EE2703 : Applied Programming Lab
Assignment 7
Report

Kesava Aruna Prakash R L
EE20B061

April 13, 2022

Abstract

This week's assignment focuses on analysing signals using the Digital Fourier Transform(**DFT**) using `pylab.fft` module. The DFT is also calculated for different windows to find the closest fit to the actual Continuous Time Fourier Transform(**CTFT**) of a Gaussian function.

Spectrum of Different Signals

For finding and plotting the spectrum, a generic function is used and it has certain arguments like sampling rate, and a few arguments to define plots like title, limiting values of x-axis etc.
Function definition:

```
def fft_modified(function,sampling_rate,x_limiter,tolerance,title,phasepoints=1)

    N = len(function)
    T = N/(sampling_rate)

    w_n = np. linspace(-sampling_rate*np.pi, sampling_rate*np.pi, N+1)
    w_n = w_n[:-1]

    f_n = pl.fftshift(pl.fft(function))/(len(function))

    # Plotting Magnitude and Phase Plots of the DFT

    pt.plot(w_n,np.abs(f_n),'r')
    pt.xlim(-x_limiter,x_limiter)
    pt.xlabel("Frequency")
    pt.ylabel("Magnitude of DFT")
    pt.title(f"DFT for {title}")
    pt.show()
    pt.plot(w_n,np.angle(f_n),'wo')

    # Finding the phase points at relevant points
    ii = np.where(abs(f_n)>tolerance)
    pt.plot(w_n[ii],np.angle(f_n[ii]),'go')
    if phasepoints == 1:
        for i, j in zip(w_n[ii], np.angle(f_n[ii])):
            if abs(j)< 1e-6:
                j=0
```

```
            pt.text(i, j+0.25, '({}, {})'.format(i, j))
    pt.xlabel("Frequency")
    pt.ylabel("Phase of DFT")
    pt.title(f"DFT for {title}")
    pt.xlim(-x_limiter,x_limiter)
    pt.show()
    if phasepoints == 2:
        print(f"Relevant Points in the Phase Plot for {title}")
        for i in range(len(ii)):
            print(w_n[ii[i]],np.angle(f_n[ii[i]]))
```

Example : 1 Spectrum of $sin(5t)$

The spectrum of the signal $sin(5t)$ has two peaks, one at 5 rad/s and the other at -5 rad/s. This can be visualised by writing $sin(5t)$ in the exponential form.

$$sin(5t) = (\frac{e^{5t}}{2j} - \frac{e^{-5t}}{2j})$$

We took a sampling rate of 128 points for t in range $[0, 2\pi]$. So sampling frequency is $\frac{128}{2\pi}$.

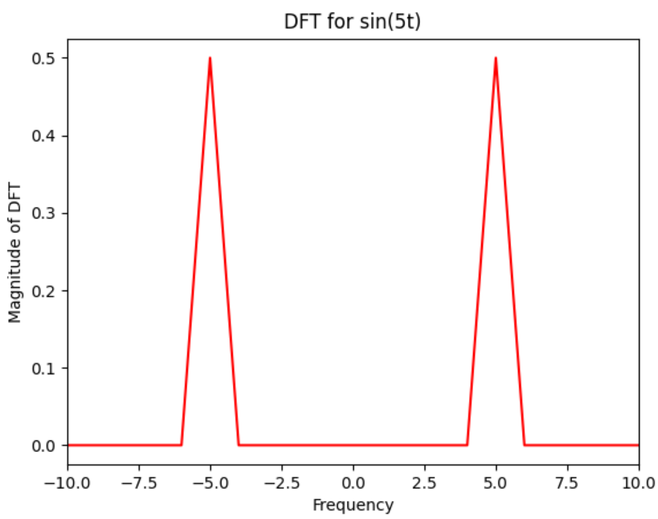

Figure 1: Spectrum of $sin(5t)$: Magnitude

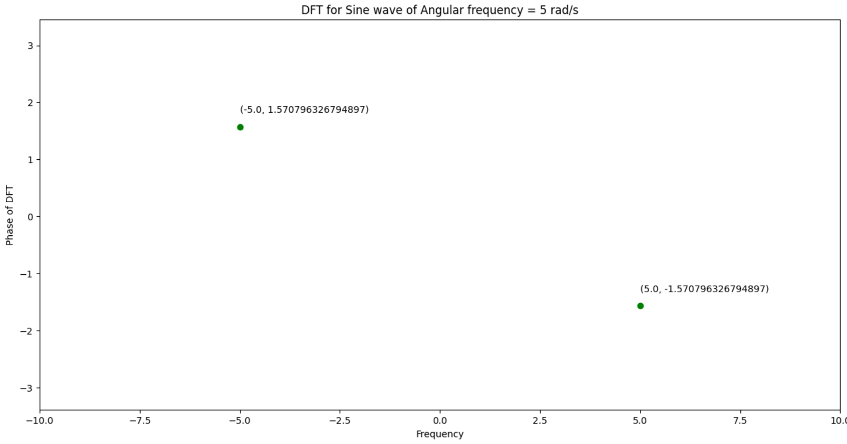

Figure 2: Spectrum of $sin(5t)$: Phase

Example : 2 Spectrum of $(1 + 0.1cos(t))cos(10t)$

The time frame of the sample is increased so that the individual peaks can be observed. Expected phase plot is obtained.

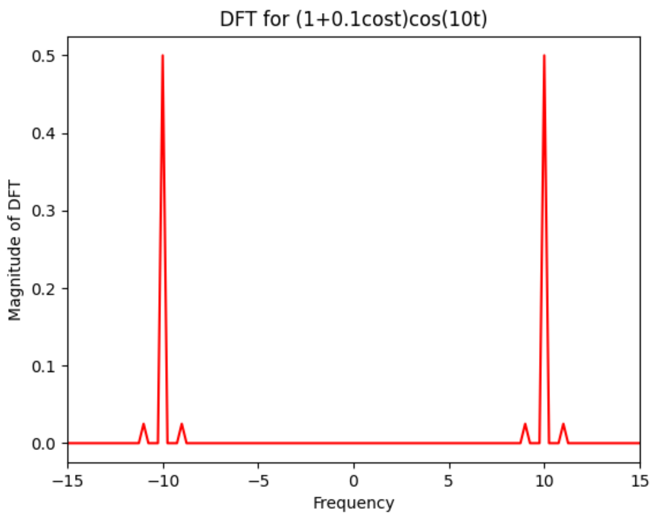

Figure 3: Spectrum of $(1 + 0.1cos(t))cos(10t)$: Magnitude

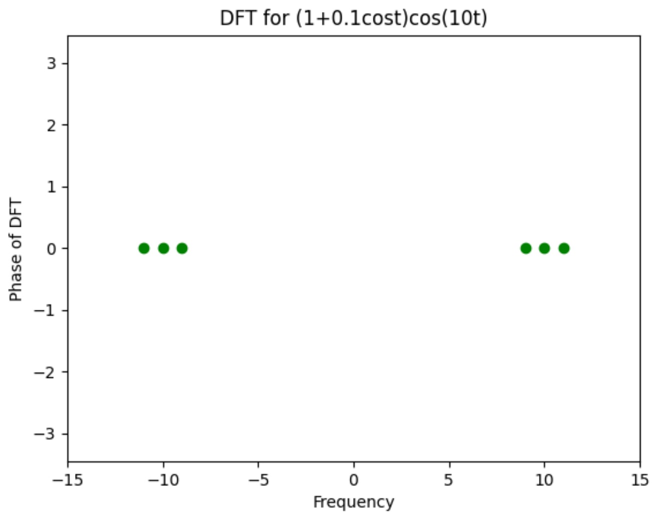

Figure 4: Spectrum of $(1 + 0.1cos(t))cos(10t)$: Phase

Spectrum of $sin^3(t)$

This signal can be expressed as a sum of sine waves using this identity:
$\sin^3(t) = \frac{3}{4}\sin(t) - \frac{1}{4}\sin(3t)$
It is expected to have 4 peaks at frequencies ± 1 rad/s and ± 3 rad/s, and phases similar to that expected from a sum of sinusoids.

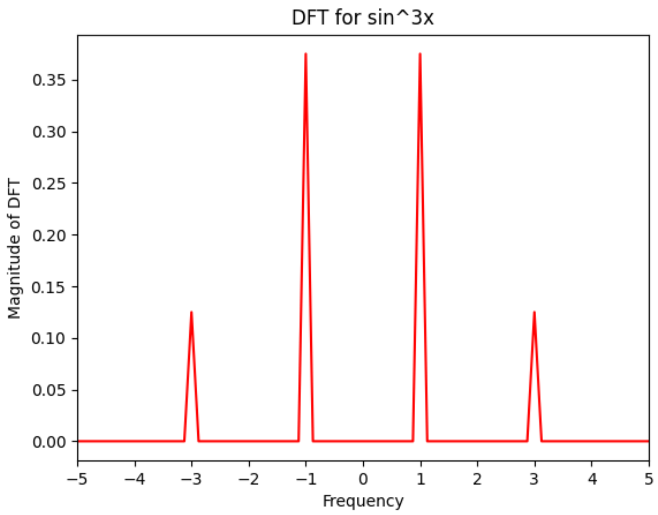

Figure 5: Spectrum of $f(t) = sin^3(t)$: Magnitude

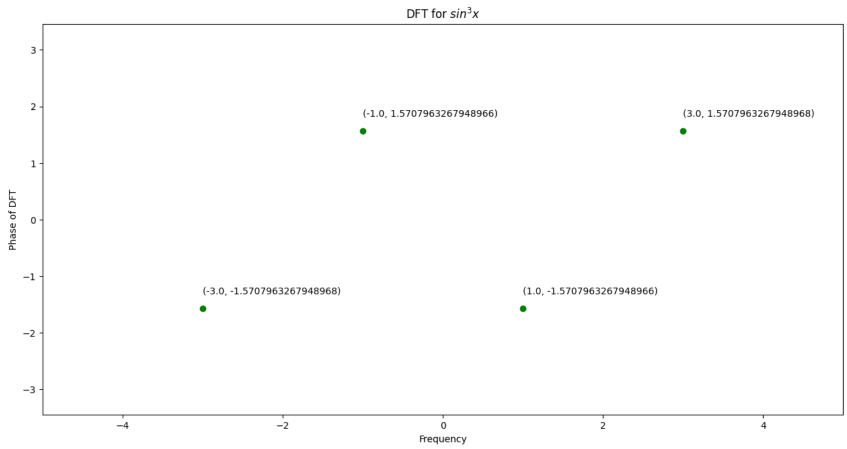

Figure 6: Spectrum of $f(t) = sin^3(t)$: Phase

Spectrum of $cos^3(t)$

This signal can be expressed as a sum of cosine waves using this identity:
$\sin^3(t) = \frac{3}{4}\cos(t) + \frac{1}{4}\cos(3t)$
It is expected to have 4 peaks at frequencies ± 1 rad/s and ± 3 rad/s, and phase to be zero at the peaks.

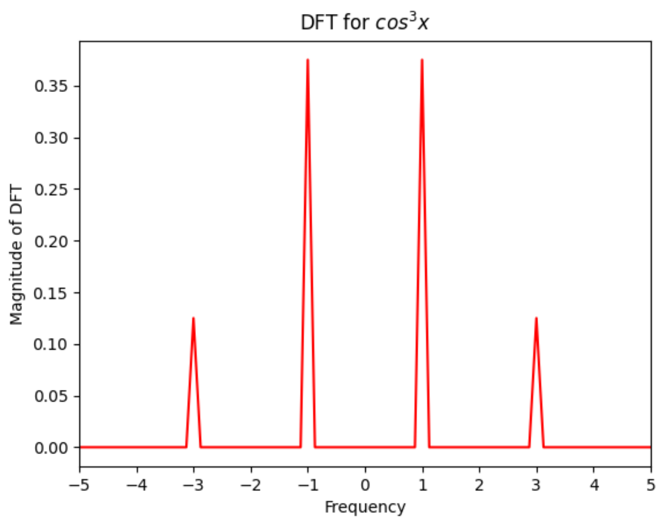

Figure 7: Spectrum of $f(t) = cos^3(t)$: Magnitude

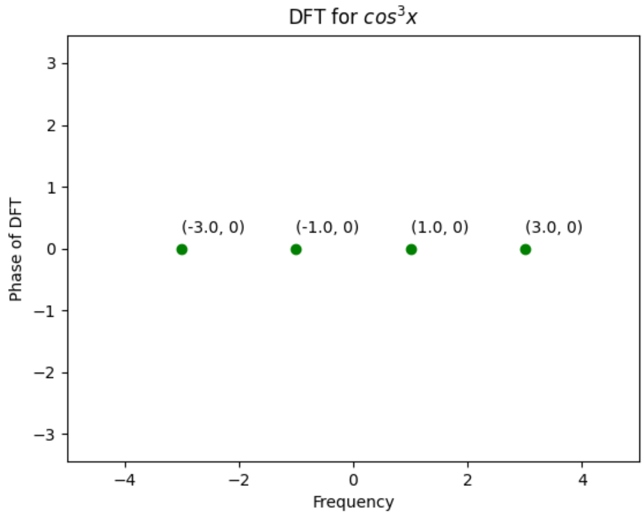

Figure 8: Spectrum of $f(t) = cos^3(t)$: Phase

Spectrum of $cos(20t + 5cos(5t))$

The signal is a frequency modulated signal. The number of peaks has clearly increased. The energy in the side bands is comparable to that of the main signal.

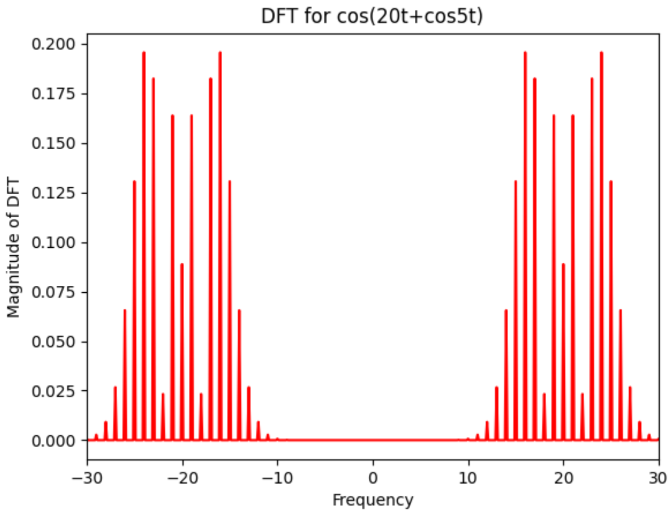

Figure 9: Spectrum of $cos(20t + 5cos(5t))$: Magnitude

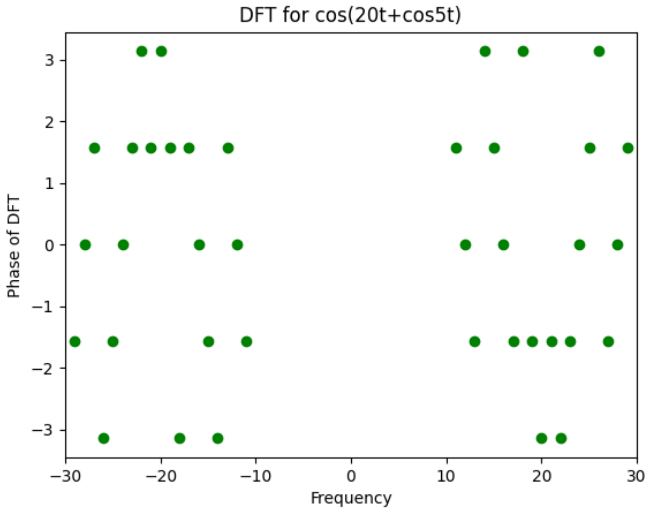

Figure 10: Spectrum of $cos(20t + 5cos(5t))$: Phase

Continuous Time Fourier Transform(CTFT) of Gaussian Functions

The Fourier transform of a signal $x(t)$ is defined as follows:

$$X(\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt$$

We can approximate this by the Fourier transform of the windowed version of the signal $x(t)$, with a sufficiently large window as Gaussian curves tend to 0 for large values of t . Let the window be of size T. We get:

$$X(\omega) \approx \frac{1}{2\pi}\int_{-T/2}^{T/2} x(t)e^{-j\omega t}dt$$

The expression for the given Gaussian is :

$$x(t) = e^{\frac{-t^2}{2}}$$

The **CTFT** is given by:

$$X(j\omega) = \frac{1}{\sqrt{2\pi}}e^{\frac{-\omega^2}{2}}$$

By increasing the time frame T, we can find suitable approximation for the **CTFT** of the Gaussian function with some tolerance(less than 1e-6). We are starting with $T = 2\pi$ and increasing the window to twice the initial size. Function for executing the above:

```
def gauss(t):
    return np.exp(-0.5*(t**2))

# Continuous time Fourier transfrom of exp(-t^2/2) is 1/(2pi)^0.5 * exp(-w^2/2)

def actual_ft(t):
    return np.exp(-(t**2)/2)/np.sqrt(2*np.pi)

def estimated_dft(tolerance, samples, x_limit):
    T = 2*np.pi
    N = samples
    error = 1

    while error > tolerance:
        #print(error)
        tn = np.linspace( -T/2, T/2, N+1)
        tn = tn[:-1]

        wn = np. linspace(-N*np.pi/T, N*np.pi/T, N+1)
        wn = wn[:-1]

        fn = pl.fftshift(pl.fft(gauss(tn)))/(N*2*np.pi/T)

        error = np.sum(np.abs(np.abs(fn)-np.abs(actual_ft(wn))))

        T = T*2
        N = N*2
    print(f"True error : {np.sum(np.abs(np.abs(fn)-np.abs(actual_ft(wn))))}")
    print(f"Samples = {N} and Time Period = {T}")
    pt.plot(wn,abs(fn),'k')
    pt.xlim(-x_limit,x_limit)
    pt.xlabel("Frequency")
    pt.ylabel("Magnitude of DFT")
    pt.title(f"DFT for Gaussian function ")
    pt.show()
```

```
pt.plot(wn,abs(actual_ft(wn)),'g')
pt.title("Plot of Actual CTFT of the Gaussian Function")
pt.xlabel("Frequency")
pt.ylabel("CTFT Magnitude")
pt.xlim(-x_limit,x_limit)
pt.show()
```

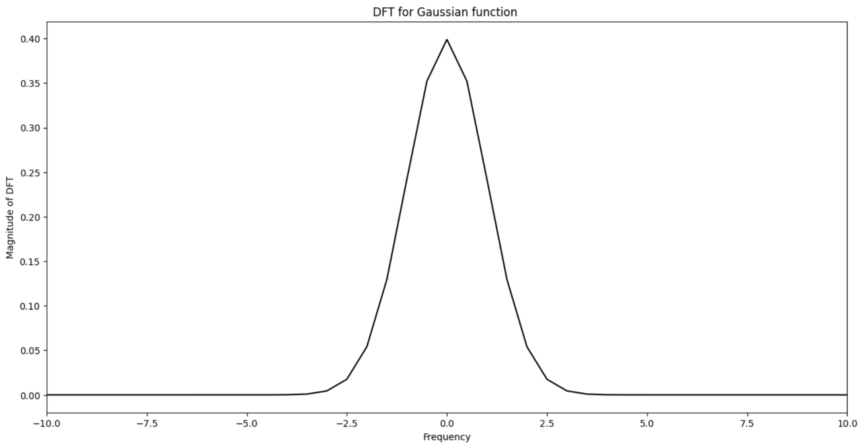

Figure 11: Estimated CTFT of Gaussian

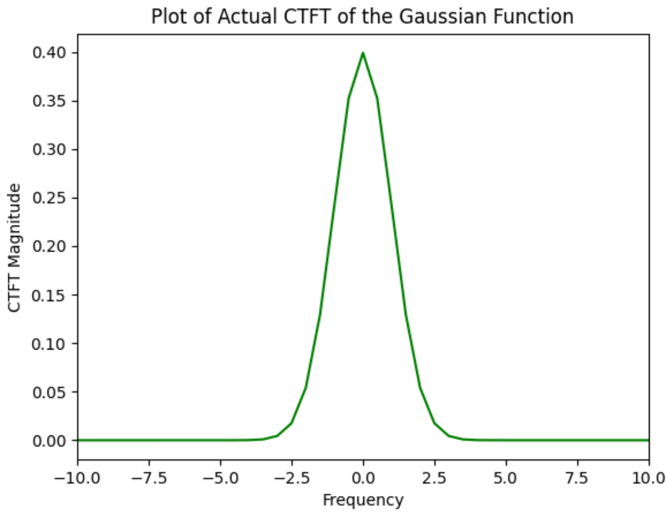

Figure 12: Expected CTFT of Gaussian

The summation of the absolute values of the error, final number of samples and final window size are given as:
True error : 5.314017021279596e-09
Samples = 512 and Time Period = 25.132741228718345 (8π)

Conclusion

- **DFT**s of different signals are determined and compared with the expected values.
- Adjusting the time frame T and number of samples N can give better visualisation of the **DTF**s to **CTFT**s.
- **CTFT** for Gaussian function is estimated using **DFT**s by choosing a suitable time frame.
- The suitable time frame was found out such that the absolute sum of error was less than a predetermined tolerance value.