ROSE-HULMAN INSTITUTE OF TECHNOLOGY
Department of Electrical and Computer Engineering

ECE 300 Signals and Systems | Winter 2005
RDT

Filter Design and Measurement
Lab 07

by Bruce A. Black

Objectives

In this project a square wave is to be passed through a simple low-pass filter and the output measured in the laboratory. MATLAB will be used to generate a Fourier series representation of the filter output, and you will be able to see how adding the complex exponential components of the Fourier series produces the actual output waveform.

Equipment

Agilent Function Generator
Digital Oscilloscope
Orange Butterworth Filter
BNC T-connector
50 Ω termination
Floppy disk

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

$$b_k = H\left(k\omega_0\right) a_k .$$

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

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

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

1. The transfer function of a Butterworth filter of "order *n*" has *n* poles (and no zeros) in the left half plane. These poles are equally spaced around the circumference of a circle whose radius is equal to the 3 dB frequency of the filter in rad/s. The Orange Boxes are Butterworth filters of order five. The filters are designed such that the 3 dB frequency is approximately 3 kHz. Using the pole-zero diagram of Fig. 1, calculate the locations of the poles of $H(s)$. Use these to obtain an *analytic* expression for the frequency response $H(\omega)$. You may assume that $H(0) = 1$.

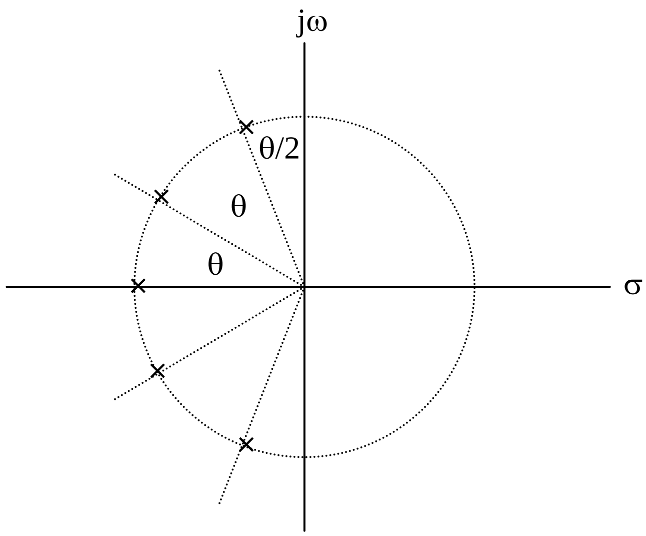

Figure 1: Pole-Zero Diagram of a Butterworth Filter

2. The input $x(t)$ to the filter will be a square wave with a period T_0 = 1 ms, 50 % duty cycle, 0.5 V DC offset, and a peak-to-peak amplitude of 1 V. (That is, the square wave switches between zero and one volt.) Find the coefficients a_k of the Fourier series for the input. Include terms for k running from $k = -9$ to $k = 9$. Using MATLAB, plot the magnitude and phase spectrum for the a_k.

3. Using your frequency response $H(\omega)$, calculate the coefficients b_k of the filter output for $k = -9$ to $k = 9$. Using MATLAB, plot the magnitude and phase spectrum for the b_k.

4. Using MATLAB, generate and plot the output waveform $y(t)$. The most efficient way to do this is to use the functions you developed previously.

5. Compare the input waveform to the output waveform by plotting both on the same graph. Explain the differences between the two waveforms.

Procedure

Filtering for Real

1. Obtain an orange filter from your instructor. Measure and record the actual 3 dB bandwidth of the filter.
2. Use the function generator to generate the waveform described in the prelab. Be sure to properly account for the necessary 50 Ω load.
3. Now insert the Orange Filter between the function generator and the load. Observe the output signal on the digital oscilloscope. Set the oscilloscope so that the display looks as much like your prelab prediction of the output waveform as possible. Now capture the oscilloscope display into a file. Load the file into MATLAB and verify that you can plot it.

Filtering in MATLAB

4. In MATLAB, generate a square wave as described above (Check out the MATLAB command `square`.) Use the time array from your oscilloscope data to determine the sampling interval and the number of points for your square wave.
5. MATLAB can "construct" a fifth-order Butterworth filter with the command

   ```
   [b,a]=butter(5,2*f3dB*deltat);
   ```

   where `f3` is the 3 dB frequency of your filter and `deltat` is the time spacing between points in your input signal `x`. The results *b* and *a* represent the numerator and denominator of the transfer function H(s). Now pass the input `x` through the filter by using the command

   ```
   y=filter(b,a,x);
   ```

6. Plot your theoretical, measured, and simulated output waveforms on the same plot, using the conventions discussed in Lab 1. Discuss any significant deviations.

Report

You should have three different versions of the filter response to a square wave: The Fourier series prediction from the pre-lab, the measured filter output from lab, and the output from the MATLAB filter. Plot all three of these on a single graph. Print it out and tape it in your lab notebook. Comment on the similarity between the three plots. Would the Fourier prediction have approximated the measured output more accurately if you had included more terms in the Fourier series? Be sure that all members of your lab group sign the lab notebook, and hand the notebook in at the end of lab.