EE2703 : Applied Programming Lab
Assignment 10
Linear and Circular Convolution

Debojyoti Mazumdar
EE20B030

June 1, 2022

0.1 Abstract

Aim of this assignment is to know how to get the linear convolution and circular convolution of two signals using python.

0.2 Introduction

We have 3 ways to convolve two signals in python and they are as follows.

`Linear Convolution:`
We can do a linear convolution of two signals $y[n]$ and $h[n]$ using the direct summation algorithm which would include two for loops.

```
for n corresponding to x: # iterate over indices in x
  y[n]=0;                 # initialize y to zero
  for k in range(K):      # iterate over indices in h
    y[n] += x[n-k]*h[k]   # compute the convolution sum
  #end for
#end for
```

We can do this linear convolution using just one for loop by using the python function *np.convolve*.

`Circular Convolution:`
To do a circular convolution we first make the non-periodic signals periodic.
let $x[n]$ be a sequence of N values.Then

$$\tilde{x}[n] = \begin{cases} x[n] & 0 \leq n < N \\ x[n-N] & N \leq n < 2N \\ x[n+N] & -N \leq n < 0 \end{cases}$$

This way we convert the two non-periodic signals $x[n]$ and $h[n]$ to periodic signals.
Now we use normal convolution on these periodic signals to obtain the circular convolution of the non-periodic signals.

$$\tilde{y}[n] = \sum_{m=0}^{N-1} \tilde{x}[n-m]\tilde{h}[m]$$

`Circular Convolution as linear convolution using aliasing:`
To do this we use the following algorithm.

- Suppose $h[]$ fits into a 2^m window. We zero pad $h[k]$ if required.
- Divide $x[]$ into sections 2^m long.
- Apply circular convolution to each section of $x[]$.
- Stitch the outputs back together.

0.3 Assignment

0.3.1 Question 1 and Question 2

Here, we are going to plot the magnitude and phase response of the signal given in h.csv.
To do so we first read the h.csv file and put the values into a list. To do that we first download the file and then write the following code to read it.

```
def Q1(printing=False, filename="h.csv"):
    f = open(filename)

    b = f.readlines()
    for i in range(len(b)):
        b[i] = float(b[i].strip("\n"))
    f.close()

    b = np.array(b)

    if printing:
        print("The coefficients are :\n")
        print(b)

    return b
```

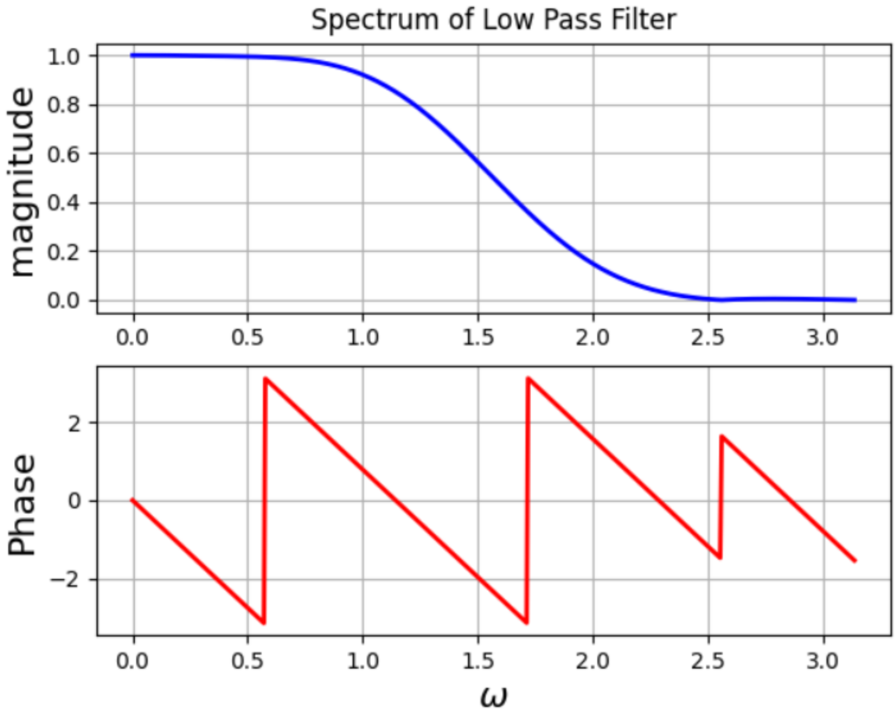
Then we use $scipy.signal.freqz()$ to obtain the spectrum and the phase of the signal.
So we have to write just one line of code to obtain it.

```
w, h = sig.freqz(Q1())
```

Now we have the code to plot it.

```
figure()
subplot(2, 1, 1)
plot(w, abs(h), 'b', lw=2)
ylabel(r"magnitude", size=16)
title(r"Spectrum of Low Pass Filter")
grid(True)
subplot(2, 1, 2)
plot(w, angle(h), 'r', lw=2)
ylabel(r"Phase", size=16)
xlabel(r"$\omega$", size=16)
grid(True)
show()
```

Which when run we get the following plot.



0.3.2 Question 3

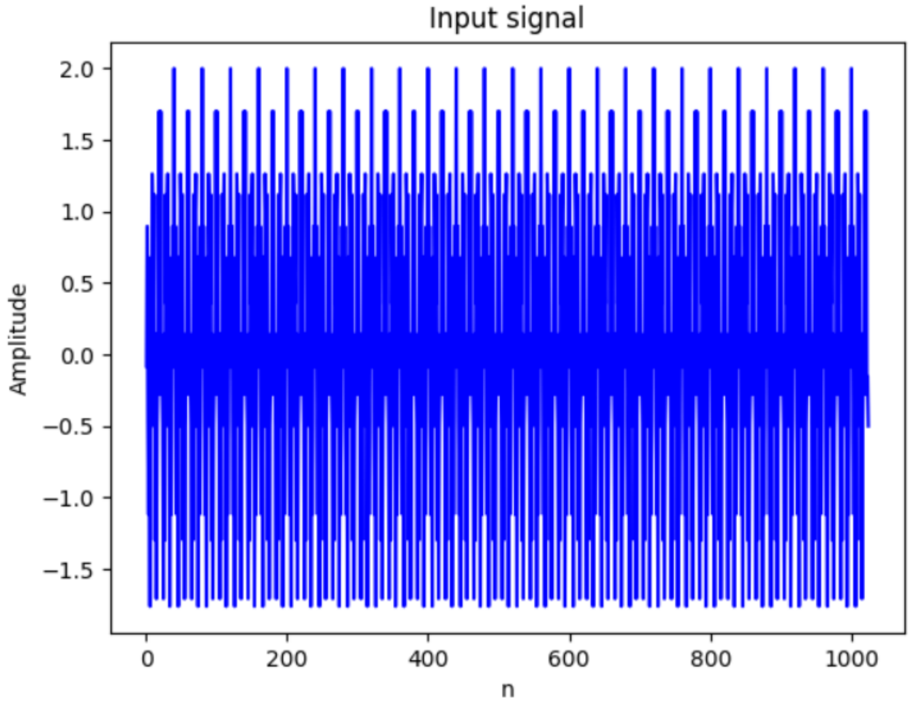
Now we generate the following sequence.

$$x = cos(0.2\pi n) + cos(0.85\pi n)$$

Where n goes from 1 to 2^{10}.
To do so we write the following function.

```
def Q3(plotting=False):
    n = np.arange(1, (2**10)+1)
    x = np.cos(0.2*pi*n) + np.cos(0.85*pi*n)

    if plotting:
        title("Input signal")
        plot(n, x, "r")
        show()

    return x, n
```

Running the above code with plotting=True we get the following as the input signal.

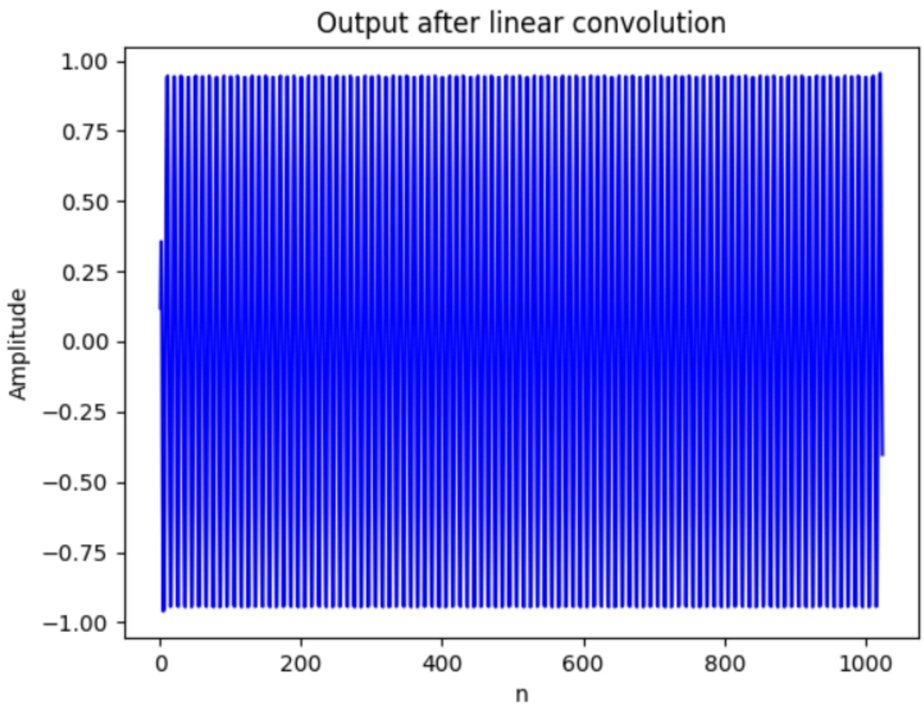


0.3.3 Question 4

Here we do a linear convolution between $x[n]$, generated from the function written in Question 3, and $h[n]$.
To do that we write the following code.

```
def Q4(plotting=False):

    b = Q1()
    x, n = Q3()

    y = np.convolve(x, b, mode="same")

    if plotting:
        title("Output after linear convolution")
        plot(n, real(y), "b")
        show()

    return y
```

Running the above function with $plotting = True$, we have the following plot.



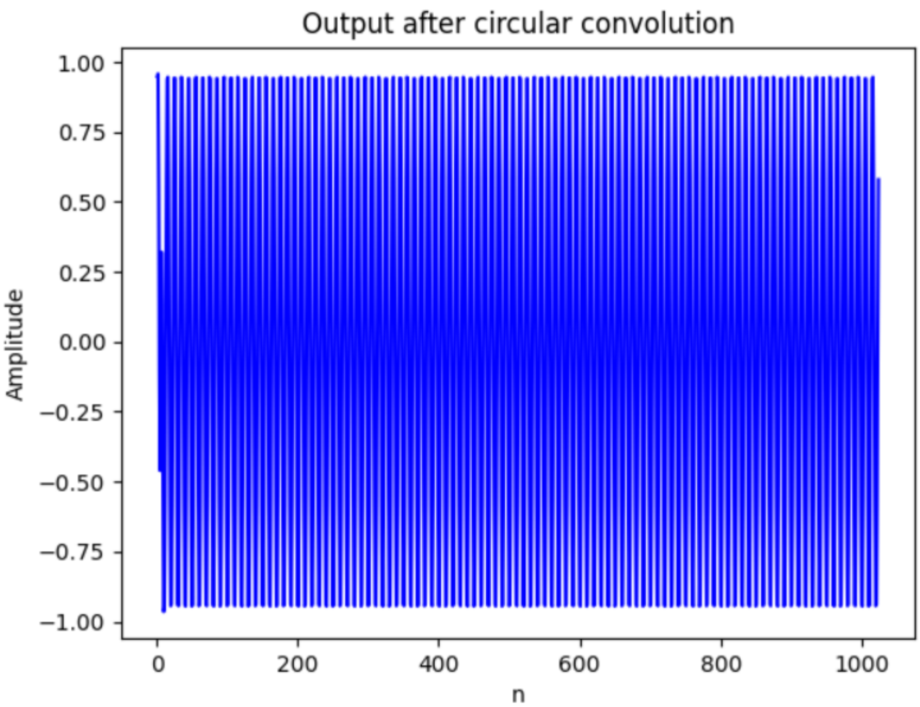
0.3.4 Question 6

Now we find the convolution of $x[n]$ and $h[n]$ using circular convolution.

To find it we write the following code.

```
# circular convolution
def Q5(plotting=False):

    x, n = Q3()
    w, h = Q2()
    b = Q1()

    y = ifft(fft(x)*fft(concatenate((b, zeros(len(x)-len(b))))))

    if plotting:
        title("Output after circular convolution")
        plot(n, real(y), "b")
        show()

    return y
```

Running the above code with $plotting = True$ we get the following plot.



0.3.5 Question 6

Now we find the linear convolution using the circular convolution. The steps to do so are as follows.

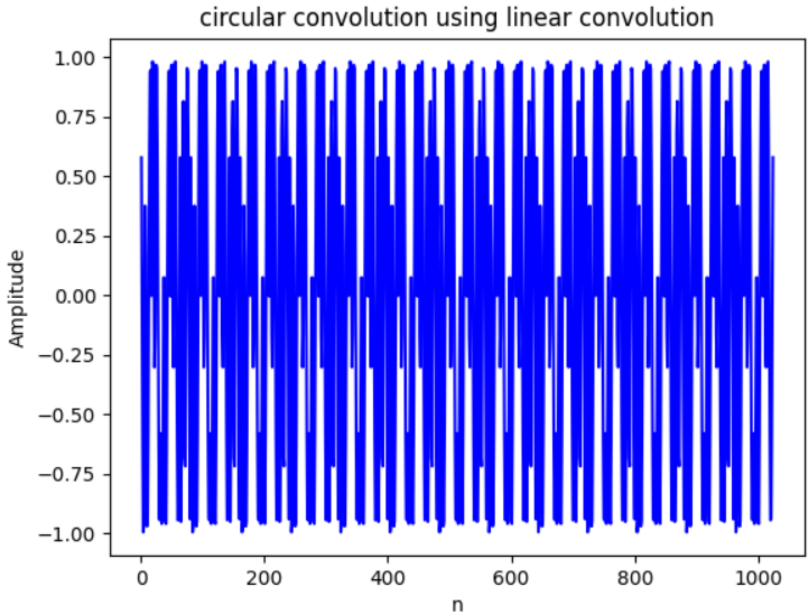
- Suppose $h[]$ fits into a 2^m window. We zero pad the $h[k]$ if required.
- Divide $x[]$ into sections 2^m long.
- Apply circular convolution to each section of $x[]$.
- Stitch the outputs back together.

Following the steps as given in the question we write the following code.

```
# circular convolution using linear convolution
def Q6(plotting=False):

    x, n = Q3()
    b = Q1()

    P = len(b)
    m = int(ceil(log2(P)))

    b_padded = np.concatenate((b, zeros((2**m)-P)))

    len_of_zero_array = (int(ceil(len(x)/2**m)))*(int(2**m))-len(x)
    x_padded = np.concatenate((x, np.zeros(len_of_zero_array)))
```

```
    y = []

    for i in range(int(len(x_padded)/(2**m))):
        x_i = np.concatenate((x_padded[i*(2**m):(i+1)*(2**m)], np.zeros(P-1)))
        x_i = x_padded[i*(2**m):(i+1)*(2**m)]
        y_i = ifft(
            fft(x_i)*fft(b_padded))
        y = np.concatenate((y, y_i))

    if plotting:
        title("some outout")
        plot(n, real(y), "b")
        show()

    return y
```

Running the above code with $plotting = True$ we get the following plot.



0.3.6 Question 7

In this question we will be using the Zadoff-Chu sequence.

We will be plotting the correlation of the Zadoff-Chu sequence.
To do we first need to write the code for reading the file.

```
def file_reading_for_Q7(filename="x1.csv", printing=False):
    f = open(filename)

    b = f.readlines()
    for i in range(len(b)):
```

```python
        b[i] = complex(b[i].strip(" ").strip("i\n")+"j")
    f.close()

    b = np.array(b)

    if printing:
        print("The coefficients are :\n")
        print(b)

    return b
```

Then we write the following code to find and plot the correlation of the Zadoff-Chu sequence.

```python
# Circular correlation
def Q7():
    x = file_reading_for_Q7()

    x2 = np.roll(x, 5)

    cor = np.fft.ifftshift(np.correlate(x2, x, 'full'))
    n = linspace(0, len(cor)-1, len(cor))

    title("Circular correlation")
    plot(n, abs(cor), "b")
    xlim(0, 20)
    show()
```

Running the above code we get the following plot as the correlation.

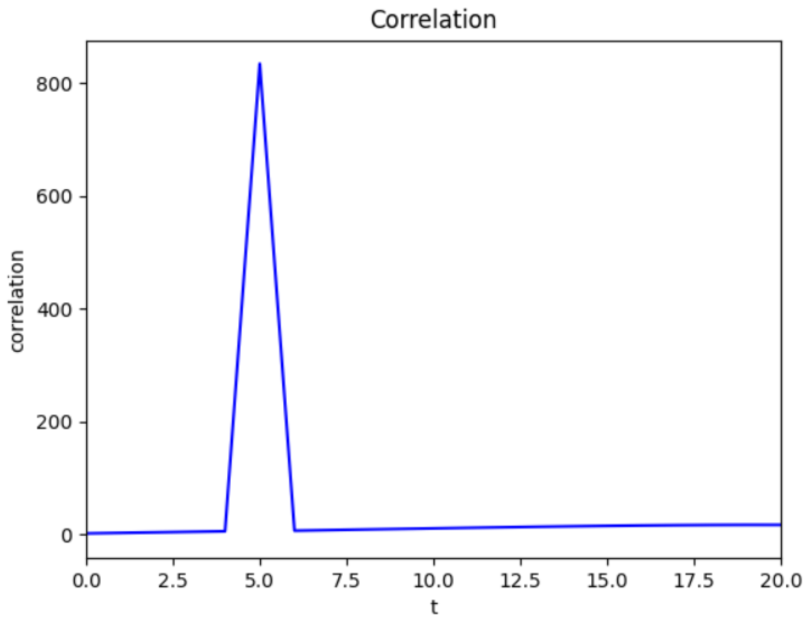


Figure 1: time-frequency plot of the DFT of chirped signal

We see that there is a peak in the correlation at t=5 which is the delay. Thus, we have verified the property of the Zadoff-Chu Sequence.

0.4 Conclusions

In this assignment we have explored different algorithms for convolution. We explored Linear Convolution, Circular convolution and a hybrid between the two. After that we verified the properties of the given Zadoff-Chu Sequence.