Filter Design and Measurement
Lab 09

by Bruce A. Black, Robert Throne, and Mario Simoni

Objectives

In this lab we will examine filters in different ways. We will first investigate three different kinds of lowpass filters and the properties they possess. These filters illustrate the types of trade-offs we must make when designing a system. Finally we will use a filter to measure the relevant "bandwidth" of a signal and look at the effects of filtering on a "real-world" signal.

Background

A periodic signal can be represented by the complex exponential form of the Fourier series. When a periodic signal is applied to the input of a filter, each of the harmonic components of the input signal experiences an amplitude and phase change caused by the filter. At the filter output the harmonic components add together to produce the output waveform. The amplitude and phase changes experienced by each of the input components combine to make the output signal different from the input signal in a predictable way. This difference between the input and output waveform is called distortion.

Stated mathematically, suppose $x(t)$ is a periodic input signal with period T_0, H(ω) is the frequency response of the filter (i.e. the Fourier transform of its input response h(t)), and $y(t)$ is the filter output. Then the input $x(t)$ can be written

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{jk\omega_0 t}, \text{ where } \omega_0 = 2\pi/T_0 .$$

As the input signal passes through the filter, the input coefficients a_k become altered by the filter to become the output coefficients b_k, where

$$b_k = H(k\omega_0)\, a_k .$$

The output $y(t)$ is then given by

$$y(t) = \sum_{k=-\infty}^{\infty} b_k e^{jk\omega_0 t} = \sum_{k=-\infty}^{\infty} H(k\omega_0)\, a_k e^{jk\omega_0 t}$$

In practice there is not usually an infinite number of components to be added to produce either $x(t)$ or $y(t)$; only components with significant amplitudes need to be included.

Pre-Lab Review Lab 6.

Part 1: Filtering Periodic Signals

This is a continuation of the work you did in **Lab 6.** Now we are going to concentrate a bit more on the filters and examine some trade-offs between different common types of lowpass filters. Much of what follows is the same as for **Lab 6**, but there are a few changes at the end when we use different types of filters.

a) Use your code from **Lab 6** to determine the Complex Fourier series representation using 20 terms of the following periodic function (defined over one period).

$$x(t) = \begin{cases} 0 & -2 \le t < -1 \\ 1 & -1 \le t < 2 \\ 3 & 2 \le t < 3 \\ 0 & 3 \le t < 4 \end{cases}$$

b) For the majority of the filters it uses, Matlab assumes filter has the form

$$H(s) = \frac{b_N s^N + b_{N-1} s^{N-1} + \cdots + b_2 s^2 + b_1 s + b_0}{a_N s^N + a_{N-1} s^{N-1} + \cdots + a_2 s^2 + a_1 s + a_0}$$

Hence, in order to represent any filter, Matlab just uses an array for the *b* coefficients and an array for the *a* coefficients. For example, we might have two variables (arrays) B and A to store the coefficients. These variables would be

$$B = \begin{bmatrix} b_N & \ldots & b_1 & b_0 \end{bmatrix}$$
$$A = \begin{bmatrix} a_N & \ldots & a_1 & a_0 \end{bmatrix}$$

Let's assume we want to use a 10^{th} order Butterworth lowpass filter with a frequency cutoff of $20\,\omega_0$. Use the help command to look up the Matlab function **butter.** You should return the coefficients in two arrays, i.e. your command should be [B,A] = butter(…..); Note that we are constructing an **analog** filter here, so read **all** of the description for the **butter** command.

c) We now again need to find the variable H0 = $H(0)$ and the variable (array) $H = \begin{bmatrix} H(j\omega_0) & H(j2\omega_0) & \ldots & H(jN\omega_0) \end{bmatrix}$. To find $H(0)$ we use the fact that

$$H(0) = \frac{b_0}{a_0}$$

If the *b* and *a* coefficients are stored in arrays *B* and *A*, we can write

H0 = B(end)/A(end);

Where **end** tells Matlab to get the last element of the array. The command **freqs** will be helpful for finding the variable (array) H.

d) Plot the Fourier series representation for both the input signal $x(t)$ and the output signal $y(t)$ on the same graph for N=25 terms, *using different line types and a legend.* The title of your graph should indicate that you are using a Butterworth filter. If you have done everything correctly your graph should look like that in Figure 1. *Print out this graph and attach it to the worksheet at the end of this lab.*

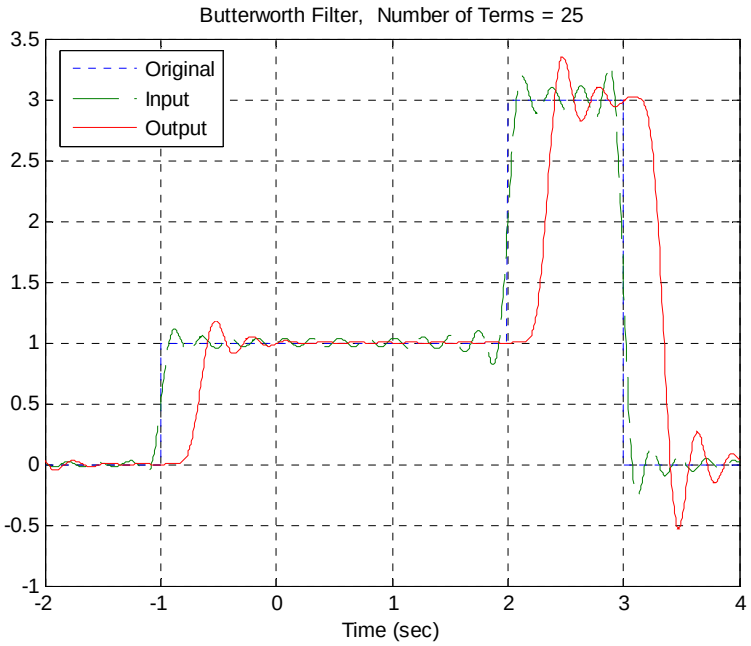

Figure 1. Input and Fourier series representation of the input and output using a 10^{th} order Butterworth filter.

It will be useful to look at a frequency response plot of the filter you are using. You will need to make three plots on one page using the **subplot** command. Type **orient tall** before any of the plotting so you can use more of the page.

e) The first plot is the magnitude of the transfer function as a function of frequency. You will need to use the **abs** command. You should also put a **grid** on your figure.

f) The second plot is the phase of the transfer function (in degrees) as a function of frequency. In order to see the phenomena we want to see use the sequence of commands **unwrap(angle(H))*180/pi**. The most important command here is **unwrap**, which allows angles of more than 180 degrees

g) The third plot is the magnitude of the transfer function (in dB) as a function of frequency. Here you need to use the commands **semilogx** and **log10**.

Be sure to include the type of filter you are using in your title. If you have done everything correctly, you should get a plot like that shown in Figure 2. *Print out this graph and attach it to the worksheet at the end of the lab.*

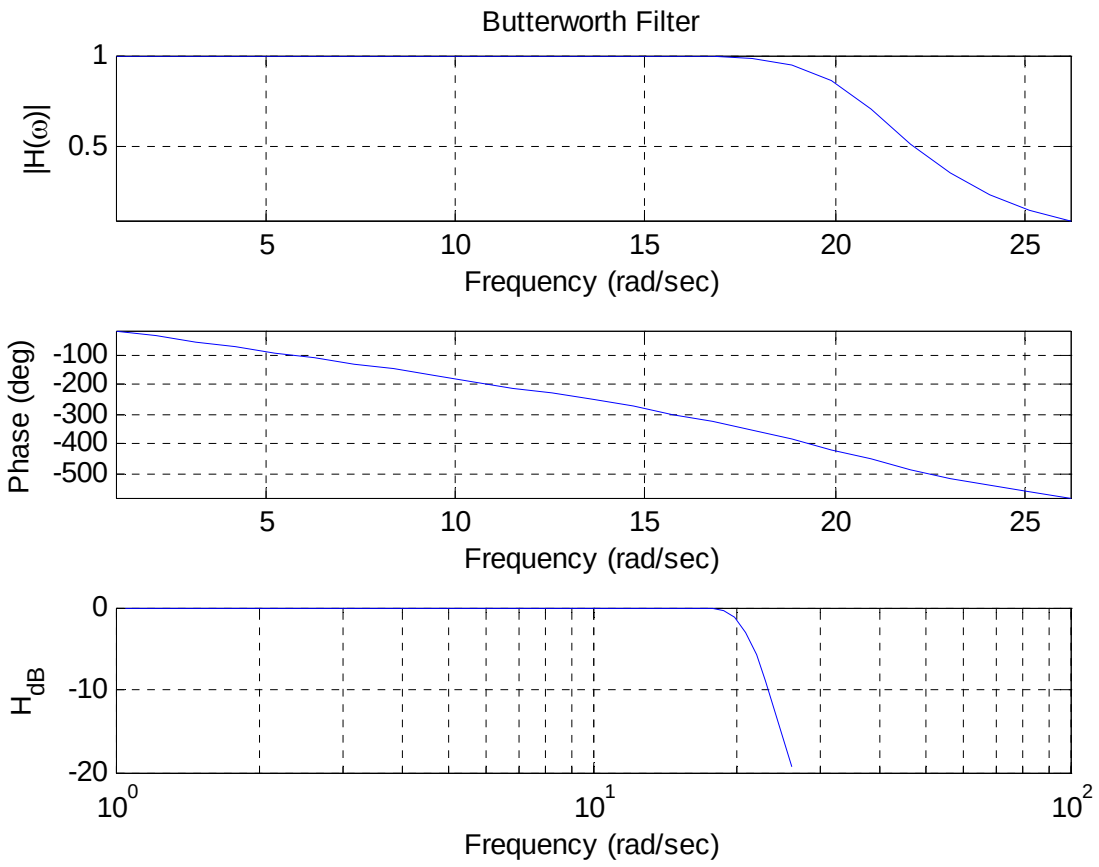

Figure 2. Frequency response of the 10th order Butterworth filter.

h) If the phase of the filter is exactly linear then we should have the relationship

$$-slope\ of\ \ filter\ (in\ \sec) = time\ delay$$

Using a straight edge (or ruler), draw a line on your graph between the initial point on the phase graph and the value of the phase at about $20\,\omega_0$. Estimate the delay between the input and output signal (use Matlab to zoom in on the signal and measure this accurately, don't just eyeball it!) and compare this to the predicted delay. Fill in the table on the worksheet.

i) Repeats parts **b-h** using a Bessel filter (use the command **besself**). Note that for this filter the frequency you specify can be considered the maximum frequency at which the phase will be linear. Determine, by trial and error, the frequency you need to enter in order for the cutoff frequency to be at approximately $20\,\omega_o$. For this use multiples of ω_o, such as $30\,\omega_o$, $40\,\omega_o$, etc. *Be sure to print out both graphs.*

j) Repeat parts **b-h** using a Cheyshev filter (use the command **cheby1** and the defaults if you don't know what values to use). *Be sure to print out both graphs.*

k) In the Chebyshev filter, try varying the R parameter from 0.5 (the default) to 2.0. What happens to the filter?

Part 2: Measuring the Bandwidth of a Signal

Even though the Fourier Transform of a signal can take up the entire spectrum to infinite frequency, you don't necessarily need the entire spectrum to get the relevant "information" from a signal. In this part of the lab, you will be applying filters to the ECE300 welcome message that you heard on the first day of class. The goal will be to find the minimum bandwidth that includes all of the necessary "information", and then relate this information to the power spectrum of the signal.

When MATLAB is used to filter a signal, everything must be done with discrete-time signal processing tools because it is being implemented on a computer. You will learn more about these tools in ECE380. For this lab the signal has been sampled at a sufficiently high frequency to approximate the continuous time equivalent. The only difference as far as you are concerned is how the filter cutoff frequency is defined. Instead of an absolute frequency, it must be normalized to half the sampling frequency. This has already been accounted for in the supplied script file so that you can still specify the absolute frequency in units of Hz.

a) Copy the files `welcomeMsg.mat` and `lab8msg.m` to your working directory for lab8. Look over the script file and make sure you understand what it is doing. ***You will need to add the appropriate axis labels and titles to all of your plots before printing them out.***

b) Start by using a lowpass filter with a cutoff frequency of 50 Hz. Look at the spectrum and the time waveform and listen to the message. *Describe what happened to the signal in the time and frequency domain on the lab worksheet. Also explain why the speaker doesn't produce any sound for the filtered signal.*

c) Increase the cutoff frequency of the LPF in regular increments until you don't hear much difference between the "words" of the original message and the filtered message. For each increment, observe the figures that illustrate the time and frequency representation of the signal. *When you find the final cutoff frequency, record it on the lab worksheet and print out the two figures to turn in with your lab.*

d) Now switch to a highpass filter and start with a cutoff frequency of 4kHz. Look at the spectrum and the time waveform and listen to the message. *Describe what happened to the signal in the time and frequency domain on the lab worksheet. Also describe the sound that you hear from the speaker.*

e) Decrease the cutoff frequency of the highpass filter in regular increments until you don't hear much difference between the "words" of the original and filtered messages. For each increment, observe the figures that illustrate the time and frequency representation of the signal. *When you find the final cutoff frequency, record it on the lab worksheet and print out the two figures to turn in with your lab.*

f) Now make a bandpass filter with the two cutoff frequencies that you found from the highpass and lowpass filters. Zoom in on the x-axis to t=2.4-2.5 seconds for both the original and filtered signals. *Print the figures of the time waveforms and the spectrums and turn them in with your lab.*

Lab 09
Instructor Verification Sheet

Names_____Date:_____

At the end of this lab you should have attached the following plots:

- *Two plots for the Butterworth filter (input/output and transfer function characteristics)*
- *Two plots for the Bessel filter*
- *Two plots for the Chebyshev filter*
- *The six plots from Part 3 (lowpass time/ freq, highpass time/freq, bandpass time/freq)*
- *One plot showing the original and filtered signals for 2.4<t<2.5*
- *Matlab code for parts 2e and 2f*

Part 1: Measurements of Different Filters

Filter Type	Predicted Delay (-slope of filter)	Measured Delay (time domain)
10^{th} order Butterworth		
10^{th} order Bessel		
10^{th} order Chebyshev		

Part 2: Measuring the Bandwidth of a signal

Lowpass filter cutoff frequency	
Highpass filter cutoff frequency	

Question 1: From Part 1, which filter type has the most linear phase over the passband? This filter will have the smallest ***phase distortion***.

Question 2: From Part 1, which filter has the flattest magnitude response over the passband? This filter will have the smallest ***magnitude distortion***.

Question 3: From Part 1, which filter has the steepest rolloff after the cutoff frequency ?

Question 4: From Part 1, what is the effect of varying R in the Chebyshev filter?

Question 5: What happened to the WelcomeMsg signal in the time and frequency domain when the lowpass filter cutoff frequency was 50 Hz?

Question 6: What happened to the WelcomeMsg signal in the time and frequency domain when the highpass filter cutoff frequency was 4 kHz?

Question 7: In terms of signal power, what relationship do you notice between the bandwidth of the bandpass filter and the spectrum of the original WelcomeMsg signal?

Question 8: Looking at the filtered and unfiltered versions of the WelcomeMsg signal between t=2.4-2.5 seconds, what distortion do you notice in the filtered version? Also looking at the unfiltered signal, explain why it is impossible to filter this signal without distortion.