ECE 464 Computing Project 2

Thomas Snyder

November 2020

1 Introduction

Computing project 2 is a continuation of previous work, and is the second of a series of three works that will ultimately simulate a digital communication system. Prior work in Computing Project 1 has created a transmitter and channel that generates and sends a 2-bit sequence via quadrature phase-shift keying (QPSK). In this work, the signal transmitted through the channel is received, demodulated, and downsampled to the proper symbol rate. In the final work of this sequence, an equalizer, a matched filter and quantizer will be implemented to extract the original 2-bit sequence from the transmitted signal.

This project implemented the signal reconstruction from a modulated and noisy transmission. The input will be the transmitted signal from previous work. The output will be a time-domain signal that closely resembles the sequence after it was upsampled and passed through the square-root-raised (SRR) cosine pulse pulse shaping function in Computer Project 1. Within the module, the signal will be filtered and subsampled to a rate that matches the symbol rate of four samples per symbol. There will be additive noise introduced by the channel, but the general shape and spectrum should match. This report will detail the process and design decisions made to reconstruct each signal from the transmitted signal.

2 Methods

Bandpass Filtering Once the modulated sequence has reached the receiver, additive zero-mean noise has been introduced through the channel. This noise must be removed as much as possible before continuing to create a clear signal at the end of the demodulation process. A bandpass filter is needed to remove all noise outside the bandwidth of the original signal. The question becomes: What center frequency and bandwidth must the filter have? It was known that the signal was modulated with a $\sin(\omega_c)$ and $\cos(\omega_c)$, and that the upsampled bandwidth of the signal was $B = \frac{3\pi}{160}$ due to the SRR cosine pulse. Thus, the center frequency of the signal was ω_c, and the bandwidth for this signal was $B = \frac{3\pi}{160}$, which allows the design on a bandpass filter with identical center frequency and passband bandwidth. The windowing method was employed to design this filter. The method is as follows. Define some linear phase FIR filter with ideal frequency response

$$H_d(e^{j\omega}) = \begin{cases} e^{-j\frac{\omega(M-1)}{2}} & 0 \le |\omega| < \omega_c \\ 0 & \text{otherwise.} \end{cases}$$

Then the inverse Fourier transform of $H_d(\omega)$ is defined below, where the windowing method truncates the impulse response to length M.

$$h_d[n] = \begin{cases} \frac{\sin(\omega_b(n-(M-1)/2)}{\pi(n-(M-1)/2)} & 0 \le n \le M-1 \\ 0 & \text{otherwise.} \end{cases}$$

ω_b is the cutoff frequency of the filter. In this case, it was set to $\frac{3\pi}{160}$. M is some arbitrary length that may be empirically determined. Note that a choice of M affects the delay of the filter. Larger M causes larger delay, but more accurate passband and stopband characteristics. Now, the low-pass filter was shifted to a bandpass filter via the frequency shift property of the Fourier transform:

$$x[n]e^{-jat} \overset{\text{FT}}{\longleftrightarrow} X(e^{j(\omega-a)}).$$

Viewing the cosine as the sum of complex exponentials, the filter was multiplied by a cosine function with $\omega = \omega_c$ to shift the filter into a bandpass filter.

$$\cos\omega = \frac{e^{j\omega} + e^{-j\omega}}{2}$$

Once the impulse response of the ideal filer was computed, it was multiplied by a windowing function to improve stopband attenuation. In this work, a Blackman window was used, which is defined for $0 \leq n \leq M$ as

$$w[n] = 0.42 - 0.5 * \cos\left(\frac{2\pi n}{M}\right) + 0.08 * \cos\left(\frac{4\pi n}{M}\right).$$

For the linear-phase filter, a time-delay was introduced, which was accounted for by adding $(M-1)/2$ zeros to the end of the signal before convolving with the filter. To shift the signal back, the front $(M-1)/2$ samples were truncated. After filtering noise, the signal is sampled to simulate analog to digital conversion.

A/D Conversion The analog to digital (A/D) conversion within the digital system was simulated via subsampling the channel signal. It was known from Computing Project 1 that the symbol rate of the channel signal was 80 samples per symbol. Sampling the signal meant decreasing this rate, specifically to eight samples per symbol, which is eight times the original symbol rate. To accomplish this, the filtered channel signal $r[n]$ was subsampled by a factor of 10, which had the effect of expanding the spectrum of the signal by that factor:

$$x[n] = r[10n]$$
$$X(\omega) = R\left(\frac{\omega}{10}\right)$$

This meant that the carrier frequency of the modulated signals became 4.4π. It was known that any signaled sampled by impulse train was $2\pi-$periodic, thus the carrier frequency within the first period was 0.4π. The subsampled signal was filtered based on this fact, using the same technique detailed in **Bandpass Filtering** with $\omega_c = 0.4\pi$ and bandwidth $B = \frac{3\pi}{16}$ because the bandwidth was also expanded by the sampling factor. The gain of the filter was set to 10 because the sampling period is $N = 10$. Once the signal is "converted" to a digital signal, demodulation may occur to separate the different sequences into their respective signals.

Demodulation Demodulation is the process of recovering the original signals by centering the carrier frequency to zero. This was achieved by multiplying the signal by its original modulation function; however, due to the intermediate frequency change from subsampling the signal with the A/D converter, the center frequency of the demodulation functions were changed to match the new center frequency of the signal, which was $\omega_c = 0.4\pi$. The A/D converted signal was multiplied by $\cos(0.4\pi)$ to produce the $b_1[n]$ signal, and the $b_2[n]$ signal was produced by multiplying by $\sin(0.4\pi)$. Note that no extra phase was added to the sine signal because linear-phase filter phase shifts result in time-shift, which has been accounted for after every filter in the process. While centering the spectrum, demodulation also introduced high-frequency artifacts of the spectrum. These were removed with a low-pass filter. The low-pass filter used the same windowing technique described in **Bandpass Filtering**, only this time, the impulse response was not multiplied by the cosine function because it was not necessary to shift the filter to a different frequency band. The bandwidth for this filter was $\frac{3\pi}{16}$ because demodulation did not introduce any spectrum expansion or compression. After this block, the signals were subsampled by a factor of two in the same manner as discussed in **A/D Conversion**. The filter that was used had bandwidth $\frac{3\pi}{8}$ for the expansion by two caused by the subsampling.

3 Results and Discussion

This module in the computing project sequence filtered the noise from a transmitted signal, simulated an A/D converter via subsampling, and demodulated and subsampled the signal to the original sequences $b_1[n], b_2[n]$ at the symbol rate just after the SRR cosine pulse shaping occurred. The results of each of these steps are shown and discussed. This section focuses on three points in the process: after bandpass filtering, after A/D subsampling, and after demodulation.

Bandpass Filtering Filtering the channel noise is straightforward to observe. When the signal is observed before and after noise, it is clear that the bandpass filter was successfully implemented to filter the noise in the signal. As can be observed from Figure 1 and 3, the noise has been filtered while leaving the original transmitted signal as it was. The filter is quite effective.

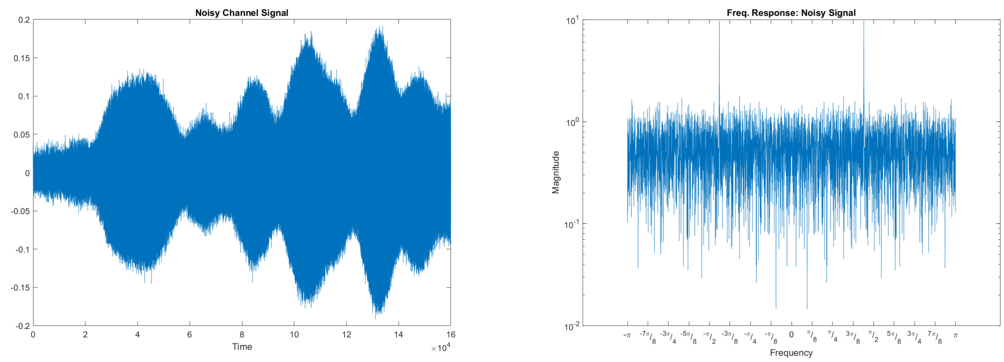

Figure 1: Unfiltered channel signal, frequency response.

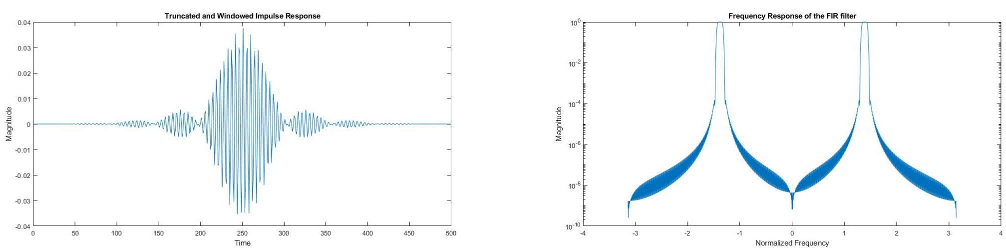

Figure 2: Bandpass filter for removing noise induced by the channel.

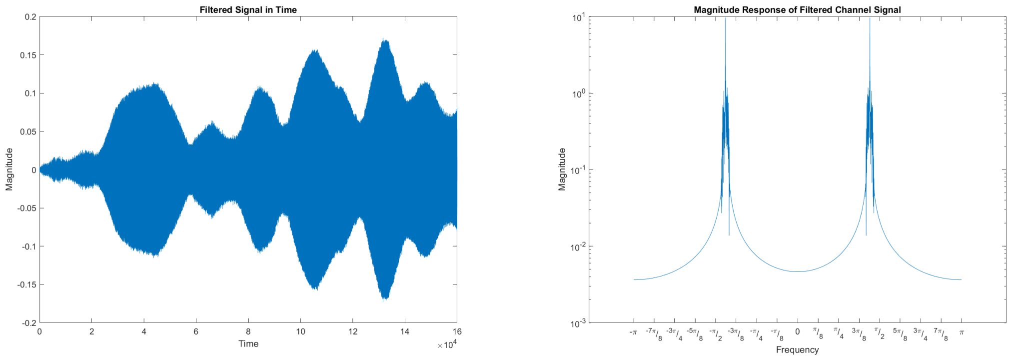

Figure 3: Channel signal after filtering the noise with a bandpass filter.

A/D Conversion When a signal is subsampled by a factor, α, frequency domain spectrum will expand by the same factor α while also becomeing $2\pi-$periodic. Given the original carrier frequency of $\omega_c = 0.44\pi$, the subsampled signal should have carrier frequency $\omega_c = 0.4\pi$ due to the 2π periodicity of the signal. The signal will then be filtered through a bandpass filter to rid any artifacts caused by the subsampling process. Figure 4 shows the results of the A/D conversion subsampling. As is observed, the A/D converted signal is an expanded form of its original. The bandpass filter (Figure 5) removes the signal outside the passband. The noise seen in the spectrum may be attributed to residual channel noise.

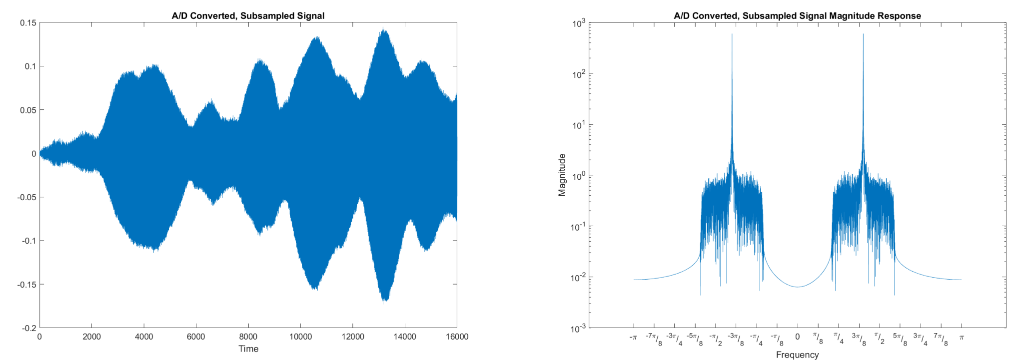

Figure 4: A/D converted signal after bandpass filtering. Notice that the signal looks similar to the filtered channel signal, only it is slightly lower in frequency and expanded.

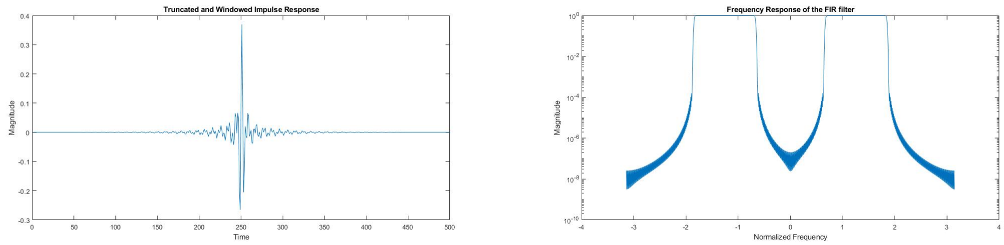

Figure 5: A/D bandpass filter.

Demodulation Finally, when the signal has been prepared and sampled, it may be demodulated to remove the carrier frequency and leave the original signal. The expected result of this process is to have the signal spectrum centered around zero with the time-domain envelope similar to the transmitted signal before modulation. Figure 6 shows the unfiltered demodulated signal. The demodulation process added high-frequency artifacts, which must be filtered out. Figure 7 is the low-pass filter used to perform removal, and Figure 8 is the result. Note that there is a single copy of the signal in the spectrum, and a clear envelope of $b_1[n]$. Only $b_1[n]$ is shown because the same process follows for $b_2[n]$, only a sine function is used in place of cosine for demodulation.

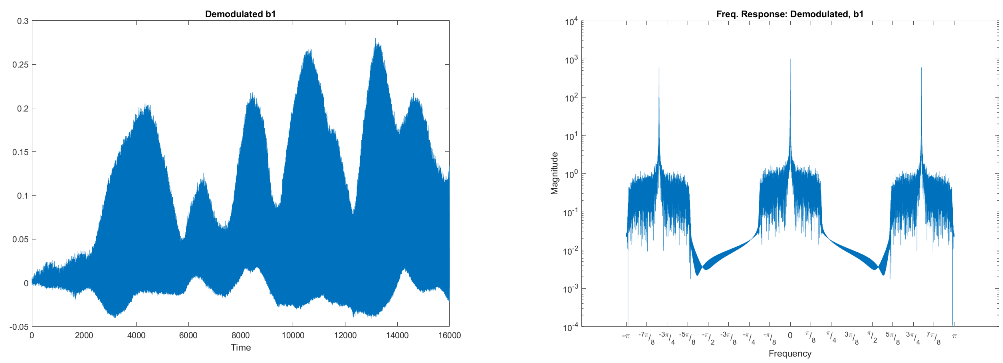

Figure 6: Demodulated signal before low-pass filtering.

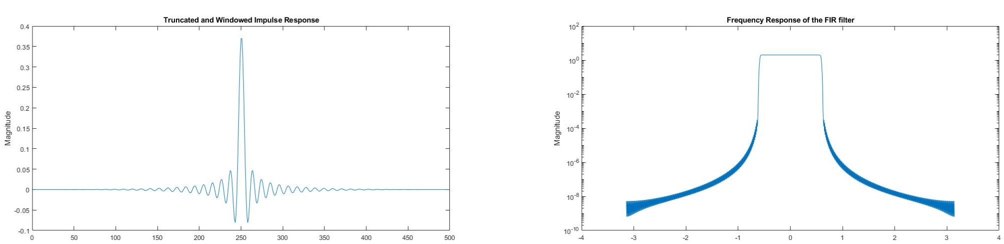

Figure 7: The demodulation low-pass filter, Bandwidth $B = \frac{3\pi}{16}$

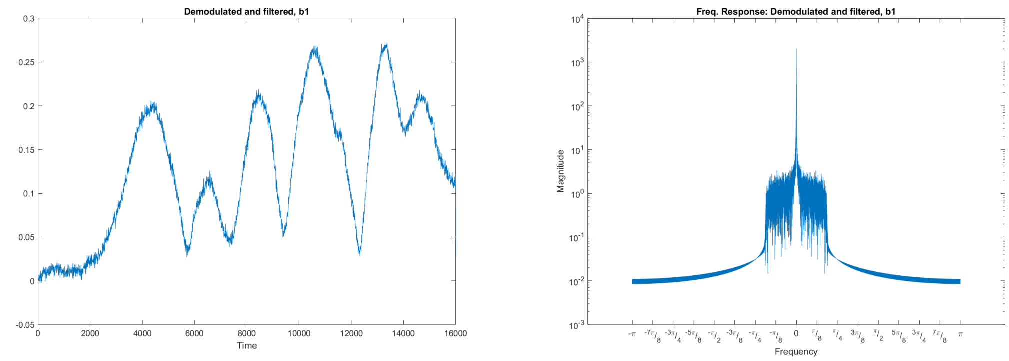

Figure 8: Demodulated, filtered signal.

Final Signal Comparing the signal after all processes (including the final downsample by a factor of two) with its original version just after the SRR cosine pulse interpolation in the first module yields similar signals. This implies that the processes within this module have been implemented correctly. Figure 9 shows the original signal just after the pulse shaping procedure. Figure 10 shows the recovered signal. Note the identical timescale, implying the recovered signal has been downsampled to the correct symbol rate. The envelope similarity also proves that the signal has been properly recovered.

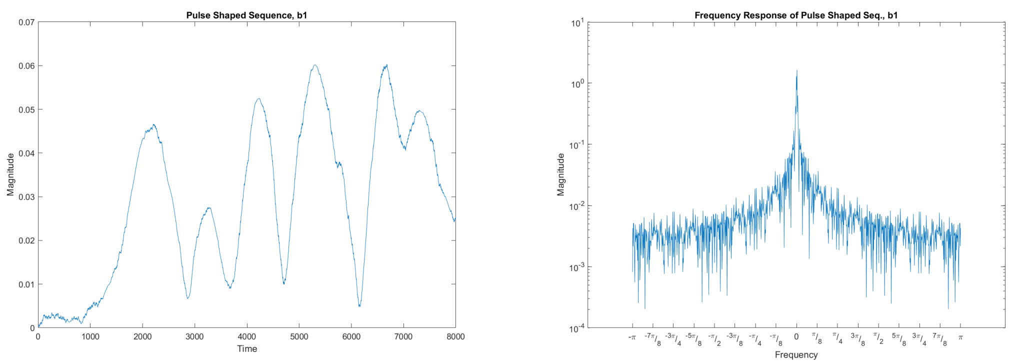

Figure 9: SRR cosine pulse shaped sequence.

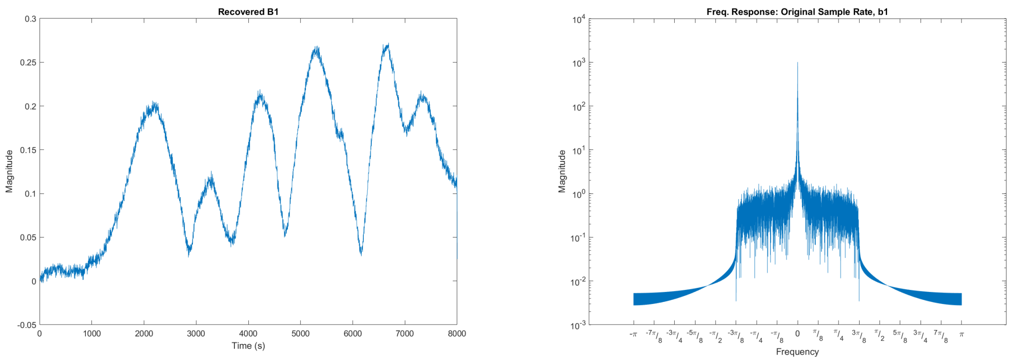

Figure 10: Recovered pulse shaped sequence.

4 Conclusion

This project took a modulated, noisy signal that was the combination of two signals modulated with different phase, and recovered the signals that were modulated. The process involved implementing a bandpass filter to attenuate noise, simulating an A/D converter through subsampling, and creating a demodulator to recover the signals from the sampled channel signal.

This project worked with sampling, modulation, and FIR filter creation. I now have a greater understanding of subsampling a signal. Before the project, I knew of demodulation and how to do it, but I had never implemented anything in real-life. It's been really rewarding for me to see it all come together. The generation of an FIR filter with the windowing method was also a new concept I put into practice. I now understand the trade-offs of longer versus shorter FIR filters, and have created a new understanding of the common windows by implementing them.