EE2703 : Applied Programming Lab
Assignment 8

Sasanapuri Uttam Raj
EE20B118

April 2022

1 Introduction

- We analyse and use DFT to find the Fourier transform of periodic signals and non periodic ones using fast fourier transform algorithms which are implemented in python using ff t and ff tshif t which is used to center the fourier spectra of a discrete signal.
- The discrete Fourier transform (DFT) converts a finite sequence of equally-spaced samples of a function into a same-length sequence of equally-spaced samples of the discrete-time Fourier transform (DTFT), which is a complex-valued function of frequency.
- Let suppose f[n] are the samples of some continous function f(t) then we define the Z transform as

$$F(z) = \sum_{n=-\infty}^{\infty} f(n) z^{-n}$$

- Replacing z with $e^{j\omega}$ we get DTFT of the sampled function

$$F(e^{j\omega}) = \sum_{n=-\infty}^{\infty} f(n) e^{-j\omega n}$$

- F($e^{j\omega}$) is continuous and periodic. f[n] is discrete and aperiodic. Suppose now f[n] is itself periodic with a period N, i.e.,

$$f[n+N] = f[n]$$

- Then, it should have samples for its DTFT. This is true, and leads to the Discrete Fourier Transform or the DFT.
- Suppose f[n] is a periodic sequence of samples, with a period N. Then the DTFT of the sequence is also a periodic sequence F[k] with the same period N.

$$F[k] = \sum_{n=0}^{N-1} f[n] e^{\frac{2\pi nk}{N}} = \sum_{n=0}^{N-1} f[n] W^{nk}$$

$$f[n] = \frac{1}{N} \sum_{k=0}^{N-1} F[k] W^{nk}$$

- Here W = $e^{-j\frac{2\pi}{n}}$ is used simply to make the equations less cluttered and k is sampled values of continuous variable ω at multiples of 2π.
- What this means is that the DFT is a sampled version of the DTFT, which is the digital version of the analog Fourier Transform .In this assignment, we want to explore how to obtain the DFT, and how to recover the analog Fourier Tranform for some known functions by the proper sampling of the function

2 Python code

2.1 Question 1

```
#-----------------------------------Main Program Code-------------------------------------

#-----------------------------------example1:sin(5x)--------------------------------------
x1=np.linspace(0,2*np.pi,128)
y1=np.sin(5*x1)
Y1=np.fft.fft(y1)

magphaplot(x1,y1,Y1,r"Spectrum of $\sin(5t)$")

#-----------------------------------example2:sin(5x)--------------------------------------

x2=np.linspace(0,2*np.pi,129);x2=x2[:-1]
y2=np.sin(5*x2)
Y2=np.fft.fftshift(np.fft.fft(y2))/128.0
w2=np.linspace(-64,63,128)
xlim2 = [-10,10]

magphaplot1(Y2,w2,r"Spectrum of $\sin(5t)$",xlim2)

#-----------------------------------example3:(1+0.1cost)*cos(10t)-------------------------

t=np.linspace(0,2*np.pi,129);t=t[:-1]
y3=(1+0.1*np.cos(t))*np.cos(10*t)
Y3=np.fft.fftshift(np.fft.fft(y3))/128.0
```

```
w3=np.linspace(-64,63,128)

xlim3 = [-15,15]
magphaplot1(Y3,w3,r"Spectrum of $\left(1+0.1\cos\left(t\right)\right)\cos\left(10t\right)$"

#-----------------------------------example4:(1+0.1cost)*cos(10t)-------------------------

t1=np.linspace(-4*np.pi,4*np.pi,513);t1=t1[:-1]

y4=(1+0.1*np.cos(t1))*np.cos(10*t1)
Y4=np.fft.fftshift(np.fft.fft(y4))/512.0

w4=np.linspace(-64,64,513);w4=w4[:-1]

magphaplot1(Y4,w4,r"Corrected Spectrum of $\left(1+0.1\cos\left(t\right)\right)\cos\left(10t
```

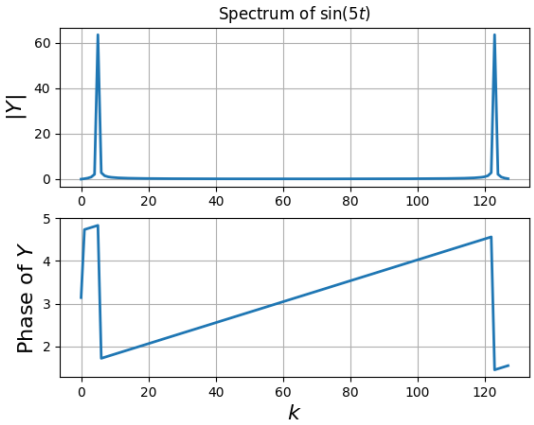

Figure 1: Incorrect Fourier spectrum plots of sin(5t)

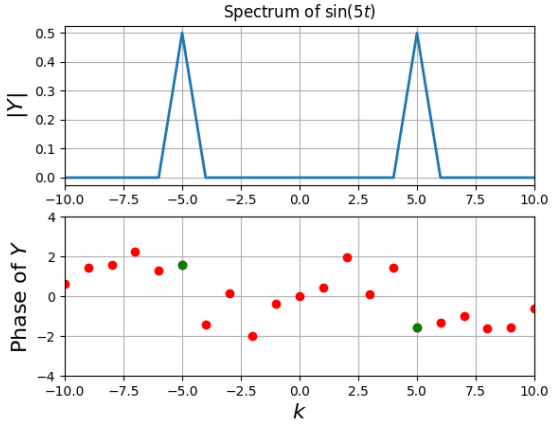

Figure 2: Correct Fourier spectrum plots of sin(5t)

2.1.1 Results and Discussion :

- The initial plot is not wrong, but is poorly labelled in x and y axes, so can be generally assumed to be incorrect. The correct plot is got from using the ff tshif t() function, which shifts the fft() function output between a window of interest.
- As we observe the plot frequency contents are of $\omega = 5rads^{-1}$, -$5rads^{-1}$
- Since everything consists of sin terms so phase is zero and π alternatively. For amplitude of the spectra we analyse the fourier transform of sin(5t) which is derived above.
- The red points are phase of points with Magnitude response less than 10^{-3}

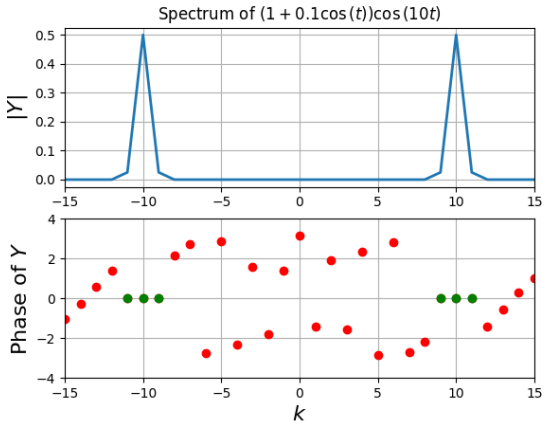

Figure 3: Incorrect Fourier spectrum plots of (1 + 0.1 cos (t)) cos (10t)

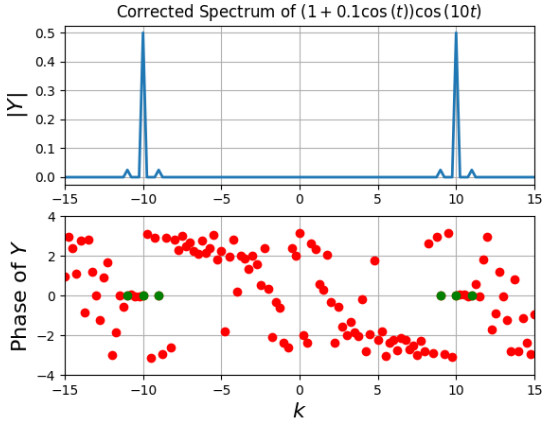

Figure 4: Correct Fourier spectrum plots of (1 + 0.1 cos (t)) cos (10t)

2.1.2 Results and Discussion :

- The first plot is incorrect because we did not take enough window size for the plots to be perfectly shown, thus the smaller frequency of 2π gets hidden. To bring them out we must increase number of points in time domain, for the smaller frequencies to show up, whose output is shown in second plot.

- As we observe the plot it has center frequencies of $\omega = 10$, -10 from carrier signal and as expected we get side band frequencies at $\omega = \pm 9, \pm 11$ .Since amplitude of the message signal is changed by a carrier signal cos(10t). It is called as amplitude modulation. And the amplitude of the side band frequencies are obtained from fourier transform expression.
- Phase spectra is 0 since only cos terms are present.
- The red points are phase of points with Magnitude response less than 10^{-3}

2.2 Question 2:

- To find Discrete Fourier Transform DFT of $sin^3(t)$ and $cos^3(t)$.
- Plot and analyse the spectrum obtained for both the functions given above.
- Cross validate the spectrum obtained with what is expected.
- To compare the spectrum obtained for $sin^3(t)$, we use

$$sin^3(t) = \frac{3}{4}sin(t) - \frac{1}{4}sin(3t)$$

- So the fourier transform of $sin^3(t)$ using above relation is

$$(sin^3(t)) \rightarrow \frac{3}{8j}(\delta(\omega - 1) - \delta(\omega + 1)) - \frac{1}{8j}(\delta(\omega - 3) - \delta(\omega + 3))$$

- Similarly $cos^3(t)$ is given by

$$cos^3(t) = \frac{3}{4}cos(t) + \frac{1}{4}cos(3t)$$

- So the fourier transform of $cos^3(t)$ using above relation is

$$(cos^3(t)) \rightarrow \frac{3}{8j}(\delta(\omega - 1) + \delta(\omega + 1)) + \frac{1}{8j}(\delta(\omega - 3) + \delta(\omega + 3))$$

- So using this we compare the plots of Magnitude and phase spectrum obtained using DFT and analyse them.

Code:

```
#-----------------------------------Question2:(cost)^3-----------------------------------


y5=(np.cos(t1))**3
Y5=np.fft.fftshift(np.fft.fft(y5))/512.0
w5=np.linspace(-64,64,513);w5=w5[:-1]

magphaplot1(Y5,w4,r"Spectrum of $cos^3\left(t\right)$",xlim3,False)

#-----------------------------------Question2:(sint)^3-----------------------------------


y6=(np.sin(t1))**3
Y6=np.fft.fftshift(np.fft.fft(y6))/512.0

magphaplot1(Y6,w4,r"Spectrum of $sin^3\left(t\right)$",xlim3,False)
```

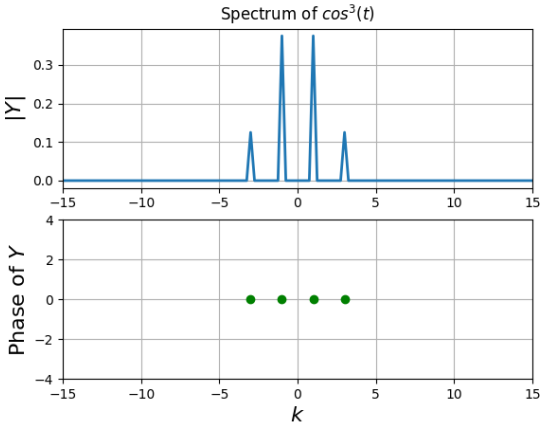

Figure 5: Spectrum of $sin^3(t)$

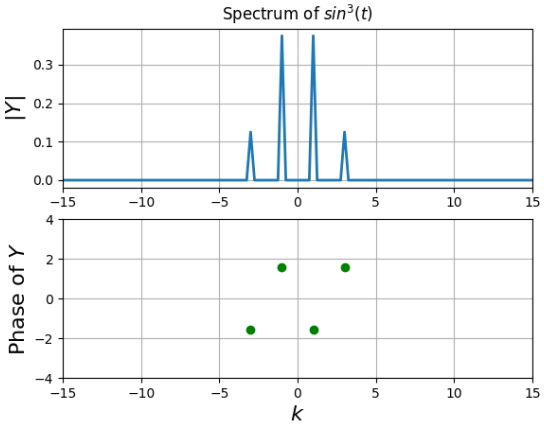

Figure 6: Spectrum of $cos^3(t)$

2.2.1 Results and Desicussion :

- As we observe the plot frequency contents are of ω = 1, -1, 3, -3 and with their amplitude in 1:3 ratio
- For $sin^3(t)$, since everything consists of cos terms so phase is zero. But due to lack of infinite computing power they are nearly zero in the order of

- For $cos^3(t)$, since everything consists of sin terms so phase is zero and π alternatively.

2.3 Question 3:

- To generate the spectrum of cos(20t + 5 cos(t)).
- Plot phase points only where the magnitude is significant.
- Analyse the spectrums obtained.

Code:

```
#-----------------------------------Question3:cos(20t)+5cost-------------------------------

y7=np.cos((20*t1)+(5*np.cos(t1)))
Y7=np.fft.fftshift(np.fft.fft(y7))/512.0

xlim4 = [-35,35]

magphaplot1(Y7,w4,r"Spectrum of $cos(20t + 5cos(10t))$",xlim4,False)
```

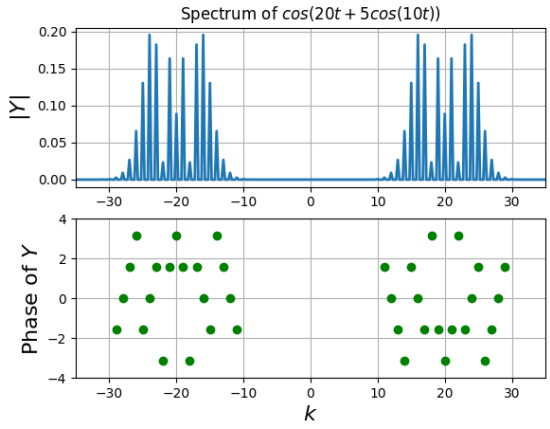

Figure 7: Spectrum of cos(20t + 5 cos(t)

2.3.1 Results and Discussion :

- As we observe the plot that its a Phase modulation since phase of the signal is varying proportional to amplitude of the message signal being $\omega = 20$ and infinite side band frequencies which are produced by 5 cost. since cos(t) is infinitely long signal. But the strength of the side band frequencies decays or very small which are away from center frequency or carrier frequency component as we observe from the plot.

- Phase spectra is a mix of different phases from [-π, π] because of phase modulation, i.e since the phase is changed continuously wrt time, the signal can represent either a sine or cosine depending on the phase contribution from cos(t).

2.4 Question 4:

- To generate the spectrum of the Gaussian $e^{-\frac{t^2}{2}}$ band limited in frequency and aperiodic in time domain find Fourier transform of it using DFT and to recover the analog fourier transform from it.

$$e^{-}\frac{t^2}{2} \rightarrow \frac{1}{\sqrt{2\pi}} e^{-}\frac{\omega^2}{2}$$

- To find the normalising constant for DFT obtained we use following steps to derive it :

- window the signal $e^{-\frac{t^2}{2}}$ by rectangular function with gain 1 and window size 'T' which is equivalent to convolving with T sinc(ωT) in frequency domain. So as T is very large the sinc(ωT) shrinks, we can approximate that as $\delta(\omega)$. So convolving with that we get same thing.

- Windowing done because finite computing power and so we cant represent infinetly wide signal.

- Now we sample the signal with sampling rate N,which is equivalent to convolving impulse train in frequency domain

- And finally for DFT we create periodic copies of the windowed sampled signal and make it periodic and then take one period of its Fourier transform i.e is DFT of gaussian.

- Following these steps we get normalising factor of Window size/(2π Sampling rate)

$$e^{-}\frac{t^2}{2} \leftrightarrow \frac{1}{\sqrt{2\pi}} e^{-}\frac{\omega^2}{2}$$

$$rect(\frac{t}{T}) = 1 for |t| < T$$

$$rect(\frac{t}{T}) = 0 otherwise$$

- For windowing the signal, we will multiply with the rectangular function,

$$y(t) = gaussian(t) * rect(\frac{t}{T}$$

- Then we will perform convolution in fourier domain. That is it is multiplication in fourier.
- Now sampling the outcome with a period of $2\pi/T_s$ and solving further we will get the multiplication factor to be

$$const = \frac{T}{T_S 2\pi}$$

- To find the Discrete Fourier transform equivalent for Continous Fourier transform of Gaussian function by finding absolute error between the DFT obtained using the normalising factor obtained with exact Fourier transform and find the parameters such as Window size and sampling rate by minimising the error obtained with tolerance of 10^{-5}
- To generate the spectrum.
- Now we use this directly and plot the expected output along with estimated output using ff tshif t. It is shown below along with code
- Analyse the spectrums obtained.

Code:

```
#-----------------------------------Question4:e^(t^2)/2)----------------------------------

# To plot FFT of gaussian signal
T = 8*np.pi  #Time period
N = 128   # Samples
Yold=0
tolerance=1e-6   #Accuracy
err=tolerance+1
iters = 0
#iterative loop to find window size
while err>tolerance:
    x = np.linspace(-T/2,T/2,N+1)[:-1]
    w = np.linspace(-N*np.pi/T,N*np.pi/T,N+1)[:-1]
    y = np.exp(-0.5*x**2)
```

```
    Y = np.fft.ifftshift(np.fft.fft(np.fft.fftshift(y)))*T/(2*np.pi*N)
    err = sum(abs(Y[::2]-Yold))
    Yold = Y
    print(err)
    iters+=1
    T*=2
    N*=2

#The DTF of gaussian
Y_exp = 1/np.sqrt(2*np.pi)np.exp(-w*2/2)

#calculating error
true_error = sum(abs(Y-Y_exp))
print("True error: ",true_error)
print("samples = "+str(N)+" time period = "+str(T/np.pi)+"*pi")

#Expected output
xlim6 = [-15,15]
magphaplot1(Y_exp,w,r"Expected Spectrum of $e^{\frac{t^{2}}{2}}$",xlim6)


#Estimated output
magphaplot1(Y,w,r"Estimated Spectrum of $e^{\frac{t^{2}}{2}}$",xlim6)
```

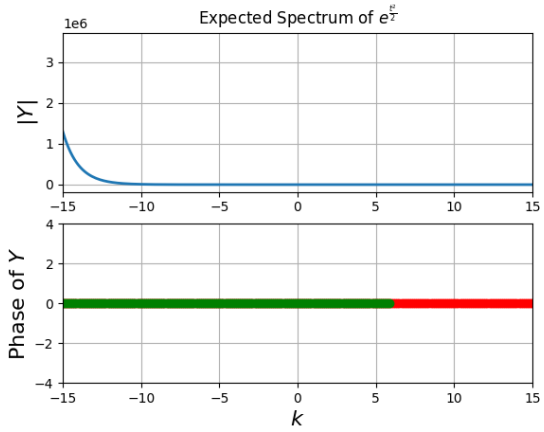

Figure 8: Expected Spectrum of $e^{\frac{t^2}{2}}$

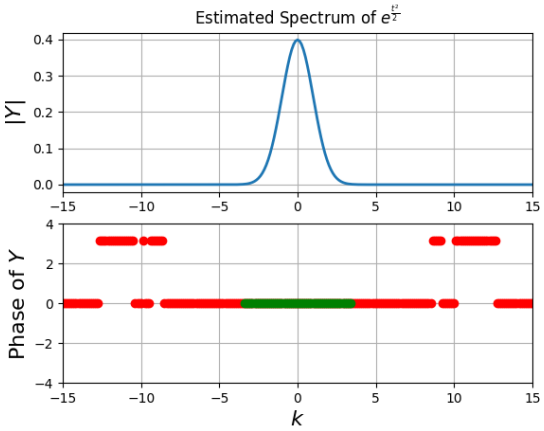

Figure 9: Expected Spectrum of $e^{\frac{t^2}{2}}$

2.4.1 Results and discussion :

- True error: 1.4532298948786486e-14 samples = 512 time period = 32.0*pi
- From the outputs, we can conclude that the estimated output is same as the expected output. The green circles in phase denote the phase for magnitudes greater than 10^{-3} which is clearly shown in the plot itself.

2.5 Conclusion

- From this assignment, we understood how to use FFT on various signals from the command fft, ifft and fftshift from pylab, which is the only library used for the entire assignment. We understood the importance of scaling of magnitude, and shift axis for the exact frequency axis. We also learnt how sampling frequency and time period is important for a correct result. We can now use the same methods to use FFT for many other input signals.
- We used fast Fourier transform method to compute DFT as it improves the computation time from O(n^2) $\rightarrow$ O(nlog2(n))
- FFT works well for signals with samples in 2^k, as it divides the samples into even and odd and goes dividing further to compute the DFT.
- That's why we use no of samples in the problems above taken as powers of 2.