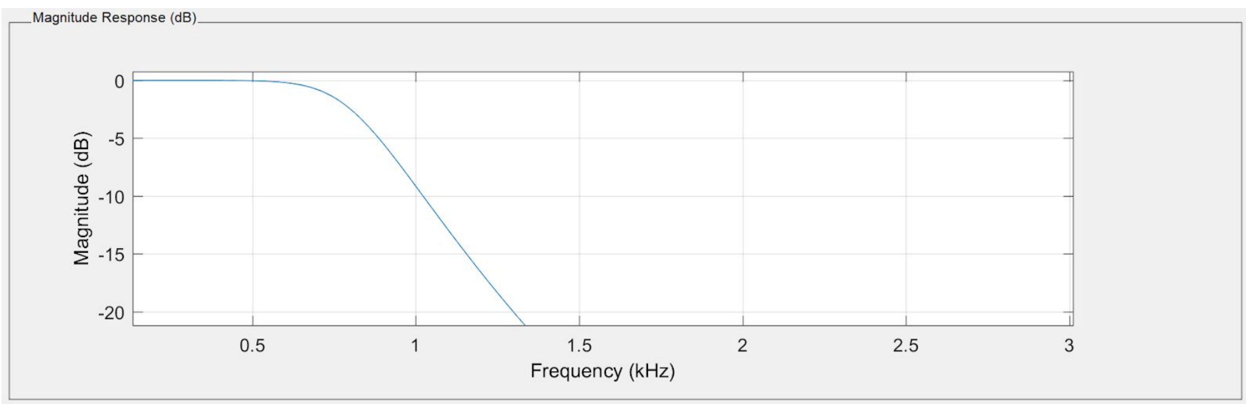

Figure 2.2 Zoomed-In Magnitude Response of the Butterworth IIR Low-Pass Filter

```
G =

    0.0028
    0.0027
    0.0511
    1.0000

>> SOS

SOS =

  Columns 1 through 4

    1.0000    2.0000    1.0000    1.0000
    1.0000    2.0000    1.0000    1.0000
    1.0000    1.0000         0    1.0000

  Columns 5 through 6

   -1.9246    0.9358
   -1.8296    0.8403
   -0.8979         0
```

Figure 2.3 List of Gain and Second-Order Section Entries

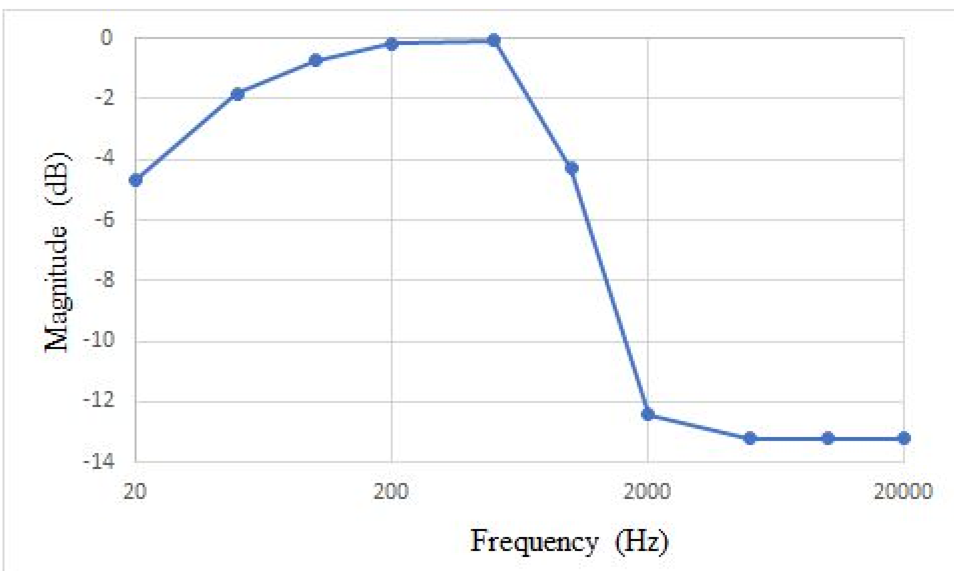

Figure 2.4 Bode Plot Frequency Response of the LCDK

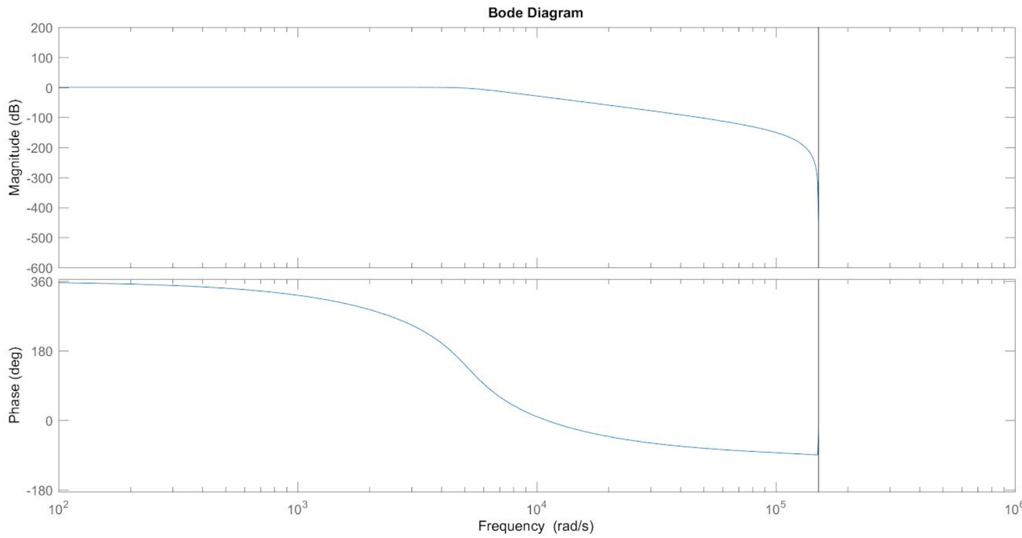

Figure 2.5 Matlab Generated Bode Plot Frequency Response; Notice when you zoom in between 20 Hz to 20 kHz, the bode plots from this Figure as well as Figure 2.4 are equivalent.