Analog and Digital Mixing of Signals

Marta Pastore

March 7, 2014

Abstract

The goal of this lab is to understand the difference between analog and digital mixing of two signals. Ideally, the result of the two should be similar but we experienced trouble with the DC offset in the digital case. Furthermore, in this lab we were able to set the tap coefficients for a 5/8 FIR filter, which are listed in Table 2. These coefficients produce a respose close to the expected response of the filter which is shown on Figure 20.

1 Introduction

In this lab we were introduced to various advanced equipment used for the mixing of signals and processing of these signals. More specifically, we explored the functions of a field programmable gate array or FPGA when inputing two SRS oscillator signals. The FPGA contains a series of programmable logic blocks and can therefore perform different tasks at once, each completed independently from one another. The FPGA we used has a built in analog-to-digital converter or an ADC card to which we connected the input signals. The connections to the ADC card determines whether the signal is sampled or mixed. In addition, we explored the workings of a finite impulse response filter or FIR filter inside the FPGA. The way an FIR filter works is it takes a time series of samples, uses these to implement a convolution, multiplies each sample by some coefficient and adds the results together. The coefficient values are calculated for a specific frequency response of the FIR filter. With the understanding of the FPGA and FIR filter, we were able to implement a Digital-Down-Converter (DDC) that mixes and filters our input signal.

2 Methods

We began by exploring how to control the SRS Function Generators and the Pulsar sampler from the command line. To do this the DFEC module was imported in our python code to set the values for the SRS and Pulsar. Within the module, the function set srs was used to control the two SRS used.To control how to take samples, the sampler function was used.

2.1 Nyquist Sampling and Aliasing

From Nyquist sampling it is known that one must sample at a rate that is more than twice the maximum frequency in the sprectrum of the signal in order to avoid aliasing. To explore this idea we took samples from a single sine wave. We chose a sample frequency, ν_{sampl}, of 100kHz. The signal frequency, ν_{sig}, was set to be some fraction of ν_{sampl}. The values of the signal frequency for which data was obtained are listed in Table 1. At each ν_{sig}, N

ν_{sig}	fxν_{sampl}
10KHz	$0.1\nu_{sampl}$
20kHz	$0.2\nu_{sampl}$
30kHz	$0.3\nu_{sampl}$
40kHz	$0.4\nu_{sampl}$
50kHz	$0.5\nu_{sampl}$
60kHz	$0.6\nu_{sampl}$
70kHz	$0.7\nu_{sampl}$
80kHz	$0.8\nu_{sampl}$
90kHz	$0.9\nu_{sampl}$

Table 1: The values of the signal fequency for which data was collected.

contiguous samples are taken and we set N equal to 256. For each data set, we plotted the digitally sampled waveform versus time and the Fourier power spectrum.

Next, we set ν_{sig} equal to ν_{sampl} which is 100kHz and took 256 samples. A plot of the sampled waveform versus time was obtained and is shown in Figure 1. We then violated the Nyquist's criterion by setting ν_{sig} equal to $29.9\nu_{sampl}$ and a plot of the samples waveform versus time was created and is shown in Figure 2.

Now we caculate the power spectrum for many more than N output frequencies over the Nyquist range $-\nu_{sampl}/2$ to $+\nu_{sampl}/2(1-2/N)$ This means we make the frequency increment much smaller than $\Delta\nu = \nu_{sampl}/N$ We chose N=10,000 in order to achieve this and we set the $\nu_{sig} = 0.1\nu_{sampl}$, where ν_{sampl} is 100kHz. We then plotted the sampled waveform vs time and the power spectrum which is shown in Figure 3.

2.2 Basic Double Sideband (DSB) Mixer Operation

In this section we used two SRS synthesizer oscillators as inputs to a mixer to explore the spectra and the waveform in the DSB mixing process. The SRS synthesizer works up to 30MHz. We set one of the synthesizers as our local oscillator with frequency ν_{lo} equal to 300kHz. The other synthesizer was set as a signal oscillator with frequencies $\nu_{sig} = \nu_{lo} \pm \delta\nu$. Here $\delta\nu$ is the frequency difference and we set that equal to 5kHz. We sampled for both the sum and differance frequencies using two types of mixers.

2.2.1 The Mini-Circuits ZAD-1 Mixer

We combined the two signals, ν_{lo} and ν_{sig}, using a Mini-Circuits ZAD-1 mixer, which has three ports. We connected ν_{lo} and ν_{sig} to the two outer ports R and L. The middle port,

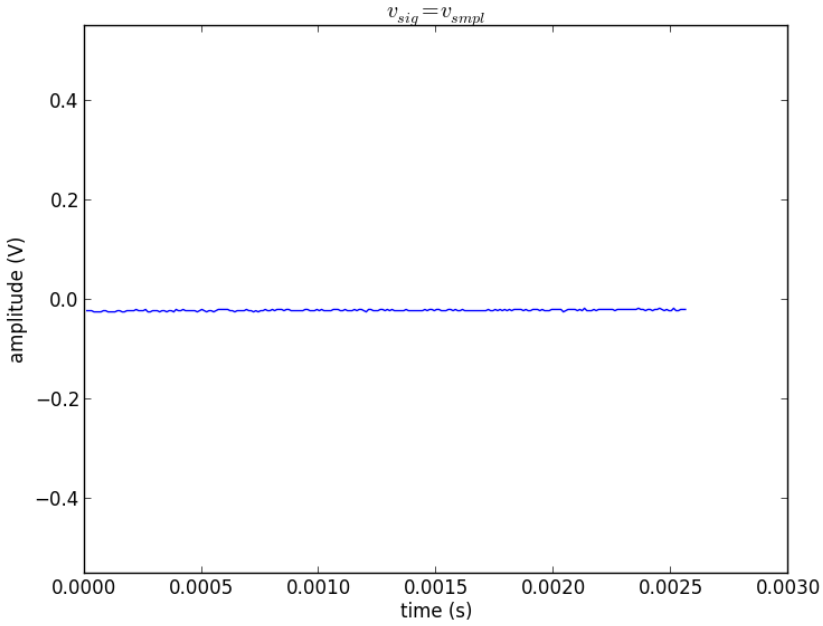

Figure 1: A plot of the sampled waveform versus time for $\nu_{sig} = \nu_{sampl} = 100$kHz.

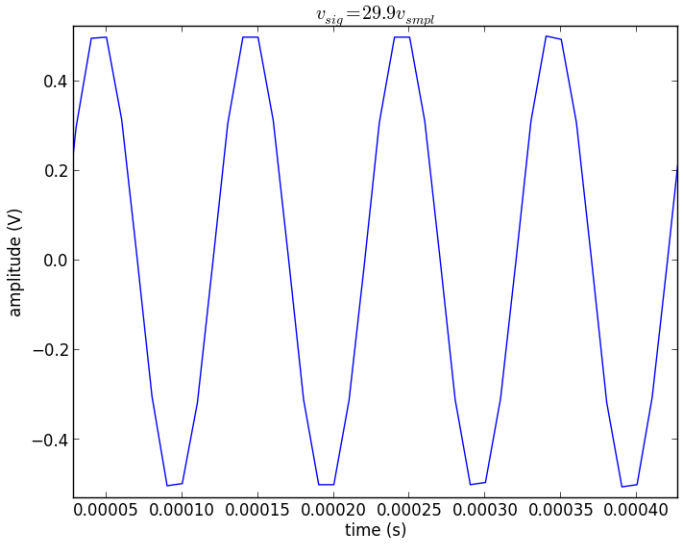

Figure 2: A plot of the sampled waveform versus time for $\nu_{sig} = 29.9\nu_{sampl}$ with $\nu_{sampl} = 100$kHz.

I, is the output with the mixed frequencies and this was connected to a DC sampler. We plotted the sampled signal and the power spectra versus frequency for both cases, $\nu_{sig} = \nu_{lo} \pm \delta\nu$, and they are shown in Figure 4 and 5. For the $\nu_{sig} = \nu_{lo}$ - $\delta\nu$ case, we took the Fourier transform of the waveform and removed the sum frequency component by zeroing both the real and imaginary portions. We then attempted to recreate the signal from the filtered

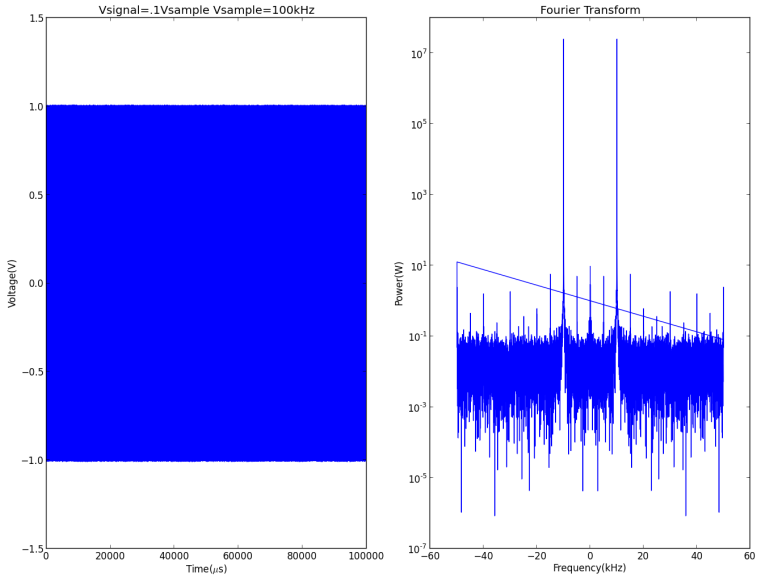

Figure 3: A plot of the sampled waveform vs time and the power spectrum for N = 10,000.

transform by taking the inverse fourier transform. A plot of the sampled signal versus time was obtained.

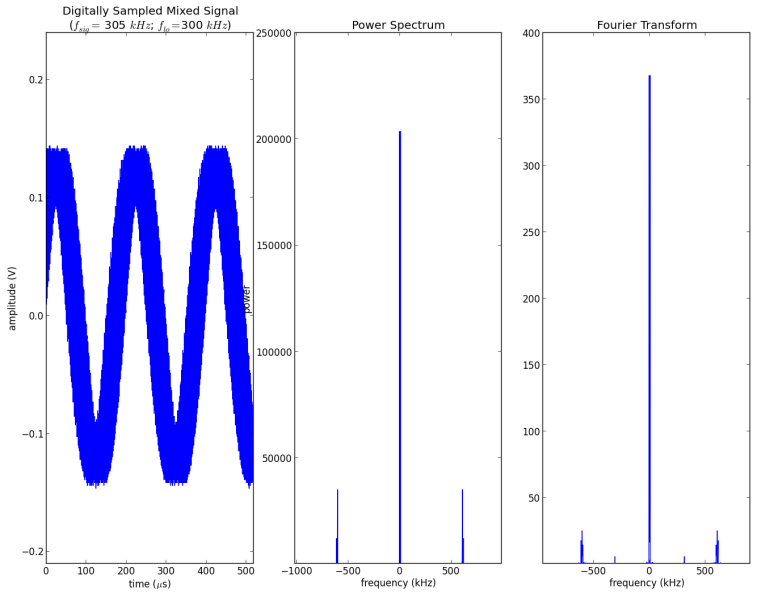

Figure 4: A plot of the sampled signal, the power spectra, and the fourier transform of ν_{sig} = ν_{lo} + $\delta\nu$.

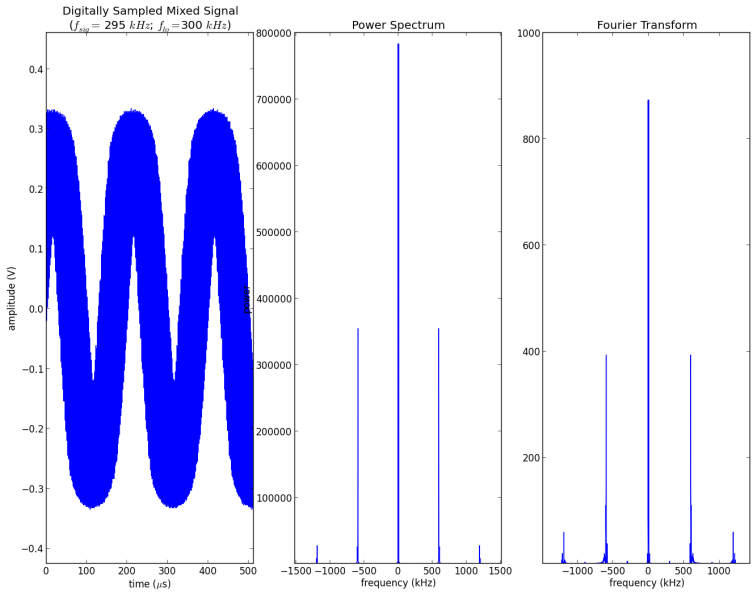

Figure 5: A plot of the sampled signal, the power spectra, and the fourier transform of ν_{sig} = ν_{lo} - $\delta\nu$.

2.2.2 Digital Mixing on an FPGA

We digitally mixed the same input frequencies as for the Mini-Circuits case, ν_{lo} =300kHz and ν_{sig} = 295kHz, by collecting data using an FPGA hosted on the ROACH. The FPGA has an ADC card with various ports but for this case we connected our two signals to ports 2 and 3 in order to obtain a digitalized mixed signal. In this case the sample frequency was preset to 200MHz. A plot of the power spectra versus frequency was obtained for the mixed signal.

Next, instead of mixing two input signals digitally, we mixed a single signal with a local oscillator that is derived from the sample clock of the FPGA. Our signal frequency was set to 2MHz and we connected this signal to port 4 on the ADC card. We used

```
echo -ne "x00\x00\x00\x02" > lo_freq
```

to set the local oscillator frequency. What we are doing here is mixing our signal frequency once with a cosine wave and with a sine wave of the chosen oscillator frequency. A plot of the resulting waveform and power spectrum is shown in Figure 6 and 7.

2.3 Setting FIR filter coefficient

We are now including eight registers for inputting the coefficients for the FIR filter circuit in the FPGA. Each coefficient has two registers one for the real part, $coeff_{real}$, and one for the imaginary part, $coeff_{imag}$. The coefficients were chosen as to implement a 5/8 bandpass filter. This means that when we divide the band into 8 positive and negative frequency

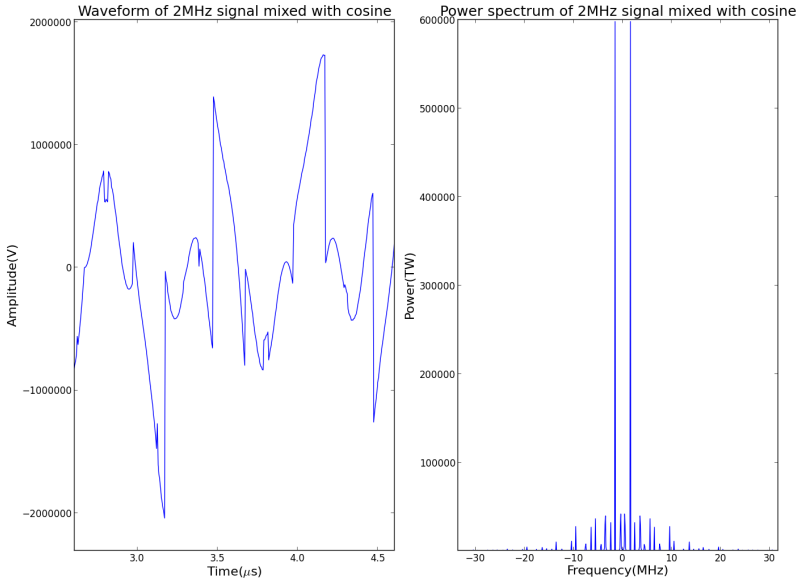

Figure 6: A plot of the waveform and power spectrum of a 20MHz signal wave mixed with a cosine wave.

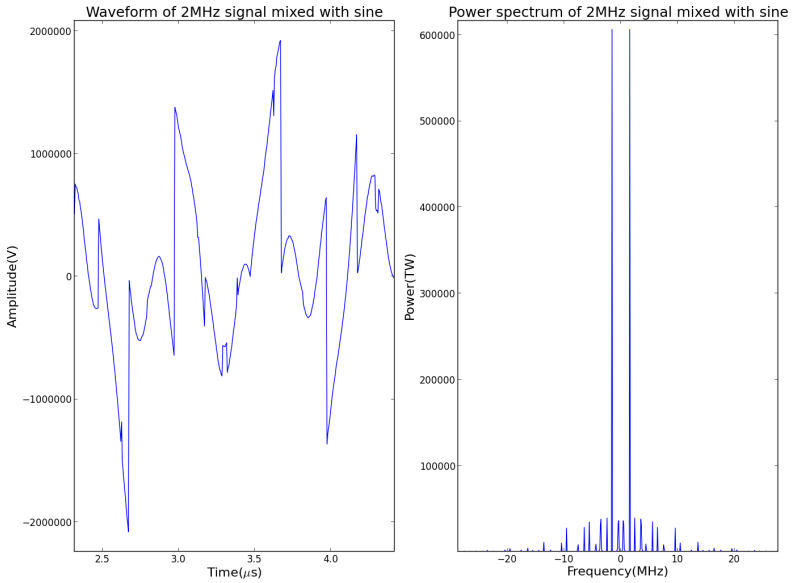

Figure 7: A plot of the waveform and power spectrum of a 20MHz signal wave mixed with a sine wave.

channels, we would extract the five channels centered around zero. The coefficient values chosen are shown in Table 2.

Frequency(MHz)	Coefficient
-100	0.1250000
-75	-0.0517767
-50	-0.01250000
-25	0.03017767
0	0.6250000
25	0.3017767
50	-0.1250000
75	-0.0517767

Table 2: The coefficient values for the 5/8 bandpass FIR filter.

3 Results

3.1 Effects of the Nyquist criterion

We digitally sampled at a frequency of 100kHz and took data for the values of ν_{sig} shown on Table 1. In this case the Nyquist frequency is half of the sampling frequency or 50kHz. At this frequency we are sampling every half period; therefore we only expect the peaks and troughs of the signal to be sampled. The resulting plot we obtained at the Nyquist frequency is shown in Figure 8 and it agrees with what we expect from sampling at the Nyquist frequency.

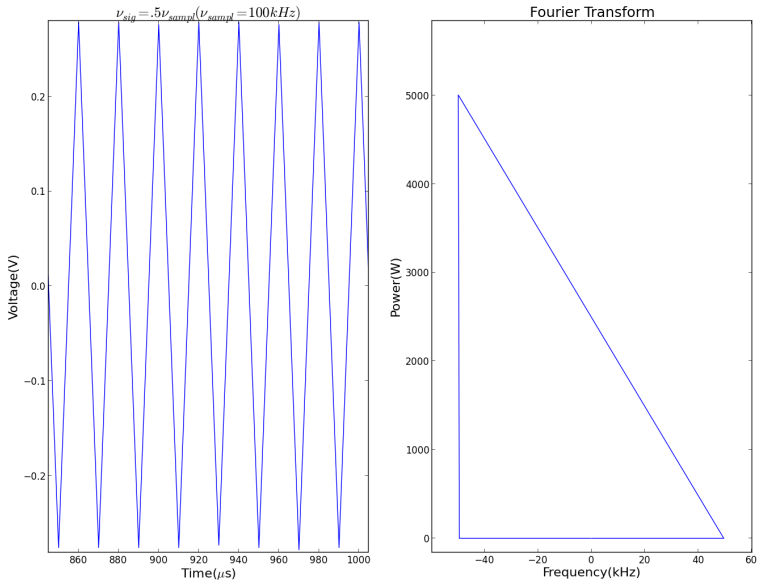

Figure 8: A plot of the sampled waveform and power spectrum at $\nu_{sig} = 0.5\nu_{sampl}$, Nyquist frequency.

For the values of ν_{sig} we are undersampling the sinusoid and this allows there to be an

alias for each ν_{sig}. An alias is a function that produces the same set of samples. For example when we plot the sampled waveform and fourier transform at $\nu_{sig} = 0.1\nu_{sampl}$ and $\nu_{sig} = 0.9\nu_{sampl}$, shown in Figure 9 and 10, we can see that the peaks of the fourier transform in both cases occur at 10kHz eventhough they are different signal frequencies. This is known as aliasing and we experienced this also for $\nu_{sig} = 0.2\nu_{sampl}$ with $\nu_{sig} = 0.8\nu_{sampl}$, shown in Figure 11 and 12, which peak at 20kHz, for $\nu_{sig} = 0.7\nu_{sampl}$ with $\nu_{sig} = 0.3\nu_{sampl}$, shown in Figure 13 and 14,which peak at 30kHz and for $\nu_{sig} = 0.6\nu_{sampl}$ with $\nu_{sig} = 0.4\nu_{sampl}$, shown in Figure 15 and 16, which peak at 40kHz. It can be noted that the original ν_{sig} is fully recovered at the frequencies below the Nyquist frequency but decreased at frequecies above the Nyquist frequency. This is observed because the Nyquist frequency is the highest frequency that can be coded at a given sampling rate in order to be able to fully reconstruct the signal.

In Figure 1 the plot for $\nu_{sig} = \nu_{sampl} = 100$kHz is shown and we can see that the result is approximately a straight line close to zero amplitude. This is because we are sampling once per period so the data results in a straight line.

In Figure 2 we explored the case of $\nu_{sig} = 29.9\nu_{sampl}$, a signal frequecy way below the Nyquist frequency. Here we determine a frequency of 10kHz compared to a signal frequency of 2.99MHz. We are taking so little samples per period that we are no where near our original frequency. Once again an example of how the Nyquist frequency is the highest frequency at which we can get back our original frequency.

Furthermore, in Figure 3 we illustrate the sample frequency waveform and the power spectrum of the case in which we set the number of samples, N, equal to 10,000. This will make the frequency increments smaller. Making the output frequencies closer together gives a more continuous frequency coverage in the power spectrum. In the power spectrum shown in figure 3, there are two peaks at around 10kHz and -10kHz but there is a series of smaller power outputs on either side of the two main peaks. Because an FFT has a finite length, there is a window on the data that is taken. A window in the time domain results in a convolution in the frequency domain with the transform of the window. This transform is a Sinc function which has infinite width. Therefore, the ripples of the Sinc function shown up a leakage far from the spectral peaks that do not fit the FFT's length.

Looking in Figure 3 at the frequency values of the highest peaks, we get a difference between the two peaks of about 1.17kHz. The total time span over which the samples were taken is 100,000μs. The inverse of this value is 10Hz.

3.2 Analog and Digital DSB Mixing Observations

In Figures 4 and 5, the data for mixing with a Mini-Circuits ZAD-1 mixer for two cases, $\nu_{sig} = \nu_{lo} \pm \delta\nu$, is illustrated. For the addition case, Figure 4, it can be seen that in both the fourier and power plots there is a peak close to zero, which represents the difference between two frequencies, +5kHz and -5KHz. There are also peaks at +605kHz and -605kHz; these represent the addition of our two input frequencies. In this case the lower sideband goes from -5kHz to -605kHz and the upper sideband goes from +5kHz to +605kHz. Similarly for the differance case, Figure 5, there are peaks at +5kHz and -5kHz as well as +595kHz and -595kHz. Here the lower sideband ranges from -5kHz to -595kHz and the upper sideband gors from +5kHz to +595kHz.

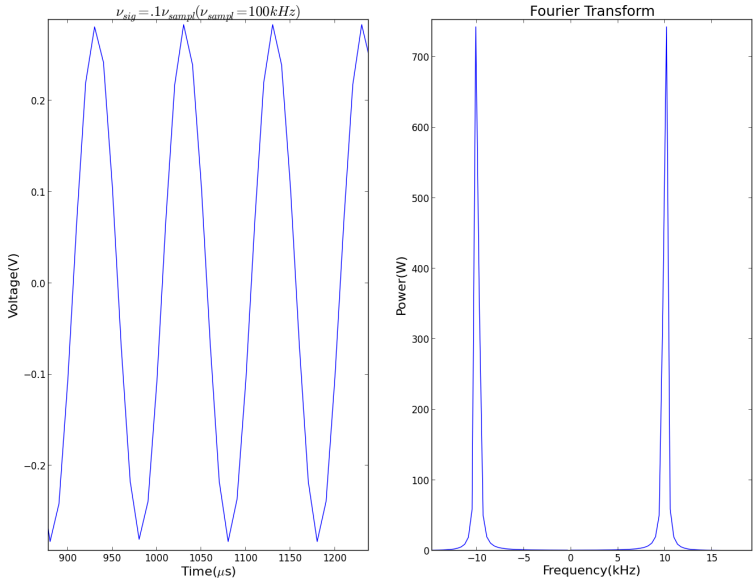

Figure 9: A plot of the sampled waveform and power spectrum at $\nu_{sig} = 0.1\nu_{sampl}$

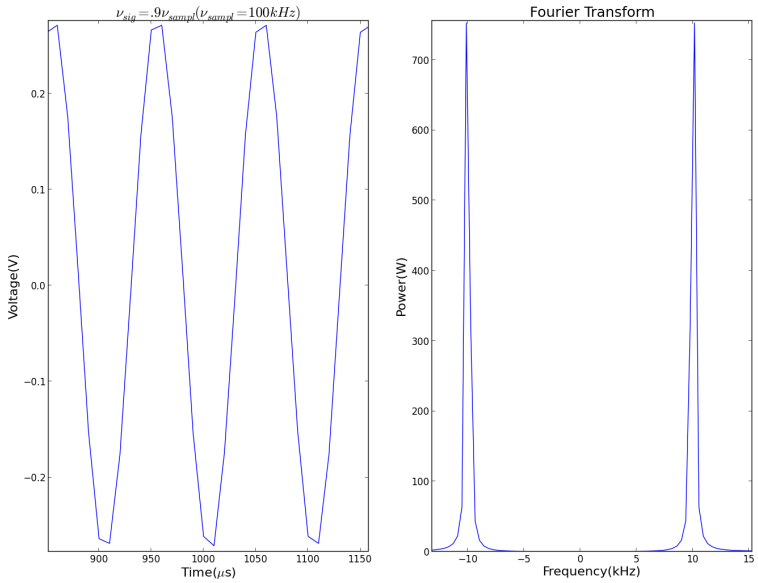

Figure 10: A plot of the sampled waveform and power spectrum at $\nu_{sig} = 0.9\nu_{sampl}$.

Focusing now on the data from the difference case, Figure 4, the waveform plotted matched the trace seen on the oscilliscope. Furthermore, looking at the fourier transform plot, we zeroed out frequencies that were higher than -5kKhz and +5kHz. This is known as fourier filtering. When we took the inverse transform on this new data we obtained the

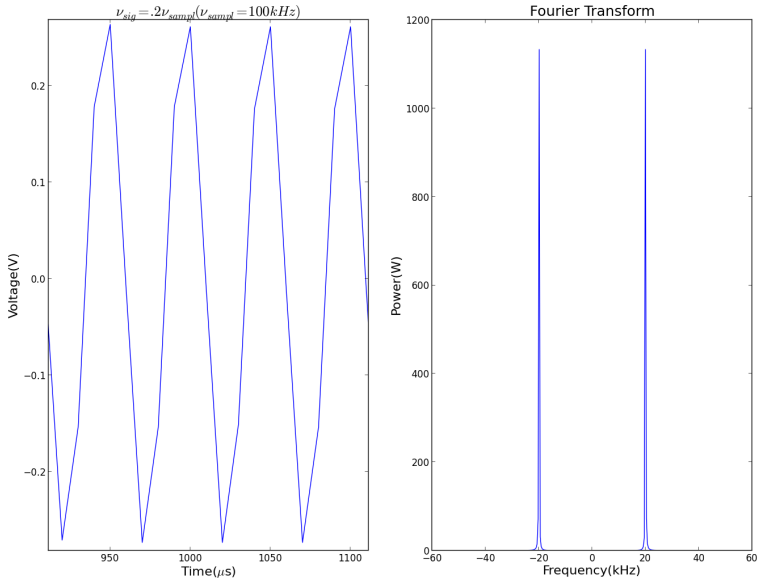

Figure 11: A plot of the sampled waveform and power spectrum at $\nu_{sig} = 0.2\nu_{sampl}$.

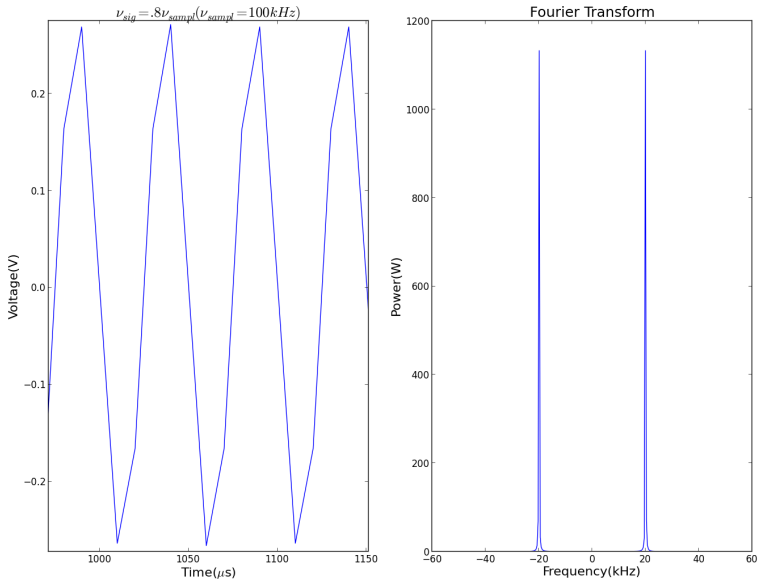

Figure 12: A plot of the sampled waveform and power spectrum at $\nu_{sig} = 0.8\nu_{sampl}$.

waveform shown in Figure 17. Here we have a waveform of frequency 5kHz just like we wanted. We checked to see if there were any imaginary components in the waveform and there were and it is propably because we did not zero out all of the negative frequencies.

The result of mixing the two signals $\nu_{sig} = 295$kHz and $\nu_{lo} = 300$kHz using digital DSD

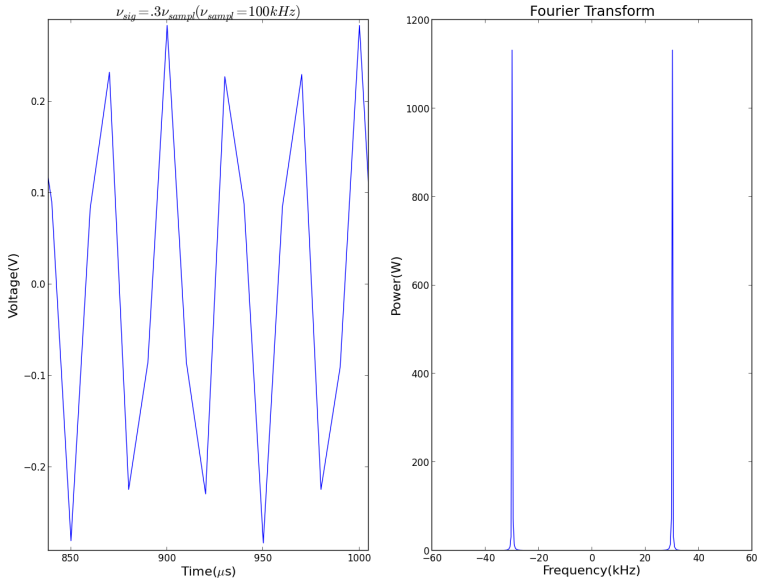

Figure 13: A plot of the sampled waveform and power spectrum at $\nu_{sig} = 0.3\nu_{sampl}$.

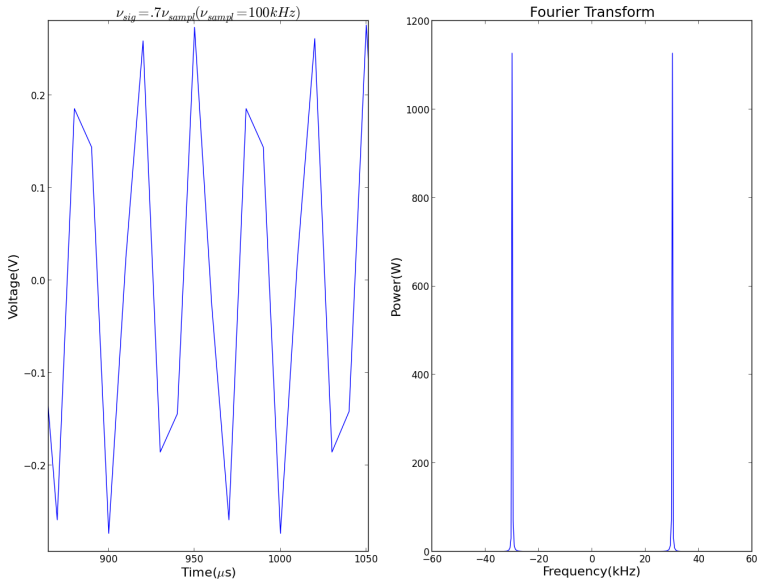

Figure 14: A plot of the sampled waveform and power spectrum at $\nu_{sig} = 0.7\nu_{sampl}$.

mixing with the FPGA is shown in Figure 18. What we expected to see was a waveform of frequency 5kHz and peaks in the power spectrum for the two input frequencies and the addition and the difference of the two frequencies like we observed in the analog mixing case shown in Figure 5. Instead we only observed peaks at around 300kHz. This means that

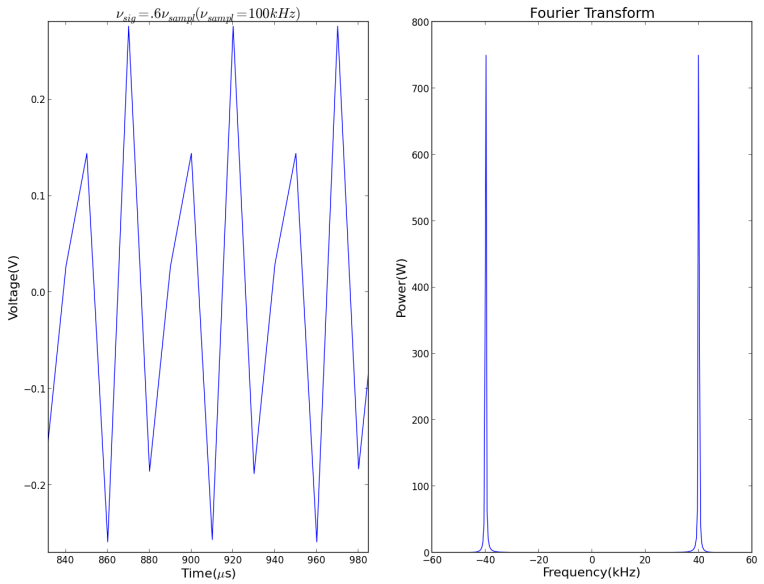

Figure 15: A plot of the sampled waveform and power spectrum at $\nu_{sig} = 0.6\nu_{sampl}$.

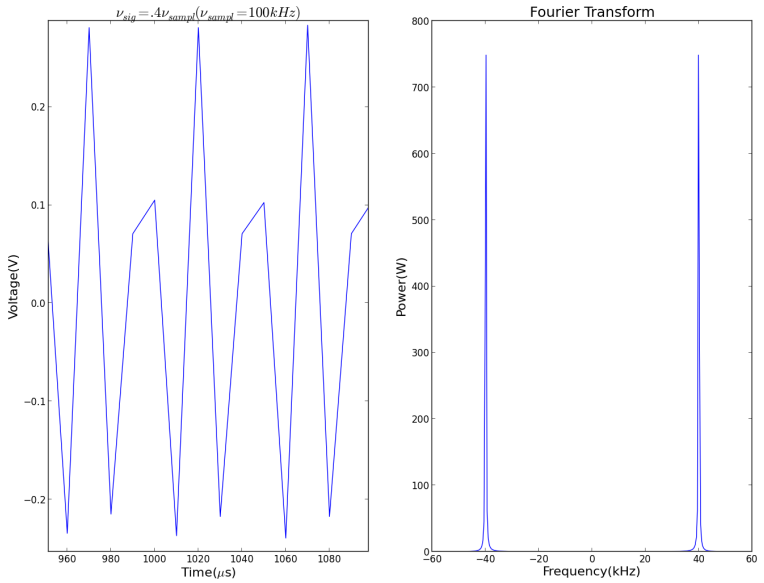

Figure 16: A plot of the sampled waveform and power spectrum at $\nu_{sig} = 0.4\nu_{sampl}$, Nyquist frequency.

our signal did not get mixed properly. The reason for this result has to do with DC offset. Since we are taking 2048 samples at 200MHz, we have a a resolution of 100kHz. Our $\delta\nu$ is only 5kHz which is much smaller than the resolution frequency. So we do not have enough

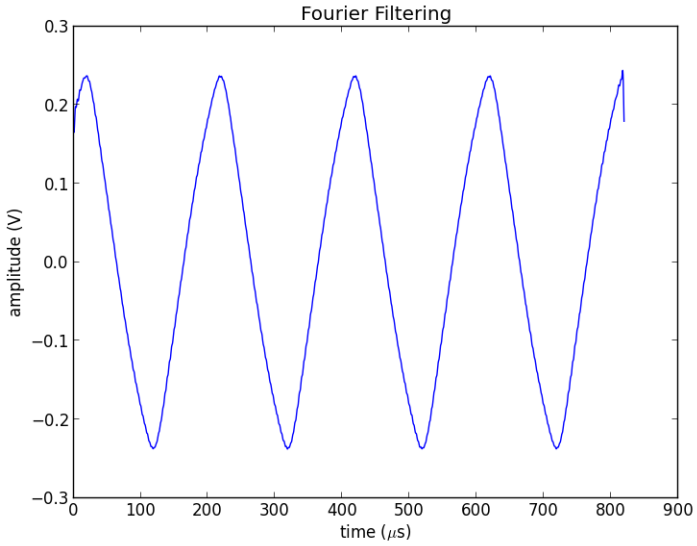

Figure 17: A plot of the fourier filtered waveform.

resolution to separate 5kHz and 0kHz, which is the DC frequency. Because the ROACH has a DC offset, our 5kHz signal is being overlooked by this offset. Also, this offset makes it so that when any other frequency is mixed with the DC signal, the same frequency appears as the output. This explains why we only see a peak at around 300kHz.

When looking at Figures 6 and 7, these show our 2MHz signal mixed with a cosine signal and a sine signal, repectively. Here the high peaks in the power spectrum happen at around +2MHz and -2MHz which corresponds to our signal frequency. We took the data for the cosine and sine mixing and combined it to reconstruct the complex valued sinudoid that this digital Single-Sideband (SSB) mixer outputs. The result of the combined power spectrum and the waveform of the extracted sinusoid are shown in Figure 19. The highest peak in the power spectrum occurs at approximately 1.5MHz and we chose to filter out this specific frequency. We did this by setting the power to zero for all the other frequencies and the waveform of this filtered frequency is shown in Figure 19. The frequency of the plotted waveform is about 1.3MHz, which is really close to the aimed 1.5MHz peak.

When comparing between SSB and DSB mixing we look at Figures 17 and 19. In Figure 17 we see the waveform of a 5kHz fourier filtered signal. Similarly, in Figure 19 we see the waveform of a 1.5MHz fourier filtered signal. This waveform was obtained usign a SSB mixer and it is safe to say that it looks more like a sine wave than Figure 19, which was obtained using a DSB mixer. It seems that SSB mixer is better at fourier filtering a signal than the DSB mixer. Now focusing on Figures 18 and 19, we can compare the power spectrum that results from DSB mixing and SSB mixing, respectively. Here we can see that Figure 18 has two main peaks that corresponds to the double sideband, while Figure 19 only has one main peak that represents a single sideband. In addition, the power spectrum of the SSB mixer, Figure 19, has a couple of tiny peaks. It should result in just a single peak. This could be a sign that the power spectrum is more realistic for the DSB than the SSB in our case, since in the DSB we get exactly two peaks in the power spectrum.

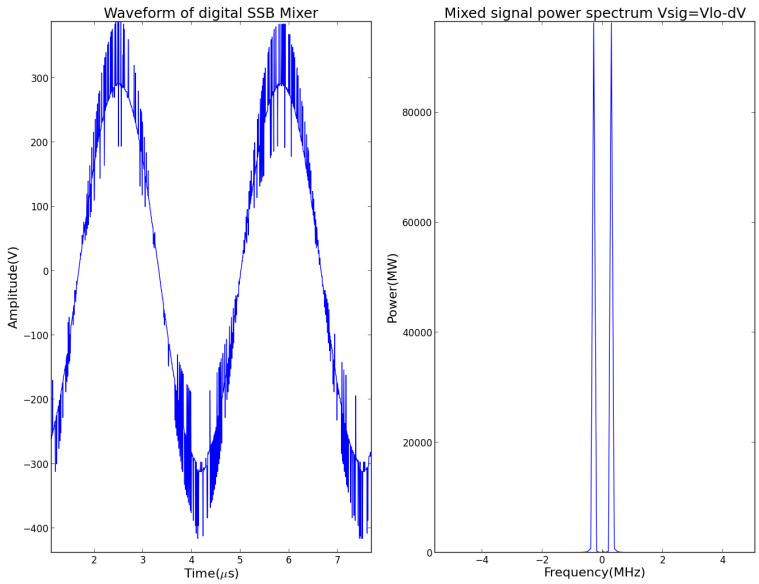

Figure 18: A plot of the waveform and power spectrum that shows the results of digitally mixing $\nu_{sig} = 295$kHz and $\nu_{lo} = 300$kHz with an FPGA.

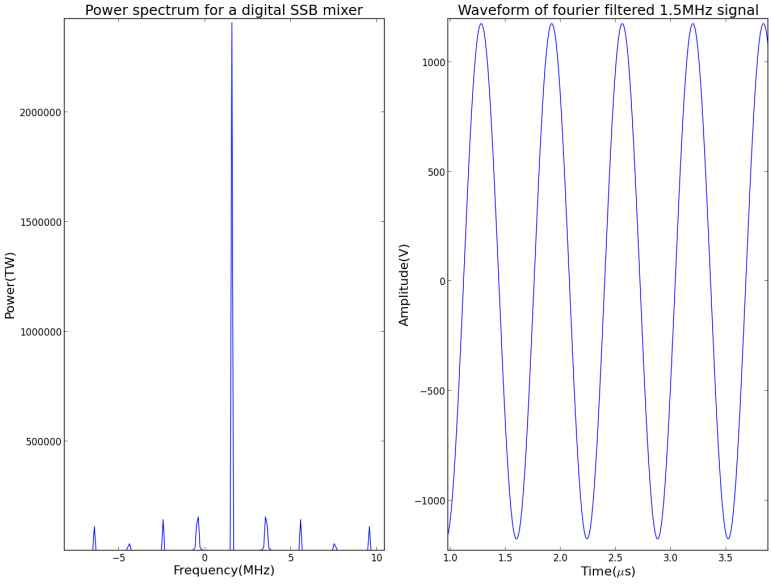

Figure 19: The first plot shows the power spectrum of the combined sine and cosine respose of a 2MHz signal.The second plot shows the waveform of the extracted 1.5MHz peak.

3.3 FIR filter response

The coefficients chosen for the FIR filter are shown in Table 2. The expected response of our 5/8 bandpass filter is shown in Figure 20. Our signal frequency for this case is 200MHz but

since we know we must sample at half the signal frequency, our filter only takes in frequencies between 100MHz and -100MHz to avoid aliasing. The response of our filter in time domain is also shown in Figure 20 and here we see the coefficients with respect to time.

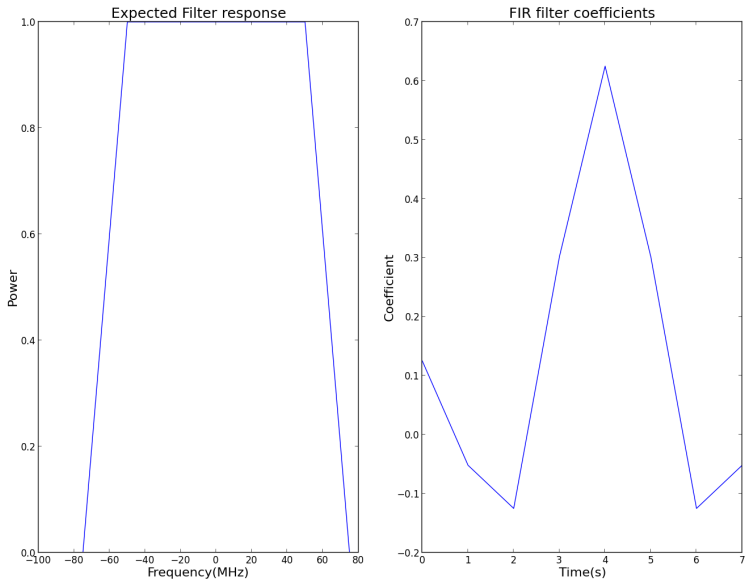

Figure 20: The first plot shows the expected response of a 5/8 bandpass filter on a 200MHz signal.

4 Conclusion

In this lab we mixed two signals using an analog and a digital mixer. We plotted out data and found out that the power spectrum for these two cases is not the same. For the digital mixing we did not see any signs of mixing which was due to DC offset described earlier. In the future we would like to fix this problem by paying more attention to the value of the input frequency and making sure that the difference between the two is closer to the resolution of the mixer.

Furthermore, we will like to test our coefficients by programing them to the FIR filter running on the FPGA and empirically characterize the shape of our filter.

5 Acknowledgements

I would like to thank my lab partner Kyle Moses for his contributions to the results of this lab. I would also like to thank Karto Keatings and Baylee Bordwell for helping with the obtaining and interpretation of my data.