ASSIGNMENT 10: Linear and Circular Convolution

Hariharan P EE20B042

April 28, 2022

1 Brief Up:

The tasks of this assignment is the following:

- Realising linear and circular convolutions
- Using circular convolution to implement linear convolution
- Autocorrelation of Zadoff-Chu Sequence

2 plotting function:

```
def plot_fun(x,y,x1='w',y1='Magnitude',t1='xyz'):
    plot(x,y)
    xlabel(x1)
    ylabel(y1)
    title(t1)
    grid(True)
    show()
```

3 Interpreting the filter:

We use freqz to convert the given filter from the time domain to the frequency domain.

```
details =np.zeros(12)
i = 0
with open("h.csv", 'r') as file:
    for line in file:
        details[i] = float(line)
        i+=1

w,h = sig.freqz(details)
```

```
plot_fun(w,abs(h),'w','Magnitude','Low pass filter')
plot_fun(w,angle(h),'w','Phase','Low pass filter')
```

The plots obtained represent an LPF

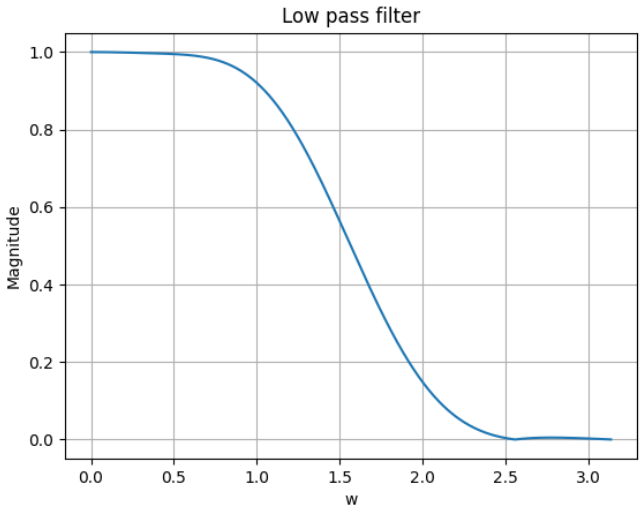


Figure 1: Magnitude plot of transfer function of filter

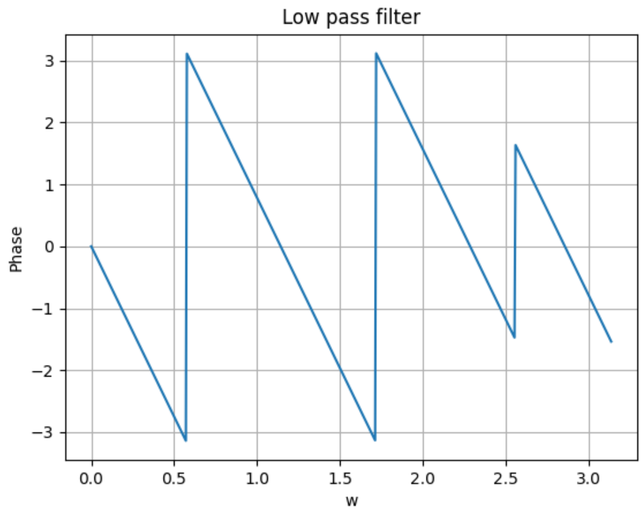


Figure 2: Phase plot of transfer function of filter

4 Generating given signal:

We generate the input signal given.

```
n = arange(2**10)
x = cos(0.2*pi*n) + cos(0.85*pi*n)
plot_fun(n,x,'n','Amplitude','Input signal')
```

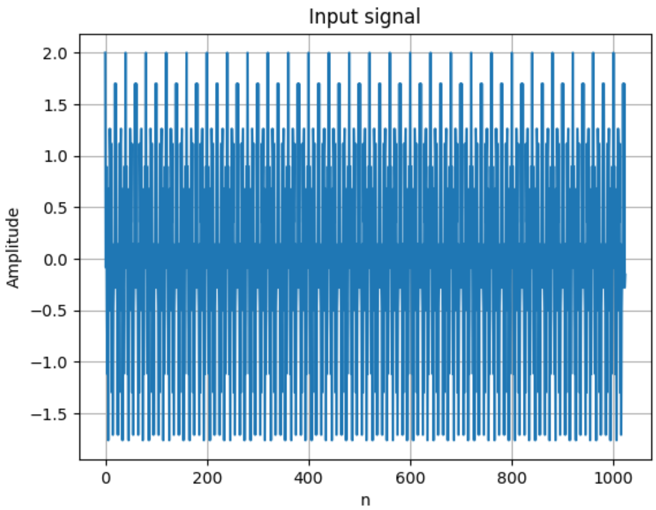


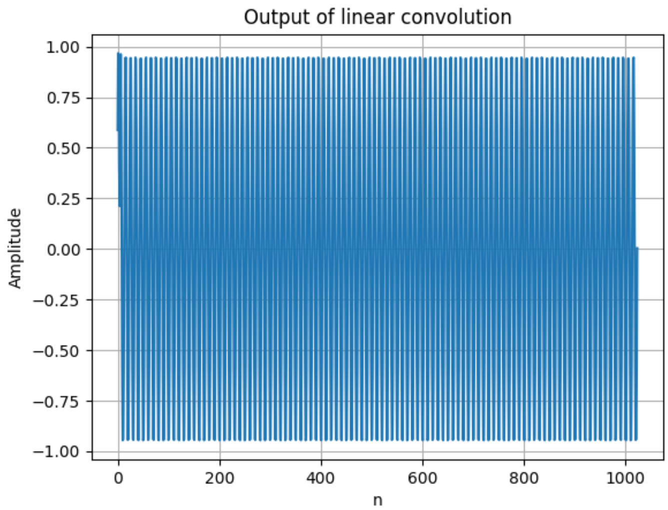
Figure 3: Fourier transform

5 Linear convolution:

We do linear convolution using the formula

$$y[n] = \sum_{k=0}^{n-1} x[n-k]h[k] \tag{1}$$

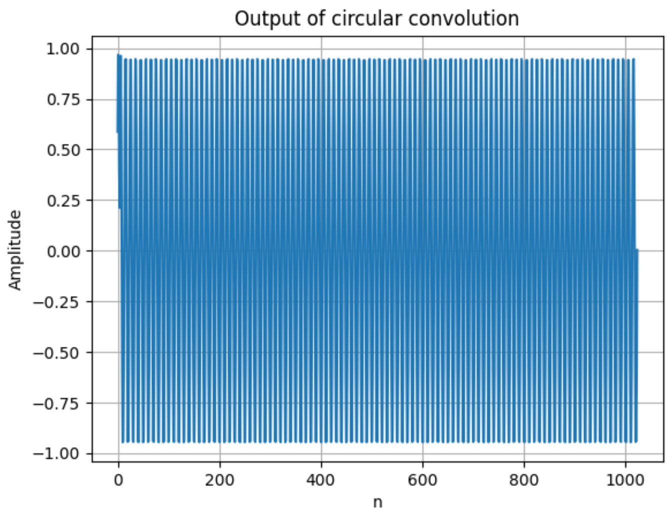
```
y = np.zeros(len(x))
for i in arange(len(x)):
    for k in arange(len(details)):
        y[i]+=x[i-k]*details[k]
plot_fun(n,y,'n','Amplitude','Output of linear convolution')
```



6 Circular Convolution:

To solve this more efficiently we shift to circular convolutions, where we convert to the frequency domain, multiply and go back to the time domain.

```
y=ifft(fft(x)*fft(concatenate((details,zeros(len(x)-len(details))))))
plot_fun(n,real(y),'n','Amplitude','Output of circular convolution')
```

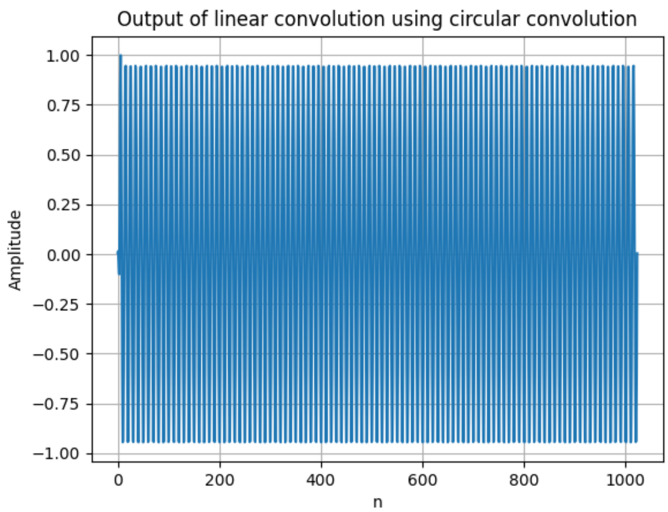


7 Linear convolution using circular convolution:

- We observe that this efficient solution is non causal and requires the entire signal to perform a convolution.
- So, We implement a linear method of circular convolution so that we need not depend on the next infinite values in the input but only a finite number of values(this is still non causal but the delay in the response is lower)

```
def circular_conv(x,h):
    P = len(h)
    n_ = int(ceil(log2(P)))
    h_ = np.concatenate((h,np.zeros(int(2**n_)-P)))
    P = len(h_)
    n1 = int(ceil(len(x)/2**n_))
    x_ = np.concatenate((x,np.zeros(n1*(int(2**n_))-len(x))))
    y = np.zeros(len(x_)+len(h_)-1)
    for i in range(n1):
        temp = np.concatenate((x_[i*P:(i+1)*P],np.zeros(P-1)))
        y[i*P:(i+1)*P+P-1] += np.fft.ifft(np.fft.fft(temp) * np.fft.fft(
    return y

y = circular_conv(x,details)
plot_fun(n,real(y[:1024]),'n','Amplitude','Output of linear convolution u
```



8 Correlation of Zadoff Chu Sequence:

We analyse the Zadoff Chu Sequence which has the following properties:

- complex sequence,
- constant amplitude sequence,
- The auto correlation of a Zadoff–Chu sequence with a cyclically shifted version of itself is zero,
- Correlation of Zadoff–Chu sequence with the delayed version of itself will give a peak at that delay.t

The output obtained for correlation with a shifted version of itself was completely in line with these properties Given above.This is what we should graphically show.

```
lines = []
with open("x1.csv",'r') as file2:
    csvreader = csv.reader(file2)
    for row in csvreader:
        lines.append(row)
lines2 = []
for line in lines:
    line = list(line[0])
    try :
        line[line.index('i')]='j'
        lines2.append(line)
    except ValueError:
        lines2.append(line)
        continue
x = [complex(''.join(line)) for line in lines2]
x2 = np.roll(x,5)
cor = np.fft.ifftshift(np.correlate(x2,x,'full'))
xlim(0,25)
plot_fun(linspace(0,len(cor)-1,len(cor)),abs(cor),'t','Correlation','auto
```

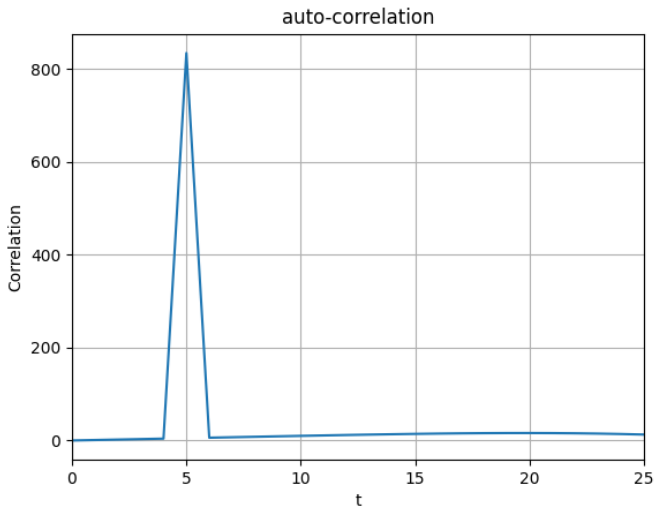


Figure 4: Correlation between Zadoff Chu Sequence and its 5samples shifted version

9 Conclusion:

- We implemented different algorithms for convolution.
- We see that using circular convolution to figure out linear convolution is more efficient.
- We see that the peak is obtained at the nth sample is the zadoff sequence is correlated with its nth sample shifted version.