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

Bilgin, Mert Can
e2655181@ceng.metu.edu.tr

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

June 5, 2023

---

1. (a)

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

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

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

(b)

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

By taking the inverse Fourier Transform, we get:

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

(c)

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

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

$$Y(jw) = X(jw)H(jw)$$

$$\frac{1}{jw+2}.\frac{jw-1}{jw+1} = \frac{A}{jw+2} + \frac{B}{jw+1}$$

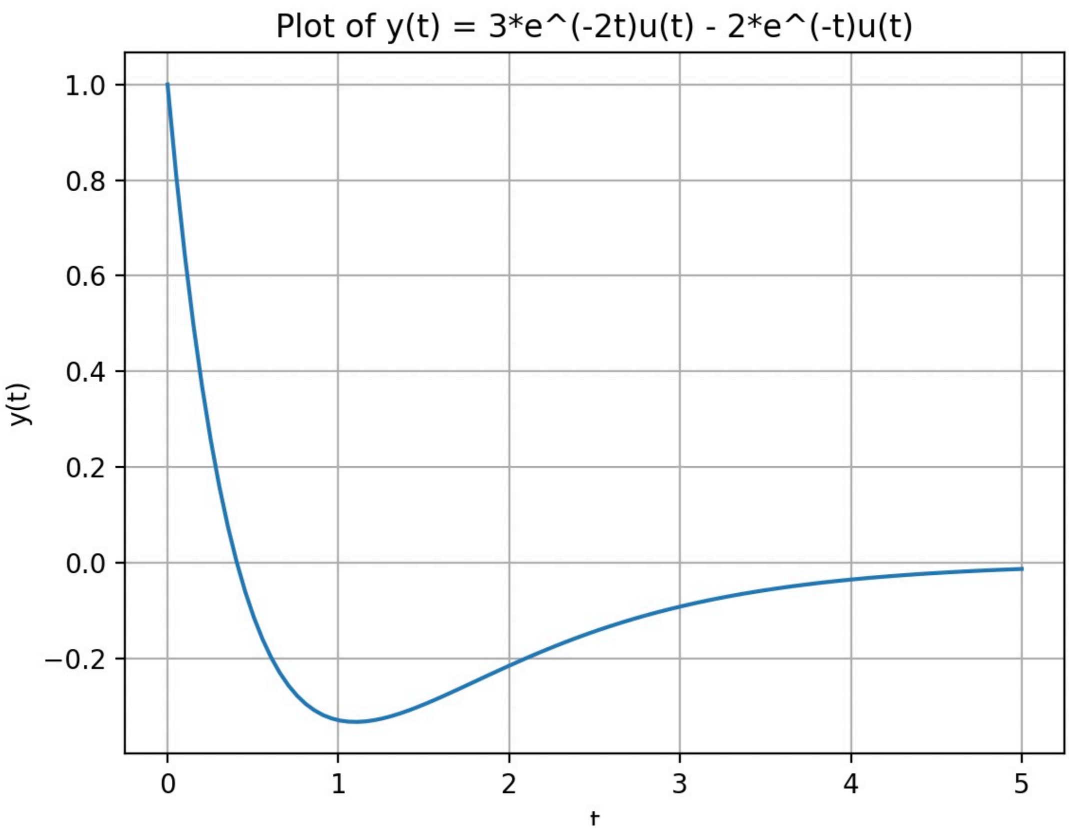
$$A(jw+1) + B(jw+2) = jw - 1$$

If jw = -1, B = -2 ,and if jw = -2, A = 3

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

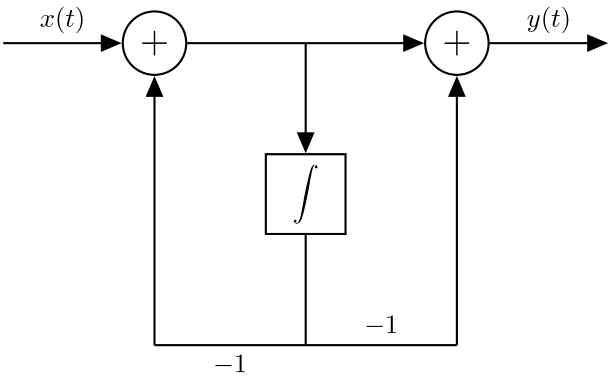
By taking the inverse fourier transform, we get:

$$y(t) = 3e^{-2t}u(t) - 2e^{-t}u(t)$$


Plot of y(t) = 3*e^(-2t)u(t) - 2*e^(-t)u(t)
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
y(t)
0
1
2
3
4
5
t

(d)

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



2. (a)

$$y[n+1] - \frac{1}{2}y[n] = x[n+1]$$
$$e^{jw}Y(e^{jw}) - \frac{1}{2}Y(e^{jw}) = e^{jw}X(e^{jw})$$
$$Y(e^{jw})(e^{jw} - \frac{1}{2}) = e^{jw}X(e^{jw})$$
$$\frac{Y(e^{jw})}{X(e^{jw})} = H(e^{jw}) = \frac{e^{jw}}{e^{jw} - \frac{1}{2}}$$

(b)

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

Multiply the second term by -2/-2:

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

After taking the inverse fourier transform, we get:

$$\delta[n] - 2^n u[-n]$$

(c)

$$x[n] = (\frac{3}{4})^n u[n]$$
$$X(e^{jw}) = \frac{1}{1 - \frac{3}{4}e^{-jw}}$$
$$Y(e^{jw}) = X(e^{jw})H(e^{jw})$$
$$Y(e^{jw}) = (\frac{1}{1 - \frac{3}{4}e^{-jw}})(1 - \frac{1}{1 - 2e^{jw}})$$
$$Y(e^{jw}) = \frac{1}{1 - \frac{3}{4}e^{-jw}} - \frac{1}{1 - 2e^{jw} - \frac{3}{4}e^{-jw} + \frac{3}{2}}$$
$$Y(e^{jw}) = \frac{1}{1 - \frac{3}{4}e^{-jw}} - \frac{1}{\frac{5}{2} - 2e^{jw} - \frac{3}{4}e^{-jw}}$$
$$y[n] = (\frac{3}{4})^n u[n] - F^{-1}(\frac{1}{\frac{5}{2} - 2e^{jw} - \frac{3}{4}e^{-jw}})$$

3. (a)

$$H(jw) = H_1(jw)H_2(jw) = (\frac{1}{jw+1})(\frac{1}{jw+2}) = \frac{1}{(jw)^2 + 3jw + 2}$$
$$H(jw) = \frac{Y(jw)}{X(jw)} = \frac{1}{(jw)^2 + 3jw + 2}$$
$$(jw)^2Y(jw) + 3jwY(jw) + 2Y(jw) = X(jw)$$
$$\frac{d^2y(t)}{dt} + 3\frac{dy(t)}{dt} + 2y(t) = x(t)$$

(b)

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

$$H(jw) = (\frac{1}{jw+1})(\frac{1}{jw+2}) = \frac{A}{jw+1} + \frac{B}{jw+2}$$

$$=> A(jw+2) + B(jw+1) = 1$$

If jw = -2, B = -1, and if jw = -1, A = 1. Hence,

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

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

(c)

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

It can be written as:

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

After taking the inverse fourier transform, we get:

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

4. (a)

$$H(e^{jw}) = H_1(e^{jw}) + H_2(e^{jw}) = \frac{3}{3+e^{-jw}} + \frac{2}{2+e^{-jw}} = \frac{12+5e^{-jw}}{6+5e^{-jw}+e^{-2jw}}$$

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

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

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

(b)

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

If we multiply the first term by (1/3)/(1/3), and multiply the second term by (1/2)/(1/2), we get:

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

(c) After taking the inverse fourier transform of the frequency response, we get:

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

5. ```
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
    even = fft(x[0::2])
    odd = fft(x[1::2])
    T = [np.exp(-2j * np.pi * k / N) * odd[k] for k in range(N // 2)]
    return [even[k] + T[k] for k in range(N//2)] + [even[k] - T[k] for k in range(N//2)]

def ifft(X):
    N = len(X)
    if N <= 1:
        return X
    even = ifft(X[0::2])
    odd = ifft(X[1::2])
    T = [np.exp(2j * np.pi * k / N) * odd[k] for k in range(N // 2)]
    return np.array([(even[k]+T[k])/2 for k in range(N//2)]+[(even[k]-T[k])/2 for k in
    range(N//2)])

Read the encoded voice message
sample_rate, encoded_audio = wavfile.read('encoded.wav')
# Convert the audio to a mono channel if needed
if len(encoded_audio.shape) > 1:
    encoded_audio = encoded_audio[:, 0]

# Apply the decoding process using custom FFT and IFFT
frequency_domain = fft(encoded_audio)
midpoint = len(frequency_domain) // 2
positive_frequencies = frequency_domain[midpoint:]
negative_frequencies = frequency_domain[:midpoint]
reversed_domain = np.concatenate((negative_frequencies[::-1], positive_frequencies[::-1]))
decoded_audio = ifft(reversed_domain).real

# Plot the frequency domain magnitude of the encoded and decoded signals
frequency_bins = np.arange(len(encoded_audio))
frequency_bins = frequency_bins * sample_rate / len(encoded_audio)
plt.figure(figsize=(12, 4))
plt.subplot(1, 2, 1)
plt.plot(frequency_bins, np.abs(frequency_domain))
plt.title('Frequency Domain Magnitude (Encoded)')
plt.xlabel('Frequency')
plt.ylabel('Magnitude')
plt.subplot(1, 2, 2)
plt.plot(frequency_bins, np.abs(reversed_domain))
plt.title('Frequency Domain Magnitude (Decoded)')
plt.xlabel('Frequency')
plt.ylabel('Magnitude')
plt.tight_layout()
plt.show()

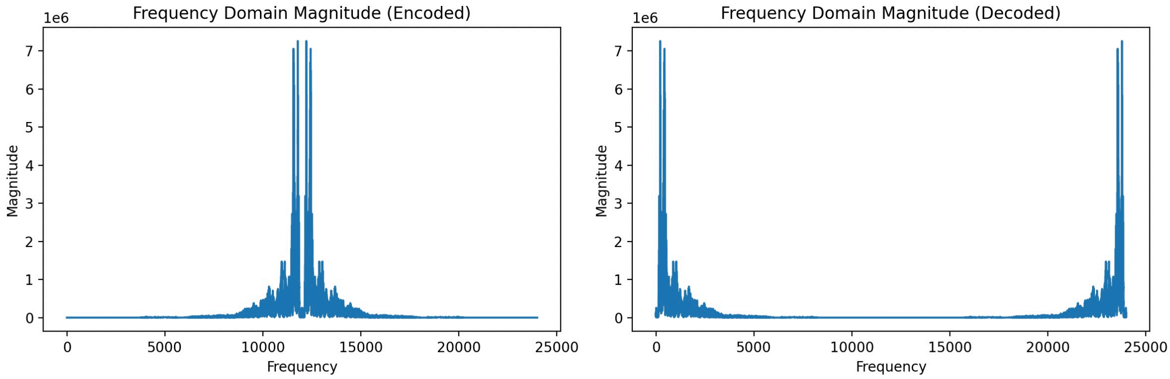
# Plot the time domain signals of the encoded and decoded audio
time = np.arange(len(encoded_audio)) / sample_rate
plt.figure(figsize=(12, 4))
plt.subplot(1, 2, 1)
plt.plot(time, encoded_audio)
plt.title('Encoded Audio (Time Domain)')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.subplot(1, 2, 2)
plt.plot(time, decoded_audio)
plt.title('Decoded Audio (Time Domain)')
```
```

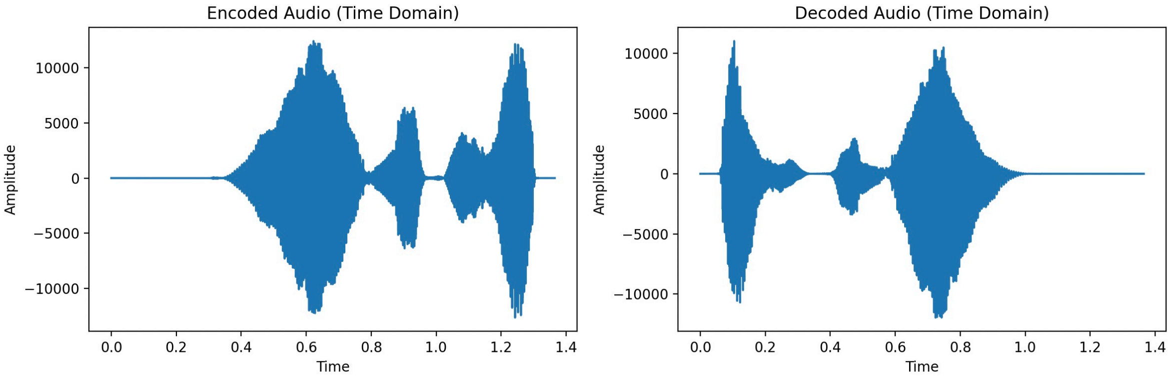
```
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.tight_layout()
plt.show()

# Listen to the decoded audio message
# Write the decoded audio to a file
wavfile.write('decoded.wav', sample_rate, np.int16(decoded_audio))

print("Decoding complete! Decoded audio saved as 'decoded.wav'.")
```

The message is: "I have a dream"


Frequency Domain Magnitude (Encoded)
1e6
Magnitude
Frequency
Frequency Domain Magnitude (Decoded)
1e6
Magnitude
Frequency


Encoded Audio (Time Domain)
Amplitude
Time
Decoded Audio (Time Domain)
Amplitude
Time