实 验 报 告

电子信息技术 学院 21 级 04 班　　学号 PB21061374　　姓名 赵栋　　日期 2023-12-10

实验四 窗函数

4.1 实验目的

(1) 理解窗函数的基本性能，掌握主瓣宽度与旁辨电平的控制原则。

(2) 探究两类相位特性对信号处理的影响。

4.2 实验原理

(1) 用窗函数截取序列做 DFT 分析，频谱泄露体现在宽阔的主瓣和旁瓣上；

(2) 控制窗函数的主瓣宽度和旁瓣电平，可以控制改善泄露对信号频谱识别的影响；

(3) 当幅度相差较大的两个信号同时存在时，需要仔细设计窗函数的主瓣宽度和旁瓣电平，以免弱信号淹没在强信号的旁瓣或主瓣中。

4.3 实验内容和分析

(1) 设

$$x_1(n) = 31.6e^{j\frac{3\pi}{7}n} + 0.005e^{j\frac{4\pi}{5}n}, 0 \le n \le 1023.$$

分别使用矩形窗、Hamming 窗对$x_1(n)$做 DFT 得到$X_1(k)$，画出的幅度谱分别如图 4-1 和图 4-2 所示。

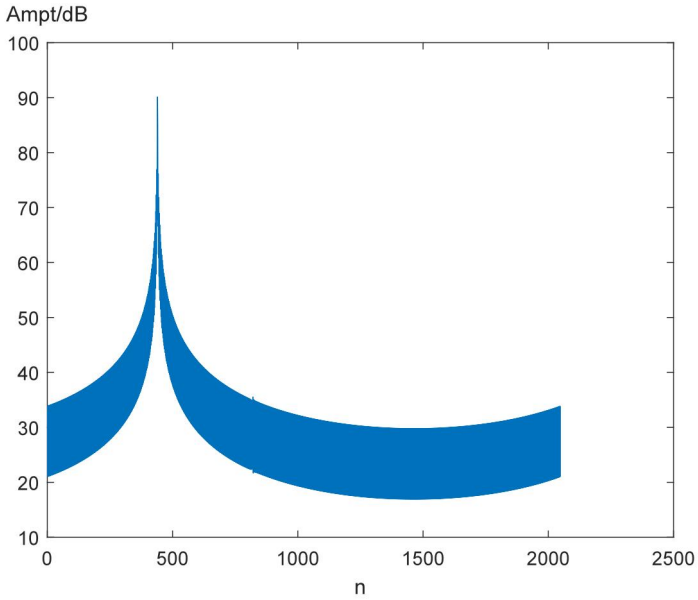


图 4-1　$x_1(n)$被矩形窗截取后 DFT 幅度谱

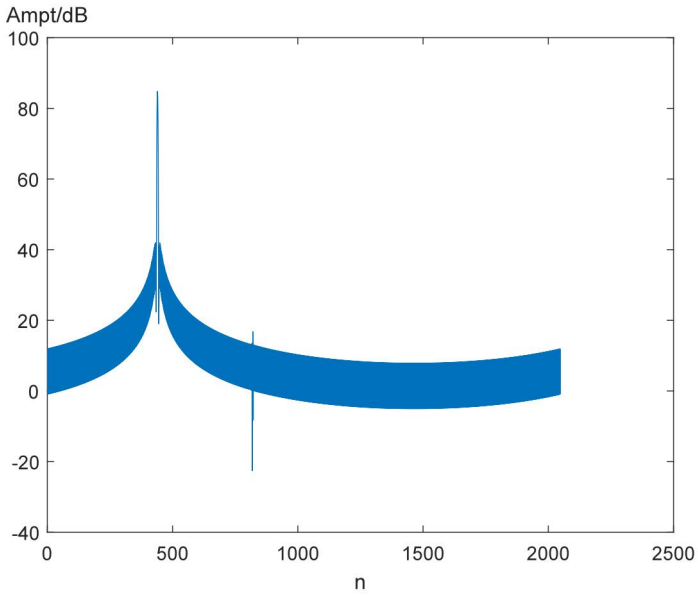


图 4-2　$x_1(n)$被 Hamming 窗截取后 DFT 幅度谱

可以看出矩形窗对$x_1(n)$截取之后的信号进行 DFT 变换得到的幅度谱主要包含的是主瓣信号，小信号则被主瓣信号淹没；而对于 Hamming 窗截得的信号进行 DFT 变换得到的幅度谱在旁瓣处有一个尖锐信号，对应的是$x_1(n)$中的小信号分量，说明 Hamming 窗的截取能更好地保留原有信号中包括小信号的所有信息。

(2) 设

$$x_2(n) = 31.6e^{j\frac{3\pi}{7}n} + 10e^{j\left(\frac{1}{7}+\frac{1}{1024}\right)3\pi n}, 0 \le n \le N-1.$$

使用 Blackman 窗，分别在$N = 1024$ 和$N = 2048$ 两种情况下，对$x_2(n)$做 DFT 得到$X_2(k)$，画出幅度谱，如图 4-3 和 4-4 所示。

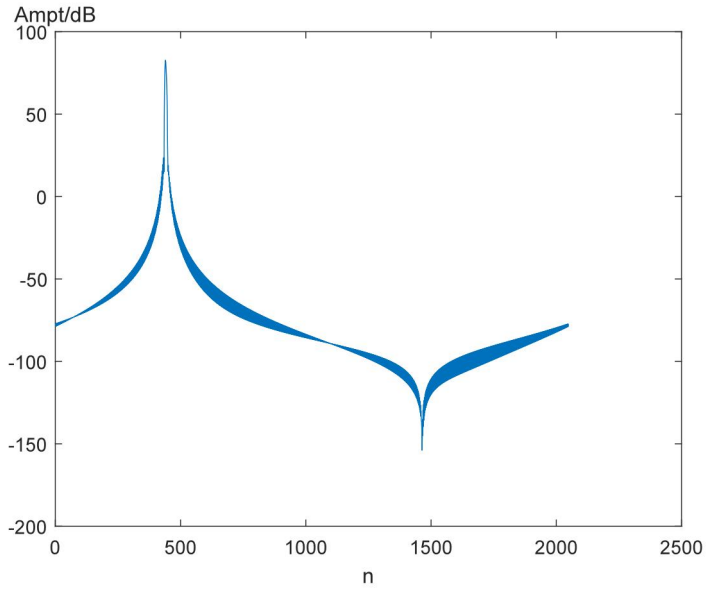


图 4-3　$x_2(n)$被 Blackman 窗$(N=1024)$截取后 DFT 幅度谱

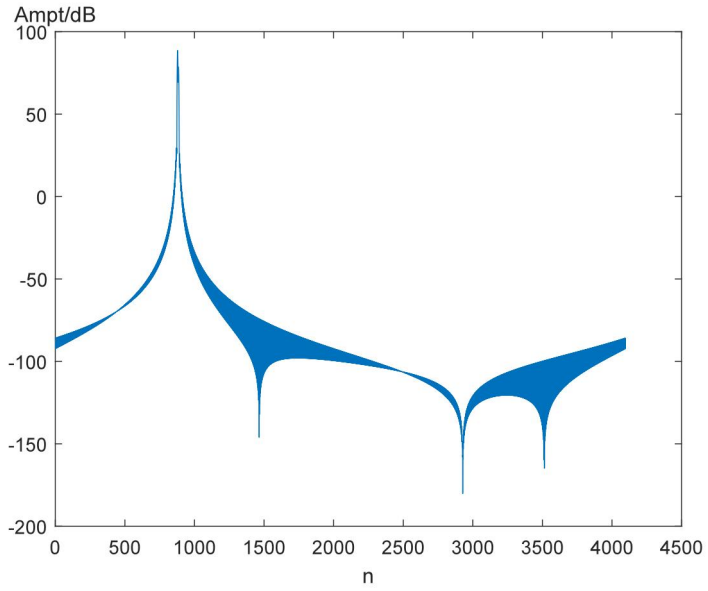
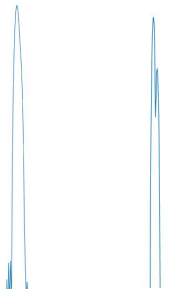


图 4-4　$x_2(n)$被 Blackman 窗$(N=2048)$截取后 DFT 幅度谱

对比$N=1024$ 和$N=2048$ 的幅度谱主瓣，如图 4-5 所示，可以看出$N=1024$ 时，高频分量淹没在低频分量的主瓣中；$N=2048$ 时，低频分量的主瓣变窄，高频分量显现，此时$x_2(n)$对应的高频信号和低频信号均能在幅度谱中显现。



(a) (b)

图 4-5 Blackman 窗截取幅度谱主瓣对比

$(a)N = 1024$

$(b)N = 2048$

4.4 实验总结

本实验通过对比了不同类型的窗函数对特定信号进行截取后的幅度谱特点展示了不同窗函数的特性。矩形窗是一种较为常见的窗函数，截取简单但是难以保留小信号分量，Hamming 窗则可以有效解决这个问题。对于 Blackman 窗，截取长度越高，主瓣宽度越窄，因此在信号含有多个频率差距较大的信号时，需要选取足够长的 Blackman 窗函数，才能避免高频分量淹没在低频分量的主瓣中。

实验五 FFT 算法

5.1 实验目的

(1) 加深对快速傅里叶变换（FFT）的理解。
(2) 实际编程实现 FFT 算法。

5.2 实验原理

编程实现一个 16 点 DFT 的基-2 快速算法。

5.3 实验内容和分析

设$x(n)=2\,sin\left(\frac{\pi}{4}n\right)+sin\left(\frac{5\pi}{8}n\right)+sin\left(\frac{3\pi}{4}n\right),n=0,1,2,...,15.$

(1) 对序列$x(n)$做 DFT，使用 MATLAB 内置的 stem 函数画出幅度谱，得到的图像如图 5-1 所示。

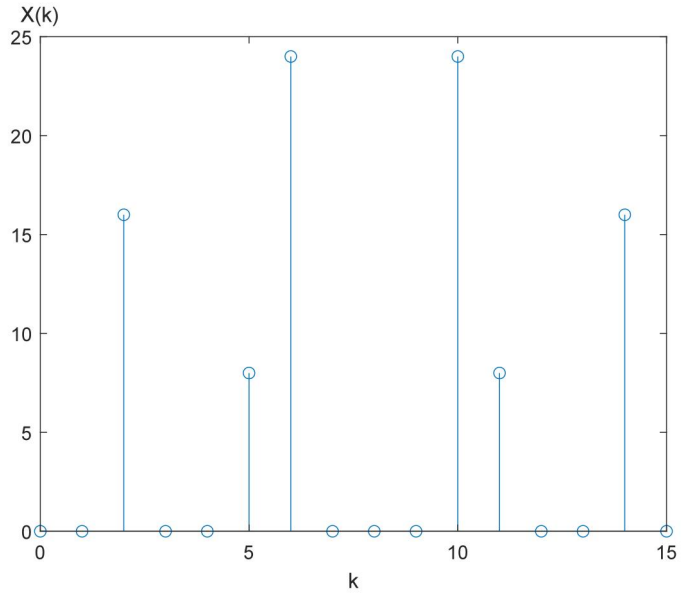


图 5-1 $x(n)$进行 DFT 变换后的幅度谱

(2) 编制按时间抽取的基 2FFT 算法程序，要求顺序输入、反序输出，对序列做 FFT。一个 16 点的 FFT 蝶形图如图 5-2 所示，可以看出对于每一级的蝶形图进行的 FFT 运算，都是把输入从中间分成两个部分然后再进行交叉运算并乘以对应的旋转因子W_N^k，由此可以得到如下所示的代码，具体细节已在注释说明。

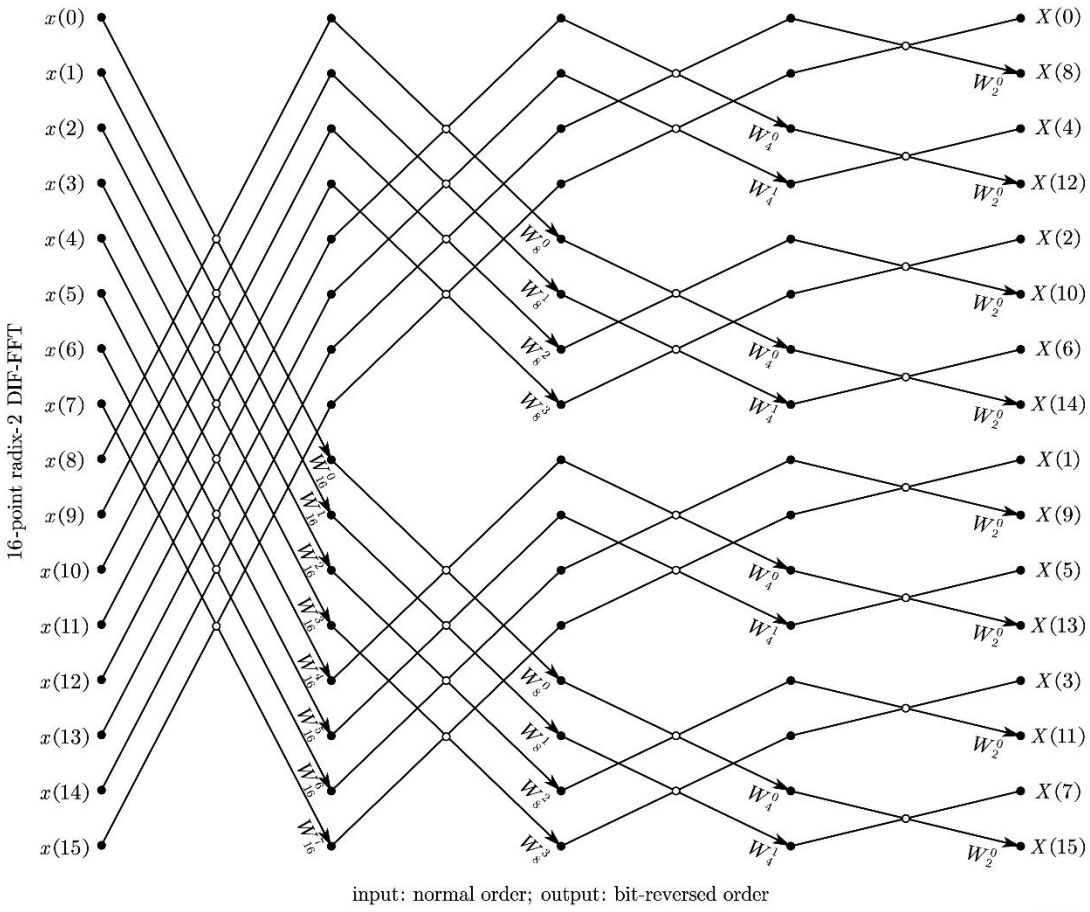


图 5-2　16 点基 2FFT 蝶形图(顺序输入，反序输出)

```
%顺序输入，反序输出的基 2-FFT 运算函数
function X = radix2FFT(x, N)
    %计算 FFT 蝶形图层数
    round = log2(N);
    i = 1;
    j = N/2;
    %计算每层蝶形图的输出
    for m = 1:round
        temp = zeros(2*i, j);
        %进行蝶形运算,将每行数据拆成两行进行蝶形运算
        for l = 1:i
            X1 = x(l, 1:j);
            X2 = x(l, j+1:2*j);
            %计算旋转因子系数
            W = exp(-1j * 2*pi / N * (0:j-1) * i);
            temp(2*l-1: 2*l, 1:j) = [X1+X2; (X1-X2).*W];
        end
        x = temp;
```

```
        i = i*2;
        j = j/2;
    end
    %将最后计算的 x 转置得到倒序输出结果
    X = x';
end
```

在命令行输出反序结果的幅度值，即调用 X2 = radix2FFT(x, N)，得到的输出为

```
 列 1 至 5

  -0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 - 0.0000i   0.0000 + 0.0000i   0.0000
+16.0000i

  列 6 至 10

  -0.0000 -24.0000i  -0.0000 +24.0000i  -0.0000 -16.0000i  -0.0000 - 0.0000i   0.0000
- 0.0000i

  列 11 至 15

  -0.0000 + 8.0000i   0.0000 + 0.0000i   0.0000 - 0.0000i  -0.0000 - 8.0000i   0.0000
+ 0.0000i

  列 16

  -0.0000 + 0.0000i
```

然后调用函数 X2=bitrevorder(X2)，将输出结果从反序转换为顺序，画出幅度谱，结果如图 5-3 所示。对比使用 DFT 函数得到的结果相比完全一样，说明该 FFT 符合编写要求。

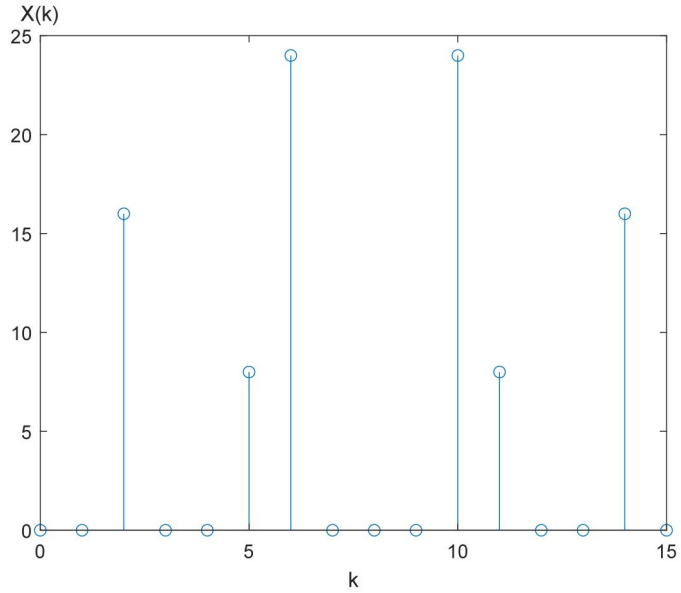


图 5-3　$x(n)$进行自制 FFT 变换后的幅度谱

5.4　实验总结

本次实验尝试了根据基 2FFT 原理的蝶形图来手动编写 FFT 函数实现快速傅里叶变换。在实现 FFT 时需要注意根据蝶形图进行合适的分组，注意运算前和运算后的数据大小，并乘以正确的参数就能实现 FFT 变换。注意这里编写的 FFT 必须满足数据长度为 2 的若干次方，如果输入数据不满足这个条件需要通过补 0 实现。