# EE432: Digital Signal Processing Fall 2011

# Project 07: Frequency Response $H(\Omega)$

**Assigned: Tues 10/18/11** **Due: Tues 10/25/11**

This project is intended to give you some practice working with filter frequency responses.

## I. MATLAB Tools for Frequency Response

In MATLAB, we normally use the *abs* function to find the magnitude, and the *angle* function to find the phase of a complex function like a filter's frequency response.

1. On a 2x1 subplot, use the *abs* and *angle* functions to plot the magnitude and phase response (in degrees) of a digital filter with frequency response given by:

$$H(\Omega) = \frac{0.85}{1 - 0.15e^{-j\Omega}}.$$

For these plots, let the value of digital frequency $\Omega$ range between $-3\pi$ and $+5\pi$. Use a small enough increment for $\Omega$ to get smooth curves. Check your plots to make sure that:

Magnitude response should be an even function.
Phase response should be an odd function.
DTFT is periodic in digital frequency, with period $2\pi$.

2. Since the magnitude response is an even function, the phase response is an odd function, and the DTFT is periodic with period $2\pi$, we only really need to plot digital frequency from 0 to $\pi$ radians in order to know the DTFT for all values of digital freqency.

Now generate the same two subplots, only now with magnitude in dB ( $G_{dB} = 20\log_{10} G$ ), and phase in radians. For these plots, only determine $H(\Omega)$ for $0 \leq \Omega \leq \pi$.

3. Determine the difference equation for this filter.

4. Now, with the coefficients of the difference equation, use MATLAB's *fvtool* function (short for *filter visualization tool*) to analyze the filter. Type >> help *fvtool* to determine how to use it. The *fvtool* window has buttons that allow you to see the magnitude response (default), the phase response, the pole-zero plot, and can also give implementation information about the filter (e.g., how many multiplies and adds to determine an output value). Answer the following questions :

   a. What is the range of values of digital frequency on the x-axis? Why is it not 0 radians to $\pi$ radians?

   b. What type of filter is this (e.g. HPF, LPF, etc.) ?

   c. MATLAB uses a direct-form two transposed implementation of this filter. In this implementation, how many multiplies are there to compute each output value? How many adds?

d. Turn in the magnitude AND phase plot of this filter from *fvtool*. There is a button in the *fvtool* window that will put them on the same plot.

**II. Transfer Functions**

1. In the frequency domain, a comb filter will pass frequencies that occur at regular intervals. The magnitude response has the appearance of a comb, and the frequencies associated with the spikes are called harmonics. It can be used to remove noise that lies at frequencies between the spikes. A comb filter can be implemented as an FIR filter, or an IIR filter. For an FIR implementation, the general form of the frequency response is:

$$H(\Omega) = 1 - \alpha e^{-jM\Omega}$$

and for the IIR implementation, the general form is:

$$H(\Omega) = \frac{1}{1 - \alpha e^{-jN\Omega}}$$

For each of the following frequency responses of comb filters, output the pole-zero plot and the magnitude response (not in dB) of the filter using *fvtool*. Based on your plots, determine the harmonic digital frequencies for which they are designed.

a. $H(\Omega) = 1 - 0.7e^{-j6\Omega}$.

b. $H(\Omega) = \dfrac{1}{1 - 0.5e^{-j8\Omega}}$

2. The *fvtool* function can also be used to plot the impulse and step responses, and give you the filter coefficients. Use *fvtool* to plot the impulse responses for each of these comb filters.

**III. Filtering**

1. Download the "NoisyDrizzle.wav" file from the course website and listen to it. It is corrupted by some harmonic interference. Filter this signal with a comb filter that has the following frequency response, then listen to the result. Use:

$$H(\Omega) = \frac{1}{1 - 0.5e^{-j11\Omega}}$$

If you play back the filtered signal, you will find that this filter does not do a good job of filtering out the noise. But what is interesting is to see how the shape of the frequency response of the filter shapes the frequency content of the signal. Run *fvtool* with a third input, the signal itself. If *x* is the name of the signal that is input to the filter, then try:

```
>> fvtool(b,a,x)
```

will display the magnitude response of the filter *and* also the frequency content of the signal *x*. If you view the frequency content of the signal, the three annoying tones that I added to the *Drizzle* signal should be apparent. In a later lab, we will use techniques like this to determine how we wish to shape the frequency content of a signal with a frequency filter.

If the filtered signal is called *y*, now run:

```
>> fvtool(b,a,y)
```

And you will see the shape of the frequency spectrum of the filtered signal. Since the process of filtering is done using convolution in the time domain, in the frequency domain we are multiplying the frequency content of the signal by the frequency response of the filter. Note how the shape of the filtered signal's frequency content follows the shape of the filter's magnitude response. Unfortunately, this comb filter was not a good one to handle the annoying tones.

What are the approximate frequencies (in Hz, NOT radians) of the harmonic noise, based on your *fvtool* display?

**Turn in your answers to the questions asked and the plots called for.**