Indian Institute of Technology, Madras
Department of Electrical Engineering
Applied Programming Lab

Lab Report
Assignment 9

Nithin Uppalapati
EE18B035

26-06-2020

Contents

1. Abstract

In the previous experiment we dealt with periodic and symmetric signals for obtaining DFTs. In the real world we do not have periodic signals because we can capture only a finite amount of data. So in reality we have to deal with non periodic and asymmetric signals.

2. Introduction

In this assignment we are going to compute DFTs of periodic but asymmetric signals. We are also going to implement windowing of signals to get better signal spectrum.

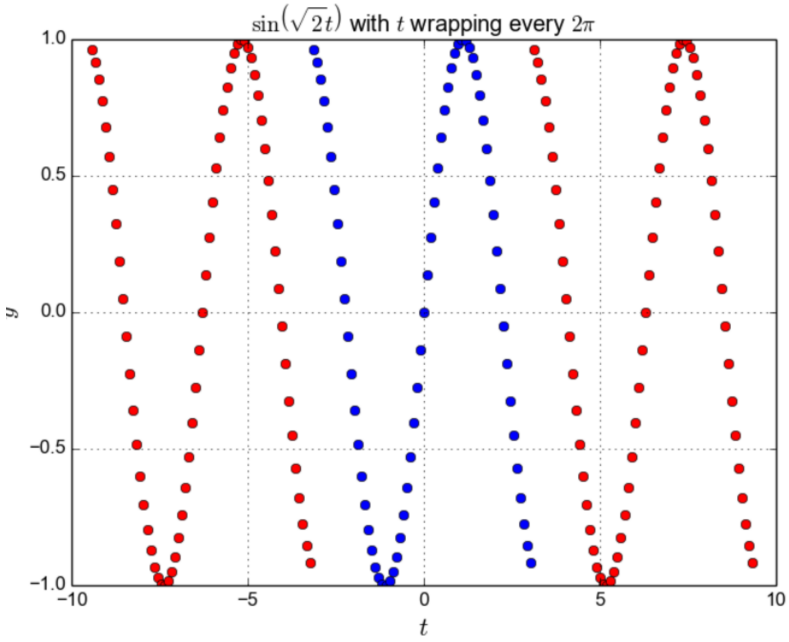
3. Analysis of an Asymmetric Signal

If we have a signal, lets say cos(t). If we plot the DFT of cos(t) we expect and obtain two spikes at 1 and -1.// But if we do the same thing to $\sin(\sqrt{2}t)$, we don't get what we expect. We will analyse why that is the case.

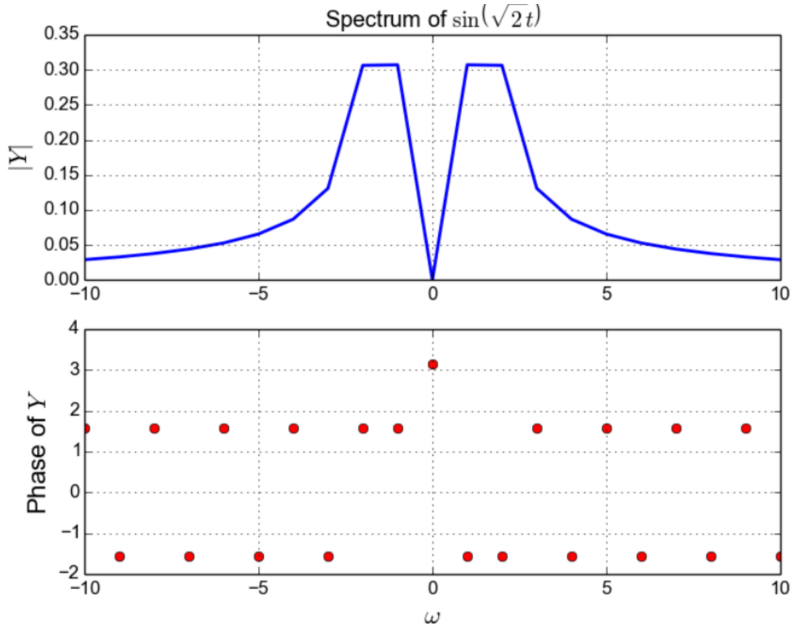
3.1 Plotting of asymmetrical signal :

Let us plot the signal $\sin(\sqrt{2}t)$, we know that the signal is periodic but when we consider signal between $-\pi$ to π and try to repeat it we do not get a proper sinusoid.

```
from pylab import *
t1=linspace(-pi,pi,65);t1=t1[:-1]
t2=linspace(-3*pi,-pi,65);t2=t2[:-1]
t3=linspace(pi,3*pi,65);t3=t3[:-1]
y=sin(sqrt(2)*t1)
figure(3)
plot(t1,y,'bo',lw=2)
plot(t2,y,'ro',lw=2)
plot(t3,y,'ro',lw=2)
ylabel(r"$y$",size=16)
xlabel(r"$t$",size=16)
title(r"$\sin\left(\sqrt{2}t\right)$ with $t$ wrapping every $2\pi$ ")
grid(True)
show()
```



We can clearly see that the signal is not $\sin(\sqrt{2}t)$, and so the spectrum of the signal looks as below.




```
from pylab import *
t=linspace(-pi,pi,65);t=t[:-1]
dt=t[1]-t[0];fmax=1/dt
y=sin(sqrt(2)*t)
y[0]=0
y=fftshift(y) # make y start with y(t=0)
```

```
Y=fftshift(fft(y))/64.0
w=linspace(-pi*fmax,pi*fmax,65);w=w[:-1]
figure()
subplot(2,1,1)
plot(w,abs(Y),lw=2)
xlim([-10,10])
ylabel(r"$|Y|$",size=16)
title(r"Spectrum of $\sin\left(\sqrt{2}t\right)$")
grid(True)
subplot(2,1,2)
plot(w,angle(Y),'ro',lw=2)
xlim([-10,10])
ylabel(r"Phase of $Y$",size=16)
xlabel(r"$\omega$",size=16)
grid(True)
show()
```

Due to the jump at $-\pi$ and π we do not get two peaks and instead get a maximum which is decaying. Due to this we cannot find the frequency of a periodic but asymmetric signal if we have its spectrum.To make the peaks more visible we window the signal.

3.2 Windowing

If we window the signal by damping the near the end of the function as it jumps we get a signal closer to the original. This is known as a window sequence and the DFT is obtained by convolution both the signals.

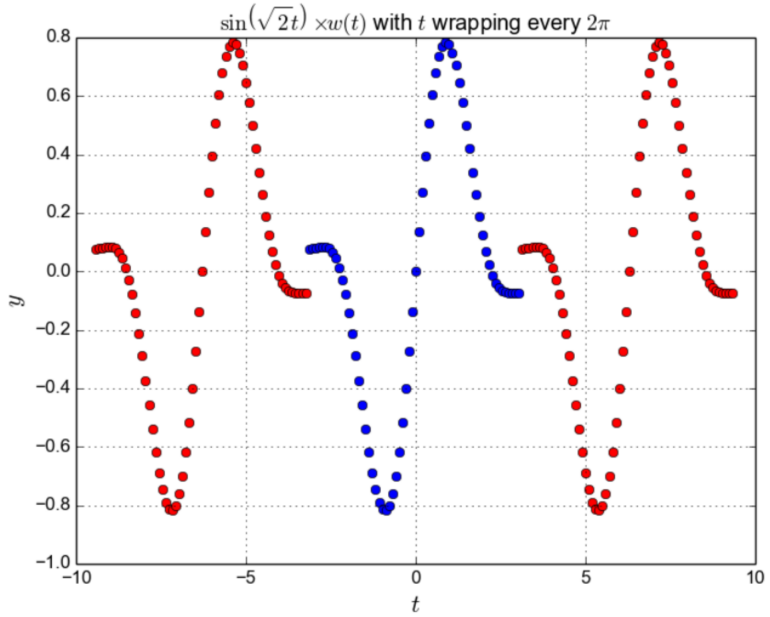
$$g(n) = f(n)w(n)$$
$$G_k = \sum_{n=0}^{N-1} F_n W_{k-n}$$

In this assignment we are going to be using a window called Hamming Window.
i.e.

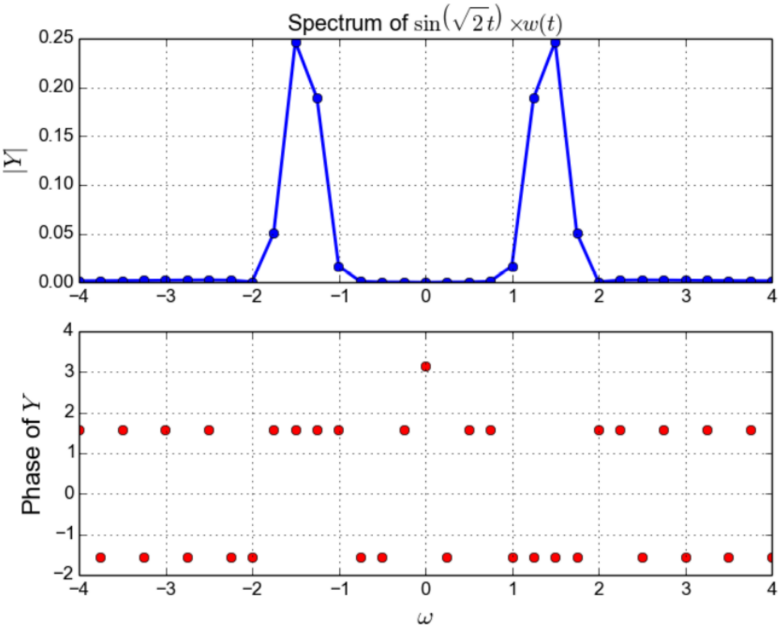
$$w[n] = \begin{cases} 0.54 + 0.46cos(2\pi n/N - i), & |n| < \frac{N-1}{2} \\ 0, & \text{else} \end{cases}$$

3.3 Plot with Windowing

If we multiply sin($\sqrt{2}$t) with w[n] we get a windowed signal with damping at the end.



If we obtain DFT of the signal with previous procedure we get a better spectrum. If we increase the sampling then we get a much better plot with defined peaks.

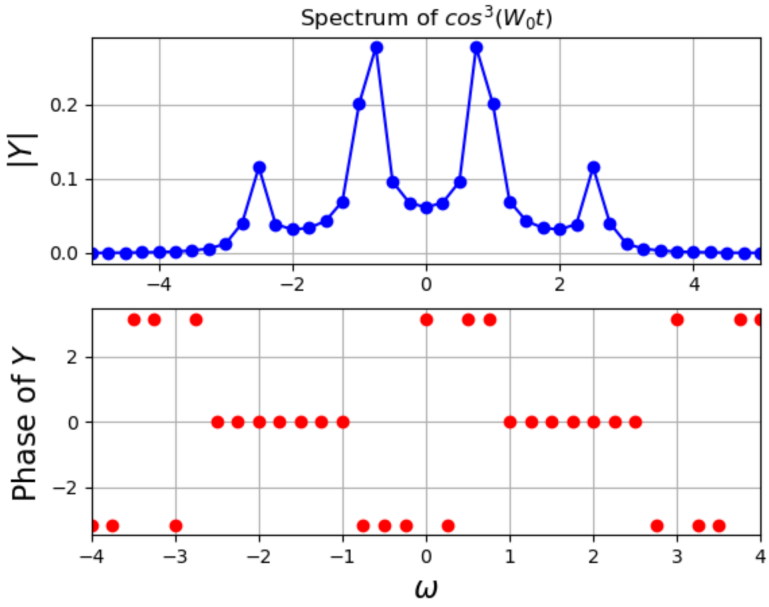


4. The Assignment

4.1 Problem 1

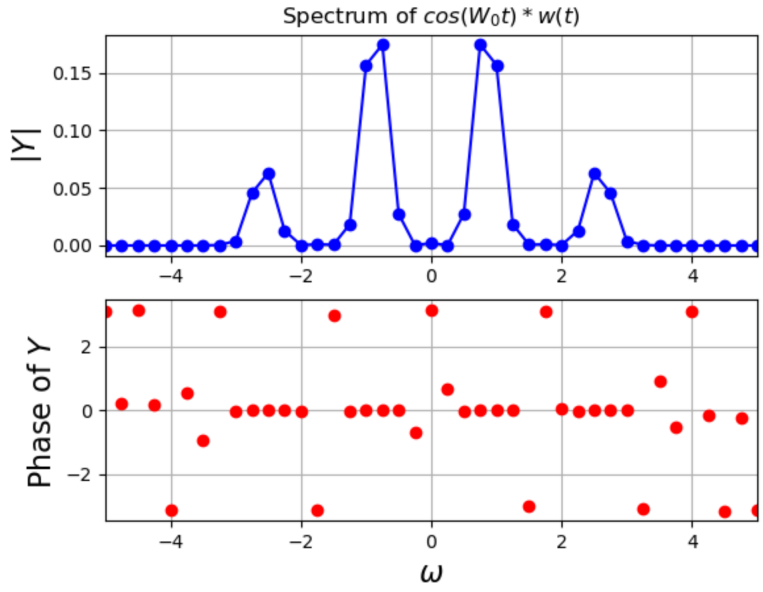
In the first problem we have to find the frequency spectrum of $cos^3(0.86t)$. Similar to previous assignment we first sample the signal. We then find the DFT of the signal and and plot it against frequency. In this assignment we additionally multiply a window with same amount of samples and then find the DFT.
We first find the spectrum of the signal without windowing.

```
N=2**8
w_0=0.86
t=linspace(-4*pi,4*pi,N+1);t=t[:-1]
dt=t[1]-t[0];fmax=1/dt
n=arange(N)
wnd=fftshift(0.54+0.46*cos(2*pi*n/(N-1)))
y=cos(w_0*t)**3
y[0]=0
Y=fftshift(fft(y))/N
w=linspace(-pi*fmax,pi*fmax,N+1);w=w[:-1]
figure()
subplot(2,1,1)
plot(w,abs(Y),'bo-')
xlim([-5,5])
ylabel(r"$|Y|$",size=16)
title(r"Spectrum of $cos^3(W_0t)$")
subplot(2,1,2)
plot(w,angle(Y),'ro')
xlim([-4,4])
ylabel(r"Phase of $Y$",size=16)
xlabel(r"$\omega$",size=16)
```



Now we obtain the spectrum after windowing.

```
y=(cos(w_0*t)**3)*wnd
y[0]=0
y=fftshift(y)
Y=fftshift(fft(y))/N
w=linspace(-pi*fmax,pi*fmax,N+1);w=w[:-1]
figure()
subplot(2,1,1)
plot(w,abs(Y),'bo-')
xlim([-5,5])
ylabel(r"$|Y|$",size=16)
title(r"Spectrum of $cos(W_0t)*w(t)$")
subplot(2,1,2)
plot(w,angle(Y),'ro')
xlim([-5,5])
ylabel(r"Phase of $Y$",size=16)
xlabel(r"$\omega$",size=16)
```



We obtain much better peaks in the spectrum when we window the signal.

4.2 Problem 2

In this problem we have a 128 element vector of the form $cos(w_0t + \delta)$. Here w_0 lies between 0.5 and 1.5 and δ is arbitrary. We will take an arbitrary w_0 and δ and try to predict the values from its spectrum.

```
w0=0.84
delta=0.9
N=128
n=arange(N)
wnd_2=fftshift(0.54+0.46*cos(2*pi*n/N-1))
t=linspace(-pi,pi,N+1);t=t[:-1]
dt=t[1]-t[0];fmax=1/dt
w=linspace(-pi*fmax,pi*fmax,N+1);w=w[:-1]
y_2=cos(w0*t+delta)*wnd_2
y_2[0]=0
figure()
plot(t,y_2)
Y_2=fftshift(fft(y_2)/N)
figure()
subplot(2,1,1)
xlim([-4,4])
plot(w,abs(Y_2))
subplot(2,1,2)
xlim([-4,4])
```

```
plot(w,angle(Y_2),'ro')
w_2p = sum(abs(Y_2**1.7*w))/sum(abs(Y_2)**1.7)
del_p=angle(Y_2[::-1][argmax(abs(Y_2[::-1]))])
if del_p>=0:
  del_p=pi-del_p
else:
  del_p=pi+del_p
print('Actual Values : W={} Delta={}'.format(w0,delta))
print('Predicted Values:W= {}Delta={}'.format(w_2p,del_p))
```

We estimate ω_0 by finding estimation of spectrum.

$$\omega_0 = \frac{\sum(Yw)^p}{\sum(w)}$$

On trying different values of p, for p = 1.7 we found the best estimate of ω_0. For estimating phase we can just find the angle of spectrum at peak value of the magnitude since the phase doesn't change much.

4.3 Problem 3

In this problem we have to find the values of w_0 and δ but with noise. Noise can be generated using ***0.1rand(128)*** with max amplitude of 0.1 and of 128 samples.

```
y_3=y_2+0.1*randn(N)
Y_3=fftshift(fft(y_3)/N)
w_2p = sum(abs(Y_3**2.4*w))/sum(abs(Y_3)**2.4)
del_p=angle(Y_2[argmax(abs(Y_3))])
if del_p>=0:
del_p=pi-del_p
else: del_p=pi+del_p
```

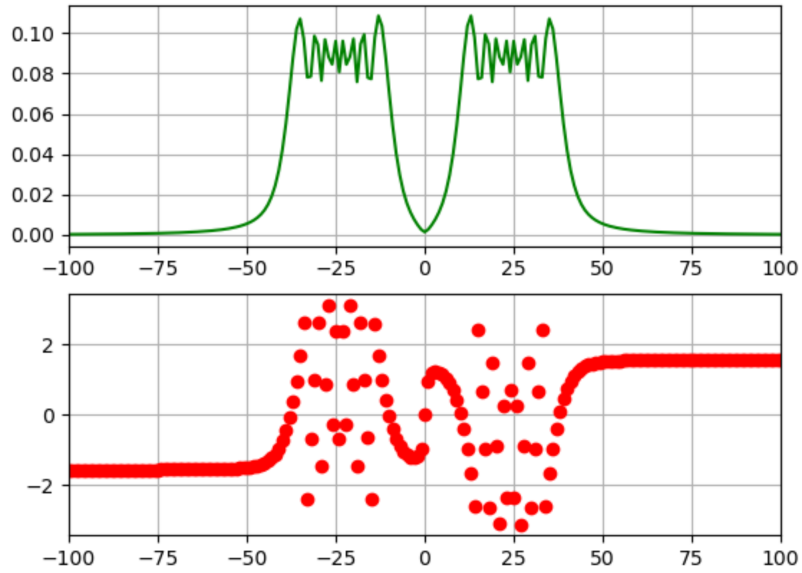
Similar to previous problem we estimate w_0. But in this case the coefficients p = 2.4 for obtaining better estimates. Similarly phase closely corresponds to peak value of magnitude.

4.4 Problem 4

In this problem we have to find DFT of

$$cos(16(1.5 + t/2\pi)t)$$

We have to sample from $-\pi$ to π in 1024 steps. This is known as a "chirped" signal, and its frequency continuously changes from 16 to 32 radians per second.




```
N=2**10
t=linspace(-pi,pi,N+1);t=t[:-1]
dt=t[1]-t[0];fmax=1/dt
y_4=cos(16*t*(1.5+t/(2*pi)))
Y_4=fftshift(fft(y_4))/N
w=linspace(-pi*fmax,pi*fmax,N+1);w=w[:-1]
figure()
subplot(2,1,1)
plot(w,abs(Y_4),color='green')
xlim([-100,100])
subplot(2,1,2)
plot(w,angle(Y_4),'ro')
```

4.5 Problem 5

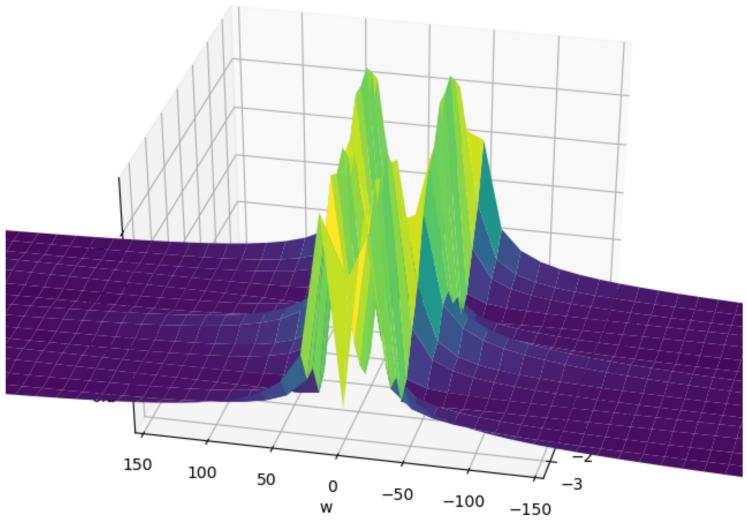
In the last problem we have to break the 1024 vector into pieces that are 64 samples wide. Extract the DFT of each and store as a column in a 2D array. Then we have to plot the array as a surface plot to show how the frequency of the signal varies with time.

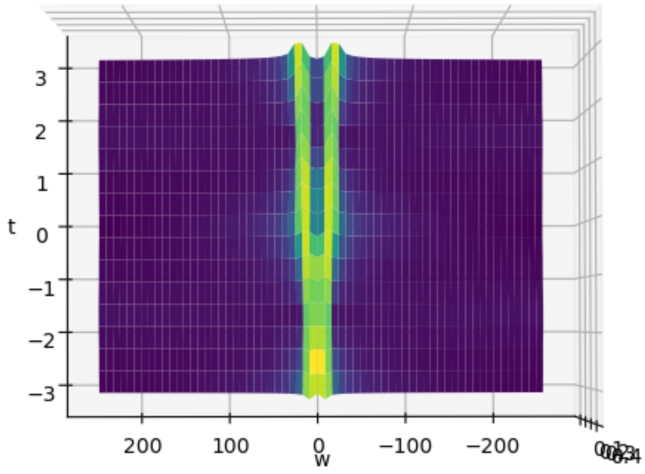
```
Y=[]
for i in range(16):
  tlim_1= -pi + 2*pi*i/16
  tlim_2= -pi + 2*pi*(i+1)/16
  t=linspace(tlim_1,tlim_2,65);t=t[:-1]
  y=cos(16*t*(1.5+t/(2*pi)))
  y[0]=0
```

```
  Y.append((fftshift(fft(y))/64))

Yd=array(Y)
t1=linspace(-pi,pi,16)
w=linspace(-pi*fmax/2,pi*fmax/2,65);w=w[:-1]
t1 , w= meshgrid(t1,w)
ax= Axes3D(figure())
surf = ax.plot_surface(t1,w,abs(Yd).T,
rstride=1, cstride=1, cmap='viridis')
xlabel('t')
ylabel('w')
```

We first find DFT for different time intervals in lengths of 64 and 16 sets.Now we form a mesh-grid of t,w and plot DFT of the signal on Z-axis.





From the figures we can clearly see that the peaks are getting separated as time goes on.This represents different frequencies in the signal.

5. Observations

Asymmetric signals are not easy to deal with and we have to chose proper windows to get defined peaks in the spectrum. We saw that we do not get exact results when trying to estimate frequency of signals and with the added noise we get even worse results.
We have to be careful in dealing with these signals and obtaining proper results.
Sometimes it might be better to look at signals in time domain and we can obtain more information. The conclusion is that we can come up with intelligent ways in processing the signal better.