ASSIGNMENT 9: Spectra of non-periodic signals

Hariharan P EE20B042

April 28, 2022

1 Brief Up:

The tasks of this assignment is the following:

- Obtaining the DFT of non-periodic functions
- Seeing how the use of Hamming Window make the DFT better
- Analyze graph

2 Introduction:

- We focus on finding transforms of non periodic functions in this assignment. Due to the discontinuity when periodically extended, fourier components in frequencies other than the sinusoids frequency are also produced which decay as $\frac{1}{\omega}$, due to Gibbs phenomenon.
- To tackle this, hamming window is used. This make the signal square integrable, and that the function goes sufficiently rapidly towards 0 , also to make infinitely long signal to a finite signal.
- We also perform a sliding DFT on a chirped signal and plot the results

3 Initialising necessary functions:

We declare the functions to efficiently run the program.

- A function to find the DFT spectrum and plot is initialised.

```
def spectrum_plot(lim,n,f,t_check=0,show_check = True,
t_lims = False,windowing= False,xlim1=10,
title = r"Spectrum of $\sin\left(\sqrt{2}t\right)$"
,xlabel = r"$\omega$",ylabel1= r"$|Y|$",
ylabel2 = r"Phase of $Y$"):
    if(t_lims):
```

```
        t = t_check
    else:
        t=linspace(-lim,lim,n+1)[:-1]
    dt=t[1]-t[0]
    fmax=1/dt
    y = f(t)
    if (windowing):
        m=arange(n)
        wnd=fftshift(0.54+0.46*cos(2*pi*m/n))
        y = y*wnd
    y[0]=0
    y=fftshift(y)
    Y=fftshift(fft(y))/float(n)
    w=linspace(-pi*fmax,pi*fmax,n+1)[:-1]

    mag = abs(Y)
    phase = angle(Y)
    if(show_check):
        figure()
        subplot(2,1,1)
        plot(w,mag,lw=2)
        xlim([-xlim1,xlim1])
        ylabel(ylabel1,size=16)
        title(title)
        grid(True)
        subplot(2,1,2)
        ph[where(mag<3e-3)] = 0
        plot(w,phase,'ro',lw=2)
        xlim([-xlim1,xlim1])
        ylabel(ylabel2,size=16)
        xlabel(xlabel,size=16)
        grid(True)
        show()
    return w,Y
```

- Functions of the signals are initialised.

```
def sinsq(t):
    return sin(sqrt(2)*t)

def cos3(t,w0=0.86):
    return (cos(w0*t))**3

def cosine(t,w0=1.5,delta=0.5):
```

```
    return cos(w0*t + delta)

def noisycosine(t,w0=1.5,delta=0.5):
    return cos(w0*t + delta) + 0.1*randn(128)

def chirp(t):
    return cos(16*(1.5 + t/(2*pi))*t)
```

- A function to find the omega and delta using weighted mean is initialised.

```
def omega_delta(w,Y):
    ii = where(w>0)
    omega = (sum(abs(Y[ii])**2*w[ii])/sum(abs(Y[ii])**2))
    i = abs(w-omega).argmin()
    delta = angle(Y[i])
    print ("omega = ", omega)
    print ("delta = ", delta)
```

4 DFT of $sin(\sqrt{2}t)$:

4.1 Without Hamming Window:

Code to compute and plot of Spectrum of $sin(\sqrt{2}t)$ without hamming window is given below

```
w,Y = spectrum_plot(pi,64,sinsq,xlim1= 10,windowing=False,
title = r"Spectrum of $\sin\left(\sqrt{2}t\right)$ without windowing ")
```

Since the DFT is computed over a finite time interval, discontinuities occur for these aperiodic signals. These discontinuities lead to non harmonic components in the FFT which decay as $\frac{1}{\omega}$.

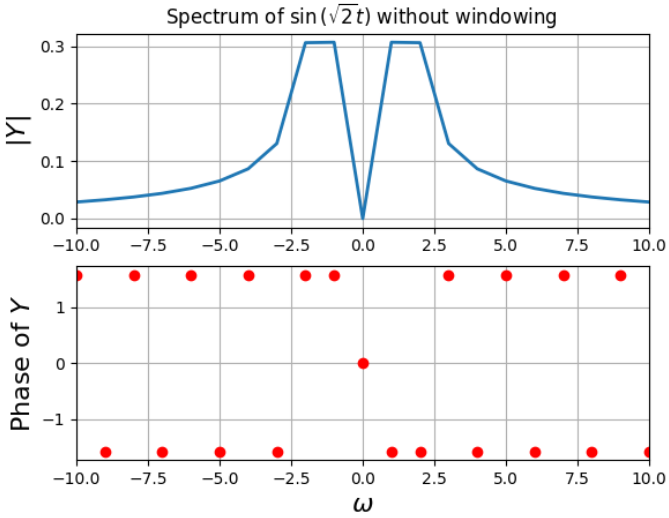


Figure 1: Spectrum of $\sin(\sqrt{2}t)$

4.2 With Hamming Window:

The hamming window removes discontinuities by attenuating the high frequency components that cause the discontinuities. The hamming window function is given by

$$x[n] = 0.54 + 0.46cos(\frac{2\pi n}{N-1}) \tag{1}$$

We now multiply our signal with the hamming window and periodically extend it. The discontinuities almost vanish. The spectrum that is obtained with a time period 8π is given below:

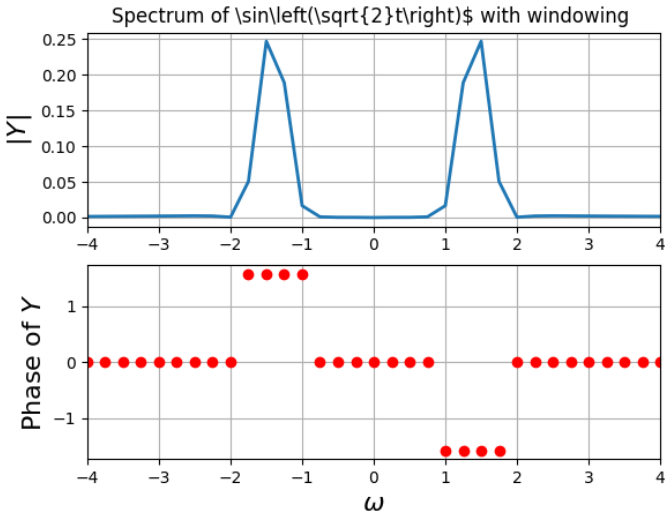


Figure 2: Spectrum of $\sin(\sqrt{2}t) * w(t)$

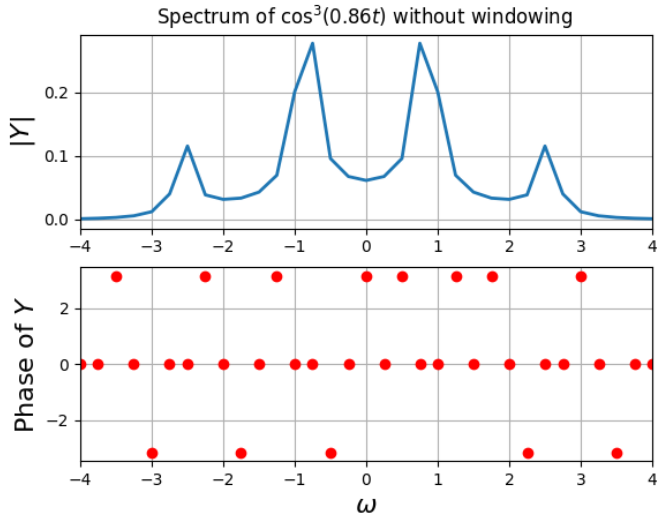
5 DFT of $cos^3(0.86t)$:

5.1 Without hamming window:

In this question, we shall plot the FFT of $cos^3(0.86t)$

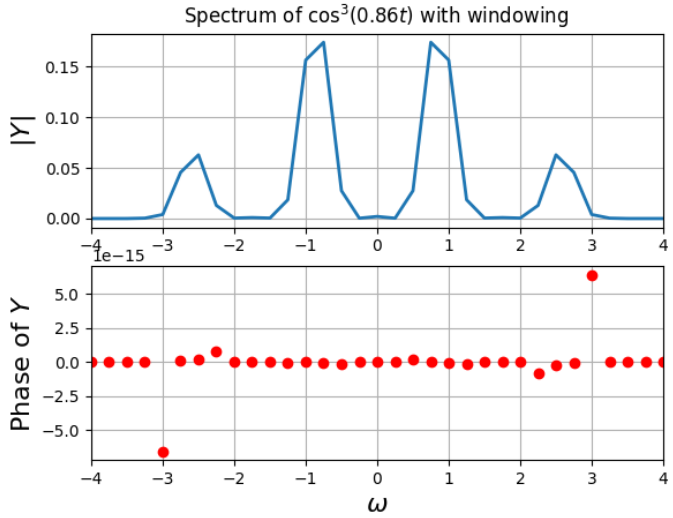
```
w,Y = spectrum_plot(4*pi,256,cos3,xlim1= 4,windowing=False,
title = r"Spectrum of $\cos^{3}(0.86t)$ without windowing ")
```

The FFT without the hamming Window:



The FFT with the hamming Window:

```
w,Y = spectrum_plot(4*pi,256,cos3,xlim1= 4,windowing=True,
title = r"Spectrum_of_$\cos^{3}(0.86t)$_with_windowing_")
```



We notice that a lot of the energy is stored in frequencies that aren't a part of the signal. After windowing, these frequencies are attenuated and hence the peaks are sharper in the windowed function. It is still not an impulse because the convolution with the Fourier transform of the windowed function smears out the peak

6 DFT of $\cos(\omega t + \delta)$ and Estimating ω and δ:

- The resolution is not enough to obtain the 0 directly if the spectrum is obtained. Since the resolution of the frequency axis is not enough, the peak won't be visible clearly. So a statistic derivation is necessary to estimate value of 0. Thus, we can compute 0 by taking a weighted average of all the weighted with the magnitude of the DFT.
- can be found by calculating the phase of the DFT at o nearest to estimated using the above statistic.
- This works because the phase of cos(ot+) when = 0 is 0, so when its not its , so we can estimate it by this approach.

We estimate ω and δ for a signal $\cos(\omega t + \delta)$ for 128 samples between $[-\pi, \pi)$. We then extract the digital spectrum of the signal and find the two peaks at $\pm\omega_0$, and estimate ω and δ.

6.1 Without white noise:

```
w,Y = spectrum_plot(pi,128,cosine,xlim1= 4,windowing=True,
title = r"Spectrum of $\cos(w_0t+\delta)$ with windowing ")
omega_delta(w,Y)
```

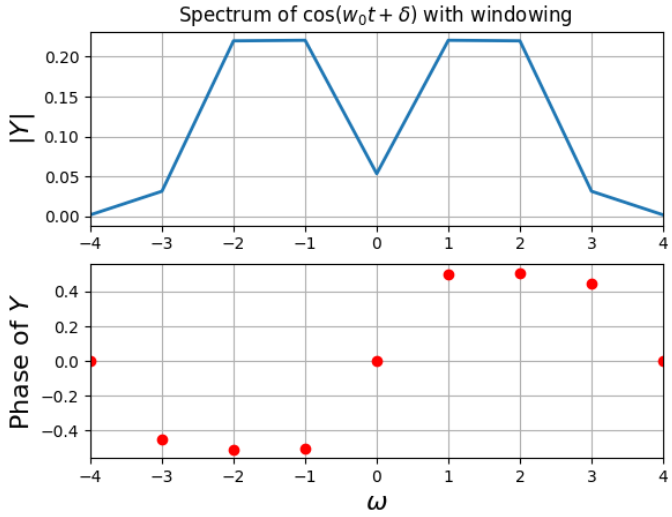


Figure 3: Fourier transform of $cos(1.5t + 0.5)$

omega = 1.5163179648582412
delta = 0.506776265719626

6.2 With white noise:

We repeat the exact same process but with noise added to the original signal.

```
w,Y = spectrum_plot(pi,128,noisycosine,xlim1= 4,windowing=True,
title1 = r"Spectrum of noisy $\cos(w_0t+\delta)$ with windowing ")
omega_delta(w,Y)
w,Y= spectrum_plot(pi,1024,chirp,xlim1= 60,windowing=False,
```

omega = 1.9604500369109494
delta = 0.5073357754766181

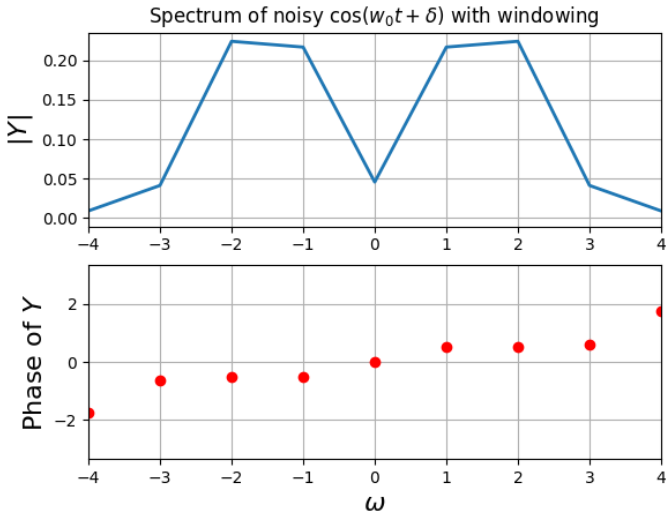


Figure 4: Fourier transform of noisy $cos(1.5t + 0.5)$

7 Chirped Signal Spectrum:

- We analyze a chirp signal which is an FM signal where frequency is directly proportional to time.A chirp signal we shall consider is given by

$$f(t) = cos(16t(1.5 + \frac{t}{2\pi})) \tag{2}$$

- Its frequency continuously changes from 16 to 32 radians per second. This also means that the period is 64 samples near and is 32 samples near +

- Due to Gibbs phenomenon, the frequency range is large. On windowing, only frequencies between 16 and 32 rad/s remain.

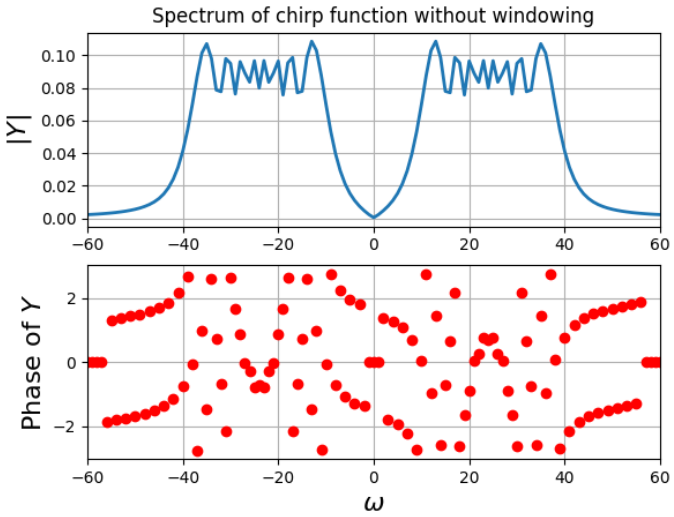


Figure 5: window-less Chirp function fourier transform

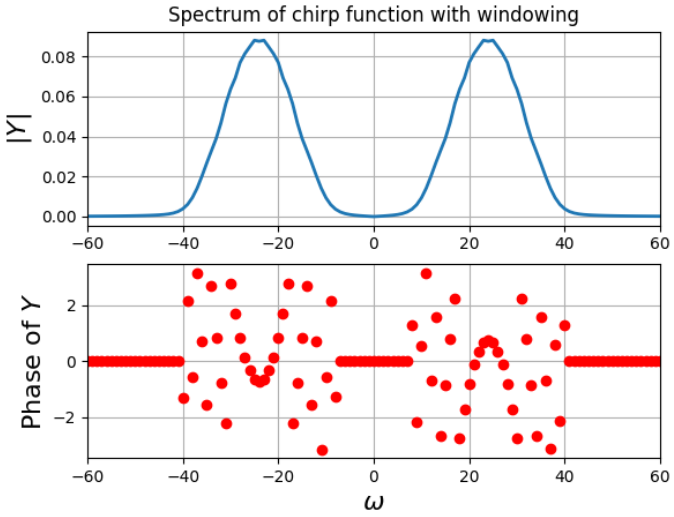


Figure 6: windowed Chirp function fourier transform

8 Frequency vs Time for chirp signal:

- For the above mentioned chirped signal, 16 pieces of 64 samples wideare made out of 1024 samples.
- We then extract the DFT of each and store as a column in a 2D array. Then plot the array as a surface plot to show how the frequency of the

signal varies with time.

- This is a "time- frequency" plot,and we get localized DFTs . We do this for both phase and magnitude.

```
t = linspace(-pi,pi,1025); t = t[:-1]
dt = t[1]-t[0]; fmax = 1/dt
t_array = split(t,16)
Y_mag = zeros((16,64))
Y_phase = zeros((16,64))
for i in range(len(t_array)):
    w,Y = spectrum_plot(0,64,f = chirp,show_check = False,t_lims=True,
    t_check = t_array[i],xlim1= 60,windowing=True,
    title = r"Spectrum of chirp function")
    Y_mag[i] =  abs(Y)
    Y_phase[i] =  angle(Y)

t= t[::64]
w = linspace(-fmax*pi,fmax*pi,65); w = w[:-1]
t,w = meshgrid(t,w)

fig1 = figure(9)
ax = fig1.add_subplot(111, projection='3d')
surf=ax.plot_surface(w,t,Y_mag.T,cmap='viridis',linewidth=0, antialiased=
fig1.colorbar(surf, shrink=0.5, aspect=5)
ax.set_title('magnitude surface plot');
ylabel(r"$\omega\rightarrow$")
xlabel(r"$t\rightarrow$")
show()
fig1 = figure(10)
ax = fig1.add_subplot(111, projection='3d')
surf=ax.plot_surface(w,t,Y_phase.T,cmap='viridis',linewidth=0, antialiase
fig1.colorbar(surf, shrink=0.5, aspect=5)
ax.set_title('phase surface plot');
ylabel(r"$\omega\rightarrow$")
xlabel(r"$t\rightarrow$")
show()
```

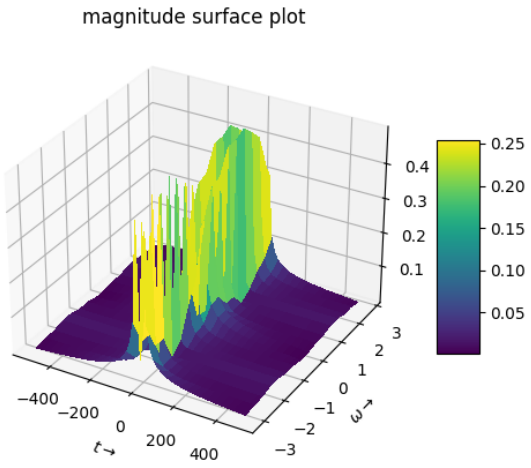


Figure 7: Chirp function, —Fourier transform—

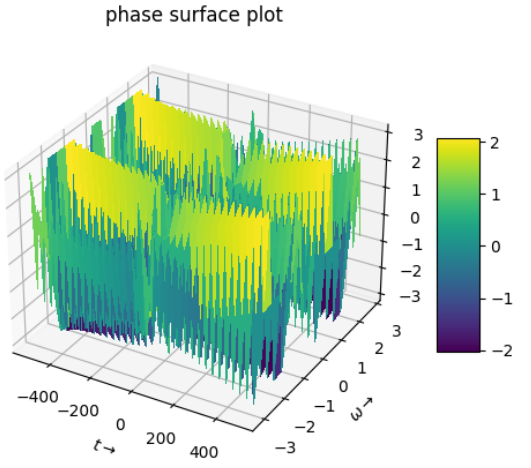


Figure 8: Chirp function, Phase of Fourier transform

9 Conclusion:

- We have emphasised the importance and necessity of windowing in the case of non-periodic series in DFT's. In particular this is to mitigate the effect of Gibbs phenomena owing to the discontinuous nature of the series $\tilde{x}[n]$ realised by a discrete fourier transform.
- We addressed the time varying spectra for a chirped signal, where we plot fourier spectra for different time slices of a signal.
- Existence of two peaks (slow growth), vanishing of chirp effects in case of a windowed transform, and a phase plot that periodically varies with reduced phase near maximum values are also observed for a chirp signal.