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

1) (*Do not use Matlab for this question*.) Consider the convolutive system/filter

$y[n] = \frac{1}{4}x[n] - \frac{1}{2}x[n-1] + \frac{1}{4}x[n-2]$.

a) Calculate the system's impulse response $h[n]$, and sketch it (e.g. with a bar plot).

b) Calculate $y[n]$ for the case $x[n] = \delta[n]$, and sketch $x[n]$ and $y[n]$ (e.g. with a bar plot).

c) Calculate $y[n]$ for $x[n] = \delta[n] + \delta[n-1] + \delta[n-2]$, and sketch $x[n]$ and $y[n]$ (e.g. with a bar plot).

d) Calculate $y[n]$ for the case $x[n] = \begin{cases} 1 & n \text{ even} \\ 0 & n \text{ odd} \end{cases}$, and sketch $x[n]$ and $y[n]$ (e.g. with a bar plot). How often does $x[n]$ repeat? How often does $y[n]$ repeat? What each of their average values? The average value of a signal is also called its zero frequency (Fourier) component.

e) Is this filter causal? Why or why not?

f) Does this filter show any indications of being low-pass or smoothing? Why or why not?

g) Does this filter show any indications of being an edge detector? Why or why not?

2) Using Matlab, create a pure tone sine wave of frequency 250 Hz, duration 0.5 s, with a compact disc's sampling frequency of 44,100 Hz (sample duration T_s is the reciprocal of the sampling frequency):

```
f = 250 % Hz
T = 0.5 %s
sf = 44100 % Hz
Ts = 1/sf % s
n = [1:T*sf]; % from here on in, the semicolons are really important, since...
N = length(n) %    ...this should be large
t = n*Ts;
x = sin(2*pi*f*t);
```

a) Using headphones, or at least better speakers than the typical laptop, play the signal as a sound:
`soundsc(x, sf)`

b) Save the sound as a .wav file: `wavwrite(x, sf, 'my250HzWavFile.wav')`

c) Find the file you just created and play it by double clicking its icon. Verify that it sounds the same as when you played in Matlab.

d) Read the same sound you just created back into Matlab, and play it again:

```
[y,sfy] = wavread('my250HzWavFile.wav');
soundsc(y,sfy)
```

e) Repeat (a) - (d) for a 500 Hz tone (and change the name of the file accordingly).

3) Continuing in Matlab, generate a signal which is approximately Gaussian white noise, of the same duration and sampling frequency as in problem (3) above, using (the built-in function) `randn()`:

```
x = randn(size(t)); % randn generates random numbers with a Gaussian distribution
```

a) Using headphones, or at least better speakers than the typical laptop, play the sound in Matlab using `soundsc()`.

b) What is the maximum of `x`? What is the minimum of `x`?

c) To save a sound as a .wav file, the signal must be scaled to so that its maximum is less than or equal to 1 and its minimum is greater than or equal to –1. We can do this like so:

```
wavExtrema = [max(x) -min(x)]
wavScale = max(wavExtrema)
xScaled = x/wavScale;
```

Why does this work? Why do we need *both* `max(x)` and `min(x)`? Why is it `-min(x)`?

d) Save the scaled sound as `noise.wav`. Find the file you just created and play it by double clicking it. Make sure it sounds the same as in (a).

e) Make a new sound, of Diff{x[n]},

```
xDiff = diff(x);
```

and play it using `soundsc()`(). Does it sound different than the original? How so? Describing how sounds differ is difficult, so use whatever words work for you. (For sounds, Diff is an example of a "pre-emphasis" filter.) Save the *scaled* sound as `diffnoise.wav` (i.e. as above, between -1 and 1).

f) Make a new sound of a double application of Diff{}:

```
xDiffDiff = diff(diff(x));
```

and play it using `soundsc()`. Does it sound different than the other two? How so? Save the *scaled* sound as `diffdiffnoise.wav`

Email the five .wav files (2b, 2e, 3d, 3e, 3f) to `jzsimon@gmail.com`.