Week 5 Research Report

Adam Kamrath

July 2023

1 Background

This report details the progress made during the fifth week of the ongoing USRP research project. In the previous week, a significant milestone was achieved as we successfully received and decoded a signal from the ISC-MR102-USB card reader.

2 Introduction

This week marked significant progress in our understanding of filters and decoding techniques, specifically aimed at the ISC-MR102-USB card reader signals. The main objective was to efficiently receive and decipher signals emitted by ISO15693 cards, which proved to be difficult.

3 Methods

The focus for the first part of this week was again the ISC-MR102-USB card reader signals. It was observed that when the signal is unfiltered, the binary sequence becomes embedded within a sine wave with a frequency of 166.66 Hz. This characteristic complicates the decoding process of the binary sequence, as depicted in Figure 1 on page 2.

To simplify the decoding process, a filter was employed to modify the signal. Since the sine wave had a frequency of 166.66 Hz, the filter needed to eliminate this frequency. Considering that the binary sequence operates at a higher frequency than the sine wave, a high-pass filter was chosen as the solution. To create the high-pass filter, the Digital Signal Processing add-on for Matlab was utilized.

The filter parameters were set as follows: a sampling rate of 200,000 Hz, a stop frequency of 500 Hz, a pass band frequency of 1000 Hz, and an order of 50. Implementing this filter significantly facilitated the subsequent decoding and analysis of the data. The resulting signal, post-filter application, can be observed in Figure 2 on page 3. The MATLAB code for the high-pass filter from Code 1 on page 3.

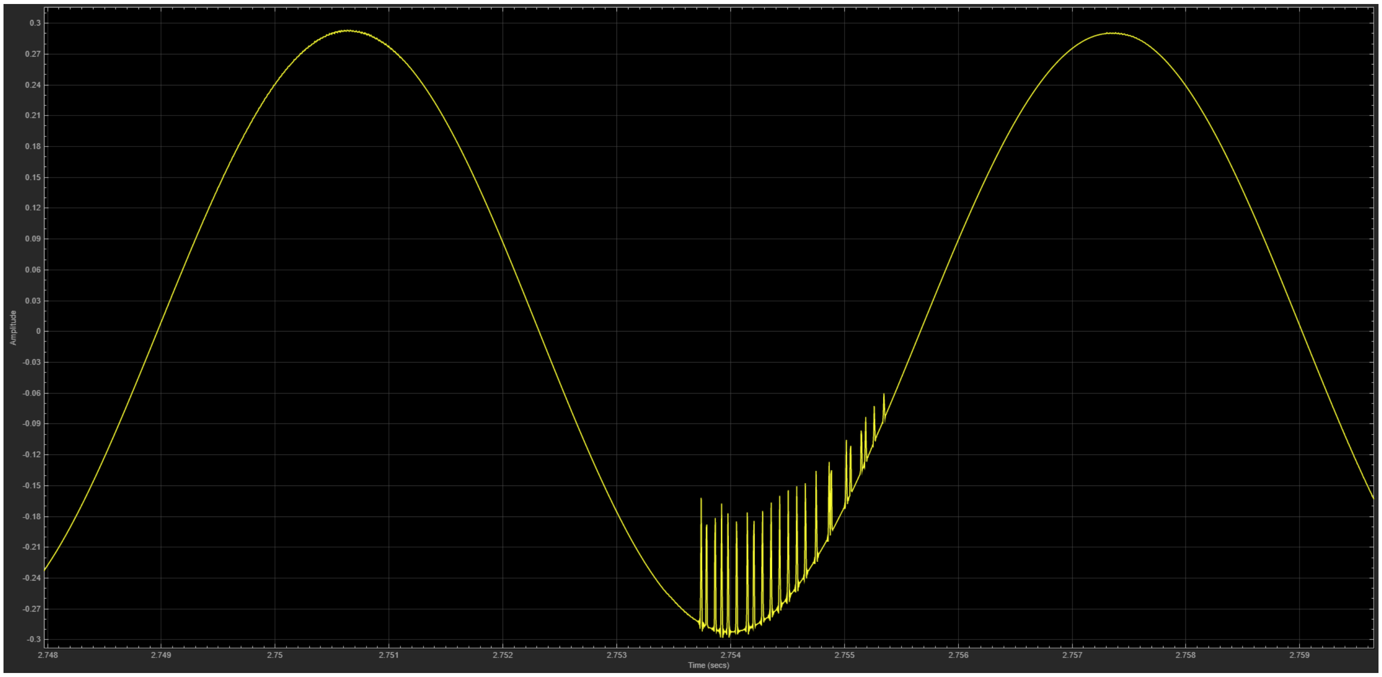

Figure 1: Unfiltered Inventory Request

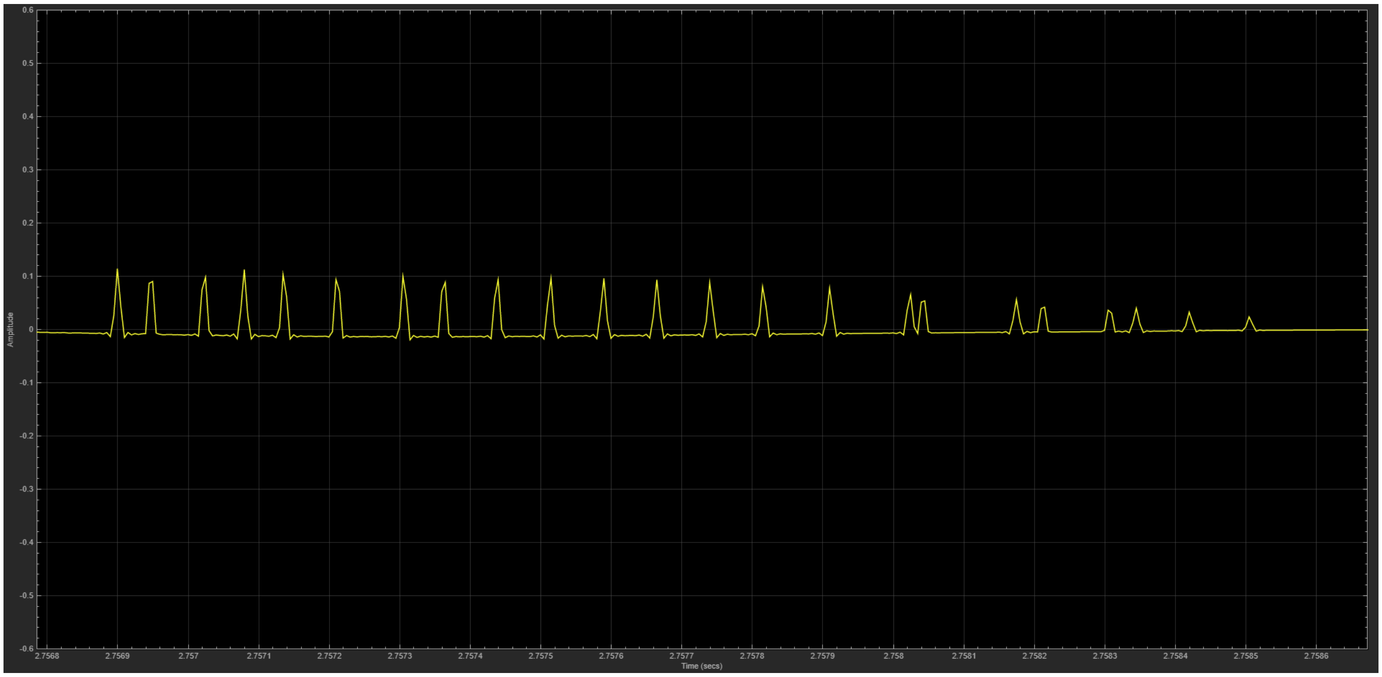

Figure 2: Filtered Inventory Request

Code 1: Initial High-pass Filter Code

```
sampling_rate = 200000;
highpass = dsp.HighpassFilter(StopbandFrequency=500, ...
    SampleRate=sampling_rate, ...
    PassbandFrequency= 1000);
```

Upon investigation, it was discovered that the USRP receiver might exhibit a frequency offset from the intended center frequency. This means that when the center frequency was set to 13.56 MHz, the receiver might not receive the signal precisely at that frequency due to the USRP's internal clock. To address this issue, frequency calibration was employed to mitigate the offset.

Initially, the frequency offset calibration example provided here was attempted. However, this approach proved ineffective in resolving the problem. Consequently, an alternative solution was devised using two separate Simulink programs.

The first Simulink program utilized a USRP device to transmit a 100 Hz sine wave precisely over the 13.56 MHz frequency. In the second program, the transmitted sine wave was received over the same frequency. A spectrum analyzer was then employed to determine the actual frequency at which the received sine wave was observed. This approach facilitated accurate evaluation of the received frequency, enabling further analysis of the frequency offset issue.

During the latter part of the week, the focus shifted towards the challenging task of receiving and decoding the signal transmitted by an ISO15693 card. This process presented significant difficulties. To specifically target the frequency of the card's signal, a band-pass filter was implemented. According to the ISO15693 document, cards utilizing a single sub-carrier and a high data rate emit pulses at 423.75 KHz. To isolate these pulses, a band-pass filter was designed to allow signals within the range of 400 KHz to 450 KHz. The MATLAB code for implementing the band-pass filter can be found in Code 2. Additionally, the graph illustrating the characteristics of the filter is presented in Figure 3.

Code 2: Band-pass Filter

```
bandpassSpecs = fdesign.bandpass('N,Fst1,Fp1,Fp2,Fst2,
   C',300,4e5,4.25e5,4.5e5,4.75e5,2e6);
bandpassSpecs.Stopband1Constrained = true;
bandpassSpecs.Astop1 = 60;
bandpassSpecs.Stopband2Constrained = true;
bandpassSpecs.Astop2 = 60;

BandpassFilter = design(bandpassSpecs, 'Systemobject',
    true);
fvtool(BandpassFilter)
measure(BandpassFilter)
release(BandpassFilter);
```

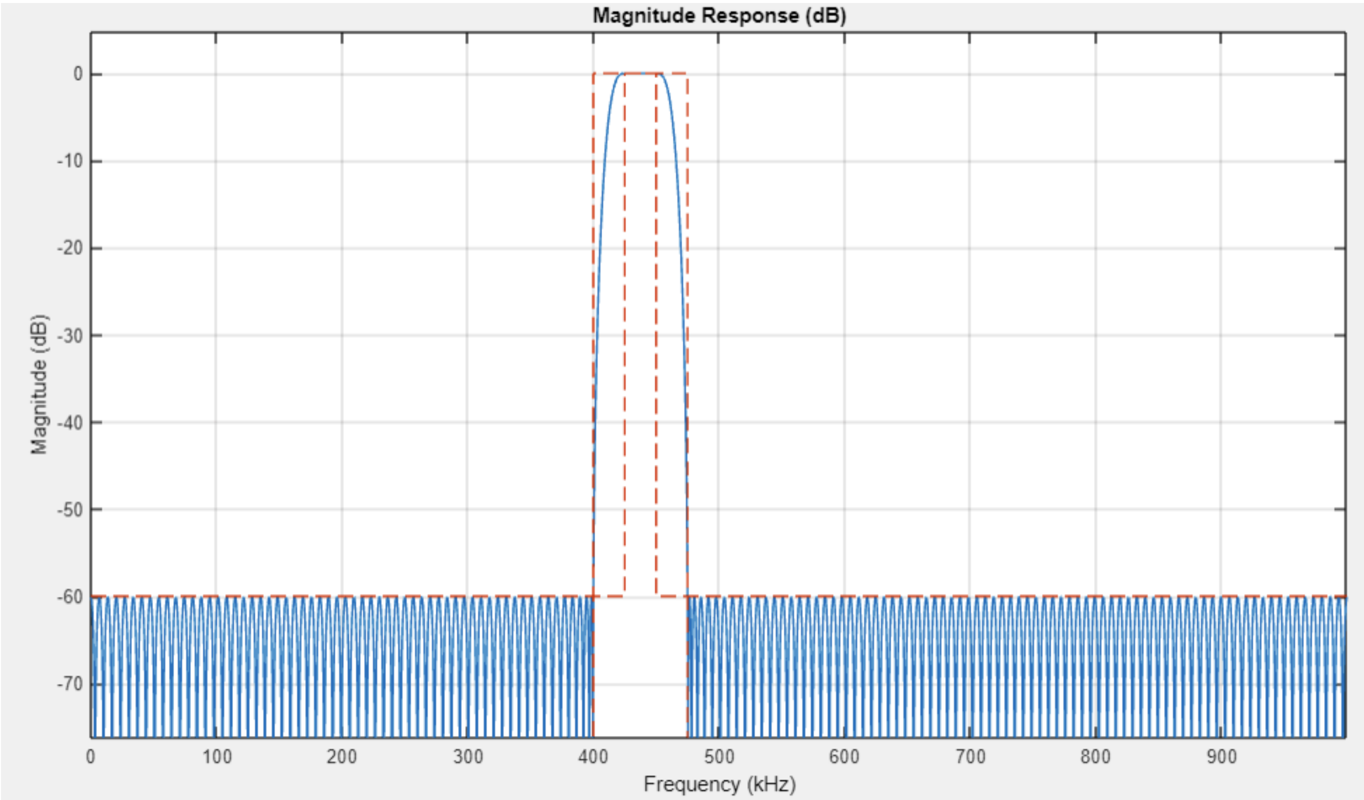

Figure 3: Band-pass Filter

4 Discussion of Results

In regards to the ISC-MR102-USB card reader signals, Figure 2 demonstrates that even after applying a filter, the readability of the data remains challenging due to sharp pulses, making it difficult to discern the signal's starting and ending points. To address this issue, a higher sample rate of 2 MHz was employed during data capture and analysis. This increased sample rate provided a more detailed representation of the signal's actual characteristics. Consequently, modifications had to be made to the high-pass filter as well, which is explained in Code 3 on page 6. For a visual depiction of the signal post high sampling rate and altered high-pass filter, refer to Figure 4 on page 6.

Code 2: Altered High-pass Filter Code

```
Fs = 2000000;
filtertype = 'FIR';
Fpass = 10000;
Fstop = 5000;
Rp = 0.1;
Astop = 60;
HighpassFilter = dsp.HighpassFilter(SampleRate=Fs,...
                                    FilterType=filtertype,...
                                    PassbandFrequency=Fpass,...
                                    StopbandFrequency=Fstop,...
                                    PassbandRipple=Rp,...
                                    StopbandAttenuation=Astop);
```

Figure 4: Higher Sampled Filtered Request

Regarding the frequency offset calibration, the spectrum analysis depicted in Figure 5 reveals that the receiver is receiving the signal at a frequency 20 kHz higher than the intended center frequency of 13.56 MHz. To rectify this issue, the center frequency of the receiver can be adjusted to 13.58 MHz, ensuring that the correct frequency is received. The spectrum analysis after applying the offset can be observed in Figure 6, demonstrating the successful correction of the frequency offset.

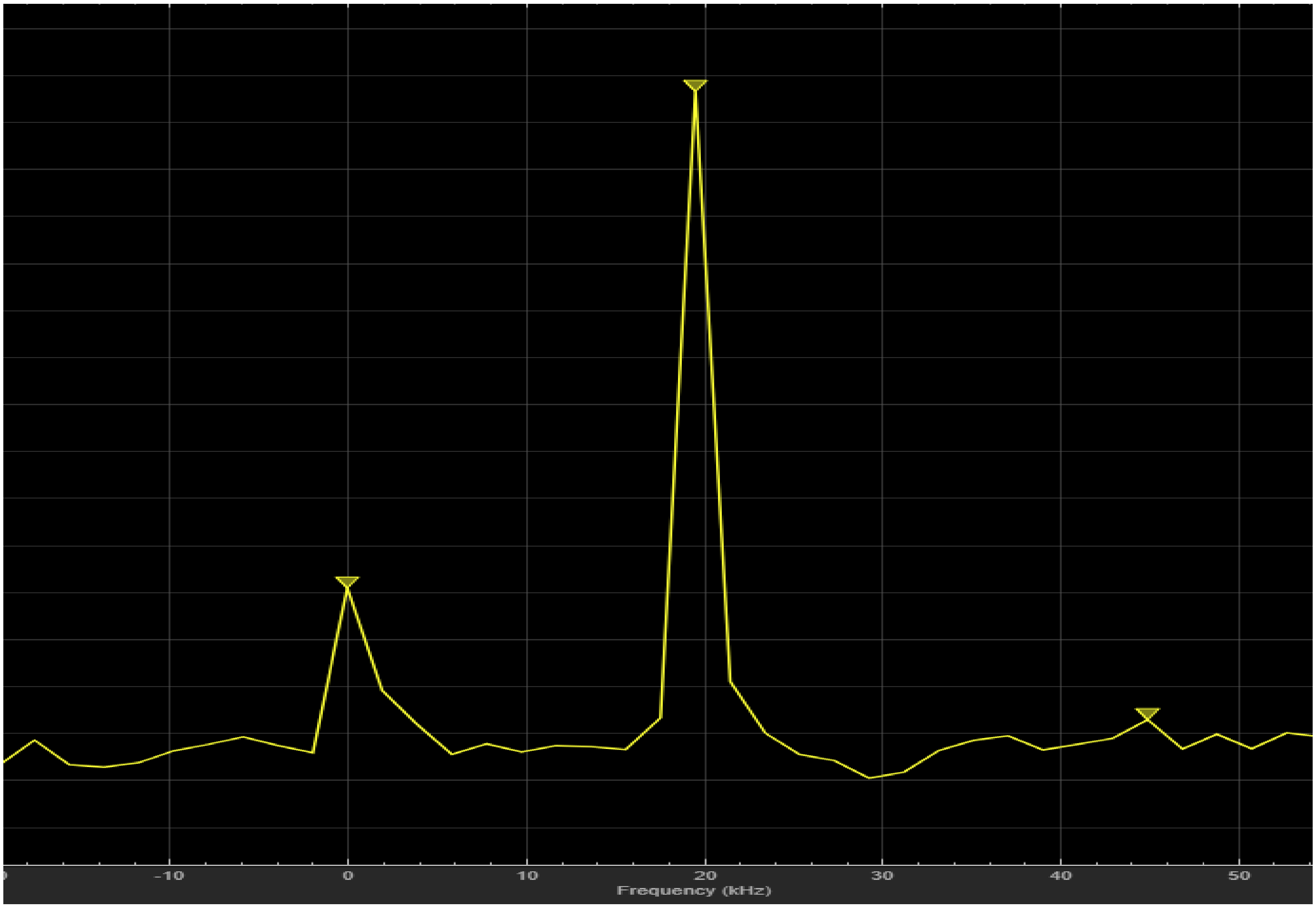

Figure 5: Frequency Offset 13.56 MHz

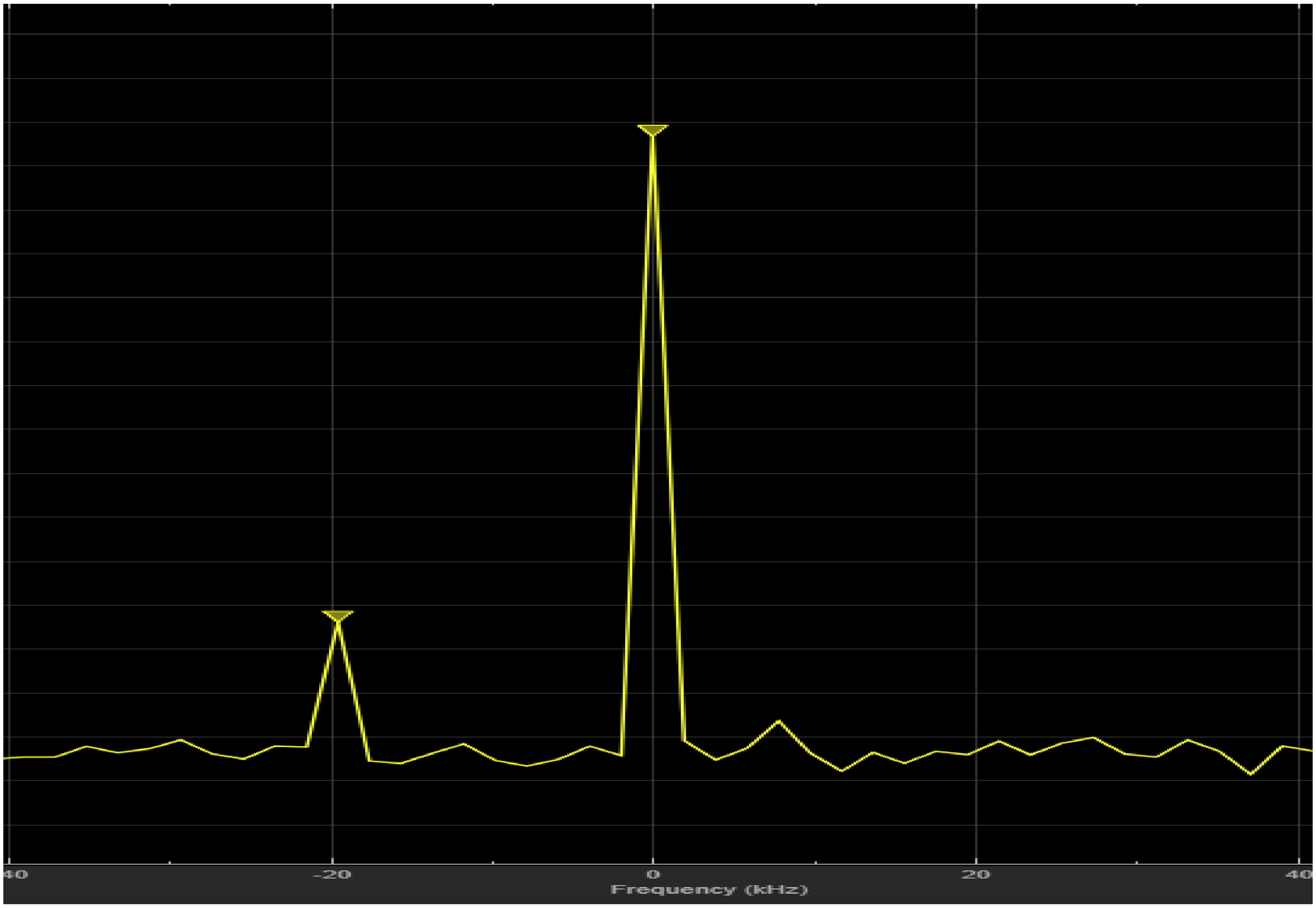

Figure 6: Fixed Frequency Offset 13.58 MHz

Despite the apparent success of the frequency offset calibration for the sine wave, it failed to yield the desired results when receiving the signal from the ISC-MR102-USB card reader. In this scenario, the pulses embedded within the 166.66 MHz sine wave were absent. Figure 7 depicts the received signal at the center frequency of 13.56 MHz, while Figure 8 represents the received signal at 13.58 MHz. A clear distinction can be observed: the 13.56 MHz signal exhibits distinct pulses that contain the binary sequence, whereas the 13.58 MHz signal lacks this crucial information. It is plausible that during the frequency offset calibration, the USRP may have transmitted at a frequency 20 kHz higher than the center frequency, while the receiving component used the correct center frequency. If this were the case, the frequency offset adjustment would be unnecessary.

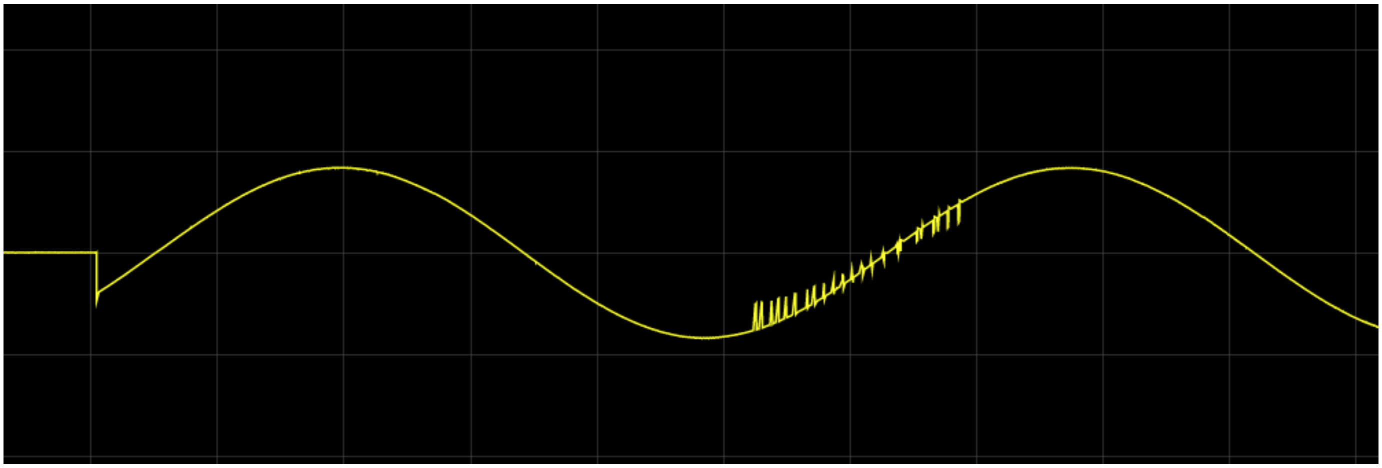

Figure 7: Card Signal at 13.56 MHz

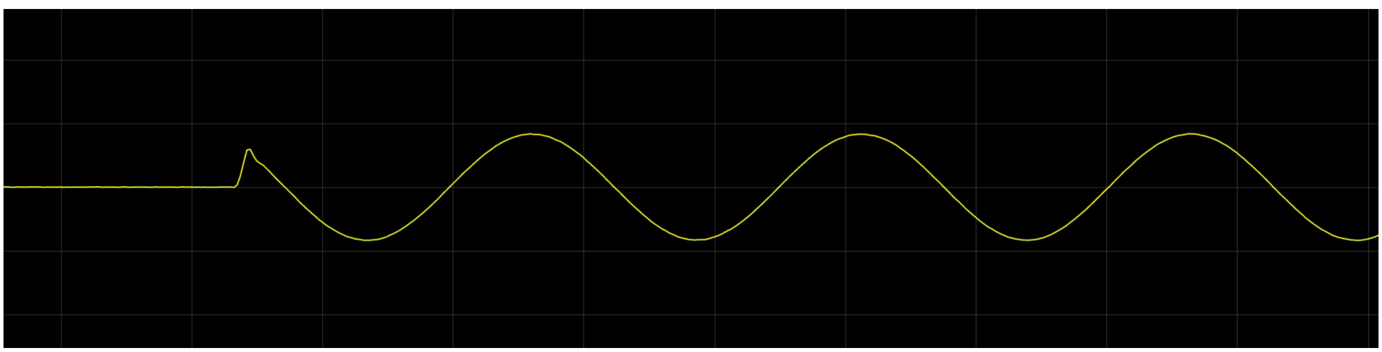

Figure 8: Card Signal at 13.58 MHz

In order to identify the card's signal, the data underwent a two-step filtering process. First, it was passed through a high-pass filter, followed by the application of a band-pass filter. The filtered data was then plotted in both the time domain (Figure 9) and the frequency domain (Figure 10). Analysis of the frequency domain graph reveals that the highest amplitude corresponds to the desired 423.75 KHz frequency, indicating the presence of the card's signal. However, it is apparent from the time domain plot that the card's signal has not been successfully isolated. Instead, the observed signals in the time domain primarily consist of noise originating from the signals emitted by the card reader.

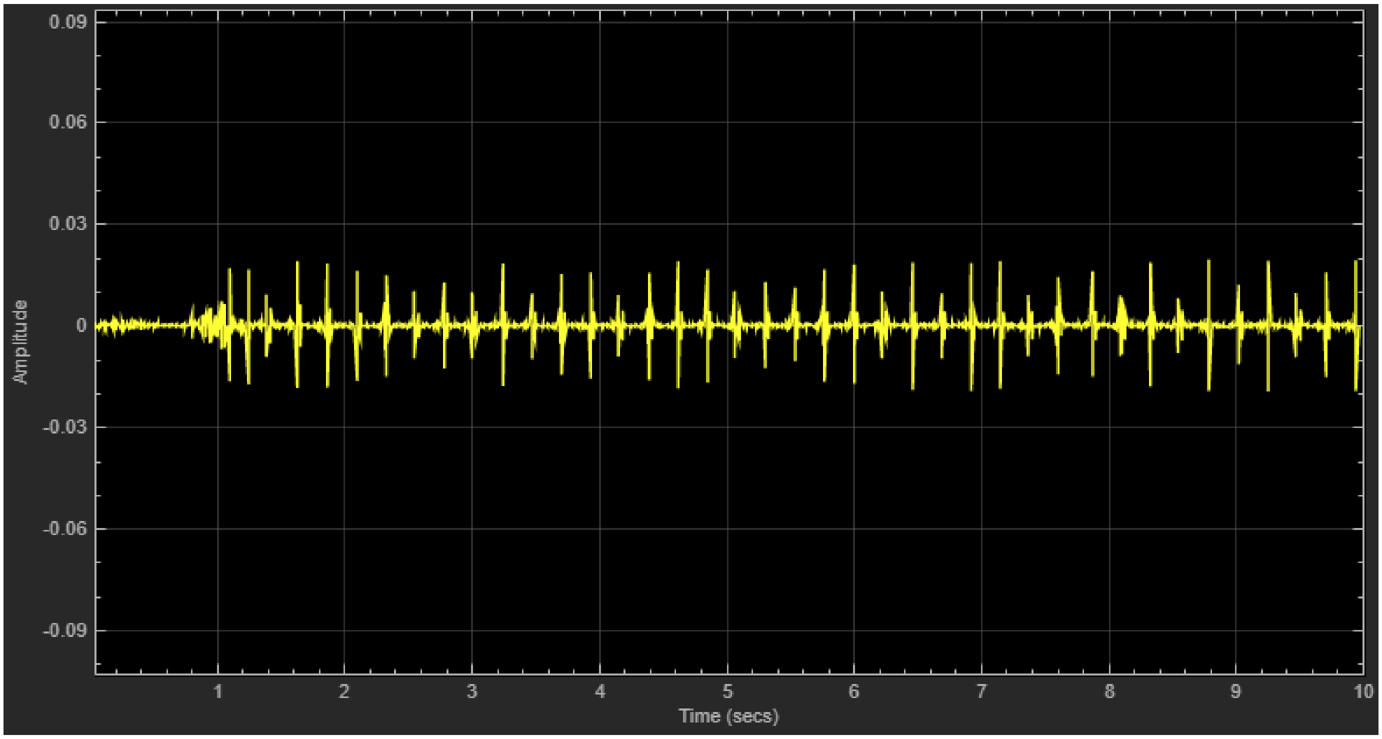

Figure 9: Band-pass Time Domain

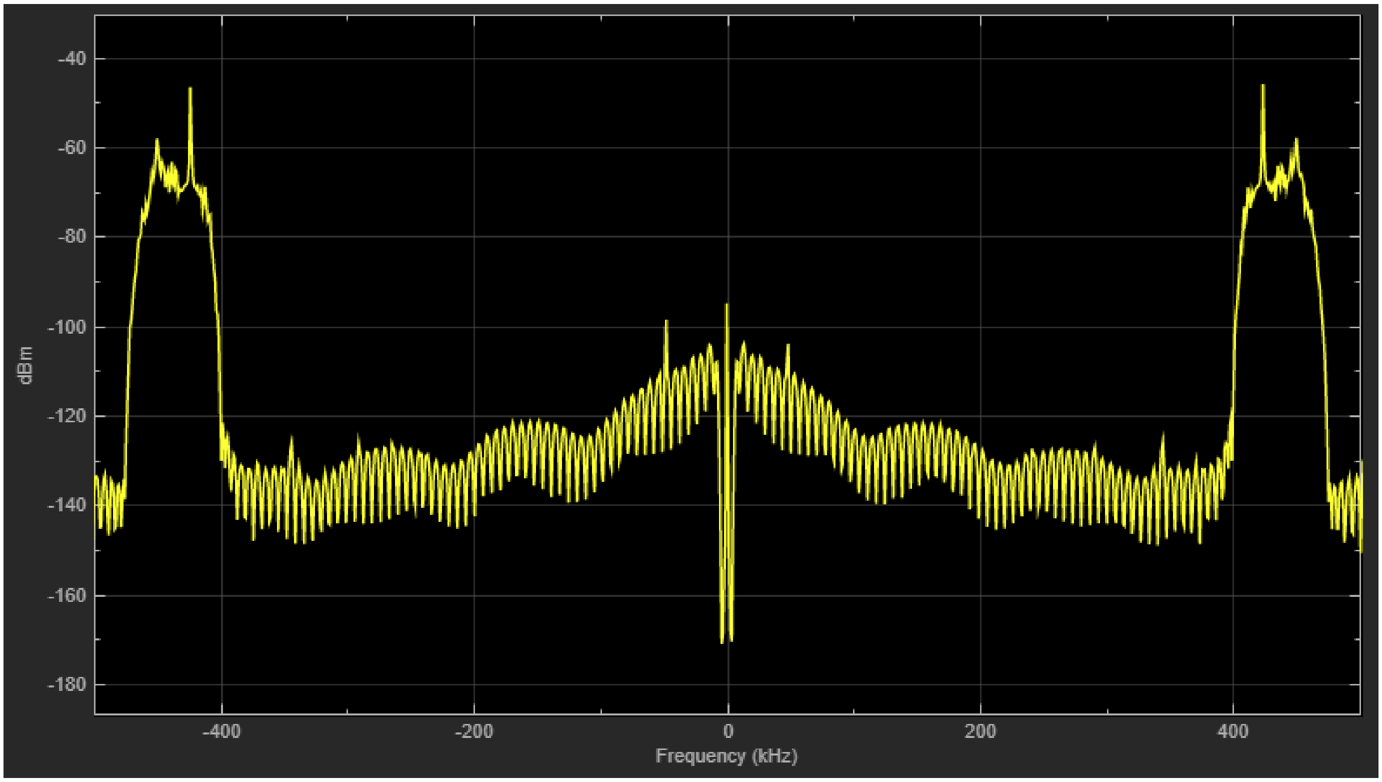

Figure 10: Band-pass Frequency Domain

5 Conclusion

In summary, this week's focus involved the utilization of filters to extract specific components of the signals received from both the card reader and the card itself. The binary sequence emitted by the card reader was successfully isolated using the employed filters. However, despite the efforts, the signal originating from the card could not be effectively separated from the surrounding noise. Further exploration and refinement may be necessary to address this challenge in isolating the card's signal.

6 References

Frequency Offset Calibation: https://www.mathworks.com/help/supportpkg/usrpradio/ug/frequency-offset-calibration-with-usrp-tm-hardware.html