BIOL 708L/NACS 728B Lab #3
Fall 2011

The Effect of Simple Filters on Sinusoids

The goal of this lab is to make sinusoids a range of frequencies, and then see what effect various filters have.

1) Create this collection of cosine signals of varying frequency:

```
n=[-7:8].'
N=length(n)
f=[0:8]./N
cossigs=cos(2*pi*n*f)
```

   a) Plot the cosine signals so they are each on individual subplots using:

```
for m=1:length(f); subplot(9,1,m);bar(n,cossigs(:,m)); axis([n(1) n(end) -2 2]); end
```

   b) Why is this range of f a "natural" range to examine? Why not end at N–1 instead? What do the cosine signals for f outside of this range, but less than N, look like?

2) The most straightforward way to use a convolutive system/filter in Matlab is with the `filter` function. Its usage, for general input `x` and general output `y`, is:

```
y = filter(h, 1, x);
```

where `h` is the impulse response of the specific system/filter and the "`1`" will be explained in a future lab.

Use the moving difference system $y[n] = Diff\{x[n]\}$, which is defined to be $y[n] = x[n] - x[n-1]$. Therefore we can see that it has an impulse response $h_{Diff}[n]$ = […,0,0,1,–1,0,0,…,], or, $h_{Diff}[n] = \delta[n] - \delta[n-1]$. In Matlab we represent it in an intermediate notation:

```
hDiff = [1 ; -1]
```

i.e., it is a column whose elements are the coefficients of the impulse response beginning at $n = 0$.

   a) Enter that definition into Matlab. Apply it to the cosine signals:

```
y=filter(hdiff,1, cossigs)
```

   b) Is this filter causal? How do you know?

   c) Plot the outputs on individual subplots, as in (1a).

   d) How does amplitude of the responses' oscillations change with frequency (ignore the first few points: the byproduct of initial condition assumptions made by Matlab)? Does this system allow low frequency inputs to pass through? High frequency? Can you call it a simple low or high pass filter?

3) Repeat (2), but instead of the edge detector, use impulse response of the 2-point-moving-average convolutive filter using: `htwo = [1 1].'/2`

4) Repeat (2), but instead of the edge detector, use impulse response of the 3-point-moving-average convolutive filter using: `hthree = [1 1 1].'/3`