Mitchell Larson
EE 3221
12/18/17
Lab3

1) Square Wave Response

Upon running the program square_intr.c, the following square wave was observed

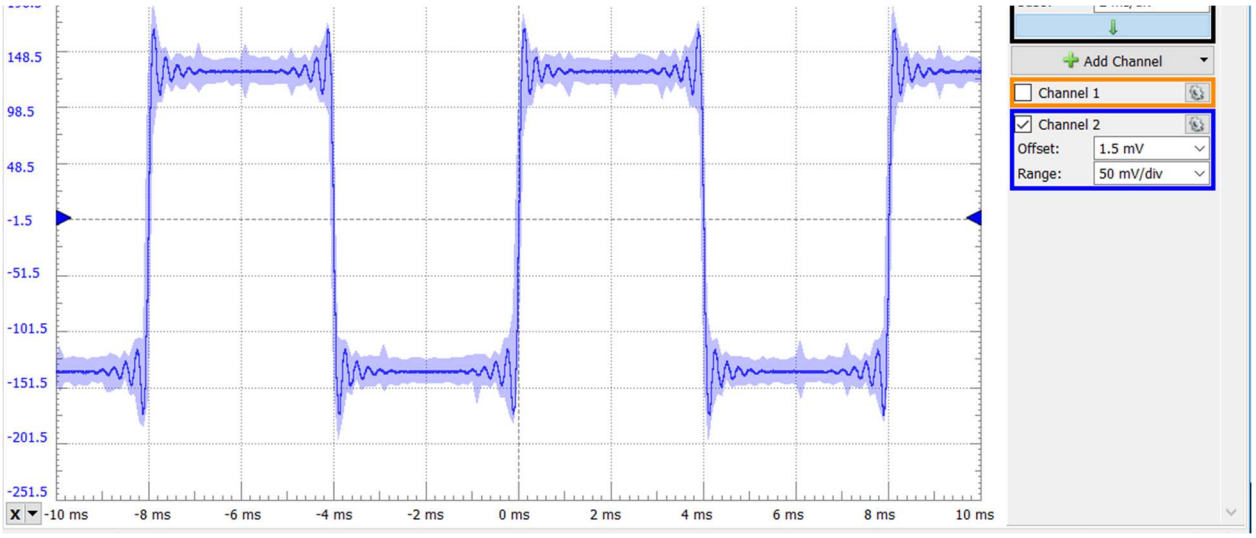

Figure 1 - The output of square_intr.c measured from an oscilloscope.

Based on the results observed above, the impulse response of the DAC and reconstruction filter was predicted to look like a series of spikes in the positive y direction every 8 ms and negative every 8 ms shifted from the origin by 4 ms. It was predicted that the spikes would not be perfectly defined due to the filter that the signal was run through. A perfect spike would require all frequencies; however, this is not possible in a real-life setup. The logic for this is the impulse response is the derivative of the unit step response (shown above). The derivative of the unit step response should be a series of spikes with smaller impurities located next to the large spikes.

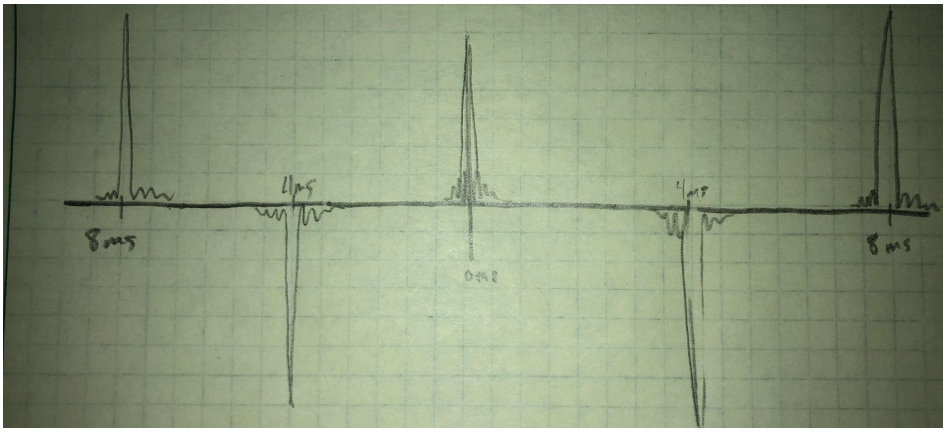

Figure 2 - Predicted impulse response

Mitchell Larson
EE 3221
12/18/17
Lab3

2) Impulse Response

The program square_intr.c was modified to provide an approximation to the DAC and reconstruction filter impulse response. The results are shown below:

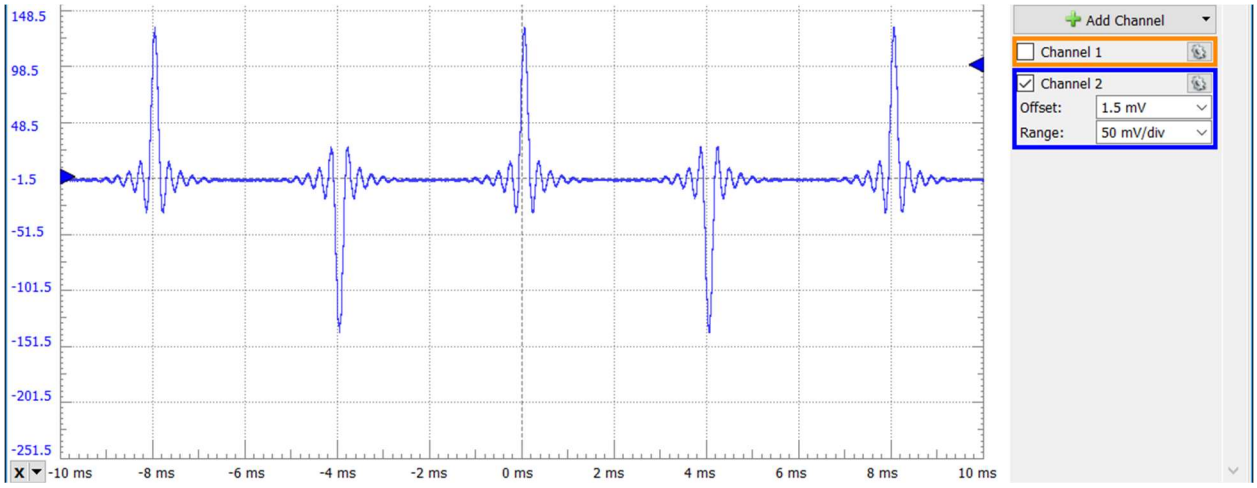

Figure 3 - DAC and reconstruction filter impulse response.

```
const int16_t square_table[loop_size] = {
10000, 00000, 00000, 00000, 00000, 00000, 00000, 00000,
00000, 00000, 00000, 00000, 00000, 00000, 00000, 00000,
00000, 00000, 00000, 00000, 00000, 00000, 00000, 00000,
00000, 00000, 00000, 00000, 00000, 00000, 00000, 00000,
-10000, -00000, -00000, -00000, -00000, -00000, -00000, -00000,
-00000, -00000, -00000, -00000, -00000, -00000, -00000, -00000,
-00000, -00000, -00000, -00000, -00000, -00000, -00000, -00000,
-00000, -00000, -00000, -00000, -00000, -00000, -00000, -00000};
```

Figure 4 - Code Modifications to produce the impulse response.

The observed results seemed to confirm the prediction that was made in the previous step. The impulse response resulted in positive spikes every 8ms starting at 0ms and negative spikes every 8ms starting at 4ms. In addition, due to the inability to reconstruct every frequency, the spikes portray some imperfection with characteristics similar to a sinc function. This result is interesting to note because the fourier transform pair of a rectangular signal is a sinc function, which could explain the sinc characteristics introduced into the result.

To produce the impulse response, every table entry except for the 1st and the 33rd were replaced with a value of 0. This would produce a positive and negative spike for each cycle.

Mitchell Larson
EE 3221
12/18/17
Lab3

3) Magnitude Frequency Response of DAC Reconstruction Filter – fs = 8 kHz

The prbs_intr.c program was run and both the time and frequency domain responses were captured. The program was responsible for generating random signals with varying frequencies.

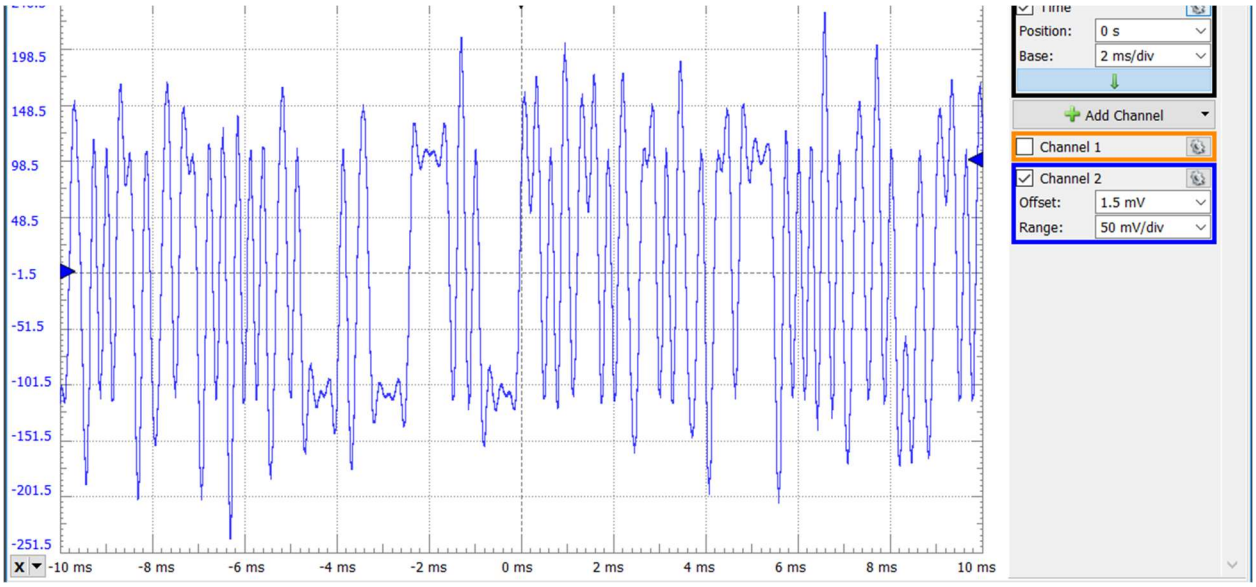

Figure 5 - Time domain response captured with fs = 8 kHz

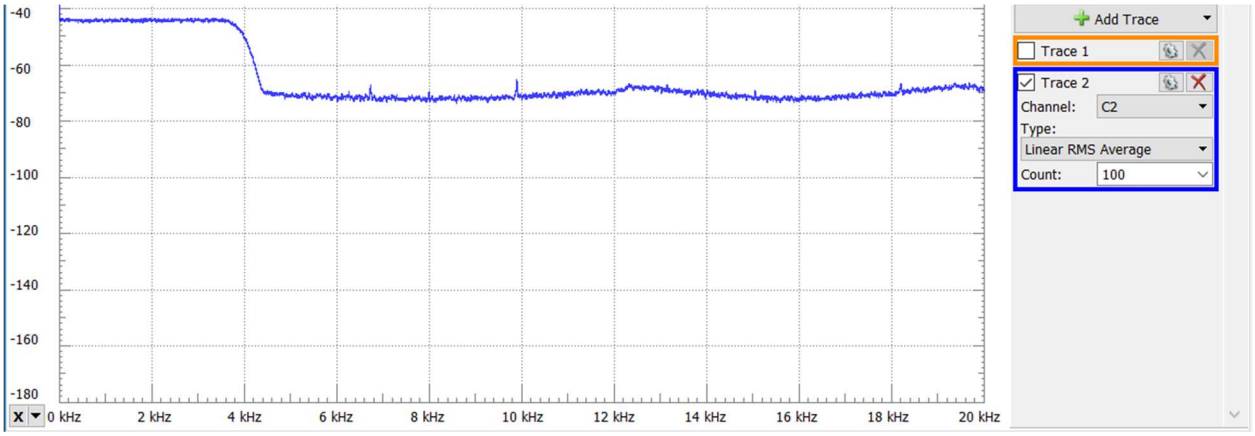

Figure 6 - Frequency domain response captured with fs = 8kHz

As expected, the filter filtered out all frequencies that were above a frequency of about 4 kHz. This should be expected, as the Nyquist Theorem states that aliasing will occur is frequencies more than twice the sampling frequency are recorded. By filtering out all frequencies that are greater than 4 kHz, the system is able to avoid aliasing when reconstructing the signal. If signals greater than 4 kHz were recorded, aliasing would occur, and it would not be possible to reconstruct the original signal from the existing data. It should be noted that the anti-aliasing filter is not perfect, and thus some signal greater than 4 kHz may be recorded, thus introducing

some error into the reconstructed filter. In a real-world system, it is difficult and expensive to eliminate all sources of minor error and build highly defined filters.

4) Magnitude Frequency Response of DAC Reconstruction Filter – fs = 8 kHz

The prbs_intr.c program was run again, this time with a sampling frequency of 48 kHz. The results were captured and are shown below.

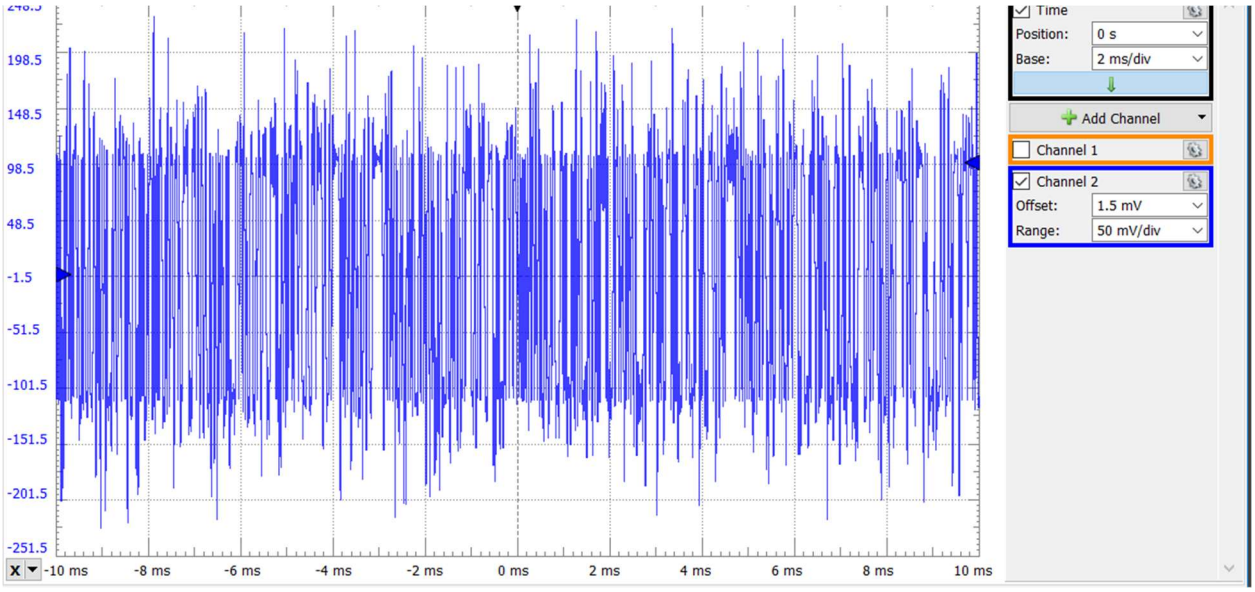

Figure 7 - Time domain response captured with fs = 48 kHz

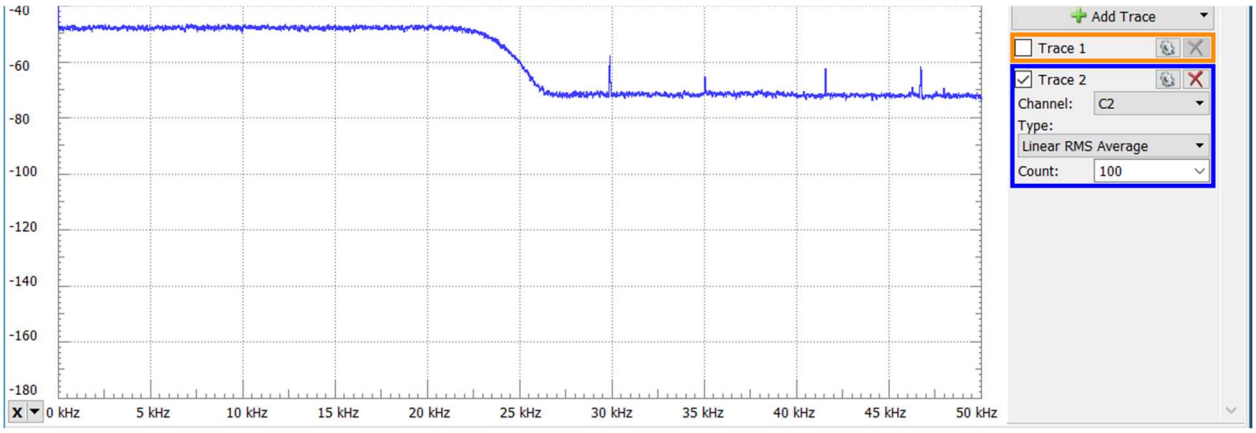

Figure 8 - Frequency domain response captured with fs = 48kHz

The behavior captured when the sampling frequency was increased to 48 kHz was similar to what was observed when the sampling frequency was 8 kHz. Again, to satisfy the Nyquist Theorem and allow accurate reconstruction of the original signal, all frequencies greater than fs/2 were attenuated so that they would not introduce aliasing into the system. This time, instead of the cutoff frequency being roughly 4 kHz, the anti-aliasing filter is eliminating frequencies that are greater than roughly 24 kHz, which is half of the sampling frequency.

5) Step Response of Anti-Alias Filter

A step response signal was generated using the analog discovery kit and the samples were plotted in matlab. The results are shown below.

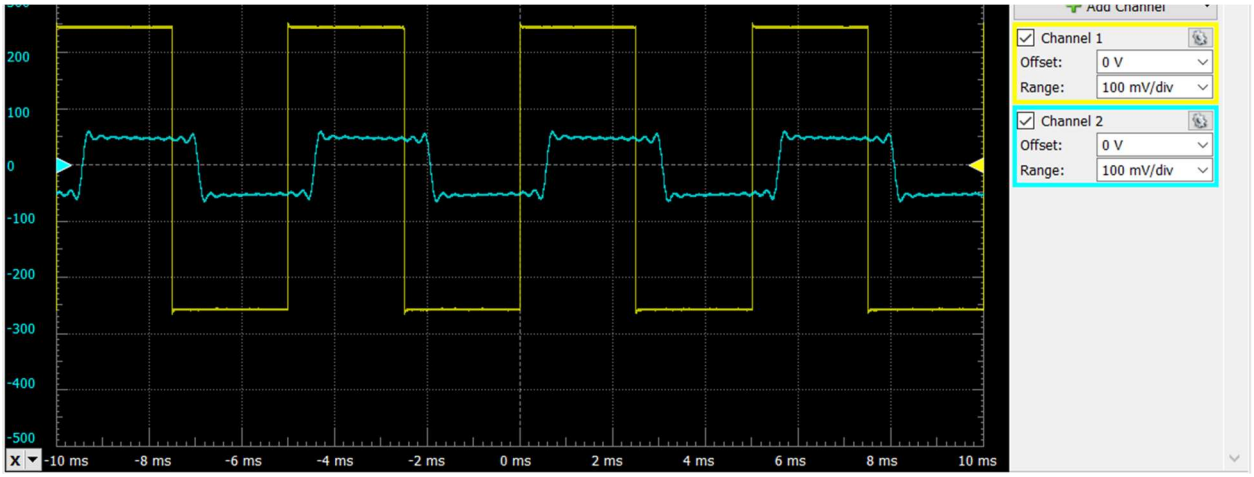

Figure 9 - The generated signal is shown in yellow with the reconstructed signal shown in blue.

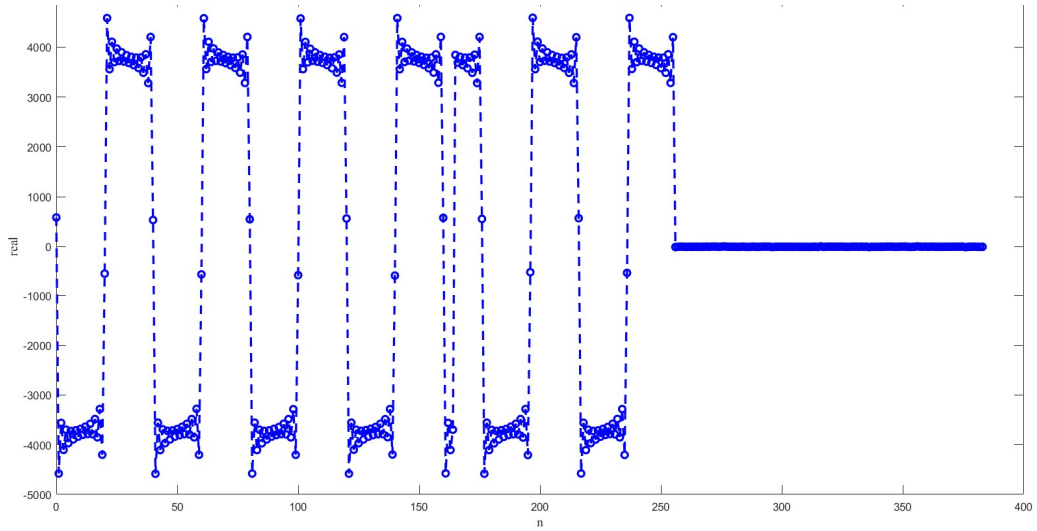

Figure 10 - Sampled signals of square input signal plotted in Matlab

The results of plotting the sampled signals appear to be very similar to the step response recorded with the DAC reconstruction filter. Both samples show imperfect square waves with several small peaks and valleys near the edges of the squares. One of the difference between the two is the samples taken and plotted in Matlab appear to have rising and decreasing linear slopes between the edges of the square. The response captured from the DAC reconstruction filter was relatively flat compared to the Matlab plot. The amplitude of the sampled signal also appears to have had a gain applied to it. This is most likely due to restrictions put in place for safety reasons.

6) Impulse Response of Anti-Alias Filter

The impulse response of the anti-alias filter was observed by reducing the duty cycle of the generated signal to 1%. Changing the duty cycle to 1% simulated an impulse. The results are recorded below.

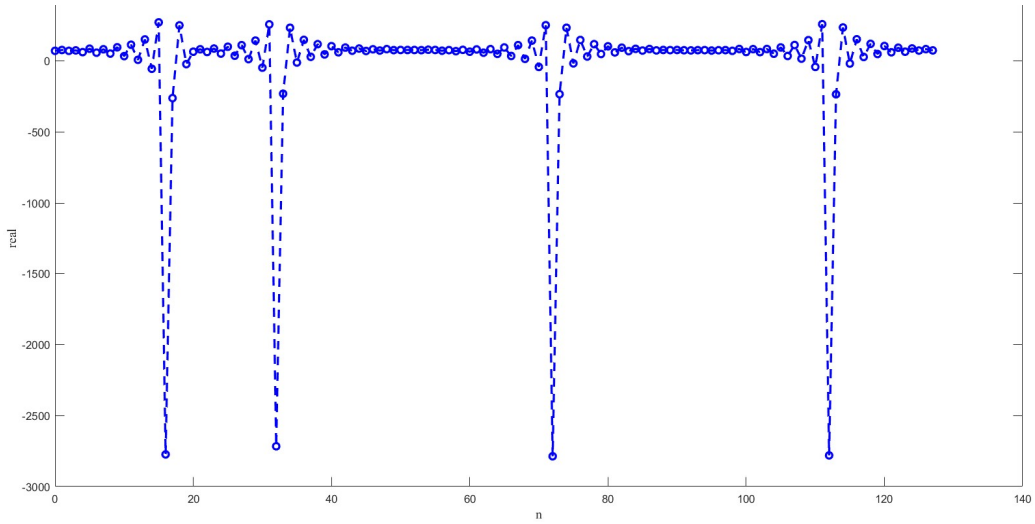

Figure 11 - Sampled signals of impulse input signal plotted in Matlab

At first glance, it may appear that the signal was not properly sampled, however the Nyquist Theorem ensures that this is not the case. With a sampling frequency of 8 kHz and a maximum signal frequency of 200 Hz, the Nyquist Theorem is easily satisfied. It is possible the impulse signals are just not being sampled at a frequency which catches the impulse responses in the negative y direction. The results captured in this step seem to closely resemble the results that were captured using the DAC reconstruction filter in the beginning of the lab. In particular, there are imperfect spikes with several smaller peaks and valleys nearby. These imperfections should be expected when using a physical anti-aliasing filter. The impulse response is made of all frequencies, but the anti-aliasing removes higher frequencies to prevent aliasing, which causes the imperfect sampled impulse response. It should be noted that the two leftmost spikes are

not spaced correctly, but this is not an error in sampling. This is a consequence of using a ring buffer being to store the samples of the signal.

7) Characteristics of WM8731 antialiasing filter

For each of the sinusoids with differing frequencies below, the sampled signal was captured using a sampling frequency of 8 kHz. Thus, the effects of aliasing should be seen in any sinusoid with a frequency greater than or equal to 4 kHz. The captured sampled signals are shown below.

- fs = 8000 Hz | f = 100 Hz

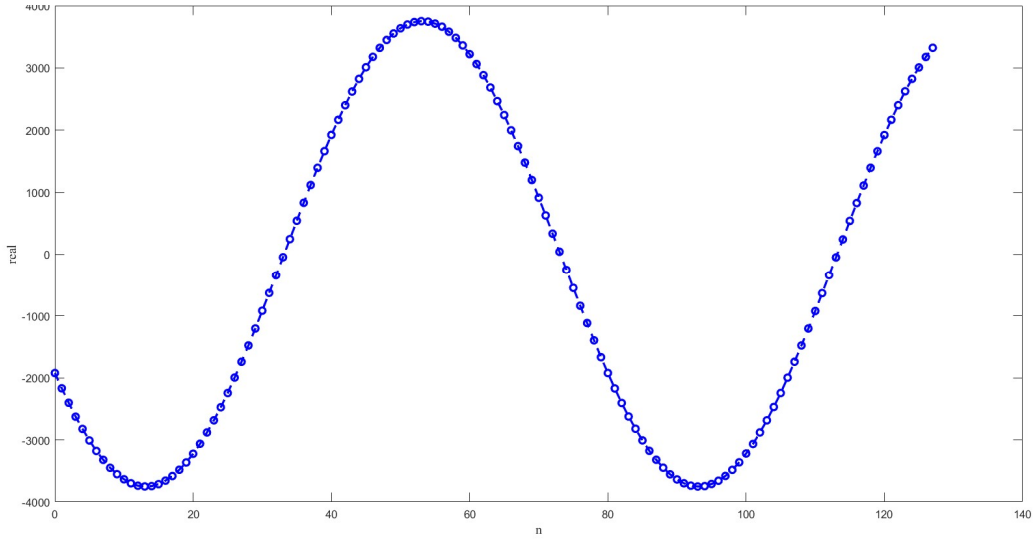

Figure 12 - Matlab plot of sampled 100 Hz sinusoid

As expected, the 100 Hz sinusoid can be correctly sampled without any effects of aliasing. With a sampling frequency much greater than the frequency of the sinusoid, its detail is captured in great detail. Aliasing does not occur in this situation since 2f < fs, and thus, the original signal can be reconstructed.

- fs = 8000 Hz | f = 200 Hz

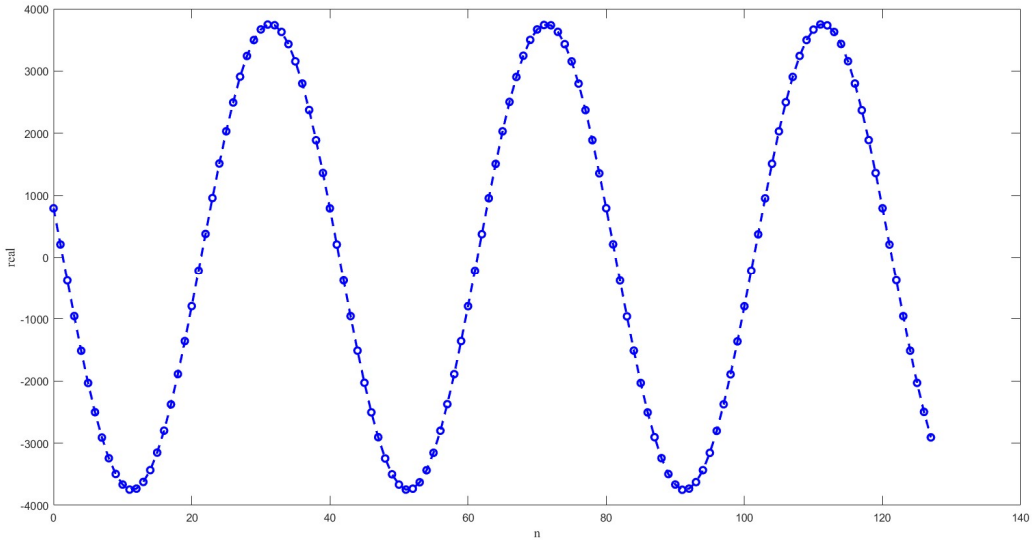

Figure 13 - Matlab plot of sampled 200 Hz sinusoid

Again, as expected, aliasing cannot occur here since the sampling frequency is more than twice the frequency of the signal. Since this is the case, the plot will exhibit any effects of the anti-aliasing filter and can be easily reconstructed into the original signal. This plot shows a little bit less detail than the previous since the difference between the sampling rate and the frequency of the signal is a little smaller.

- fs = 8000 Hz | f = 500 Hz

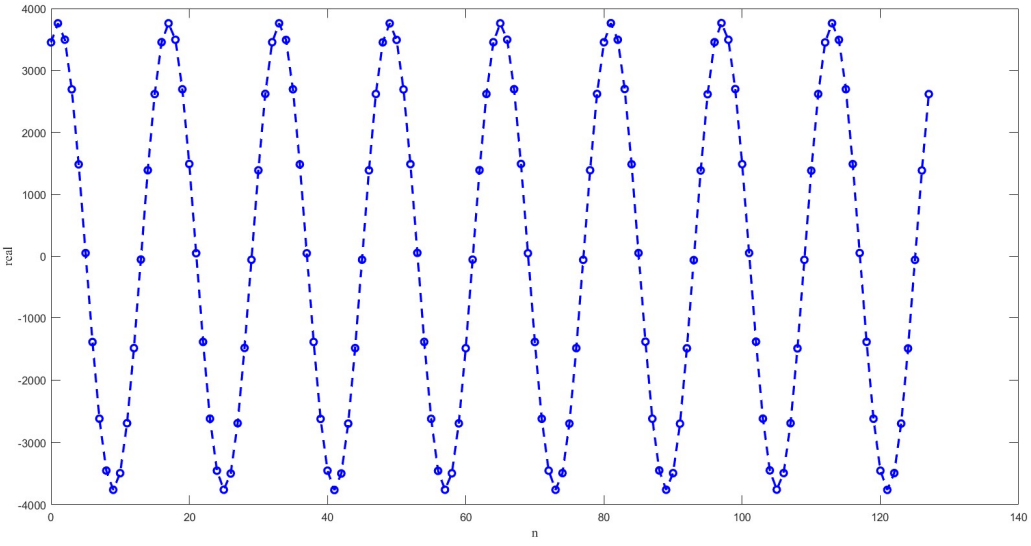

Figure 14 - Matlab plot of sampled 500 Hz sinusoid

Again, as expected, aliasing cannot occur here since the sampling frequency is more than twice the frequency of the signal. Since this is the case, the plot will exhibit any effects of the anti-aliasing filter and can be easily reconstructed into the original signal. This plot shows a little bit less detail than the 100 and 200 Hz signals since the difference between the sampling rate and the frequency of the signal is a little smaller.

- fs = 8000 Hz | f = 1000 Hz

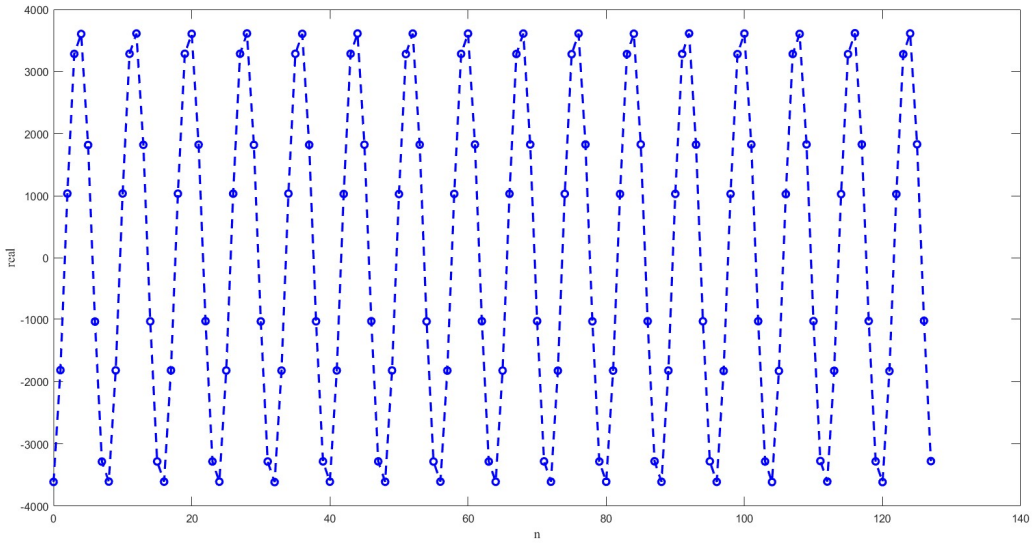

Figure 15 - Matlab plot of sampled 1 kHz sinusoid

Since the sampling frequency is still more than twice the frequency of the sinusoid, the effects of the anti-aliasing filter are not apparent. It is interesting to note that sine detail of the original signal is no longer as definitive when looking at the sampled data. It is easy to see the periodic characteristics, but the peaks and valleys are not as smooth.

- fs = 8000 Hz | f = 2000 Hz

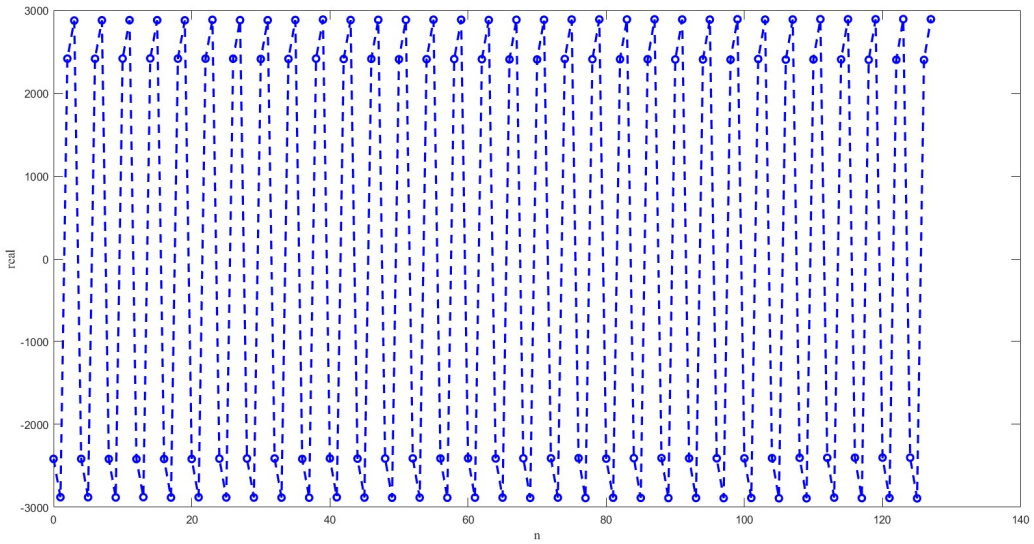

Figure 16 - Matlab plot of sampled 2 kHz sinusoid

At this point, the sampled data is starting to not visually represent the original signal. It is now appearing to be sharp, almost square like cycles. While the signal integrity may appear to be compromised visually, the signal is still preserved and can be reconstructed since the frequency of the sinusoid is less than half of the sampling frequency. Because of this, the effects of the anti-aliasing filter should not be present in the sampled and reconstructed signal.

- fs = 8000 Hz | f = 3500 Hz

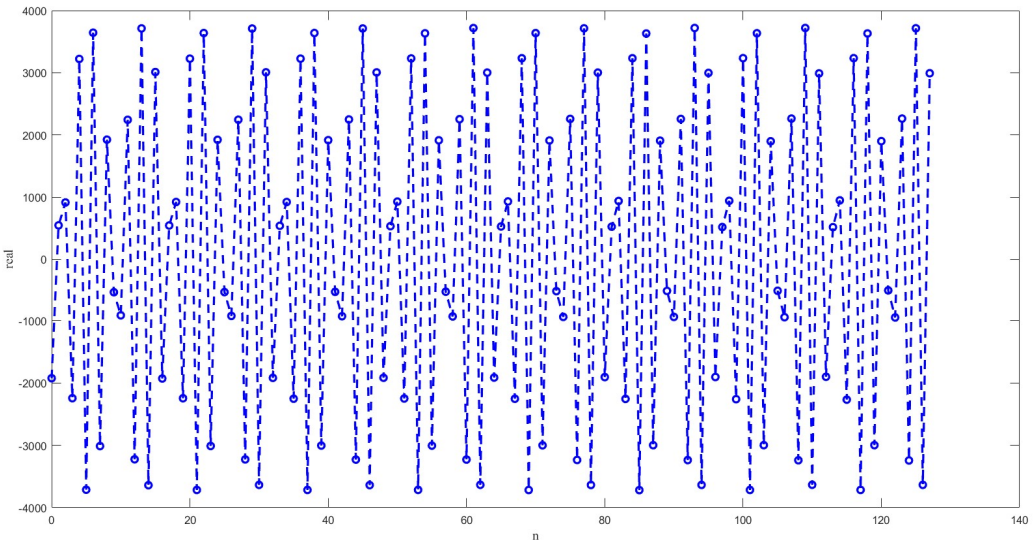

Figure 17 - Matlab plot of sampled 3.5 kHz sinusoid

The sampled data no longer appears to represent the original sinusoidal signal. While this is true, the signal can still be reconstructed, and since the frequency of the sinusoid is less than half the sampling frequency, the anti-aliasing filter should have no effects of the accuracy of the sampled and reconstructed data. Signals with a higher frequency of 3500 Hz will start to approach the frequency where the effects of the anti-aliasing filter will be apparent.

- fs = 8000 Hz | f = 4500 Hz

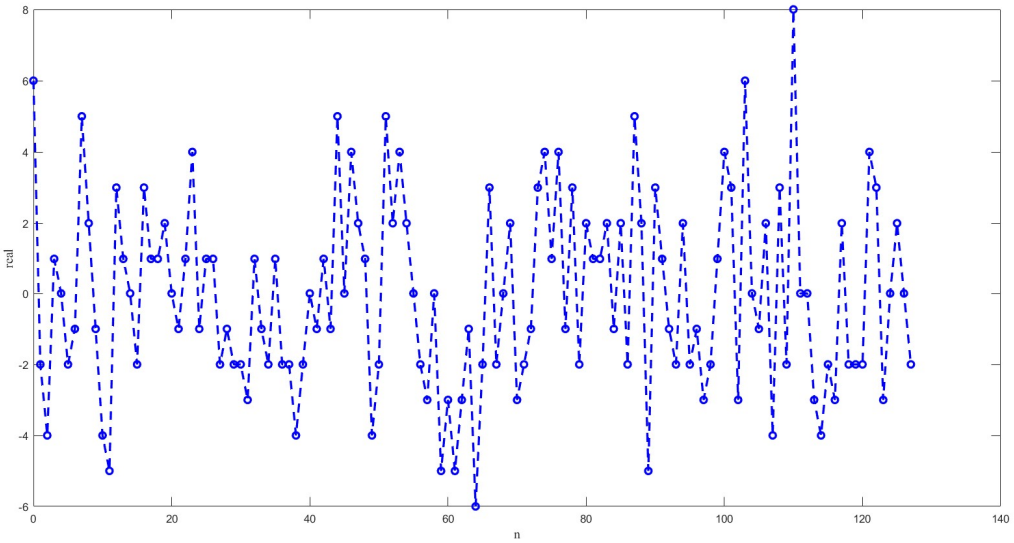

Figure 18 - Matlab plot of 4.5 kHz sinusoid at 8 kHz.

Since the frequency of the sinusoid is not greater than half the sampling frequency, the effects of the anti-aliasing filter are now seen. The anti-aliasing filter is now filtering out the input signal, which results in the sampled data being extremely small. This small magnitude sampled data is most likely due to noise on the wires. Essentially, the original signal has been completely lost since the sampling frequency has not been adjusted. For the anti-aliasing filter to not filter out the original signal, the sampling frequency would need to be adjusted to at least 9 kHz.

- fs = 8000 Hz | f = 5000 Hz

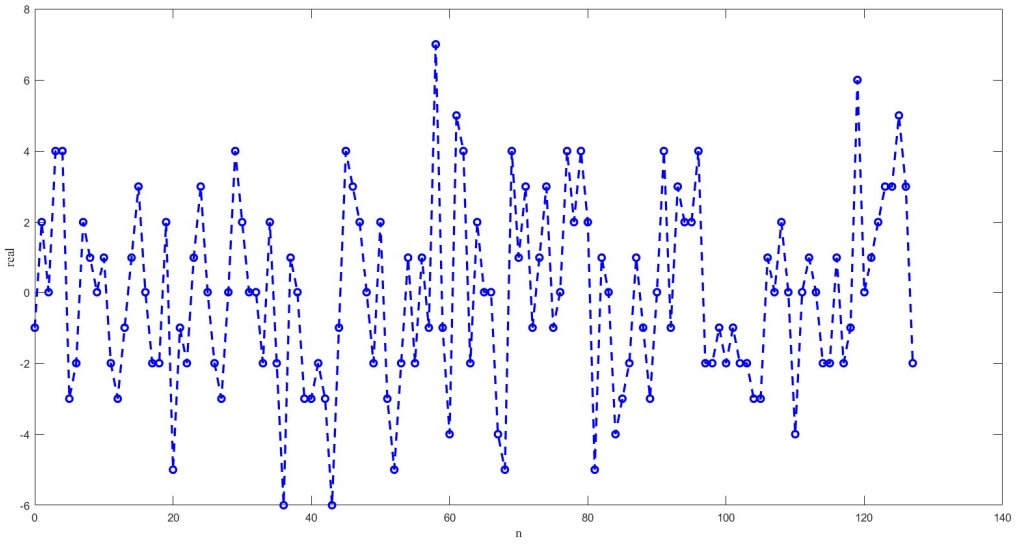

Figure 19 - Matlab plot of 5 kHz sinusoid at 8 kHz.

The same reasoning that was used to explain the results for the 4.5 kHz signal could be used to explain this result as well. Since the frequency of the sinusoid is not less than half of the sampling frequency, the Nyquist Theorem is not satisfied, and the anti-aliasing filter removes the original signal to prevent aliasing. This means that the original frequency is now lost. To correctly sample the sinusoid, the sampling frequency should be increased so that it is greater than 10 kHz.

- fs = 8000 Hz | f = 10000 Hz

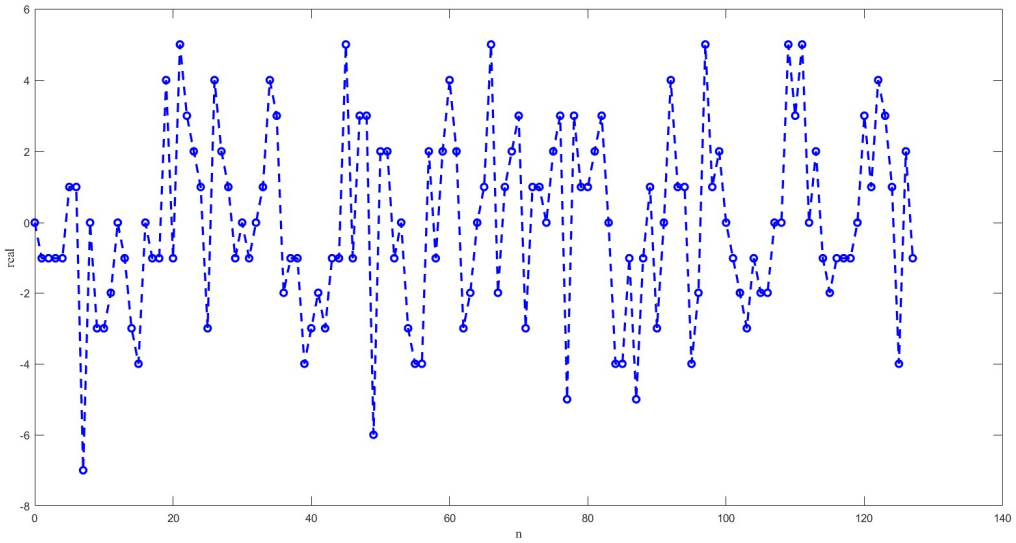

Figure 20 - Matlab plot of 5 kHz sinusoid at 8 kHz.

Mitchell Larson
EE 3221
12/18/17
Lab3

The same phenomenon is occurring in this result as the previous two. The original signal is being filtered out since its frequency is not less than half the sampling frequency. The sampled data is not representative of the original signal. In order to properly capture the 10 kHz sinusoid, the sampling frequency should be increased so that it is greater than 20 kHz. This would prevent the anti-aliasing filter from filtering out the original signal so that it could be reconstructed.