EE2703 : Applied Programming Lab
Assignment - 9

Jawhar S
EE20B049

May 12, 2022

Introduction

In this assignment, we continue our analysis of signals using Fourier Transforms. This time, we focus on finding transforms of non periodic functions. These functions have a discontinuity when periodically extended.
The discontinuity causes fourier components in frequencies other than the sinusoids frequency which decay as $\frac{1}{\omega}$, due to Gibbs phenomenon. We resolve this problem using a hamming window in the case of this assignment.
We use this windowed transform to analyse signals known to contain a sinusoid of unknown frequencies and extract its phase and frequency.

Function to plot Spectrum

We will be reusing the same function to plot the spectrum of different signals
The Python code is as follows:

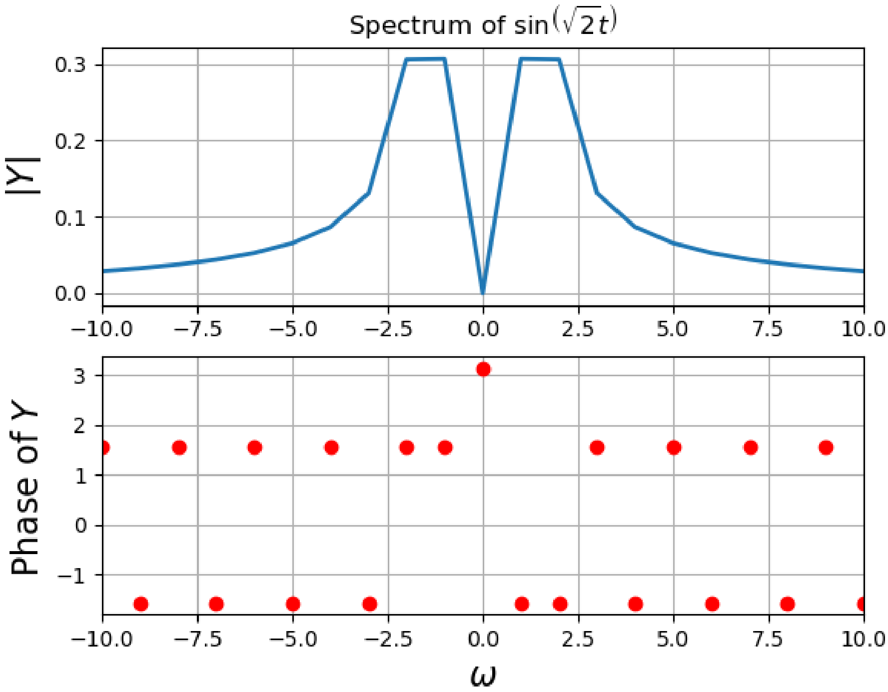
```
def get_dft(f, T_L, T_R, N, title_str, plot_flag = True, limx = 10, wnd_func

    t = linspace(T_L, T_R, N+1)[:-1]
    dt = t[1] - t[0]
    fmax = 1/dt
    n = arange(N)
    flt_N = float(N)

    if func_mode:
        y = f(t)
    else:
        y = f

    if wnd_func != None:
        wnd = wnd_func(n/flt_N)
        y = y*wnd

    y[0] = 0 # sample corresponding to -tmax should be set to 0
    y = fftshift(y)
    Y = fftshift(fft(y))/flt_N
    w = linspace(-pi*fmax, pi*fmax,  N+1)[:-1]

    if plot_flag:
        figure()
        subplot(2, 1, 1)
```

```
        plot(w, abs(Y), lw = 2)
        xlim([-limx, limx])
        ylabel(r"$|Y|$", size = 16)
        title(title_str)
        grid(True)

        subplot(2, 1, 2)
        plot(w, angle(Y), 'ro', lw = 2)
        xlim([-limx, limx])
        ylabel(r"$\angle Y$", size = 16)
        xlabel(r"$\omega$", size = 16)
        grid(True)
        show()

    return Y

fig_sno = 1
```

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

Without Hamming window

Since the DFT is computed over a finite time interval, We actually plotted the DFT for this function



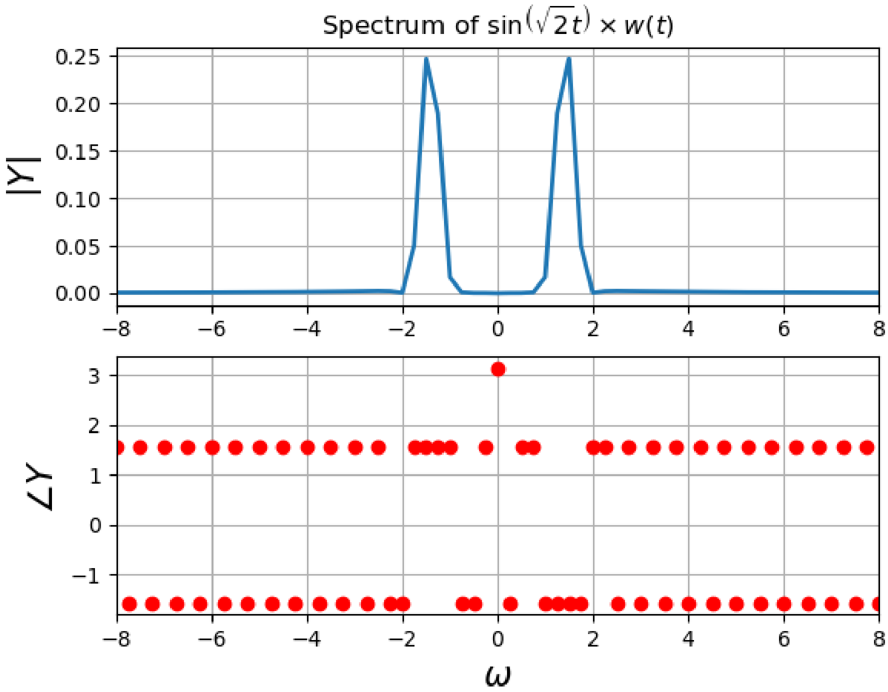

```
t = linspace(-pi, pi, 65)[:-1]
dt = t[1]-t[0]
fmax = 1/dt
y = sin(sqrt(2)*t)
y[0] = 0 # the sample corresponding to -tmax should be set zero
y = fftshift(y) # make y start with y(t = 0)
Y = fftshift(fft(y))/64.0
w = linspace(-pi*fmax, pi*fmax, 65)[:-1]
```

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

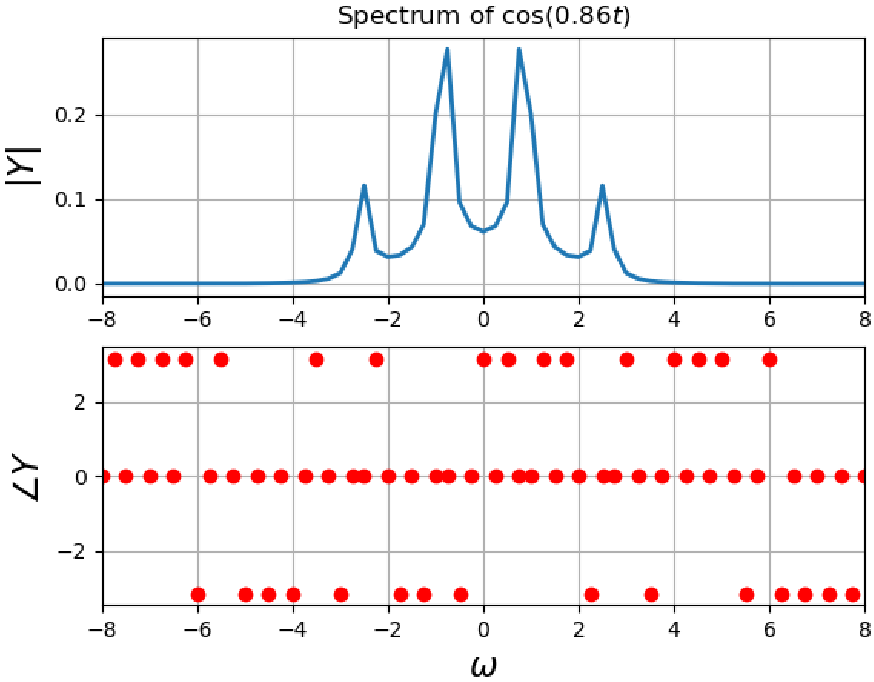
$$x[n] = 0.54 + 0.46cos(\frac{2\pi n}{N-1}) \tag{1}$$

We now multiply our signal with the hamming window and periodically



extend it.
The Python code is as follows:

```
 t = linspace(-pi, pi, 65)[:-1]
dt = t[1]-t[0]
fmax = 1/dt
y = t
y[0] = 0 # the sample corresponding to -tmax should be set zero
y = fftshift(y) # make y start with y(t = 0)
Y = fftshift(fft(y))/64.0
w = linspace(-pi*fmax, pi*fmax, 65)[:-1]

f = lambda x: sin(sqrt(2)*x)
wnd_func = lambda x: fftshift(0.54+0.46*cos(2*pi*x))
Y = get_dft(f, -pi, pi, 64, \
    r"Spectrum of $\sin\left(\sqrt{2}t\right)\times w(t)$", \
    True, 8, wnd_func)
```



Spectrum of $cos^3(0.86t)$

Without Hamming window

```
f = lambda x: (cos(0.86*x))**3
Y = get_dft(f, -4*pi, 4*pi, 256, \
    r"Spectrum of $\cos\left(0.86t\right)$", \
    True, 8, None)
```

With Hamming window

```
f = lambda x: (cos(0.86*x))**3
wnd_func = lambda x: fftshift(0.54+0.46*cos(2*pi*x))
Y = get_dft(f, -4*pi, 4*pi, 256, \
    r"Spectrum of $\cos\left(0.86t\right)$", \
    True, 8, wnd_func)
```

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

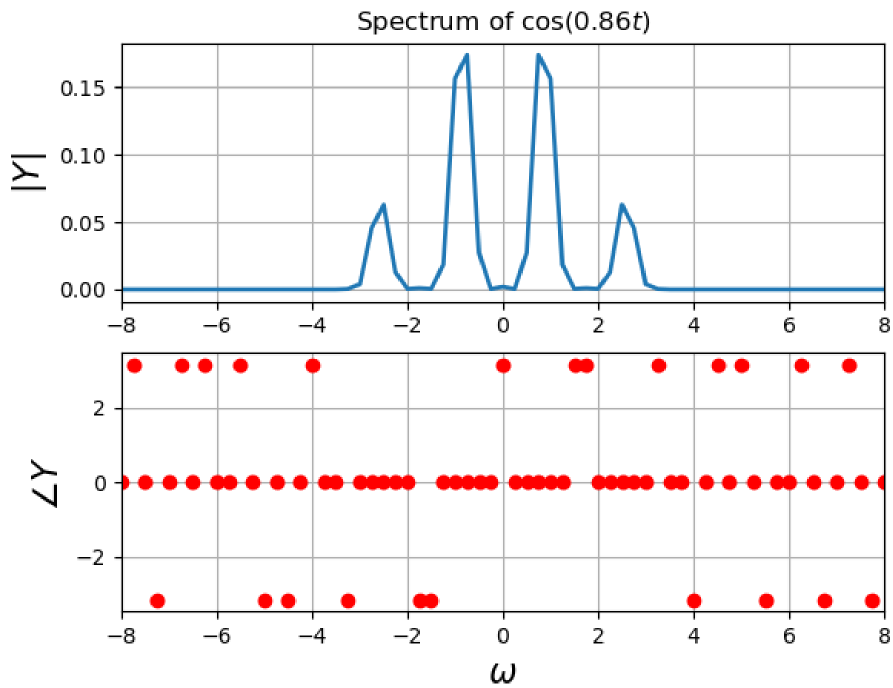


Figure 1: Magnitude and Phase Plot of sin(5t)

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

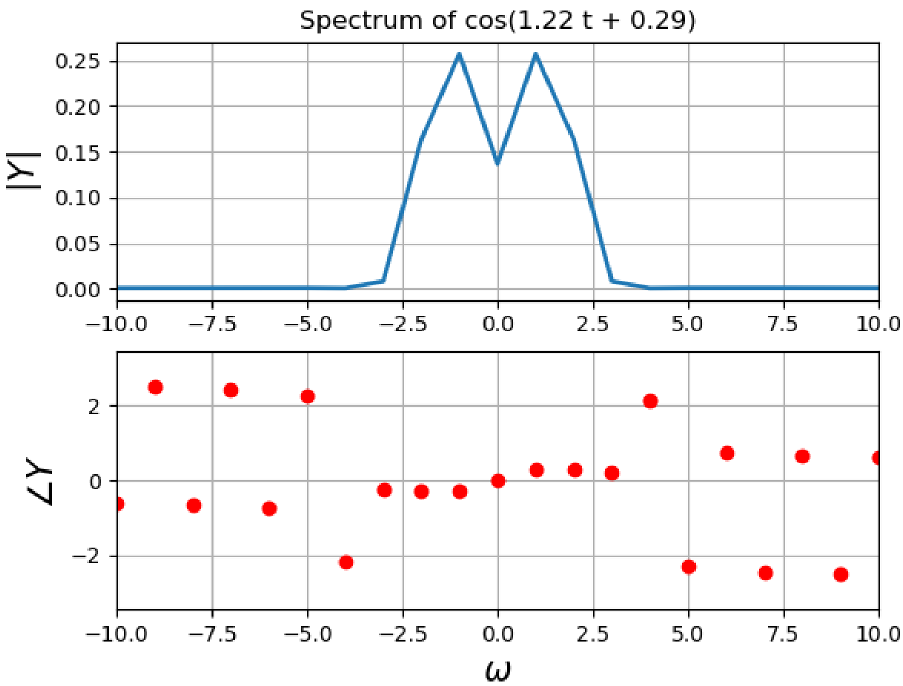
Estimate ω and δ for a signal $\cos(\omega t + \delta)$

Without noise

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

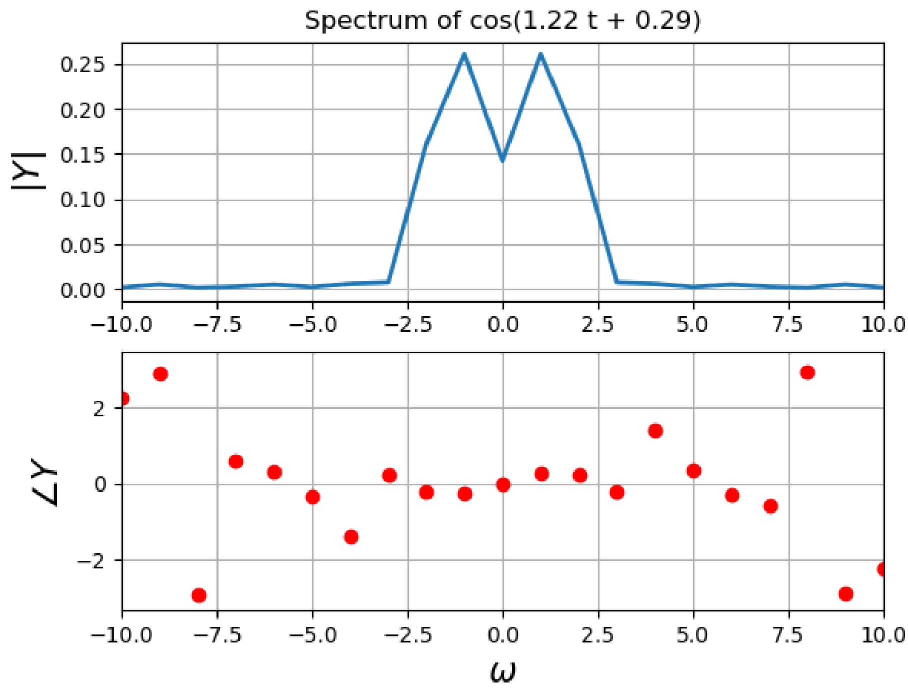
```
   Y_clean = get_dft(y_clean, -pi, pi, N, \
    "Spectrum of cos(%.2f t + %.2f)" % (w0, d), \
    True, 10, wnd_func, False)
Y_noisy = get_dft(y_noisy, -pi, pi, N, \
    "Spectrum of cos(%.2f t + %.2f)" % (w0, d), \
    True, 10, wnd_func, False)

# Set low magnitudes to 0 for better estimation
abs_Y_clean = abs(Y_clean)
abs_Y_clean[abs_Y_clean<1e-3] = 0

# Set low magnitudes to 0 for better estimation
abs_Y_noisy = abs(Y_noisy)
abs_Y_noisy[abs_Y_noisy<5e-2] = 0

# Get average frequency weighted by magnitude
w0_est_clean = average(w[N//2:N//2+5], weights=abs_Y_clean[N//2:N//2+5])
w0_est_noisy = average(w[N//2:N//2+5], weights=abs_Y_noisy[N//2:N//2+5])

# Get average phase weighted by magnitude
d_est_clean = (average(angle(Y_clean[N//2+1:]), weights=abs_Y_clean[N//2+1:]) \
                - average(angle(Y_clean[:N//2]), weights=abs_Y_clean[:N//2]))/2
d_est_noisy = (average(angle(Y_noisy[N//2+1:]), weights=abs_Y_noisy[N//2+1:]) \
                - average(angle(Y_noisy[:N//2]), weights=abs_Y_noisy[:N//2]))/2
```

We estimate omega by performing a Mean average of ω over the magnitude of $|Y(j\omega)|$. For delta we consider a widow on each half of ω (split into positive and negative values) and extract their mean slope. The intuition



behind this is that, a circular shift in the time domain of a sequence results in the linear phase of the spectra.

Actual w0, d = [1.22291617] [0.29296623]
Estimated w0 (clean, noisy): 1.0766767584640384 1.0326437769558694
Estimated d (clean, noisy): 0.29508367503323496 0.2600460947698232



With noise

We repeat the exact same process as question 3 but with noise added to the original signal.

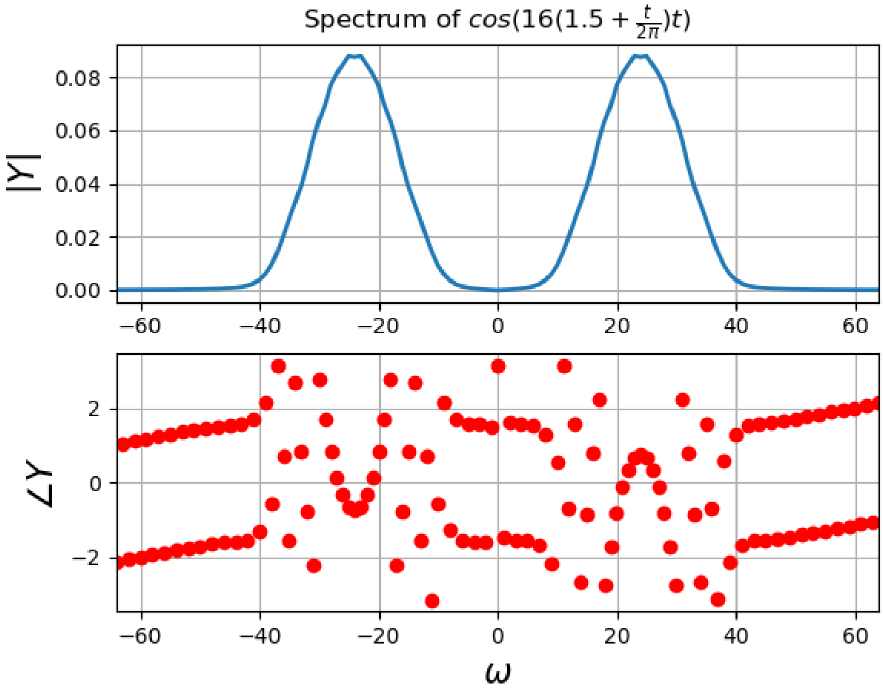
Spectrum of Chirped Signal

In this question 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

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

The FFT of the chirp is given by: We note that the frequency response is spread between 5-50 rad/s. A large section of this range apears due to Gibbs phenomenon. On windowing, only frequencies between 16 and 32 rad/s remain.

```
N = 1024
f = lambda x: cos(16*x*(1.5 + x/(2*pi)))
Y = get_dft(f, -pi, pi, N, \
```



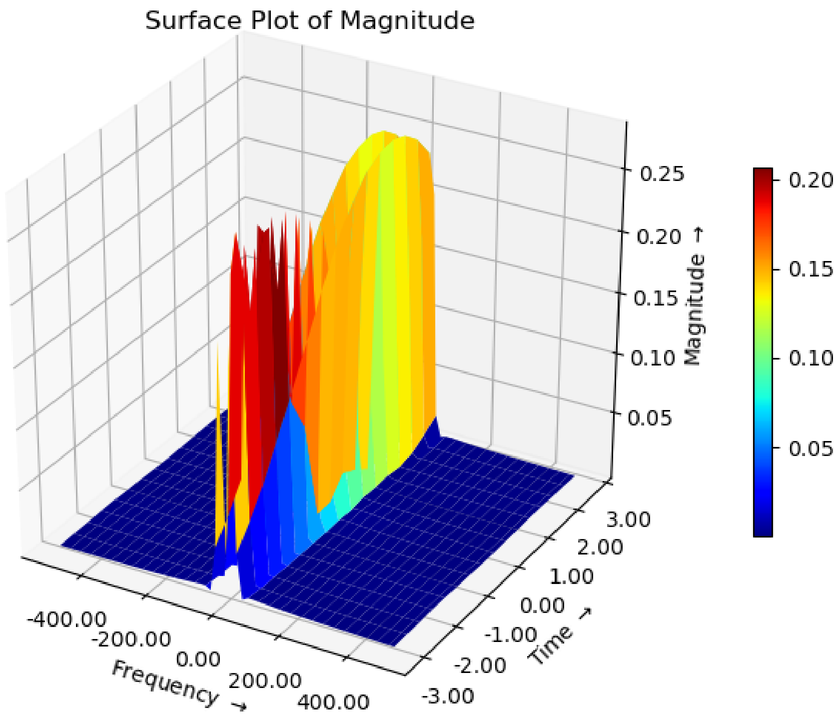

```
r"Spectrum of $cos(16(1.5+\frac{t}{2\pi})t)$", \
True, 64, wnd_func)
```

Time-Frequency Plot

For the same chirped signal, we 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 plot the array as a surface plot to show how the frequency of the signal varies with time. This is new. So far we worked either in time or in frequency. But this is a "time- frequency" plot, where we get localized DFTs and show how the spectrum evolves in time. We do this for both phase and magnitude. Let us explore their surface plots.

```
t = linspace(-pi, pi, N+1)[:-1]
dt = t[1] - t[0]
fmax = 1/dt
blocks = split(t, 16)        # split into 16 blocks, 1024/64 = 16
edges = [[blk[0], blk[63]] for blk in blocks]   # Get block edges
```

```
# Find DFT of each block
Y_arr = array([get_dft(f, edge[0], edge[1], 64, \
    r"Spectrum of $cos(16(1.5+\frac{t}{2\pi})t)$", \
    False, 64, wnd_func, "") for edge in edges])

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

t_g,w_g = meshgrid(t,w)

fig = figure()
ax = p3.Axes3D(fig)
surf = ax.plot_surface(w_g,t_g, abs(Y_arr.T), cmap=cm.jet)      # Reverse the co
# Formatting to truncate decimal numbers
ax.yaxis.set_major_formatter(FormatStrFormatter('%.2f'))
ax.xaxis.set_major_formatter(FormatStrFormatter('%.2f'))
ax.zaxis.set_major_formatter(FormatStrFormatter('%.2f'))
xlabel(r"Frequency $\rightarrow$")
ylabel(r"Time $\rightarrow$")
ax.set_zlabel(r"Magnitude $\rightarrow$")
ax.set_title("Surface Plot of Magnitude")
fig.colorbar(surf, shrink=0.5)
show()
```



Conclusion

In this assignment we have covered the requirement 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.

The general properties of a fourier spectra for a chirped signal are observable in the time avrying plots , ie..., 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.

The last question addresses the time varying spectra for a chirped signal, where we plot fourier spectra for different time slices of a signal. We noted the case of sparse number of slices and hence took more closely spaced slices.