Post-Lab Report

ECEN 489 Lab 1 – Signal Processing Concepts Review

Braxton Wade – Section 501

TA: Haotian Zhao

Introduction

Overall, this lab is an overview of basic signal processing concepts. The lab will cover concepts such as basic digital filters, sampling, and a discrete Fourier transform in simulation. In the physical lab, the sampling of signals will be tested and their discrete forms observed.

Section 1: Digital Filters

Part A:

An example of a FIR digital filter and IIR digital filter can be observed below:

This is primarily because the finite impulse response filter is only dependent on the state of the signal back to a set number of samples before the current output. For example, the output if this specific filter is only dependent on the most recent three samples. However, for the infinite response filter, there is a clear dependency on the past instances of the output signal. When the transfer function is solved for the output, there is a recursive dependence on the past versions of the output.

The plots of the above transfer functions can be seen below:

A graph of a function

AI-generated content may be incorrect.

Figure 1: Transfer function of the IIR filter stated above

This shows the IIR filter to have a pole at 0 and a zero at 0.209 Fs

A graph of a diagram

AI-generated content may be incorrect.

Figure 2: Transfer function of the IIR filter stated above

This shows the zeroes of the FIR filter to have a zero at 0.228 Fs and no poles.

Part B:

Of the two filters listed in the lab manual, the FIR and IIR filters are as follows:

This is because the IIR filter, when solved for the output, has clear dependencies on past versions of the output and is a recursive function. The FIR filter is finite and has a dependency on only the past 5 input samples.

Graphs of the transfer functions can be found below:

A graph of a graph of a graph

AI-generated content may be incorrect.

Figure 3: Transfer function of the FIR filter provided in the lab manual

This graph shows no zeroes and poles at 0.2 and 0.4 Fs respectively. This graph confirms the calculated result with simulation.

A graph with a line

AI-generated content may be incorrect.

Figure 4: Transfer function of the IIR filter provided in the lab manual

This graph shows a zero at 0.25 Fs and poles at 0 and 0.5 Fs. This simulation confirms the calculated results.

Part C:

The IIR filter observed above has the potential to be unstable. This is because the output has the potential to be infinite if the input is exactly on a multiple of Fs. This can be seen in the simulation above where the magnitude of the output becomes infinite at these frequencies.

Contrary to this, the FIR filter output, while having poles, is inherently stable. This is because the output becomes essentially zero at these points. This won’t cause any instability in the output.

Section 2: Sampling

A graph with red and blue lines

AI-generated content may be incorrect.

Figure 5: DT Representation of the 800MHz signal sampled at 500MHz

A graph of a function

AI-generated content may be incorrect.

Figure 6: DT Representation of the 300MHz signal sampled at 500MHz

With both of these samples, it can be observed that the signal is being sampled at a rate that is less than twice within the period of the signal. This means that the signal cannot be reliably constructed back to its original form because it doesn’t meet the Nyquist Criteria. This can be seen in the graphs because, even in this small frame, the signal can be observed to be possibly many different frequencies.

For example, the figure below shows a graph at a different frequency overlaid on this one that matches the sampled data. It can be observed that the discrete points intersect the continuous points of both the 300MHz and the 800MHz signal in figure 7 and that a 200MHz signal will intersect exactly with the 300MHz signal in figure 8.

A graph of a function

AI-generated content may be incorrect.

Figure 7: Re-Simulation of the 800MHz discrete signal at 500MHz with aliasing frequency of 300MHz also plotted continuously

A graph of a function

AI-generated content may be incorrect.

Figure 8: Re-Simulation of the 300MHz discrete signal at 500MHz with aliasing frequency of 200MHz also plotted continuously

Part B:

As a whole, this signal cannot be recovered exactly due to the Nyquist criteria. The resolution isn’t fine enough to sample the signal at least twice each period. Therefore, it is entirely unknown what the signal could’ve been from the sampled data. If the assumption is made that the signal is within the Nyquist frequency, it would’ve been reconstructed as a 300MHz as that’s the aliasing frequency. The same can be said about the 300MHz signal being reconstructed at 200MHz.

The only solution to sample these frequencies accurately would be to increase the sampling frequency to at least double the frequency of the fastest signal. This meets the Nyquist criteria and will allow for accurate reconstruction of the signal.

Part C:

A zero-order hold system is a reconstruction system where a set of discrete signal is reconstructed as close to the continuous version of itself as possible. This is done by taking the current point and holding its value until the next point appears. In an ideal system, the pulse width, W, is equivalent to 1/T where T is the sampling frequency.

The reconstruction of the symbol can be represented by the function:

This creates an instantaneous reconstruction of a sample of data. This can be used to convert any individual point in time and find its value from a given dataset. The equivalent point in the dataset is found by dividing the current time by the ideal pulse width, W. An assumption is made that point 0 is at time 0. It is then rounded up because it is stated that the sampling point is at the end of the pulse width.

This will create a direct interpolation where the sample value is held up until the time it was taken and will then transition instantly to the next sample.

Part D/E:

The signal in question for these parts is a pure sinewave with a frequency of 300MHz. It was sampled at a frequency of 800MHz and then reconstructed using the sinc interpolation method.

A diagram of a signal

AI-generated content may be incorrect.

Figure 9: The original signal of 300MHz, sampled at 800MHz, and reconstructed signal graphed for a sampling beginning at zero and lasting for 9 periods

Mean Squared Error = 0.030215654405303296

A diagram of a signal

AI-generated content may be incorrect.

Figure 10: The original signal of 300MHz, sampled at 800MHz, and reconstructed signal graphed for a sampling beginning at Ts/2 and lasting for 9 periods

Mean Squared Error = 0.008780766528583223

A diagram of a signal

AI-generated content may be incorrect.

Figure 11: The original signal of 300MHz, sampled at 500MHz, and reconstructed signal graphed for a sampling beginning at zero and lasting for 9 periods

Mean Squared Error = 0.9364164738528349

A graph of a signal

AI-generated content may be incorrect.

Figure 12: The original signal of 300MHz, sampled at 500MHz, and reconstructed signal graphed for a sampling beginning at Ts/2 and lasting for 9 periods

Mean Squared Error = 0.9577593259192416

A graph of a signal

AI-generated content may be incorrect.

Figure 13: The original signal of 300MHz, sampled at 1000MHz, and reconstructed signal graphed for a sampling beginning at zero and lasting for 9 periods

Mean Squared Error = 0.006855272551350668

A diagram of a signal

AI-generated content may be incorrect.

Figure 14: The original signal of 300MHz, sampled at 1000MHz, and reconstructed signal graphed for a sampling beginning at Ts/2 and lasting for 9 periods

Mean Squared Error = 0.0008302934263999465

In these cases, there is a clear trend in this sampling. The first observable trend is that the error of the reconstruction decreases as sampling increases. This is partially due to the reconstruction methods. The more points that the reconstruction method has to go on, the closer to the original signal the reconstruction will be. This is especially true for the sampling at 500MHz. The signal of 300MHz is above the Nyquist rate. Therefore, the reconstruction will create a signal at the aliasing frequency of 200MHz. Because of this, the second pattern isn’t observed here.

The second pattern is that the shifted error where the sampling begins at Ts/2 is much less than the error when the sampling begins at zero. This is largely because of the proximity of the samples at the ends to the ends of the continuous sample. With the first sample, one of the points is a full sampling cycle away from the end of the continuous signal. This allows the reconstructed signal to deviate much farther from the ideal signal. However, with the samples starting at Ts/2, the reconstruction only has half a sampling period on each end of the signal to deviate. This allows a much tighter fit to the true signal towards the end of the signal.

Section 3: Discrete Fourier Transform

Part A:

A graph of a signal

AI-generated content may be incorrect.

Figure 15: A 50 Point DFT of a 2MHz signal sampled at 5MHz

Part B:

A graph of a signal

AI-generated content may be incorrect.

Figure 16: A 50 Point DFT of a 200MHz and 400MHz signal sampled at 1GHz

Yes, the 200MHz and the 400MHz parts of the frequency can be easily picked out of this FFT. While it has a wide range due to the resolution, it can be easily identified as being centered at the frequencies of interest.

Part C:

A graph of a signal

AI-generated content may be incorrect.

Figure 17: A 50 Point DFT of a 200MHz and 400MHz signal sampled at 500MHz

In this figure, while the 200MHz signal is clearly sampled at 500MHz due to it being below the Nyquist frequency, the 400MHz signal has been aliased down to 100MHz. This is to be expected with the change in sampling rate and is mathematically supported.

Part D:

A Blackman window was applied to each of the waveforms. However, because the Fourier transforms were already ideal representations, it didn’t affect the output in a positive way.

A graph with a line

AI-generated content may be incorrect.  
Figure 18: A 50 Point DFT of a 2MHz signal sampled at 5MHz with a Blackman Window

A graph of a signal

AI-generated content may be incorrect.

Figure 19: A 50 Point DFT of a 200MHz and 400MHz signal sampled at 1GHz with a Blackman Window

A graph of a signal

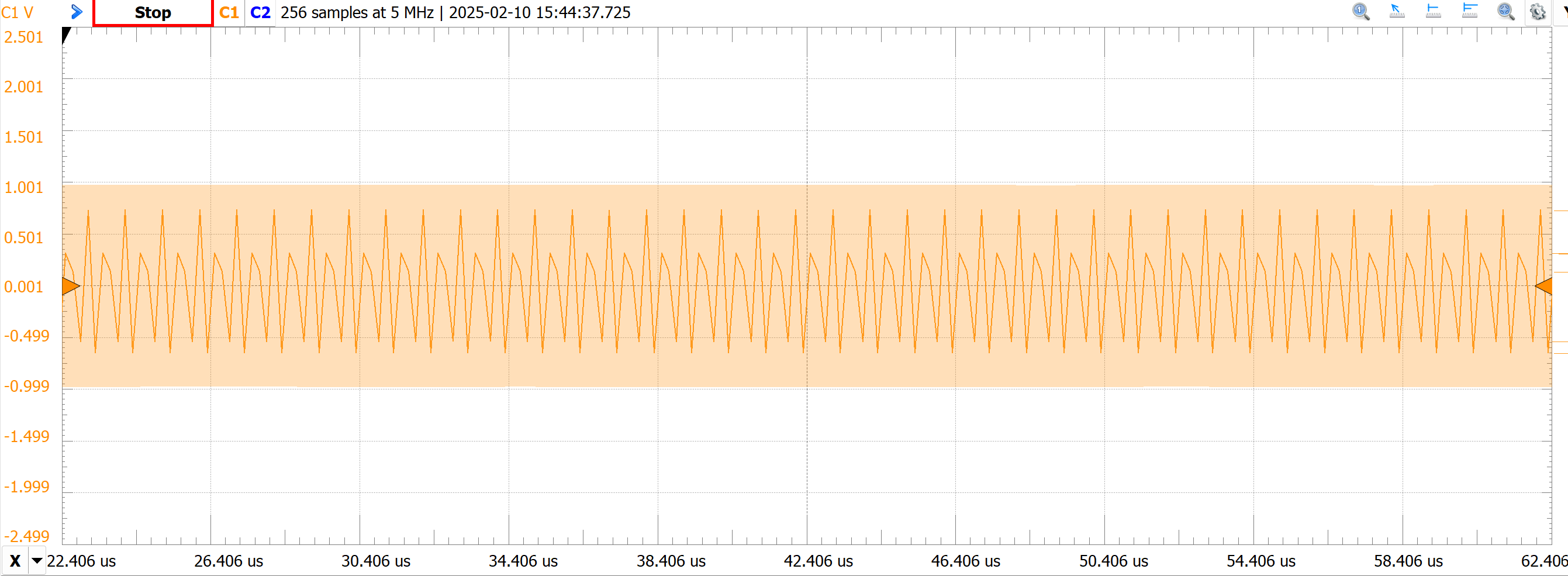
AI-generated content may be incorrect.

Figure 20: A 50 Point DFT of a 200MHz and 400MHz signal sampled at 500MHz with a Blackman Window

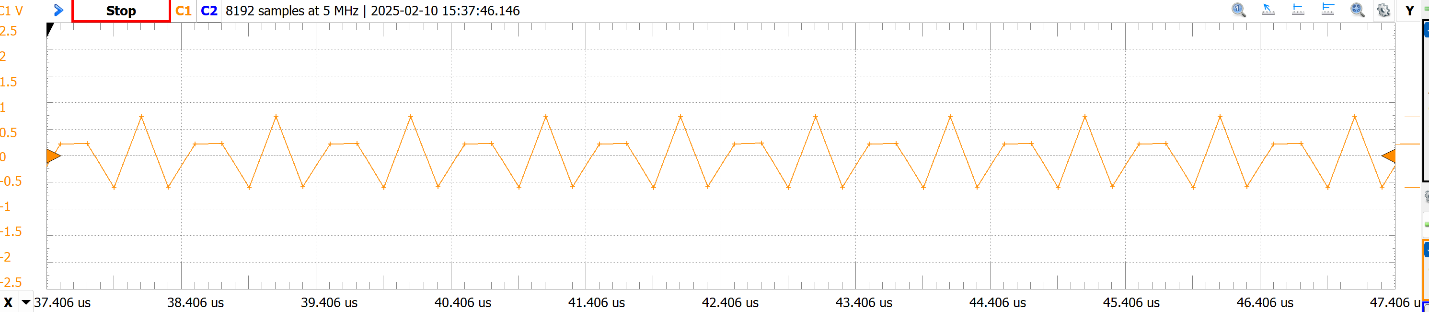
The primary difference observed here was that the Blackman window caused the signal to spread into other bins aside from its ideal position. The only reason that this didn’t offer an improvement was because the signals fell exactly on a DFT bin in each of these cases. If the frequencies fell between bins, the Blackman window would help prevent leaking of the signal from bin to bin.

Section 4: Lab Measurements

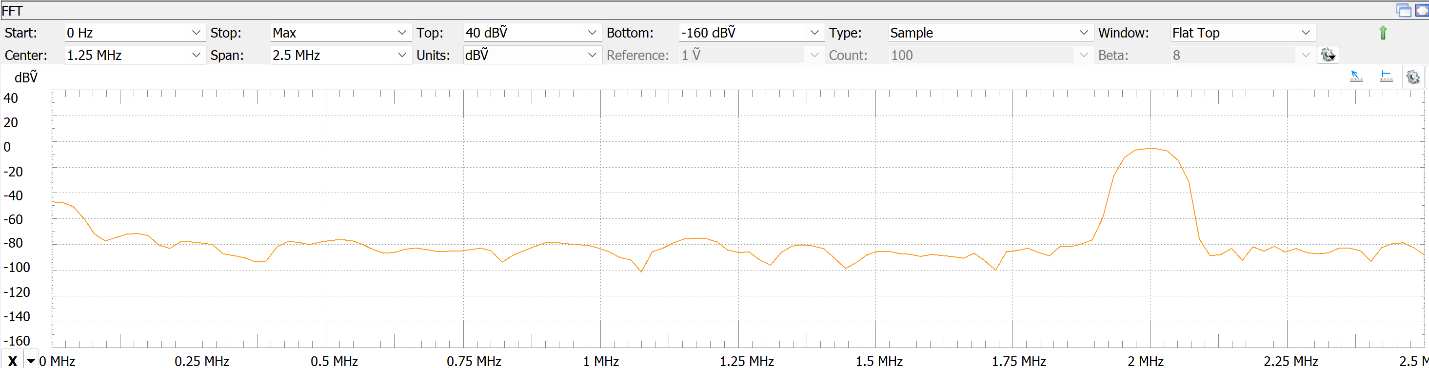
This lab portion was done with a perfect sinewave input signal at 2MHz



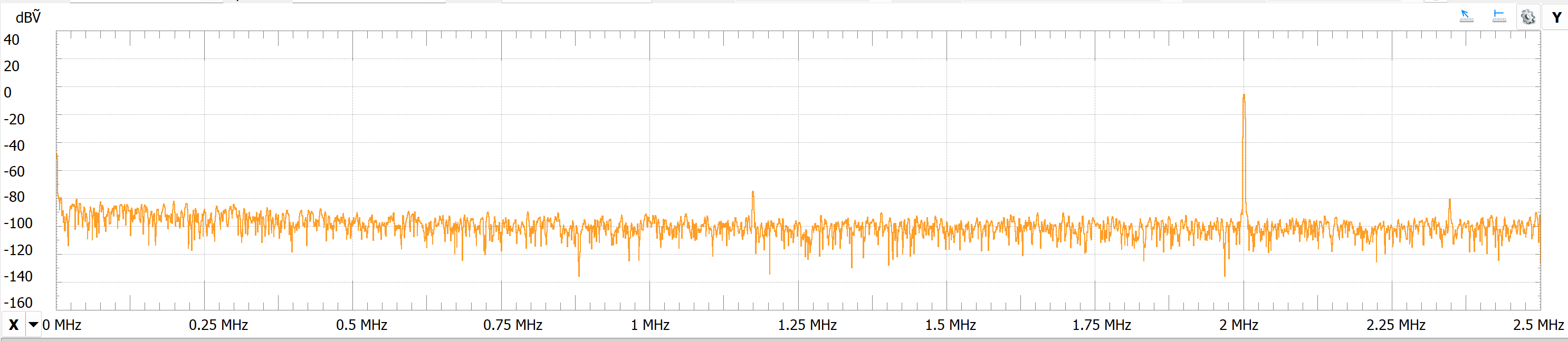
Measurement 1: A screenshot of the scope waveform at 5MHz and 256 samples



Measurement 2: A screenshot of the scope waveform at 5MHz and 8k samples



Measurement 3: A screenshot of the scope DFT at 5MHz and 256 samples

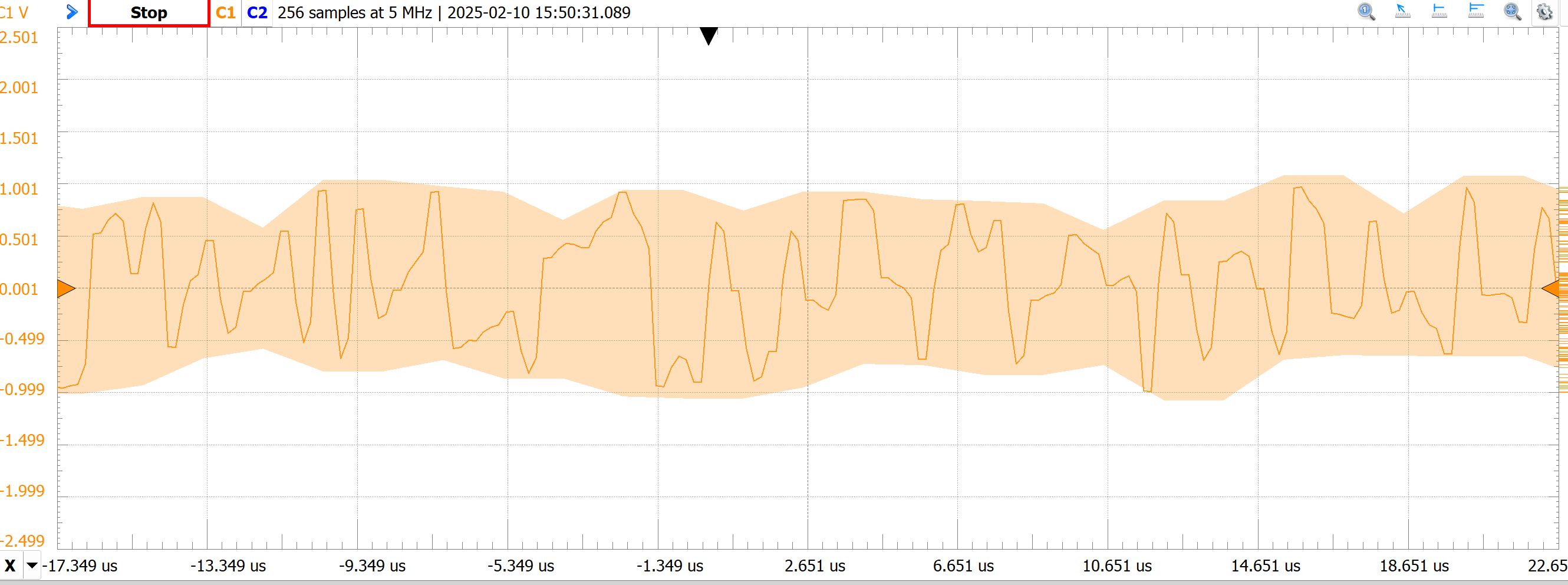


Measurement 4: A screenshot of the scope DFT at 5MHz and 8k samples

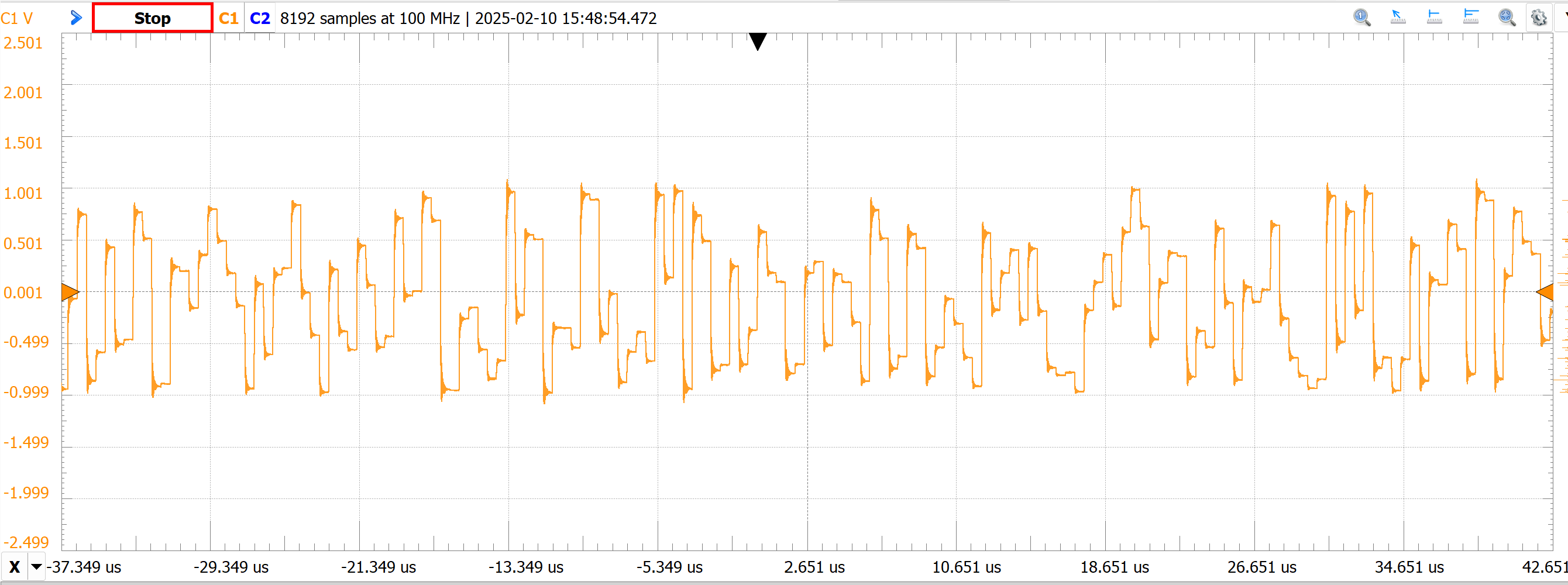
For this lab, the requested resolution was a total of 256 samples across the entire spectrum. I also included a higher resolution of samples to create a more detailed version of the DFT at the same sampling frequency.

In this DFT, the input is a pure sinewave. This can be observed because there is a single spike, nearly reaching 0dB, at the frequency of the sinewave. The only other significant spike in the DFT is at zero and that is likely because of any DC bias that may be present.

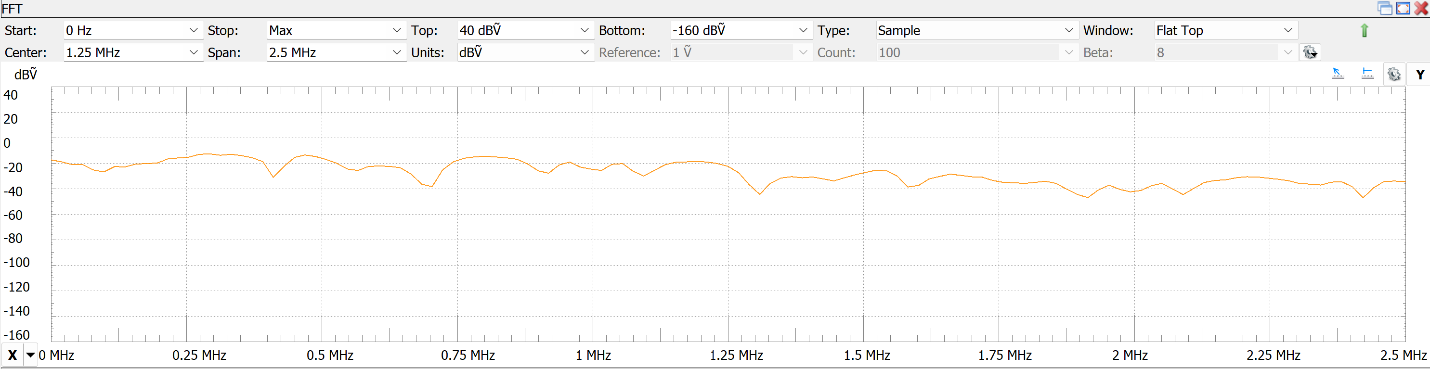
This next portion of the lab was done with a noise signal at the same frequency



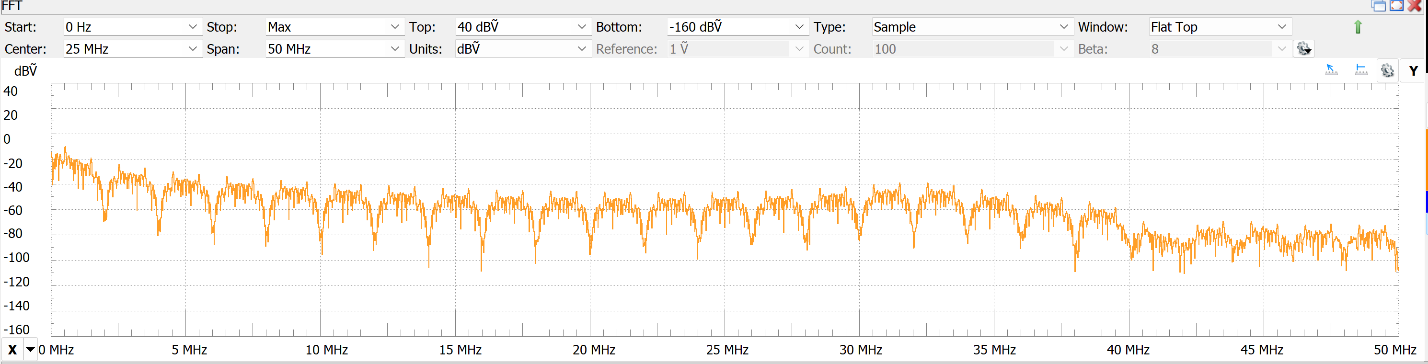
Measurement 5: A screenshot of the scope waveform at 5MHz and 256 samples



Measurement 6: A screenshot of the scope waveform at 100MHz and 8k samples



Measurement 7: A screenshot of the scope DFT at 5MHz and 256 samples



Measurement 8: A screenshot of the scope DFT at 100MHz and 8k samples

In the 5MHz resolution scope measurement, the noise appears to have a uniform presence across all frequencies. This is what we would expect with a signal of pure noise. However, this changes when the resolution is increased. The DFT at 100MHz shows a clear dip at each multiple of the programmed noise frequency.

Section 5: Code Created for Simulations

A screenshot of a computer program

AI-generated content may be incorrect.

Code 1: A function that will plot the transfer function of any filter if given the coefficients and title of graph

A screenshot of a computer program

AI-generated content may be incorrect.

Code 2: The inputs using the transfer function generator above for part 1

A screen shot of a computer program

AI-generated content may be incorrect.

Code 3: The code used to graph and sample the sinewave for part 2a

A screen shot of a computer screen

AI-generated content may be incorrect.

Code 4: Signal reconstruction and mean squared error code

A screen shot of a computer program

AI-generated content may be incorrect.

Code 5: FFT Calculating and plotting code

A screenshot of a computer program

AI-generated content may be incorrect.

Code 6: FFT with Blackman window