Assignment 9

Kathir Pagalavan EE20B056

April 29, 2022

Aim

To plot DFT spectra of non-periodic functions and to use hamming windows to make the DFT better.

Assignment examples

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

We will plot the DFT spectra of $sin(\sqrt{2}t)$ both with and without windows.The python code for the same is attached below

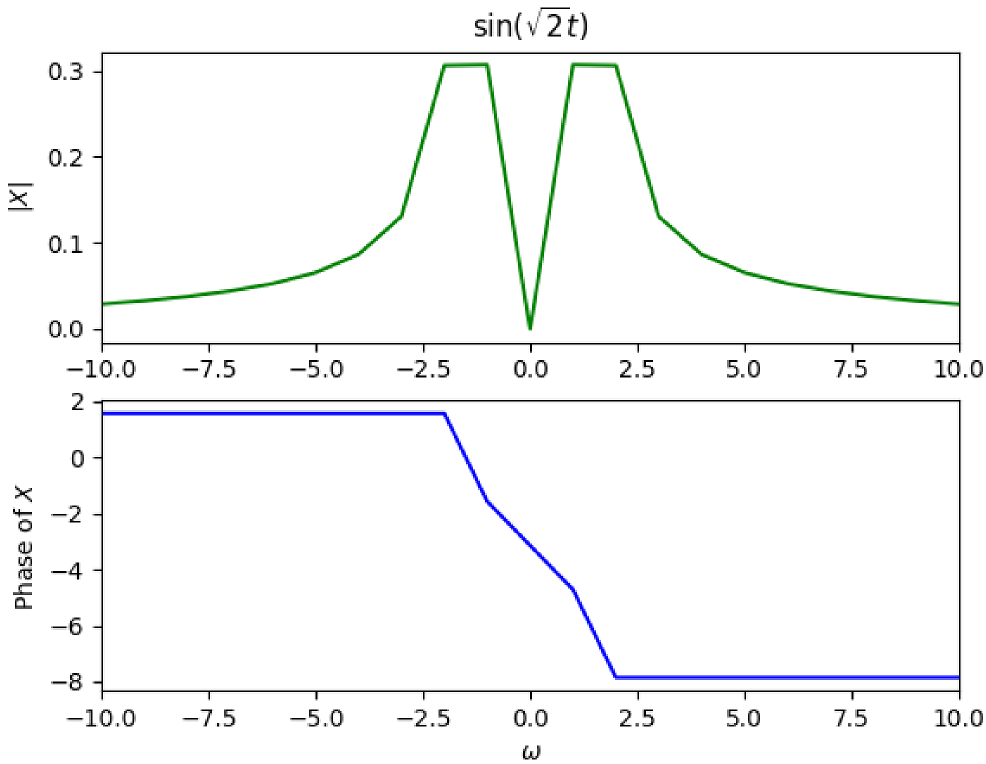

Figure 1: Spectrum of $sin(\sqrt{2}t)$

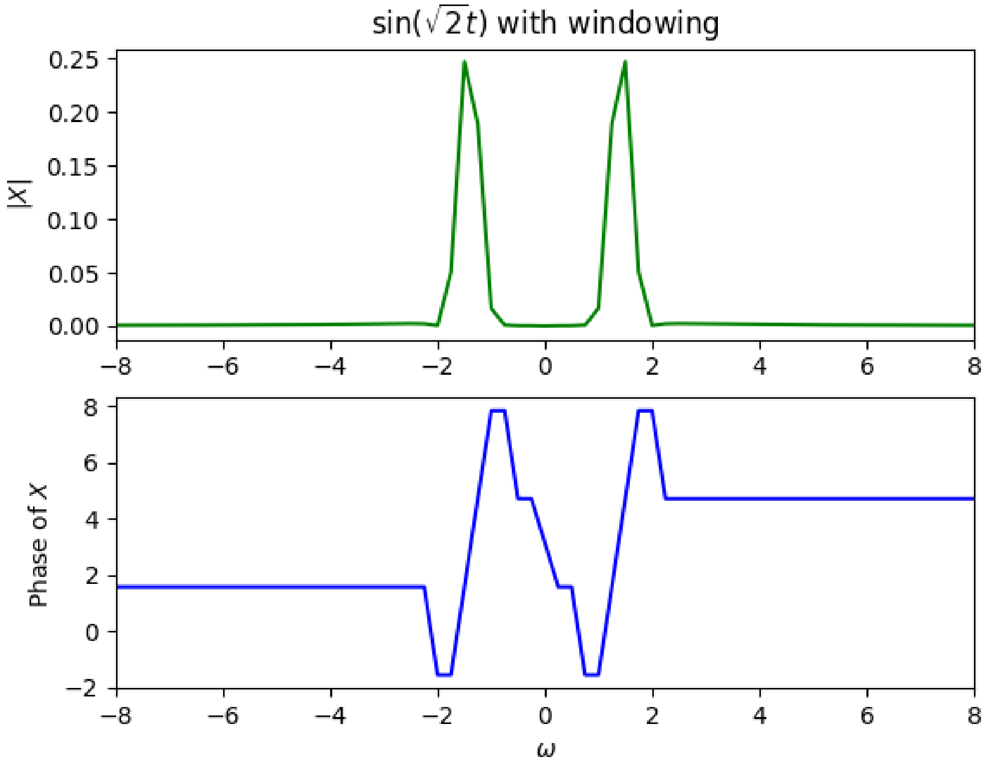

Figure 2: Spectrum of $sin(\sqrt{2}t)$ with windowing

DFT of $cos^3(\omega_o t)$

We find the DFT of $cos^3(\omega_o t)$ function with $\omega_o t = 0.86$ with and without hamming windows. The python code for the same is attached below.

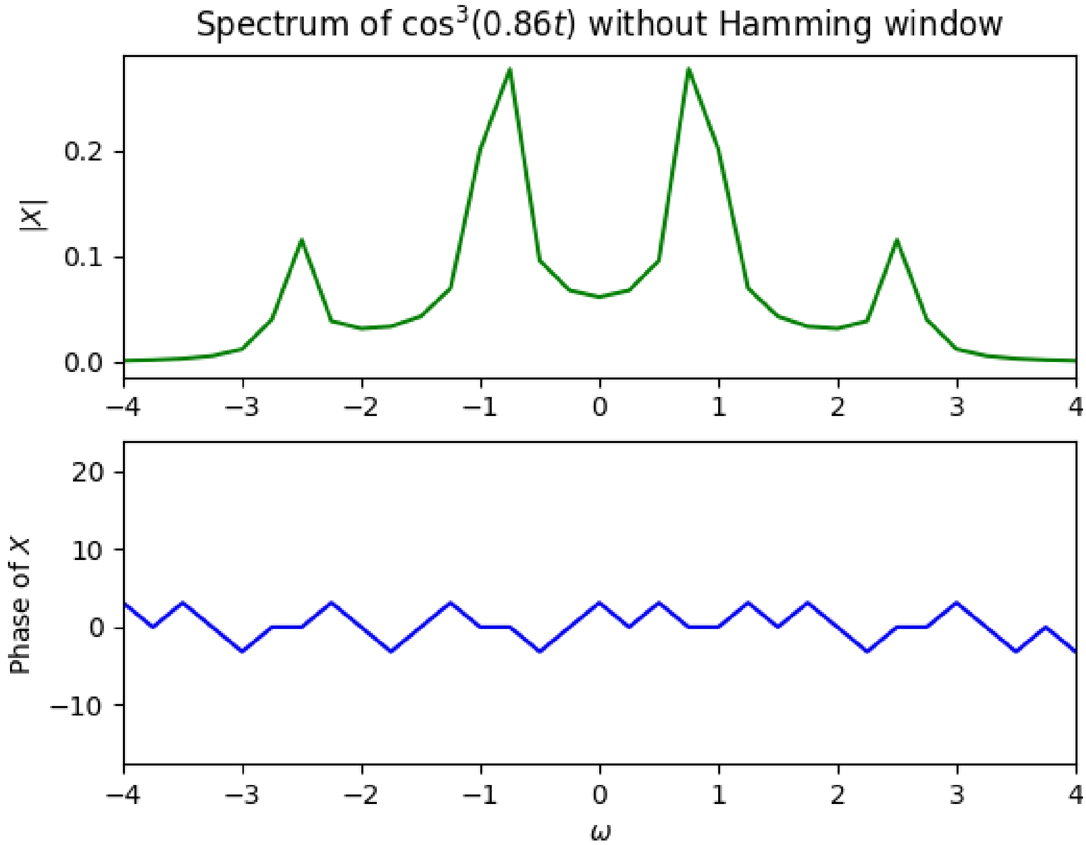

Figure 3: Spectrum of $\cos^3(0.86t)$

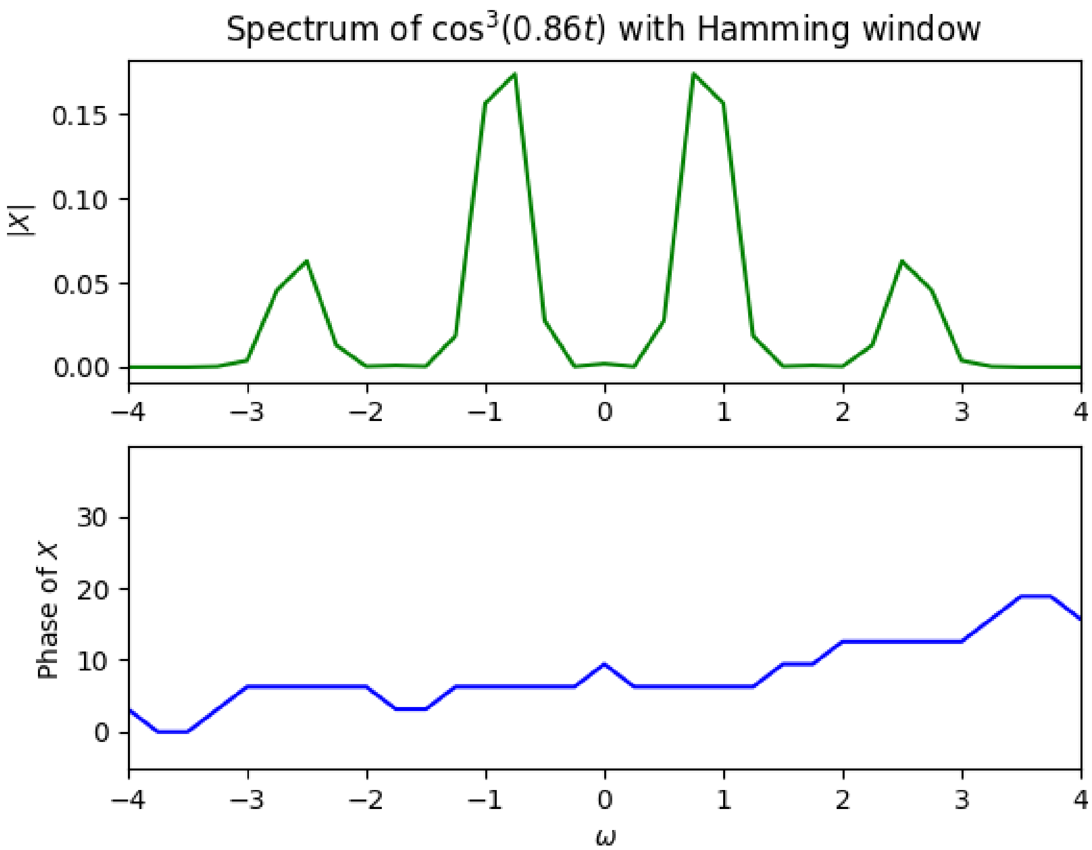

Figure 4: Spectrum of $\cos^3(0.86t)$ with hamming window

Estimating ω_o and δ

The peak will not be visible since the resolution is not enough and hence we find ω_o by taking weighted means of phases using magnitude squared weights.and δ is found by calculating phase at that point.The same task is repeated for a noised version of the same function.The code for the same is attached below

```
#3 w0 and d determination from a given cos(w0t+d) sample vector
# w0=1.234 d=0.456

w0 = 1.456
d = 0.456

t = linspace(-pi,pi,129)[:-1]
dt = t[1]-t[0]; fmax = 1/dt
n = arange(128)
wnd = fftshift(0.54+0.46*cos(2*pi*n/128))
y = cos(w0*t + d)*wnd
```

```
y[0]=0
y = fftshift(y)
Y = fftshift(fft(y))/128.0
w = linspace(-pi*fmax,pi*fmax,129); w = w[:-1]
dftplot(y,w,4,r"Spectrum of $\cos(w_0t+\delta)$ with Hamming window")

# w0 is calculated by finding the weighted average of all w>0. Delta is found by calcu
ii = where(w>=0)
w_cal = sum(abs(Y[ii])**2*w[ii])/sum(abs(Y[ii])**2)
i = abs(w-w_cal).argmin()
delta = angle(Y[i])
print("Calculated value of w0 without noise: ",w_cal)
print("Calculated value of delta without noise: ",delta)

#4 w0 and d determination from a given noised cos(w0t+d) sample vector
y = (cos(w0*t + d) + 0.1*randn(128))*wnd
y[0]=0
y = fftshift(y)
Y = fftshift(fft(y))/128.0
dftplot(y,w,4,r"Spectrum of a noisy $\cos(w_0t+\delta)$ with Hamming window")

# w0 is calculated by finding the weighted average of all w>0. Delta is found by calcu
ii = where(w>=0)
w_cal = sum(abs(Y[ii])**2*w[ii])/sum(abs(Y[ii])**2)
i = abs(w-w_cal).argmin()
delta = angle(Y[i])
print("\nCalculated value of w0 with noise: ",w_cal)
print("Calculated value of delta with noise: ",delta)
```

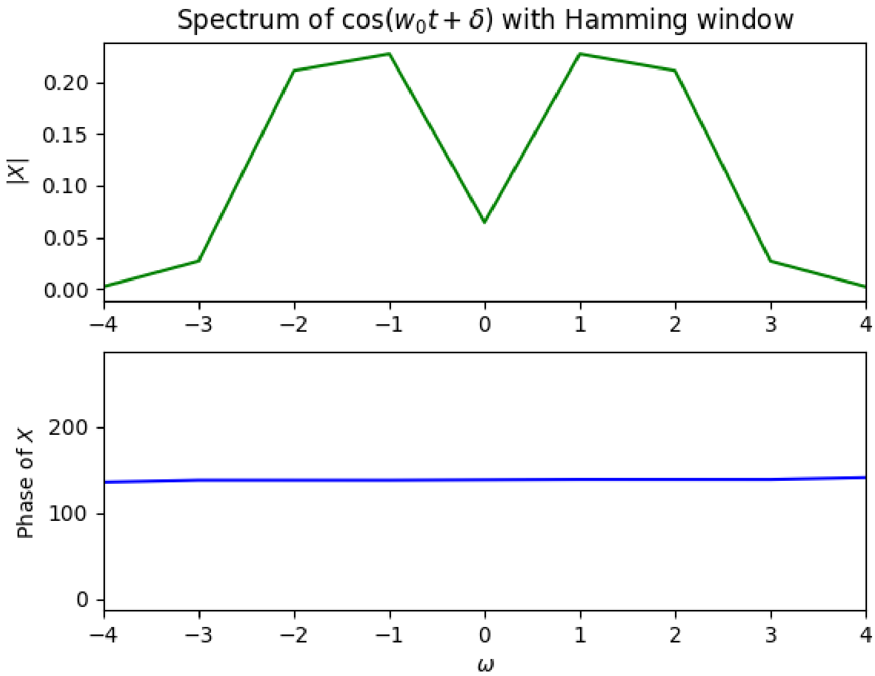

Figure 5: Spectrum of $\cos(w_0 t + \delta)$

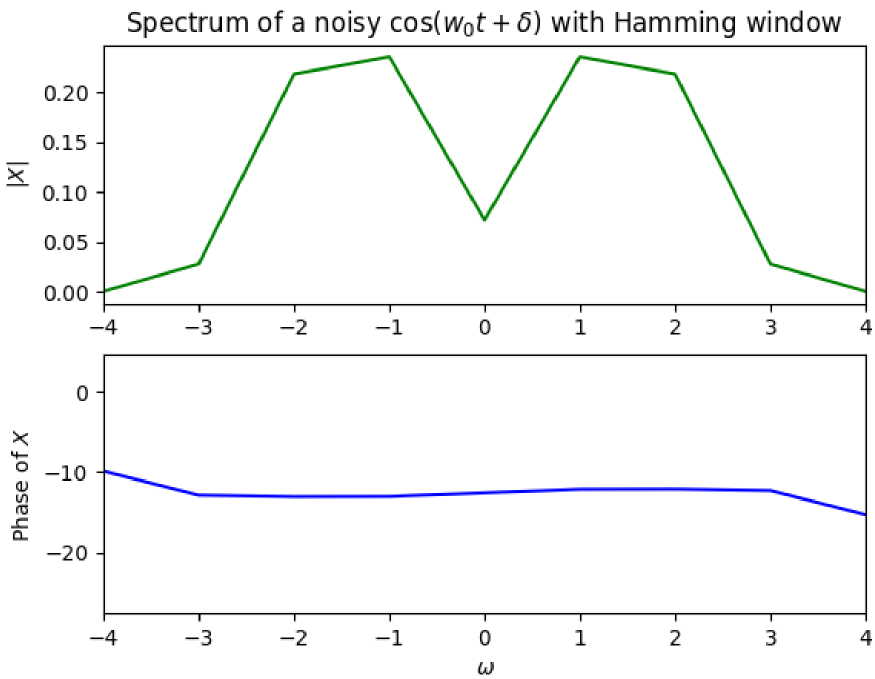

Figure 6: Spectrum of noised $\cos(w_0 t + \delta)$

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

Its frequency continuously changes from 16 to 32 radians per second. This also means that the period is 64 samples near $+\pi$ and is 32 samples near -π.The python code for calculating and plotting the above chirped signal is attached below

```
#5 DFT of cos(16(1.5+t/2pi)t)

t = linspace(-pi,pi,1025); t = t[:-1]
dt = t[1]-t[0]; fmax = 1/dt
n = arange(1024)
wnd = fftshift(0.54+0.46*cos(2*pi*n/1024))
y = cos(16*t*(1.5 + t/(2*pi)))*wnd
y[0]=0
y = fftshift(y)
Y = fftshift(fft(y))/1024.0
w = linspace(-pi*fmax,pi*fmax,1025); w = w[:-1]
dftplot(y,w,100,r"Spectrum of chirped function")
```

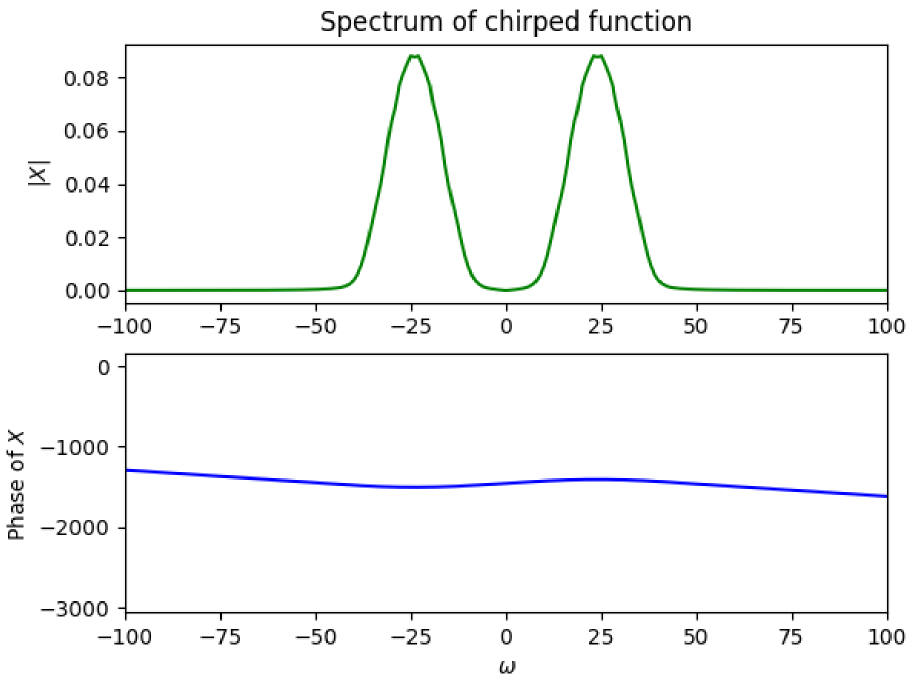

Figure 7: Spectrum of chirped function

Surface plot of chirped signal

The 1024 samples of the chirped signal are broken down into 64-sample pieces and DFT for each of them are found and plotted to show how the frequency varies with time.The code for the above is attached below

```
#6 surface plot of 64 samples wide DFTs vs t
t_array = split(t,16)
Y_mag = zeros((16,64))
Y_phase = zeros((16,64))

for i in range(len(t_array)):
        n = arange(64)
        wnd = fftshift(0.54+0.46*cos(2*pi*n/64))
        y = cos(16*t_array[i]*(1.5 + t_array[i]/(2*pi)))*wnd
        y[0]=0
        y = fftshift(y)
        Y = fftshift(fft(y))/64.0
        Y_mag[i] = abs(Y)
        Y_phase[i] = angle(Y)

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

fig1 = figure(7)
ax = fig1.add_subplot(111, projection='3d')
surf=ax.plot_surface(w,t,Y_mag.T,cmap=cm.jet,linewidth=0, antialiased=False)
fig1.colorbar(surf, shrink=0.5, aspect=5)
ax.set_title('surface plot');
ylabel(r"$\omega\rightarrow$")
xlabel(r"$t\rightarrow$")

show()
```

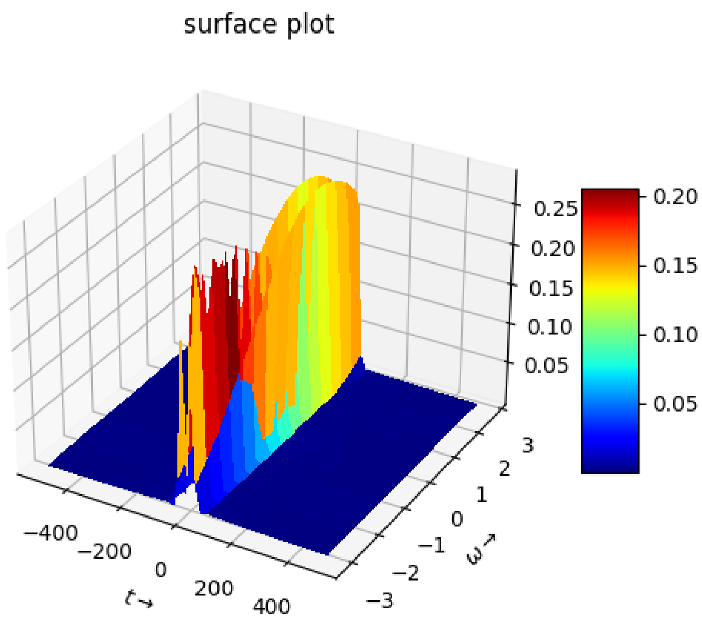

Figure 8: Spectrum of chirped function

Conclusion

We obtained DFT for non-periodic functions and used hamming windows to improve the results.We made a surface plot to see how frequency varies with time.