

ECE 3793 – Project 2
Continuous-Wave Modulation and Demodulation

Jack Gross and Griffin Hill
ECE 3793 – Fall 2025

November 25, 2025

1 Introduction

This project serves as an introduction to basic continuous-wave (CW) modulation principles for radio communications. In Part 1, we derive the fundamental mathematics of CW modulation using Fourier transform pairs and properties learned throughout the course. In Part 2, we practice these techniques in MATLAB, demodulating a provided dataset containing multiple audio clips modulated at different carrier frequencies.

The goal of this project is to develop an understanding of basic CW modulation and demodulation for radio communications. By deriving the fundamental calculations using Fourier transform techniques, then applying these principles to demodulate a dataset of various modulated audio clips, we develop a better understanding of these communication concepts.

2 Methods and Derivations for Part 1

The fundamental mathematics required to perform basic modulation are as follows. Performing modulation requires the original signal and a carrier signal. We will start with a single audio tone at 500 Hz and a carrier signal at F_c :

$$\begin{aligned}x(t) &= \cos(2\pi 500t) \\x_c(t) &= \cos(2\pi F_c t)\end{aligned}$$

Calculation 1 Multiply the original signal with the carrier signal in time domain to get the transmitted signal.

$$x_{\text{TX}}(t) = x(t)x_c(t) = \cos(2\pi \cdot 500t) \cos(2\pi F_c t). \quad (1)$$

Using this trigonometric identity

$$\cos A \cos B = \frac{1}{2} [\cos(A + B) + \cos(A - B)], \quad (2)$$

$$x_{\text{TX}}(t) = \frac{1}{2} [\cos(2\pi(F_c + 500)t) + \cos(2\pi(F_c - 500)t)]. \quad (3)$$

we see the transmitted signal is the sum of two scaled signals centered at F_c , one shifted by -500Hz, the other shifted 500Hz.

Calculation 2 Take the Fourier Transform to get frequency domain transmitted signal. Using this CTFT pair

$$\cos(2\pi F_0 t) \longleftrightarrow \frac{1}{2} [\delta(F - F_0) + \delta(F + F_0)], \quad (4)$$

$$X_{\text{TX}}(F) = \frac{1}{4} [\delta(F - F_c - 500) + \delta(F + F_c + 500) + \delta(F - F_c + 500) + \delta(F + F_c - 500)]. \quad (5)$$

we see the Fourier transform of the transmitted signal consists of four scaled impulses at $F = \{F_c + 500, F_c - 500, -F_c - 500, -F_c + 500\}$

Next we will check check our previous calculations by assessing each signal individually.

Calculation 3 Individual transforms found by using CTFT pair in equation (4):

$$X(F) = \frac{1}{2} [\delta(F - 500) + \delta(F + 500)], \quad (6)$$

$$X_c(F) = \frac{1}{2} [\delta(F - F_c) + \delta(F + F_c)]. \quad (7)$$

Calculation 4 Using multiplication-convolution theorem and convolution shifting,

$$\begin{aligned}
 X_{\text{TX}}(F) &= X(F) * X_c(F) \\
 X_{\text{TX}}(F) &= X(F) * \frac{1}{2} [\delta(F - F_c) + \delta(F + F_c)] \\
 X_{\text{TX}}(F) &= \frac{1}{2} [X(F - F_c) + X(F + F_c)] \\
 X_{\text{TX}}(F) &= \frac{1}{4} [\delta(F - F_c - 500) + \delta(F + F_c + 500) + \delta(F - F_c + 500) + \delta(F + F_c - 500)].
 \end{aligned}$$

This is the same as our previous calculation.

Next we take a signal that only contains frequencies within the range $(-22\text{kHz}, 22\text{kHz})$. This reflects the audible range of frequencies which human beings can noticeably hear.

Calculation 5 New audio signal $x_a(t) \longleftrightarrow X_a(F)$ nonzero only for $|F| < 22\text{ kHz}$,

$$x_{\text{TX}}(t) = x_a(t) \cos(2\pi F_c t) \longrightarrow X_{\text{TX}}(F) = \frac{1}{2} [X_a(F - F_c) + X_a(F + F_c)]. \quad (8)$$

The transmitted signal $X_{\text{TX}}(F)$ is two copies of $X_a(F)$ shifted by $\pm F_c$, scaled by $\frac{1}{2}$. Considering only positive values of F , the bandwidth of $X_a(F)$ is 22 kHz, reflecting the audible frequency range of humans. We also see that the bandwidth of $X_{\text{TX}}(F)$, for $F > 0$, is the bandwidth:

$$B = F_c + 22\text{kHz} - (F_c - 22\text{kHz}) = 44\text{kHz} \quad (9)$$

The bandwidth of $X_{\text{TX}}(F) = 44\text{kHz}$ is twice the bandwidth of $X_a(F) = 22\text{kHz}$. This is the positive sideband of the transmitted signal in the frequency domain. This positive sideband of the transmitted signal in the Frequency domain is centered at F_c and shifted by $\pm 22\text{kHz}$. This implies that the carrier signal must be greater than 22kHz so that the full spectrum of the original is preserved.

Next we will down-convert the received signal, which has inevitably picked up some interference, using Fourier concepts.

Calculation 6 Fourier Transform of $x_{\text{RX}}(t) = x_{\text{TX}}(t) + x_I(t)$

$$X_{\text{RX}}(F) = X_{\text{TX}}(F) + X_I(F). \quad (10)$$

Calculation 7 Substituting the expression from Calculation 5,

$$X_{\text{RX}}(F) = \frac{1}{2} [X_a(F - F_c) + X_a(F + F_c)] + X_I(F). \quad (11)$$

Calculation 8 Down-conversion of received signal:

$$x_d(t) = x_{\text{RX}}(t) \cos(2\pi F_c t) \longleftrightarrow X_d(F) = \frac{1}{2} [X_{\text{RX}}(F - F_c) + X_{\text{RX}}(F + F_c)].$$

Expand:

$$X_d(F) = \frac{1}{2} X_a(F) + \frac{1}{4} [X_a(F - 2F_c) + X_a(F + 2F_c)] + \frac{1}{2} [X_I(F - F_c) + X_I(F + F_c)]. \quad (12)$$

Calculation 9 Inverse Fourier transform using convolution shifting and multiplication-convolution theorem:

$$x_d(t) = \frac{1}{2} x_a(t) + \frac{1}{2} x_a(t) \cos(4\pi F_c t) + x_I(t) \cos(2\pi F_c t). \quad (13)$$

After applying the inverse Fourier transform, we have the down-converted signal containing 3 signals: the scaled original signal, the scaled and modulated original signal, and some interference. We will create and apply a low pass filter to derive our original signal.

Calculation 10 Ideal low-pass filter with 22kHz cutoff:

$$H(F) = \text{rect}\left(\frac{F}{44 \times 10^3}\right) \quad (14)$$

Calculation 11 Impulse response of the ideal LPF:

$$h(t) = 44 \times 10^3 \text{sinc}(44 \times 10^3 t) \quad (15)$$

Calculation 12 Applying the filter, $X_{\text{out}}(F) = X_a(F)H(F)$. The interference and carrier signals are outside the passband of the LPF, so they are zeroed out and we are left with:

$$X_{\text{out}}(F) = \frac{1}{2}X_a(F), \quad |F| \leq 22 \text{ kHz}. \quad (16)$$

Calculation 13 Taking the inverse fourier transform, the final recovered audio signal is

$$x_{\text{out}}(t) = \frac{1}{2}x_a(t). \quad (17)$$

3 Methods for Part 2

The MATLAB script was developed to implement the receiver mathematics derived in Part 1, enabling the demodulation and filtering of the audio waveforms from the provided dataset. The dataset, loaded using the `load` command, includes the received signal `rx_data`, sampling frequency $F_s = 529200$ Hz, carrier spacing, and interpolation factor. The time vector t was generated based on F_s and the length of `rx_data`. The time-domain representation of the received signal, as illustrated in Plot 1 (Figure 1), does not provide any significant information. The signal contains many high-frequency carriers layered on top of each other, making it difficult to make sense of the plot as it is essentially just a block of noise. The signal is precisely 10 seconds long.

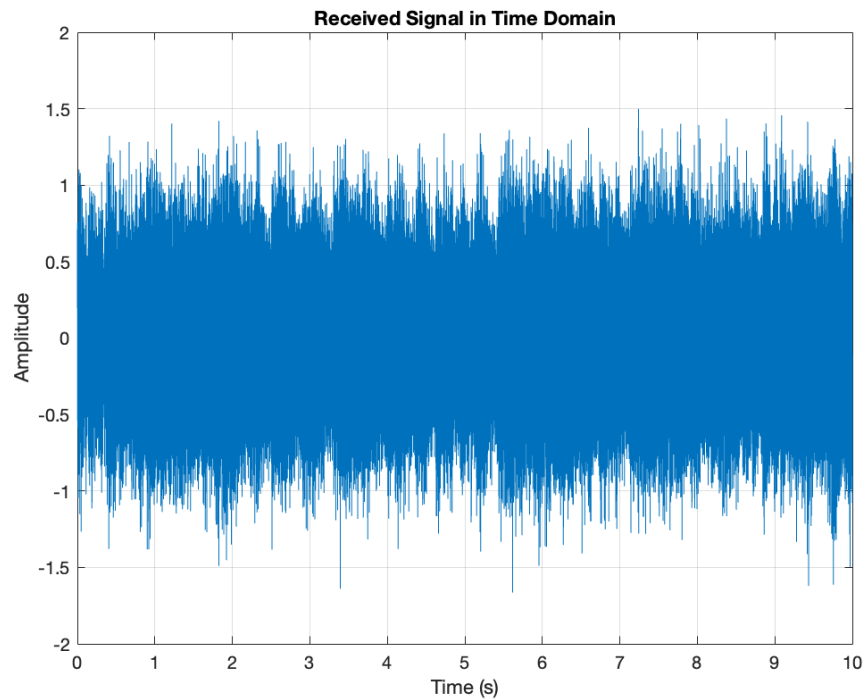


Figure 1: Plot 1: Received Signal in Time Domain.

Downsampling and playback revealed an unpleasant sounding signal due to the interference. To gain better insight, the signal was analyzed in the frequency domain using the Fast Fourier Transform (FFT). Plot 2 (Figure 2) shows 5 strong clusters of energy at the carrier frequencies of approximately 45 kHz, 90 kHz, 135 kHz, 180 kHz, and 225 kHz. There is no signal content within 20 kHz of $F = 0$.

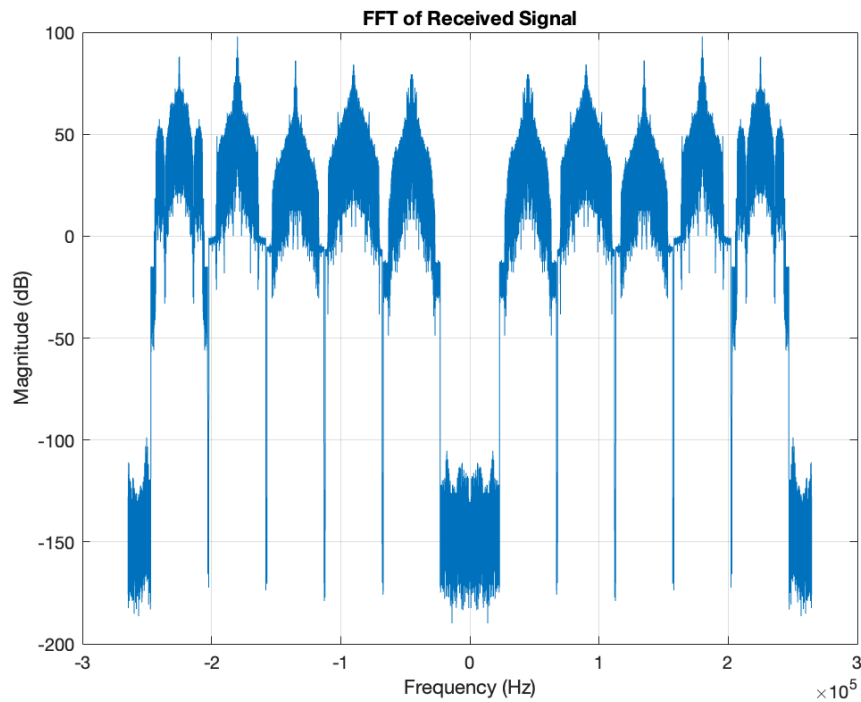


Figure 2: Plot 2: Spectrum of Received Signal.

The frequencies at which the five humps appear correspond directly to the carrier frequencies used in the dataset. Each hump is centered at one of the carrier frequencies, showing that each audio waveform was modulated onto its own unique carrier. A zoomed view around one carrier, such as 45 kHz (Plot 3, Figure 3), reveals that the hump around the carrier frequency is symmetric. The upper and lower sidebands are the same shape and extend an equal distance from the carrier. This symmetric structure occurs because the transmitted signal is real-valued, which forces the Fourier transform to be conjugate-symmetric. As a result, amplitude modulation produces mirror-image sidebands around each carrier frequency. Since the original audio signal is real-valued, its Fourier transform is conjugate-symmetric. If the signal is the product of the real cosine carrier, the spectrum changes to F_c and $-F_c$. The two shifted copies preserve the original conjugate symmetry, resulting in symmetric upper and lower sidebands around each carrier frequency. The hump around each carrier in the FFT is symmetric as modulation keeps the actual intrinsic symmetry of this real-valued signal.

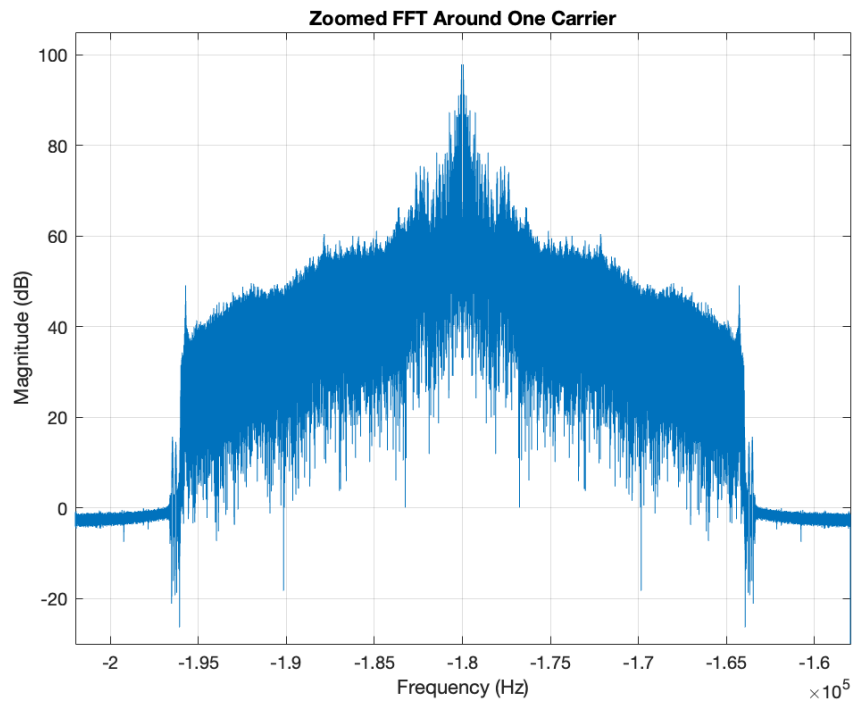


Figure 3: Plot 3: Zoomed Spectrum Around One Carrier.

For down-conversion, a carrier frequency was selected (e.g., $F_c = 45$ kHz), and the signal was multiplied by the cosine at that frequency. The spectrum of the down-converted signal (Plot 4, Figure 4) shows every signal as a hump centered around its carrier frequency. After down conversion, the hump of the chosen carrier is shifted down to $F=0$ while the other humps move to different frequencies. Unlike the RF spectrum, the down-converted FFT clearly shows strong energy at baseband, indicating that the selected channel has been brought back to audio frequencies. Yes, after down conversion, the selected carrier's spectrum is shifted to baseband, so there is now strong signal content centered at $F=0$.

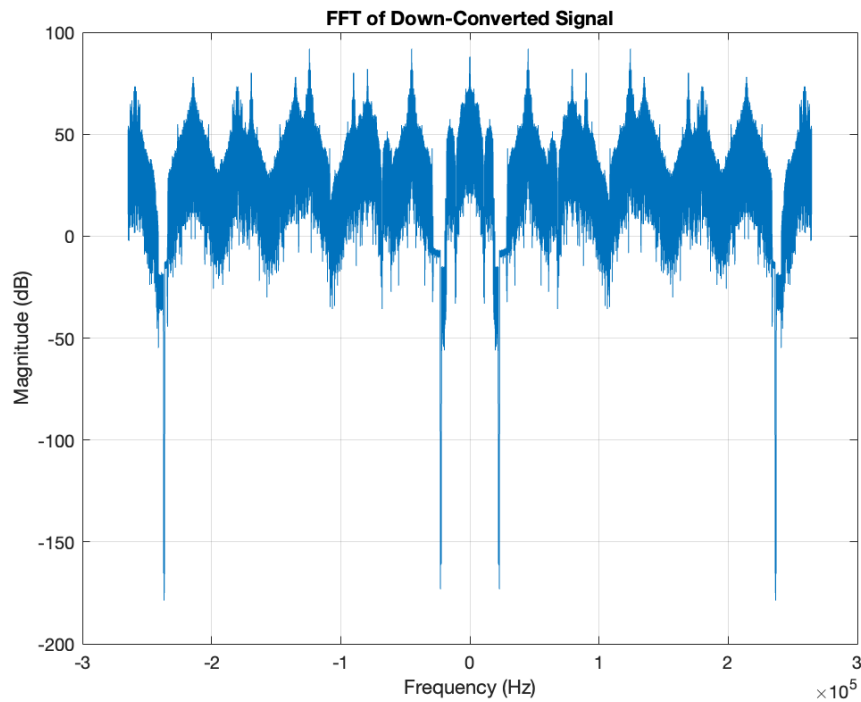
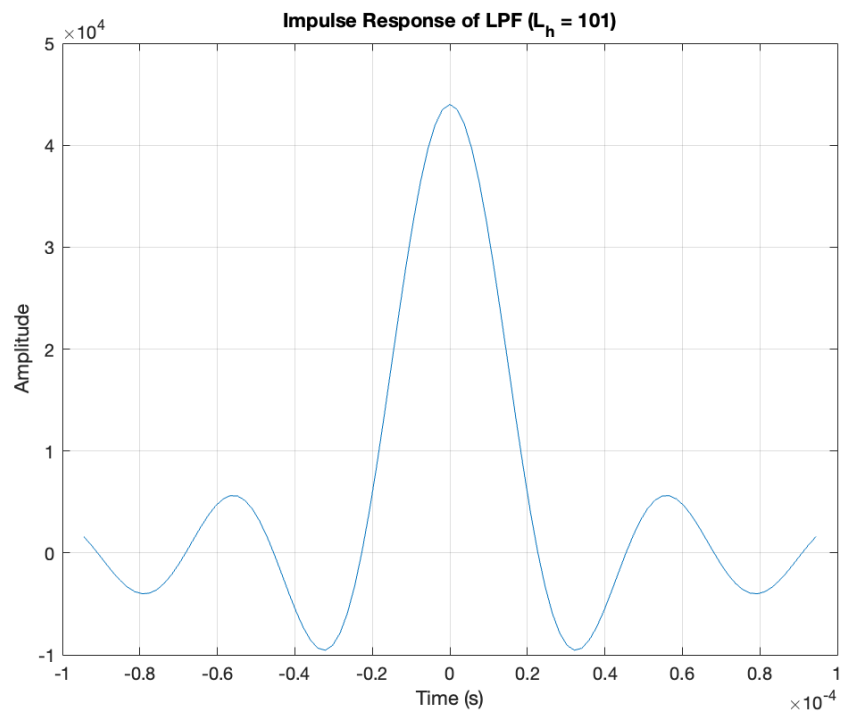
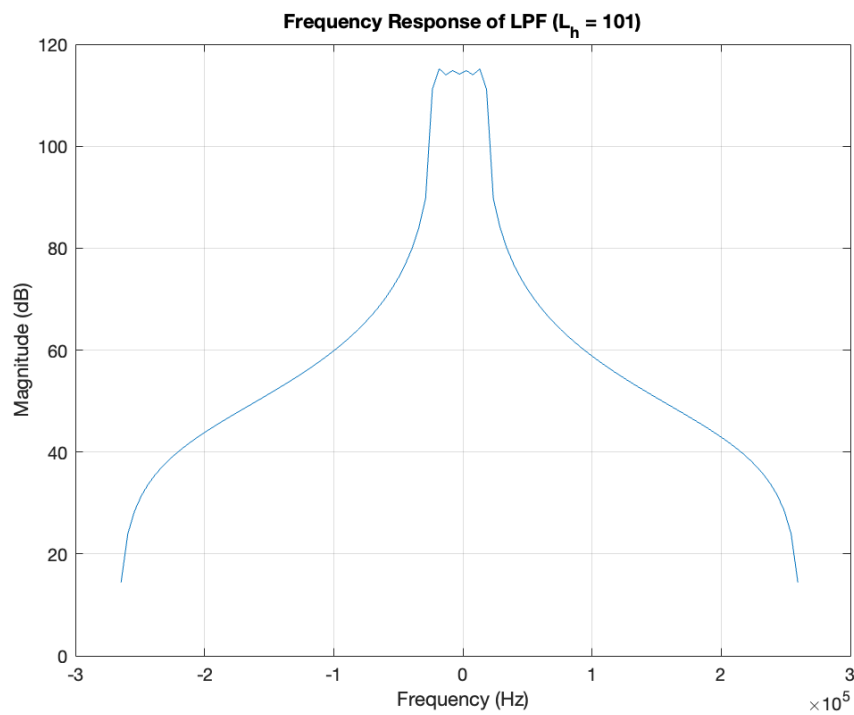


Figure 4: Plot 4: Spectrum of Down-Converted Signal.

Playback at this stage showed the audio can be heard faintly but is noisy because the signal still contains high-frequency components and interference from the other channels. The LPF has not been applied yet; hence, the audio is not clear. The low-pass filter was implemented with a truncated sinc impulse response. For $L_h = 101$, the impulse response is shown in Plot 5 (Figure 5), and its frequency response in Plot 6 (Figure 6). The frequency response $H(F)$ looks like a rectangular low-pass filter (a rect function) centered at $F=0$.

Figure 5: Plot 5: Truncated Impulse Response ($L_h = 101$).Figure 6: Plot 6: Frequency Response ($L_h = 101$).

Increasing to $L_h = 1001$ (Plot 7, Figure 7), the transition band begins to clear up and be much sharper bringing the LPF closer to its ideal function. The passband also becomes flatter, and the cutoff becomes steeper.

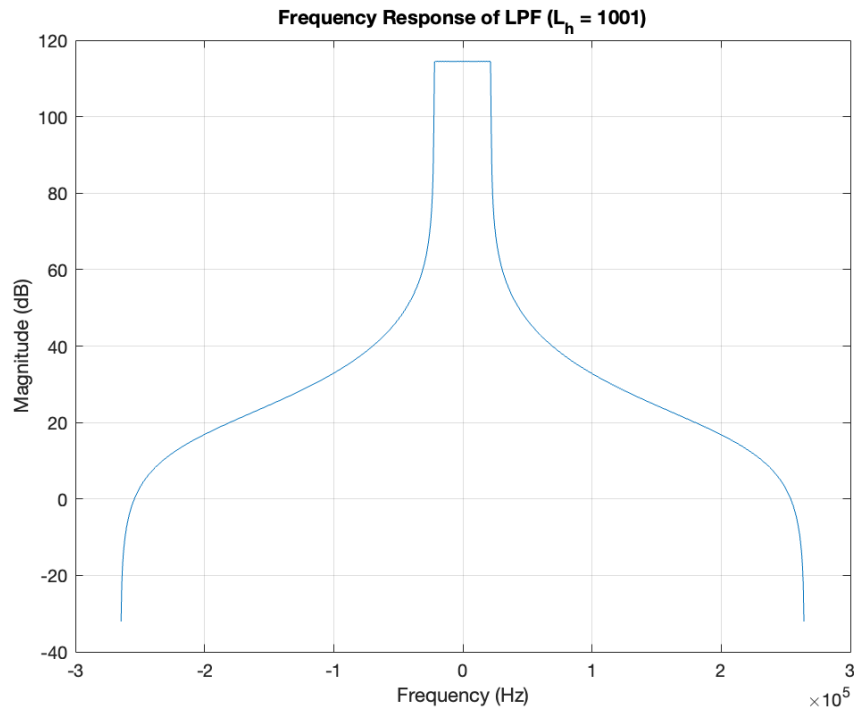


Figure 7: Plot 7: Frequency Response ($L_h = 1001$).

After convolution with the filter, the spectrum of the output (Plot 8, Figure 8) shows that after filtering, only the baseband audio content remains. The unwanted high-frequency terms and interference are removed, leaving a clean audio spectrum around $F = 0$.

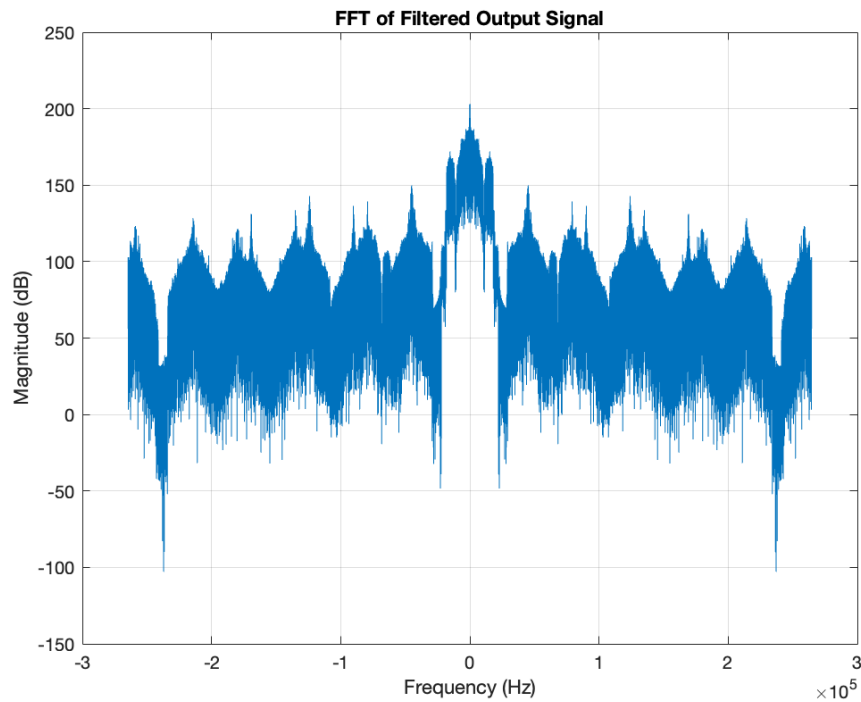


Figure 8: Plot 8: Spectrum of Filtered Output.

The resulting audio signal, after filtering and downsampling, is clear and easy to understand. For the final carrier F5, the recovered signal plays the clip of “Never Gonna Give You Up” by Rick Astley. Changing the carrier extracts different clips, showcasing the selectivity of the process.

4 Conclusion

This project looked through the basics of CW modulation. A signal is mixed through modulation so that it can be efficiently transmitted, then a received signal containing the transmitted signal and some interference is down-converted and filtered to retrieve the original signal. Part 1 used Fourier concepts and filtering to show the process of modulation and demodulation. The derivations revealed that audio signals are typically limited to 22kHz (the audible frequency range for a human), revealing that modulation requires carrier frequencies that are greater than 44kHz for efficient transmission. In Part 2, these modulation techniques were implemented in MATLAB to retrieve the 5 different audio clips which had been modulated using 5 different carrier frequencies.

A MATLAB Code

Listing 1: Complete MATLAB Script for Part 2

```

1 %% ECE 3793 - Project 2
2 % Griffin Hill and Jonathan Gross
3 clear; close all; clc;
4
5 % load project data
6 load("Project_2_Data.mat");

```

```

7
8 %% Step 1
9 % time interval for rx signal
10 Ts = 1/Fs;
11 t_1 = 0:Ts:(length(rx_data)-1)*Ts;
12
13 % Plot 1: Plot rx_data vs. t_1
14 figure;
15 plot(t_1, rx_data);
16 ylim([-2, 2]);
17 xlabel('Time (s)');
18 ylabel('Amplitude');
19 title('Received Signal in Time Domain');
20 grid on;
21
22 % downsample rx_data
23 % rx_downsample = rx_data(1:interp_factor:end);
24 % audiodevreset; % reset audio output device
25 % soundsc(rx_downsample,44100);
26
27 %% Step 2
28 % compute fft
29 X_RX = fftshift(fft(rx_data));
30
31 % Plot 2: Make a plot of the Fourier transform of the data vs. f_axis_rx
32 figure;
33 X_RX_db = mag2db(abs(X_RX));
34 f_axis_rx = -Fs/2 + (0:(length(X_RX)-1))*Fs/length(X_RX);
35 plot(f_axis_rx, X_RX_db);
36 xlabel('Frequency (Hz)');
37 ylabel('Magnitude (dB)');
38 title('FFT of Received Signal');
39 grid on;
40
41 % Plot 3: Make another plot of the FFT but zoomed in to be centered on
42 % one of the humps.
43 figure;
44 plot(f_axis_rx, X_RX_db);
45 xlim([-202000, -157950]);
46 ylim([-30, 105]);
47 xlabel('Frequency (Hz)');
48 ylabel('Magnitude (dB)');
49 title('Zoomed FFT Around One Carrier');
50 grid on;
51
52 %% Step 3
53 % carrier signal with Fc=Fn=n*carrier_spacing
54 n = 5;
55 Fc = n*carrier_spacing;
56 x_c = cos(2*pi*Fc*t_1);
57
58 % down-converted signal
59 x_d = rx_data.*x_c;
60
61 % compute fft
62 X_d = fftshift(fft(x_d));
63
64 % Plot 4: Plot the FFT of x_d
65 figure;

```

```

66 X_d_db = mag2db(abs(X_d));
67 f_axis_d = -Fs/2 + (0:(length(X_d)-1))*Fs/length(X_d);
68 plot(f_axis_d, X_d_db);
69 xlabel('Frequency (Hz)');
70 ylabel('Magnitude (dB)');
71 title('FFT of Down-Converted Signal');
72 grid on;
73
74 % playback downsampled audio signal
75 % x_d_downsample = x_d(1:interp_factor:end);
76 % audiodevreset; % reset audio output device
77 % soundsc(x_d_downsample, 44100);
78
79 %% Step 4
80 % new time interval
81 L_h = 101;
82 t_h = -Ts*((L_h-1)/2):Ts:Ts*((L_h-1)/2);
83
84 % impulse response
85 h = 44e3*sinc(44e3*t_h);
86
87 % Plot 5: Make a plot of your impulse response for L_h = 101
88 figure;
89 plot(t_h, h);
90 xlabel('Time (s)');
91 ylabel('Amplitude');
92 title('Impulse Response of LPF (L_h = 101)');
93 grid on;
94
95 % frequency response using fft
96 H = fftshift(fft(h));
97
98 % Plot 6: Plot the magnitude of H in dB
99 H_db = mag2db(abs(H));
100 f_axis_h = -Fs/2 + (0:(L_h-1))*Fs/L_h;
101 figure;
102 plot(f_axis_h, H_db);
103 xlabel('Frequency (Hz)');
104 ylabel('Magnitude (dB)');
105 title('Frequency Response of LPF (L_h = 101)');
106 grid on;
107
108 % redefine h with L_h = 1001
109 L_h = 1001;
110 t_h = -Ts*((L_h-1)/2):Ts:Ts*((L_h-1)/2);
111 h = 44e3*sinc(44e3*t_h);
112 H = fftshift(fft(h));
113 H_db = mag2db(abs(H));
114 f_axis_h = -Fs/2 + (0:(L_h-1))*Fs/L_h;
115
116 % Plot 7: Make another plot of the new H in dB
117 figure;
118 plot(f_axis_h, H_db);
119 xlabel('Frequency (Hz)');
120 ylabel('Magnitude (dB)');
121 title('Frequency Response of LPF (L_h = 1001)');
122 grid on;
123
124 %% Step 5

```

```
125 % output signal
126 x_out = conv(x_d, h);
127
128 % fft of output
129 X_out = fftshift(fft(x_out));
130
131 % Plot 8: Plot the magnitude of the FFT of x_out in dB
132 X_out_db = mag2db(abs(X_out));
133 f_axis_out = -Fs/2 + (0:(length(X_out)-1))*Fs/length(X_out);
134 figure;
135 plot(f_axis_out, X_out_db);
136 xlabel('Frequency (Hz)');
137 ylabel('Magnitude (dB)');
138 title('FFT of Filtered Output Signal');
139 grid on;
140
141 % playback downsampled audio signal
142 x_out_downsample = x_out(1:interp_factor:end);
143 audiodevreset; % reset audio output device
144 soundsc(x_out_downsample, 44100);
```