ECEn 390 Milestone 2 Report

Objectives

The purpose of this milestone is to (1) design 10 bandpass filters, (2) design a decimating system and analyze noise aliasing, and (3) analyze the full system with noise.

Task 1

Summary

In this task I created 10 IIR bandpass filters of length 11 centered at the frequencies of each of the ten players for the laser tag system. I chose a bandwidth of 50Hz, which was very narrow and well within the specification. Even on the logarithmic frequency scale, the filters did not overlap. After creating square waves at each of the 10 player frequencies and passing them through the 10 IIR filters, I was able to see that each filter successfully filtered for the correct corresponding frequency signals. Thus we can use these 10 filters to compute the energy of a signal passing through them and identify the largest energy in order to identify hits from players in the laser tag game.

Specifications

- Butterworth IIR bandpass filter configuration with center frequencies of each player
- Filter length of 11
- Equal filter bandwidths for all 10 filters
- Attenuation of adjacent player frequency <-20dB
- Player frequencies: 1471, 1724, 2000, 2273, 2632, 2941, 3333, 3571, 3846, and 4167 Hz.
- Pulse length 200ms

Design

The filter coefficients for each of the 10 filters are shown in the tables below.

Table 1. Denominator Coefficients (IIR Filter A).

Filter 1	Filter 2	Filter 3	Filter 4	Filter 5	Filter 6	Filter 7	Filter 8	Filter 9	Filter 10
1	1	1	1	1	1	1	1	1	1
-5.96377	-4.63779	-3.05913	-1.40717	0.820109	2.708087	4.947984	6.170189	7.409291	8.574306
19.12534	13.50222	8.641749	5.690414	5.167376	7.831907	14.69161	20.12723	26.85794	34.30658
-40.3415	-26.156	-14.2788	-5.73747	3.258035	12.20161	29.08241	42.97419	61.57879	84.03529
61.53747	38.58967	21.30227	11.95803	10.3929	18.6515	43.17984	65.95805	98.25826	139.2851
-70.0197	-43.039	-22.1939	-8.54353	4.810178	18.75816	48.44079	75.23044	113.5946	163.0512
60.29881	37.81293	20.8735	11.71735	10.18372	18.27609	42.3107	64.63041	96.28045	136.4815
-38.7338	-25.1136	-13.7098	-5.50883	3.1282	11.71536	27.92343	41.26159	59.12474	80.68629
17.99353	12.70318	8.130355	5.353679	4.861593	7.368439	13.82219	18.93613	25.26853	32.27636
-5.49791	-4.27551	-2.82016	-1.29725	0.756045	2.496542	4.561466	5.688198	6.830506	7.904514
0.903328	0.903328	0.903328	0.903328	0.903328	0.903328	0.903328	0.903328	0.903328	0.903328

Table 2. Numerator Coefficients (IIR Filter B).

Filter 1	Filter 2	Filter 3	Filter 4	Filter 5	Filter 6	Filter 7	Filter 8	Filter 9	Filter 10
9.09E-10	9.09E-10	9.09E-10	9.09E-10	9.09E-10	9.09E-10	9.09E-10	9.09E-10	9.09E-10	9.09E-10
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
-4.55E-09	-4.55E-09	-4.55E-09	-4.55E-09	-4.55E-09	-4.55E-09	-4.55E-09	-4.55E-09	-4.55E-09	-4.55E-09
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
9.09E-09	9.09E-09	9.09E-09	9.09E-09	9.09E-09	9.09E-09	9.09E-09	9.09E-09	9.09E-09	9.09E-09
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
-9.09E-09	-9.09E-09	-9.09E-09	-9.09E-09	-9.09E-09	-9.09E-09	-9.09E-09	-9.09E-09	-9.09E-09	-9.09E-09
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
4.55E-09	4.55E-09	4.55E-09	4.55E-09	4.55E-09	4.55E-09	4.55E-09	4.55E-09	4.55E-09	4.55E-09
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
-9.09E-10	-9.09E-10	-9.09E-10	-9.09E-10	-9.09E-10	-9.09E-10	-9.09E-10	-9.09E-10	-9.09E-10	-9.09E-10

The plots of the frequency-domain transfer functions of all 10 filters are shown in Figure 1 below. The upper plot is the plot with linear scaling and the lower plot is with logarithmic scaling (in decibels).

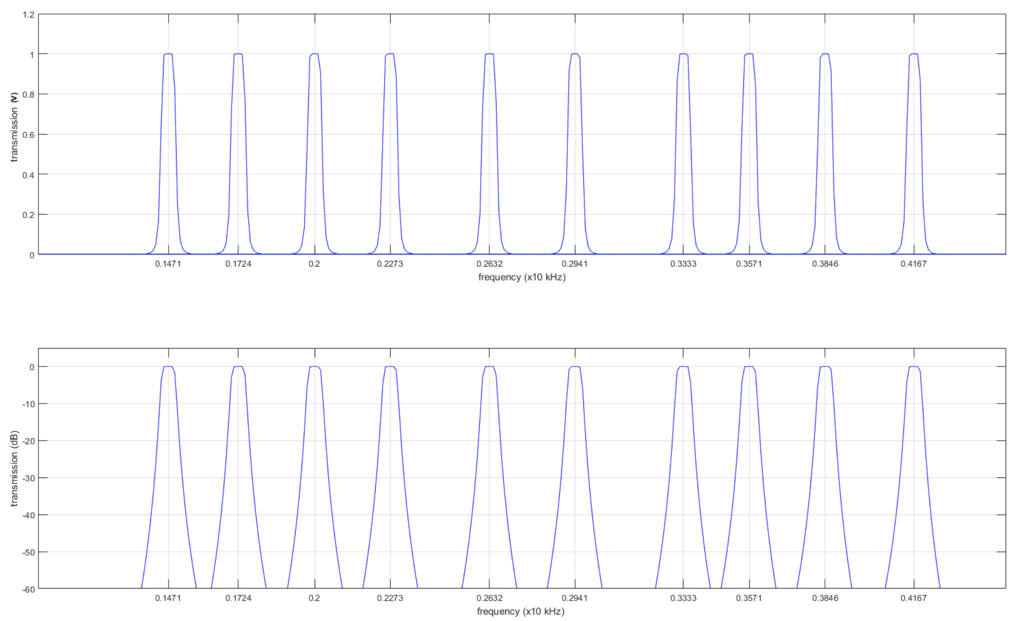

Figure 1. Frequency-domain transfer functions.

For each of the filters, the bandwidth I selected was 50Hz. The filters are also all of order 10 (length 11). Because of the narrow bandwidth, the filters far exceeded the specification that adjacent player frequency attenuation was less than 20dB.

The energy of a signal can be calculated with the following equation:

$$E = \int_{-\infty}^{\infty} |x(t)|^2 \tag{1}$$

Using the ‘square’ command in MATLAB, I created 10 different square waves at each of the 10 player frequencies to pass through the filters. Since each of these is a discrete set of data, Equation *(1)* becomes

$$E = \sum_{n=0}^{\text{Length}-1} |x[n]|^2 \qquad (2)$$

In MATLAB, I used Equation *(2)* to compute the energy of each square wave for each of the ten filters. The results are displayed in *Figure 2* below.

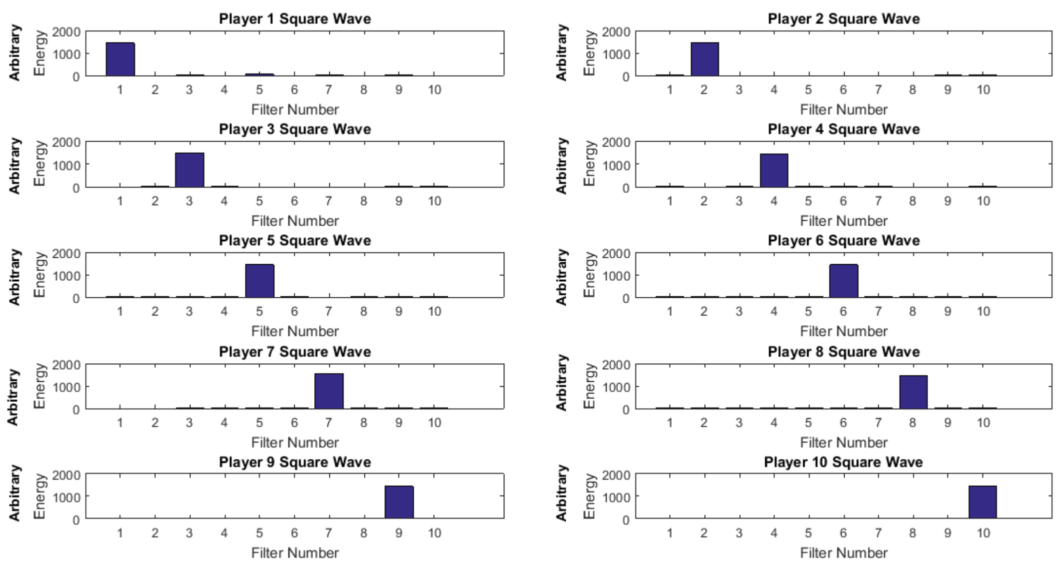

Figure 2. Energy of Square Waves through Bandpass Filters

Some of the frequencies, notably Player 1’s frequency, have harmonics, so there is some nonzero energy computed for the signals as they pass through other filters. However the maximum energy of each signal corresponds to the appropriate filter for each square wave frequency.

Task 2

Summary

In this task I created an FIR filter of length 81 with a corner frequency of 7 kHz to act as an anti-aliasing filter for decimating signals for the laser tag system. Using a blackman window with a bandwidth of 14kHz I was able to achieve a maximum player variation of 0.0318 dB and an out of band rejection of 54.8922 dB.

Specifications

- Anti-aliasing FIR filter
- Maximum filter length: 81

- Maximum player variation: 1dB
- Out of band rejection: 40dB
- Stopband: 10kHz < f < 50kHz

Background (Goals)

We want to design an anti-aliasing filter so we can decimate our signal from 100 kHz to 10 kHz without unwanted aliasing. For example, *Figure 3* shows a synthesized signal with two frequencies, 3 kHz and 6 kHz, sampled at 100 kHz and plotted in the time and frequency domains.

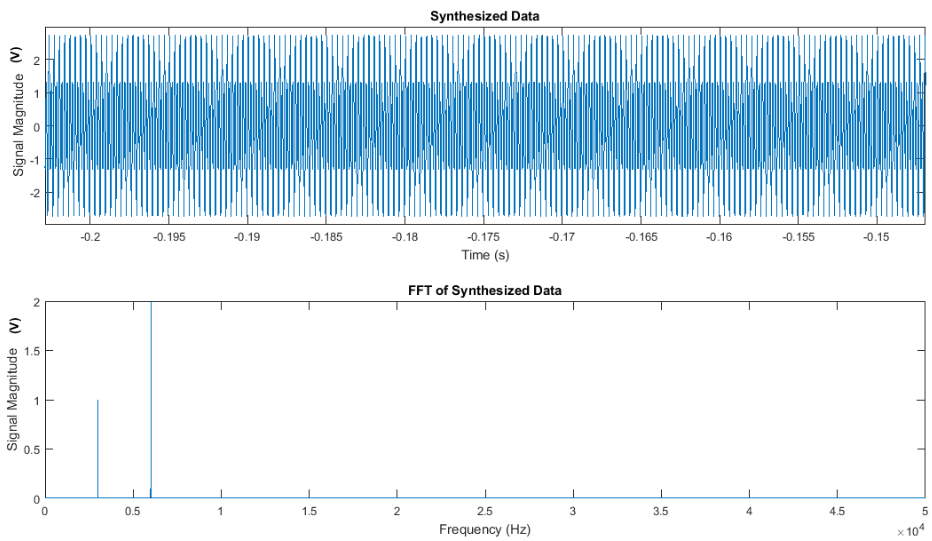

Figure 3. Synthesized Sample.

The signal was then down-sampled to 10 ksamples/sec and plotted in *Figure 4* below.

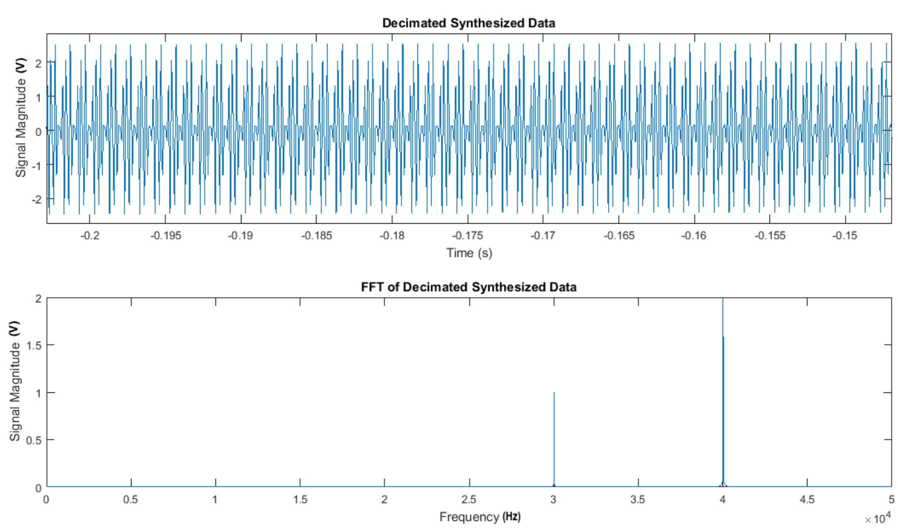

Figure 4. Down-sampled Synthesized Sample.

The 3kHz frequency was unaffected, but the 6kHz frequency aliases into a 4kHz frequency, because the sampling rate is 10kHz (10kHz – 6kHz = 4kHz). This is exactly the kind of aliasing that we want to filter out in our design.

Design

I opted for a blackman window with a bandwidth of $2*f_c = 2*7kHz = 14kHz$ for my filter design. The filter has length 81 with the following B coefficients:

Table 3. Numerator Coefficients (FIR Filter B).

0.00E+00	-4.60E-06	-1.62E-05	-2.40E-05	-1.04E-05	4.20E-05	1.41E-04	2.75E-04	4.07E-04	4.80E-04
4.23E-04	1.74E-04	-2.92E-04	-9.39E-04	-1.65E-03	-2.24E-03	-2.46E-03	-2.09E-03	-9.76E-04	8.75E-04
3.25E-03	5.70E-03	7.60E-03	8.27E-03	7.08E-03	3.71E-03	-1.74E-03	-8.63E-03	-1.58E-02	-2.15E-02
-2.39E-02	-2.14E-02	-1.27E-02	2.57E-03	2.38E-02	4.93E-02	7.66E-02	1.03E-01	1.24E-01	1.38E-01
1.43E-01	1.38E-01	1.24E-01	1.03E-01	7.66E-02	4.93E-02	2.38E-02	2.57E-03	-1.27E-02	-2.14E-02
-2.39E-02	-2.15E-02	-1.58E-02	-8.63E-03	-1.74E-03	3.71E-03	7.08E-03	8.27E-03	7.60E-03	5.70E-03
3.25E-03	8.75E-04	-9.76E-04	-2.09E-03	-2.46E-03	-2.24E-03	-1.65E-03	-9.39E-04	-2.92E-04	1.74E-04
4.23E-04	4.80E-04	4.07E-04	2.75E-04	1.41E-04	4.20E-05	-1.04E-05	-2.40E-05	-1.62E-05	-4.60E-06
0.00E+00									

For the sake of space, these were put in the table from left-to-right, top-to-bottom, though they are in fact part of one vector of length 81.

The transfer function of the filter is plotted in the frequency domain below.

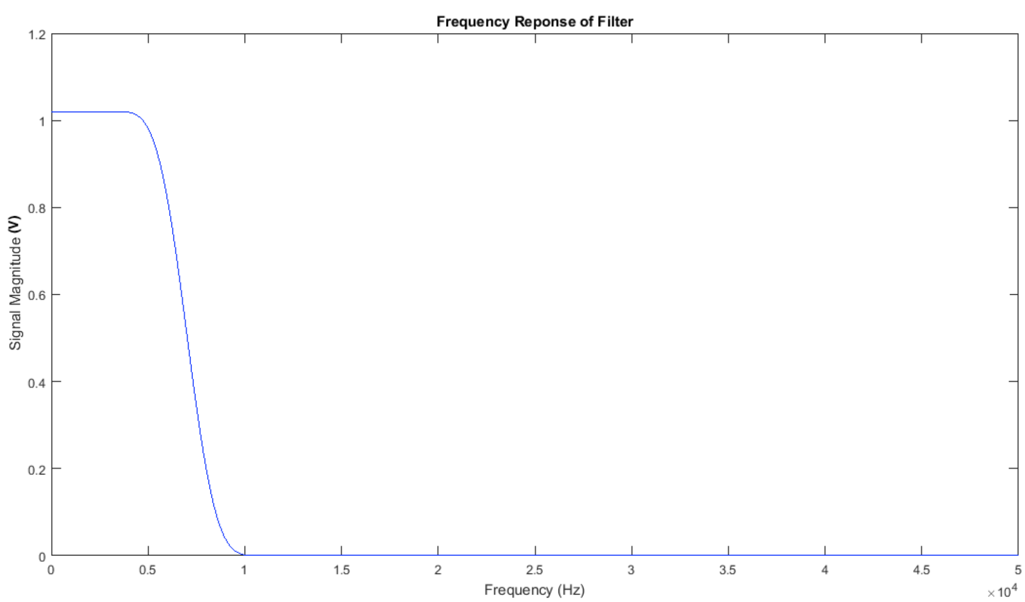

Figure 5. Frequency response of low-pass filter.

We had previously sampled some optical noise by pointing our receiver at the fluorescent lights and collecting a 500ms sample at a sampling rate of 100kHz. I then down-sampled this signal from 100 ksamples/sec to 10 ksamples/sec, first without any filtering and second after first passing it through my low-pass FIR filter. The frequency-domain plots of these two signals are shown in *Figure 6* below.

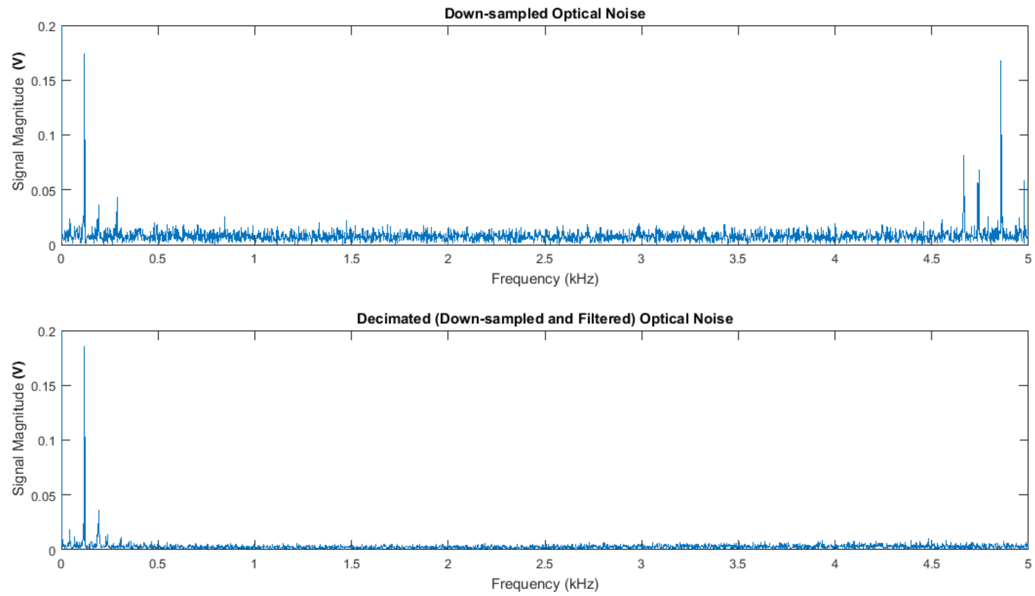

Figure 6. Down-sampled and Decimated Optical Noise.

The aliasing that occurred between 4.5 and 5 kHz was eliminated by the low-pass filter.

Verification

The first specification is a maximum filter length of 81. Since my filter length is 81, this expectation is met.

The second specification is a maximum player variation of 1dB.

The gain of each of the 10 players is shown below.

Table 4. Player Variation Gains.

Player 1	Player 2	Player 3	Player 4	Player 5	Player 6	Player 7	Player 8	Player 9	Player 10
1.0204	1.0205	1.0205	1.0205	1.0204	1.0203	1.0203	1.0202	1.0195	1.0168

Thus our maximum variation is $20|\log_{10} 1.0168/1.0205| = \boxed{0.0318\ dB < \ 1.0\ dB}$, so this meets the in band uniformity specification.

The third specification is an out of band rejection of 40 dB.

The gain of the filter frequency response from 10kHz to 50 kHz is shown below:

Table 5. Out of Band Gains.

10kHz	15kHz	20kHz	25kHz	30kHz	35kHz	40kHz	45kHz	50kHz
0.001800492	2.56E-05	5.69E-06	1.97E-06	8.80E-07	4.70E-07	2.93E-07	2.16E-07	1.95E-07

So our minimum out of band rejection is $20|\log_{10} 0.001800492| = \boxed{54.8922\ dB > \ 40\ dB}$, which meets the specification.

Task 3

Summary

I integrated the filters from tasks 1 and 2 together to build the whole signal-processing algorithm in MATLAB. I then tested the system with a sample signal. I did this by adding a noise sample to a square wave at the player 1 frequency and then running it through the algorithm. The algorithm is as follows: first pass it through an anti-aliasing filter (the FIR filter from task 2), then down-sample from F_s = 100kHz to F_s = 10kHz, then pass it through bandpass filters at the 10 player frequencies (the IIR filters from Task 1), and finally compute the energies of the resulting signals and compare them to determine which player did the shooting.

Specifications

- Combined signal processing algorithm
- Anti-aliasing filter from Task 2
    - FIR filter
    - 81 ‘b’ coefficients
    - 1 nonzero coefficient of 1 for ‘a’ coefficients
- Bank of bandpass filters from Task 1
    - IIR filters
    - 11 ‘a’ and 11 ‘b’ coefficients
    - center-of-passband frequencies are the 10 player frequencies

Design

I used the coefficients from Tasks 1 and 2 and the ‘filter’ command in MATLAB to create the FIR and IIR filters.

Verification

First I generated a square wave for player 1 with a frequency of 1471 Hz and amplitude of 0.1V as shown in *Figure 7* below.

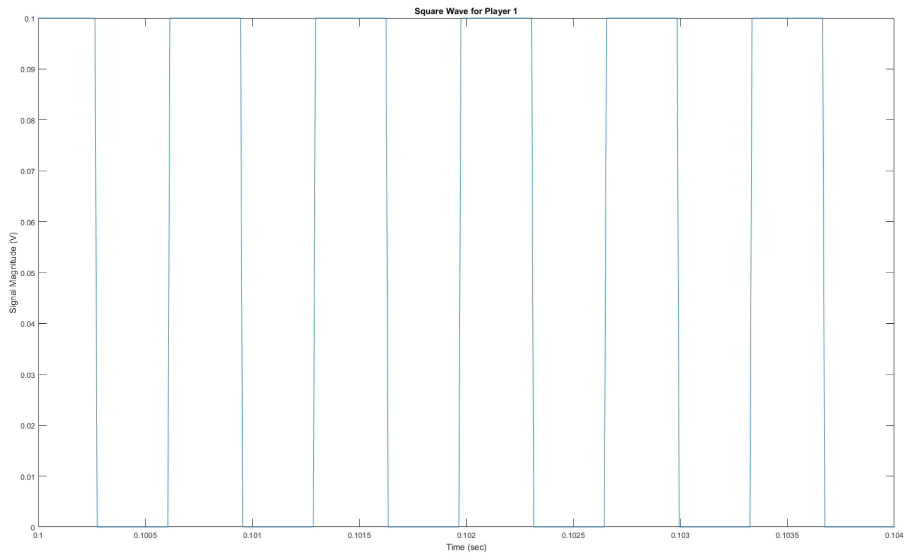

Figure 7. Player 1 Square Wave

The y-axis goes from -0.1V to 0.1V so we know we have the correct amplitude. 1471Hz * .004 ms = 5.884, which makes sense since there are 6 peaks shown.

I also imported the optical noise from lights.mat. A portion of the optical noise data plotted in the time domain is shown in *Figure 8* below.

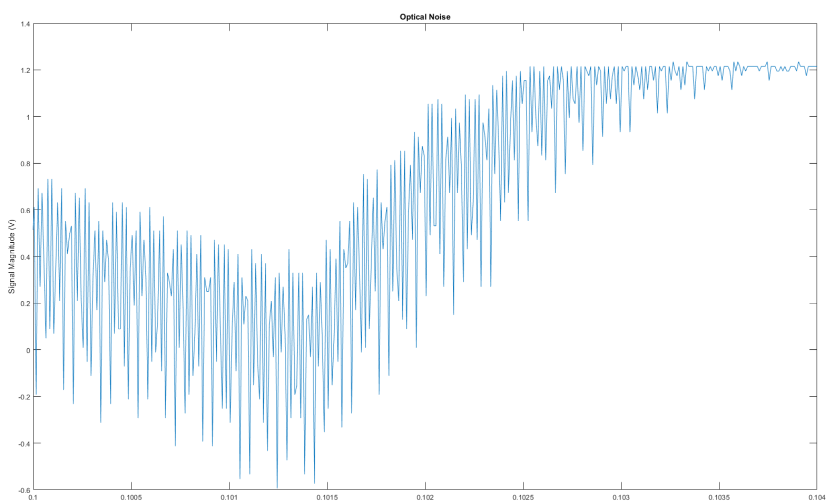

Figure 8. Optical Noise Sample.

Then I combined the player 1 square wave with the optical noise. *Figure 9* shows a portion of the signal plotted in the time domain and the signal plotted in the frequency domain.

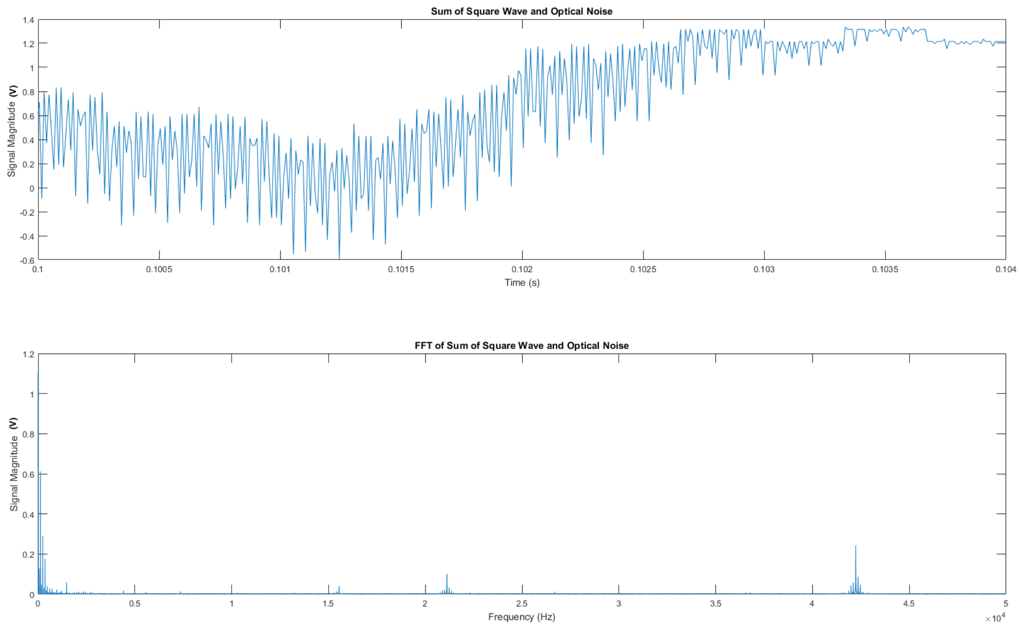

Figure 9. Sum of Player 1 Square Wave and Optical Noise.

This plot makes sense because the noise sample is shifted up or down 0.1V depending on the portion of the square wave that was added. It is especially obvious at the end of the 400ms sample. In the frequency domain we see a large spike around 1471Hz because of the player 1 square wave and smaller spikes around 21kHz and 42kHz from the noise.

I then decimated the signal by passing it through my FIR filter and then down-sampling from F_s=100kHz to F_s=10kHz. The resulting signal is shown below.

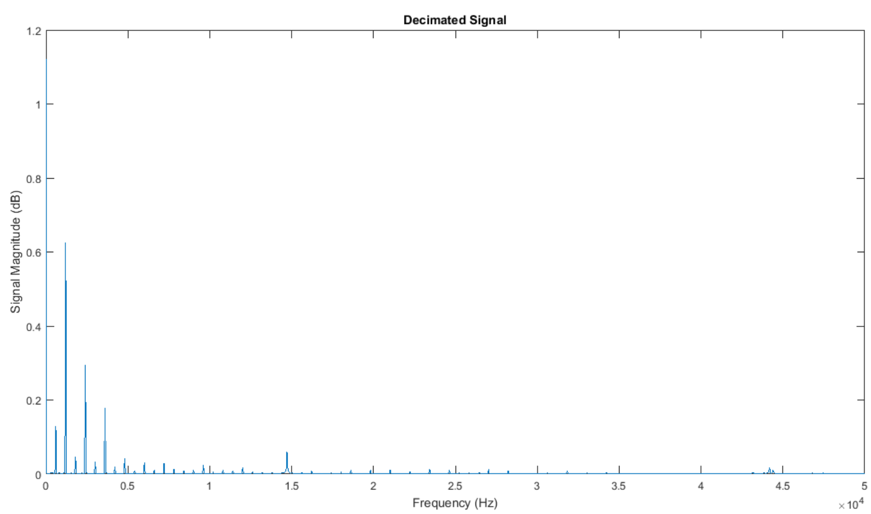

Figure 10. Decimated Signal

This looks pretty similar to the signal in *Figure 9*, without any extra aliased signals, so it appears to have been decimated correctly.

Next I passed the signal through the IIR filter centered at Player 1's frequency of 1471 Hz. The frequency domain plot is shown in *Figure 11* below.

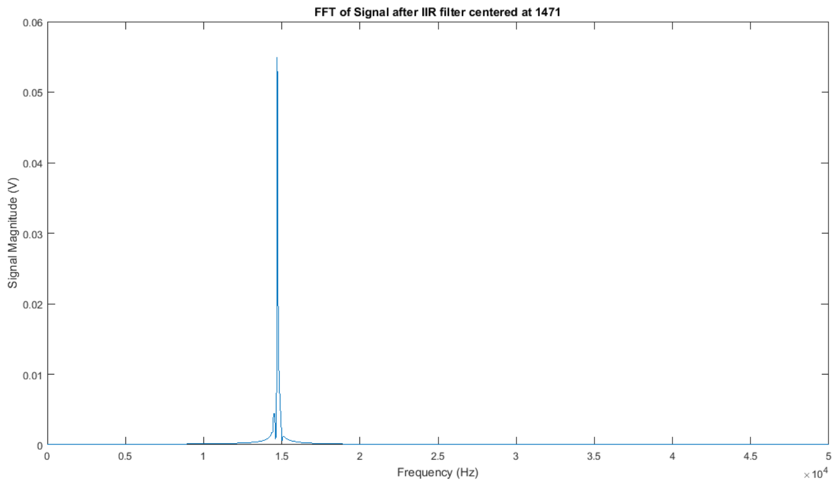

Figure 11. Signal After IIR Filter 1

Note that the spike goes up to almost .06 V, which makes sense since the original signal was 0.1 V. It was attenuated a little bit, but is still a large spike.

Next I passed the signal through the IIR filter centered at player 2's frequency of 1724 Hz, as shown in *Figure 12* below.

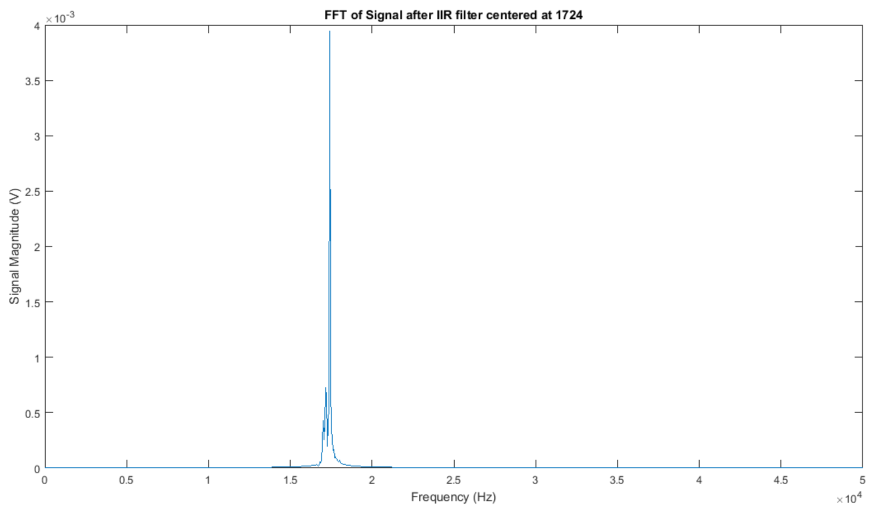

Figure 12. Signal After IIR Filter 2.

The magnitude of this spike is only .004 V, which is much lower than the magnitude of the previous one. Thus we see that this signal has been attenuated significantly by this filter.

Finally, I passed the signal through all 10 filters and computed the energy of each of the filtered signals. The 10 energies are displayed in the bar chart below.

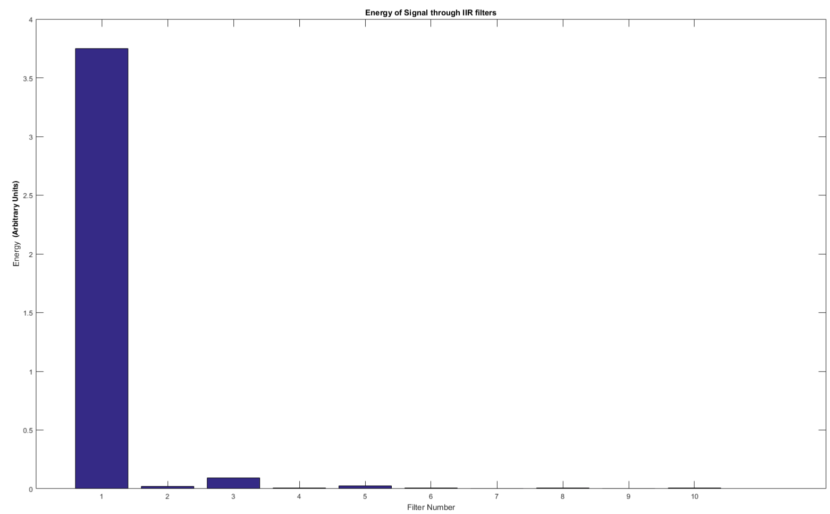

Figure 13. Energy of Signal after each IIR filter.

Note that in the bar chart (and the table) the first filter has a significantly larger energy than that of any of the other filters. The system was able to detect a signal from Player 1.

Conclusion

This was a pretty intense lab. It took a bit of time to understand exactly what was going on in each of the tasks, even from a high level, but I feel that I do finally have an adequate understanding both on a high level and on a low level of what the goals and results of this lab were. Additionally, this lab was a good review of MATLAB, particularly how to plot signals in both time and frequency domains. I think the most confusing parts of this lab were how to know the best way to plot certain things in the frequency domain (e.g. using freqz for filter responses vs. using fft for observing a signal in the frequency domain) and the scaling of each plot (remembering to divide by n or 2n in order to get the signal into a magnitude with units of volts, or using 20log to get it in dB). Overall, though, I feel like I now have a more solid understanding of signal processing, especially for this particular project.