Assignment No 10

Sumanth R Hegde, EE17B032

April 10, 2019

1 Aim

In this week's assignment, we explore linear and circular convolutions and their implementation using the DFT.

2 FIR filter

The discrete time signal is extracted from the csv file using the `genfromtxt` function. Using scipy.signal.freqz , we obtain the DFT of the FIR filter, along with the corresponding frequencies. The magnitude plot (in dB scale) and phase plot are as follows : From the magnitude plot, we notice that

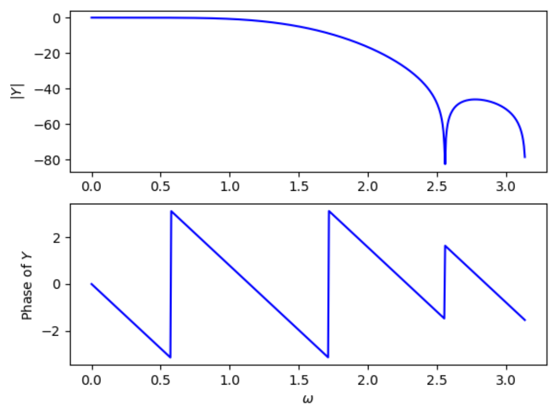

Figure 1: Magnitude and Phase response of the filter

the system acts a low pass filter. The phase plot shows that the system introduces a constant group delay.

3 Response for a mixed frequency input

The following code snippet generates x = $cos(0.2\pi n) + cos(0.85\pi n)$ for $n = 1, 2, 3, ...2^{10}$

```
n = linspace(1,2**10,2**10)
x = cos(0.2*pi*n) + cos(0.85*pi*n)
```

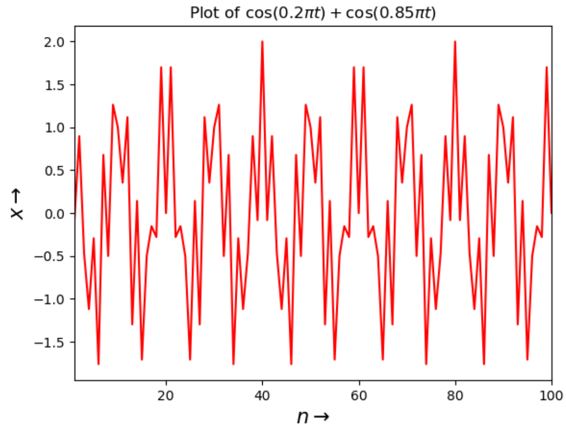

Figure 2: The mixed frequency signal

The output $y = x * h$ is obtained using the np.convolve function. The output is as follows :

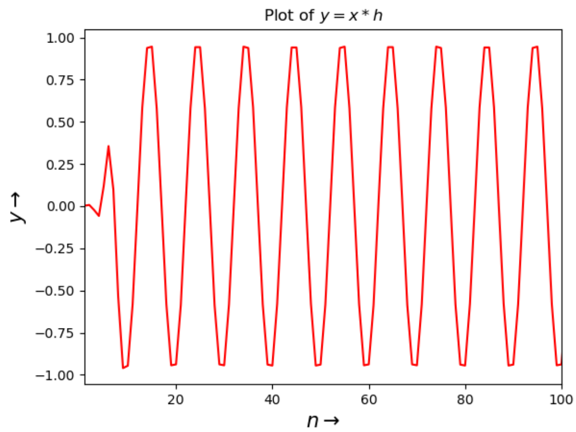

Figure 3: Output obtained using linear convolution

Notice that the high frequency variations in $x[n]$ have been attenuated by the system. The low frequency component remains.

3.1 Using DFT to perform Linear Convolution

The length of the output $y[n]$ is expected to be `len(x)+len(h)-1`. We also know that the number of frequencies in the DFT of a signal is equal to the number of samples in the time domain. Thus, the input $x[n]$ and the filter $h[n]$ are zero padded, so that the DFT of the $y[n]$ has the appropriate number of samples. The following code snippet does the job :

```
x_ = np.concatenate((x,zeros(len(h)-1)))
y1= np.fft.ifft(np.fft.fft(x_) *
np.fft.fft( np.concatenate( (h,np.zeros(len(x_)-len(h))) )))
```

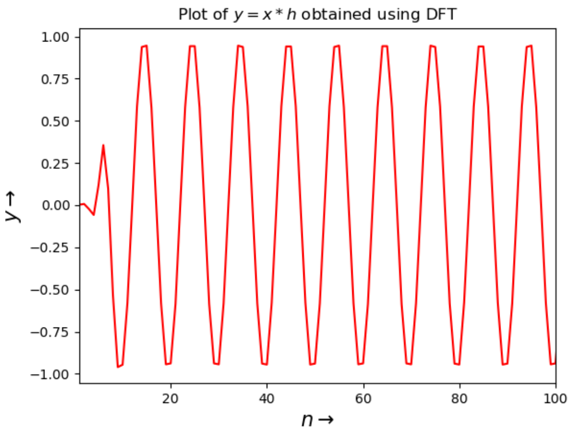

Figure 4: $y[n]$ obtained using the DFTs of $x[n]$ and $h[n]$

We see that the signals obtained using `np.convolve` and the DFT are identical.

3.2 Using Circular Convolution to perform Linear Convolution

The main steps followed in the implementation are :

- We zero pad the $h[n]$ to fit a 2^m window
- $x[n]$ is broken into sections 2^m long
- Now, each section of $x[n]$ is convolved with $h[n]$ using the corresponding DFTs. Appropriate padding is aded to ensure that the length of the output is as expected.
- Each succeeding convolution adds to the already computed value of $y[n]$ at each index n, which may be updated by the previous convolution

The following code snippet implements the above algorithm:

```
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( np.concatenate( (h_,np.zeros(len(temp)-len(h_))) ))).real
    return y
```

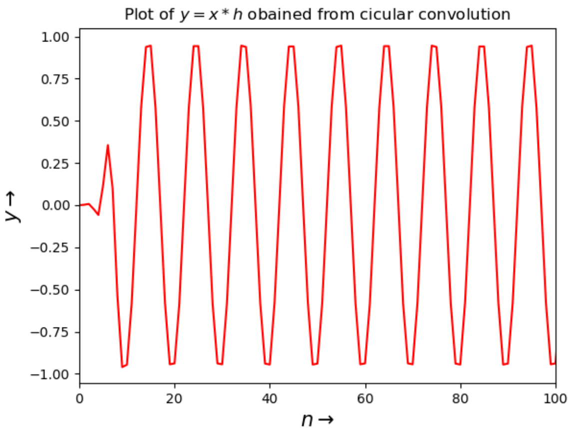

Figure 5: Output obtained using linear convolution

The output obtained is identical to the ones obtained using the previous methods. In fact, the error is of the order of 10^{-16}, possibly due to the finite accuracy in calculating the DFT.

4 Circular Correlation

Consider the Zadoff-Chu sequence, a commonly used sequence in communication. The properties of the sequence are :

- It is a complex sequence.
- It is a constant amplitude sequence.

- The auto correlation of a ZadoffChu sequence with a cyclically shifted version of itself is zero, except at the shift
- Correlation of ZadoffChu sequence with the delayed version of itself will give a peak at that delay.

The auto correlation property is verified using the following code snippet :

```
# x is the Zadoff-Chu sequence
X = np.fft.fft(x)
x2 = np.roll(x,5)
cor = np.fft.ifftshift(np.correlate(x2,x,'full'))
```

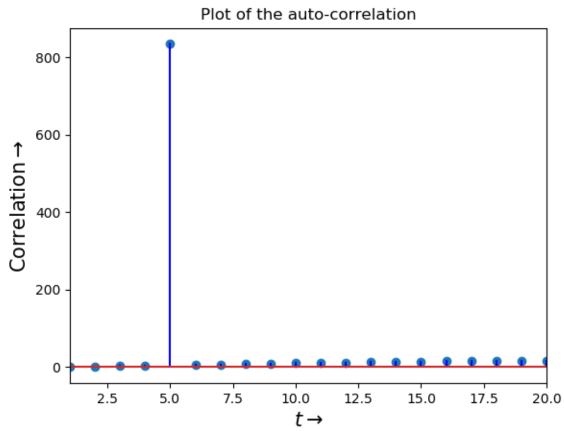

Figure 6: Auto-correlation of the sequence with a cyclically shifted version of itself

Auto-correlation of $x[n]$ and a cyclic shifted version of $x[n]$, with a shift of 5 to the right gives a peak at 5, and vanishes everywhere else.

5 Conclusion

Linear convolution algorithm implemented using a direct summation is non-optimal and computationally expensive. A faster way to perform the convolution is to use the DFTs of the input and the filter. Circular convolution can be used for the implementation of linear convolution, with a much faster computation speed. The magnitude and phase response of a low pass filter were studied and the system's output, for a mixed frequency signal was obtained through three different methods. For the Zadoff-Chu sequence, the auto correlation output with a cyclically shifted version of itself was found to be non zero only at the point corresponding to the shift.