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

Karacık, Ahmet Furkan
e2310209@ceng.metu.edu.tr

LastName2, FirstName2
exxxxxxx@ceng.metu.edu.tr

June 6, 2023

---

1. (a)

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

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

$$\Rightarrow jwY(jw) + Y(jw) = jwX(jw) - X(jw)$$

$$\Rightarrow \frac{dy(t)}{dt} + y(t) = \frac{dx(t)}{dt} - x(t)$$

(b)

$$h(t) \xleftrightarrow{FT} H(jw)$$

$$\frac{jw-1}{jw+1} = \frac{jw}{1+jw} - \frac{1}{1+jw} \xleftrightarrow{FT} \frac{d(e^{-t}u(t))}{dt} - e^{-t}u(t)$$

$$= 2e^{-t}u(t)$$

where $t > 0$

(c)

$$y = y_h + y_p$$

$$y_h = Ae^{st} \Rightarrow Ase^{st} + Ae^{st} = 0 \Rightarrow Ae^{st}(s+1) = 0 \Rightarrow s = -1$$

$$\Rightarrow y_h(t) = Ae^{-t}$$

$$y_p = Kx(t) = Ke^{-2t}u(t) \Rightarrow -2Ke^{-2t} + Ke^{-2t} = -2e^{-2t} + e^{-2t}$$

$$\Rightarrow -Ke^{-2t} = -e^{-2t} \Rightarrow K = 1$$

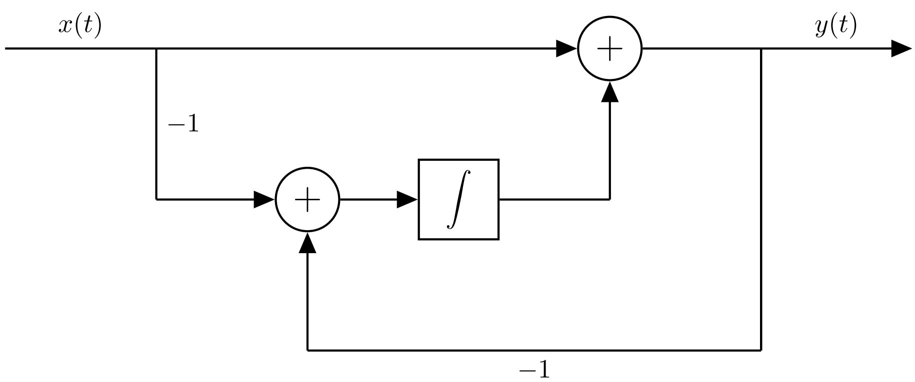
$$\Rightarrow y_p(t) = x(t)$$

$$y(t) = Ae^{-t} + e^{-2t}$$

$$y(0) = 0 \Rightarrow A + 1 = 0 \Rightarrow A = -1$$

$$\Rightarrow y(t) = (e^{-2t} - e^{-t})u(t)$$

(d)



2. (a)

$$y[n] \overset{FT}{\longleftrightarrow} Y(e^{jw}) \Rightarrow y[n+1] = e^{jw}Y(e^{jw})$$

$$x[n] \overset{FT}{\longleftrightarrow} X(e^{jw}) \Rightarrow x[n+1] = e^{jw}X(e^{jw})$$

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

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

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

Divided all terms with $2e^{jw}$.

(b) From the table 5.2, we see that the inverse transform of $H(e^{jw})$ gives us:

$$h[n] = (\frac{1}{2})^n u[n]$$

(c)

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = \sum_{k=-\infty}^{\infty} (\frac{3}{4})^k u[k](\frac{1}{2})^{n-k}u[n-k]$$

$$= \sum_{k=0}^{n} (\frac{3}{2})^k (\frac{1}{2})^k (\frac{1}{2})^n 2^k$$

$$\sum_{k=0}^{n} (\frac{3}{2})^k (\frac{1}{2})^n = (\frac{1}{2})^n \sum_{k=0}^{n} (\frac{3}{4})^k$$

$$= (\frac{1}{2})^n \frac{1-(\frac{3}{4})^{n+1}}{1-\frac{3}{4}}$$

$$= (\frac{1}{2})^n (4 - \frac{3^{n+1}}{4^n})$$

3. (a) Let $G(jw) = X(jw)H_1(jw)$ and $Y(jw) = G(jw)H_2(jw)$ Then,

$$Y(jw) = X(jw)H_1(jw)H_2(jw) = X(jw)\frac{1}{jw+1}\frac{1}{jw+2}$$

$$\Rightarrow Y(jw)(jw+1)(jw+2) = X(jw)$$

$$\Rightarrow (jw)^2Y(jw) + 3jwY(jw) + 2Y(jw) = X(jw)$$

$$\Rightarrow y''(t) + 3y'(t) + 2y(t) = x(t)$$

(b) From $Y(jw) = X(jw)H_1(jw)H_2(jw)$ we see that if $Y(jw) = H(jw)X(jw)$ Then,

$$H(jw) = H_1(jw)H_2(jw)$$

Let $x(t) \overset{FT}{\longleftrightarrow} H_1(jw) = \frac{1}{jw+1}$ and $y(t) \overset{FT}{\longleftrightarrow} H_2(jw) = \frac{1}{jw+2}$ Then,

$$x(t) = e^{-t}u(t)$$

$$y(t) = e^{-2t}u(t)$$

We also see that

$$(x*y)(t) \overset{FT}{\longleftrightarrow} H_1(jw)H_2(jw) = H(jw)$$

It is clear that if we find $H(jw)$ we can find $h(t)$ using Inverse Fourier Transform. To obtain $H(jw)$ we can use the convolution of input and output signals.

$$(x*y)(t) = \int_{-\infty}^{\infty} x(\tau)y(t-\tau)d\tau$$

$$= \int_{-\infty}^{\infty} e^{-\tau}u(\tau)e^{-2(t-\tau)}u(t-\tau)d\tau$$

$$= e^{-2t}\int_{0}^{t} e^{-\tau}e^{2\tau}d\tau = e^{-2t}(e^{\tau})\Big|_0^t$$

$$\Rightarrow h(t) = (e^{-t} - e^{-2t})u(t)$$

(c)

$$Y(jw) = X(jw)H(jw) = jwH(jw)$$
$$y(t) \overset{FT}{\longleftrightarrow} Y(jw)$$
$$\frac{dh(t)}{dt} \overset{FT}{\longleftrightarrow} jwH(jw)$$
$$\Rightarrow y(t) = \frac{dh(t)}{dt} = (-e^{-t} + 2e^{-2t})u(t)$$

4. (a)

$$Y(e^{jw}) = X(e^{jw})(H_1(e^{jw}) + H_2(e^{jw}))$$
$$= X(e^{jw})(\frac{3}{3+e^{-jw}} + \frac{2}{2+e^{-jw}})$$
$$= X(e^{jw})(\frac{12+5e^{-jw}}{6+5e^{-jw}+e^{-2jw}})$$
$$\Rightarrow e^{-2jw}Y(e^{jw}) + 5e^{-jw}Y(e^{jw}) + 6Y(e^{jw}) = 5e^{-jw}X(e^{jw}) + 12X(e^{jw})$$
$$\Rightarrow y[n-2] + 5y[n-1] + 6y[n] = 5x[n-1] + 12x[n]$$

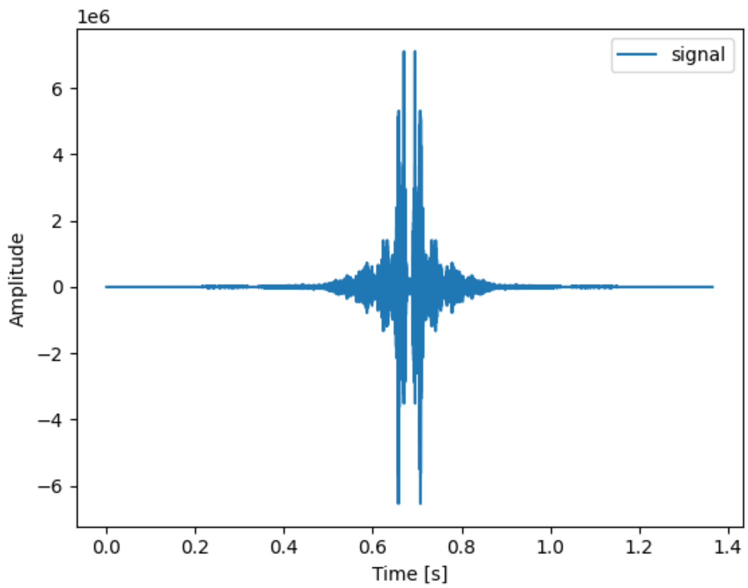
(b)

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

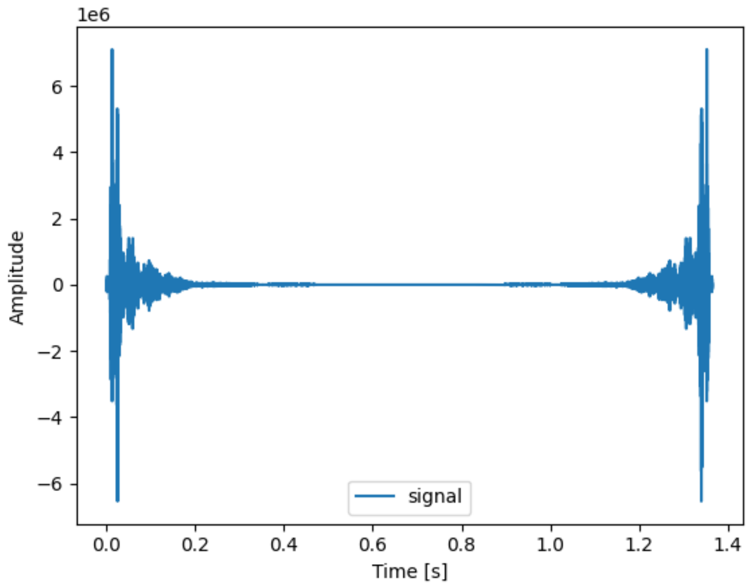
(c)

$$h[n] \overset{FT}{\longleftrightarrow} H(e^{jw}) = \frac{3}{3+e^{-jw}} + \frac{2}{2+e^{-jw}} = \frac{1}{1+\frac{1}{3}e^{-jw}} + \frac{1}{1+\frac{1}{2}e^{-jw}}$$
$$\Rightarrow h[n] = (-\frac{1}{3})^n u[n] + (-\frac{1}{2})^n u[n]$$

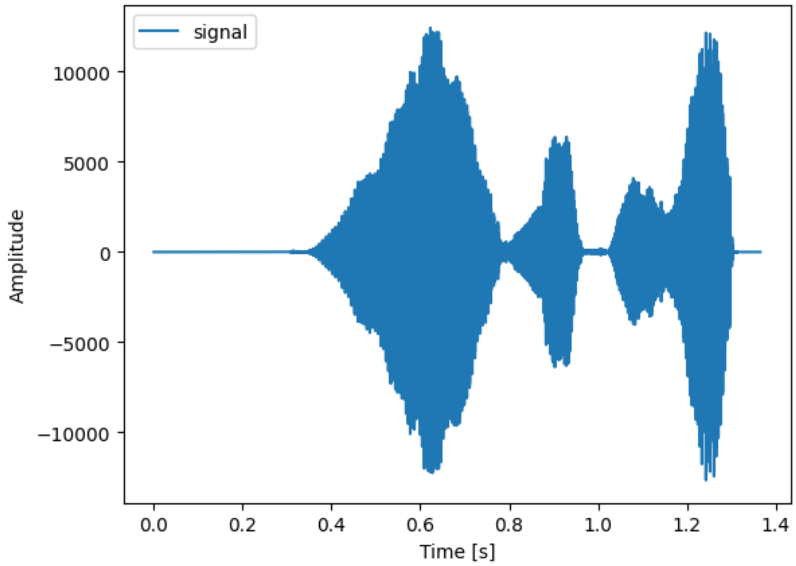
5. Frequency domain magnitude plot of encoded message:



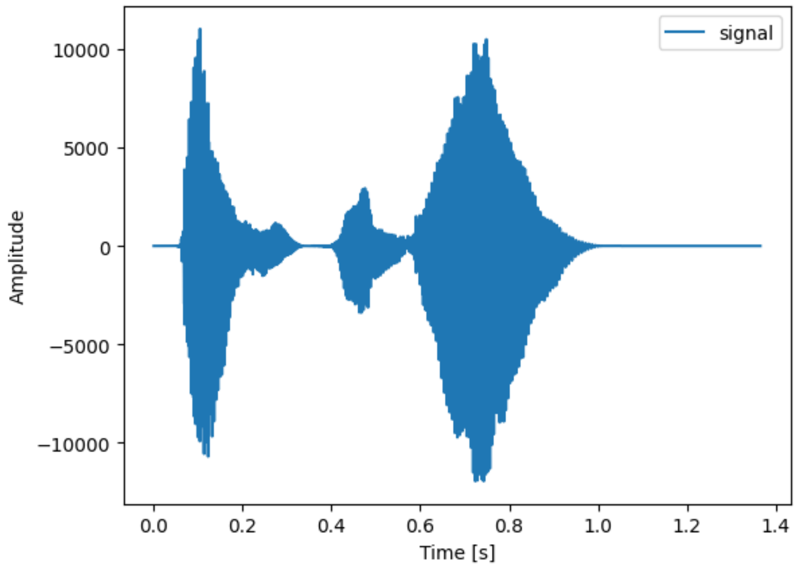
Frequency domain magnitude plot of decoded message:



Time domain magnitude plot of encoded message:



Time domain magnitude plot of decoded message:



Code:

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

# fast fourier transformation function
def fft(x):
    x = np.asarray(x, dtype=float)
    N = x.shape[0]
    n = np.arange(N)
    k = n.reshape((N,1))
    M = np.exp(-2j * np.pi * k * n / N)
    return np.dot(M, x)

# inverse fast fourier transformation function
def ifft(X):
    X = np.asarray(X, dtype=complex)
    N = X.shape[0]
    n = np.arange(N)
    k = n.reshape((N,1))
    M = np.exp(2j * np.pi * k * n / N)
    return np.dot(M, X) / N

# get data
path = r"/content/encoded.wav"
samplerate, data = wavfile.read(path)
length = data.shape[0] / samplerate

# obtain fourier domain representation of the signal
fourier_domain_signal = fft(data)

# apply decoding recipe step 2
X_prime_1 = fourier_domain_signal[:len(fourier_domain_signal)//2]
X_prime_2 = fourier_domain_signal[len(fourier_domain_signal)//2:]

X_1 = []
for i in range(1, len(X_prime_1)+1):
    X_1.append(X_prime_1[-i])
X_2 = []
for i in range(1, len(X_prime_2)+1):
    X_2.append(X_prime_2[-i])

X_prime = np.concatenate((X_1, X_2), axis=None)

# return to time domain
time_domain_signal = ifft(X_prime)

# convert signal to wav file
wavfile.write("decoded_message.wav", samplerate, time_domain_signal.astype(np.int16))

# plot function
def plot(data_to_plot):
    time = np.linspace(0., length, data_to_plot.shape[0])

    plt.plot(time, data_to_plot[:], label="signal")

    plt.legend()

    plt.xlabel("Time [s]")

    plt.ylabel("Amplitude")

    plt.show()
```