Assignment 9:
Spectra of non-periodic Signals

Saurav Sachin Kale, EE19B141

May 12, 2021

1 Aim

- To analyse the magnitude and phase plots of the frequency response of non-periodic discrete time signals, and minimize the associated problems which arise from Gibbs phenomenon.
- To perform corrective filtering in order to better extract the frequencies in the non periodic signals by applying a Hamming Window.
- Extracting coefficients in a sinsoid from the frequency response.

2 Procedure

2.1 Analysing the frequency spectrum of $sin(\sqrt{2}t)$

We have the signal $sin(\sqrt{2}t)$. We sample this signal in an interval of $[-\pi, \pi)$ and obtain its DFT by using the `fft()` and `fftshift()` functions, with a small change.

- `y[0] = 0` is done because the function is odd, but the samples are not symmetric. Unsymmetric samples result in residual components in phase because of the following argument:

  $$y[0] = 0$$

  For $i \in \{1, 2...., \frac{N}{2} - 1\}$

  $$y[i] = -y[N-i]$$

  $$y[\frac{N}{2}] = sin(t_{\frac{N}{2}}) = sin(-t_{max})$$

  $$Y[k] = \sum_{n=0}^{N-1} y[n]exp(-j\frac{2\pi kn}{N})$$

  $$Y[k] = \sum_{n=0}^{\frac{N}{2}-1} y[n](exp(j\frac{2\pi kn}{N}) - exp(-j\frac{2\pi kn}{N})) + y[\frac{N}{2}]exp(j\pi k)$$

  $$Y[k] = -2j\sum_{n=0}^{\frac{N}{2}-1} y[n]sin(\frac{2\pi kn}{N}) + (-1)^k y[\frac{N}{2}]$$

  We can see that the DFT is not completely imaginary, even if we know CTFT will be purely imaginary, since it is a sinusoid. Hence we need to ensure $y[\frac{N}{2}] = 0$. That is the purpose behind `y[0] = 0`

With this we obtain the following frequency spectrum

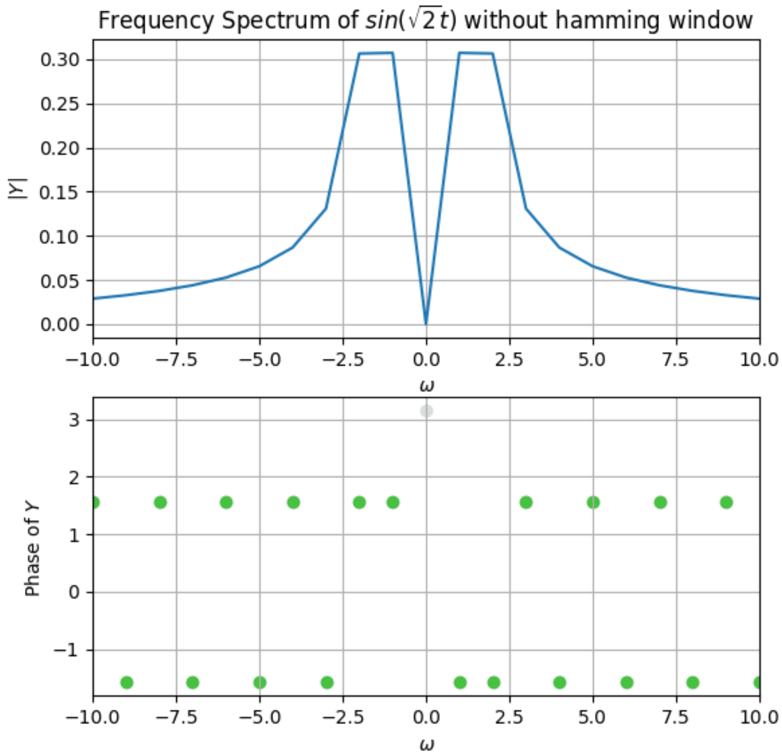

Figure 1: Frequency Spectrum of $sin(\sqrt{2}t)$ sampled 64 times in an interval of $[-\pi, \pi)$

This is not as expected. We should have got two peaks approximately at $\pm\sqrt{2}$. The reason behind this is that the DFT imposes periodicity in the sampled function despite it being dependent only on the samples. If we look at the region we have sampled, and construct a periodic function with it we get:

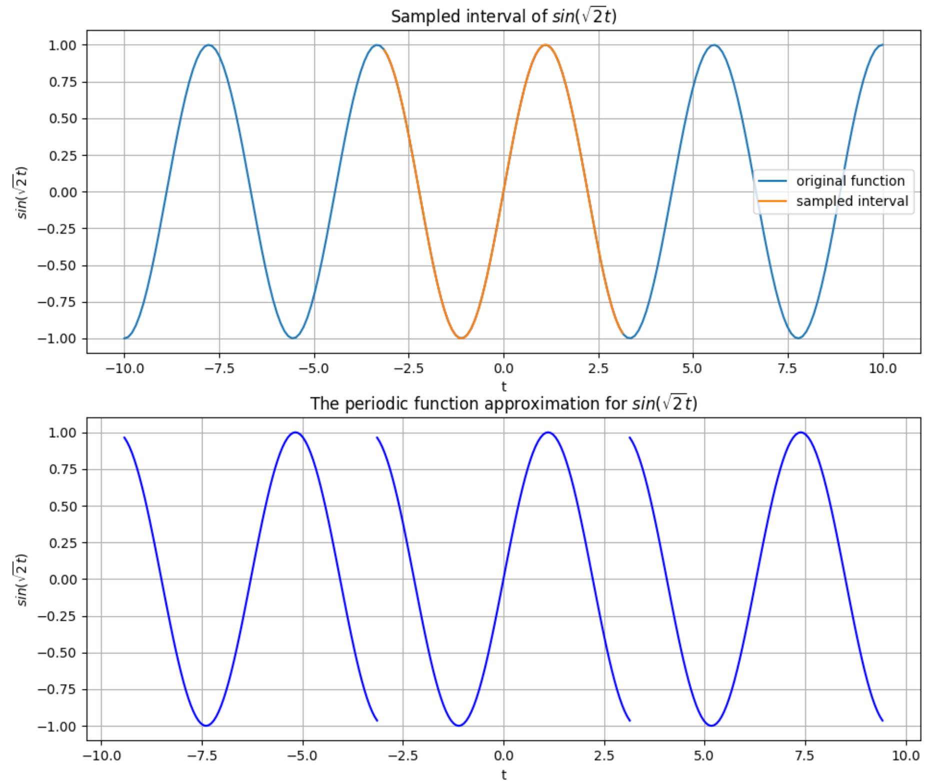

Figure 2: Periodic representation of $sin(\sqrt{2}t)$

Now we can see clearly why we are getting so much variation from the expected plot.

- We observe the discontinuities which happen between each section of 64 samples
- Discontinuities cause Gibbs Phenomenon when signal is reconstructed from transform domain
- This results in significant magnitudes even at higher frequencies

We can see the effect of this by plotting the Magnitude response of the unit Ramp function:

$$r(t) = t \text{ for } -\pi < t < \pi$$

Fourier Series of this ramp is:

$$f(t) = 2(\frac{sint}{1} - \frac{sin2t}{2} + \frac{sin3t}{3} - ...)$$

Frequency samples decay as $\frac{1}{\omega}$. That is a slope of -20dB/decade in logarithmic (base 10) scale. If we plot this we get:

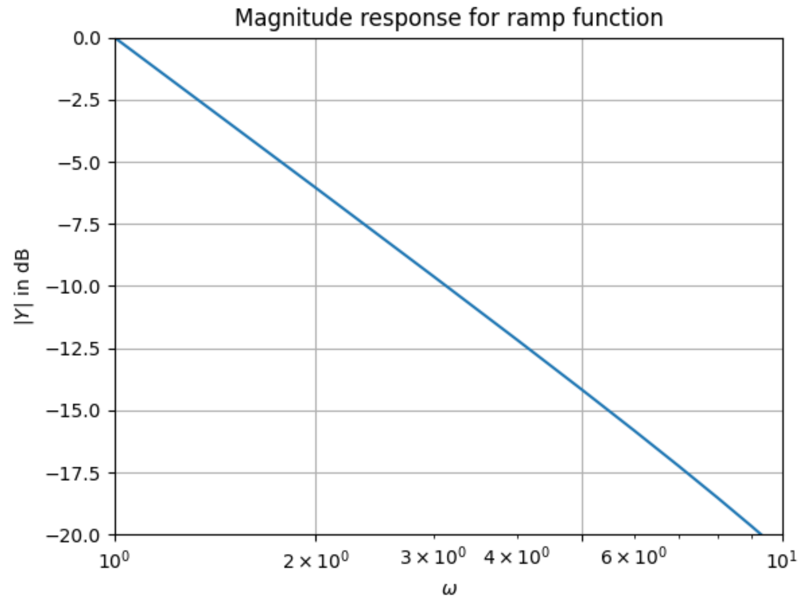

Figure 3: Magnitude Response for ramp function

It is as expected. The big jumps at each period can be interpreted as ramp functions if we join them linearly, and so we get a very slowly decaying response instead of sharp peaks.
In order to reduce this variation we multiply a Hamming Window function to our signal in order to minimize this discontinuity. The hamming window function we will be using is the following:

$$w[n] = \begin{cases} 0.54 + 0.46cos(\frac{2\pi n}{N-1}) & \text{for } |n| \leq \frac{N-1}{2} \\ 0 & \text{otherwise} \end{cases}$$

On multiplying we can see our periodic function now becomes:

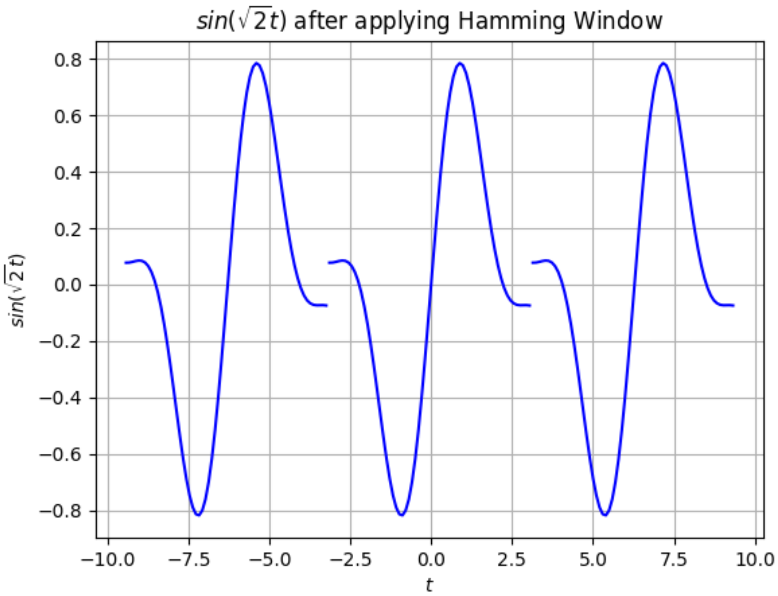

Figure 4: New periodic representation of $sin(\sqrt{2}t)$ after applying Hamming Window

We can see how the discontinuity is minimized. Plotting the DFT for $sin(\sqrt{2}nT_s)w[n]$ we obtain the following frequency spectrum

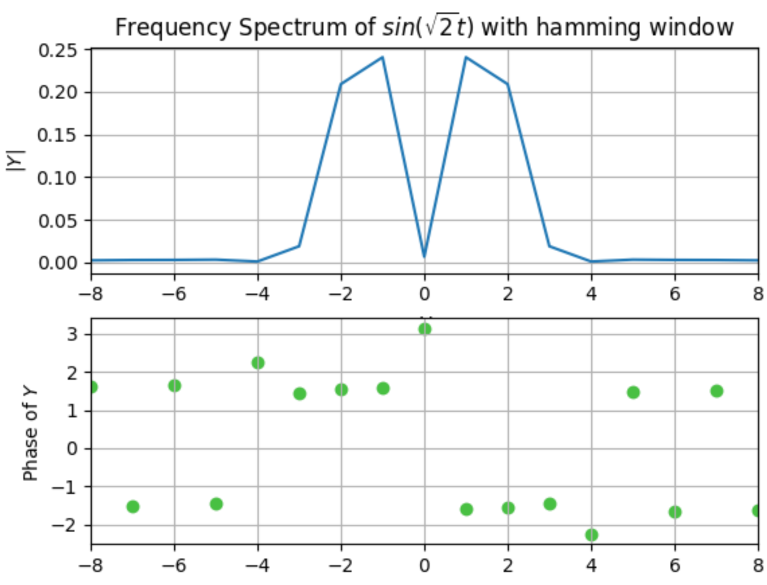

Figure 5: Frequency spectrum of $sin(\sqrt{2}nT_s)w[n]$ (Hamming windowed function)

Magnitude plot is much closer to our expectations now. But it is still too wide. We need better resolution to get better defined peaks. So we increase the time interval to $[-4\pi, 4\pi)$ and the samples to 256.

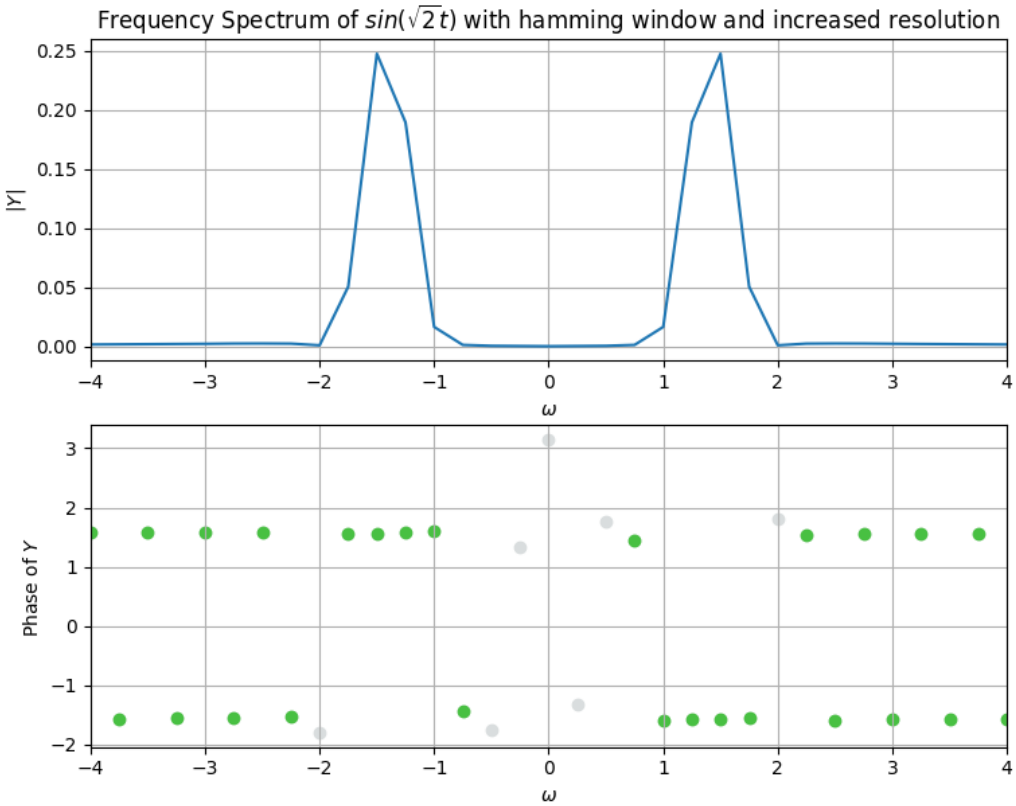

Figure 6: Better resolution plot for $sin(\sqrt{2}t)$ with Hamming window

This is much better than before. Now we have well defined peaks in the region approximately between 1 and 2 which is what we expect for $sin(\sqrt{2}t)$, because

$$sin(\sqrt{2}t) = \frac{1}{2j}e^{j\sqrt{2}t} - \frac{1}{2j}e^{-j\sqrt{2}t} = -\frac{j}{2}e^{j\sqrt{2}t} + \frac{j}{2}e^{-j\sqrt{2}t}$$

We can verify that the phase is also matching. The magnitude will not match because we have multiplied the Hamming window function which imposes its own magnitude. The peak is not for a single value of ω though. It is a wider peak across multiple values of ω. That is due to the fact that whenever the delta function appears in the transform of $sin(\sqrt{2}t)$, it convolves with the transform of $w(t)$ and result is the transform of $w(t)$ appears at that spot.

2.2 Analysing the frequency spectrum of $cos^3(0.86t)$

We now have the signal $cos^3(0.86t)$, and we perform the same procedure on this signal and plot the frequency spectrum.

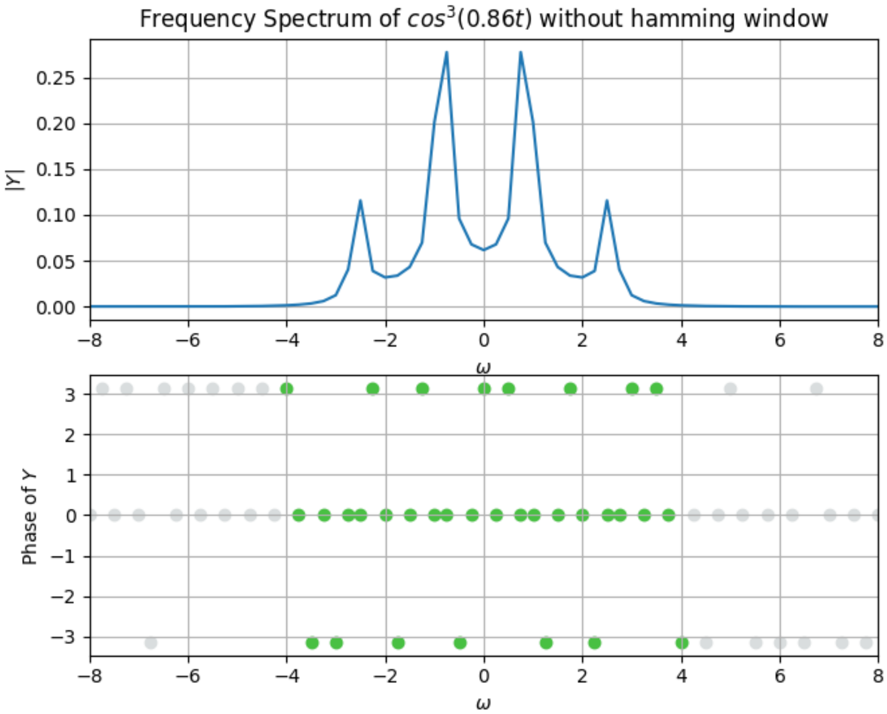

Figure 7: Frequency spectrum of $cos^3(0.86t)$ without Hamming window

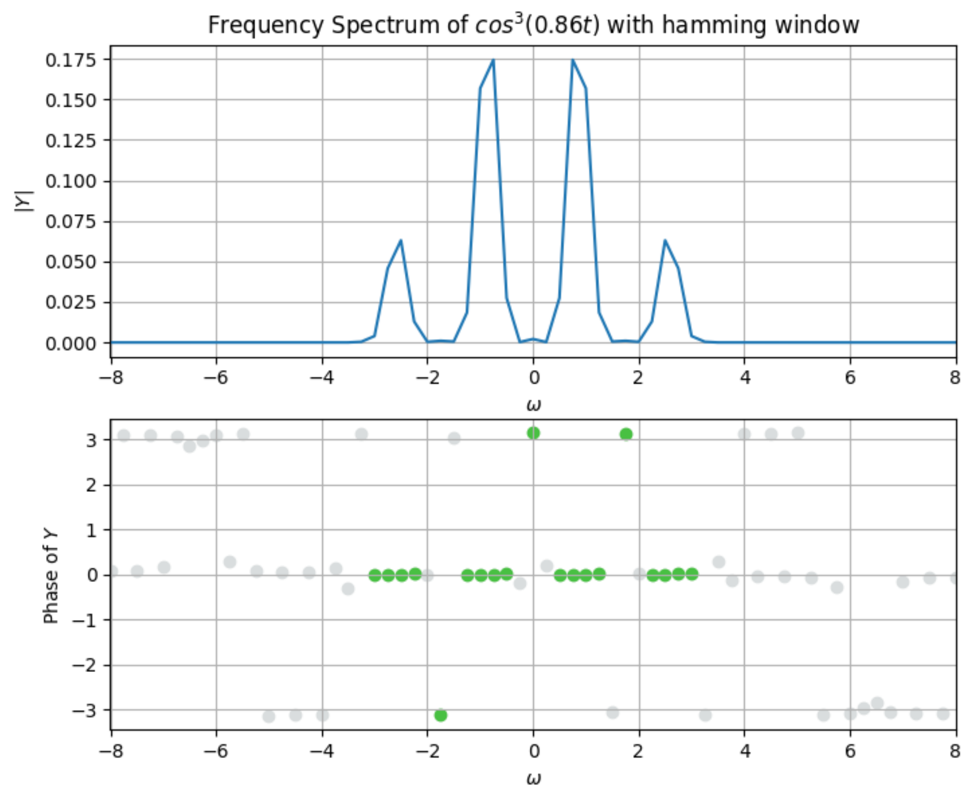

Figure 8: Frequency spectrum of $cos^3(0.86t)$ with Hamming window

Which is as expected because:

$$cos(3(0.86)t) = 4cos^3(0.86t) - 3cos(0.86t)$$

$$cos^3(0.86t) = \frac{1}{4}cos(3(0.86)t) + \frac{3}{4}cos(0.86t)$$

$$cos^3(0.86t) = \frac{1}{8}e^{j3(0.86)t} + \frac{1}{8}e^{-j3(0.86)t} + \frac{3}{8}e^{-j0.86t} + \frac{3}{8}e^{j0.86t}$$

We can verify that the peaks are approximately at the correct spot, and the phase also matches.

2.3 Frequency spectrum of $cos(\omega_o t + \delta)$

We have the signal $cos(\omega_o t + \delta)$ for any arbitrary ω_o such that $0.5 < \omega_o < 1.5$ and δ. Here we chose $\omega_o = 0.8$ and $\delta = 0.78$.

$$cos(\omega_o t + \delta) = \frac{1}{2}e^{j\delta}e^{j\omega_o t} + \frac{1}{2}e^{-j\delta}e^{-j\omega_o t}$$

Therefore, we expect that the phase of the peaks will be equal to δ, and the peaks would be located at $\pm\omega_o$. We use the same procedure to plot the frequency spectrum.

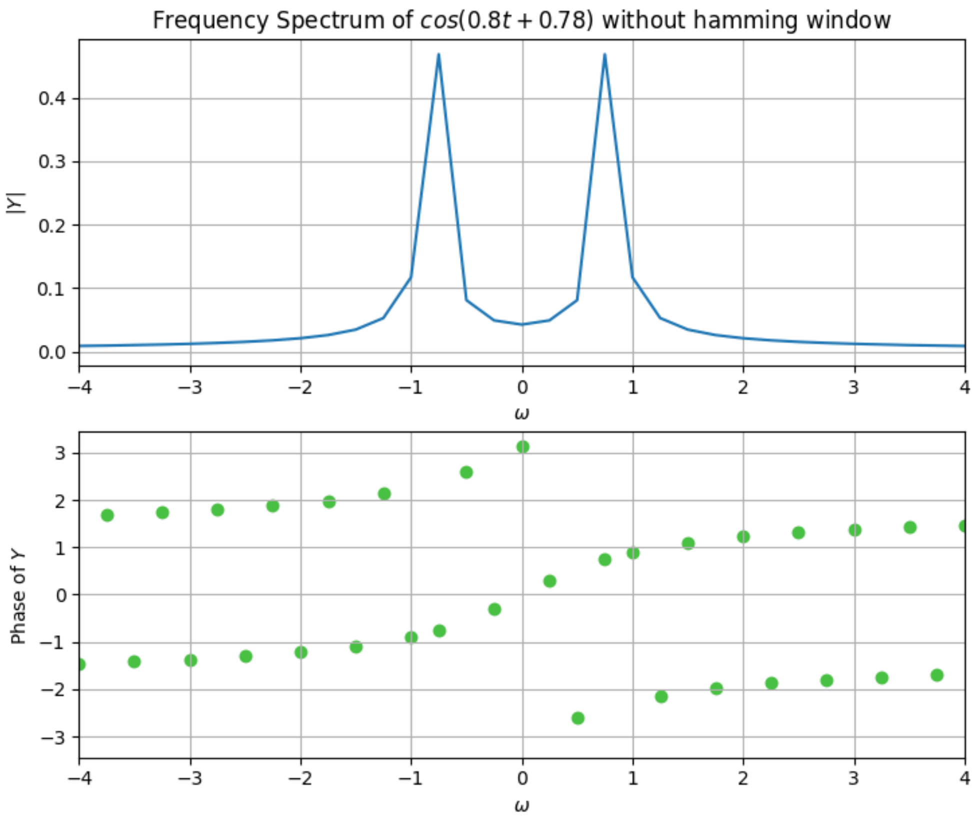

Figure 9: Frequency spectrum of $cos(\omega_o t + \delta)$ without Hamming Window

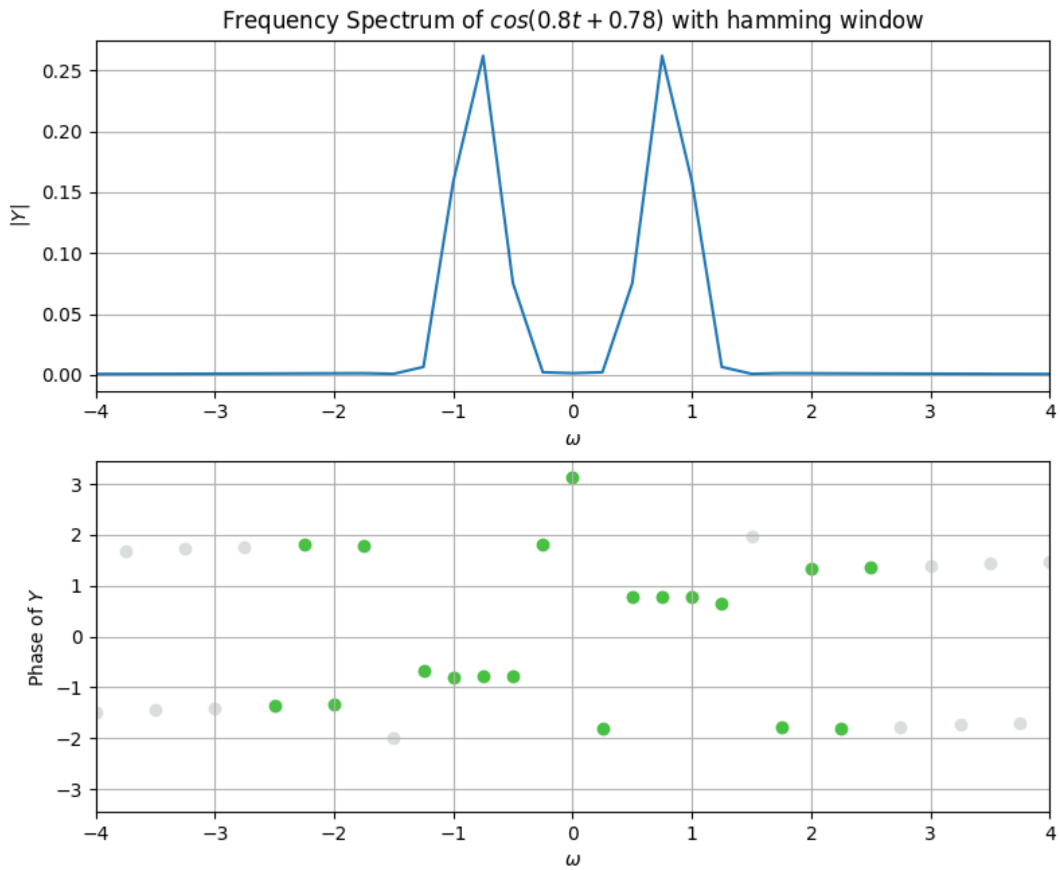

Figure 10: Frequency spectrum of $cos(\omega_o t + \delta)$ with Hamming Window

In addition to this, we can now use this data to estimate the value of ω_o using a wieghted average. We narrow down the location of the peaks by first getting those values of ω for which magnitude is sufficiently high. We then perform a wieghted average with the magnitude squared being the wieght for all the frequencies under the peak.

```
# estimate the Wo and Delta from the frequency response data
def estimateWoAndDelta(w, mag, phase):
    # find the location near the peaks
    actualMag = p.where(mag > 0.2)
    #print(w[actualMag])
    # take weighted average across the peaks
    wWeightedAvg = p.sum((mag[actualMag]**2) * abs(w[actualMag]))/p.sum(mag[actualMag]
    # take simple average of absolute value of the phases at the peaks
        # (got better results from simple avg rather than weighted here)
    phaseEstimate = p.mean(abs(phase[actualMag]))
    # print
    print("Estimate␣for␣Wo:␣", wWeightedAvg)
    print("Estimate␣for␣delta:␣", phaseEstimate)
```

For the phase, we take a simple average of absolute values all the values in the peak regions. We do this because since coefficients are real, the negative phases will simply represent the conjugates of the positive phases. So we avoid the zero mean by taking absolute value of phases. We chose to take simple average because it produced better results than wieghted average. We obtain the following output for $cos(0.8t+0.78)$:

```
Q3
No hamming
Estimate for Wo:  0.75
Estimate for delta:  0.7466104540943095
Hamming
Estimate for Wo:  0.75
Estimate for delta:  0.7749261729797845
```

Which is very close to the actual values.

2.4 Frequency spectrum of $cos(\omega_o t + \delta)$ with added white gaussian noise

We have the signal $cos(\omega_o t + \delta)$ with added white gaussian noise. We obtain that noise by using the randn() function and the amplitude 0.1 as follows

```
def noisyCosW0tPlusDelta(x):
        return p.cos(0.8*x + 0.78) + 0.1*p.randn(len(x))
```

We now perform the same analysis for this

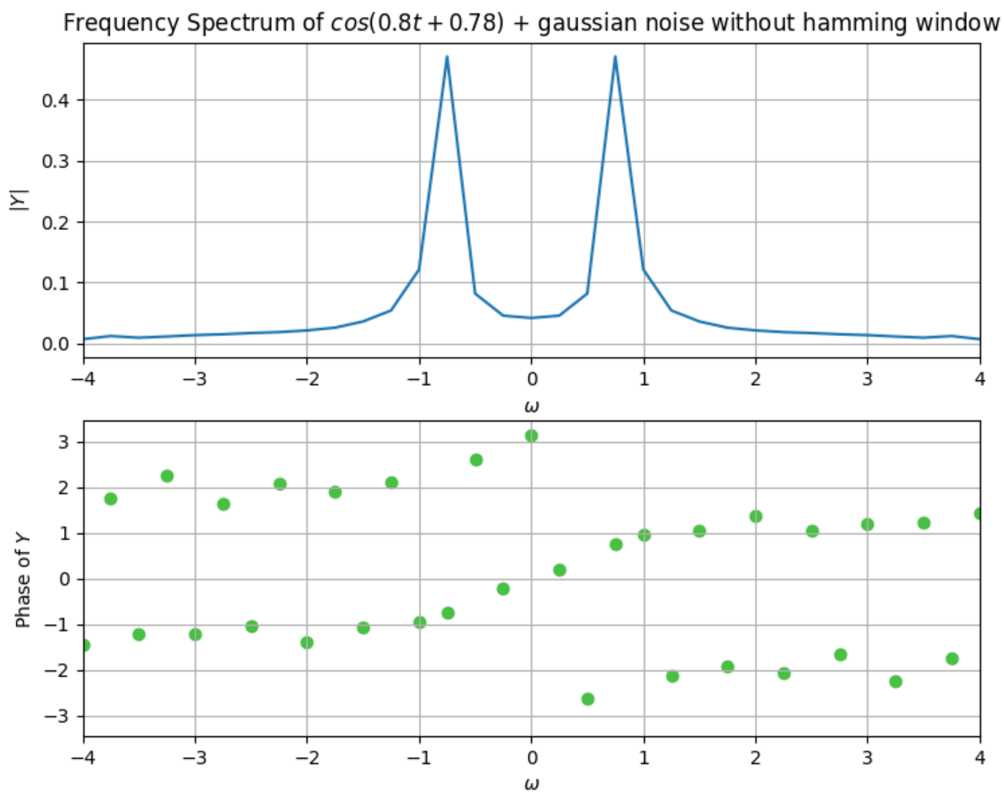

Figure 11: Frequency spectrum of $cos(\omega_o t + \delta)$ with noise, without Hamming Window

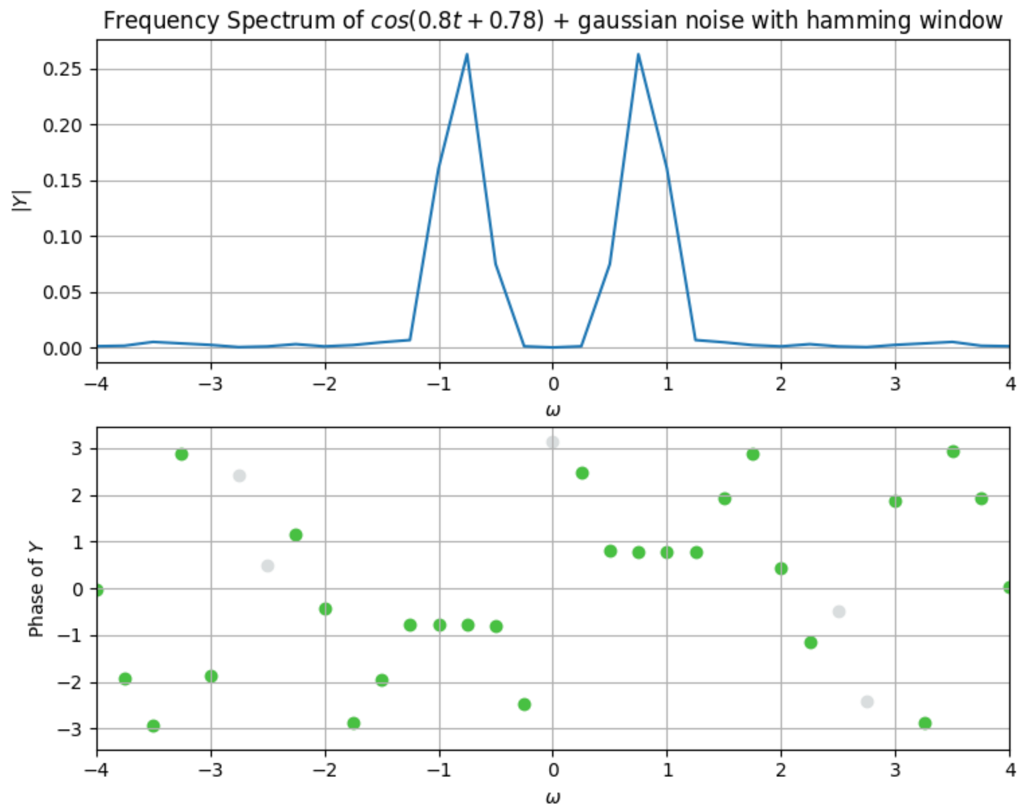

Figure 12: Frequency spectrum of $cos(\omega_o t + \delta)$ with noise, with Hamming Window

We see that there isn't really much difference in the frequency spectra with or without the noise. Only there is very slight distortion. The estimation data is again quite similar:

```
Q4
No hamming
Estimate for Wo:  0.7500000000000001
Estimate for delta:  0.7428681726674076
Hamming
Estimate for Wo:  0.75
Estimate for delta:  0.7774617919261039
```

We can see that it is quite close to actual values.

2.5 Frequency spectrum of a Chirped signal

We have the signal $cos(16(1.5 + \frac{t}{2\pi})t)$. This is known as a chirped signal, and its frequency changes. We now plot its frequency response for an interval of $[-\pi, \pi)$ and 1024 samples.

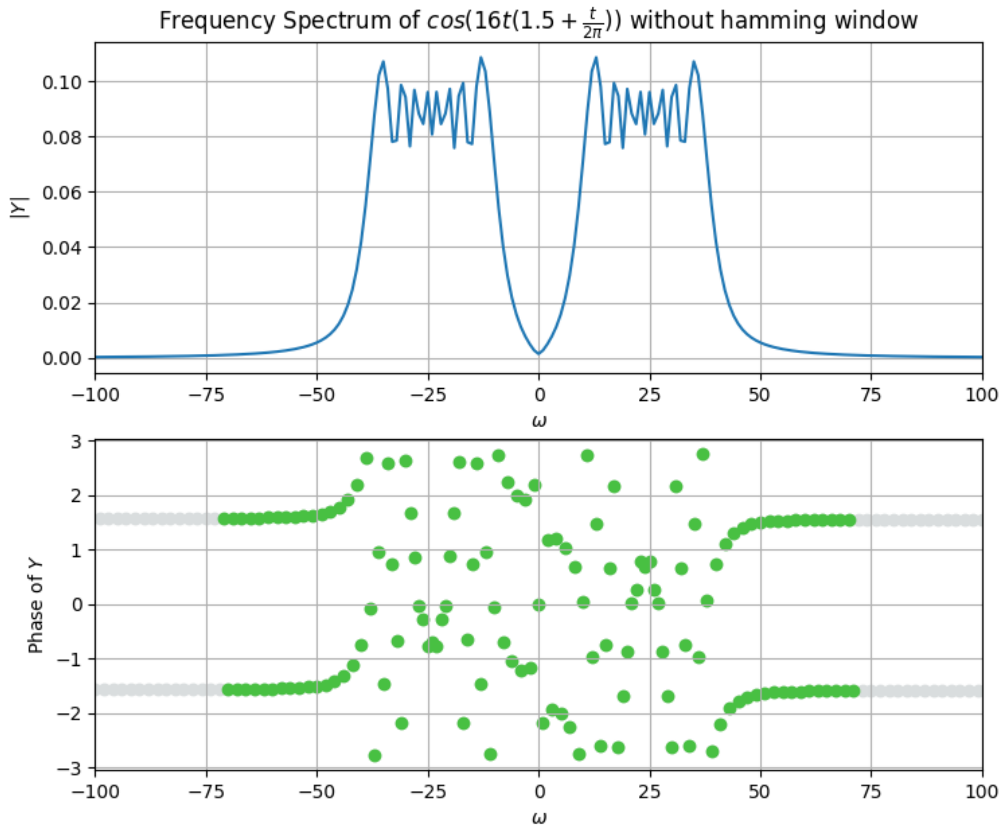

Figure 13: Frequency spectrum of $cos(16(1.5+\frac{t}{2\pi})t)$ without Hamming Window

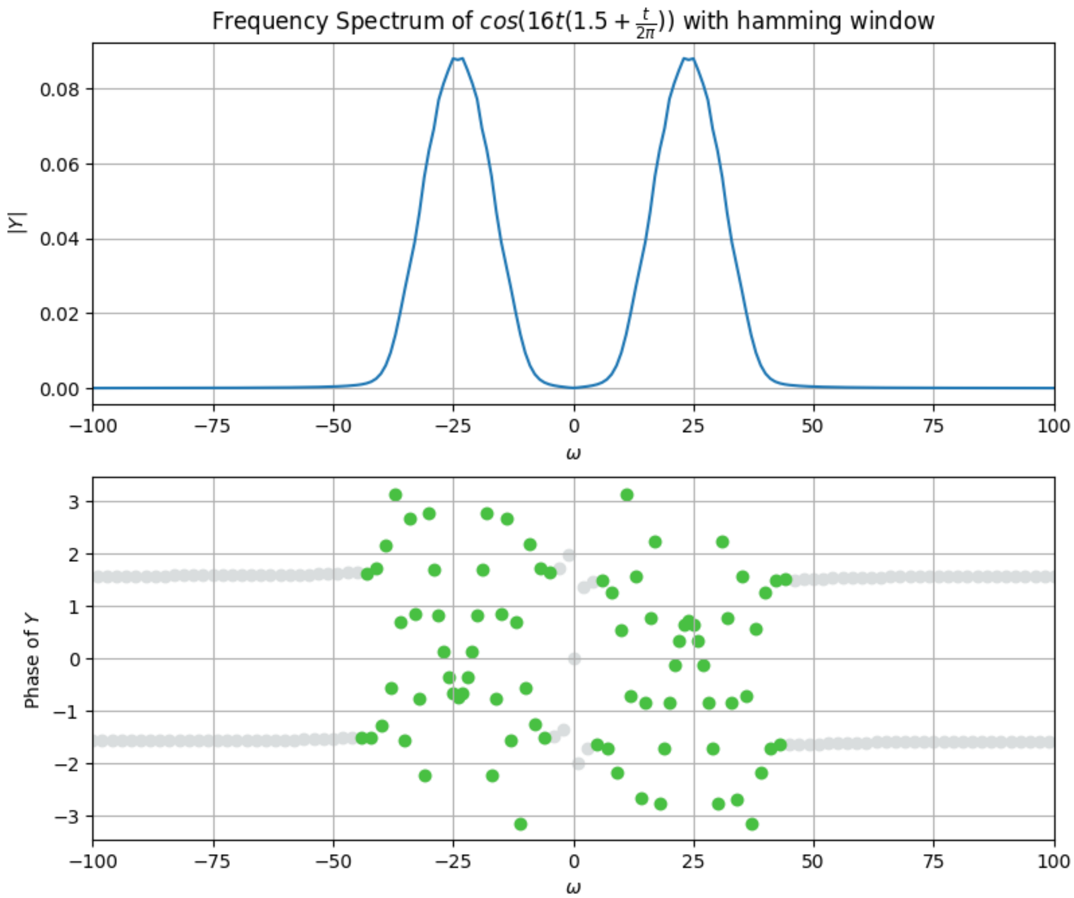

Figure 14: Frequency spectrum of $cos(16(1.5 + \frac{t}{2\pi})t)$ with Hamming Window

This is as expected because the frequency response has frequencies from 16 to 32 rad/s. The hamming window further confines the frequencies in that range.

2.6 Time Frequency Plot of a Chirped signal

We now plot the Time Frequency plot of the signal $cos(16(1.5+\frac{t}{2\pi})t)$. We break the interval of 1024 samples from $[-\pi, \pi)$ into 16 contiguous pieces of 64 samples each. We then find the DFT of each piece. Since each piece corresponds to a different time interval, we can plot magnitude as a function of time and frequency. We do this in the following way:

```
#define the full interval in 1024 samples
t_full = p.linspace(-p.pi, p.pi, 1025)[:-1]
# split it into 16x64
t_broken = p.reshape(t_full, (16, 64))
# mag and phase arrays for each interval
mags = []
phases = []
```

```
# define the w, sampling frequency is still same, despite the breaking up in 16 interv
w = p.linspace(-512, 512, 65)[:-1]
# for each interval we find FFT and append it to mags and phases
for t in t_broken:
    y = chirp(t)
    y[0] = 0
    y = p.fftshift(y)
    Y = p.fftshift(p.fft(y))/64

    mags.append(abs(Y))
    phases.append(p.angle(Y))

mags = p.array(mags)
phases = p.array(phases)
```

We now plot magnitude and phase surfaces.

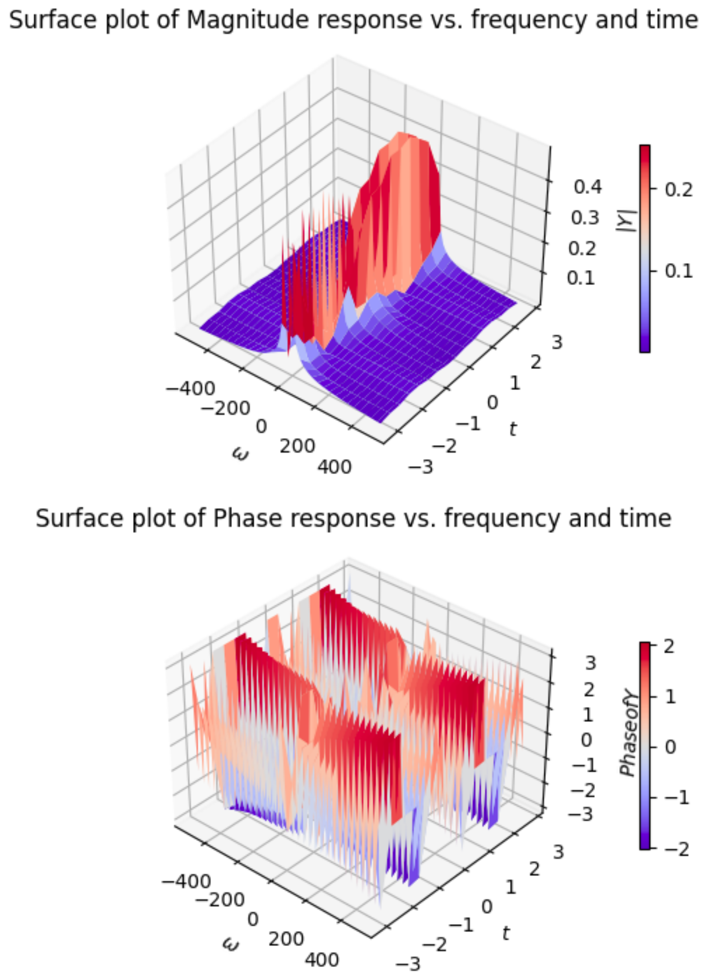

Figure 15: Frequency Time surface of $cos(16(1.5 + \frac{t}{2\pi})t)$ without Hamming Window

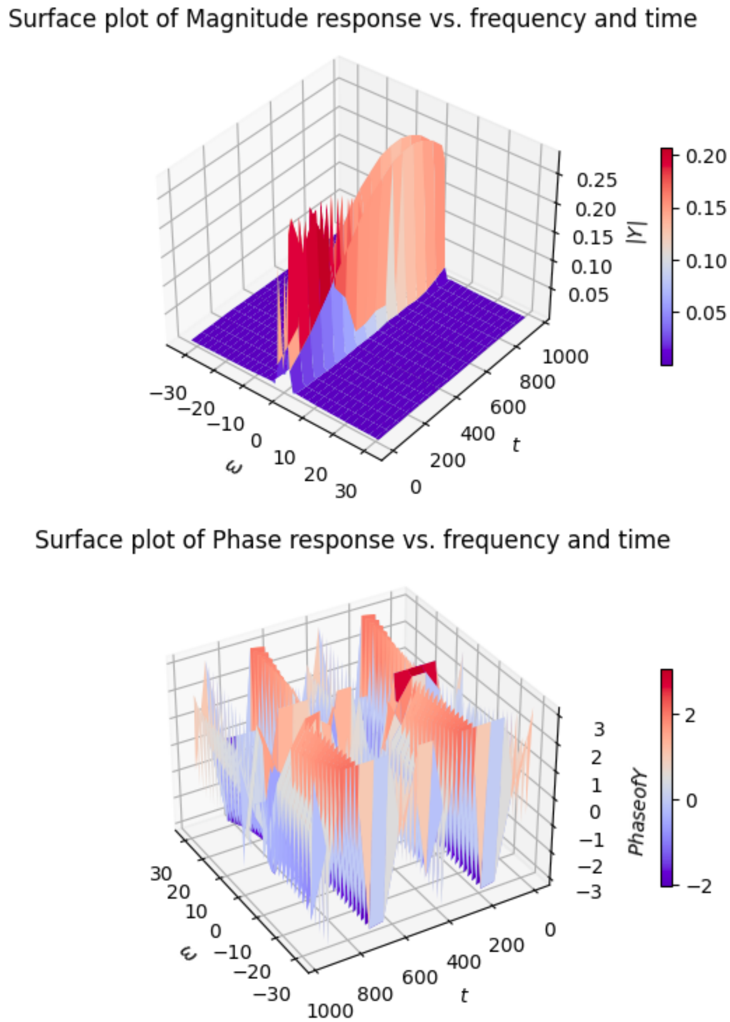

Figure 16: Frequency Time surface of $cos(16(1.5 + \frac{t}{2\pi})t)$ with Hamming Window

This is how the DFT varies with time. The gap between the peaks increases with time. The hamming window narrows down the peaks a little more.

3 Conclusions

- The spectra of several non periodic signals were plotted
- The effect of discontinuities is clearly seen, understood, and minimized by applying a hamming window. This results in more well defined peaks

- Extracting coefficients using the frequency spectrum of a sinusoid was done using averaging methods on the DFT data to sufficient accuracy.
- The Time Frequency surface plot of a chirped signal was analysed to understand the time variation of the DFT