EE2703 : Applied Programming Lab
Experiment 8

Mukhesh Pugalendhi Sudha
EE18B114

April 21, 2020

Abstract

The problem is to use `numpy.fft` library to compute the forward fourier transform and inverse fourier transform.

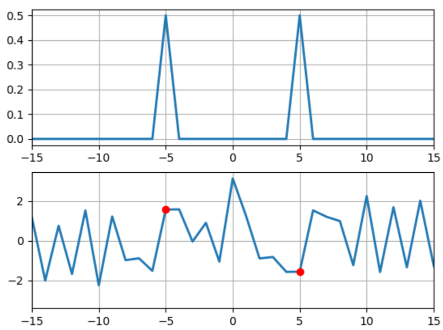
Introduction

We use `fft()` and `ifft()` along with various methods in pylab to visualize the given time domain signal in the frequency domain. Different signals like $sin(5t)$, $(1+0.1cos(t))cos(10t)$, $sin^3(t)$, $cos^3(t)$, $cos(20t+5cos(t))$ and $exp(-t^2/2)$ are transformed to get their respective frequency spectrum.

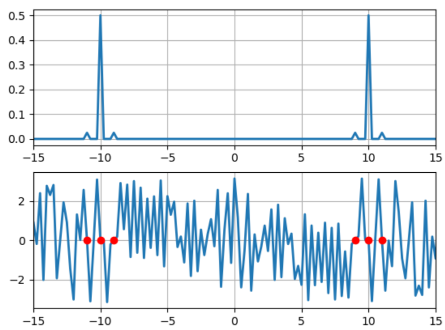
Results

1. Examples

The signals given in the examples are transformed using the given methods.



$sin(5t)$

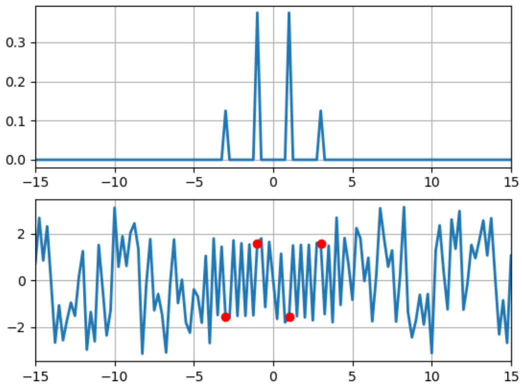


$(1+0.1cos(t))cos(10t)$e

2. $sin^3(t)$ and $cos^3(t)$

The expansion of $sin^3(t)$ is as follows:

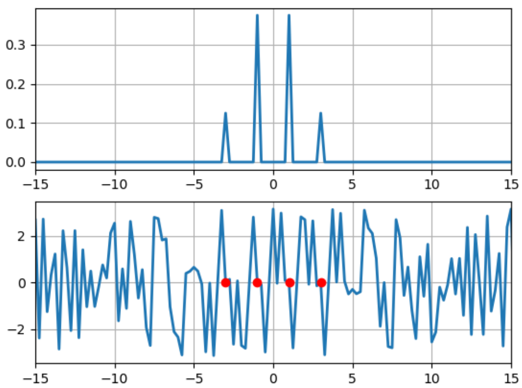
$$sin^3(t) = \frac{1}{8}(-je^{-3jt} + 3je^{-jt} - 3je^{jt} + je^{3jt})$$



$sin^3(t)$

Peaks are found as per the above expansion. Similarly,

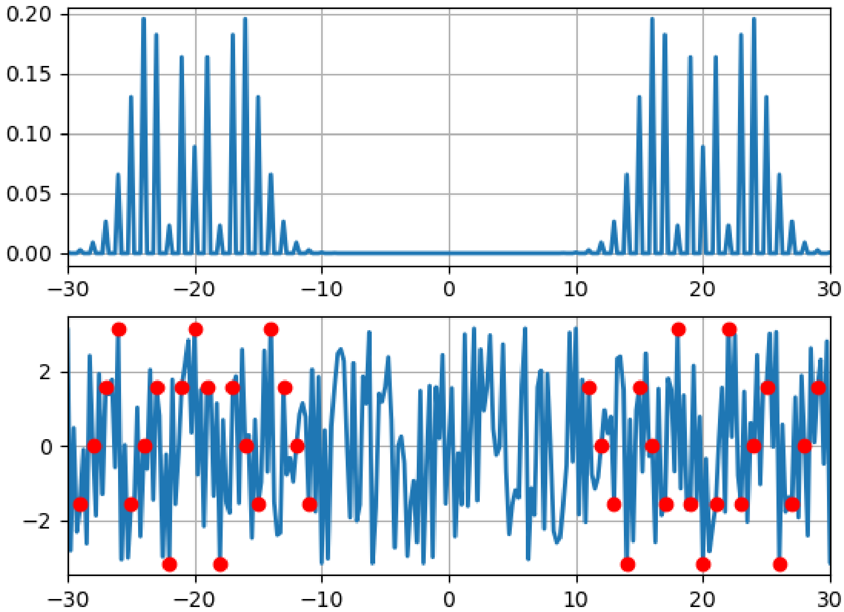
$$cos^3(t) = \frac{1}{8}(e^{-3jt} + 3e^{-jt} + 3e^{jt} + e^{3jt})$$



$cos^3(t)$

Phase at peaks = 0 as expected.

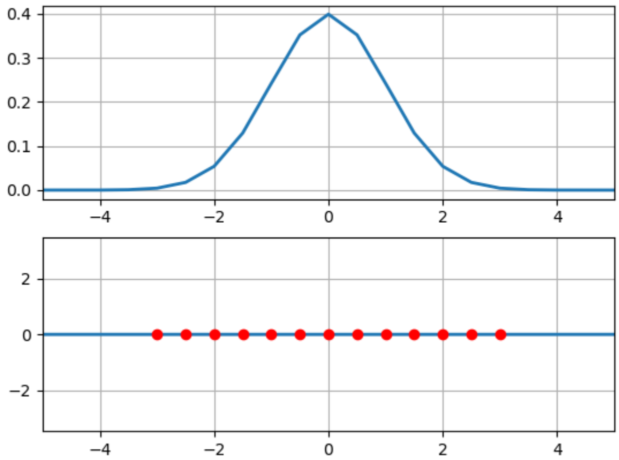
3. $cos(20t + 5cos(t))$



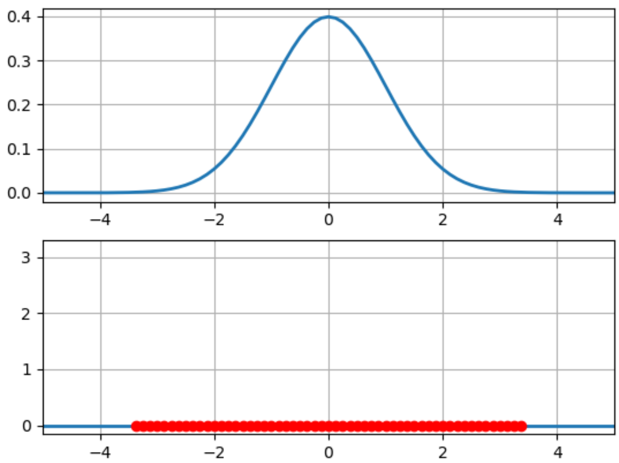
There seems to be a band of frequencies around the main frequency of 20 rad / s.

4. Response of Gaussian

To find the time range that gives `1e-6` accuracy, the sum of absolute differences compared with actual ω dependent Gaussian is computed for various ranges starting from PI, 2PI, ... The range which gives an error less than `1e-6` accuracy is 4PI. It gives an accuracy of `5.3505e-09`



However, if range is 16PI, we get an accuracy of `1.8794e-14` and the plot looks smoother.



.

Conclusion

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