Fast Fourier Transform

袁略真

3130103964

生物信息学

浙江大学

2016 年 4 月 18 日

1 傅里叶变换

傅里叶分析用于将时域信号转换为频率域形式, 在数据中识别出频率组成.

当对时域函数在$[0, 2\pi)$周期采样,对于获得离散数据可以通过离散傅里叶变换分解为不同频率的振幅和相位关系. 离散傅里叶变换可看做使用如下的三角函数进行插值:

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)e^{i2nk\pi/N}$$

$$n = 0, 1, ..., N-1, \ k = 0, 1, ..., N-1$$

其中,$X(k)$是频率为$f_k = k/2\pi$的分振动(频率域中的基)的振幅.

Fast Fourier Transform(FFT) accelerate the calculation of $X(k)$. However, the algorithm add constrains on data. The data should generate periodically on the $[0, 2\pi)$(on the time axis), and the number N of data points should be $N = 2^r, \ r = 2, 3, ...$.

Under these constrains, the FFT has some tricks and can shrink the time complex from $\Omega(N^2)$(solving systems of linear equation, I'm not quite sure for all the related algorithm. Gaussian Elimination leads to $O(n^3)$ complexity[1].) to $O(NlogN)$. For simple DFT(discrete fourier transform) the time complexity is $O(n^2)$[2].

2 Result

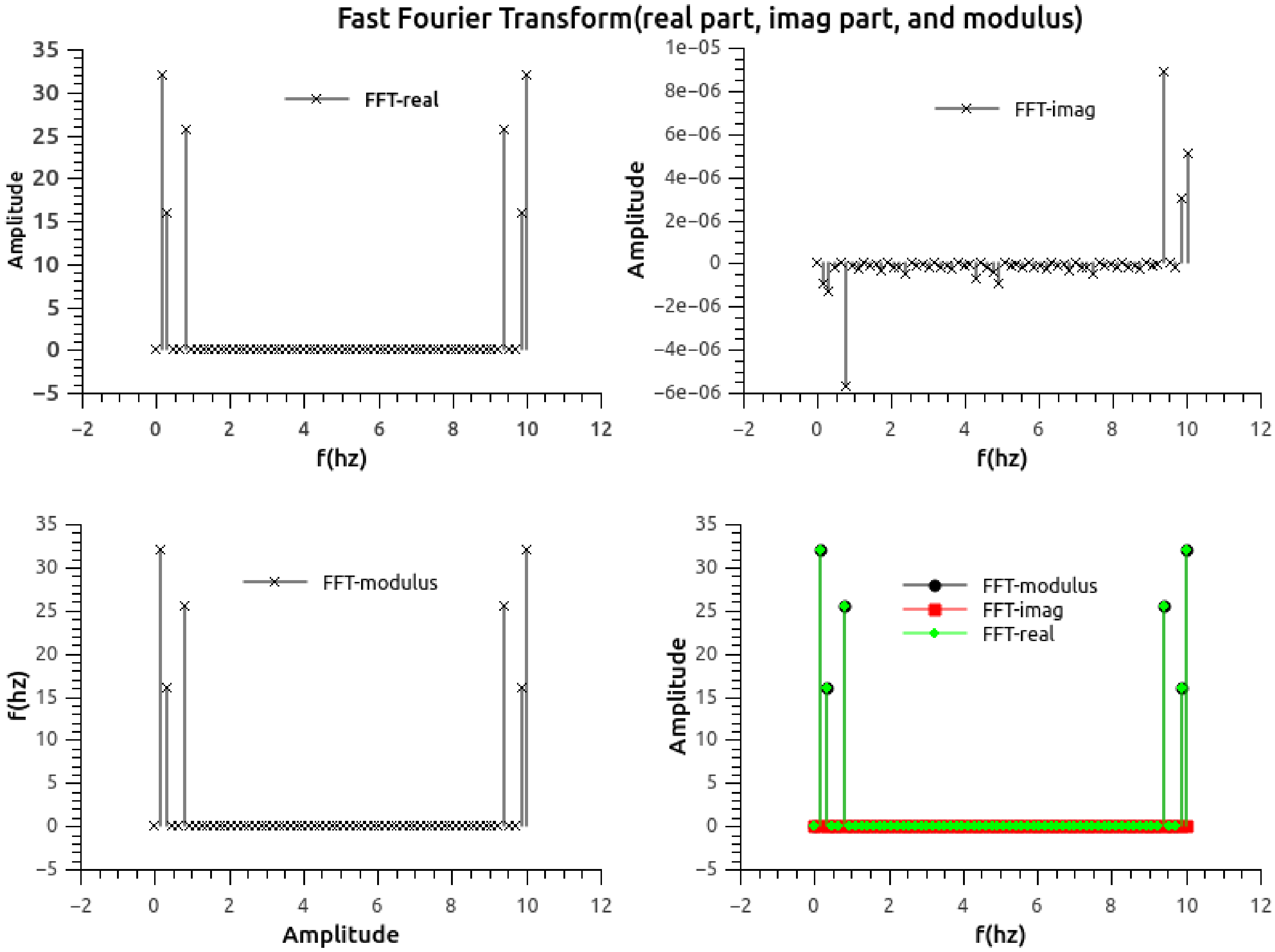

Figure 1: Frequency spectrum after the FFT. The function used is $x(t) = \ cos(t) + 0.5 * cos(2 * (t)) + 0.8 * cos(5 * (t))$. 64 data points are generated periodically from $[0, 2\pi)$.

References

[1] Gaussian elimination – Wikipedia(Computational efficiency section)
`https://en.wikipedia.org/wiki/Gaussian_elimination`

[2] Fast Fourier transform – Wikipedia
`https://en.wikipedia.org/wiki/Fast_Fourier_transform`