Assignment-10: Linear and Circular Convolution

Syam SriBalaji T

EE20B136

May 14, 2022

EE2703 :Jan-May 2022

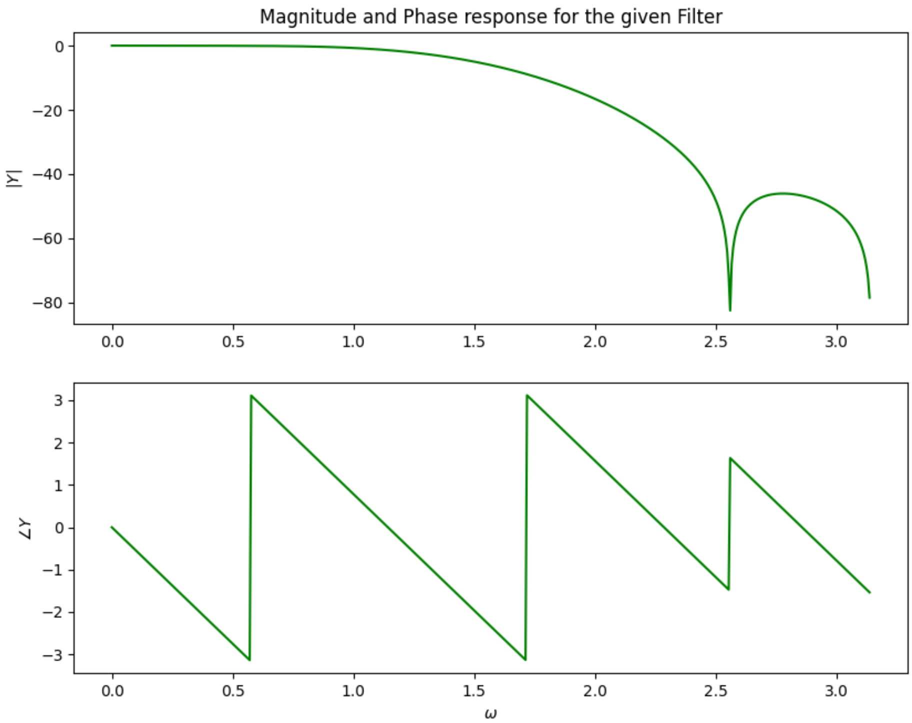
Introduction:

In this assignment, we will be working on

- Linear Convolution
- Circular Convolution
- Circular Convolution using Linear Convolution
- Correlation output of Zadoff-Chu sequences

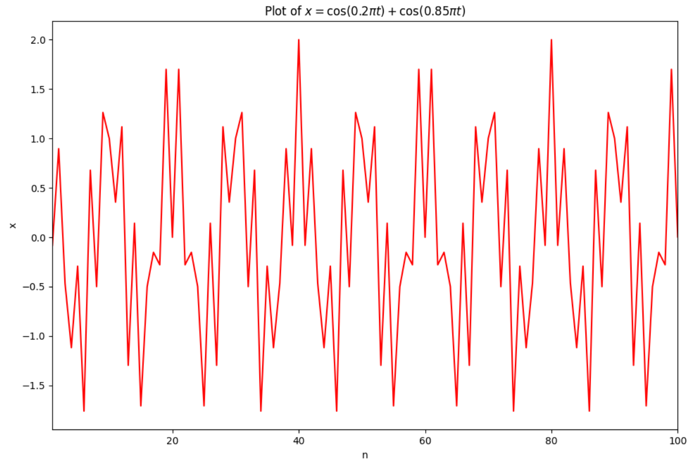
Magnitude and Phase plot of the given filter:

Firstly we read the file *'h.csv'* which contains coefficients for an FIR filter. Now we use *'sig.freqs()'* from scipy library to convert given values from time domain to frequency domain.



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

Here is the normal plot of $cos(0.2\pi n) + cos(0.85\pi n)$, we using *'linspace()'* to create the points in x-axis.

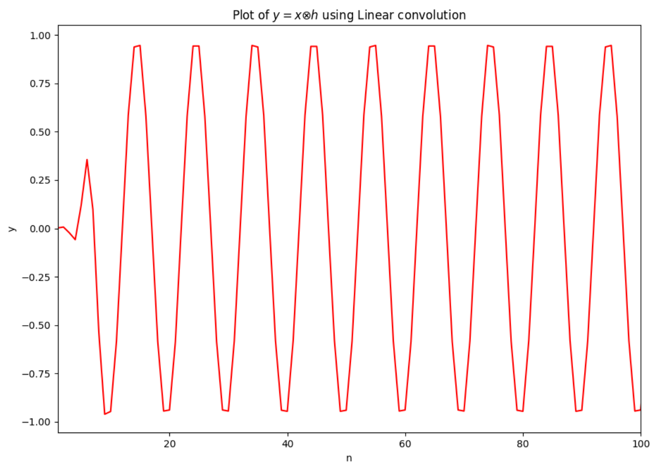


Linear convolution:

Linear convolution for a sequence is-

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

While in Python will be simply using *'convolve()'* from numpy library and solve linear convolution of x which we got from previous question.



Circular convolution:

The DFT supporting convolution is given by-

$$\tilde{x}[n] = \frac{1}{N} \times \sum_{k=0}^{N-1} \tilde{X}[k] e^{j(\frac{2\pi}{N})kn}$$
$$\tilde{X}[k] = \sum_{k=0}^{N-1} \tilde{n}[n] e^{-j(\frac{2\pi}{N})kn}$$

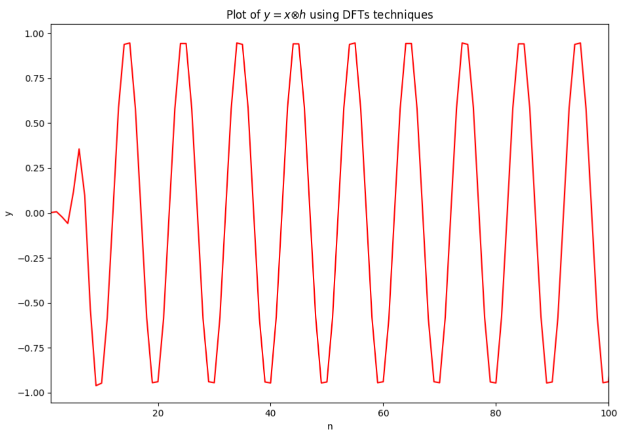
here $\tilde{x}$ refers that the sequence of N values are extended into infinite periodic sequence.

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

Thus, the circular convolution is given by-

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

The length of the output y[n] is expected to be len(x)+len(h)-1. The input x[n] and transfer sequence h[n] are done with zero padding because the number of frequency in DFT and time domain are same. So now in Python we simply use *'concatenate'* from numpy library. to obtain circular convolution.



Linear convolution using Circular convolution:

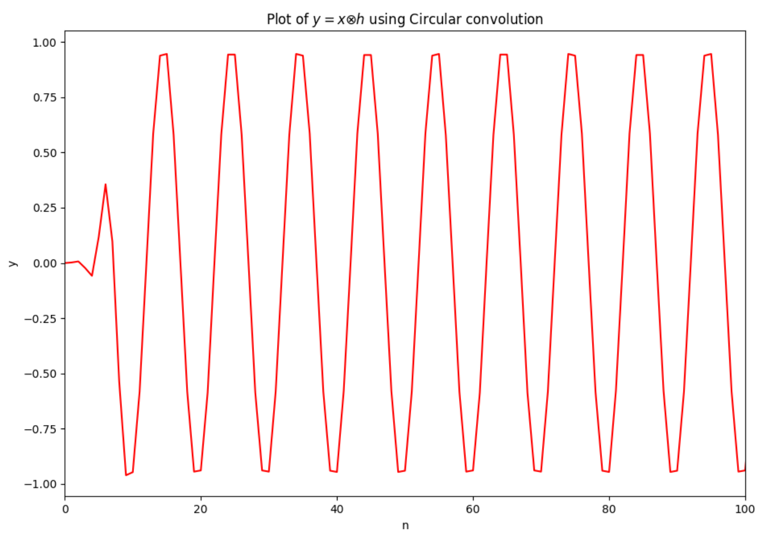
For this method, the following procedure is to be followed-

1. When h[n] fits in 2^m window so it is zero padded
2. x[n] is broken into 2^m divisions
3. Appropriate padding is added to each DFTs
4. Bring back all the outputs and combining it

So we process it by-

```
def Circ_conv(x,h):
    P1 = len(h)
    n_ = int(ceil(log2(P1)))
    h_ = concatenate((h,zeros(int(2**n_)-P1)))
    P2 = len(h_)
    n1 = int(ceil(len(x)/2**n_))
    x_ = concatenate((x,zeros(n1*(int(2**n_))-len(x))))
    y = zeros(len(x_)+len(h_)-1)
    for i in range(n1):
        temp = concatenate((x_[i*P2:(i+1)*P2],zeros(P2-1)))
        y[i*P2:(i+1)*P2+P2-1] += np.fft.ifft(np.fft.fft(temp) * np.fft.fft( concatenate
        ( (h_,zeros(len(temp)-len(h_))) ))).real
    return y
```

From that, we plot the Circular convolution using Linear convolution of x. And also we can note that this is similar to the graph which we got earlier.

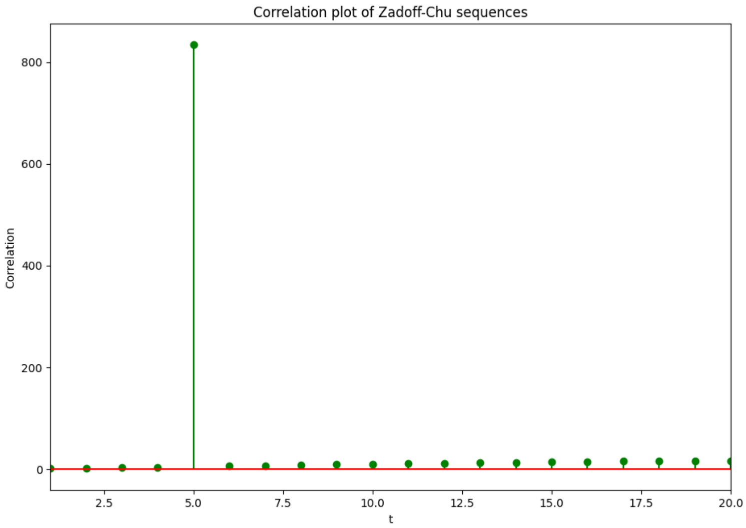


Correlation output plot of Zadoff-Chu sequences:

The properties of Zadoff-Chu sequence are-

1. It is a complex and constant amplitude sequence.
2. Auto correlation of it's sequence with a cyclically shifted version of itself is zero.
3. Correlation of it's sequence with the delayed version of itself will give a peak at that delay.

Thus, after Auto-correlating Zandoff Chu sequence, we get-



And we can view that the peak is present at 5.

Conclusion:

Thus, in this assignment we understood that we can do Linear convolution, Circular convolution, Correlation for various different kinds of input signals in Python easily.

Thank you!