Digital Signal Processing

Ameya Chatur
BM20BTECH11002

Contents

***Abstract*—This manual provides a simple introduction to digital signal processing.**

1 Software Installation

Run the following commands

```
sudo apt−get update
sudo apt−get install libffi−dev libsndfile1 python3
    −scipy python3−numpy python3−matplotlib
sudo pip install cffi pysoundfile
```

2 Digital Filter

2.1 Download the sound file from

```
wget https://github.com/yashrajput22/EE3900
    −22/blob/master/codes/Section−2/
    Sound_Noise.wav
```

2.2 You will find a spectrogram at https://academo.org/demos/spectrum-analyzer. Upload the sound file that you downloaded in Problem **??** in the spectrogram and play. Observe the spectrogram. What do you find?
Solution: There are a lot of yellow lines between 440 Hz to 5.1 KHz. These represent the synthesizer key tones. Also, the key strokes are audible along with background noise.

2.3 Write the python code for removal of out of band noise and execute the code.
Solution:

```
import soundfile as sf
from scipy import signal

#read .wav file
input_signal,fs = sf.read("Sound_Noise.wav
    ")
#sampling frequency of Input signal
sampl_freq=fs
# print("Sample Frequency ",sampl_freq)
#order of the filter
order=4
#cutoff frquency 4kHz
cutoff_freq=4000.0
#digital frequency
Wn=2*cutoff_freq/sampl_freq
# b and a are numerator and
    denominatorpolynomials respectively
b, a = signal.butter(order,Wn, 'low')
#filter the input signal with butterworth filter
# output_signal = signal.filtfilt(b, a,
    input_signal)
output_signal = signal.lfilter(b, a,
    input_signal)
#write the output signal into .wav file
sf.write('Sound_With_ReducedNoise.wav',
    output_signal, fs)
```

2.4 The output of the python script in Problem **??** is the audio file Sound_With_ReducedNoise.wav. Play the file in the spectrogram in Problem **??**. What do you observe?
Solution: The key strokes as well as background noise is subdued in the audio. Also, the signal is blank for frequencies above 5.1 kHz.

3 Difference Equation

3.1 Let

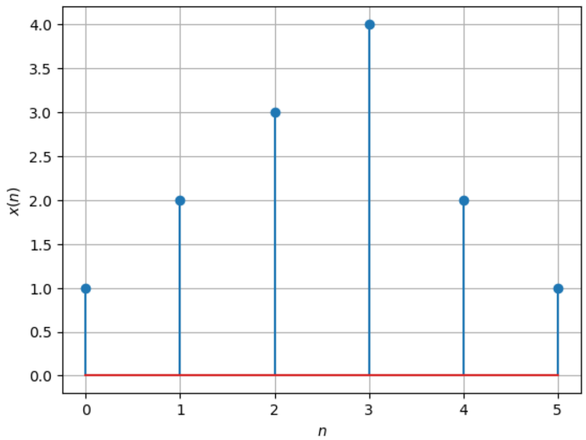
$$x(n) = \left\{ \underset{\uparrow}{1}, 2, 3, 4, 2, 1 \right\} \tag{3.1}$$

Sketch $x(n)$.
Solution:

```
import numpy as np
import matplotlib.pyplot as plt

x=np.array([1.0,2.0,3.0,4.0,2.0,1.0])

plt.stem(range(0,len(x)),x)
plt.ylabel("$x(n)$")
```

```
plt.xlabel("$n$")
plt.grid()
plt.show()
```

The above code yields

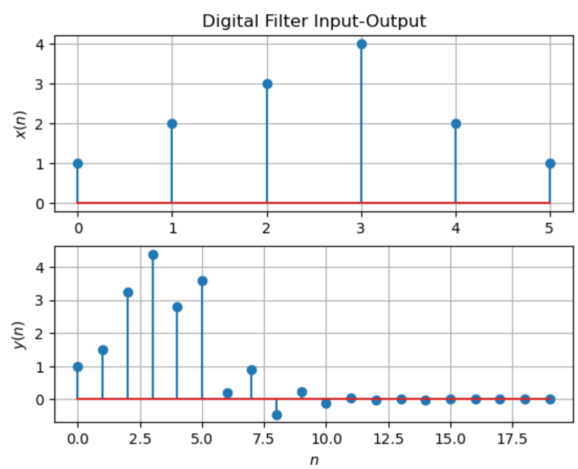


3.2 Let

$$y(n) + \frac{1}{2}y(n-1) = x(n) + x(n-2),$$
$$y(n) = 0, n < 0 \quad (3.2)$$

Sketch $y(n)$.
Solution:

```
import numpy as np
import matplotlib.pyplot as plt
#If using termux
import subprocess
import shlex
#end if

x=np.array([1.0,2.0,3.0,4.0,2.0,1.0])
k = 20
y = np.zeros(20)


y[0] = x[0]
y[1] = -0.5*y[0]+x[1]

for n in range(2,k-1):
        if n < 6:
                y[n] = -0.5*y[n-1]+x[n]+x
                    [n-2]
        elif n > 5 and n < 8:
                y[n] = -0.5*y[n-1]+x[n-2]
        else:
                y[n] = -0.5*y[n-1]
print(y)

#subplots
plt.subplot(2, 1, 1)
plt.stem(range(0,6),x)
plt.title('Digital Filter Input-Output')
plt.ylabel('$x(n)$')
plt.grid()# minor


plt.subplot(2, 1, 2)
plt.stem(range(0,k),y)
plt.xlabel('$n$')
plt.ylabel('$y(n)$')
plt.grid()# minor

#If using termux
# plt.savefig('../figs/xnyn.pdf')
# plt.savefig('../figs/xnyn.eps')
# subprocess.run(shlex.split("termux-open ../
    figs/xnyn.pdf"))
#else
plt.show()
```



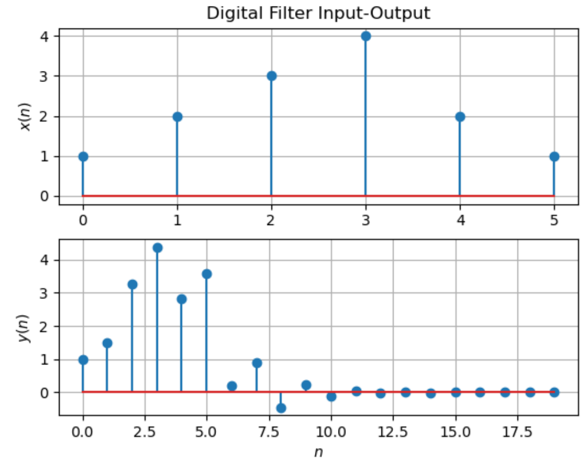
3.3 Repeat the above exercise using a C code.
Solution: The following C code generates data and saves it to a .dat file

```
#include <stdlib.h>
#include <stdio.h>
#define k 20

int main(){
    double x[6]={1,2,3,4,2,1};
    double y[k]={0};
    y[0]=x[0];
    y[1]=x[1]- 0.5*y[0];
    for(int i=2;i<k;i++){
```

```
        if (i<6)
            y[i]=−0.5∗y[i−1] + x[i]+x[i−2];
        else if(i<8)
            y[i]=−0.5∗y[i−1] +x[i−2];
        else
            y[i]=−0.5∗y[i−1];
    }
    int axes_x[sizeof(x)/sizeof(int)]={0};
    int axes_y[k]={0};

    FILE∗ fpy;
    fpy=fopen("3_3_data_y.dat","w");
    for(int i=0;i<k;i++){
        fprintf(fpy,"%f\n",y[i]);
    }
    fclose(fpy);
    FILE∗ fpx;
    fpx=fopen("3_3_data_x.dat","w");
    for(int i=0;i<6;i++){
        fprintf(fpx,"%f\n",x[i]);
    }
    fclose(fpx);
    return 0;
}
```

The following Python code sketches $x(n)$ and $y(n)$

```
import numpy as np
import matplotlib.pyplot as plt

y=np.loadtxt('3_3_data_y.dat',dtype="
    double")
x=np.loadtxt('3_3_data_x.dat',dtype="
    double")

plt.subplot(2,1,1)
plt.stem(range(0,len(x)),x)
plt.title("Digital Filter Input−Output")
plt.ylabel("$x(n)$")
plt.grid()

plt.subplot(2,1,2)
plt.stem(range(0,len(y)),y)
plt.ylabel("$y(n)$")
plt.xlabel("$n$")
plt.grid()
plt.show()
```



4 Z-TRANSFORM

4.1 The Z-transform of $x(n)$ is defined as

$$X(z) = \mathcal{Z}\{x(n)\} = \sum_{n=-\infty}^{\infty} x(n)z^{-n} \tag{4.1}$$

Show that

$$\mathcal{Z}\{x(n-1)\} = z^{-1}X(z) \tag{4.2}$$

and find

$$\mathcal{Z}\{x(n-k)\} \tag{4.3}$$

Solution: From (**??**),

$$\mathcal{Z}\{x(n-1)\} = \sum_{n=-\infty}^{\infty} x(n-1)z^{-n} \tag{4.4}$$

$$= \sum_{n=-\infty}^{\infty} x(n)z^{-n-1} = z^{-1}\sum_{n=-\infty}^{\infty} x(n)z^{-n} \tag{4.5}$$

resulting in (**??**). Similarly, it can be shown that

$$\mathcal{Z}\{x(n-k)\} = z^{-k}X(z) \tag{4.6}$$

4.2 Obtain $X(z)$ for $x(n)$ defined in problem **??**.
Solution:

$$Z(x(n)) = \sum_{n=-\infty}^{\infty} x(n)z^{-n} \tag{4.7}$$

$$= x(0)z^{0} + x(1)z^{-1} + x(2)z^{-2} + x(3)z^{-3} + \tag{4.8}$$

$$x(4)z^{-4} + x(5)z^{-5}$$

$$= 1 + 2z^{-1} + 3z^{-2} + 4z^{-3} + 2z^{-4} + z^{-5} \tag{4.9}$$

4.3 Find

$$H(z) = \frac{Y(z)}{X(z)} \tag{4.10}$$

from (**??**) assuming that the Z-transform is a linear operation.

Solution:

Applying (**??**) in (**??**),

$$Y(z) + \frac{1}{2}z^{-1}Y(z) = X(z) + z^{-2}X(z) \tag{4.11}$$

$$\implies \frac{Y(z)}{X(z)} = \frac{1 + z^{-2}}{1 + \frac{1}{2}z^{-1}} \tag{4.12}$$

4.4 Find the Z transform of

$$\delta(n) = \begin{cases} 1 & n = 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.13}$$

and show that the Z-transform of

$$u(n) = \begin{cases} 1 & n \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.14}$$

is

$$U(z) = \frac{1}{1 - z^{-1}}, \quad |z| > 1 \tag{4.15}$$

Solution: It is easy to show that

$$\delta(n) \overset{\mathcal{Z}}{\rightleftharpoons} 1 \tag{4.16}$$

and from (**??**),

$$U(z) = \sum_{n=0}^{\infty} z^{-n} \tag{4.17}$$

$$= \frac{1}{1 - z^{-1}}, \quad |z| > 1 \tag{4.18}$$

using the fomula for the sum of an infinite geometric progression.

4.5 Show that

$$a^n u(n) \overset{\mathcal{Z}}{\rightleftharpoons} \frac{1}{1 - az^{-1}} \quad |z| > |a| \tag{4.19}$$

Solution:

$$\mathcal{Z}\{a^n u(n)\} = \sum_{n=-\infty}^{\infty} a^n u(n) z^{-n} \tag{4.20}$$

$$= \sum_{n=-\infty}^{\infty} u(n)(az^{-1})^n \tag{4.21}$$

$$= \sum_{n=0}^{\infty} (az^{-1})^n, \quad \left|az^{-1}\right| < 1 \tag{4.22}$$

$$\tag{4.23}$$

$$= \frac{1}{1 - az^{-1}}, \quad |a| < |z| \tag{4.24}$$

using the fomula for the sum of an infinite geometric progression.

4.6 Let

$$H(e^{j\omega}) = H(z = e^{j\omega}). \tag{4.25}$$

Plot $|H(e^{j\omega})|$. Comment. $H(e^{j\omega})$ is known as the *Discret Time Fourier Transform* (DTFT) of $x(n)$.

Solution: The graph is symmetric and periodic. It is achieves a high of value 4 and a minimum value between 0 - 0.5. It is bounded between (0, 4) with period of 2π

$$H\left(e^{j\omega}\right) = \frac{1 + e^{-2j\omega}}{1 + \frac{e^{-j\omega}}{2}} \tag{4.26}$$

$$\implies \left|H\left(e^{j\omega}\right)\right| = \frac{\left|1 + e^{-2j\omega}\right|}{\left|1 + \frac{e^{-j\omega}}{2}\right|} \tag{4.27}$$

$$= \frac{\left|1 + e^{2j\omega}\right|}{\left|e^{2j\omega} + \frac{e^{j\omega}}{2}\right|} \tag{4.28}$$

$$= \frac{|1 + \cos 2\omega + j\sin 2\omega|}{\left|e^{j\omega} + \frac{1}{2}\right|} \tag{4.29}$$

$$= \frac{\left|4\cos^2(\omega) + 4j\sin(\omega)\cos(\omega)\right|}{|2e^{j\omega} + 1|} \tag{4.30}$$

$$= \frac{|4\cos(\omega)|\,|\cos(\omega) + j\sin(\omega)|}{|2\cos(\omega) + 1 + 2j\sin(\omega)|} \tag{4.31}$$

$$\therefore \left|H\left(e^{j\omega}\right)\right| = \frac{|4\cos(\omega)|}{\sqrt{5 + 4\cos(\omega)}} \tag{4.32}$$

The following code plots $\left|H\left(e^{j\omega}\right)\right|$

```
import numpy as np
```

```
import matplotlib.pyplot as plt

#DTFT
def H(z):
        num = np.polyval([1,0,1],z**(-1))
        den = np.polyval([0.5,1],z**(-1))
        H = num/den
        return H

#Input and Output
omega = np.linspace(-3*np.pi,3*np.pi,100)

#subplots
plt.plot(omega, abs(H(np.exp(1j*omega))))
plt.title('Filter Frequency Response')
plt.xlabel('$\omega$')
plt.ylabel('$|H(e^{\jmath\omega})| $')
plt.grid()# minor

plt.show()
```

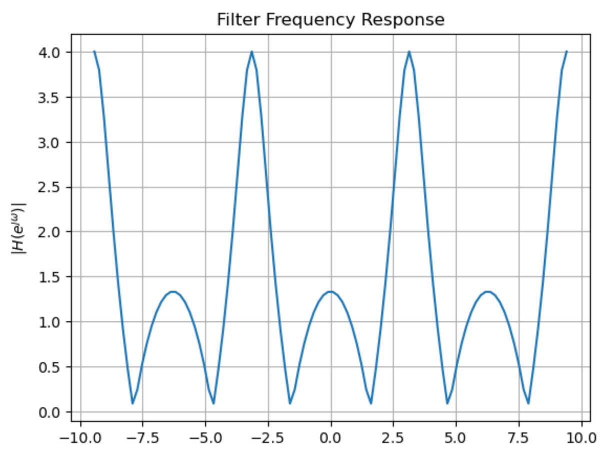


Fig. 4.6: $h(n)$ as the inverse of $H(z)$

4.7 Express $h(n)$ in terms of $H\left(e^{j\omega}\right)$.

Solution:

$$H\left(e^{j\omega}\right) = \sum_{n=-\infty}^{\infty} h(n)\, e^{-j\omega n} \tag{4.33}$$

$$\int_{-\pi}^{\pi} H\left(e^{j\omega}\right) e^{j\omega k} d\omega = \sum_{n=-\infty}^{\infty} h(n) \int_{-\pi}^{\pi} e^{-j\omega n} e^{j\omega k} d\omega \tag{4.34}$$

(4.35)

$$\int_{-\pi}^{\pi} e^{j\omega(n-k)} d\omega = \begin{cases} 2\pi & n = k \\ 0 & \text{otherwise} \end{cases} \tag{4.36}$$

$$\int_{-\pi}^{\pi} H\left(e^{j\omega}\right) e^{j\omega k} d\omega = h(n)\, 2\pi \tag{4.37}$$

$$\int_{-\pi}^{\pi} H\left(e^{j\omega}\right) e^{j\omega k} d\omega = 2\pi h(n) \tag{4.38}$$

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} H\left(e^{j\omega}\right) e^{j\omega k} d\omega = h(n) \tag{4.39}$$

5 Impulse Response

5.1 Using long division, find

$$h(n), \quad n < 5 \tag{5.1}$$

for H(z) in (**??**).
Solution:

$$H(z) = \frac{1 + z^{-2}}{1 + \frac{1}{2} z^{-1}} \tag{5.2}$$

Let $z^{-1} = x$,then, by polynomial long division we get

$$\begin{array}{r} 2x-4 \\ \frac{1}{2}x+1 \overline{\big) \; x^2 \qquad +1} \\ \underline{-x^2-2x} \\ -2x+1 \\ \underline{2x+4} \\ 5 \end{array}$$

$$\implies (1+z^{-2}) = (\frac{1}{2}z^{-1}+1)(2z^{-1}-4)+5 \tag{5.3}$$

$$\implies \frac{(1+z^{-2})}{\frac{1}{2}z^{-1}+1} = (2z^{-1}-4) + \frac{5}{\frac{1}{2}z^{-1}+1} \tag{5.4}$$

$$\implies H(z) = (2z^{-1}-4) + \frac{5}{\frac{1}{2}z^{-1}+1} \tag{5.5}$$

Now, consider $\frac{5}{\frac{1}{2}z^{-1}+1}$
The denominator $\frac{1}{2}z^{-1}+1$ can be expressed as sum of an infinite geometric progression, which as its first term equal to 1 and common ratio $\frac{-1}{2}z^{-1}$
Therefore, we can write $\frac{5}{\frac{1}{2}z^{-1}+1}$ as $5\left(1+\left(\frac{-1}{2}z^{-1}\right)+\left(\frac{-1}{2}z^{-1}\right)^2+\left(\frac{-1}{2}z^{-1}\right)^3+\left(\frac{-1}{2}z^{-1}\right)^4+\ldots\right)$
Therefore, H(z) can be given by,

$$H(z) = (2z^{-1}-4) + \frac{5}{\frac{1}{2}z^{-1}+1} \tag{5.6}$$

$$\tag{5.7}$$

$$= 2z^{-1} - 4 + 5 + \frac{-5}{2}z^{-1} + \frac{5}{4}z^{-2} + \frac{-5}{8}z^{-3} + \frac{5}{16}z^{-4} + \ldots \tag{5.8}$$

$$\implies H(z) = 1z^0 + \frac{-1}{2}z^{-1} + \frac{5}{4}z^{-2} + \frac{-5}{8}z^{-3} + \frac{5}{16}z^{-4} + \ldots \tag{5.9}$$

Comparing the above expression to (**??**) we get h(n) for n<5 as,

$$h(0) = 1 \tag{5.10}$$

$$h(1) = \frac{-1}{2} \tag{5.11}$$

$$h(2) = \frac{5}{4} \tag{5.12}$$

$$h(3) = \frac{-5}{8} \tag{5.13}$$

$$h(4) = \frac{5}{16} \tag{5.14}$$

5.2 Find an expression for $h(n)$ using $H(z)$, given that

$$h(n) \overset{\mathcal{Z}}{\rightleftharpoons} H(z) \tag{5.15}$$

and there is a one to one relationship between $h(n)$ and $H(z)$. $h(n)$ is known as the *impulse response* of the system defined by (**??**).
Solution: From (**??**),

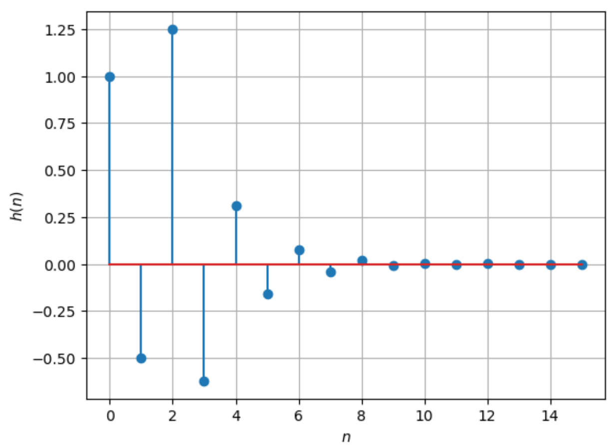
$$H(z) = \frac{1}{1+\frac{1}{2}z^{-1}} + \frac{z^{-2}}{1+\frac{1}{2}z^{-1}} \tag{5.16}$$

$$\implies h(n) = \left(-\frac{1}{2}\right)^n u(n) + \left(-\frac{1}{2}\right)^{n-2} u(n-2) \tag{5.17}$$

using (**??**) and (**??**).

5.3 Sketch $h(n)$. Is it bounded? Justify theoritically.
Solution: The following code plots Fig. **??**.

```
import numpy as np
import matplotlib.pyplot as plt

n=np.arange(14)
un=(−1/2)**n

hn1=np.pad(un,(0,2),'constant',
      constant_values=(0))
hn2=np.pad(un,(2,0),'constant',
      constant_values=(0))

hn=hn1+hn2

plt.stem(range(0,len(hn)),hn)
plt.grid()
plt.ylabel("$h(n)$")
plt.xlabel("$n$")
plt.show()
```



From (**??**) we know that

$$h(n) = \left(-\frac{1}{2}\right)^n u(n) + \left(-\frac{1}{2}\right)^{n-2} u(n-2) \quad (5.18)$$

Implies we can write that

$$h(n) = \begin{cases} 0 & ,n < 0 \\ \left(\frac{-1}{2}\right)^n & ,0 \le n < 2 \\ 5\left(\frac{-1}{2}\right)^n & ,n \ge 2 \end{cases} \quad (5.19)$$

A sequence is said to be bounded when

$$|x_n| \le M, \forall n \in \mathcal{N} \quad (5.20)$$

Now consider (**??**),
For n<0,

$$|h(n)| \le 0 \quad (5.21)$$

For $0 \le n < 2$,

$$|h(n)| = (\frac{1}{2})^n \quad (5.22)$$
$$\implies |h(n)| \le 1 \quad (5.23)$$

For $n \ge 2$,

$$|h(n)| = 5(\frac{1}{2})^n \quad (5.24)$$
$$\implies |h(n)| \le 5 \quad (5.25)$$

From above we can say that,

$$M = max\{0, 1, 5\} \quad (5.26)$$
$$= 5 \quad (5.27)$$

Therefore since M exists and is a real value, we can say that h(n) is bounded.

5.4 Convergent? Justify using the ratio test.
Solution:
Yes, it is convergent. We can clearly see in the plot it is not tending to infinite and remain finite.
For large n, we see that

$$h(n) = \left(-\frac{1}{2}\right)^n + \left(-\frac{1}{2}\right)^{n-2} \quad (5.28)$$
$$= \left(-\frac{1}{2}\right)^n (4+1) = 5\left(-\frac{1}{2}\right)^n \quad (5.29)$$
$$\implies \left|\frac{h(n+1)}{h(n)}\right| = \frac{1}{2} \quad (5.30)$$

and therefore, $\lim_{n\to\infty}\left|\frac{h(n+1)}{h(n)}\right| = \frac{1}{2} < 1$. Hence, we see that $h(n)$ converges.

5.5 The system with $h(n)$ is defined to be stable if

$$\sum_{n=-\infty}^{\infty} h(n) < \infty \quad (5.31)$$

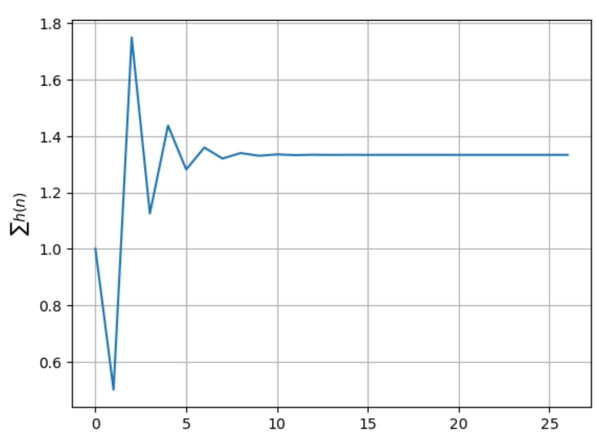
Is the system defined by (**??**) stable for the impulse response in (**??**)?
Solution: By using h(n) from 5.3

$$h(n) = \left(-\frac{1}{2}\right)^n u(n) + \left(-\frac{1}{2}\right)^{n-2} u(n-2) \quad (5.32)$$
$$= \sum_{n=-\infty}^{\infty} \left(-\frac{1}{2}\right)^n u(n) + \left(-\frac{1}{2}\right)^{n-2} u(n-2) \quad (5.33)$$
$$= \sum_{n=-\infty}^{\infty} \left(-\frac{1}{2}\right)^n u(n) + \sum_{n=-\infty}^{\infty} \left(-\frac{1}{2}\right)^{n-2} u(n-2) \quad (5.34)$$
$$= \sum_{n=-\infty}^{\infty} \left(-\frac{1}{2}\right)^n + \sum_{n=-\infty}^{\infty} \left(-\frac{1}{2}\right)^{n-2} \quad (5.35)$$
$$(5.36)$$
$$= \frac{2}{3} + \frac{2}{3} < \infty \quad (5.37)$$

5.6 Verify the above result using a python code.
Solution: The following code computes and plots at each n. We can see that the sum converges to a constant value as n tends to infinity.

```
import numpy as np
import matplotlib.pyplot as plt
from sympy import N

n=np.arange(25)
```

```
un=(−1/2)**n

hn1=np.pad(un,(0,2),'constant',
     constant_values=(0))
hn2=np.pad(un,(2,0),'constant',
     constant_values=(0))

hn=hn1+hn2

nh=len(hn)
sum_hn=np.zeros(nh)
sum_hn[0]=hn[0]
for i in range(1,nh):
     sum_hn[i]=sum_hn[i−1]+hn[i]

plt.plot(range(len(hn)),sum_hn)
plt.ylabel("$\sum{h(n)}$")
plt.xlabel("$n$")
plt.grid()
plt.show()
```



5.7 Compute and sketch $h(n)$ using

$$h(n) + \frac{1}{2}h(n-1) = \delta(n) + \delta(n-2), \quad (5.38)$$

This is the definition of $h(n)$.
Solution: The following code plots Fig. **??**.

Note that this is the same as Fig. **??**.

$$h(n) + \frac{1}{2}h(n-1) = \delta(n) + \delta(n-2) \quad (5.39)$$

$$H(z) + \frac{1}{2}z^{-1}H(z) = 1 + z^{-2} \quad (5.40)$$

$$H(z) = \frac{1}{1+\frac{1}{2}z^{-1}} + \frac{z^{-2}}{1+\frac{1}{2}z^{-1}} \quad (5.41)$$

$$h(n) = \left(-\frac{1}{2}\right)^{n} u(n) + \left(-\frac{1}{2}\right)^{n-2} u(n-2) \quad (5.42)$$

```
import numpy as np
import matplotlib.pyplot as plt

n=np.arange(14)
un=(−1/2)**n

hn1=np.pad(un,(0,2),'constant',
     constant_values=(0))
hn2=np.pad(un,(2,0),'constant',
     constant_values=(0))

hn=hn1+hn2

plt.stem(range(0,len(hn)),hn)
plt.grid()
plt.ylabel("$h(n)$")
plt.xlabel("$n$")
plt.show()
```

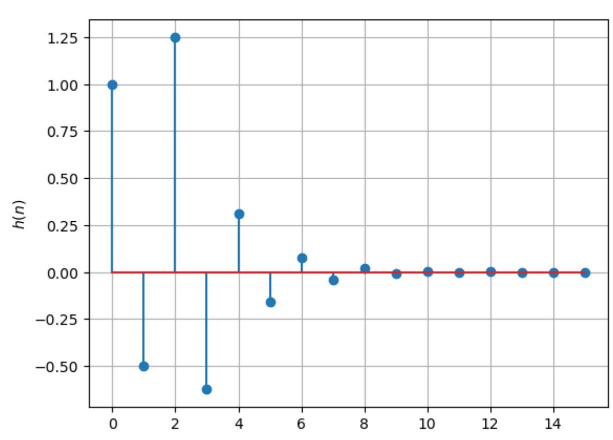


Fig. 5.7: $h(n)$ from the definition

5.8 Compute

$$y(n) = x(n) * h(n) = \sum_{n=-\infty}^{\infty} x(k)h(n-k) \quad (5.43)$$

Comment. The operation in (??) is known as *convolution*.
Solution: The following code plots $y(n)$.

```
import numpy as np
import matplotlib.pyplot as plt
from sympy import N

n=np.arange(25)
un=(−1/2)∗∗n

hn1=np.pad(un,(0,2),'constant',
     constant_values=(0))
hn2=np.pad(un,(2,0),'constant',
     constant_values=(0))

hn=hn1+hn2

nh=len(hn)
sum_hn=np.zeros(nh)
sum_hn[0]=hn[0]
for i in range(1,nh):
     sum_hn[i]=sum_hn[i−1]+hn[i]

plt.plot(range(len(hn)),sum_hn)
plt.ylabel("$\sum{h(n)}$")
plt.xlabel("$n$")
plt.grid()
plt.show()
```

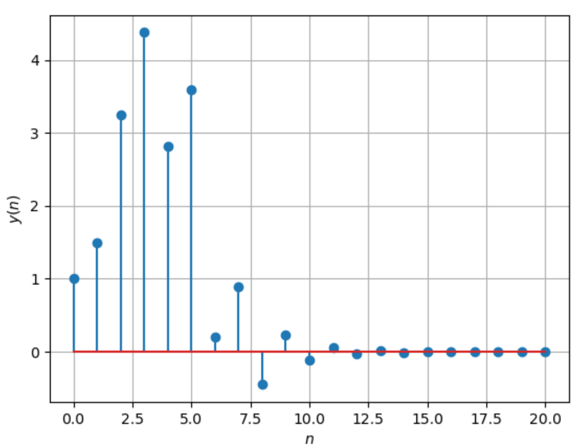


Fig. 5.8: $h(n)$ from the definition

5.9 Express the above convolution using a Teoplitz matrix.

Solution: We know that from, (??),

$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) \quad (5.44)$$

This can also be wrtten as a matrix-vector multiplication given by the expression,

$$y = T(h) * x \quad (5.45)$$

In the equation (??), $T(h)$ is a Teoplitz matrix. The equation (??) can be expanded as,

$$\mathbf{y} = \mathbf{x} \circledast \mathbf{h} \quad (5.46)$$

$$\mathbf{y} = \begin{pmatrix} h_1 & 0 & . & . & . & 0 \\ h_2 & h_1 & . & . & . & 0 \\ h_3 & h_2 & h_1 & . & . & 0 \\ . & . & . & . & . & . \\ h_{n-1} & h_{n-2} & h_{n-3} & . & . & 0 \\ h_n & h_{n-1} & h_{n-2} & . & . & h_1 \\ 0 & h_n & h_{n-1} & h_{n-2} & . & h_2 \\ . & . & . & . & . & . \\ 0 & . & . & . & 0 & h_{n-1} \\ 0 & . & . & . & 0 & h_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{pmatrix} \quad (5.47)$$

5.10 Show that

$$y(n) = \sum_{n=-\infty}^{\infty} x(n-k)h(k) \quad (5.48)$$

Solution: From (??), we substitute $k := n - k$ to get

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)\, h(n-k) \quad (5.49)$$

$$= \sum_{n-k=-\infty}^{\infty} x(n-k)\, h(k) \quad (5.50)$$

$$= \sum_{k=-\infty}^{\infty} x(n-k)\, h(k) \quad (5.51)$$

6 DFT and FFT

6.1 Compute

$$X(k) \triangleq \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}, \quad k = 0, 1, \ldots, N-1 \quad (6.1)$$

and $H(k)$ using $h(n)$.
Solution:

From **??**, we know that ,

$$x(n) = \left\{\underset{\uparrow}{1}, 2, 3, 4, 2, 1\right\} \tag{6.2}$$

Here, let, $\omega = e^{-j2\pi k}$. Then,

$$X(k) = 1 + 2\omega^{\frac{1}{5}} + 3\omega^{\frac{2}{5}} + 4\omega^{\frac{3}{5}} + 2\omega^{\frac{4}{5}} + \omega \tag{6.3}$$

Similarly, we know from (**??**),

$$h(n) = \begin{cases} 0 & , n < 0 \\ \left(\frac{-1}{2}\right)^n & , 0 \le n < 2 \\ 5\left(\frac{-1}{2}\right)^n & , n \ge 2 \end{cases} \tag{6.4}$$

Now,again let, $\omega = e^{-j2\pi k}$. Then,

$$H(k) = 1 + \frac{-1}{2}\omega^{\frac{1}{5}} + \frac{5}{4}\omega^{\frac{2}{5}} + \frac{-5}{8}\omega^{\frac{3}{5}} + \frac{5}{16}\omega^{\frac{4}{5}} + \frac{-5}{32}\omega \tag{6.5}$$

6.2 Compute

$$Y(k) = X(k)H(k) \tag{6.6}$$

Solution:
Now, from (**??**) and (**??**), we know $X(k) and H(k)$. Now, given that,

$$Y(k) = X(k) * H(k) \tag{6.7}$$

$$Y(k) = (1 + 2\omega^{\frac{1}{5}} + 3\omega^{\frac{2}{5}} + 4\omega^{\frac{3}{5}} + 2\omega^{\frac{4}{5}} + \omega) * \\ (1 + \frac{-1}{2}\omega^{\frac{1}{5}} + \frac{5}{4}\omega^{\frac{2}{5}} + \frac{-5}{8}\omega^{\frac{3}{5}} + \frac{5}{16}\omega^{\frac{4}{5}} + \frac{-5}{32}\omega) \tag{6.8}$$

$$Y(k) = 1 + \frac{3}{2}\omega^{\frac{1}{5}} + \frac{13}{4}\omega^{\frac{2}{5}} + \frac{35}{8}\omega^{\frac{3}{5}} + \frac{45}{16}\omega^{\frac{4}{5}} \\ \frac{115}{32}\omega^{\frac{5}{5}} + \frac{1}{8}\omega^{\frac{6}{5}} + \frac{25}{32}\omega^{\frac{7}{5}} - \frac{5}{8}\omega^{\frac{8}{5}} \\ - \frac{5}{32}\omega^{5} \tag{6.9}$$

where, $\omega = e^{-j2k\pi}$

6.3 Compute

$$y(n) = \frac{1}{N}\sum_{k=0}^{N-1} Y(k) \cdot e^{j2\pi kn/N}, \quad n = 0, 1, \ldots, N-1 \tag{6.10}$$

Solution: The following code plots Fig. **??** and computes $X(k)$ and $Y(k)$. Note that this is the same as $y(n)$ in Fig. **??**.

```
import numpy as np
import matplotlib.pyplot as plt

N=14
xtemp=np.array([1.0,2.0,3.0,4.0,2.0,1.0])
x=np.pad(xtemp, (0,8), 'constant',
    constant_values=(0))
n=np.arange(N)

X=np.zeros(N) + 1j*np.zeros(N)
for k in range(N):
    for i in range(N):
        X[k]+=x[i]*np.exp(-1j*2*np.pi*k*i/
            N)

un=(-1/2)**n
hn1=np.pad(un,(0,2),"constant",
    constant_values=(0))
hn2=np.pad(un,(2,0),"constant",
    constant_values=(0))
hn=hn1+hn2
H=np.zeros(N)+ 1j*np.zeros(N)
for k in range(N):
    for i in range(N):
        H[k]+=hn[i]*np.exp(-1j*2*np.pi*k*i
            /N)

Y=np.zeros(N)+1j*np.zeros(N)
for k in range(N):
    Y[k]=X[k]*H[k]
y=np.real(Y)
plt.stem(n,y)
plt.ylabel("$Y(k)$")
plt.xlabel("$k$")
plt.grid()
plt.show()
```

6.4 Repeat the previous exercise by computing $X(k), H(k)$ and $y(n)$ through FFT and IFFT.
Solution: The following python codes compute $X(k), H(k)$ and $y(n)$ through FFT and IFFT.

```
from scipy.fft import fft, ifft
import numpy as np
import matplotlib.pyplot as plt

x=np.array([1.0,2.0,3.0,4.0,2.0,1.0])
x=np.pad(x,(0,8),'constant',constant_values
    =(0))
```

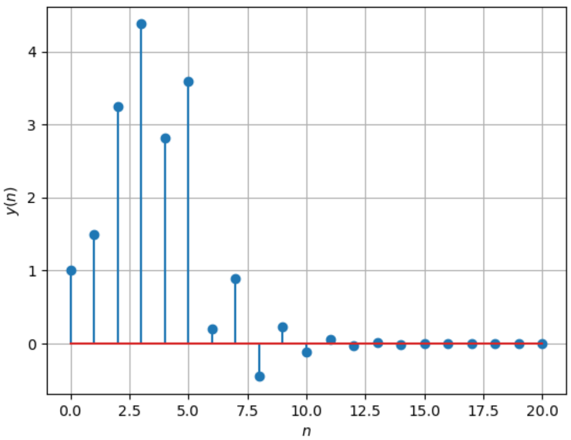


Fig. 6.3: $y(n)$ from the DFT

```
N=14
n=np.arange(N)
un=(−1/2)**n
hn1=np.pad(un,(0,2),'constant',
     constant_values=(0))
hn2=np.pad(un,(2,0),'constant',
     constant_values=(0))
hn=hn1+hn2

X=fft(x)
H=fft(hn[:N])
Y=np.zeros(N)+1j*np.zeros(N)
for i in range(N):
     Y[i]=X[i]*H[i]
y=ifft(Y)

plt.stem(range(0,N),np.real(X))
plt.title("Using FFT")
plt.ylabel("$X(k)$")
plt.grid()
plt.show()

plt.stem(range(0,N),np.real(H))
plt.title("Using FFT")
plt.ylabel("$H(k)$")
plt.grid()
plt.show()

plt.stem(range(0,N),np.real(y))
plt.title("Using FFT and IFFT")
plt.ylabel("$y(n)$")
plt.grid()
plt.show()
```

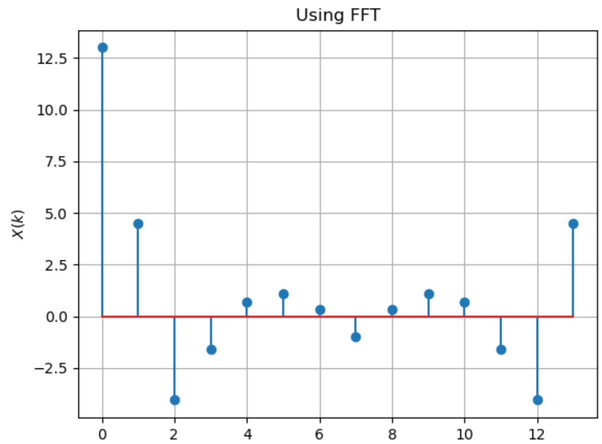


Fig. 6.4: $X(k)$ from the FFT

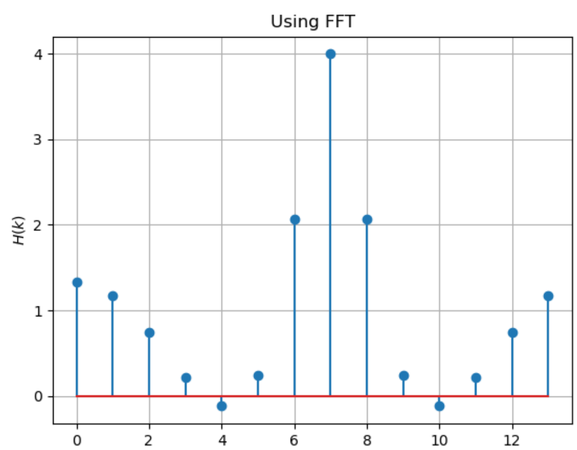


Fig. 6.4: $H(k)$ from the FFT

6.5 Wherever possible, express all the above equations as matrix equations.

Solution: We use the DFT Matrix, where $\omega = e^{-\frac{j2k\pi}{N}}$, which is given by

$$\mathbf{W} = \begin{pmatrix} \omega^0 & \omega^0 & \dots & \omega^0 \\ \omega^0 & \omega^1 & \dots & \omega^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \omega^0 & \omega^{N-1} & \dots & \omega^{(N-1)(N-1)} \end{pmatrix} \tag{6.11}$$

i.e. $W_{jk} = \omega^{jk}$, $0 \leq j, k < N$. Hence, we can write any DFT equation as

$$\mathbf{X} = \mathbf{W}\mathbf{x} = \mathbf{x}\mathbf{W} \tag{6.12}$$

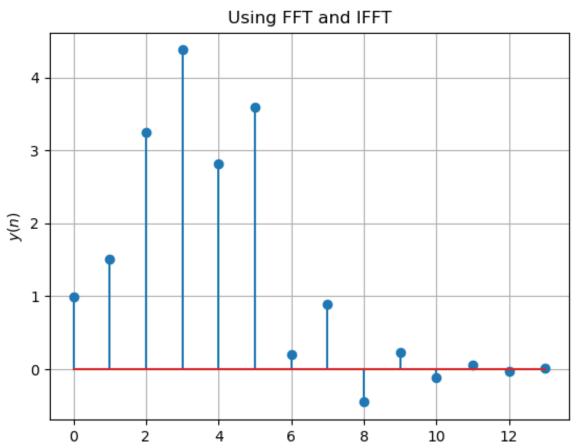


Fig. 6.4: $y(n)$ from the IFFT

where

$$\mathbf{x} = \begin{pmatrix} x(0) \\ x(1) \\ \vdots \\ x(n-1) \end{pmatrix} \tag{6.13}$$

Using (**??**), the inverse Fourier Transform is given by

$$\mathbf{x} = \mathcal{F}^{-1}(\mathbf{X}) = \mathbf{W}^{-1}\mathbf{X} = \frac{1}{N}\mathbf{W}^{\mathbf{H}}\mathbf{X} = \frac{1}{N}\mathbf{X}\mathbf{W}^{\mathbf{H}} \tag{6.14}$$

$$\implies \mathbf{W}^{-1} = \frac{1}{N}\mathbf{W}^{\mathbf{H}} \tag{6.15}$$

where H denotes hermitian operator. We can rewrite (**??**) using the element-wise multiplication operator as

$$\mathbf{Y} = \mathbf{H} \cdot \mathbf{X} = (\mathbf{W}\mathbf{h}) \cdot (\mathbf{W}\mathbf{x}) \tag{6.16}$$

The plot of $y(n)$ using the DFT matrix in Fig. (**??**) is the same as $y(n)$ in Fig. (**??**).

```
import numpy as np
from numpy.fft import fft, ifft
import matplotlib.pyplot as plt
#If using termux
#import subprocess
#import shlex
#end if

N = 14
n = np.arange(N)
fn=(−1/2)**n
hn1=np.pad(fn, (0,2), 'constant',
    constant_values=(0,0))
hn2=np.pad(fn, (2,0), 'constant',
    constant_values=(0,0))
h = hn1+hn2
xtemp=np.array([1.0,2.0,3.0,4.0,2.0,1.0])
x=np.pad(xtemp, (0,10), 'constant',
    constant_values=(0))
dftmtx = fft(np.eye(len(x)))
X = x@dftmtx
H = h@dftmtx
Y = H*X
invmtx = np.linalg.inv(dftmtx)
y = (Y@invmtx).real
#plots
plt.stem(range(0,16),y)
plt.xlabel('$n$')
plt.ylabel('$y(n)$')
plt.grid()# minor

#If using termux
#plt.savefig('../figs/6_5.png')
#subprocess.run(shlex.split("termux−open ../
    figs/yndft.pdf"))
#else
plt.show()
```

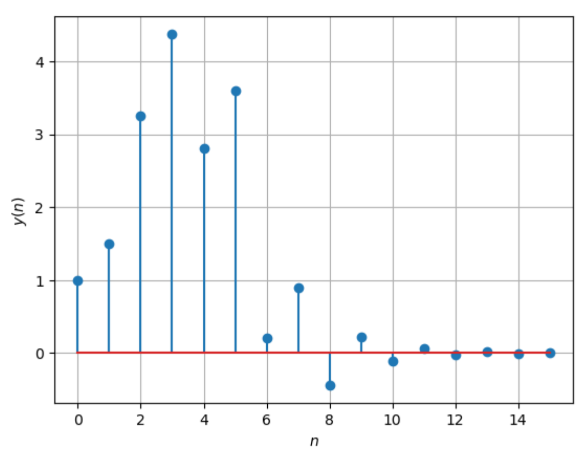


Fig. 6.5: $y(n)$ using the DFT matrix