# CENG 384 - Signals and Systems for Computer Engineers
# Spring 2023
# Homework 4

Meyvecioğlu, Hasan Ege
e2449783@ceng.metu.edu.tr

Şanlı, Enes
e2375749@ceng.metu.edu.tr

June 7, 2023

---

1. (a)

$$H(jw) = \frac{Y(jw)}{X(jw)} \tag{1}$$

$$(jw+1)Y(jw) = (jw-1)X(jw) \tag{2}$$

$$y'(n) + y(n) = x'(n) - x(n) \tag{3}$$

(b)

$$H(jw) = \frac{jw-1}{jw+1} \tag{4}$$

$$H(jw) = 1 - \frac{2}{jw+1} \tag{5}$$

$$h(t) = F^{-1}\{H(jw)\} = F^{-1}\{1 - \frac{2}{jw+1}\} \tag{6}$$

$$= \delta(t) - 2e^{-t}u(t) \tag{7}$$

(c)

$$x(t) * h(t) = y(t) \tag{8}$$

so

$$X(jw)H(jw) = Y(jw) \tag{9}$$

$$\frac{1}{jw+2}\frac{jw-1}{jw+1} = Y(jw) \tag{10}$$

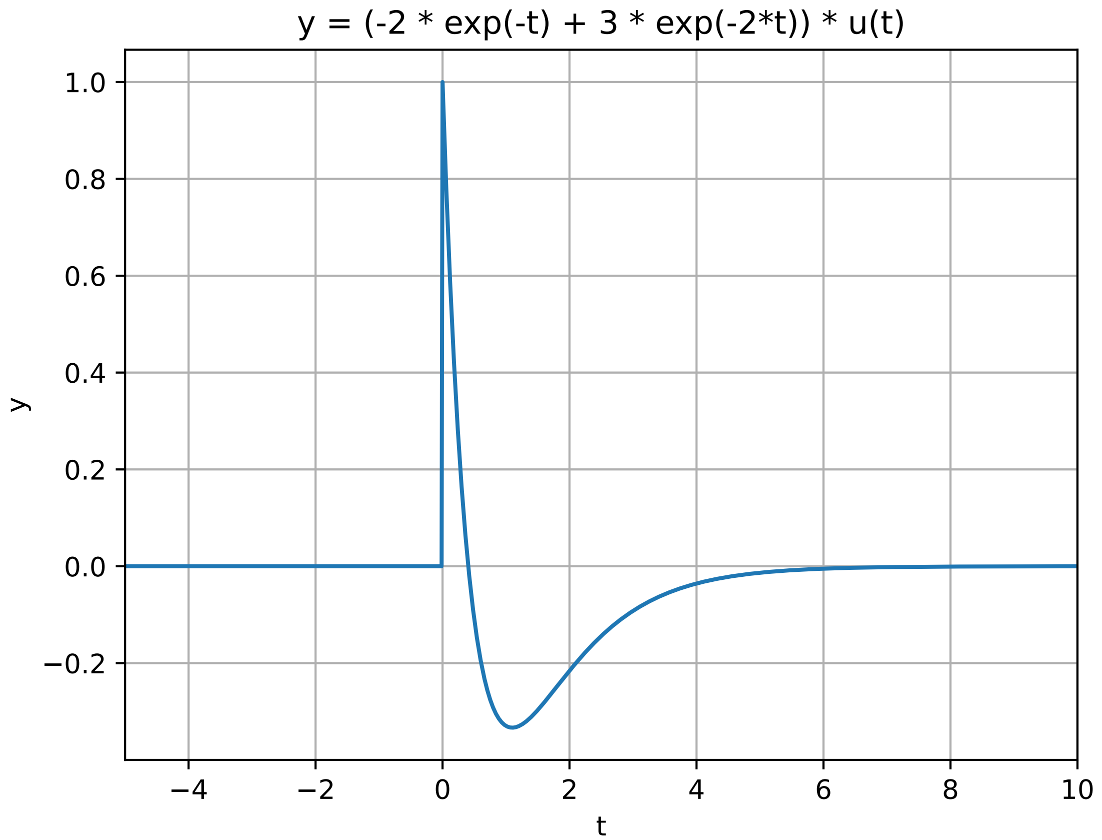
$$\frac{jw-1}{(jw+2)(jw+1)} = \frac{A}{jw+1} + \frac{B}{jw+2} \tag{11}$$

$$2A + B = -1, A + B = 1 \tag{12}$$

$$A = -2, B = 3 \tag{13}$$

$$Y(jw) = \frac{-2}{jw+1} + \frac{3}{jw+2} \tag{14}$$

$$y(t) = -2e^{-t}u(t) + 3e^{-2t}u(t) \tag{15}$$




(d)

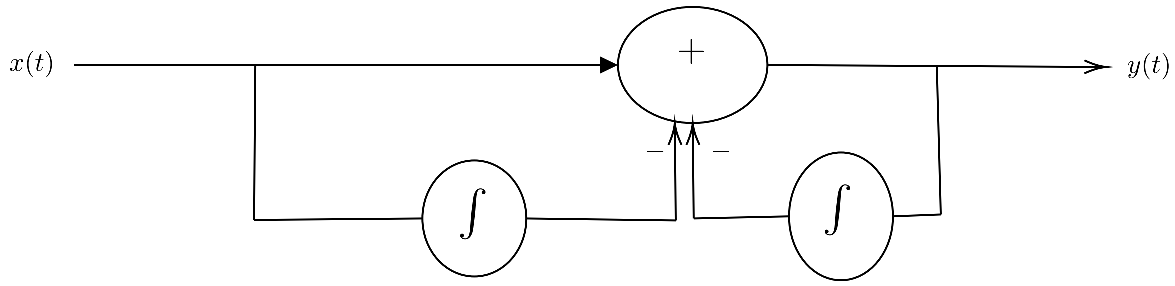
$$\frac{dy}{dt} + y = \frac{dx}{dt} - x \tag{16}$$

$$x + y = \frac{dx}{dt} - \frac{dy}{dt} \tag{17}$$

$$\int (x+y)dt = \int (\frac{dx}{dt} - \frac{dy}{dt})dt \tag{18}$$

$$\int xdt + \int ydt = x - y \tag{19}$$

$$y = x - \int xdt - \int ydt \tag{20}$$




2. (a)

$$H(e^{jw}) = \frac{Y(e^{jw})}{X(e^{jw})} \tag{21}$$

$$e^{jw}Y(e^{jw}) - \frac{1}{2}Y(e^{jw}) = e^{jw}X(e^{jw}) \tag{22}$$

$$H(e^{jw}) = \frac{Y(e^{jw})}{X(e^{jw})} = \frac{e^{jw}}{e^{jw} - \frac{1}{2}} = \frac{1}{1 - \frac{1}{2}e^{-jw}} \tag{23}$$

(b)

$$h[n] = F^{-1}\{H(e^{jw})\} = F^{-1}\{\frac{1}{1 - \frac{1}{2}e^{-jw}}\} = (\frac{1}{2})^n u[n] \tag{24}$$

(c)

$$F^{-1}\{X(e^{jw})\} = \frac{1}{1-\frac{3}{4}e^{-jw}} \tag{25}$$

$$y[n] = x[n] * h[n] \tag{26}$$

$$Y(e^{jw}) = X(e^{jw})H(e^{jw}) \tag{27}$$

$$Y(e^{jw}) = \frac{1}{1-\frac{3}{4}e^{-jw}}\frac{1}{1-\frac{1}{2}e^{-jw}} = \frac{A}{1-\frac{3}{4}e^{-jw}} + \frac{B}{1-\frac{1}{2}e^{-jw}} \tag{28}$$

$$A + B = 1, \frac{1}{2}A + \frac{3}{4}B = 0 \tag{29}$$

$$A = 3, B = -2 \tag{30}$$

$$Y(e^{jw}) = \frac{3}{1-\frac{3}{4}e^{-jw}} - \frac{2}{1-\frac{1}{2}e^{-jw}} \tag{31}$$

$$y[n] = 3(\frac{3}{4})^n u[n] - 2(\frac{1}{2})^n u[n] \tag{32}$$

3. (a)

$$H_3(jw) = H_2(jw)H_1(jw) = \frac{1}{jw^2 + 3jw + 2} \tag{33}$$

$$H_3(jw) = \frac{Y(jw)}{X(jw)} = \frac{1}{jw^2 + 3jw + 2} \tag{34}$$

$$jw^2 Y(jw) + 3jwY(jw) + 2Y(jw) = X(jw) \tag{35}$$

$$y''(t) + 3y'(t) + 2y(t) = x(t) \tag{36}$$

(b)

$$H_3(jw) = H_2(jw)H_1(jw) = \frac{1}{(jw+1)(jw+2)} = \frac{A}{jw+1} + \frac{B}{jw+2} \tag{37}$$

$$A = 1, B = -1 \tag{38}$$

$$h(t) = F^{-1}\{H(jw)\} = e^{-t}u(t) - e^{-2t}u(t) \tag{39}$$

(c)

$$y(t) = x(t) * h(t) \tag{40}$$

$$Y(jw) = X(jw)H_3(jw) \tag{41}$$

$$Y(jw) = \frac{jw}{(jw+1)(jw+2)} = \frac{A}{jw+1} + \frac{B}{jw+2} \tag{42}$$

$$2A + B = 0, A + B = 1 \tag{43}$$

$$A = -1, B = 2 \tag{44}$$

$$Y(jw) = \frac{-1}{jw+1} + \frac{2}{jw+2} \tag{45}$$

$$y(t) = -e^{-t}u(t) + 2e^{-2t}u(t) \tag{46}$$

4. (a)

$$y[n] = x[n] * h_1[n] + x[n] * h_2[n] \tag{47}$$

$$y[n] = x[n] * (h_1[n] + h_2[n]) \tag{48}$$

$$Y(e^{jw}) = X(e^{jw})(H_1(e^{jw}) + H_2(e^{jw})) \tag{49}$$

$$Y(e^{jw}) = X(e^{jw})H(e^{jw}) \tag{50}$$

$$H(e^{jw}) = H_1(e^{jw}) + H_2(e^{jw}) \tag{51}$$

$$H(e^{jw}) = \frac{5e^{-jw} + 12}{e^{-2jw} + 5e^{-jw} + 6} \tag{52}$$

$$H(e^{jw}) = \frac{Y(e^{jw})}{X(e^{jw})} \tag{53}$$

$$Y(e^{jw})(e^{-2jw} + 5e^{-jw} + 6) = (5e^{-jw} + 12)X(e^{jw}) \tag{54}$$

$$y[n-2] + 5y[n-1] + 6y[n] = 5x[n-1] + 12x[n] \tag{55}$$

(b)

$$H(e^{jw}) = H_1(e^{jw}) + H_2(e^{jw}) \tag{56}$$

$$H(e^{jw}) = \frac{5e^{-jw} + 12}{e^{-2jw} + 5e^{-jw} + 6} \tag{57}$$

(c)

$$h[n] = F^{-1}\{H(e^{jw})\} = F^{-1}\{\frac{3}{3 + e^{-jw}} + \frac{2}{2 + e^{-jw}}\} = F^{-1}\{\frac{1}{1 + \frac{1}{3}e^{-jw}} + \frac{1}{1 + \frac{1}{2}e^{-jw}}\} \tag{58}$$

$$h[n] = (-\frac{1}{3})^n u[n] + (-\frac{1}{2})^n u[n] \tag{59}$$

5. The code including the implementation of FFT and IFFT algorithms and are below:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.io.wavfile as wavfile

def fft(x):
    N = int(len(x))

    if N <= 1:
        return x

    even = fft(x[::2])
    odd = fft(x[1::2])


    X = np.zeros(N, dtype=np.complex128)
    for k in range(int(N/2)):
        factor = np.exp(-2j * np.pi * (k-1) / N)
        X[k] = odd[k] + factor * even[k]
        if k == 0:
            X[k + int (N/2)] = odd[k] + np.exp(-2j * np.pi * (int (N/2) -1) / N) * even [k]
        else:
            X[k + int (N/2)] = odd[k] - factor * even [k]

    return X


def ifft(x):
    inverse = fft(np.conj(x))/len(x)
    return np.real(np.conj(inverse))


if __name__ == '__main__':

    # read encoded.wav
    sample_rate, data = wavfile.read('encoded.wav')

    print('sample rate: ', sample_rate)

    duration = len(data) / sample_rate
    time = np.linspace(0, duration, len(data))

    # plot the encoded signal in time domain
    fig1 = plt.figure()
    plt.plot(time,data)
    plt.title('Encoded Signal in Time Domain')
    plt.xlabel('Time(s)')
    plt.ylabel('Amplitude')
    plt.show()
    fig1.savefig('encoded_time.pdf')



    frequency_domain = fft(data)
    frequencies = np.linspace(-len(data)//2, len(data)//2, len(data))

    # plot the magnitude of the frequency domain
    fig2 = plt.figure()
    plt.plot(frequencies,np.abs(frequency_domain))
    plt.title('Encoded Signal in Frequency Domain')
    plt.xlabel('Frequency(Hz)')
    plt.ylabel('Amplitude')
    plt.show()
    #save as pdf
    fig2.savefig('encoded_freq.pdf')

    modified_freq = np.concatenate((np.flip(frequency_domain[:len(frequency_domain)//2]), np.flip(
    frequency_domain[len(frequency_domain)//2:]))) #solves the secret
```

```
# plot the magnitude of the modified frequency domain
fig3 = plt.figure()
plt.plot(frequencies,np.abs(modified_freq))
plt.title('Decoded Signal in Frequency Domain')
plt.xlabel('Frequency()')
plt.ylabel('Amplitude')
plt.show()
fig3.savefig('decoded_freq.pdf')

modified_time = ifft(modified_freq)

# plot the decoded signal in time domain
fig4 = plt.figure()
plt.plot(time, modified_time)
plt.title('Decoded Signal in Time Domain')
plt.xlabel('Time(s)')
plt.ylabel('Amplitude')
plt.show()
fig4.savefig('decoded_time.pdf')

# write the modified time domain to a wav file
wavfile.write('decoded.wav', sample_rate, modified_time.real.astype(data.dtype))
```

Listing 1: Message decoder that uses FFT&IFFT implementaton

**Plots:**

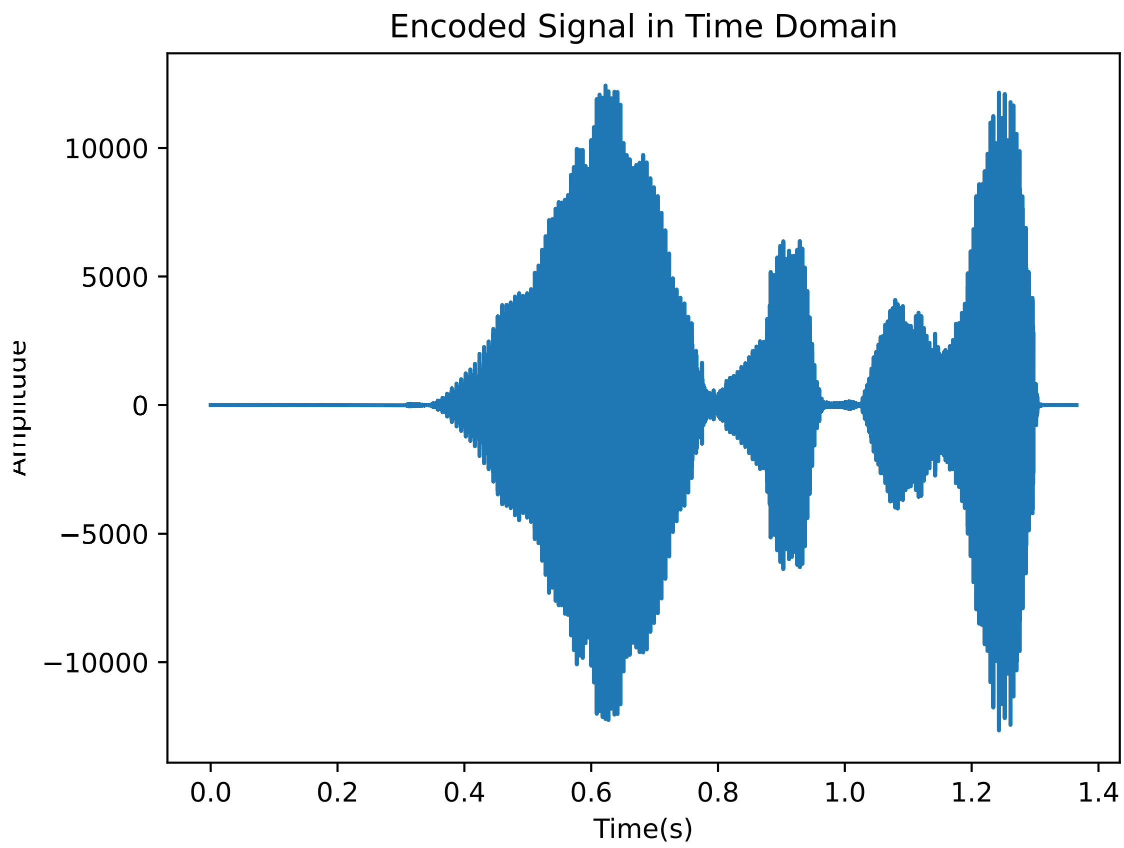



Figure 1: Encoded Signal in Time Domain

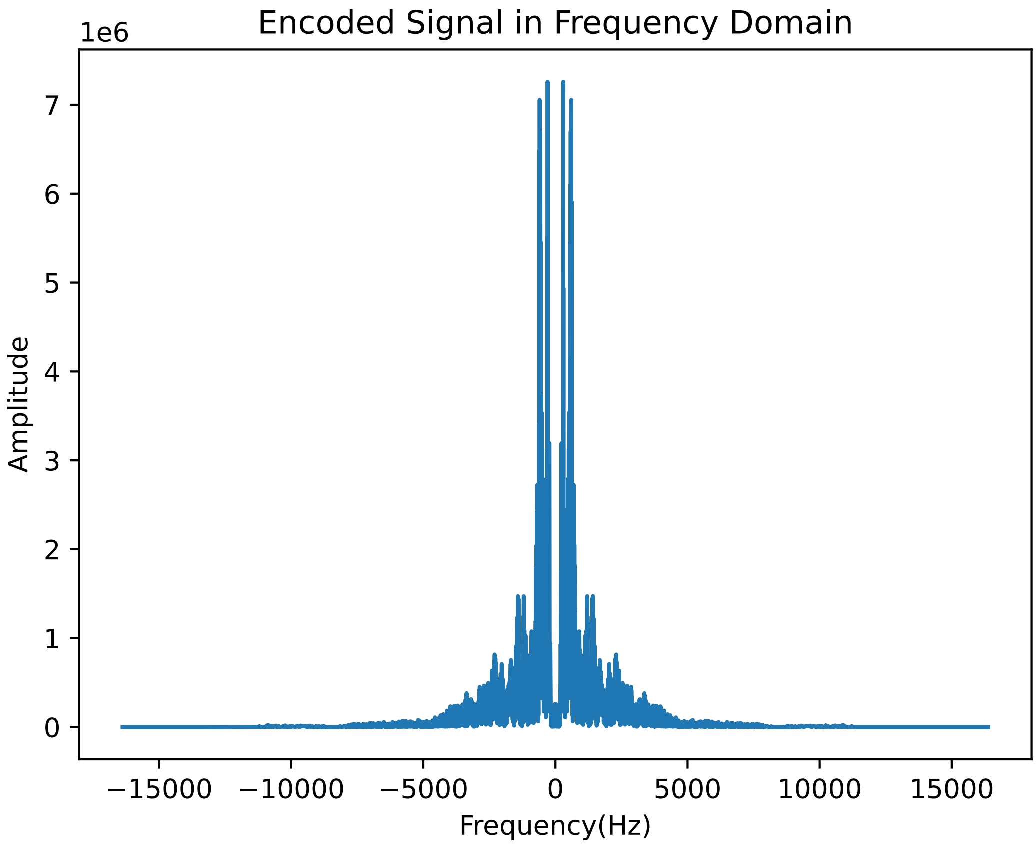



Figure 2: Encoded Signal in Frequency Domain

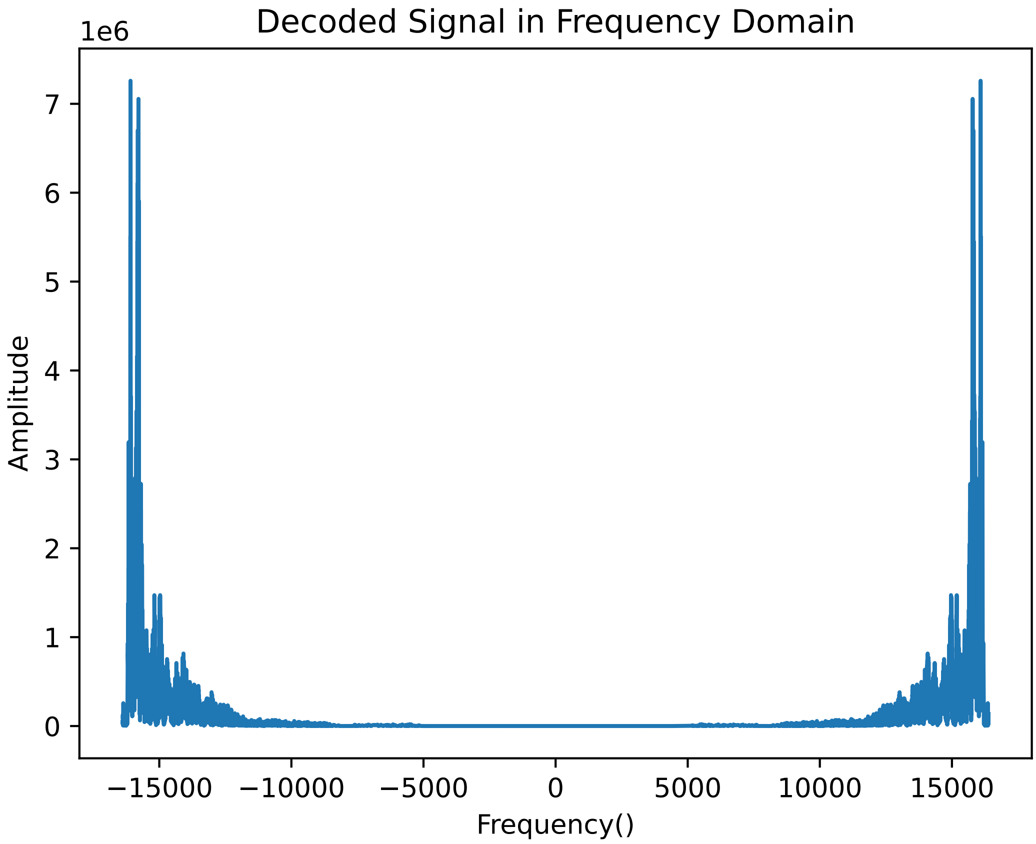



Figure 3: Decoded Signal in Frequency Domain

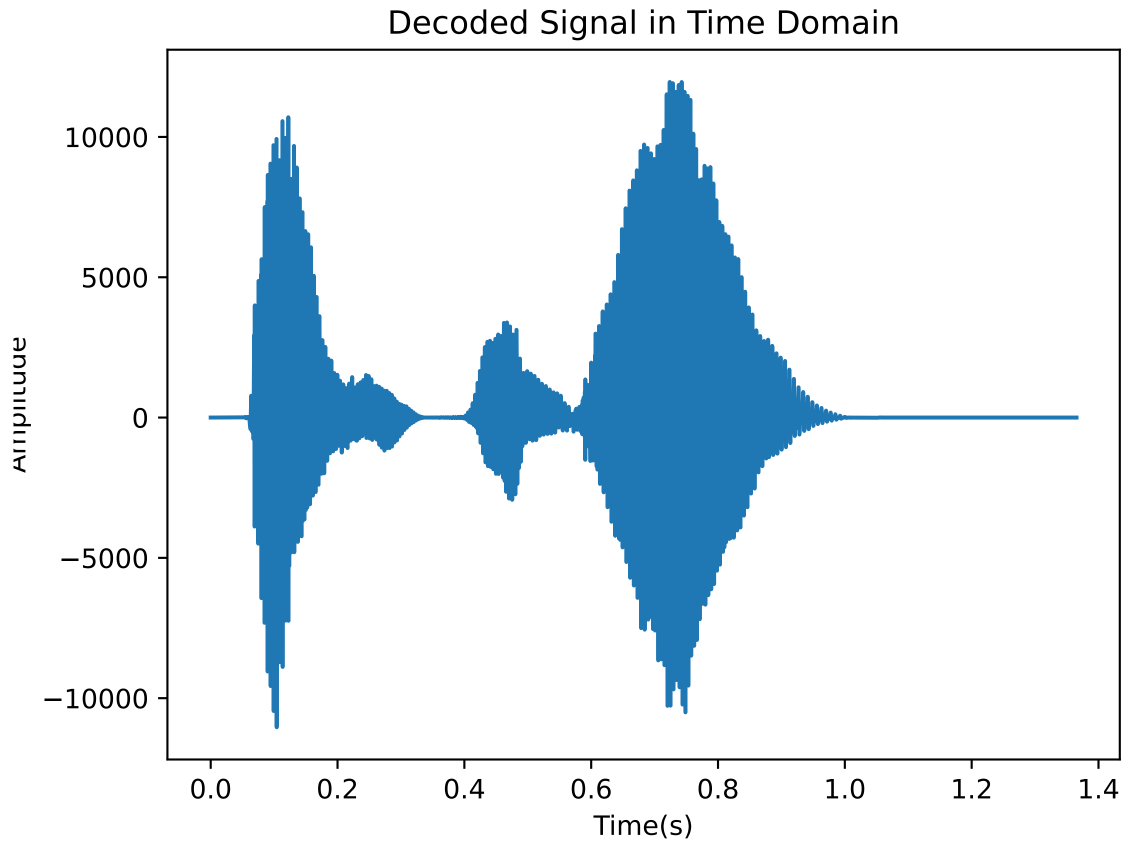



Figure 4: Decoded Signal in Time Domain

The secret message is:

***"I have a dream."***