EE218 LINEAR SYSTEMS II LABORATORY
Missouri University of Science & Technology

LAB 3: SPECTRUM OF NOISY SIGNALS

Aim: In this lab you will learn about noise signals and analyze their spectrum.

Required Background:

- Spectrum of signals: Fourier transform (FT), discrete-time Fourier transform (DTFT).
- Sampling theorem: Nyquist rate.

Introduction: So far in this lab you have been using periodic signals. These signals produce very nice looking displays on test equipment, but they are not typical of the signals you see in the real world. Most signals jump around in seemingly random ways. In this lab, we will start to look at random signals, and how you could observe them on the test equipment. Disconnect the scope probe from an oscilloscope and turn the vertical gain up all the way. On most scopes, you will not see a flat line -you will see a very erratic waveform. This waveform is usually called noise, or in this case measurement noise. You can hear noise if you tune a radio or television to a channel where there are no transmitters, and turn the volume up all the way.

There are many different types of noise. One of the most fundamental is called white noise. There are a variety of ways to generate this noise. You will find in the lab a few noise generators. Some of the spectrum analyzers have a white noise output -often located on the back panel of the equipment. If you can't find either of these, you can make white noise using Matlab. Use the following code to make 1 second of white noise, using a sampling rate of 44,100 samples per second:

```
fs=44100; t=[0:1/fs:1];
whitenoise=randn(size(t));
sound(whitenoise,fs);
```

You can play this noise through the sound card of the PC, store it in a .wav file for playing at a later time, or even burn it into a CD so you can listen to it in the comfort of your own home.

You may wonder why this particular type of noise is called white. The name comes from the optics world. White light contains all frequencies at equal brightness. Likewise, white noise contains all frequencies with equal amplitude. In your linear systems class you learned that the Dirac Delta function contained all frequencies, and you could test filters by measuring their impulse response. This is great for analysis, but it doesn't work out very well in practice. As you will see in a future lab, it is often easier to use white noise to measure the frequency response of a filter.

Suppose you are given a noise source, and you would like to figure out if the noise is white or not. An oscilloscope is not going to help you much. You need to look at the power spectrum of the signal, and see if it contains all frequencies with equal power.

A true Fourier Transform requires you observe a signal from time = -infinity to + infinity. We obviously can't do this in the lab, so the spectrum analyzer (or the FFT in Matlab) will look at a segment, or window, of the input data, and estimate the Fourier Transform from that. After the spectrum analyzer has calculated the Fourier Transform for one window, it will grab the next window and do it again, and again, and again. You will see these spectra as successive traces on the spectrum analyzer display. In the case of periodic signals every sweep looked about the same. This is no longer the case. Any individual sweep/FFT calculation will look noisy (surprise, surprise). The subsequent sweep may look quite a bit different from the current one. To get a clear view of a noise process, you need to average a number of sweeps together. After you average enough sweeps, you should see a smooth line that, in the case of white noise, will be constant for all frequencies.

If you wish, you can listen to the white noise on the speakers, it should sound like the 's' sound. A modern digital phone will have a device that analyzes your voice. When you make an 's' sound, it will not transmit the sound to the other end. Rather, it will send a special code to tell the phone at the other end of the line that you are generating white noise. It will tell the other phone the amplitude of your voice and the duration of the 's' sound. The distant phone will then turn on a white noise source, and send that to the speaker. You might think this is silly, but it takes fewer bits of information to say 'turn on your local white noise source' than it does to digitize and transmit the white noise you are producing when you say a word like 'hiss'.

If you are using a high quality audio amplifier and speaker system, be careful. If you turn the volume up very high, you may blow out your tweeters. High quality speakers usually have a combination of low-frequency speakers (sub-woofers and woofers), middle-frequency speakers (mid-range speakers) and high-frequency speakers (tweeters). The woofers produce sound roughly in the range of 20 Hz to a few hundred Hz. Mid-ranges go from a few hundred Hz to a few kHz. Tweeters go from a few kHz up to a few 10's of kHz. So tweeters cover a rage of frequencies about 10 times as large as the mid-range speakers, about 100 times as large as the woofers. This means that when you listen to white noise you will be dumping 10 times as much power into your tweeters as you are your mid-range speakers. If you turn the volume up high, you may put so much power in the tweeters that you burn them up, even though the sound doesn't seem deafening to your ears. People that test stereo equipment often want noise that delivers roughly the same amount of power to the woofers, mid-range and tweeters. Pink noise does just that -it has a power level that is inversely proportional to frequency. It sounds more like a "sh" sound.

Procedure:

1. Connect a white noise source to a spectrum analyzer. Figure out how to turn the averaging on, and determine how many sweeps you have to average together to clearly see the white noise power spectrum is constant for all frequencies.

2. Use the PC sound card to generate white noise. Analyze the signal using a spectrum analyzer. The sound card output is not called "true" white noise, it is "band limited" white noise. Over what range of frequencies does the white noise look like it has constant power?

3. Either pass white noise into the PC sound card, or generate white noise using Matlab. FFT a window of the white noise and plot the power spectrum. You should notice that it does not look very flat. Write a Matlab script to average a number of these power spectra together. How many do you have to average to get a clear picture of the white noise power spectrum?

4. When you fed periodic signals to the spectrum analyzer and PC sound card, you had to be careful not to clip the signal. Do you still have to worry about that?

For Your Report:

- Part 1: Write down data points from the noise spectrum so you can plot the power spectrum and show that it is (or is not) flat.

- Part 2: Write down data points of the band-limited noise so you can plot it's power spectrum. Make sure to give a thorough explanation of why the noise is band limited.

- Part 3: After you've written the Matlab script, use the script to average 1, 5, 10, and 50 power spectra together. Plot all 4 averaged power spectra on the SAME figure to show that as you average more, the power spectra start to look "flat". Make sure to use different plot colors and the Matlab "legend" function to label the plots correctly.

- Part 4: If you are having trouble answering this question, try simulating "clipping" in Matlab.
