

数字信号处理第三次作业

闫昊昕

2019210361

2020 年 10 月 18 日

第1题

个人编写的 n 点 FFT 程序在 `fft.py` 中定义。该函数接受两个参数为函数输入：信号点序列 x_n 以及信号点数 N 。该函数要求信号点数严格为 2 的整数次幂。

对一个输入 16384 点信号 $f(n) = \sin(2\pi \times 50n) + \cos(2\pi \times 100n)$ ，个人编写的 `fft` 程序以及 `numpy` 包自带的 `fft` 程序得到的频谱图像如下：

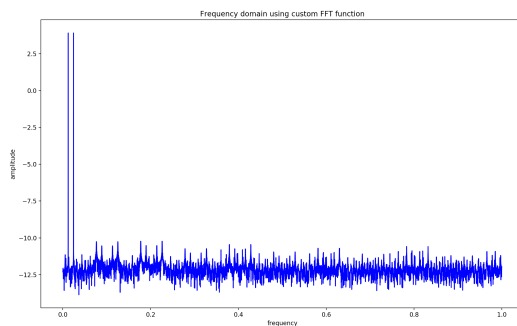


图 1: 个人编写的基 2FFT 程序结果

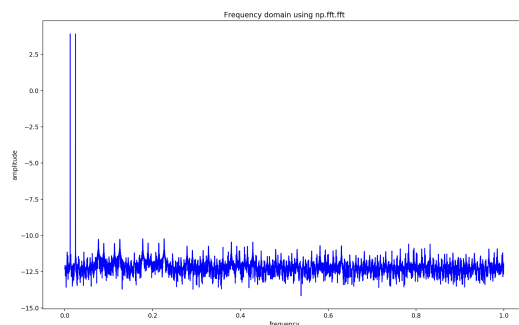


图 2: Numpy 自带的 FFT 程序结果

运行时间方面，个人编写的 FFT 程序可以在 0.2 秒左右的时间内完成对 16384 点信号的 FFT 运算，在运行速度上远远高于 DFT 算法，但和 `np.fft.fft` 函数相比运行速度仍然明显较慢，如下图所示：

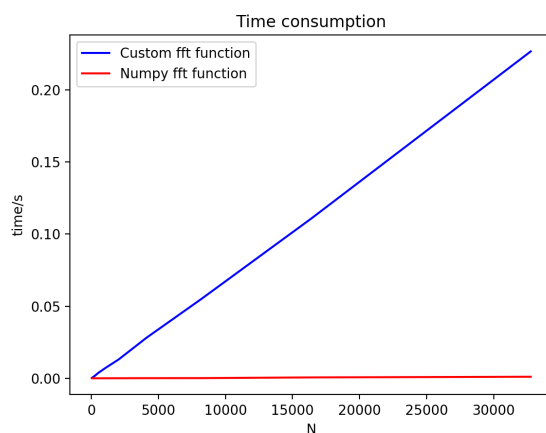


图 3: 两种程序的运行时间比较

同时可以看到随着 N 的增加程序运行速度的变化约为线性，这与 FFT 算法的时间复杂度 $O(n \log n)$ 相吻合。

第2题

可以用两种方式实现函数的自相关，一种是根据定义实现序列的自相关，另一种方式是先对序列进行 FFT 得到其频谱，再做平方运算得到功率谱，最后做 IFFT 得到自相关函数的估计，本题中自相关函数的主要实现方式为定义法，功率谱法做验证使用。

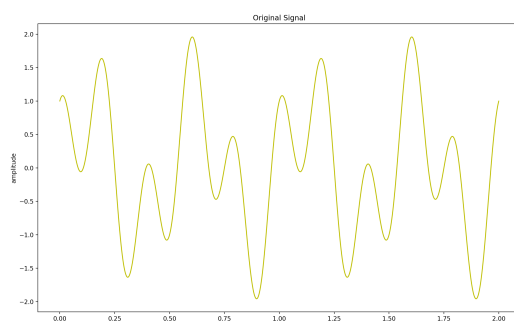


图 4: 原信号的时域表示

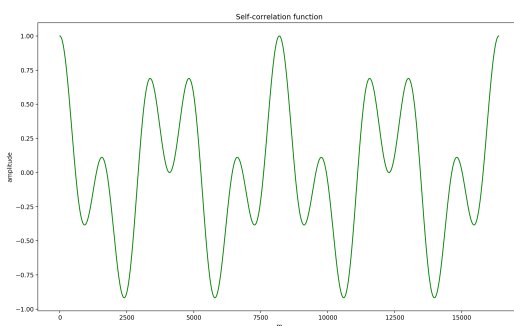


图 5: 根据定义实现的自相关函数

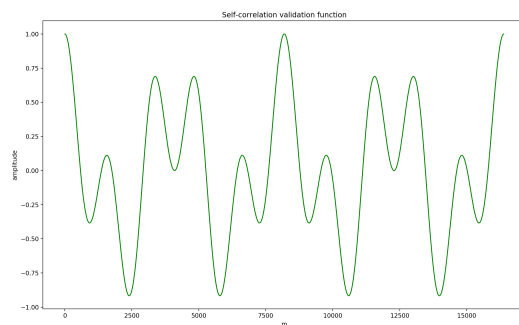


图 6: 根据功率谱原理得到的自相关函数

图 4为原信号的时域表示，图 5为通过根据定义计算的自相关函数得到的自相关函数谱，图 6为通过功率谱方法估计的自相关函数。可以看到原信号的自相关函数关于频谱中心对称，与理论相符合。