Introduction to the Network Analyzer and the Digital Fast Fourier Transform (FFT)

Eric D. Black
California Institute of Technology

April 13, 2017

1 Prelab exercises

1. Read Arfken Section 14.6 [1]. Explain how the number of operations in an ordinary, discrete Fourier transform scales as N^2, where N is the number of sample points. Do Exercises 14.6.4 and 14.6.5.

2. Obtain a copy of the original paper on the Fast Fourier Transform by Cooley and Tukey [2], and look it over. You can look it up online or ask your TA for a copy. This paper is notorious for being impenetrable, and for this class you only need to get the general gist of it. If you have the interest and the time to go through it completely you may, but it is not required to do the lab. The main things you need to take away from this paper are that the matrix elements in the discrete Fourier transform are not all independent, and that by suitable choice of the number of sample points, you can reduce the total number of operations required to calculate the Fourier transform from N^2 to $N \log N$. If you want to understand the algorithm in all its glory, the paper by Bergland will be of considerable help [3].

2 Laboratory exercises

1. **First, get out the manual!** In research you are not going to have carefully-written instructions for your labs, as you may be used to from

your classes, so you need to get comfortable looking things up in the manual. Now is a good time to start. There is a paper copy of the manual in the lab, next to the instrument, and an electronic version is also available. Contact your TA if you want the PDF version.

2. **Basic operation of the instrument:** Use a function generator to generate a 10 kHz sine wave with a 1 V amplitude. Feed this signal into Channel A of the spectrum analyzer, display it on the screen, and verify that you see a peak where you expect. Practice adjusting the vertical and horizontal scales.

   You will want to set the instrument to measure the spectrum of Channel A and to display that spectrum on the screen. This will involve a fair amount of configuration before you get your first spectrum, unless someone else has left the instrument in the correct state from its las session. The manual may be useful here.

3. **Built-in signal generator:** The SR770 contains a signal generator that can produce a wide variety of signals. Using a BNC Tee, feed the output of this internal signal generator into both the input of the Spectrum Analyzer and an oscilloscope. Experiment with the different modes, and verify that you can produce a sine signal with a frequency and amplitude you set. Try 2-Tone, Noise, and Chirp too. We'll use Noise in the next part of this lab and Chirp in a few weeks.

4. **Frequency resolution:** Most spectrum analyzers produce a spectrum by measuring the *power* in a series of frequency intervals called *bands*. The voltage displayed on the screen is the square root of this power, since power is proportional to V^2. If you vary the width of each frequency band, then, you should see the total power in each band change in a predictable way, *i.e.* proportional to the square root of the width of the band. This bandwidth is sometimes called the *linewidth*, and the SR770 lets you control it from the front panel.

   Set the spectrum analyzer to measure voltage (as opposed to PSD units, which we'll get to in a few minutes), in either mV, dbV, or dbVrms, then change the linewidth, and note how this affects the trace on the screen. The amplitude of the peaks on the screen should change as you change the bin width, and this is because the instrument is graphing the total *power* in each bin, or small frequency band. The wider the bin, the more frequencies the instrument is summing over, and therefore the more power it will report in each one.

Configure the source to generate white noise with an amplitude of 1,000 mV, and connect this output to the input of the spectrum analyzer. Measure the noise level, including an estimate of its uncertainty, for a variety of linewidths, and plot your data. Does the noise level scale with the square root of the linewidth, as expected?

Hint: You can use averaging to reduce your error bars substantially. Just be sure to hit the START button after each linewidth change. That will erase the data from the previous measurement and reduce the amount of time it takes for the trace to settle to its final value for the new one.

5. **PSD units:** Most instruments that calculate FFT's of a signal have an option to normalize the displayed trace by the bin width. The result is called a *Power Spectral Density*, or PSD for short. Enabling PSD units produces a trace on the screen whose amplitude is independent of the frequency bin width. Find this option, enable it, and re-do the measurements you made in the last section. Record your results, noting the new units of *Volts per root Hertz*. Is the amplitude of this spectrum independent of bin width, as expected?

6. **Windowing:** If you took ph3 you may remember covering the subject of *windowing* in your Mathematica homework. If not you can look it up in any good book on digital signal processing, such as Porat [5], or do a Google search on the subject. Most spectrum analyzers offer you a choice of several windows, and the SR770 is no exception.

   To see the effect of different windows on your spectrum, set the source to generate a 10kHz sine wave with an amplitude of 1,000 mV. Set your measurement center frequency also at 10 kHz, so that the resulting line will be centered on the screen, and set the span to a relatively narrow 12.2 Hz. Once the measurement has settled down switch between different windows, and see how each affects the shape of the line on the spectrum.

   Repeat the previous experiment with a much narrower span of 1.5 Hz. This will give you a linewidth of 3.81 mHz (Yes, that's *milli*-Hertz), which will result in a much longer settling time, nearly five minutes.

The results will be different from what you obtained with the larger linewidth, but not by as much as you might expect. Can you explain why?

What is the relation between linewidth and acquisition time, and why?

7. **Input range and noise floor:** At its heart, the FFT spectrum analyzer is just an Analog to Digital Converter, or ADC, connected to a computer. The ADC samples an analog signal you feed into the BNC connection on the front panel and stores that sampled (digitized) signal as a set of 1024 data points in memory. The computer then performs the FFT and any other mathematical operations on that data set and then displays the result on the screen. There are basically three stages between your analog signal and what you see on the screen. Starting at the BNC input on the front panel, first there is an anti-aliasing filter to remove any components of the input signal that are at too high a frequency to be accurately sampled by the Analog to Digital Converter later in the chain. Next comes a *preamplifier*, which can boost the strength of a weak signal to something the ADC can accurately sample. The gain of this preamplifier is adjustable, and the SR770 can monitor the strength of the signal and adjust this gain as appropriate. You can do it manually, too, by setting the *input range* to the maximum value you expect the signal to attain. If you get it wrong, and the signal is larger than you expected, don't worry. You probably won't break the spectrum analyzer. It will just show an "OverLoad" error message on the screen.

   Input range is usually denoted in units of dbV, or decibels relative to one volt.

   Disconnect your signal from the input of the SR770, and put a 50Ω terminator on the input. A 50Ω resistor should produce a broadband noise signal with an amplitude on the order of one nano-volt per root Hertz. This is called *Johnson noise* and is a fundamental property of all resistors, regardless of their construction. (In general, the power spectral density of Johnson noise in a resistor is given by

$$V_{rms,Johnson} = \sqrt{4k_BTR}$$

   where k_B is Boltzmann's constant, T is the temperature in Kelvins, and R is the resistance in Ohms.) See if you can measure this Johnson noise, and check if it agrees with the theoretical prediction. Try this measurement for several different input ranges, and plot the noise you measure as a function of the input range. You should find that, at higher input ranges, the spectrum analyzer introduces its own electronic noise, obscuring the Johnson noise of your resistive terminator. This is important to know about and to characterize if you are trying to measure small signals with your spectrum analyzer.

8. **Saving data:** When it was first introduced the SR770 had a floppy drive for saving data and/or screenshots. Our unit is part of a special manufacturing run in which the floppy drive was replaced by a USB port, but the procedures for saving data or a screenshot to a file on both media types are the same.

   For this part of the lab, generate a simple spectrum on the screen with a peak near the center of your range. Save this spectrum to a USB drive both as a data file and a graphics file. Transfer both to a computer, and open them. Use your favorite graphing program to plot the spectrum from your data file. Print out both your graph and your screenshot, and tape them into your lab notebook.

   If you have trouble using your own thumb drive there is a USB drive in the lab that was supplied by SRS with a proprietary format that may work better.

References

[1] George Arfken, *Mathematical Methods for Physicists, Third Edition*, Academic Press, Inc., (1985).

[2] James W. Cooley and John W. Tukey, *An Algorithm for the Machine Calculation of Complex Fourier Series*, Math. Comp. **19** 297-301 (1965).

[3] G. D. Bergland, *A guided tour of the fast Fourier transform*, IEEE Spectrum **6**, Issue 7, 41-52 (July 1969).

[4] *Model SR770 FFT Network Analyzer*, Stanford Research Systems, Sunnyvale, California (1992). Available online at http://www.thinksrs.com/downloads/PDFs/Manuals/SR770m.pdf.

[5] Boaz Porat, *A Course In Digital Signal Processing*, John Wiley & Sons, Inc. (1997).