Applied Programming Lab:Assignment 10

Saurabh Vaishampayan:EP17B028

May 15 2020

1 Introduction

This assignment involves Discrete Time signal processing. We import FIR filter coefficients of a low pass filter, implement linear convolution, circular convolution, implement linear convolution using circular convolution by the overlap add algorithm and compare the results. We also study the Zadoff Chu sequence for its autocorrelation proerties

2 Properties of the FIR LPF

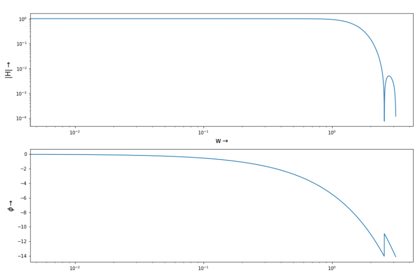
We import the coefficients from the csv file, study its magnitude and phase response, find the 3dB frequency, find zeros.

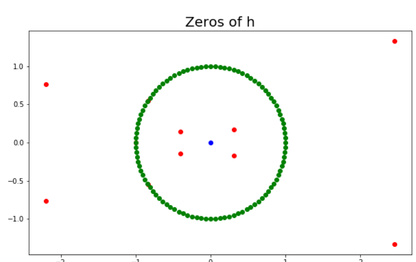
2.1 Codes

```

import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as sp
import csv

def extract(filename):
    h = []
    with open(filename,'r') as f:
        data  = csv.reader(f)
        for row in data:
            temp = row[0].split('i')
            if(len(temp)==1):
                h.append(float(temp[0]))
            if(len(temp)==2):
                temp = temp[0].split('+')
                if(len(temp)==2):
                    h.append(float(temp[0])+1j*float(temp[1].split('i')[0])
    )
                else:
                    temp = temp[0].split('-')
                    if(len(temp)==2):
                        h.append(float(temp[0])-1j*float(temp[1].split('i')
    [0]))
                    else:
                        h.append(float(temp[1])-1j*float(temp[2].split('i')
    [0]))
```

```python
    h = np.array(h)
    return h

h = extract('h.csv')
w, htransf = sp.freqz(h)
ii = np.where(np.abs(np.abs(htransf)-0.5)==np.min(np.abs(np.abs(htransf)
    -0.5)))
print(w[ii]) #3db Freq

zeros = np.roots(h)
x = np.real(zeros)
y = np.imag(zeros)
print(zeros)

theta = np.linspace(0,2*np.pi,101)[:-1]

X = np.cos(theta)
Y = np.sin(theta)

plt.figure(figsize=(10,6))
plt.plot(x,y,'ro')
plt.plot(X,Y,'go')
plt.plot(0,0,'bo')
plt.title('Zeros of h',size=20)
plt.savefig('zeros.png')
plt.show()
```

2.2 Plots





2.3 Inference

3dB Frequency = 1.5646604
We infer that h is a low pass filter with 3dB frequency of 1.5646604 rad/s. We also plot the zeros of h and find that it has 2 zeros on the unit circle: -0.83410025+0.55161288j and -0.83410025-0.55161288j

3 Filtering of input sum of sinusoids

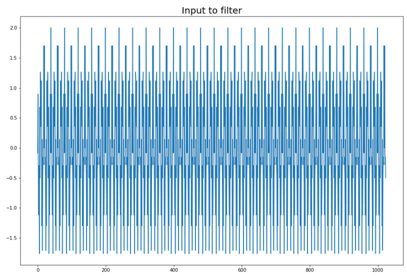
3.1 Description

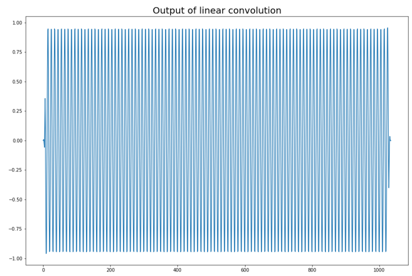
We implement filtering by three methods: Linear Convolution, Circular convolution, linear convolution using circular convolution by overlap add method and compare the results

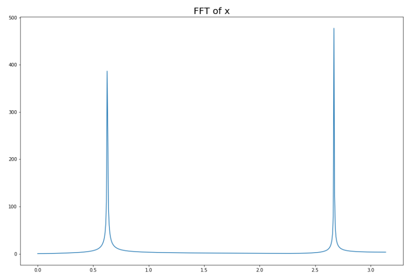
3.2 Codes

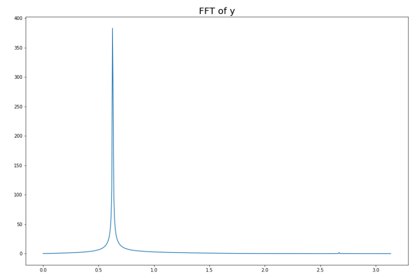
```

n = np.arange(1, (2**10)+1, 1)
x = np.cos(0.2*np.pi*n)+np.cos(0.85*np.pi*n)
plt.figure(figsize=(15,10))
plt.plot(x)
plt.title('Input to filter', size=20)
plt.savefig('x.png')
plt.show()

y = np.convolve(x, h)
plt.figure(figsize=(15,10))
plt.plot(y)
plt.title('Output of linear convolution',size=20)
plt.savefig('linout.png')
plt.show()

w, X = sp.freqz(x)
plt.figure(figsize=(15,10))
plt.plot(w, np.abs(X))
plt.title('FFT of x',size=20)
plt.show()
w, Y = sp.freqz(y)
plt.figure(figsize=(15,10))
plt.plot(w, np.abs(Y))
plt.title('FFT of y',size=20)
plt.show()

y1 = np.fft.ifft(np.fft.fft(x)*np.fft.fft(np.concatenate((h,np.zeros(len(x)
    -len(h))))))
plt.figure(figsize=(15,10))
plt.plot(y1)
plt.title('Output of circular convoltion', size=20)
plt.savefig('circout.png')
plt.show()

y2 = np.concatenate((y1, np.zeros(len(y)-len(y1))))
plt.figure(figsize=(15,10))
plt.plot(y2-y)
plt.title('Difference in output by the two methods', size=15)
plt.savefig('diff.png')
plt.show()
```

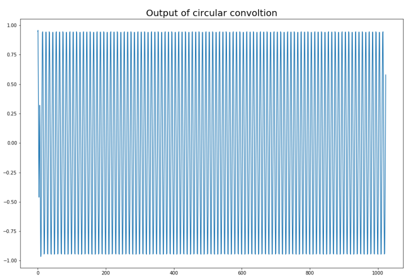
3.3 Plots

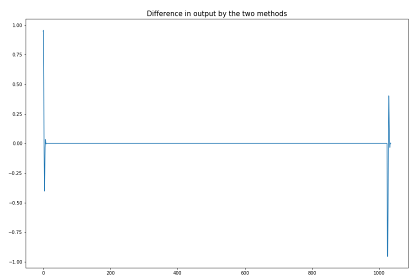












3.4 Inference

We observe that the filter filters out the higher frequency sinusoid present in the input. Also the linear convolution and the circular convolution outputs exactly match in the middle but are vastly different at the ends, as evident by the plot for their difference.

3.5 Linear Convolution using circular convolution

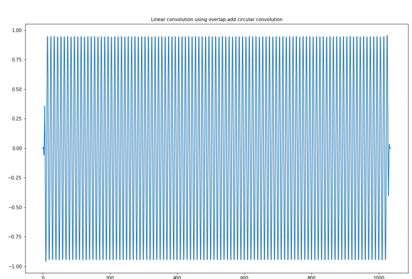
Overlap add algorithm is used here.

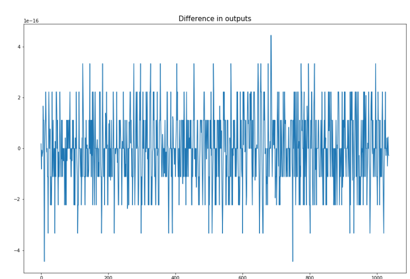
3.6 Code

```
#Linear convolution using circular convolution:

```

```python
def overlap_add_conv(x,h):

    #Zero pad h to have length as power of 2
    M = len(h)
    n_ = int(np.ceil(np.log2(M)))
    N = 2**n_
    L = N+1-M
    h_ = np.concatenate((h,np.zeros(L-1)))

    #Make slices of x which are L units long
    Nx = len(x)
    n_slices = int(np.ceil(Nx/L))
    x_ = np.concatenate((x,np.zeros(L*n_slices-Nx)))

    y = np.zeros(len(x_)+M-1)

    for i in range(n_slices):
        temp = np.concatenate((x_[i*L:(i+1)*L],np.zeros(M-1)))
        y[i*L:(i+1)*L+M-1] += np.fft.ifft(np.fft.fft(h_)*np.fft.fft(temp)).real
    return y

y3 = overlap_add_conv(x,h)
y = np.concatenate((y, np.zeros(len(y3)-len(y))))
plt.figure(figsize=(15,10))
plt.plot(y3)
plt.title('Linear convolution using overlap-add circular convolution', size=10)
plt.savefig('overlapadd.png')
plt.show()
plt.figure(figsize=(15,10))
plt.plot(y3-y)
plt.title('Difference in outputs',size=15)
plt.savefig('diff2.png')
plt.show()
```

3.7 Plots





3.8 Inference

By the plot for difference of linear convolution and overlap add convolution, we infer that they exactly match upto floating point precision of 1e-16. Overlap add method is much faster than linear convolution when the FIR filter is small compared to the input.

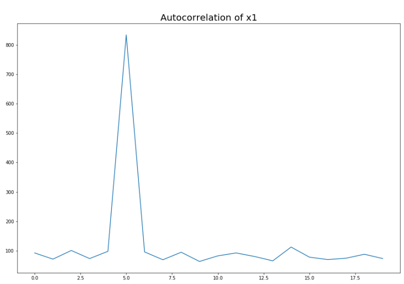
4 Zadoff Chu

We plot the correlation of a cyclically shifted Zadoff Chu sequence with itself, and analyse it

4.1 Code

```
#Question 6:
zc = extract('x1.csv')
z2 = np.roll(zc,5)

corr = np.fft.ifftshift(np.correlate(z2,zc,'full'))
plt.figure(figsize=(15,10))
plt.plot(np.abs(corr[0:20]))
plt.title('Autocorrelation of x1',size=20)
plt.savefig('x1.png')
plt.show()
```

4.2 Plots



4.3 Inference

We observe that the autocorrelation vanishes for all n except a peak at n=5, which was the shift.

5 Conclusions

- We have imported an FIR filter, studied its properties, used it for filtering the input by different methods.
- Overlap add method is a fast way of implementing DSP filters when the FIR filter is small in size compared to input. It is also practical for real time processing.
- We studied the Zadoff Chu sequence for its autocorrelation properties