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

Geçit, Emre
e2521581@ceng.metu.edu.tr

Yancı, Baran
e2449015@ceng.metu.edu.tr

June 8, 2023

---

1. (a)

$$\begin{aligned}
H(j\omega) &= \frac{j\omega - 1}{j\omega + 1} \\
\frac{Y(j\omega)}{X(j\omega)} &= \frac{j\omega - 1}{j\omega + 1} \\
Y(j\omega)(j\omega + 1) &= X(j\omega)(j\omega - 1) \\
y'(t) + y(t) &= x'(t) - x(t)
\end{aligned}$$

(b)

$$\begin{aligned}
H(j\omega) &= \frac{j\omega - 1}{j\omega + 1} \\
h(t) &= \mathcal{F}^{-1}\{H(j\omega)\} \\
&= \mathcal{F}^{-1}\{\frac{j\omega - 1}{j\omega + 1}\} \\
&= \mathcal{F}^{-1}\{\frac{j\omega + 1 - 2}{j\omega + 1}\} \\
&= \mathcal{F}^{-1}\{\frac{j\omega + 1}{j\omega + 1}\} - \mathcal{F}^{-1}\{\frac{2}{j\omega + 1}\} \\
&= \mathcal{F}^{-1}\{1\} - 2\mathcal{F}^{-1}\{\frac{1}{j\omega + 1}\} \\
&= \delta(t) - 2e^{-t}u(t)
\end{aligned}$$

(c)

$$\begin{aligned}
y'(t) + y(t) &= x'(t) - x(t) \\
y'(t) + y(t) &= -2e^{-2t}u(t) - e^{-2t}u(t) \\
y'(t) + y(t) &= -3e^{-2t}u(t) \\
y_p(t) &= Ae^{-2t} \\
y_p'(t) &= -2Ae^{-2t} \\
-2Ae^{-2t} + Ae^{-2t} &= -3e^{-2t}u(t) \\
A &= 3 \\
y_p(t) &= 3e^{-2t} \\
y_h(t) &= c_1e^{-t}u(t) \\
y(t) &= y_p(t) + y_h(t) \\
&= 3e^{-2t} + c_1e^{-t}u(t) \\
y(0) &= 0 \\
0 &= 3e^{-2(0)} + c_1e^{-0}u(0) \\
0 &= 3 + c_1 \\
c_1 &= -3 \\
y(t) &= 3e^{-2t} - 3e^{-t}u(t)
\end{aligned}$$

(d) Block diagram of the system:

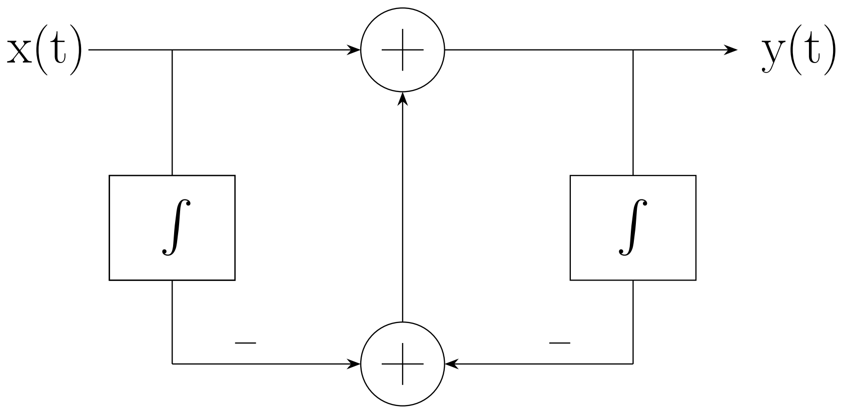

Figure 1: Block Diagram of the System

2. (a)

$$\begin{aligned}
y[n+1] - \frac{1}{2}y[n] &= x[n+1] \\
e^{j\omega}Y(e^{j\omega}) - \frac{1}{2}Y(e^{j\omega}) &= e^{j\omega}X(e^{j\omega}) \\
H(e^{j\omega}) &= \frac{Y(e^{j\omega})}{X(e^{j\omega})} \\
H(e^{j\omega}) &= \frac{e^{j\omega}}{e^{j\omega} - \frac{1}{2}}
\end{aligned}$$

(b)

$$
\begin{aligned}
H(e^{j\omega}) &= \frac{e^{j\omega}}{e^{j\omega}-\frac{1}{2}} \\
h[n] &= \mathcal{F}^{-1}\{H(e^{j\omega})\} \\
&= \mathcal{F}^{-1}\{\frac{e^{j\omega}}{e^{j\omega}-\frac{1}{2}}\} \\
&= \mathcal{F}^{-1}\{\frac{e^{j\omega}-\frac{1}{2}+\frac{1}{2}}{e^{j\omega}-\frac{1}{2}}\} \\
&= \mathcal{F}^{-1}\{\frac{e^{j\omega}-\frac{1}{2}}{e^{j\omega}-\frac{1}{2}}+\frac{\frac{1}{2}}{e^{j\omega}-\frac{1}{2}}\} \\
&= \mathcal{F}^{-1}\{1+\frac{\frac{1}{2}}{e^{j\omega}-\frac{1}{2}}\} \\
&= \mathcal{F}^{-1}\{1\}+\mathcal{F}^{-1}\{\frac{\frac{1}{2}}{e^{j\omega}-\frac{1}{2}}\} \\
&= \delta[n]+\frac{1}{2}\mathcal{F}^{-1}\{\frac{1}{e^{j\omega}-\frac{1}{2}}\} \\
&= \delta[n]+\frac{1}{2}e^{\frac{1}{2}n}u[n]
\end{aligned}
$$

(c) The initial condition is $y[a]=0, a \leq -1$.

$$
\begin{aligned}
y[n+1]-\frac{1}{2}y[n] &= x[n+1] \\
y[n+1] &= \frac{1}{2}y[n]+x[n+1] \\
y[0] &= \frac{1}{2}y[-1]+x[0] \\
&= 0+1 \\
&= 1 \\
y[1] &= \frac{1}{2}y[0]+x[1] \\
&= \frac{1}{2}+\frac{3}{4} \\
&= \frac{5}{4} \\
y[2] &= \frac{1}{2}y[1]+x[2] \\
&= \frac{5}{8}+\frac{9}{16} \\
&= \frac{19}{16} \\
y[3] &= \frac{1}{2}y[2]+x[3] \\
&= \frac{19}{32}+\frac{27}{64} \\
&= \frac{65}{64} \\
y[4] &= \frac{1}{2}y[3]+x[4] \\
&= \frac{65}{128}+\frac{81}{256} \\
&= \frac{211}{256} \\
&\dots \\
y[n] &= 2^{-n}\left(1+3\left(\left(\frac{3}{2}\right)^n-1\right)\right)
\end{aligned}
$$

3. (a)

$$\begin{aligned}
H(j\omega) &= H_1(j\omega)H_2(j\omega) \\
&= \frac{1}{j\omega+1}\frac{1}{j\omega+2} \\
&= \frac{Y(j\omega)}{X(j\omega)} \\
Y(j\omega)(j\omega+1)(j\omega+2) &= X(j\omega) \\
Y(j\omega)(j^2\omega^2+3j\omega+2) &= X(j\omega) \\
y''(t)+3y'(t)+2y(t) &= x(t)
\end{aligned}$$

(b)

$$\begin{aligned}
H(j\omega) &= H_1(j\omega)H_2(j\omega) \\
H_1(j\omega) &= \frac{1}{j\omega+1} \\
H_2(j\omega) &= \frac{1}{j\omega+2} \\
H(j\omega) &= \frac{1}{j\omega+1}\frac{1}{j\omega+2} \\
h_1(t) &= \mathcal{F}^{-1}\{H(j\omega)\} \\
&= \mathcal{F}^{-1}\{\frac{1}{j\omega+1}\frac{1}{j\omega+2}\} \\
&= \mathcal{F}^{-1}\{\frac{1}{j\omega+1}\} * \mathcal{F}^{-1}\{\frac{1}{j\omega+2}\} \\
&= e^{-t}u(t) * e^{-2t}u(t) \\
&= \int_{-\infty}^{\infty} e^{-(t-\tau)}u(t-\tau)e^{-2\tau}u(\tau)d\tau \\
&= \int_0^t e^{-(t-\tau)}e^{-2\tau}d\tau \\
&= \int_0^t e^{-t+\tau}e^{-2\tau}d\tau \\
&= \int_0^t e^{-t}e^{\tau}e^{-2\tau}d\tau \\
&= e^{-t}\int_0^t e^{-\tau}d\tau \\
&= e^{-t}\left[-e^{-\tau}\right]_0^t \\
&= e^{-t}\left[-e^{-t}+1\right] \\
&= e^{-t}-e^{-2t}
\end{aligned}$$

(c)

$$\begin{aligned}
X(j\omega) &= j\omega \\
Y(j\omega) &= H(j\omega)X(j\omega) \\
&= \frac{1}{j\omega+1}\frac{1}{j\omega+2}j\omega \\
y(t) &= \mathcal{F}^{-1}\{Y(j\omega)\} \\
&= \mathcal{F}^{-1}\{\frac{j\omega}{(j\omega+1)(j\omega+2)}\} \\
&= \mathcal{F}^{-1}\{\frac{-1}{j\omega+1}+\frac{2}{j\omega+2}\} \\
&= \mathcal{F}^{-1}\{\frac{-1}{j\omega+1}\} + \mathcal{F}^{-1}\{\frac{2}{j\omega+2}\} \\
&= -e^{-t}u(t)+2e^{-2t}u(t)
\end{aligned}$$

4. (a)

$$H(e^{j\omega}) = \frac{Y(e^{j\omega})}{X(e^{j\omega})} = \frac{5e^{-j\omega}+12}{e^{-2j\omega}+5e^{-j\omega}+6} \qquad \text{Found in part 4b}$$

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

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

(b)

$$\begin{aligned} H(e^{j\omega}) &= H_1(e^{j\omega}) + H_2(e^{j\omega}) \\ &= \frac{3}{3+e^{-j\omega}} + \frac{2}{2+e^{-j\omega}} \\ &= \frac{5e^{-j\omega}+12}{e^{-2j\omega}+5e^{-j\omega}+6} \end{aligned}$$

(c)

$$\begin{aligned} h[n] &= \mathcal{F}^{-1}\{H_1(j\omega)+H_2(j\omega)\} \\ &= \mathcal{F}^{-1}\{\frac{3}{3+e^{-j\omega}} + \frac{2}{2+e^{-j\omega}}\} \\ &= \mathcal{F}^{-1}\{\frac{3}{3+e^{-j\omega}}\} + \mathcal{F}^{-1}\{\frac{2}{2+e^{-j\omega}}\} \\ &= \frac{-1}{3}u[n] + \frac{1}{2}u[n] \end{aligned}$$

5. After decoding **encoded.wav** according to the recipe given, a wav file **decoded.wav** is obtained. When played by a media player, the decoded file says ”I have a dream”.

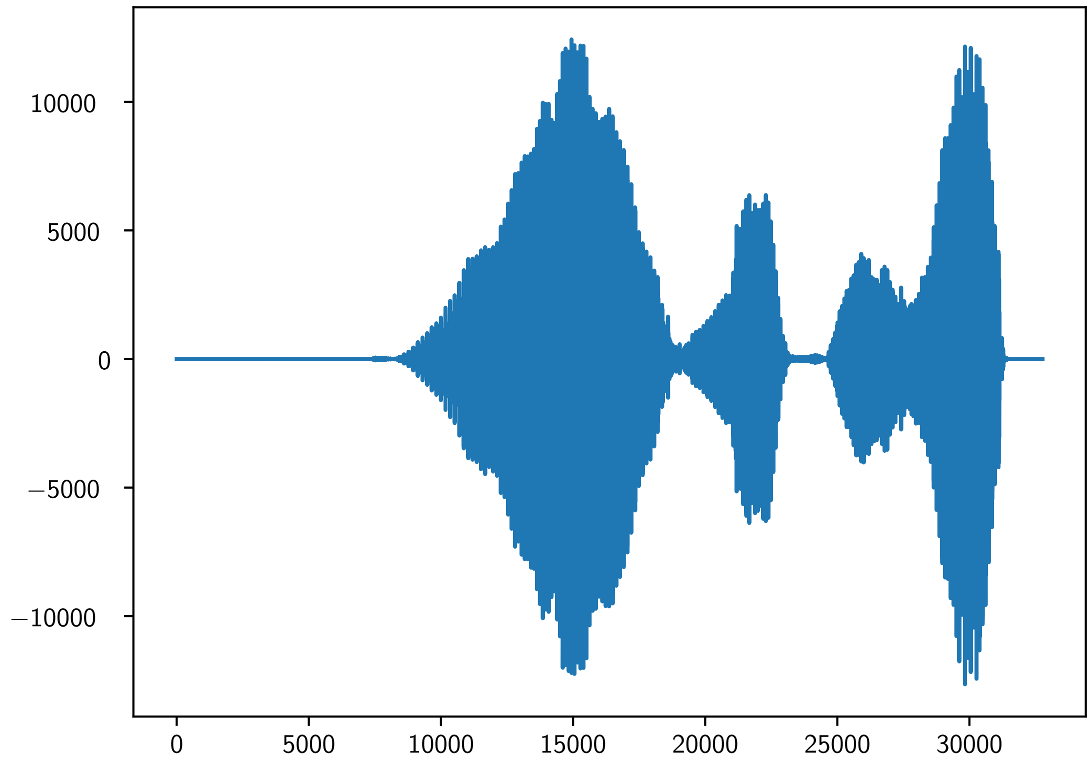

Figure 2: Time Domain Representation of The Encoded Signal

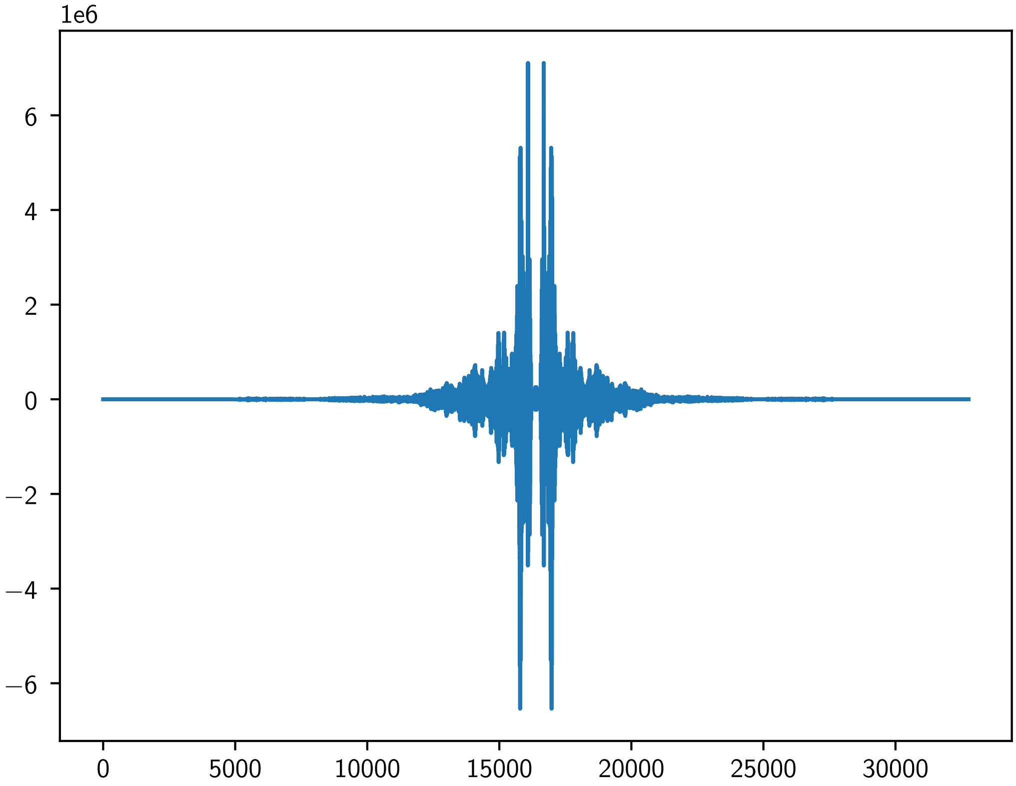

Figure 3: Frequency Domain Representation of The Encoded Signal

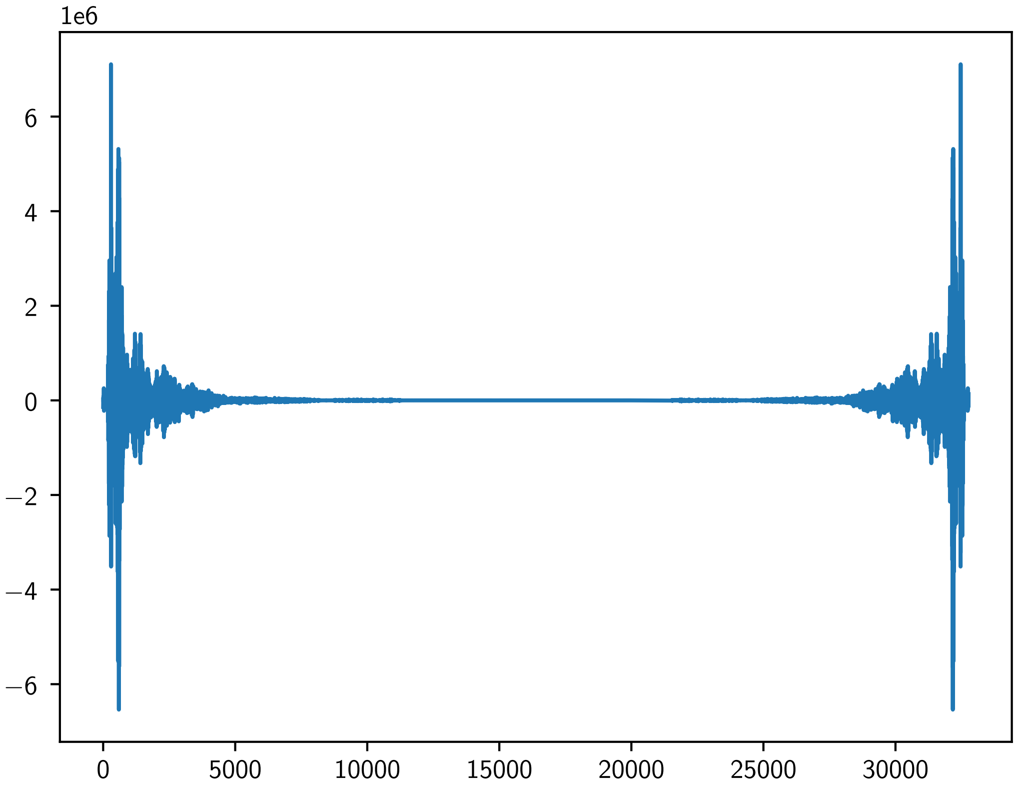

Figure 4: Frequency Domain Representation of The Decoded Signal

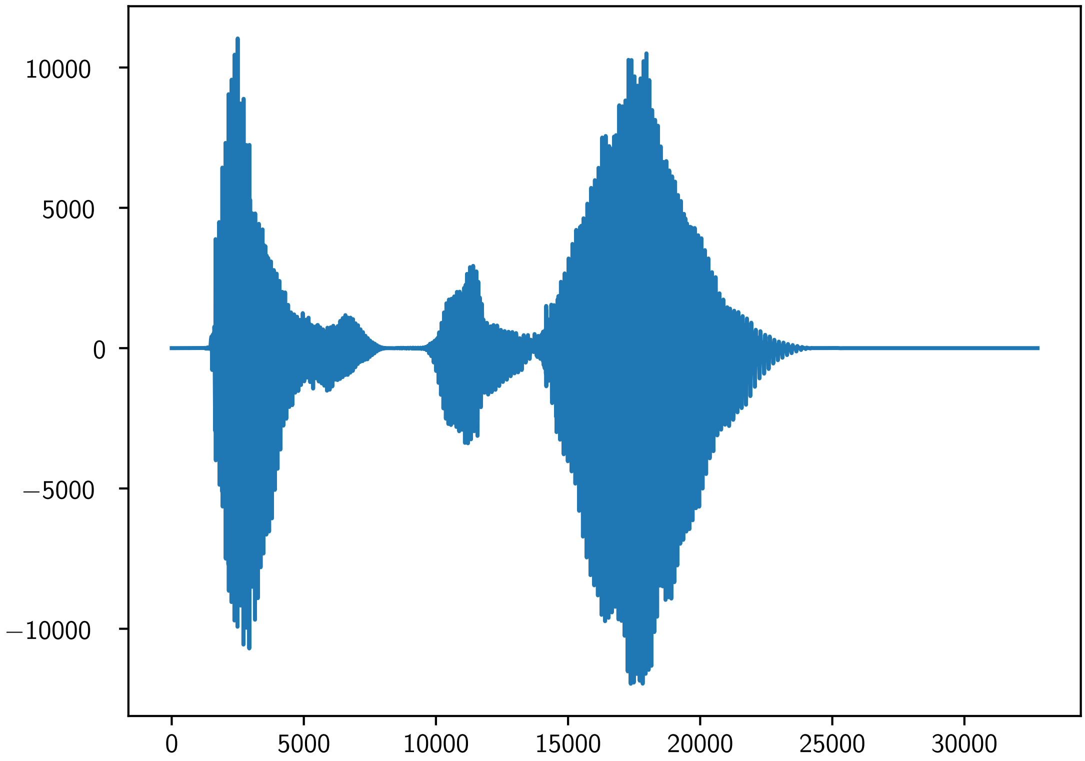

Figure 5: Time Domain Representation of The Decoded Signal

The code for the decoding process is given below.

```
import numpy
from scipy.io import wavfile
from tester import test # This is to verify the implementation, uses numpy.fft
import matplotlib
from matplotlib import pyplot

matplotlib.use("pgf")

def fft(x: numpy.ndarray) -> numpy.ndarray:
    N = len(x)
    if N <= 1:
        return x

    even = fft(x[::2])
    odd = fft(x[1::2]) * numpy.exp(-2j * numpy.pi * numpy.arange(N // 2) / N)
    return numpy.concatenate([even + odd,
                              even - odd])

def ifft(X: numpy.ndarray) -> numpy.ndarray:
    return numpy.flip(fft(X)) / len(X)

def plot(signal, name):
    pyplot.plot(signal)
    pyplot.savefig("figures/" + name + ".pgf")
    pyplot.clf()

rate, encoded = wavfile.read("encoded.wav")
plot(encoded, "encoded")
N = len(encoded)
dft_encoded = fft(encoded)
plot(dft_encoded.real, "dft_encoded")
flipped = numpy.flip(dft_encoded)
```

```
dft_decoded = numpy.concatenate([flipped[N//2:], flipped[:N//2]])
plot(dft_decoded.real, "dft_decoded")
decoded = ifft(dft_decoded)
plot(decoded.real, "decoded")
assert test(encoded, decoded)
wavfile.write("decoded.wav", rate, decoded.real.astype(numpy.int16))
```