EE2703 : Applied Programming Lab
Experiment 9

Mukhesh Pugalendhi Sudha
EE18B114

April 29, 2020

Abstract

The problem is to use `numpy.fft` library to compute the forward fourier transform of non periodic functions.

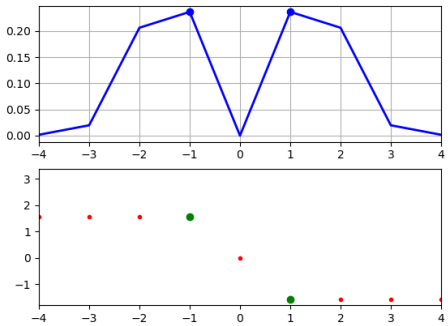
Introduction

We use `fft()` along with various methods in pylab to visualize the given non periodic time domain signal in the frequency domain. To minimize harmonics due to Gibbs phenomenon, we multiply the time domain signal by a Hamming window. This reduces the discontinuous jump in time domain.

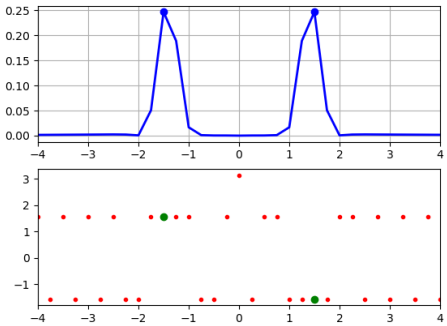
Results

1. Examples

The signal $sin(\sqrt{2}t)$ is transformed using the given methods. This is done with 64 samples and with 256 samples.



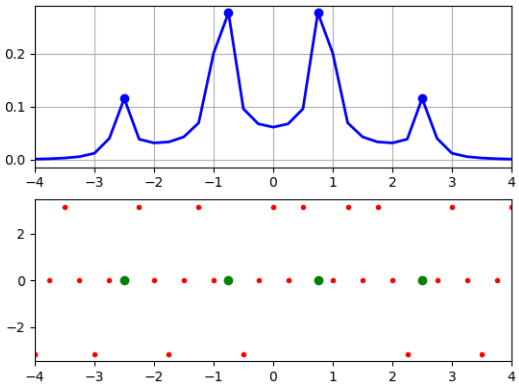
64 samples



256 samples

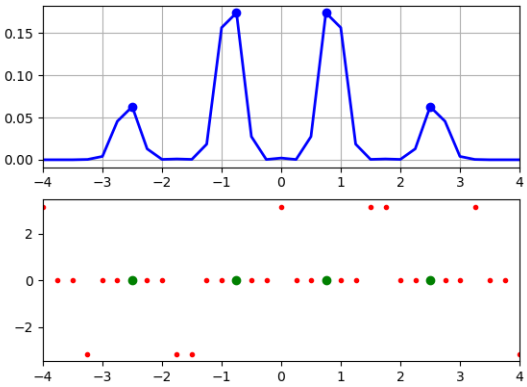
2. $cos^3(0.86 \times t)$

From the last experiment, we know that, the peaks should be at -3ω, $-\omega$, ω and 3ω.



without Hamming window

The harmonics do not die down rapidly.



with Hamming window

Harmonics disappear faster.

3. Estimation of ω_o and δ

To find ω_o, a weighted average among ω is taken with weights as $|Y(w)^2|$. This average is taken only for those values with magnitude greater than $1e-1$.
We know that,

$$cos(\omega_o t + \delta) = cos(\delta)cos(\omega_o) - sin(\delta)sin(\omega_o)$$

Using this relation, we can estimate $cos(\delta)$ and $sin(\delta)$ with `lstsq()`.

If $\omega_o = 1.4$ and $\delta = 1.2$, the estimates are,
$\omega_o = 1.4772749403447898$
$\delta = 1.2256826007486994$

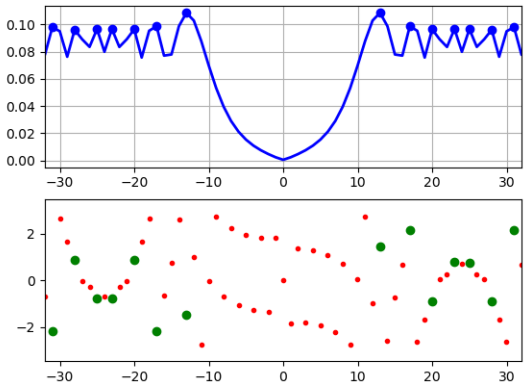
4. Addition of noise

The same process is done for signal with noise and the estimates are as follows.

If $\omega_o = 1.4$ and $\delta = 1.2$, the estimates are,
$\omega_o = 1.4042286367378962$
$\delta = 1.2950004820310628$

5. Chirped signal

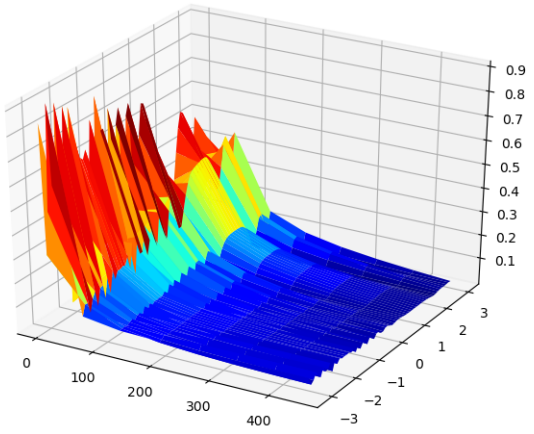
The signal $cos(16(1.5 + \frac{t}{2\pi})t)$ is given as input in 1024 steps. The spectrum is as follows.



Frequencies from 16 to 32 are present in the signal.

6. Chirped signal in chunks

The 1024 steps from the previous part is divided into chunks of 64 samples. Frequency present in each of those chunks is plotted in a 3d surface plot.



As time progresses, it can be seen that the frequency tends to become higher.

Conclusion

Thus `numpy.fft` is used to find and plot frequency spectrum of various non periodic signals.