A Description of Project 2

Zhou Kaining 3180101148

1 General Introduction

This whole project, lab assignment #2, was accomplished by me alone. The Matlab code and files are in the .rar file titled as '3180101148 Zhoukaining Lab2'. The structure of code is as follows.

For Problem 1, 2 and 3, I integrate the code into one .m file, titled as 'main.m' in the directory named 'Project 2', which is the main script. For the sake of a better presentation effect, I divide the code into coherent, relevant but relatively independent code sections which begin with '%%'.

Run the script 'main.m' for all problems. Please run the code section by section from the first one, then sequentially, in their current directory. The output is in the form of graphics and characters in the command line as well as the files.

2 Problem 1

In the lecture, we have studied Fourier transform in theory. So I try to implement Fourier transform to analyze certain sound data.

I record my own sound by phone in a file 'speech.m4a', which is the object signal to be analyzed. For sound in .m4a format, sampling frequency is 48000, and the number of track is 2 by default. First I plot the sound wave's doubled amplitude with respect to time, as Figure 1 shows, then I utilize the built-in functions, **fft** and **fftshift** to illustrate the sound signal in frequency domain shown in Figure 2.

Then I use filter to get the low and high frequency components respectively. First, I set the threshold of the division to be 4000Hz. Then I use a built-in function **fir1** to get the FIR filter with designate arguments. Figure 3 illustrates the normalized frequency response of the lowpass filter. After setting a certain type of filter, I employ it on the signal to obtain the low frequency components. Filtered signal is presented in time domain (Figure 4) and frequency domain (Figure 5).

By observing the original and lowpass filtered versions of the signal, we cannot see there are major differences, which indicates the dominant frequency is the low frequency. To compare with this result, I conduct the highpass filter experiment.

The only difference between two filters is that I replace argument 'low' with 'high' in function **fir1**. However, the result is quite distinct from the previous one. As shown in Figure 6, 7 and 8, the filtered signal changes its features largely. In time domain diagram, data look more sparse and the average amplitude reduces noticeably. In frequency domain diagram, we can see the average energy diminishes two orders of magnitude. That manifests that the high frequency component is minor. I also write two resulting data in two separate files 'Highpass version.m4a' and 'Lowpass version.m4a'.

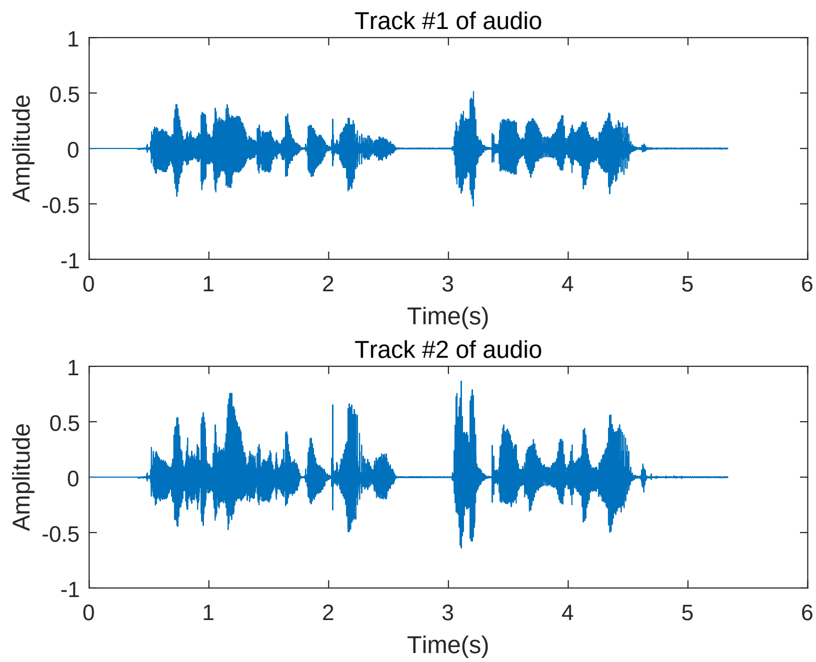

Figure 1: The input signal's wave.

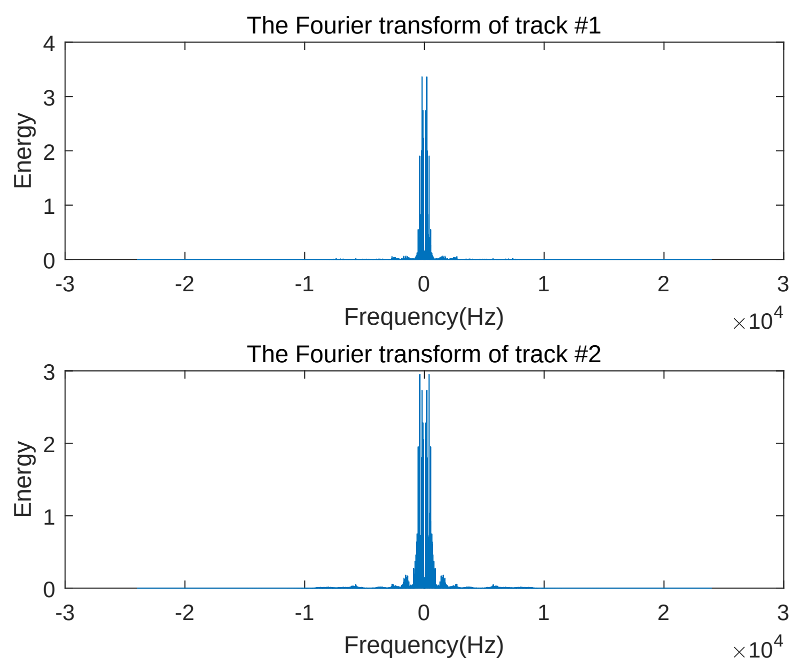

Figure 2: The input signal's frequency energy.

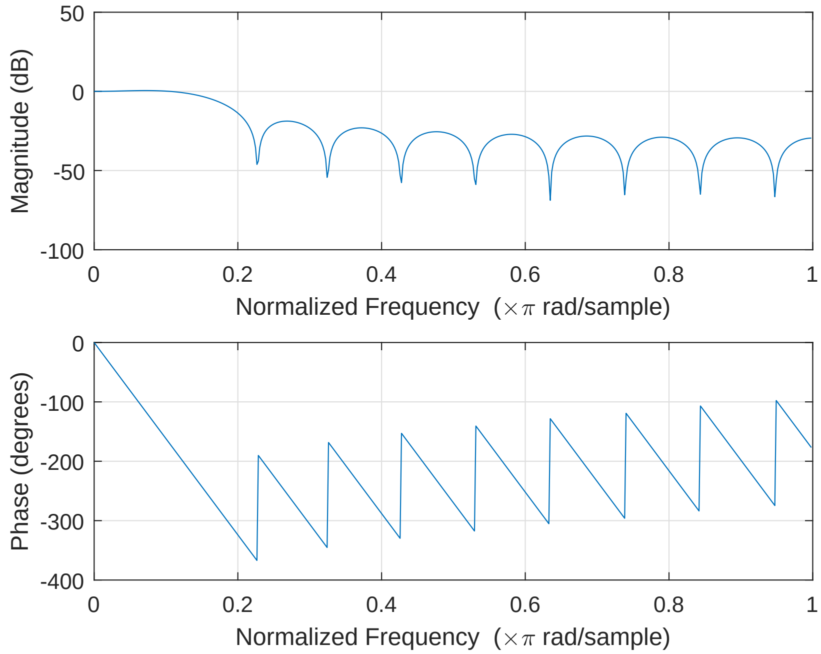

Figure 3: The bode plot of the lowpass filter.

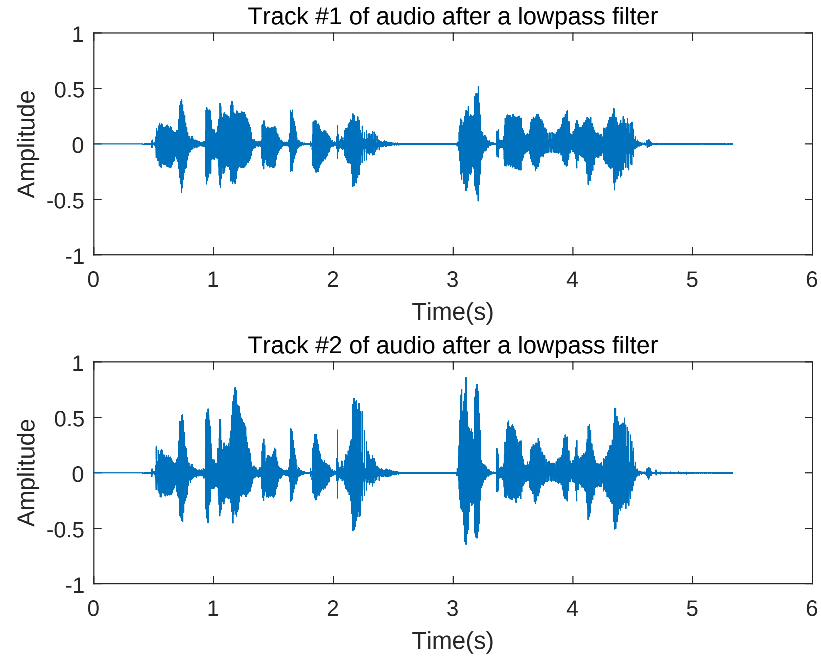

Figure 4: The signal after a lowpass filter displayed in time domain.

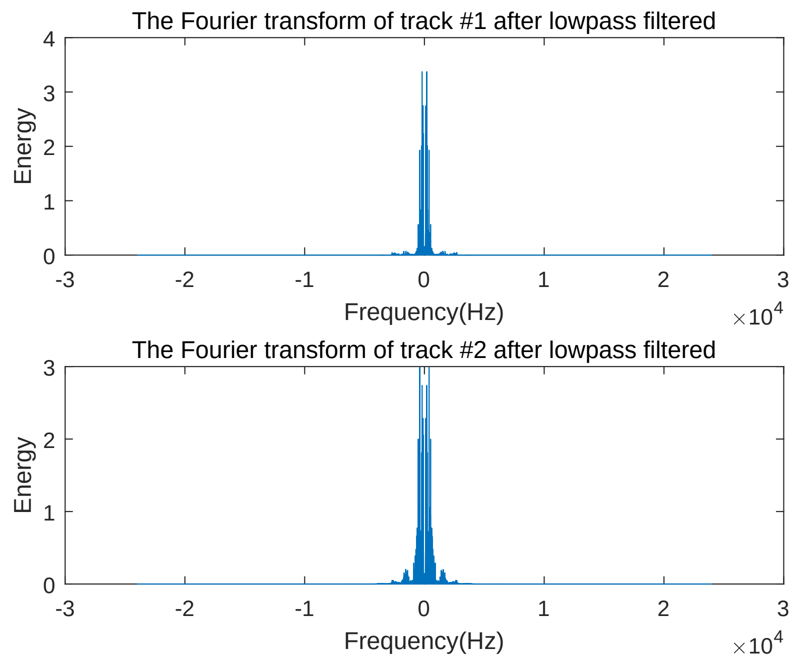

Figure 5: The signal after a lowpass filter displayed in frequency domain.

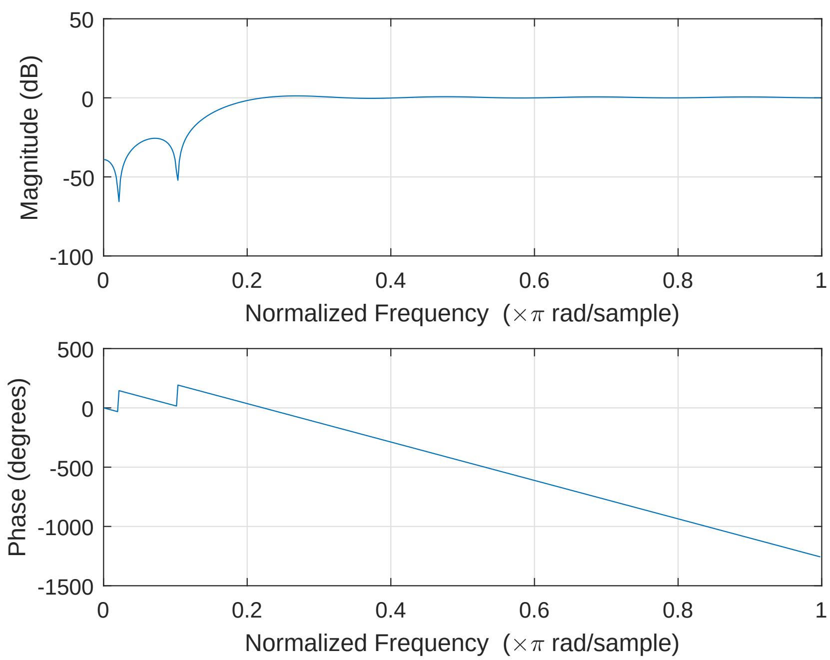

Figure 6: The bode plot of the highpass filter.

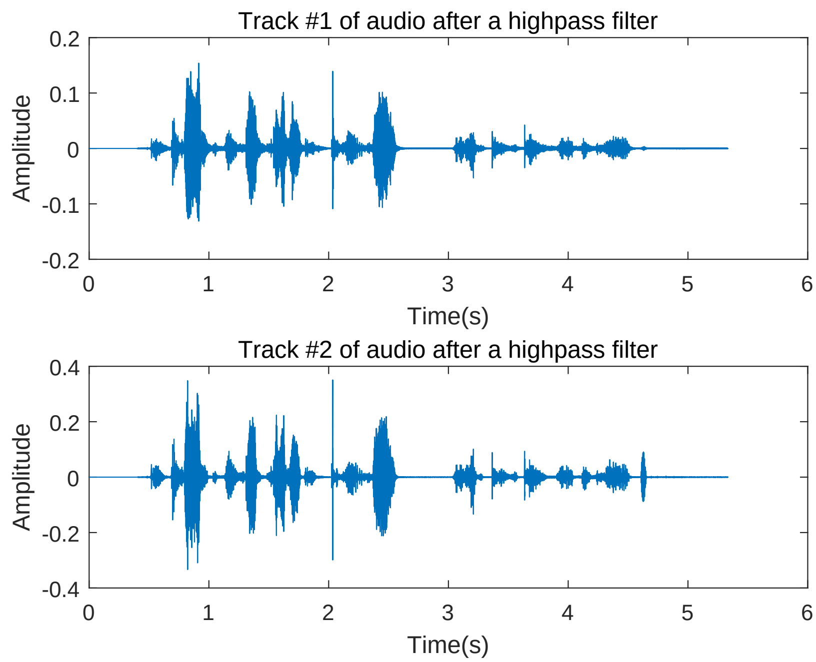

Figure 7: The signal after a highpass filter displayed in time domain.

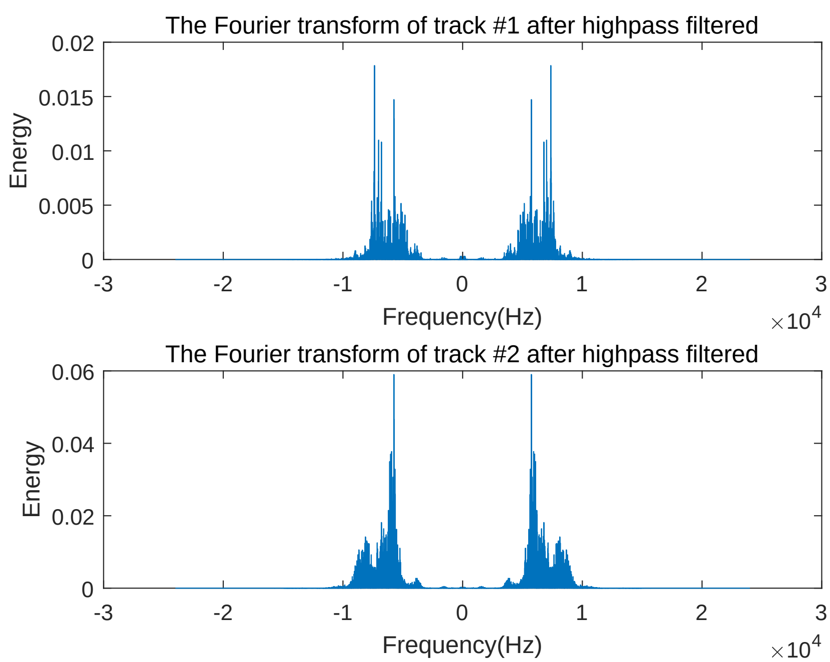

Figure 8: The signal after a highpass filter displayed in frequency domain.

3 Problem 2

3.1 Basic Problems

(a). Firstly the waveform of the original signal is shown in Figure 9. Perform Z transform on Eq. (1.1), we obtain that the system function is

$$H(z) = 1 + \alpha z^{-N}, \tag{1}$$

then I perform convolution operation to get the impulse response which is plotted in Figure 10.

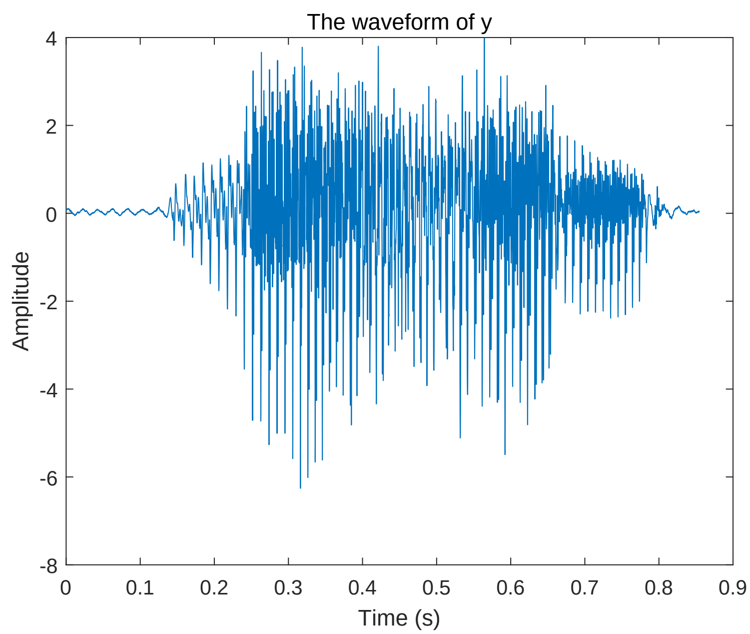

Figure 9: The original input signal in Problem 2.

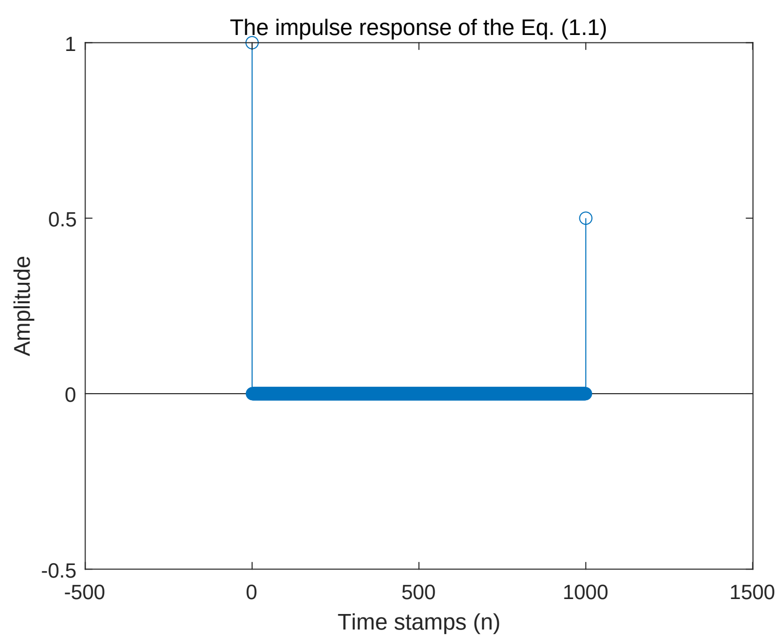

Figure 10: The impulse response of Eq. (1.1)

(b). We can obtain the system function of difference equation Eq. (1.2) with Z transform:

$$H_1(z) = \frac{1}{1 + \alpha z^{-N}}. \tag{2}$$

Multiply Eq. (1) by Eq. (2), we obtain that the overall system function is 1, so $z[n] = x[n]$ is a valid solution to the overall difference equation.

3.2 Intermediate Problems

(c). Store the finite-length impulse following the instruction. The built-in function **filter** takes in three arguments, the first and second arguments compose the function system's numerator and denominator respectively, the third is input. The impulse response of Eq. (1.2) is shown in Figure 11.

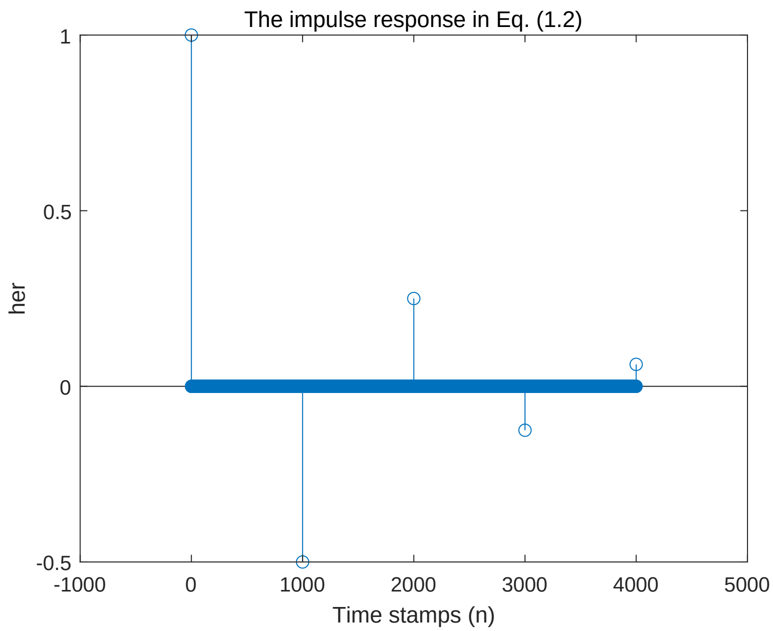

Figure 11: The impulse response of Eq. (1.2)

(d). Now we use the filter determined by Eq. (1.2) to remove the echo of y. The filtered version of sound is presented in Figure 12.

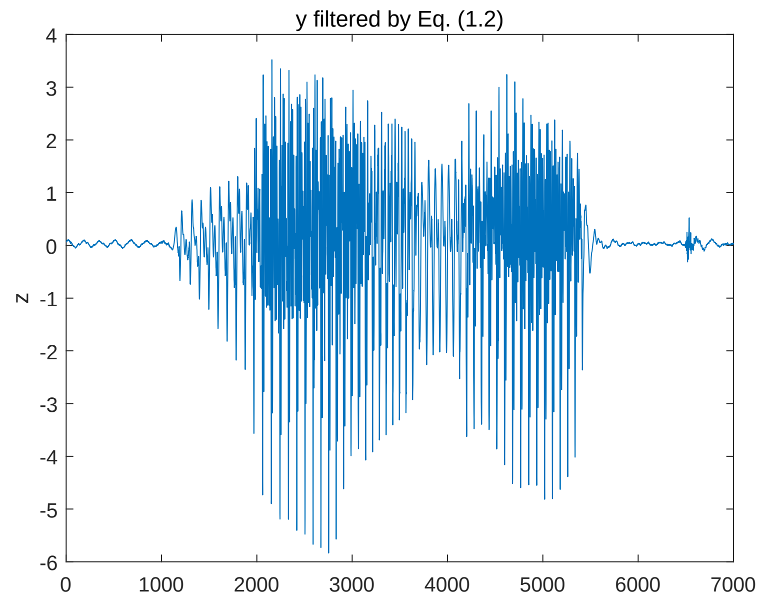

Figure 12: The original sound without echo.

(e). The convolution of **he** and **she** is illustrated in Figure 13. The result, **hoa**, is not the ideal impulse signal. When $n = 5001$, $hoa[n] = 0.0313$. It is because that the input impulse signal and the filter are not infinite, so though in theory the output of the overall impulse response of the cascaded echo system is unit impulse, it is not so in practice.

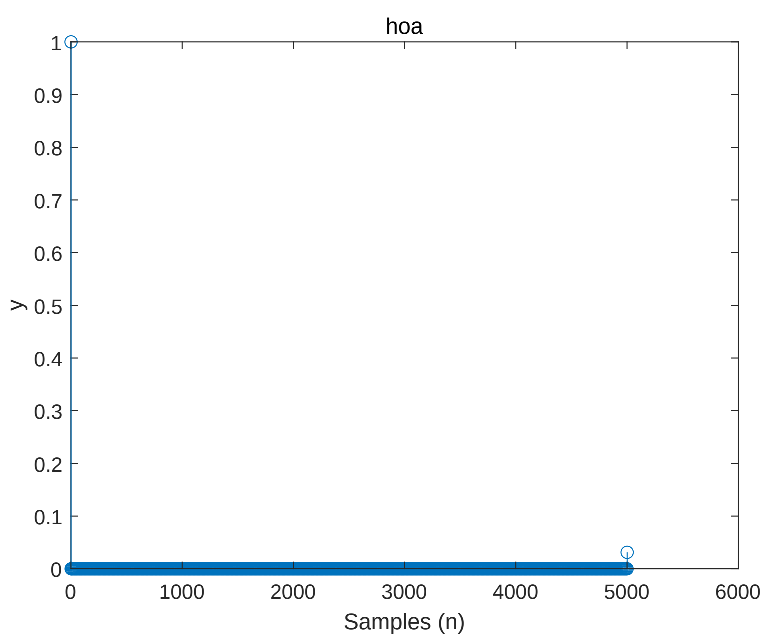

Figure 13: The original sound without echo.

3.3 Advanced Problem

(f). Starting from the given formula, I obtain R_{yy} with respect to R_{xx} as follows.

$$\begin{aligned}
y[n] &= x[n] * (\delta[n] + \alpha\delta[n-N]) \\
R_{yy}[n] &= y[n] * y[-n] \\
&= x[n] * x[-n] * (\delta[n] + \alpha\delta[n-N]) * (\delta[-n] + \alpha\delta[-n-N]) \\
&= R_{xx}[n] * (\delta[n] + \alpha\delta[n-N]) * (\delta[n] + \alpha\delta[n+N]) \\
&= R_{xx}[n] * \left(\left(1+\alpha^2\right)\delta[n] + \alpha\delta[n-N] + \alpha\delta[n+N]\right)
\end{aligned}$$

So if a sound has one echo, we can describe it by using following equation:

$$R_{yy}[n] = \left(1+\alpha^2\right) R_{xx}[n] + \alpha R_{xx}[n-N] + \alpha R_{xx}[n+N] \tag{3}$$

Suppose that we do not have information about R_{xx}, what we have is $y[n]$. So here I am going to estimate the amplitude α and delay N. I plot the autocorrelation of **y2** in Figure 14.

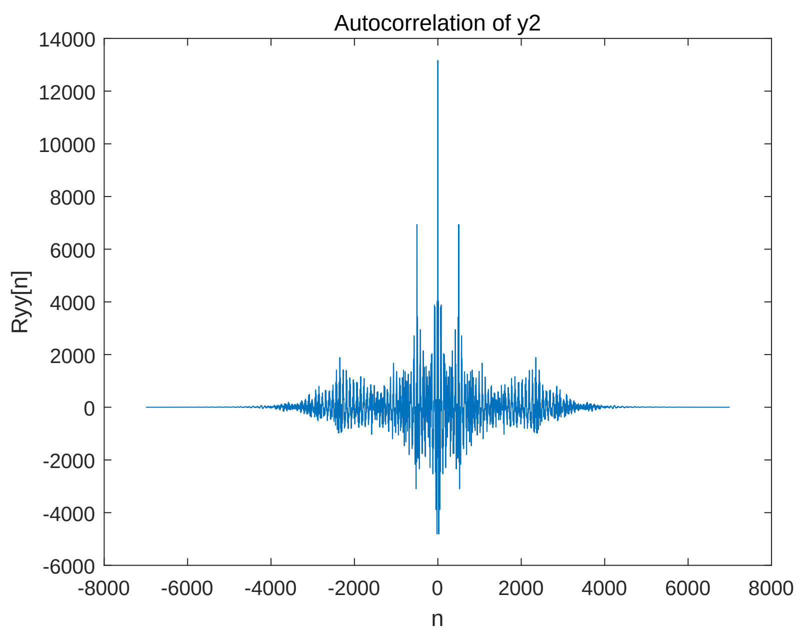

Figure 14: Autocorrelation of y2.

Then I use a built-in function **findpeaks** to find peaks and their locations which appear in Figure 14. So I can get the latency, $N = 501$. One important thing that should be noticed is that I assume $R_{xx}[0] = \beta R_{xx}[N] \neq 0, R_{xx}[2N] = R_{xx}[3N] = 0$ because that the autocorrelation is obvious when time stamps are near 0. Based on this, I set $n = 0, n = N, n = 2N$ in Eq. (3) respectively, then I get the following equations:

$$R_{yy}[0] = \left(1+\alpha^2\right) R_{xx}[0] + 2\alpha R_{xx}[N], \tag{4}$$

$$R_{yy}[N] = \left(1+\alpha^2\right) R_{xx}[N] + \alpha R_{xx}[0] + \alpha R_{xx}[2N], \tag{5}$$

$$R_{yy}[2N] = \left(1+\alpha^2\right) R_{xx}[2N] + \alpha R_{xx}[N] + \alpha R_{xx}[3N]. \tag{6}$$

Then I divide Eq. (6) by Eq. (5) and I get

$$\begin{aligned}\frac{R_{yy}[2N]}{R_{yy}[N]} &= \frac{\left(1+\alpha^2\right) R_{xx}[2N] + \alpha R_{xx}[N] + \alpha R_{xx}[3N]}{\left(1+\alpha^2\right) R_{xx}[N] + \alpha R_{xx}[0] + \alpha R_{xx}[2N]} \\ &\approx \frac{\alpha R_{xx}[N]}{\left(1+\alpha^2\right) R_{xx}[N] + \alpha R_{xx}[0]} \\ &= \frac{\alpha\beta}{\left(1+\alpha^2\right)\beta + \alpha}.\end{aligned} \tag{7}$$

Compared to α, β is relatively small. So Eq. (7) can be further reduced to

$$\frac{R_{yy}[2N]}{R_{yy}[N]} \approx \beta, \tag{8}$$

Then I divide Eq. (5) by Eq. (4) and obtain

$$\begin{aligned}\frac{R_{yy}[N]}{R_{yy}[0]} &= \frac{\left(1+\alpha^2\right) R_{xx}[N] + \alpha R_{xx}[0] + \alpha R_{xx}[2N]}{\left(1+\alpha^2\right) R_{xx}[0] + 2\alpha R_{xx}[N]} \\ &\approx \frac{\left(1+\alpha^2\right) R_{xx}[N] + \alpha R_{xx}[0]}{\left(1+\alpha^2\right) R_{xx}[0] + 2\alpha R_{xx}[N]} \\ &= \frac{\left(1+\alpha^2\right)\beta + \alpha}{\left(1+\alpha^2\right) + 2\alpha\beta}.\end{aligned} \tag{9}$$

From Eq. (7) and Eq. (9) I can solve the equations and find $\alpha \approx 0.8416$ ($\frac{R_{yy}[2N]}{R_{yy}[N]}$ and $\frac{R_{yy}[N]}{R_{yy}[0]}$ can be easily computed by given data).

Similarly, for **y3** (its autocorrelation is depicted in Figure 15), we know that

$$\begin{aligned}y[n] &= x[n] * \left(\delta[n] + \alpha_1\delta[n-N_1] + \alpha_2\delta[n-N_2]\right) \\ R_{yy}[n] &= y[n] * y[-n] \\ &= x[n] * x[-n] * \left(\delta[n] + \alpha_1\delta[n-N_1] + \alpha_2\delta[n-N_2]\right) * \left(\delta[-n] + \alpha_1\delta[-n-N_1] + \alpha_2\delta[-n-N_2]\right) \\ &= R_{xx}[n] * \left(\delta[n] + \alpha_1\delta[n-N_1] + \alpha_2\delta[n-N_2]\right) * \left(\delta[n] + \alpha_1\delta[n+N_1] + \alpha_2\delta[n+N_2]\right) \\ &= R_{xx}[n] * \{\left(1+{\alpha_1}^2+{\alpha_2}^2\right)\delta[n] + \alpha_1\delta[n-N_1] + \alpha_1\delta[n+N_1] + \alpha_2\delta[n-N_2] + \alpha_2\delta[n+N_2] + \\ &\quad \alpha_1\alpha_2\delta[n-N_1+N_2] + \alpha_1\alpha_2\delta[n-N_2+N_1]\}\end{aligned}$$

So if a sound has two echos, we can obtain its correlation by following equation:

$$\begin{aligned}R_{yy}[n] = &\left(1+{\alpha_1}^2+{\alpha_2}^2\right) R_{xx}[n] + \alpha_1\left(R_{xx}[n-N_1] + R_{xx}[n+N_1]\right) + \alpha_2\left(R_{xx}[n-N_2] + R_{xx}[n+N_2]\right) + \\ &\alpha_1\alpha_2\left(R_{xx}[n-N_1+N_2] + R_{xx}[n-N_2+N_1]\right).\end{aligned} \tag{10}$$

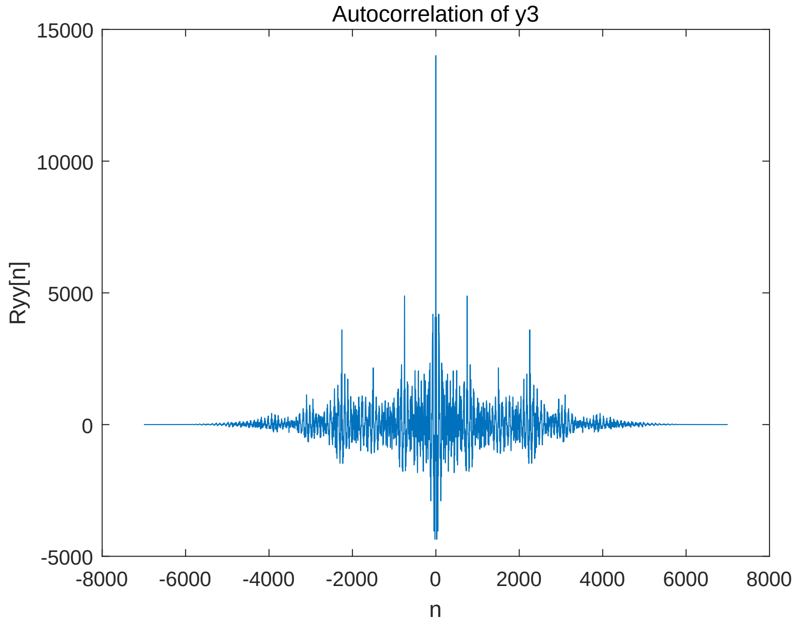

Figure 15: Autocorrelation of y3.

We can see that it has several high peaks at $n = 0, n = |N_1|, n = |N_2|, n = |N2 - N1|$. Then I set $n = 0, n = N_1, n = N_2, n = N_2 - N_1$ and have

$$R_{yy}[0] = \left(1 + {\alpha_1}^2 + {\alpha_2}^2\right) R_{xx}[0] + 2\alpha_1 R_{xx}[N_1] + 2\alpha_2 R_{xx}[N_2] + 2\alpha_1\alpha_2 R_{xx}[N_1 - N_2], \tag{11}$$

$$\begin{aligned} R_{yy}[N_1] = &\left(1 + {\alpha_1}^2 + {\alpha_2}^2\right) R_{xx}[N_1] + \alpha_1 \left(R_{xx}[0] + R_{xx}[2N_1]\right) + \alpha_2 \left(R_{xx}[N_1 - N_2] + R_{xx}[N_1 + N_2]\right) + \\ &\alpha_1\alpha_2 \left(R_{xx}[N_2] + R_{xx}[2N_1 - N_2]\right), \end{aligned} \tag{12}$$

$$\begin{aligned} R_{yy}[N_2] = &\left(1 + {\alpha_1}^2 + {\alpha_2}^2\right) R_{xx}[N_2] + \alpha_1 \left(R_{xx}[N_2 - N_1] + R_{xx}[N_2 + N_1]\right) + \alpha_2 \left(R_{xx}[0] + R_{xx}[2N_2]\right) + \\ &\alpha_1\alpha_2 \left(R_{xx}[2N_2 - N_1] + R_{xx}[N_1]\right), \end{aligned} \tag{13}$$

$$\begin{aligned} R_{yy}[N_2 - N_1] = &\left(1 + {\alpha_1}^2 + {\alpha_2}^2\right) R_{xx}[N_2 - N_1] + \alpha_1 \left(R_{xx}[N_2 - 2N_1] + R_{xx}[N_2]\right) + \alpha_2 \left(R_{xx}[-N_1] + R_{xx}[2N_2 - N_1]\right) + \\ &\alpha_1\alpha_2 \left(R_{xx}[2N_2 - 2N_1] + R_{xx}[0]\right). \end{aligned} \tag{14}$$

Now I am going to establish approximate treatment means on these equations in order to determine α_1 and α_2. From the plot we can find $N_1 = 751$, $N_2 = 2252$. Generally speaking, due to the properties of autocorrelation, the farther a sample is from 0, the closer it is to 0. What is more, I suppose that there is approximate and linearity in R_{xx}. So I set

$$R_{xx}[n] \approx 0, \quad \forall |n| \geq N_2$$

$$R_{xx}[N_2 - N_1] \approx R_{xx}[N_2 - 2N_1] \approx R_{xx}[2N_1] \approx \frac{R_{xx}[N_1]}{2}$$

With these approximations, I can further reduce Eq. (11)~(14) to following ones:

$$R_{yy}[0] \approx \left(1 + {\alpha_1}^2 + {\alpha_2}^2\right) R_{xx}[0] + 2\alpha_1 R_{xx}[N_1], \tag{15}$$

$$R_{yy}[N_1] \approx \left(1 + {\alpha_1}^2 + {\alpha_2}^2\right) R_{xx}[N_1] + \alpha_1 \left(R_{xx}[0] + \frac{R_{xx}[N_1]}{2}\right) + \alpha_2 \frac{R_{xx}[N_1]}{2}, \tag{16}$$

$$R_{yy}[N_2] \approx \alpha_1 \frac{R_{xx}[N_1]}{2} + \alpha_2 R_{xx}[0] + \alpha_1\alpha_2 R_{xx}[N_1], \tag{17}$$

$$R_{yy}[N_2 - N_1] \approx \left(1 + {\alpha_1}^2 + {\alpha_2}^2\right) \frac{R_{xx}[N_1]}{2} + \alpha_1 \frac{R_{xx}[N_1]}{2} + \alpha_2 R_{xx}[N_1] + \alpha_1\alpha_2 R_{xx}[0]. \tag{18}$$

Now we can see Eq. (15)~(18) compose a set of equations with the highest degree of 3, with respect to variable α_1, α_2, $R_{xx}[0]$ and $R_{xx}[N_1]$. I solve the equations with Matlab and select one set of reasonable solution where $\alpha_1 = 0.7786$, $\alpha_2 = 0.5028$.

In conclusion, I estimate the amplitude and delay by setting a range of values to be zero then solving the equations. And I deduce that for vector **y2**, $N = 501$ and $\alpha = 0.8416$; for **y3**, $N_1 = 751$, $\alpha_1 = 0.7786$, $N_2 = 2252$ and $\alpha_2 = 0.5028$.

4 Problem 3

4.1 Basic Problems

(a). Store signal **z=[dash dash dot dot]** and plot it in Figure 16.

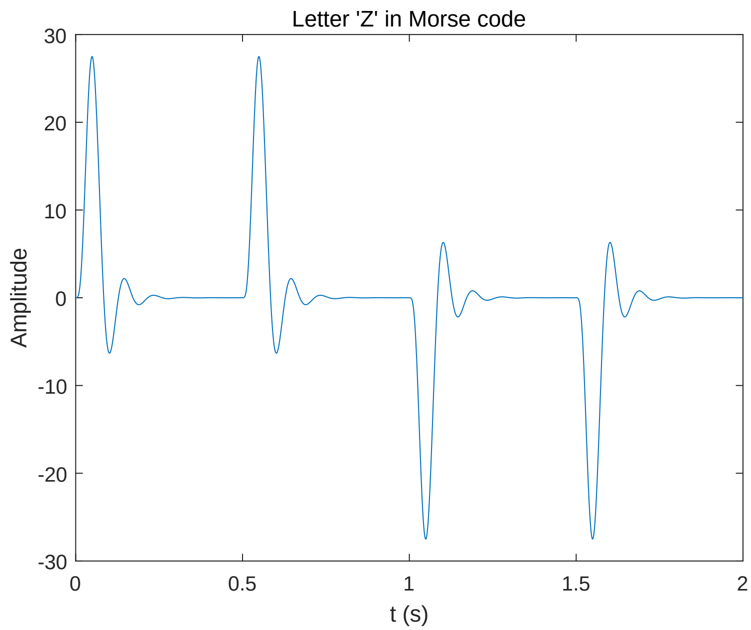

Figure 16: ‘Z’ in Morse code.

(b). Plot the frequency response of the filter using the built-in function **freqs** in Figure 17.

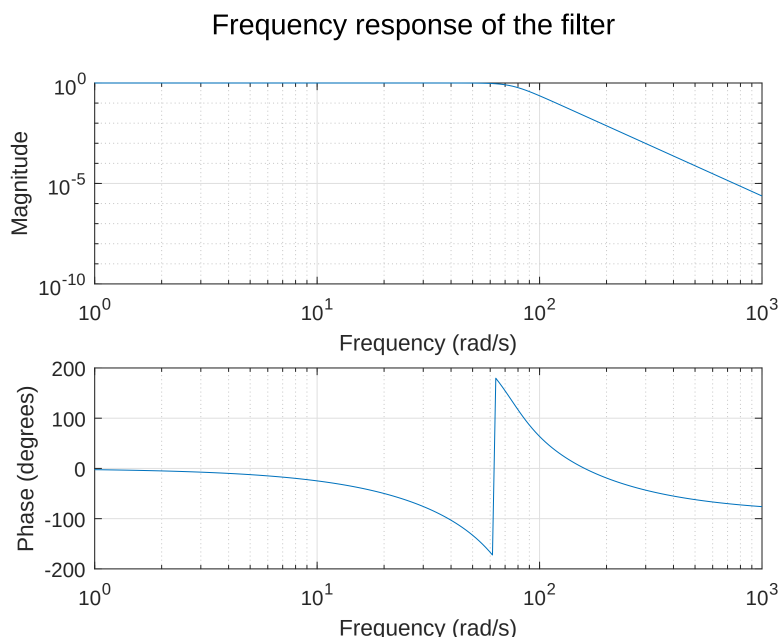

Figure 17: Frequency response.

(c). Plot the outputs **ydash** and **ydot** along with the original signal **dash** and **dot** in Figure 18.

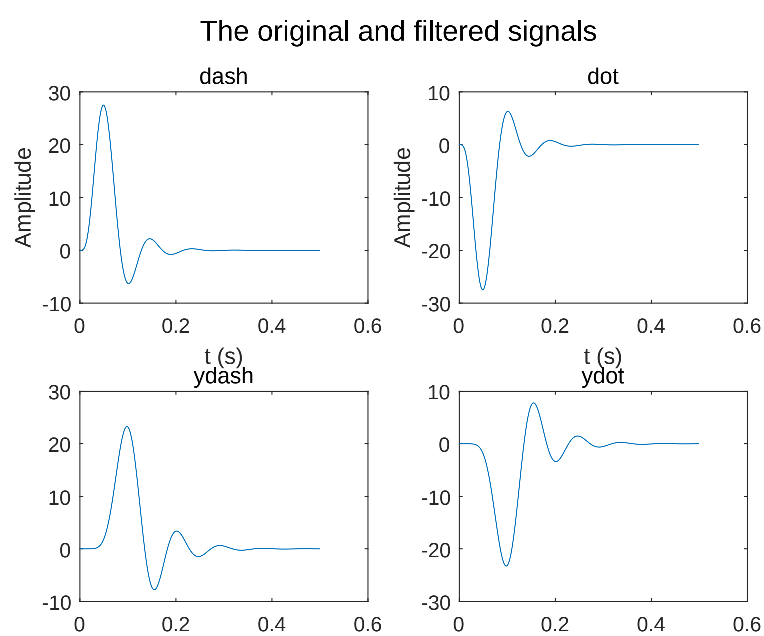

Figure 18: The original and filtered signal.

(d). Plot the modulated signal and its output in Figure 19.

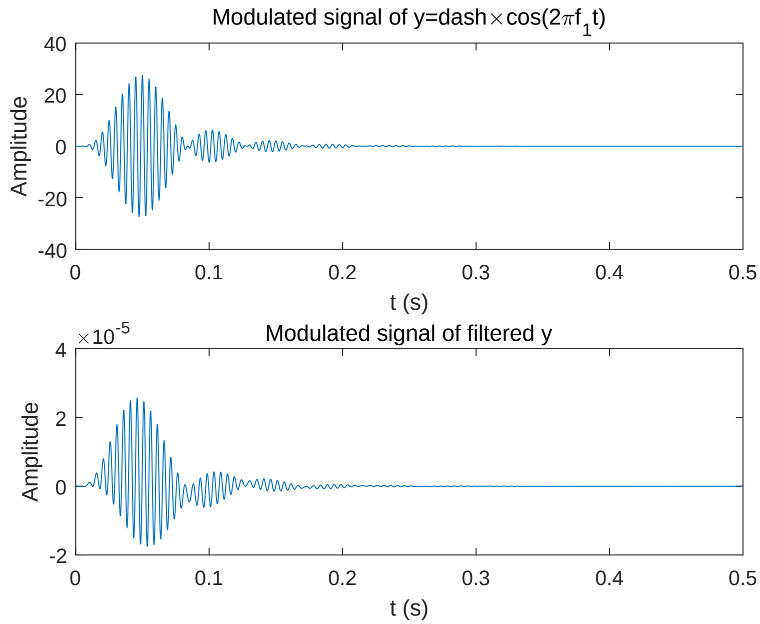

Figure 19: The modulated signal and its output.

I get a result that I would have expected. From the Figure we can find that after modulated and filtered, than amplitude decreases by 6 orders of magnitude, it proves that the energy moves outside the passband of the filter.

4.2 Intermediate Problems

(e). Let

$$\mathrm{y}_1(t) = m(t)\cos(2\pi f_1 t)\cos(2\pi f_1 t)\,, \tag{19}$$

$$\mathrm{y}_2(t) = m(t)\cos(2\pi f_1 t)\sin(2\pi f_1 t)\,, \tag{20}$$

$$\mathrm{y}_3(t) = m(t)\cos(2\pi f_1 t)\cos(2\pi f_2 t)\,, \tag{21}$$

we can determine their Fourier transform respectively.

$$\mathrm{Y}_1(j\omega) = \frac{1}{4}M\left(j\left(\omega - 4\pi f_1\right)\right) + \frac{1}{2}M\left(j\omega\right) + \frac{1}{4}M\left(j\left(\omega + 4\pi f_1\right)\right) \tag{22}$$

$$\mathrm{Y}_2(j\omega) = \frac{1}{4j}M\left(j\left(\omega - 4\pi f_1\right)\right) - \frac{1}{4j}M\left(j\left(\omega + 4\pi f_1\right)\right) \tag{23}$$

$$\mathrm{Y}_3(j\omega) = \frac{1}{4}M\left(j\left(\omega - 2\pi\left(f_1 + f_2\right)\right)\right) + \frac{1}{4}M\left(j\left(\omega - 2\pi\left(f_1 - f_2\right)\right)\right) + \frac{1}{4}M\left(j\left(\omega + 2\pi\left(f_1 - f_2\right)\right)\right) + \frac{1}{4}M\left(j\left(\omega + 2\pi\left(f_1 + f_2\right)\right)\right) \tag{24}$$

(f). Because in Eq. (2.4), $m_1(t)$ is modulated by $\cos(2\pi f_1 t)$, so I multiply $x(t)$ by $\cos(2\pi f_1 t)$, then demodulate the signal with filter, the result is $m_1(t)$ plotted in the top subfigure in Figure 20. The first letter's Morse code is **dash dot dot**, which indicates 'D'.

(g). Demodulate $x(t)$ in the similar way to obtain $m_2(t)$ and $m_3(t)$. The results are shown in the second and third subfigure in Figure 20.

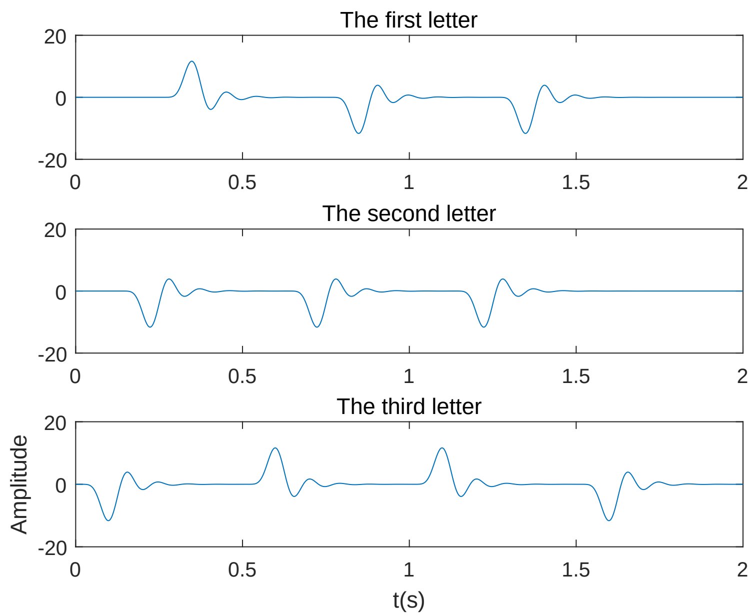

Figure 20: Three demodulation signals representing three letters.

The second letter's Morse code is **dot dot dot**, which indicates 'S'. The third one is **dot dash dash dot**, which indicates 'P'.

So the future of technology lies on DSP.

5 Contribution

Contributor: Zhou Kaining.