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

Mehrabi, Sogol
e2547321@ceng.metu.edu.tr

Qaryouti, Hashem
e265281@ceng.metu.edu.tr

May 28, 2023

---

1. (a)

$$\frac{X(j\omega)}{Y(j\omega)} = \frac{j\omega - 1}{j\omega + 1}$$
$$j\omega Y(j\omega) + Y(j\omega) = j\omega X(j\omega) + X(j\omega)$$

Using Inverse FT,

$$y'(t) + y(t) = x'(t) - x(t)$$

(b) $h(t)$ is the Inverse Fourier of $H(j\omega)$.

$$H(j\omega) = \frac{j\omega - 1}{j\omega + 1} = 1 - \frac{2}{j\omega + 1}$$
$$h(t) = \delta(t) - 2e^{-t}u(t)$$

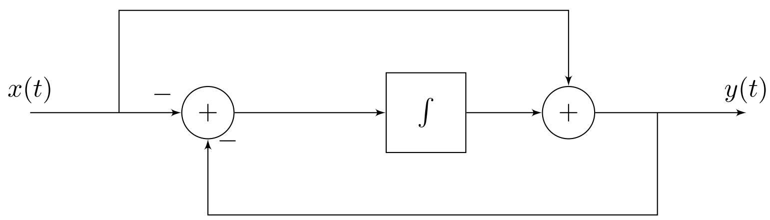
(c)

$$y'(t) + y(t) = -2e^{-2t}u(t) - e^{-2t}u(t)$$
$$y'(t) + y(t) = -3e^{-2t}u(t)$$
$$y'_H(t) + y_H(t) = 0, r + 1 = 0, r = -1$$
$$y_H(t) = ce^{-t}u(t)$$
$$y_P(t) = Ae^{-2t}u(t), y'_p(t) = -2Ae^{-2t}u(t)$$
$$A - 2A = -3, A = 3$$
$$y_P(t) = 3e^{-2t}u(t)$$
$$y(t) = ce^{-t}u(t) + 3e^{-2t}u(t)$$

$y(0) = 0$ since $y(t)$ is a function of $u(t)$, so,

$$c + 3 = 0, c = -3$$
$$y(t) = -3e^{-t}u(t) + 3e^{-2t}u(t)$$

(d)



2. (a) Rewriting the equation,

$$y[n] - \frac{1}{2}y[n-1] = x[n]$$

Taking Fourier transform of both sides,

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

(b) We can obtain $h[n]$ by taking the Inverse Fourier transform of $H(e^{j\omega})$. Hence,

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

(c) First let's obtain $Y(e^{j\omega})$ by multiplying $H(e^{j\omega})$ with the Fourier transform of $x[n]$.

$$X(e^{j\omega}) = \frac{1}{1 - \frac{3}{4}e^{-j\omega}}$$

$$Y(e^{j\omega}) = \frac{1}{1 - \frac{1}{2}e^{-j\omega}} \cdot \frac{1}{1 - \frac{3}{4}e^{-j\omega}} = \frac{A}{1 - \frac{1}{2}e^{-j\omega}} + \frac{B}{1 - \frac{3}{4}e^{-j\omega}}$$

$$A - \frac{3}{4}Ae^{-j\omega} + B - \frac{1}{2}Be^{-j\omega} = 1$$

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

$$A = -2, B = 3$$

$$Y(e^{j\omega}) = \frac{-2}{1 - \frac{1}{2}e^{-j\omega}} + \frac{3}{1 - \frac{3}{4}e^{-j\omega}}$$

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

3. (a)

$$Y(j\omega) = X(j\omega)H_1(j\omega)H_2(j\omega)$$

$$= X(j\omega)\frac{1}{(j\omega)^2 + 3j\omega + 2}$$

$$Y(j\omega)[(j\omega)^2 + 3j\omega + 2] = X(j\omega$$

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

(b)

$$H(j\omega) = H_1(j\omega)H_2(j\omega)$$

$$= \frac{1}{(j\omega)^2 + 3j\omega + 2}$$

$$= \frac{A}{j\omega + 1} + \frac{B}{j\omega + 2}$$

$$Aj\omega + 2A + Bj\omega + B = 1$$

$$A + B = 0, 2A + B = 1$$

$$A = 1, B = -1$$

$$H(j\omega) = \frac{1}{j\omega + 1} - \frac{1}{j\omega + 2}$$

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

(c)

$$Y(j\omega) = \frac{j\omega}{(j\omega)^2 + 3j\omega + 2}$$

$$= \frac{C}{j\omega + 1} + \frac{D}{j\omega + 2}$$

$$Cj\omega + 2C + Dj\omega + D = j\omega$$

$$C + D = 1, 2C + D = 0$$

$$C = -1, D = 2$$

$$Y(j\omega) = \frac{-1}{j\omega + 1} + \frac{2}{j\omega + 2}$$

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

4. (a)

$$Y(e^{j\omega}) = X(e^{j\omega})[H_1(e^{j\omega}) + H_2(e^{j\omega})]$$

$$H_1(e^{j\omega}) + H_2(e^{j\omega}) = \frac{3}{3+e^{-j\omega}} + \frac{2}{2+e^{-j\omega}}$$

$$= \frac{12+5e^{-j\omega}}{6+5e^{-j\omega}+e^{-2j\omega}}$$

$$Y(e^{j\omega})(6+5e^{-j\omega}+e^{-2j\omega}) = X(e^{j\omega})(12+5e^{-j\omega})$$

$$6y[n] + 5y[n-1] + y[n-2] = 12x[n] + 5x[n-1]$$

(b)

$$H(e^{j\omega}) = \frac{Y(e^{j\omega})}{= X(e^{j\omega})} = H_1(e^{j\omega}) + H_2(e^{j\omega})$$

$$H(e^{j\omega}) = \frac{3}{3+e^{-j\omega}} + \frac{2}{2+e^{-j\omega}}$$

(c) We can rewrite the system's frequency response as following:

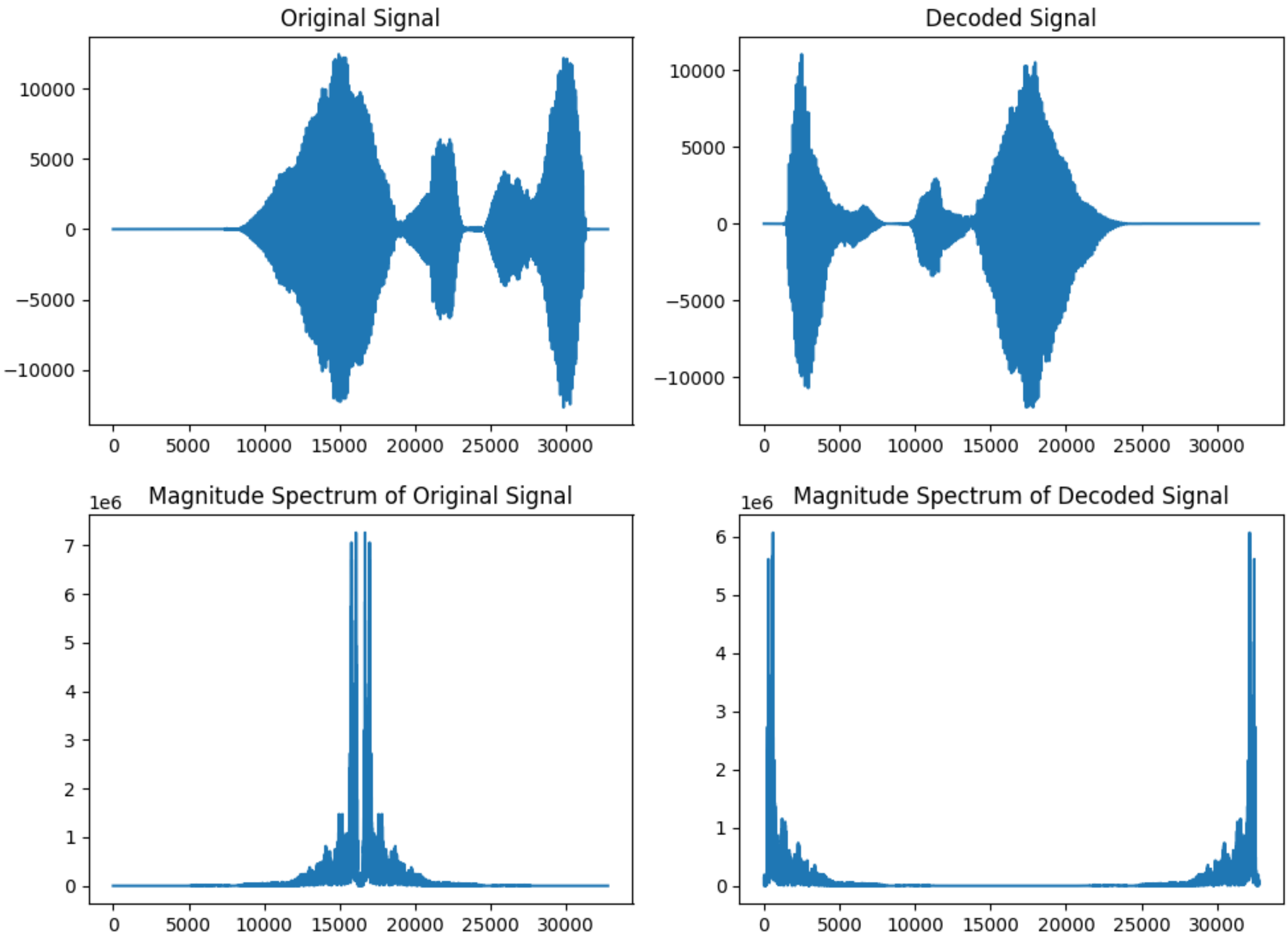
$$H(e^{j\omega}) = \frac{1}{1-(\frac{-1}{3})e^{-j\omega}} + \frac{1}{1-(\frac{-1}{2})e^{-j\omega}}$$

Taking Inverse Fourier transform,

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

5.

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

def fft(x):
    N = len(x)
    if N <= 1:
        return x
    e = fft(x[0::2])  # even
    o = fft(x[1::2])  # odd
    seq = np.array([i for i in range(N)])
    T = np.exp(-2j * np.pi * seq / N)
    first = e + T[: N // 2] * o
    second = e + T[N // 2 :] * o

    return np.concatenate([first, second])

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

def encode_decode(signal):
    transformed = fft(signal)
    half = len(transformed) // 2
    transformed = np.concatenate([transformed[:half][::-1], transformed[half:][::-1]])
    return np.real(ifft(transformed))

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

original_fft = fft(data)
decoded_data = encode_decode(data)
decoded_fft = fft(decoded_data)

#wavfile.write('decoded.wav', sample_rate, decoded_data.astype(np.int16))

plt.figure(figsize=(10, 8))
plt.subplot(2, 2, 1)
plt.title('Original Signal')
plt.plot(data)
plt.subplot(2, 2, 2)
plt.title('Decoded Signal')
plt.plot(decoded_data)
plt.subplot(2, 2, 3)
plt.title('Magnitude Spectrum of Original Signal')
plt.plot(np.abs(original_fft))
plt.subplot(2, 2, 4)
plt.title('Magnitude Spectrum of Decoded Signal')
plt.plot(np.abs(decoded_fft))
plt.show()
```



The secret message: **I have a dream.**