

ECE 5510 Fall 2006: Extra Credit Assignment

Due: Tuesday, Dec 5. Hand in written work in class; email me (npatwari@ece.utah.edu) the Matlab code and .wav file.

In this individual assignment, you will design, analyze, and produce a so-called “white noise” sound track for a CD. This is an individual assignment (strictly no group work) and will demonstrate that you can apply knowledge of LTI filters and power-spectral density to a practical audio processing problem. Up to 15 points will be added to your final exam grade (out of 100). BUT you may only earn a total of 100 points on the final exam. Treat it like a take-home exam question and do not consult with other students. You may, if you wish, consult with me during office hours (or make an appointment).

Background: People sometimes find white noise soothing. Sometimes white noise is used to cover background noise. It has been used to mask *tinnitus*, ringing of the ears. Babies sometimes fall asleep faster and sleep better in the presence of “white noise”. TVs tuned to channels that don’t exist are sometimes used to generate such noise. CDs of white noise are also commercially available. However, the background noise used for this purpose isn’t usually white, *i.e.*, flat in the spectral domain. I downloaded a clip of white noise from the web site, <http://www.purewhitenoise.com>. I loaded it into Matlab and calculated the frequency spectrum with the following statements:

```
q = wavread('pwn_wav.wav');  
plot((1:length(q))./length(q) - 0.5, fftshift(abs(fft(q))));
```

The result was the plot in Figure 1. It may be that purely white (frequency-flat) noise is not actually pleasing to the human ear, even though other random sounds are. Your ‘white noise’

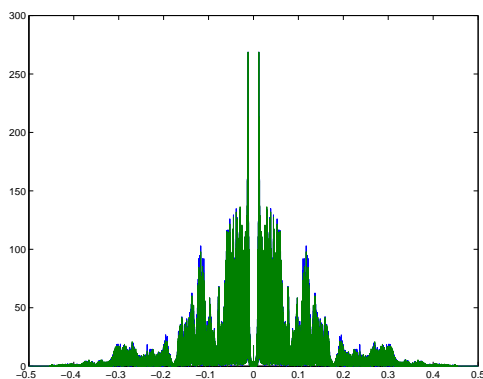


Figure 1: The FFT of the “Pure White Noise” wav clip.

audio track will be produced in Matlab to your own specifications, and your own subjective preferences. You may produce a sound that is low-pass, high-pass, band-pass, multi-band, or stop-band, or any combination of those, as you wish, so long as you believe that it produces a soothing sound. Here are the guidelines for your sound:

1. It must be the output of a linear, time-invariant filter;
2. The input to this LTI filter must be truly ‘white’, *i.e.*, be generated from an i.i.d. random sequence. This input i.i.d sequence may be Gaussian or not; depending on your preference.

3. Matlab has ± 1 thresholds for sound, and when the limits are exceeded, you can hear it, and it is not particularly soothing. So your sound track should mostly stay between -1 and +1.
4. In order to keep the volume of the tracks produced by different students the same, try to keep the standard deviation of the filter output around 0.1. (This is flexible, if you need to reduce the amplitude to stay within the -1 to +1 limits.)

You will be awarded points for the following mix of analysis and experimental verification:

1. (2 pts) Expressing (analytically) the impulse response of the discrete-time filter which you have chosen to implement in Matlab.
2. (3 pts) Expressing analytically the autocorrelation function of your chosen filter, along with verification in Matlab.
3. (3 pts) Expressing analytically the power-spectral density (PSD) of the output of your filter, along with verification in Matlab.
4. (2 pts) Expressing analytically the probability that the output exceeds Matlab's ± 1 threshold for sound, and verification in Matlab.

And the following Matlab experimental results:

1. (3 pts) Your matlab implementation, which generates three minutes of your white noise process, and then saves it to a .wav file.
2. (2 pts) My subjective rating of how 'soothing' the sound is.

You may wish to do the analysis & verification parts (above) with fewer than 3 minutes of data to save time; but the final .wav output should be a 3 minute file. Finally, please use headphones when playing sound in a public computer lab.

You may find the following matlab commands useful:

```

randn(number_of_samples,2)          % use two columns for stereo sound
filter(b,a,x) % Filter the input x
std(y)          % Calculate standard deviation
sound(y, Fs)   % Listen to vector y at sampling rate Fs
autocorr(Series , nLags);          % Compute autocorrelation
length(find(abs(y)>=1))/length(y) % fraction of output < -1 or > +1
wavwrite(y, Fs, 8, 'yourName.wav') % 8-bit .wav file creation

```

The above plot command may also be useful. Please use 'help <function>' to get more information on these Matlab functions.

I will produce and distribute all results to students in the class unless you state that you don't want your track to be distributed (maybe you already have a record deal?).