Supplement to Chapter 5 of *The Science of Digital Media* – Digital Audio Processing

Worksheet – Digital Audio Processing> Creating Filters via the Transfer Function[1]

Modeling Environment: MATLAB

Introduction:

An audio filter in the frequency domain can be defined by

$$\boldsymbol{H}(z) = \frac{\boldsymbol{Y}(z)}{\boldsymbol{X}(z)} = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots}{1 + b_1 z^{-1} + b_2 z^{-2} + ..}$$

Equation 1

.

This is called the *transfer function*. If you can put a filter specification into this form, you may be able to manipulate it algebraically to determine the zeros and poles – the places where the function goes to 0 or to a maximum. As $\boldsymbol{Y}(z)$ gets smaller, $\boldsymbol{H}(z)$ gets smaller. The limit leads to a zero of the function. As $\boldsymbol{X}(z)$ get smaller, $\boldsymbol{H}(z)$ gets larger. The limit leads to a pole of the function. Since $\boldsymbol{H}(z)$ is the frequency response, seeing where it gets larger or smaller tells you which frequencies are increased or attenuated by the filter. Let's look at how this works with three simple filters – two comb filters and an all-pass filter.

Suppose you have a delay filter defined by

$$\boldsymbol{y}(n) = \boldsymbol{x}(n) + g\boldsymbol{x}(n-m)$$

The delay in the filter is given by *m*. You can take the z-transform of both sides to get

$$\boldsymbol{Y}(z) = \boldsymbol{X}(z) + g\boldsymbol{X}(z-m)$$

By the delay property of the z-transform, this can be written as

$$\boldsymbol{Y}(z) = \boldsymbol{X}(z) + gz^{-m}\boldsymbol{X}(z)$$

from which we get

$$\boldsymbol{H}(z) = \frac{\boldsymbol{Y}(z)}{\boldsymbol{X}(z)} = 1 + gz^{-m}$$

[1]This material is based on work supported by the National Science Foundation under Grant No. DUE-0340969. This worksheet was written by Jennifer Burg.

Evaluating the z-transform at $z = e^{i\theta}$, where $\theta = \frac{2\pi k}{N}$ for $0 \le k \le N-1$, is in effect taking the discrete Fourier transform of $\boldsymbol{h}(n)$, yielding the frequency response of the filter.

MATLAB has a built-in function called freqz that will give you a frequency response graph if you give it the vectors of coefficients ***a*** and ***b*** from Equation 1.

Let's say we set the delay at $m = 20$ and g= 0.8. Then we have

$$\boldsymbol{H}(z) = 1 + gz^{-m} = \frac{1 + gz^{-m}}{1} = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + ...}{1 + b_1 z^{-1} + b_2 z^{-2} + ..}$$

$$= \frac{1 + 0 * z^{-1} + 0 * z^{-2} + ... + 0.8 * z^{-20}}{1 + 0 * z^{-1} + 0 * z^{-2} + ... + 0 * z^{-20}} =$$

This gives us the coefficients that we need for vectors ***a*** and ***b***. We create these vectors in MATLAB:

```
>>a = [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.8 ]
>>b = [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
```

You get the frequency and phase response diagrams from

```
>>freqz(a,b)
```

You can also use the fvtool, which gives the zero-pole diagram as well.

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

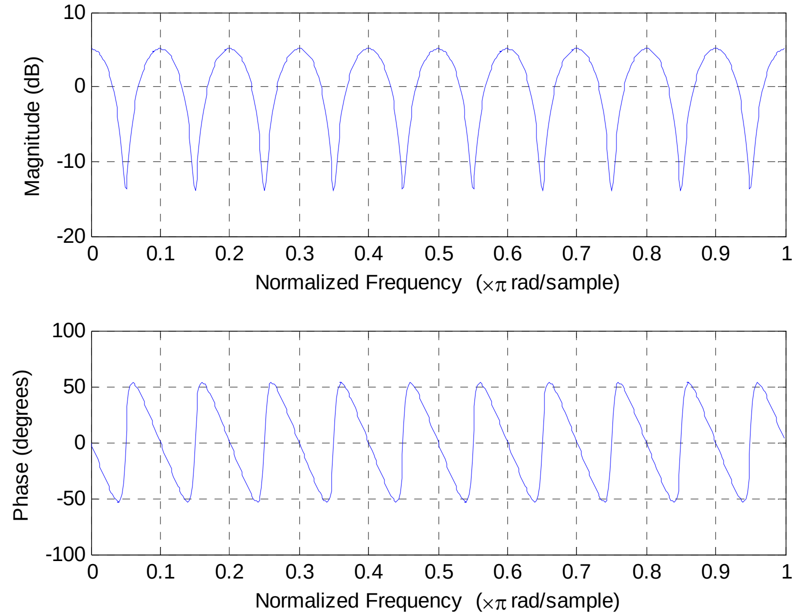

Figure 1. Frequency and phase response of a non-recursive comb filter

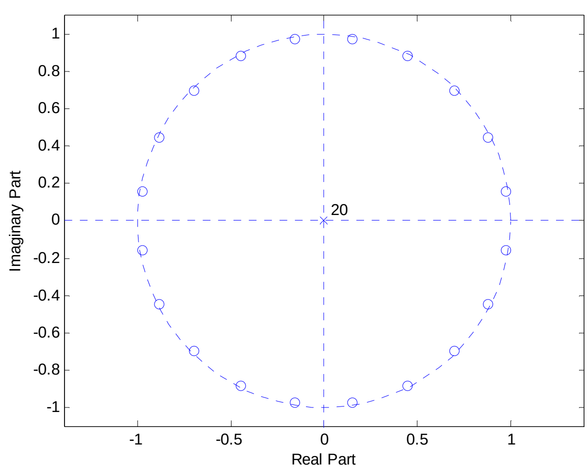

Figure 2. Zero-pole diagram for non-recursive comb filter

Exercise 1

Suppose you have the following recursive filter where m = 20 and g = 0.5:

$$\boldsymbol{y}(n) = \boldsymbol{x}(n) + g\boldsymbol{y}(n-m)$$

What is its transfer function?

Graph the frequency response, the phase response, and the zero-pole diagram in MATLAB, and describe the behavior of the filter.

Exercise 3

Suppose you have the following all-pass filter where m = 20 and g = 0.5:

$$\boldsymbol{y}(n) = -g\boldsymbol{x}(n) + \boldsymbol{x}(n-m) + g\boldsymbol{y}(n-m)$$

What is its transfer function?

Graph the frequency response, the phase response, and the zero-pole diagram in MATLAB, and describe the behavior of the filter.

Note: MATLAB's Filter Design and Analysis Tool allows you to look at the frequency, impulse, and step response of a filter along with its zero-pole diagram. You can manipulate the zeros and poles and see how this changes the behavior of a filter. To open this tool, type

```
>>fdatool
```