# 数字信号处理 大作业

2024秋季学期

姓名：陈彦旭

班级：无24

## Task 1

介绍所使用的 SFFT 算法的原理与流程，画出稀疏信号的频谱真值与 SFFT 所得频谱的对比图。

选取“哈希映射法”。

符号说明：$N$ 为信号长度，$K$ 为信号频谱稀疏度，$B$ 为分“筐”数，$d$ 为循环时用到的参数，$L$ 为循环估计的次数。

### Step 1：频谱重排

实际中的稀疏信号大多频点较为集中，为了保证频域“分筐”的时候以极大概率将不同大值频点装入不同的筐中，需要对信号进行重排，将原本集中的大值频点分散开。利用 DFT 的循环位移性质和缩放性质，在时域对信号进行等价的操作。选取随机参数 $\sigma, \tau$，进行缩放+位移：

$$p[n] = x[((\sigma n + \tau))_N] \tag{1}$$

等价于频域有：

$$P[((\sigma k))_N] = X[k] W_N^{-\tau k} \tag{2}$$

### Step 2：窗函数滤波

矩形窗函数时域为 sinc 序列，无限长，需要做截断操作，但是截断导致频域的通带和阻带变大。可以选取高斯窗函数与之时域相乘，具有平滑效果。

构造平坦的窗函数 $g[n]$ 及其频谱 $G[k]$，然后与信号时域相乘：

$$y[n] = p[n] \cdot g[n] \tag{3}$$

频域为两个信号频谱的循环卷积：

$$Y[k] = P[k] \circledast G[k] \tag{4}$$

由于 $P[k]$ 只在少数频点上有大值，可以看作若干不同幅度的 $\delta[k]$ 叠加，因此频谱 $Y[k]$ 近似为若干不同位置上的窗函数的叠加，且不同窗函数之间有重叠的概率较低。

## Step 3：降采样

对信号 $y[n]$ 时域混叠：

$$z[n] = \sum_{j=0}^{N/B-1} y[n+Bj],\, n = 0, \ldots, N-1 \tag{5}$$

对 $z[n]$ 做 $B$ 点 FFT，结果实际上是对频谱 $Y[k]$ 降采样，降采样率为 $N/B$：

$$Z[k] = Y[k\frac{N}{B}],\, k = 0, \ldots, B-1 \tag{6}$$

也就是将连续的 $N/B$ 个频点装入一个筐中，一共得到 $B$ 个筐。

## Step 4：重构恢复

取出 $Z[k]$ 中幅度最大的 $dK$ 个频点对应的坐标组成集合 $J$，集合 $J$ 是以 $z[k]$ 幅值为观测域得到的大值频点的坐标集合，每一个坐标都是一个筐，筐中装有 $N/B$ 个频点。通过哈希函数关系可以将 $J$ 映射回以 $X[k]$ 为观测域的大值频点的"原像"，得到集合 $I$，因此集合 $I = \{k \in [0, N-1] | h_\sigma(k) \in J\}$，大小为 $dKN/B$。

其中哈希函数的正确定义应该是（参考文献中并没有正确的表示出循环位移中"模"的关系，否则哈希函数将是单调增长的，不可能从 $[0, N-1]$ 映射到 $[0, B-1]$）：

$$h_\sigma(k) = \mathrm{mod}(\mathrm{round}(\mathrm{mod}(\sigma * k, N) * B/N), B),\; k = 0, \ldots, N-1 \tag{7}$$

```
hash_sigma = mod(round(mod(sigma * (0:N - 1), N) * B / N), B) + 1;
```

对于集合 $J$ 中的每一个原像频点 $k$，需要除以窗函数滤波器的增益，估计其原本的幅度值。

$$\hat{X}[k] = Z[h_\sigma(k)] W_N^{\tau k} / G[o_\sigma(k)] \tag{8}$$

所使用的偏移量函数 $o_\sigma(k)$ 将 $[0, N-1]$ 映射到 $[-N/2B, N/2B-1]$，如果加上模 $N$ 的话，就是映射到 $[0, N/2B-1], [N-N/2B, N-1]$，这样能够直接作为数组的索引：

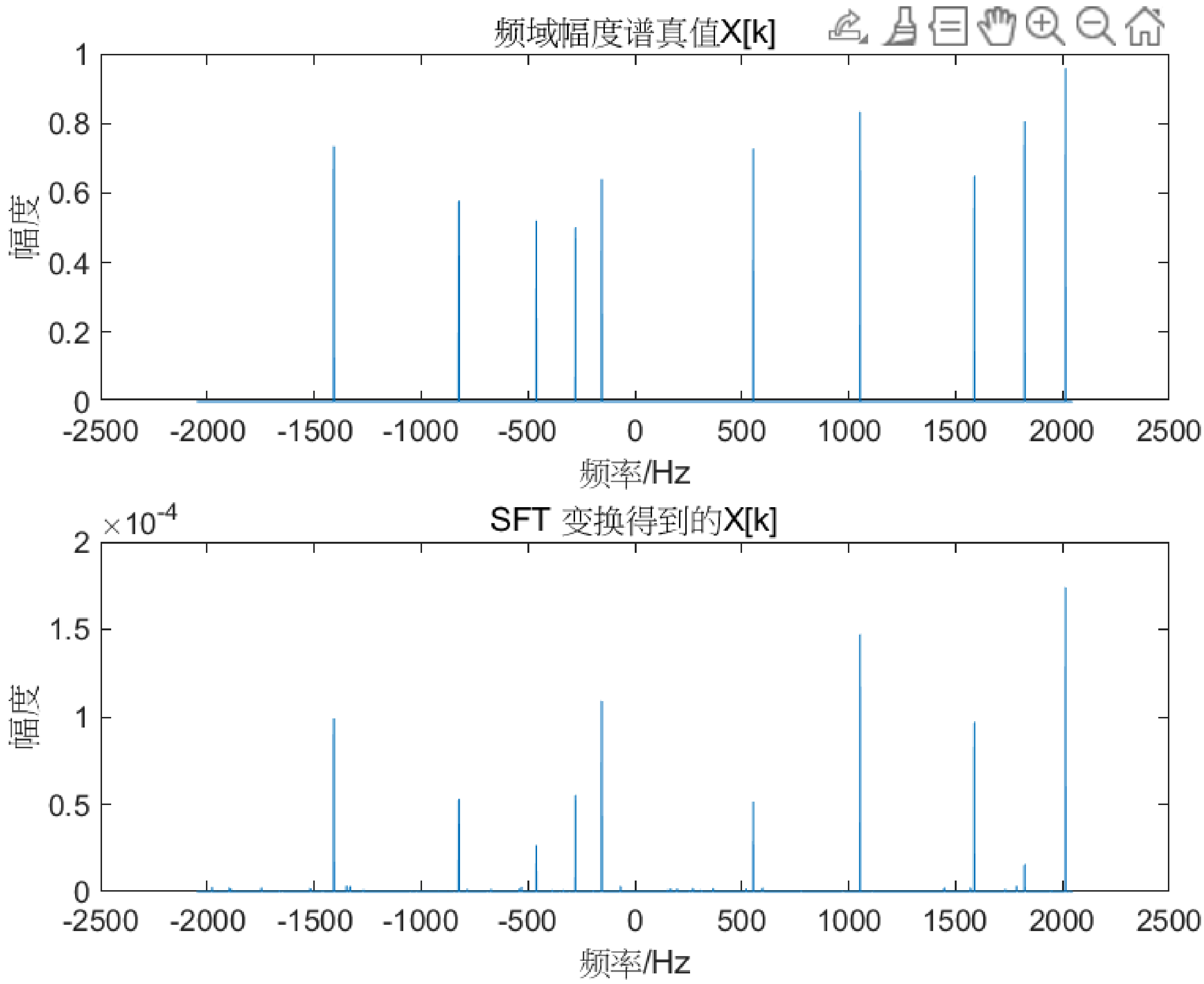
$$o_\sigma(k) = \mathrm{mod}(\sigma k - \mathrm{round}(\sigma k B/N) * N/B,\; N) \tag{9}$$

```
offset_sigma = mod(sigma * (0:N - 1) - round(sigma * (0:N - 1) * B / N) * N / B, N)
+ 1;
```

经过上述一次操作之后得到集合 $I$ 和其对应坐标估计的幅度值。循环操作 $L$ 次后，所有的集合 $I$ 组成大小为 $L \times dKN/B$ 的矩阵，找出其中出现次数超过 $L/2$ 的频点坐标（或者出现次数最多的 $K$ 个频点的坐标），作为最终所求的原始大值频点的坐标，将其在循环中出现的所有幅度值的中位数作为最终所估计的频率值。

## 仿真验证

设定实验参数为：$N=4096, K=10, d=4$。运行结果如下：




可见 SFT 算法较为准确地找出了所有稀疏的频点。幅值较小，相对大小不一致，可能是因为丢弃了大部分原始序列中的点，降采样操作也会造成能量损失。

## Task 2

以 $O(f(n,k))$ 的形式给出所使用的 SFFT 算法运行的时间复杂度。分别画出 SFFT 运行时间随信号长度 $N$ 以及信号频谱稀疏度 $k$ 变化的关系图，利用关系图验证你的推测。

根据参考文献 Ref 2 的推导，SFFT 算法的复杂度约为 $O(\log_2 n\sqrt{nk\log_2 n})$， $n$ 为信号长度且为 2 的次幂， $k$ 为频域的稀疏度。在 $k$ 为 $O(n/\log n)$ 量级的时候比 FFT 更快。

# Task 3

介绍所选择压缩感知算法的流程、原理。探究至少两种因素对 SFFT 算法与压缩感知算法性能的影响，分别画出使用两种算法得到的信号频谱估计的 $l_1$ 误差与所选择的因素之间的关系图。对比使用两种算法得到的关系图，并尝试对所得结果加以解释。

选取算法。

参考文献：

Hassanieh H, Indyk P, Katabi D, et al. Simple and practical algorithm for sparse fourier transform[C]. // Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms. [S.l.]: ACM, 2012: 1183-1194.

[1]仲顺安,王雄,王卫江,等.稀疏傅里叶变换理论及研究进展[J].北京理工大学学报,2017,37(02):111-118.DOI:10.15918/j.tbit1001-0645.2017.02.001.

[1]李非凡.稀疏傅里叶变换算法的硬件实现[D].东南大学,2023.DOI:10.27014/d.cnki.gdnau.2023.000766.