Assignment 10

Om Shri Prasath (EE17B113)

April 7, 2019

1 Introduction

- We are going to use DFT to implement convolution of two digital signals. One of the main use of DFT is to convolute digital input signal with a filter, since it is the fastest and most efficient way of doing so.
- Inverse DFT of multiplication of the DFT of two signals will give us what is known as the *circular convolution* of both the signals, which is defined as follows:

If

$$Y(m) = X(m)H(m) \tag{1}$$

then

$$y[n] = \sum_{m=0}^{N-1} x[(n-m)\%N]h[m] \tag{2}$$

i.e.

$$y[n] = x[n] \circledast h[m] \tag{3}$$

- Normally, we would like to find only the linear convolution, which is the more useful convolution of the both, but finding circular convolution is easier, thus we find linear convolution from circular convolution using the following algorithm:
  - First, we spilt the input signal into different parts, let the length of each part be L
  - Let the length of filter signal be P. Thus, pad both the signals with zero, such that length of both the signals is N = L+P-1
  - Now, do circular convolution for each section of input digital signal with the filter. We will get sections of y with N length

  - The last P-1 samples of a y section will overlap with the first P-1 samples of the next y section. Thus, by adding the sections of y appropriately, we get the final output

- Next, we analyze circular correlation, by using the example of Zadoff-Chu sequence. The Zadoff-Chu sequence have very interesting properties which will be demonstrated here.

2 Python Code

2.1 Question 1

- Import the given filter and analyze it using the scipy.signal.freqz function.

Code:

```
import scipy.signal as sp
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import time

fil = pd.read_csv("/home/omshripc/Sem 4/EE2703/Answers/h.csv",
                  header=None)
fil = fil.values
fil = fil[:, 0]
w, h = sp.freqz(fil)

plt.title("Magnitude Response of Filter")
plt.plot(w, np.abs(h))

plt.ylabel(r"$|H(j\omega)| \to$")
plt.xlabel(r"$w \to$")

plt.show()

ph = np.unwrap(np.angle(h))
plt.title("Phase Response of Filter")
plt.plot(w, ph)

plt.ylabel(r"$\angle H(j\omega) \to$")
plt.xlabel(r"$w \to$")
plt.show()
```

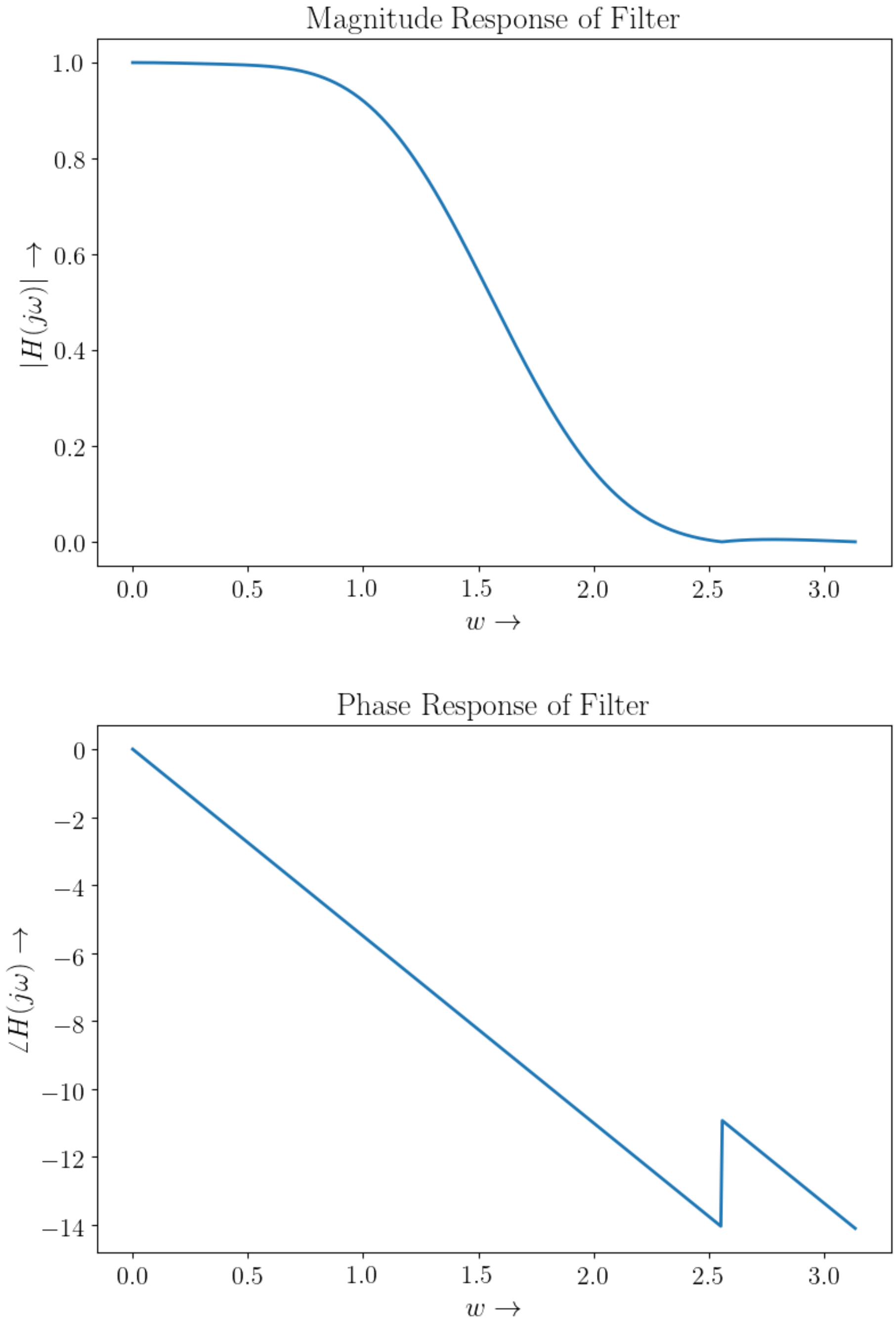


Figure 1: Magnitude and Phase Response of Filter

2.1.1 Results & Discussion:

- From the phase plot, we can see that the filter is a *Lowpass Filter*. The 3db cutoff frequency is approximately 30^o.

- From the phase plot, we can conclude that the given filter is a *Linear Fitler*. Thus, there will be a time delay in the output of the function while compared to the input function

2.2 Question 2

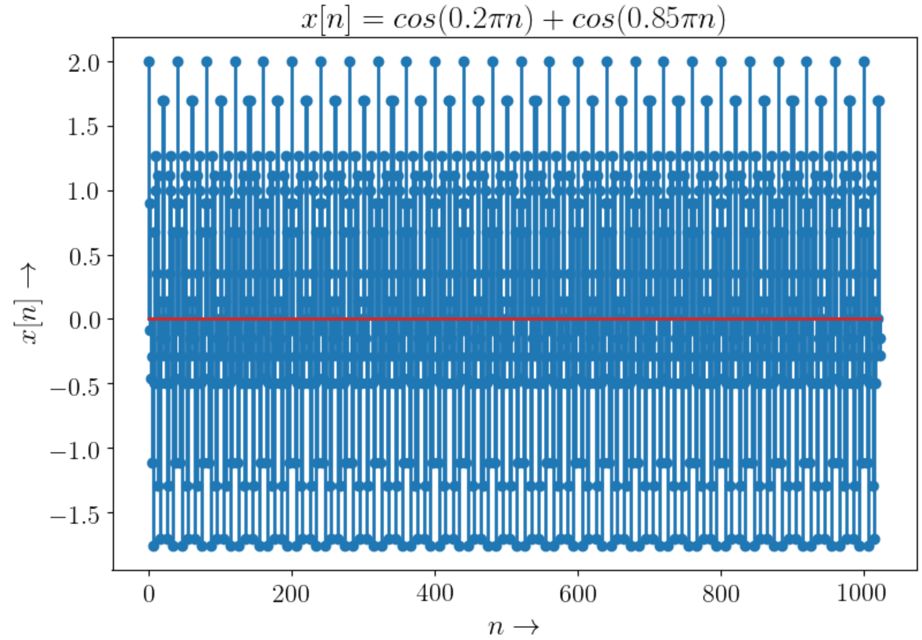
- Generate the function $x[n] = cos(0.2\pi n) + cos(0.85\pi n)$ for $n \in [0, 1024]$.
- Pass it through the filter via linear convolution and circular convolution. Observe the differences

Code:

```
n = np.linspace(0, 1024, 1025)
n = n[:-1]
x = np.cos(np.pi*0.2*n)+np.cos(0.85*np.pi*n)

plt.title(r"$x[n] = cos(0.2\pi n)+cos(0.85\pi n)$")
plt.stem(n, x)
plt.xlabel(r"$n \to$")
plt.ylabel(r"$x[n] \to$")
plt.show()

plt.title(r"$x[n]$ for n $\in$ [0,100]")
plt.stem(n, x)
plt.xlabel(r"$n \to$")
plt.ylabel(r"$x[n] \to$")
plt.xlim([0, 100])
plt.show()
```



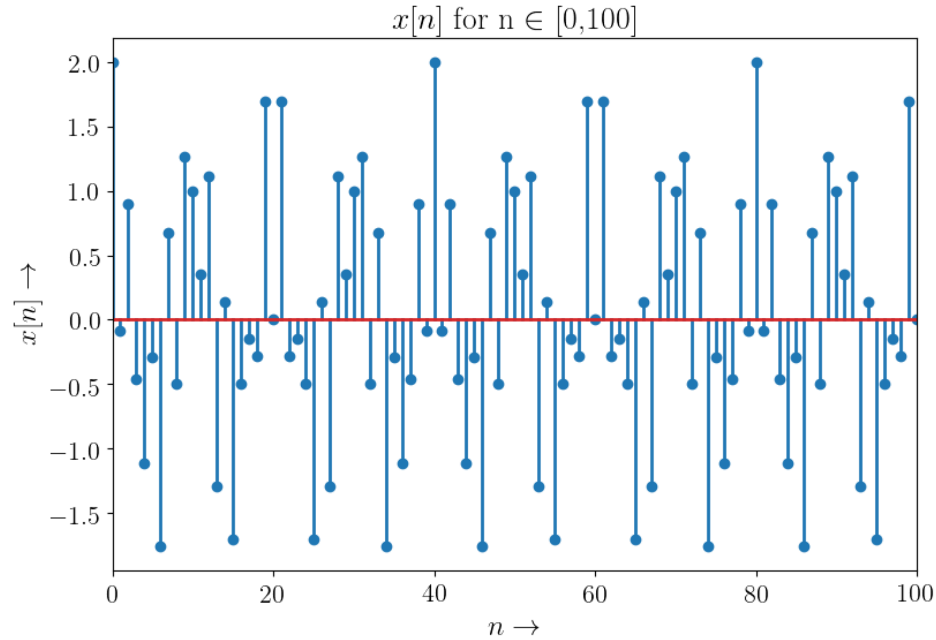


Figure 2: $x[n] = cos(0.2\pi n) + cos(0.85\pi n)$

Code:

```
y_conv = np.convolve(x, fil)

plt.title(r"$y[n] = x[n]*h[n]$")
plt.stem(y_conv)
plt.xlabel(r"$n \to$")
plt.ylabel(r"$y[n] \to$")
plt.show()

plt.xlim([0, 100])
plt.title(r"$y[n]$ zoomed to start")
plt.stem(y_conv)
plt.xlabel(r"$n \to$")
plt.ylabel(r"$y[n] \to$")
plt.show()


plt.xlim([900, 1050])
plt.title(r"$y[n]$ zoomed to end")
plt.stem(y_conv)
plt.xlabel(r"$n \to$")
plt.ylabel(r"$y[n] \to$")
plt.show()

fil_rep = np.concatenate([fil, np.zeros(len(x) - len(fil))])
y1 = np.fft.ifft(np.fft.fft(x)*np.fft.fft(fil_rep))


plt.title(r"$y'[n] = x[n] \circledast h[n]$")
plt.stem(y1)
plt.xlabel(r"$n \to$")
plt.ylabel(r"$y'[n] \to$")
plt.show()

plt.xlim([0, 100])
plt.title(r"$y'[n]$ zoomed to start")
plt.stem(y1)
plt.xlabel(r"$n \to$")
plt.ylabel(r"$y'[n] \to$")
plt.show()


plt.xlim([900, 1050])
```

```
plt.title(r"$y'[n]$ zoomed to end")
plt.stem(y1)
plt.xlabel(r"$n \to$")
plt.ylabel(r"$y'[n] \to$")
plt.show()
```

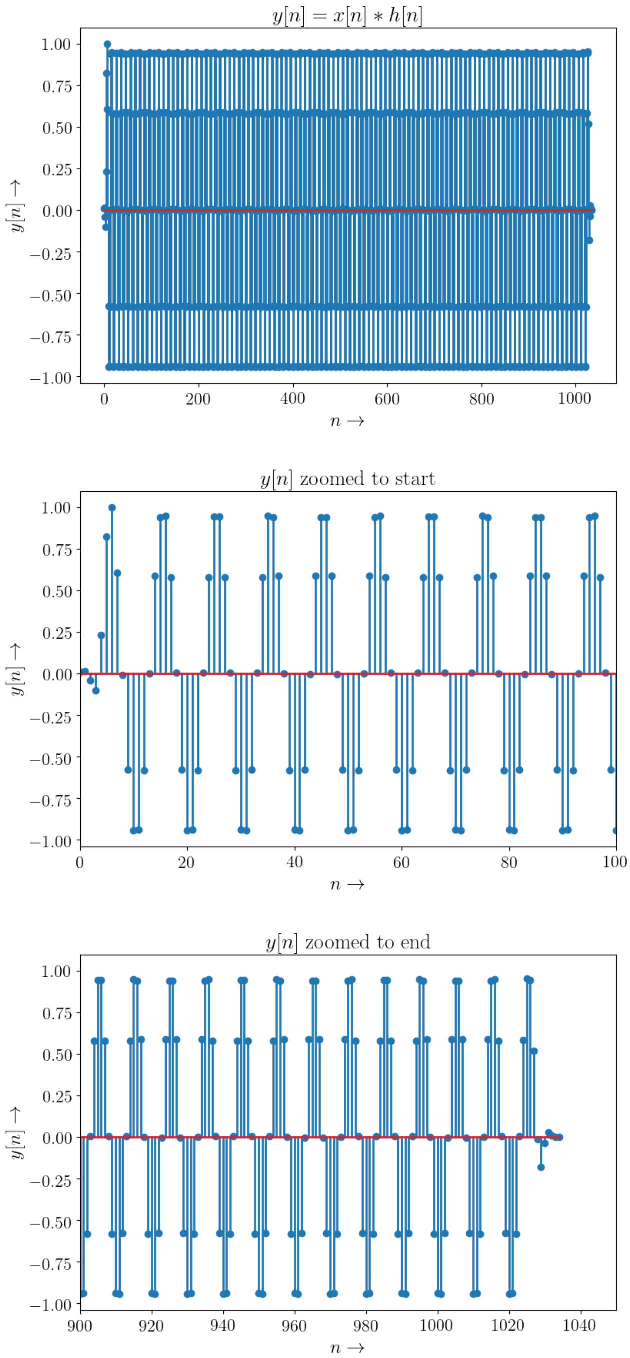


Figure 3: Linear Convolution

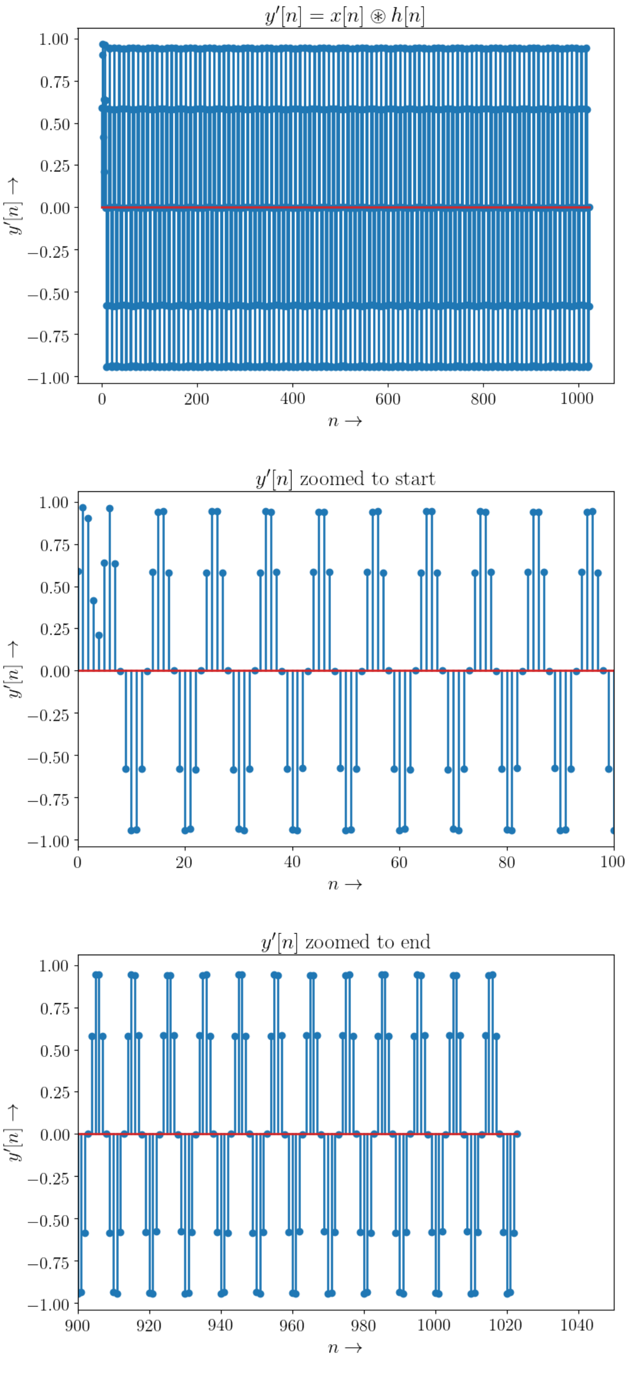


Figure 4: Circular Convolution

2.2.1 Results & Discussion:

- There are distinct differences in both the convolutions
- One of the main difference is that the output of both the convolutions are not of same size.
- Also in the start, both the convolutions do not look similar, although they do so afterwards.
- This is expected behaviour, since the method by which we find the convolutions differ greatly and this is reflected in their output

2.3 Question 3

- Now we are to convolve both the signals linearly using circular convolution

Code:

```
x_split = np.array(np.split(x, 1024/16))
P = len(fil)
L = x_split.shape[1]
N = L+P-1
y_part = np.zeros((x_split.shape[0], N))
fil_new = np.concatenate([fil, np.zeros(N-len(fil))])
for i in range(x_split.shape[0]):
    x_new = np.concatenate([x_split[i], np.zeros(N-len(x_split[i]))])
    y_part[i] = np.array(np.fft.ifft(np.fft.fft(x_new)*np.fft.fft(fil_new
y = np.zeros(len(x)+len(fil))

J = len(y_part[0])
K = J-(P-1)
for i in range(y_part.shape[0]):
    y[i*K:i*K+J] += y_part[i]


plt.title(r"$y[n] = x[n]*h[n]$ using circular convolution")
plt.stem(y)
plt.xlabel(r"$n \to$")
plt.ylabel(r"$y'[n] \to$")
plt.show()
```

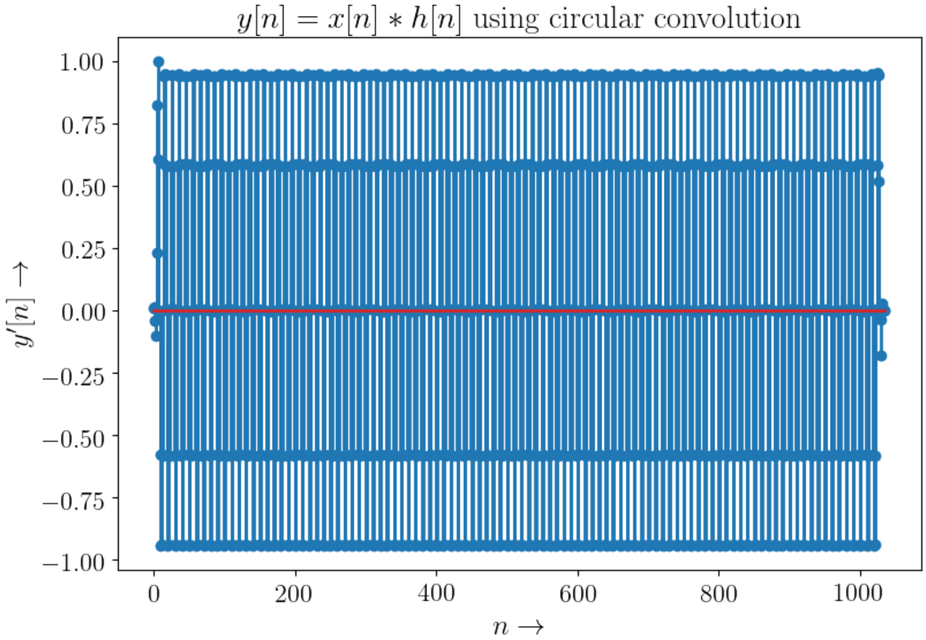


Figure 5: Linear Convolution using Circular Convolution

2.3.1 Results & Discussion:

- Thus, we have done the linear convolution using circular convolution

2.4 Question 4

- We are going to analyse the Zadoff-Chu by both linearly autocorrelating and circularly autocorrelating it with a cyclic shifted version of the signal
- Correlation is given by

$$y[n] = x[n] * h[-n] \tag{4}$$

- Autocorrelation is when a signal is correlated with itself
- The definition holds in circular correlation too, i.e.

$$y[n] = x[n] \circledast h[-n] \tag{5}$$

- In DFT domain, it becomes

$$Y[m] = X[m]H^*[m] \tag{6}$$

- $H^*[m]$ is the conjugate of $H[m]$

Code:

```
z_c = pd.read_csv("/home/omshripc/Sem 4/EE2703/Answers/x1.csv",
                  header=None).values[:, 0]

z_c = np.array([complex(z_c[i].replace("i", "j")) for i in range(len(z_c)


plt.title(r"Magnitude of Zadoff-Chu sequence ($z[n]$)")
plt.plot(np.round(np.abs(z_c)), 'bo')

plt.xlabel(r"$n \to$")
plt.ylabel(r"$z[n] \to$")
plt.show()

z_c_rot = np.concatenate([z_c[5:], z_c[0:5]])

y_corr = np.correlate(z_c, z_c_rot, "full")
n = np.linspace(-len(z_c), len(z_c), len(y_corr))

plt.title(r"$p[n]$ = Correlation of $z[n]$ with z[n] cyclically shifted b

plt.stem(n, np.abs(y_corr))
plt.xlabel(r"$n \to$")
```

```
plt.ylabel(r"$p[n] \to$")
plt.show()

plt.title(r"$p[n]$ for n $\in$ [3,7]")
plt.stem(n, np.abs(y_corr))
plt.xlim([3, 7])
plt.xlabel(r"$n \to$")
plt.ylabel(r"$p[n] \to$")
plt.show()

y_cir_conv = np.fft.ifft(np.fft.fft(z_c)*np.fft.fft(np.conj(z_c_rot)))
plt.title(
    r"$q[n]$ = Circular Correlation of $z[n]$ with z[n] cyclically shifte

plt.stem(np.abs(y_cir_conv))
plt.xlabel(r"$n \to$")
plt.ylabel(r"$p[n] \to$")
plt.show()

plt.title(r"$q[n]$ for n $\in$ [830,838]")
plt.stem(np.abs(y_cir_conv))
plt.xlim([830, 838])
plt.xlabel(r"$n \to$")
plt.ylabel(r"$p[n] \to$")
plt.show()
```

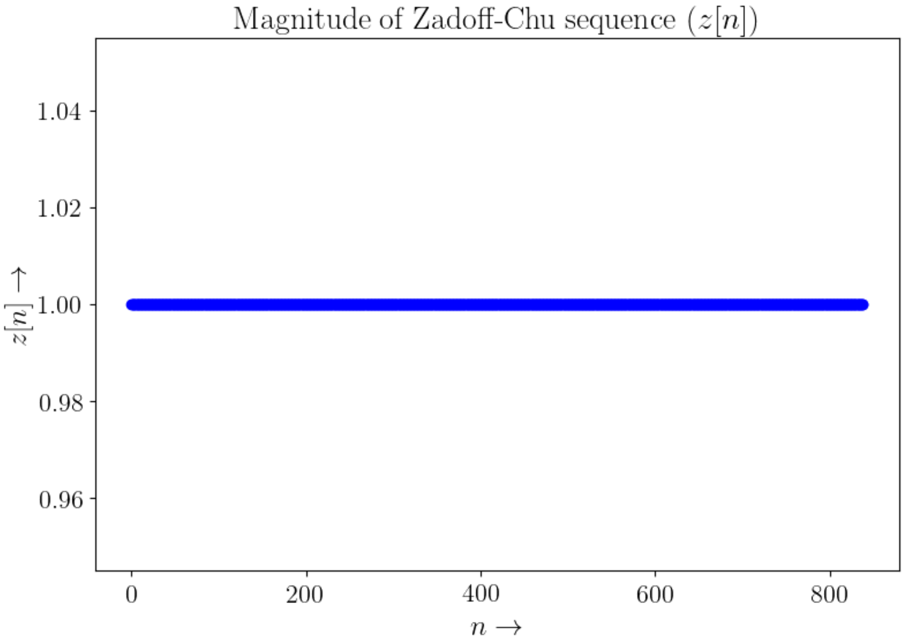


Figure 6: Magnitude of Zandoff-Chu sequence

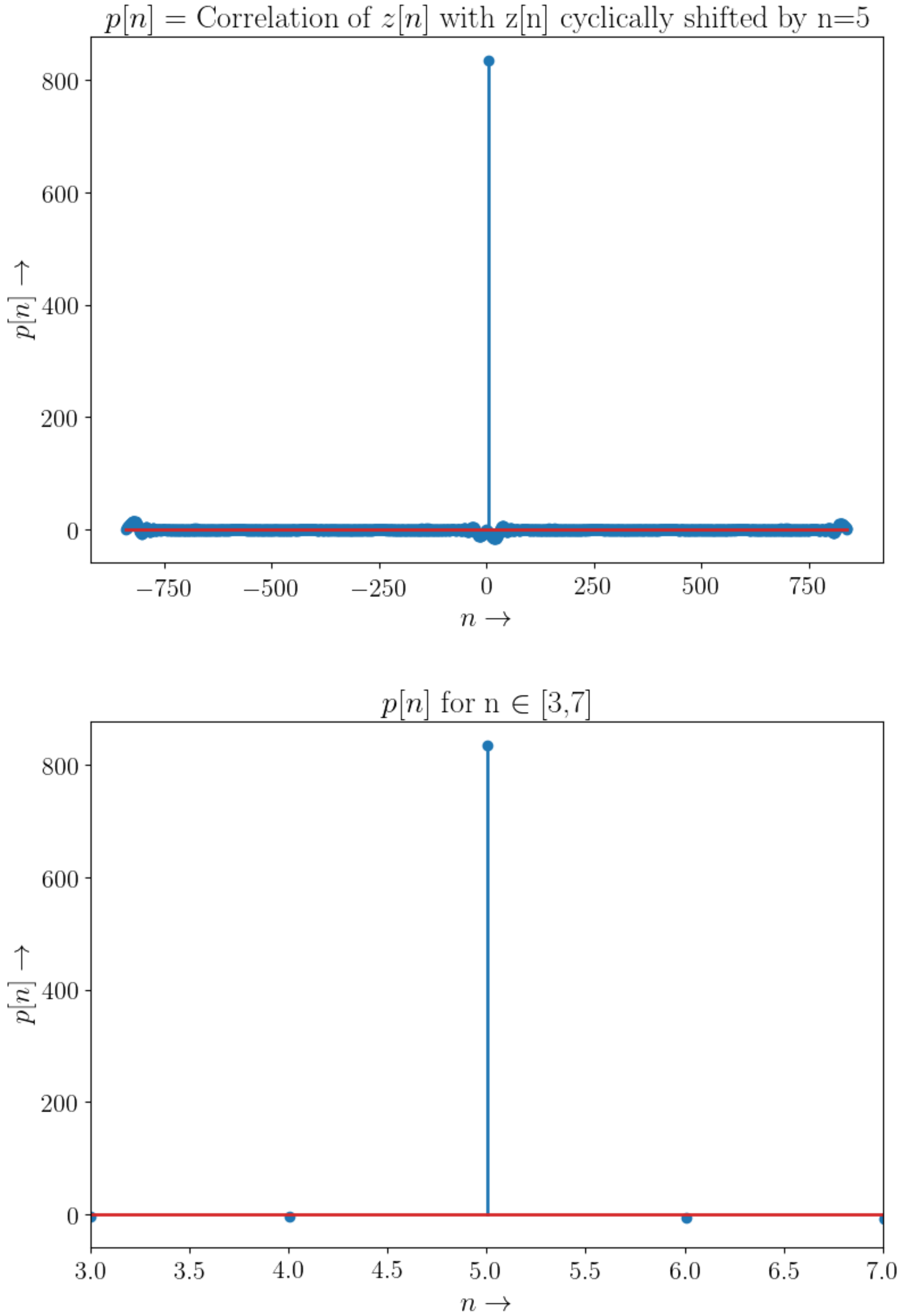


Figure 7: Linear Convolution of Zandoff-Chu sequence

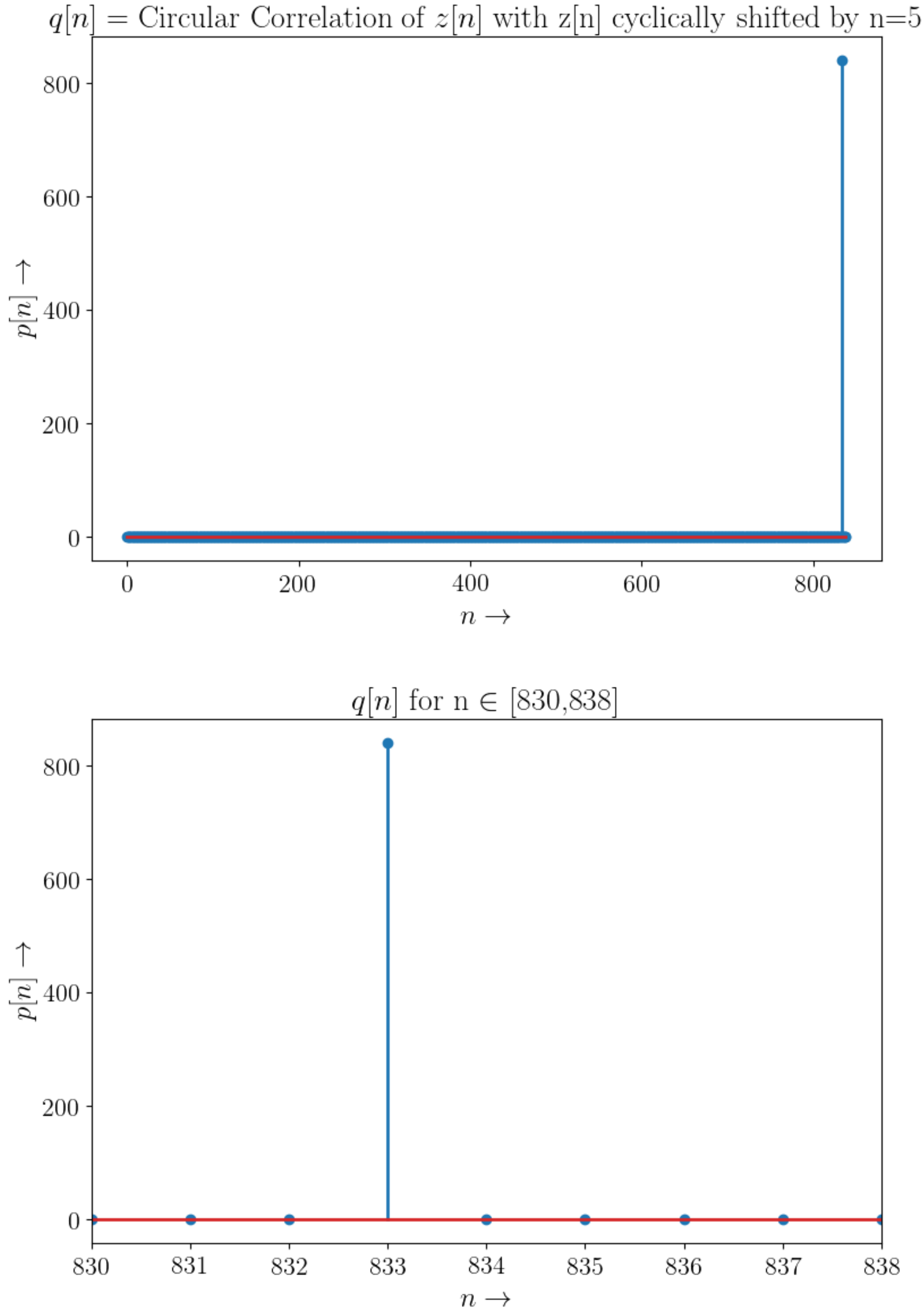


Figure 8: Linear Convolution of Zandoff-Chu sequence

2.4.1 Results & Discussion:

- We see that the magnitude of the Zandoff-Chu sequence is 1, which is a property of the sequence
- We can clearly see that there are peaks at the particular points and the function is zero at all other points
- The peak is at the value by which we cyclically shifted our signal for the linear convolution, but it is peaking at the value of shift from the last for the circular correlation
- This is the special property of the Zandoff-Chu sequence, i.e. the correlation of the sequence with a cyclic shifted version of it is zero everywhere but only peaks at the value of its shift.
- Also, since the circular correlation is defined as such, the peak now occurs at the value of shift from the last.

3 Conclusion

Thus we were able to linearly convolve two signals using circular convolution and were also able to study the property of Zandoff-Chu sequence