Frequency Domain Analysis using Pulses

The following will give background on pulse analysis of Linear Time Invariant (LTI) systems. Then some motivation for why we want to perform pulse analysis is given. Once we are motivated to actually do pulse analysis we will dive into the subtleties and real-life applications of pulse-analysis. Finally, a project will be described which will take you through the process of designing a filter and analyzing it using pulses.

1 Pulse Analysis of LTI Systems - Ideal

Pulse analysis is the process of using a pulse function as the input to an LTI system to determine its magnitude impulse response. For example, if we build a filter and wish to determine the frequencies for which it passes we could simply send a pulse through the system and take the FFT; but more on this later. For now let's look at the theory behind pulse analysis. We know from Linear Signals and Systems that the output of a filter is given by the ~~the~~ time-domain signal convolved with the time-domain transfer function. This idea is shown as a graphical system in Figure 1.

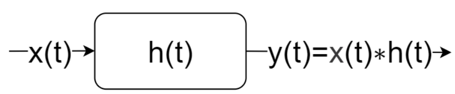

Figure 1: LTI System in the time-domain. The output is defined by the convolution of the input signal and the transfer function/impulse response of the system.

From Linear Signals and Systems we also know that convolution in the time-domain is equivalent to multiplication in the frequency-domain. That is to say that we take the fourier transform of the input signal and the impulse response and multiply them together to get the resulting signal representation in the frequency domain. This idea is shown as a graphical system in Figure 2.

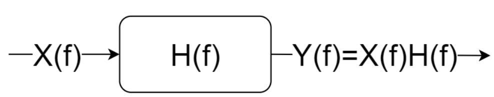

Figure 2: LTI System in the frequency-domain. The output is defined by the multiplication of the Fourier Transforms of the input signal and the transfer function/impulse response of the system.

Typically, we must perform an Inverse Fourier Transform (IFT) of the output $Y(f)$ in order to obtain the information we want. For example, if we want to know what a signal looks like after passing it through a low-pass filter we begin by taking the Fourier Transform (FT) of the signal and the FT of the filter's impulse response. Then we multiply the resulting FTs together and take the IFT of the result to obtain the filtered signal. However, some things are best left in the frequency domain. For example, the low-pass filter itself is best represented in the frequency domain as we can see the gain that the filter applies to various frequencies and we can easily see the frequencies that the filter won't pass.

Let's have a look at the Dirac-Delta function, i.e. a pulse. The dirac-delta function is defined as having a height of 1 at a specific time or frequency and 0 everywhere else. In mathematical notation the Dirac-Delta function is given by Equation 1.

$$\delta(t) = \begin{cases} 1, t = 0 \\ 0, t \neq 0 \end{cases} \tag{1}$$

The FT of the Direc-Delta function is a 1 for all frequencies and can be represented mathematically as Equation 2

$$\delta(t) \xrightarrow{\mathscr{F}} 1 \tag{2}$$

Fourier Transforms come in pairs though, so if we have a pulse in the frequency-domain the equivalent IFT is a 1 for all of time. Mathematically this is represented by Equation 3 and Equation 4 respectively.

$$\delta(f) = \begin{cases} 1, f = 0 \\ 0, f \neq 0 \end{cases} \tag{3}$$

$$1 \xleftarrow{\mathscr{F}^{-1}} \delta(f) \tag{4}$$

Let's examine the dirac-delta function in regards to the LTI system shown in Figure 1. Let $x(t) = \delta(t)$. As we know from Linear Signals and Systems, the convolution of a function with an impulse is just the function. This concept is shown in Figure 3

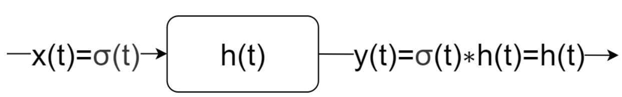

Figure 3: Dirac-Delta as the input to our LTI system. The output is simply the transfer function/impulse response of the system.

We can also represent Figure 3 in the frequency-domain. We just take the FT of the dirac delta function as our input. This concept is shown in Figure 4

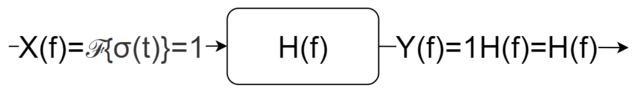

Figure 4: Dirac-Delta as the input to our LTI system represented in the frequency-domain. The output is simply the Fourier Transform of the transfer function/impulse response of the system.

What are the consequences of Figure 4? Well if we have a filter with an unknown shape, or even a known shape that we wish to verify, we see that we can obtain the frequency response by inputing an impulse into our system. This is why the transfer function is so cleverly called the impulse response! Let's move on to motivating ourselves about why we would want to use pulse analysis by looking at what the alternative is.

2 Motivation

It is a common procedure to find the frequency response of a circuit. Let's say we designed a lowpass filter circuit. The frequency response of this theoretical lowpass circuit is shown in Figure 5. As you can see this lowpass circuit only treats frequencies between 0-1500Hz. This frequency response plot contains a lot of great information but after we design a filter circuit in real-life we aren't just magically given a frequency response plot (at least not yet, that's the point of this project) that tells us how our filter acts.

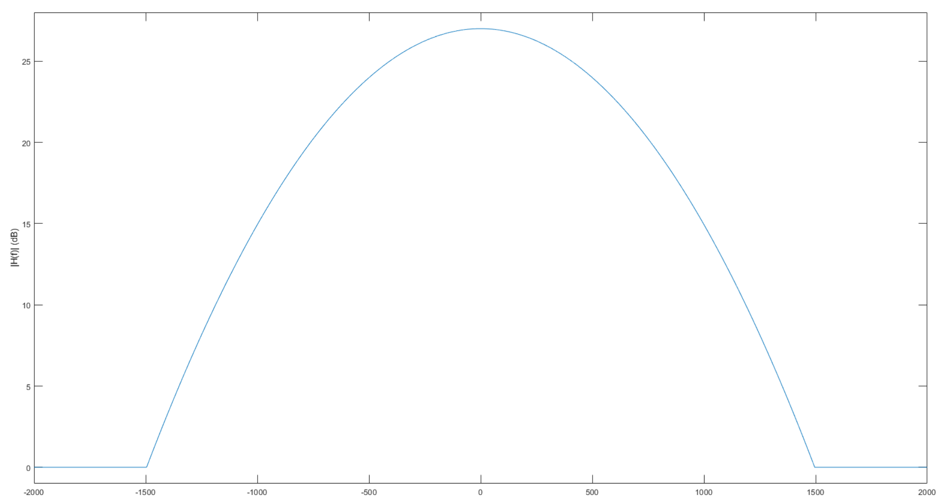

Figure 5: The frequency response of a theoretical lowpass circuit.

In order to generate a plot similar to the one in Figure 5 we must connect our circuit to a function generator and an oscilloscope. We set the function generator to output a DC 1V signal. We adjust the amplitude knobs on the oscilloscope until we can see the output and it tells us that the output of our circuit is 22.3V. We calculate the dB ratio of the output to the input using $20log10(22.3/1) = 27$dB. This gives us the point for $f = 0$ on the plot in **??**. Next we set the function generator to give us the a 1Vpp sinusoid at 100 Hz. We twist some more knobs on the oscilloscope till our signal fits on the window and we have the oscilloscope give us the peak-peak measurement of the voltage; The oscilloscope tells us that the output is 5.6Vpp. We calculate the dB ratio again $20log10(5.6/1) = 14.9363$dB. We have now obtained a second point on our plot for $f = 100$Hz. We repeat this process 18 more times to obtain the points spaced by 100 Hz all the way up to 2000 Hz. We have now spent 15 minutes of our time and obtained the plot shown in Figure 6. It's not bad, but what if I ask you to do this out to 4000 Hz or even worse 20000 Hz; it would take forever and your frequency response plot is still missing chunks and barely covers the audible frequency range!

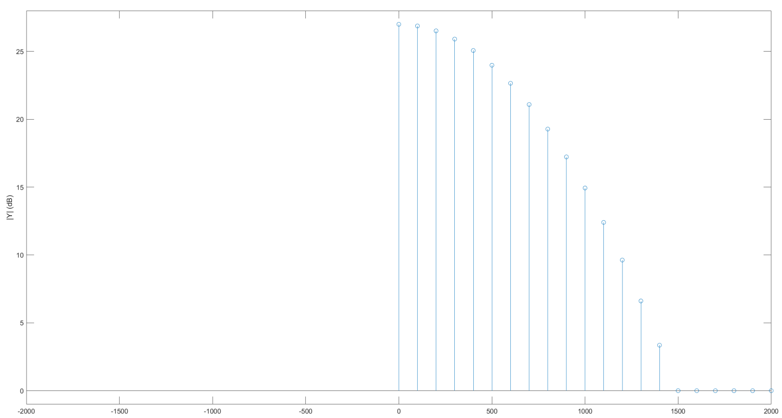

Figure 6: The frequency response plot that we get for 15 minutes of our time.

This large amount of time mindlessly collecting data points for an incomplete graph is exactly what we are trying to avoid doing by using pulses to analyze our filters. As we saw earlier in Section 1: Pulse Analysis of LTI Systems - Ideal, if we put a dirac-delta pulse into our lowpass circuit then the output of the circuit will be the frequency response $H(f)$ that we desire. To obtain this frequency response visually on the oscilloscope we simply ask it kindly to take the Fast Fourier Transform (FFT) for us. For discrete signals there is a version of the Fourier Transform called the Discrete Fourier Transform (DFT). The FFT is an improvement of the DFT for 2 based number systems (i.e. binary). The implementation details of the DFT and FFT are not important to us now as you'll learn about them in DSP, but it is good to know what's going on inside the oscilloscope. Which brings me to another point, the oscilloscopes we use are DIGITAL oscilloscopes. They captures signals using an ADC, meaning the signals are discrete and not continuous! Only the older cathode ray tube oscilloscopes show continuous signals and so they are known as analog oscilloscopes. But now that we have some motivation in mind let's look at some issues with pulse analysis that come from the fact that reailty is not ideal.

3 Pulse Analysis of LTI Systems - Reality

The first issue and probably the only real issue that we will come across when implementing pulse analysis in reality rather than in theory is that the Dirac-Delta function does not and cannot exist. The Fourier Transform of the Dirac-Delta function tells us as much; it is composed of every frequency from 0 to infinity. To put this in perspective the highest frequency a person has ever created is 10^{27} Hz i.e. there is a very very large gap of frequencies that sits between 10^{27} Hz and infinity that we cannot make. Therefore, we must choose a different pulse. One that we can create in real life. The only requirement of the pulse that we create is that its frequency response is flat (i.e. constant, like Dirac-Delta) over the frequency range which we want to investigate. One of the pulses that we can create and that we will use for this project is a rectangular pulse (although we still can't even create a perfect rectangular pulse but we can get close approximations, that's for a different day/project though). For our theories we will use a rectangular pulse that is centered at $t = 0$ and has a width of T but for the project you can use a square wave with a very small duty cycle and a slow frequency (i.e. a pulse train). The pulse trains (i.e. square wave with low duty cycle) can be crafted from a variety of sources such as a microcontroller or a 555 timers. If you don't know how to craft a pulse train with a 555 timer then check out the Art of Electronics. It is a superb textbook that not only does a great job of explaining various circuits but is also has a whole libraries worth of them. Anyways, let's study up on a single rectangular pulse. Figure 7 shows the anatomy of the rectangular pulse which we will examine and Equation 5 gives the mathematical representation of our rectangular pulse.

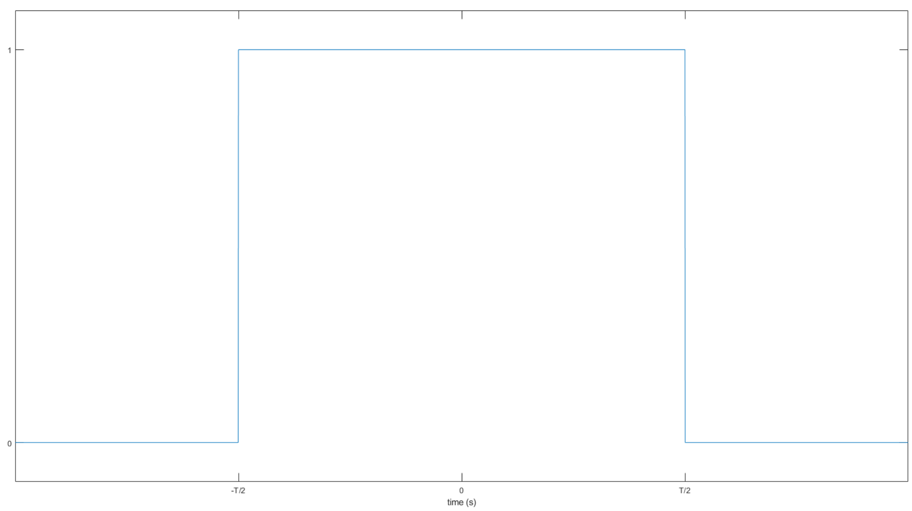

Figure 7: A rectangular pulse of width L.

$$rect_T(t) = \begin{cases} 1, -T/2 < t < T/2 \\ 0, \text{any other time} \end{cases} \tag{5}$$

Let's begin by finding the Fourier Transform of our rectangular pulse. The form of the Fourier Transform that we are interested in is given by Equation 6 because it gives us the FT in terms of the frequency, f, instead of the angular frequency, ω.

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt \tag{6}$$

Evaluating Equation 6 with $x(t) = rect_T(t)$ yields

$$RECT_T(f) = \int_{-T/2}^{T/2} e^{-j2\pi ft}dt$$

$$RECT_T(f) = -\frac{e^{-j2\pi ft}}{j2\pi f}\Big|_{t=-T/2}^{t=T/2}$$

$$RECT_T(f) = \frac{e^{j2\pi fT/2} - e^{-j2\pi fT/2}}{j2\pi f}$$

$$RECT_T(f) = \frac{sin\left(2\pi f\frac{T}{2}\right)}{\pi f}$$

$$RECT_T(f) = \frac{2\frac{T}{2}sin\left(2\pi f\frac{T}{2}\right)}{2\pi f\frac{T}{2}}$$

$$RECT_T(f) = Tsinc(2\pi f\frac{T}{2}) = Tsinc(\pi fT) \tag{7}$$

So the Fourier Transform of our rectangular pulse is simply a sinc function parameterized by the width of our rectangle. Let's see what the Fourier Transform looks like for various widths of the rectangle. Figure 8 shows rectangular pulses with $T = 0.1$, $T = 1$, and $T = 40$ respectively. Under each rectangular pulse is the corresponding Fourier Transform. There are two things to learn from Figure 8. Notice that as the rectangle gets wider the Fourier Transform gets skinnier and vice versa as the rectangle gets skinnier the Fourier Transform gets wider. If we allowed the rectangle on the left to become skinny enough so that it's width was almost 0 then the Fourier Transform would almost constant; this is because a rectangle of virtually no width is equivalent to a Dirac-Delta function and so it's Fourier Transform is a constant. Now as we move to the right in Figure 8 the rectangle is getting wider and the Fourier Transform is approaching a pulse. If we allowed the rectangle to have infinite length so that it is a constant for all of time then the Fourier Transform would become the Dirac-Delta function as expected. Those are the two things to learn. A rectangle of infinitesmal width is equivalent to a Dirac-Delta function and a rectangle of infinite width is equivalent to a constant; meaning that Fourier Pairs for the Dirac-Deltas apply to the edge cases of the rectangular function. For this reason we choose the rectangular pulse to replace our Dirac-Delta pulse.

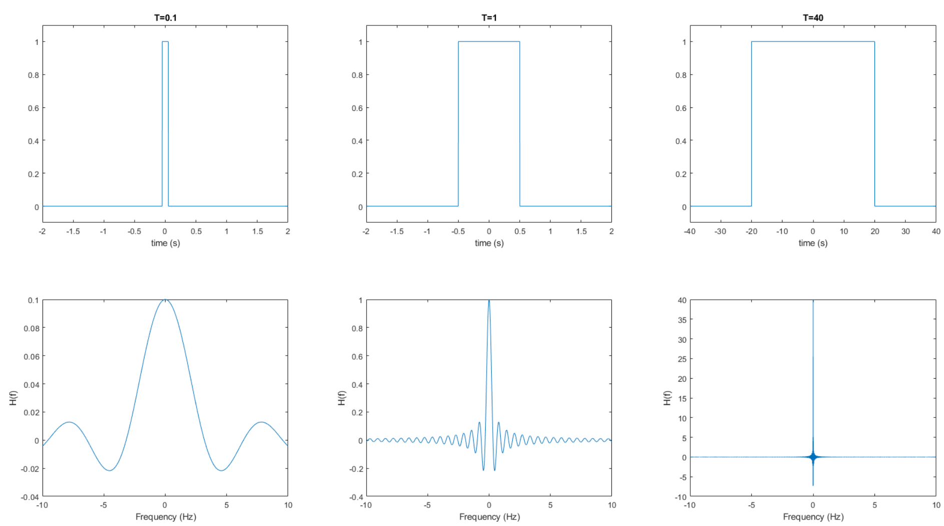

Figure 8: Variously lengthed rectangular pulses and their Fourier Transforms.

To be particular, the feature of the rectangular pulse that we will exploit is the feature where a skinny enough rectangle has a Fourier Transform that starts to flatten over the frequency-domain. This feature is exactly our requirement for pulse analysis; i.e. the Fourier Transform of the chosen pulse signal must be constant over the frequency range of interest. Now let's redo the capturing of the frequency response of our low pass filter.

4 Pulse Analysis Walkthrough

We want to repeat the experiment in Section 2: Motivation but let's require that we find the frequency response over a frequency range of 20000 Hz. We could do it by hand by twisting knobs on the function generator and oscilloscope for an hour or we could just send a pulse in and take an FFT. Let's do the latter. First we need to find a pulse skinny enough so that the magnitude of the Fourier Transform is flat over the 20kHz range. Let's take the definition of "flat" to mean that there is a -3dB drop or less across the 20kHz frequency range. Well what about a pulse of 1 millisecond (i.e. $L = 0.001$)? This should be easy enough to craft with a microcontroller or a 555 timer. Figure 9 shows the magnitude frequency response (i.e. the magnitude of the Fourier Transform converted to dB). The plot on the left shows the magnitude frequency response over a 500kHz range. The plot on the right shows the magnitude frequency response over a 40kHz range and the y-axis is scaled to show the -3dB range better. The red line shows the -3dB down range that we need to be flat over. The plot on the right shows us that for a 1 millisecond rectangular pulse the magnitude frequency response immediately drops far below the -3dB line. So 1 millisecond won't work.

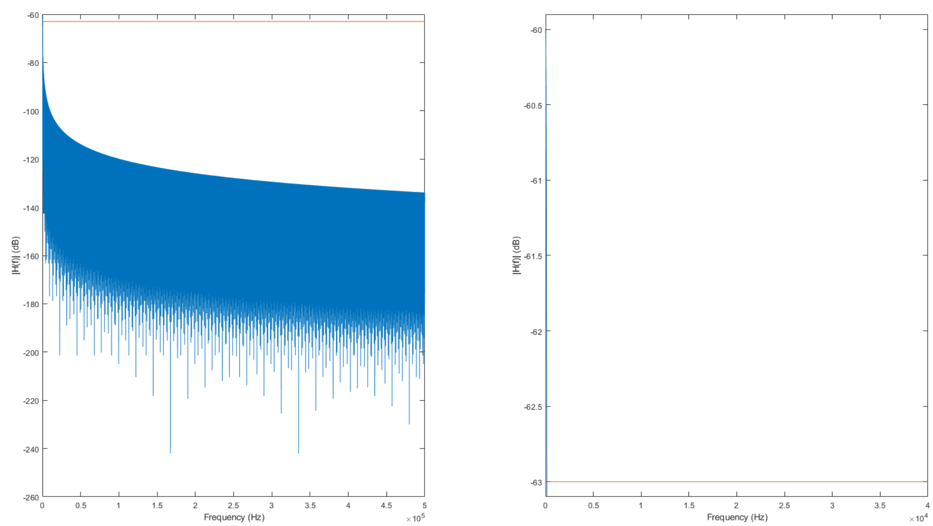

Figure 9: Magnitude Frequency Response of a Rect Pulse with T=1ms (Blue) and the -3dB down line (Red). Left: 500kHz range Right: 40kHz range

Let's try 1 microsecond, (i.e. $L = 0.000001$). Figure 10 shows the results. If we have a look at the left graph we see that the frequency spectrum has flattened quite considerably. The -3dB crossing is somewhere around 15kHz now. On the right we see that magnitude frequency response of the 1 microsecond pulse doesn't even drop half of a dB over the entire 40kHz range. Perfect!

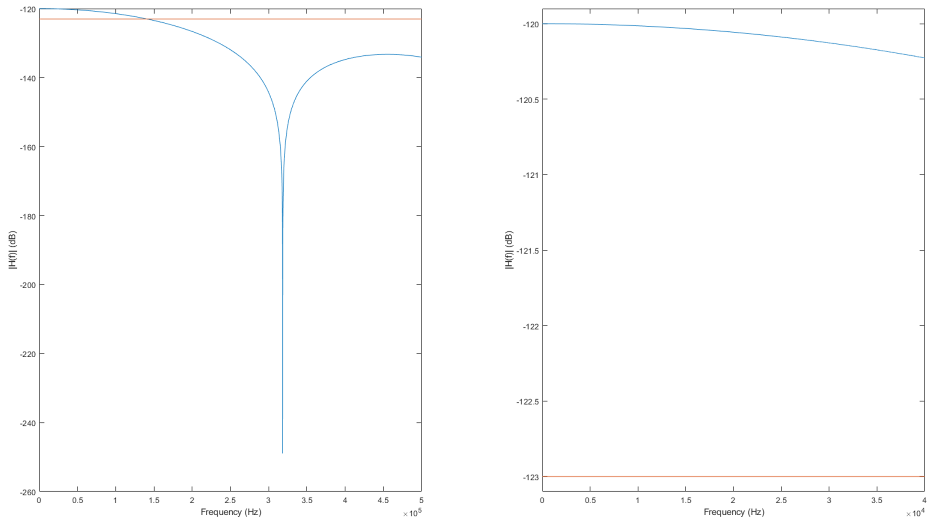

Figure 10: Magnitude Frequency Response of a Rect Pulse with T=1us (Blue) and the -3dB down line (Red). Left: 500kHz range Right: 40kHz range

~~:w~~

Could we get better? Of course! Let's see what a 1 nanosecond pulse would look like. Figure 11 shows the results. Now the frequency range is perfectly flat over not only the 40kHz range but it's also flat over the entire 500kHz range. Even better!

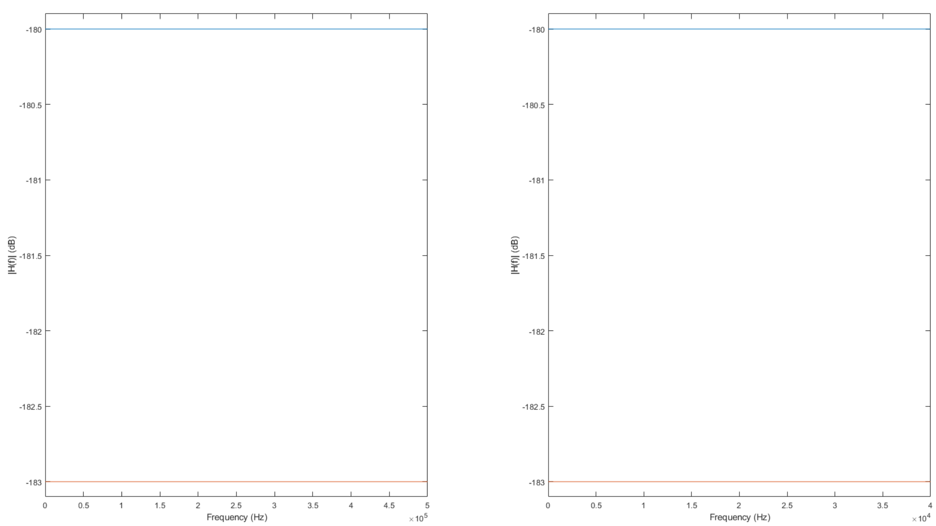

Figure 11: Magnitude Frequency Response of a Rect Pulse with T=1ns (Blue) and the -3dB down line (Red). Left: 500kHz range Right: 40kHz range

But wait. What is that. Have a look back at Figure 10 and Figure 11 and take note of the maximum y-axis value. For the 1us pulse the magnitude is about -120dB but for the 1ns pulse the magnitude is about -180dB. That is a -60 dB drop, which for the uninitiated is a ton! In terms of voltage a -60dB drop is a difference of 1000V in input to output voltage; if we put 1000V we would get out 1V. What does this mean for our pulse? It means that the thinner our pulse the more voltage we have to put in to overcome noise. In other words, if we use a 1V 1us rectangular pulse to probe our filter we would have to us a 1000V 1ns rectangular pulse to get the same output from the filter. This is problematic because transitioning from 0V to 1000V in a one nanosecond period is rather impossible as crafting even a 1V nanosecond pulse will pose some interesting electrical engineering problems. For us that means that we should not overdo things, let's use the 1us pulse as it is flat over our region of interest and it is much easier to craft.

To proceed we set our function generator to make a 1us 1V pulse and input it into our theoretical lowpass filter from Figure 5. On the oscilloscope we turn on the FFT and set the frequency range to 20kHz. What's the output look like? Pretty much the same as Figure 6 except the zeros at the end of the filter extend out to 20kHz and the spacing between our points will be much smaller (remember the FFT is a DFT which isn't continuous). Now we can have the oscilloscope generate a screenshot of the FFT if we need it for a report or we could have it output the FFT to a CSV file so we could explore it in Matlab. And how long did this take? Probably about a minute. To formalize the process we generally connect the function generator directly to the oscilloscope and output the pulse. Then we take the FFT of the pulse on the oscilloscope and ensure that whatever pulse parameters we have chosen produce a flat FFT over the frequency range of interest. Then we just unplug the function generator from the oscilloscope and plug it into the input of the circuit and the output of the circuit to the oscilloscope and observe. One last thing to note before we move on to the project. Look at Figure 5 again. Notice how the magnitude response is in the positive dB range? That's because it's been a lowpass amplifier this whole time! I was lying to you. But all the same the pulse analysis process works for examining how the amplifier responds to different frequencies. In fact, pulse analysis works for a great number of things and isn't simply constrained to filters or amplifiers. For example, you could take a group of working circuit boards and pass a pulse through their various inputs and power ports to obtain the frequency response of working boards. Then you could take a board that is broken and pass pulses through all the various ports and whichever port has a differing frequency response probably contains the components that broke. If you have a large enough data-set you could set up some machine learning algorithms to automate the process of identifying working and non-working boards; even potentially having the machine learning algorithms learn exactly which parts are broken on the board given the frequency responses.

5 Project

1. Design three filters. A passive low-pass RC filter, a passive high-pass RC filter, and an active low-pass Butterworth filter. For the passive and active lowpass filters make the cutoff frequency 25kHz. For the high-pass filter make the cutoff frequency 15kHz.

2. Find the transfer functions for these filters and plot the magnitude frequency response in dB using Matlab.

3. Build the filters on a breadboard.

4. ~~Come into the lab on one of the workshop days.~~

5. Obtain the magnitude frequency response plot of the filters over the 40kHz range by hand using steps of 1kHz.

6. Obtain the magnitude frequency response plot of the filters over the 40kHz range using pulses.

7. Connect the output of your high-pass filter to the input of the lowpass filter. Obtain the magnitude of this new filter over the 40kHz range using pulses.

8. At home plot the magnitude frequency response plot data that you obtained by hand. ~~Compare the FFT and by hand plots to your Matlab results.~~

6 Report

Give a Teare-style report (i.e. Intro, Background, Results, Discussion, Conclusion). The Background should basically re-iterate what I have said here but in your own words; we are checking for understanding so don't skip out here. The Results should include the designs of your filters and the values of specific components, Matlab plots of the magnitude response for the transfer functions of the filters, the plots of your data that you found by hand~~, and the FFT plots as generated by the oscilloscope~~. The Discussion should compare and contrast the plots from the Results section. The Intro and Conclusion should be written last and the Conclusion should include your thoughts on pulse analysis vs by-hand analysis.