

浙江大学实验报告

专业： 信息工程
姓名： 箫宇
学号：
日期： 2023 年 7 月 2 日
地点： 教 4-421

课程名称： 数字信号处理 指导老师： 徐元欣 成绩：
实验名称： 基 4-FFT 算法编程 实验类型： 验证 同组学生姓名： —

一 实验目的和要求

FFT 是快速计算 DFT 的一类算法的总称。通过序列分解，用短序列的 DFT 代替长序列的 DFT，使得计算量大大下降。基 4-FFT 是混合基 FFT 的一个特例。

通过编写基 4-FFT 算法程序，加深对 FFT 思路、算法结构的理解。

二 实验内容和步骤

编写 16 点基 4-FFT 算法的 MATLAB 程序（studentname.m 文件）。

产生 16 点输入序列 x ，用自己的学号作为前 10 点的抽样值，后面补 6 个零值抽样。算出 16 点频谱序列 X ，用 $\text{stem}(X)$ 显示频谱图形。

三 主要仪器设备

用 MATLAB。

四 操作方法和实验步骤

（参见“二、实验内容和步骤”）

五 实验数据记录和处理

1. 基 4-FFT 算法思路、流图结构简述如下

1.1 算法思路

令序列 $x(n)$ 的 N 点 DFT 的结果为 $X(k)$ ，且有 $N = 4^m$ ，现按 $((n))_4$ 的结果对序列进行分组，得

$$\begin{aligned}x^{(0)}(n) &= x(4n) \\x^{(1)}(n) &= x(4n + 1) \\x^{(2)}(n) &= x(4n + 2) \\x^{(3)}(n) &= x(4n + 3)\end{aligned} \quad 0 \leq n \leq \frac{N}{4} - 1$$

因有:

$$\begin{aligned}
 X(k) &= \sum_{l=0}^{4^{m-1}-1} x(4l)W_N^{4lk} + \sum_{l=0}^{4^{m-1}-1} x(4l+1)W_N^{(4l+1)k} + \sum_{l=0}^{4^{m-1}-1} x(4l+2)W_N^{(4l+2)k} \\
 &+ \sum_{l=0}^{4^{m-1}-1} x(4l+3)W_N^{(4l+3)k} = \sum_{l=0}^{4^{m-1}-1} x^{(0)}(l)W_{4^{m-1}}^{4lk} + W_N^k \sum_{l=0}^{4^{m-1}-1} x^{(1)}(l)W_{4^{m-1}}^{4lk} \\
 &+ W_N^{2k} \sum_{l=0}^{4^{m-1}-1} x^{(2)}(l)W_{4^{m-1}}^{4lk} + W_N^{3k} \sum_{l=0}^{4^{m-1}-1} x^{(3)}(l)W_{4^{m-1}}^{4lk} = X^{(0)}((k))_{4^{m-1}} \\
 &+ W_N^k X^{(1)}((k))_{4^{m-1}} + W_N^{2k} X^{(2)}((k))_{4^{m-1}} + W_N^{3k} X^{(3)}((k))_{4^{m-1}} \\
 0 \leq k \leq N-1 &= 4^m - 1
 \end{aligned}$$

其中:

$$\begin{aligned}
 X^{(0)}(k) &= \text{DFT}_{4^{m-1}} \{x^{(0)}(n)\} \\
 X^{(1)}(k) &= \text{DFT}_{4^{m-1}} \{x^{(1)}(n)\} \\
 X^{(2)}(k) &= \text{DFT}_{4^{m-1}} \{x^{(2)}(n)\} \\
 X^{(3)}(k) &= \text{DFT}_{4^{m-1}} \{x^{(3)}(n)\}
 \end{aligned}$$

若令 $0 \leq k \leq 4^{m-1} - 1$, 则上式可以写作:

$$\begin{cases}
 X(k) = X^{(0)}(k) + W_N^k X^{(1)}(k) + W_N^{2k} X^{(2)}(k) + W_N^{3k} X^{(3)}(k) \\
 X(k + 4^{m-1}) = X^{(0)}(k) - jW_N^k X^{(1)}(k) - W_N^{2k} X^{(2)}(k) + jW_N^{3k} X^{(3)}(k) \\
 X(k + 2 \times 4^{m-1}) = X^{(0)}(k) - W_N^k X^{(1)}(k) + W_N^{2k} X^{(2)}(k) - W_N^{3k} X^{(3)}(k) \\
 X(k + 3 \times 4^{m-1}) = X^{(0)}(k) + jW_N^k X^{(1)}(k) - W_N^{2k} X^{(2)}(k) - jW_N^{3k} X^{(3)}(k)
 \end{cases}$$

1.2 流图结构

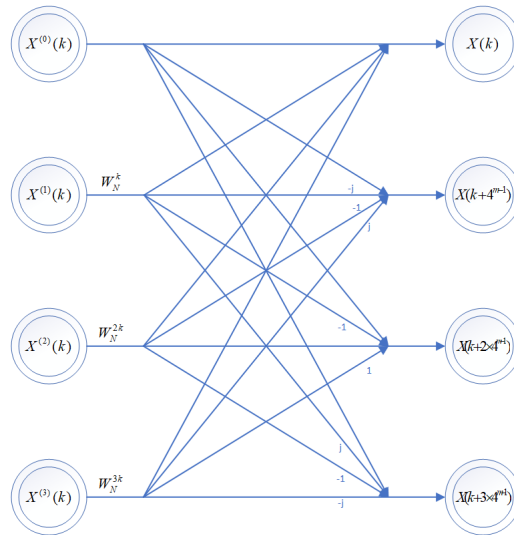


图 1: 流图结构

2. 16 点基 4-FFT 算法的流图绘出如下

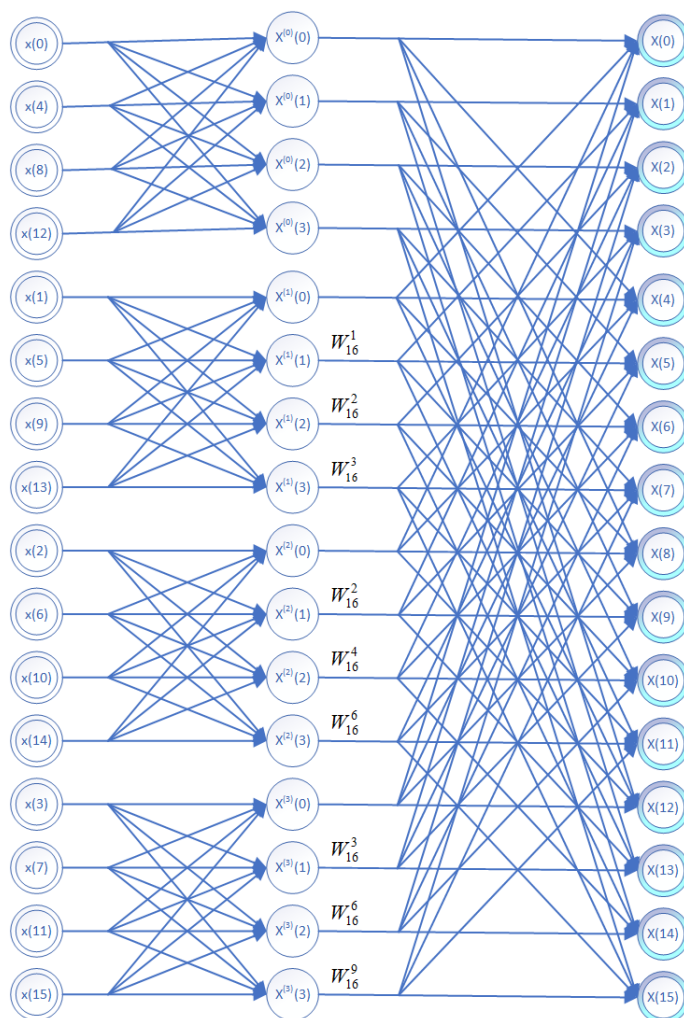


图 2: 16 点基 4-FFT 算法的流图

3. 16 点基 4-FFT 算法的 MATLAB 程序 (studentname.m) 列出如下

Listing 1: 绘图函数

```

1 %输入学号以及补零
  x = [3 1 9 0 1 0 5 0 5 5 , zeros(1 , 6)];
3 %基4蝴蝶图相加时的系数
  t = [1 1 1 1 ; 1 -1j -1 1j ; 1 -1 1 -1 ; 1 1j -1 -1j];
5 %将待fft变换的序列分解，矩阵中每一行为一组
  x1 = [x(1:4:16) ; x(2:4:16) ; x(3:4:16) ; x(4:4:16)];
7 %对每组进行4点FFT
  X1 = x1*t;
9 %得到第二级运算的旋转因子矩阵
  w = exp(-1j*pi/8);

```

```
11 W = [1 1 1 1 ; 1 w w^2 w^3 ; 1 w^2 w^4 w^6 ; 1 w^3 w^6 w^9];  
    %第二级FFT变换  
13 X2 = ((X1.*W).'*t).';  
    X = [X2(1 , 1:4) , X2(2 , 1:4) , X2(3 , 1:4) , X2(4 , 1:4)];  
15 %绘制图像  
    figure(1)  
17 n = 0:1:15;  
    stem(n , x , 'filled');  
19 figure(2)  
    stem(n , abs(X) , 'filled');  
21 figure(3)  
    stem(n , abs(fft(x)) , 'filled');
```

4. 用自己的学号构成的输入序列为（列出数值，插入图形）

4.1 数值

$x = [3 \ 1 \ 9 \ 0 \ 1 \ 0 \ 5 \ 0 \ 5 \ 5 \ 0 \ 0 \ 0 \ 0 \ 0]$

4.2 图形

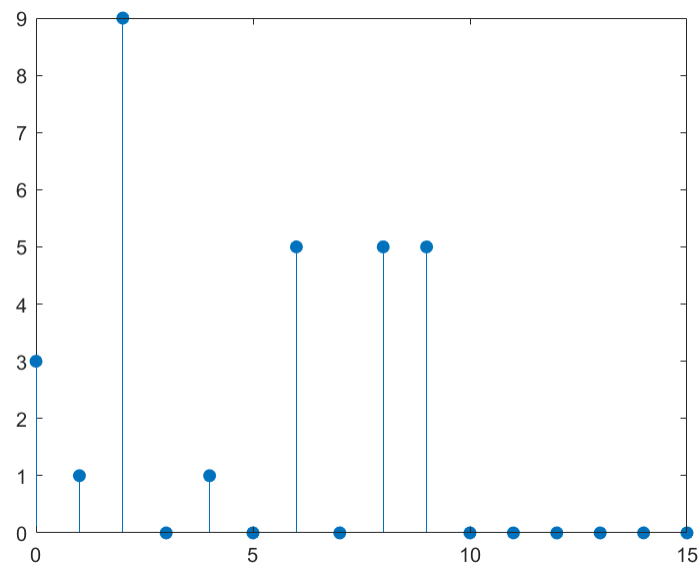


图 3: 学号

1	2	3	4	5	6	
17.0000+0.0000i	4.5239-12.4302i	2.7574+0.2426i	-3.2977-12.5950i	-5.0000+6.0000i	-6.3592+5.2040i	11.242
9	10	11	12	13	14	
17.0000+0.0000i	4.5239-12.4302i	2.7574+0.2426i	-3.2977-12.5950i	-5.0000+6.0000i	-6.3592+5.2040i	11.242

5. 对应的输出频谱序列为（列出数值，插入图形）

5.1 数值

5.2 图形

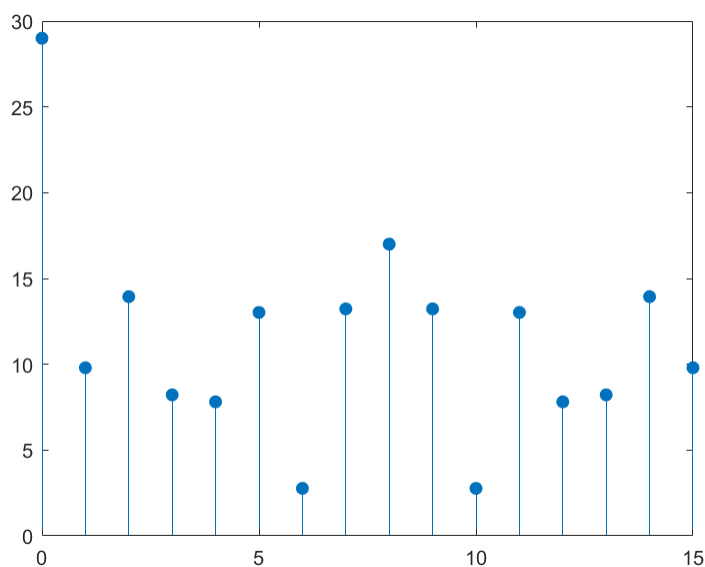


图 4: 频谱

六 实验结果与分析

与系统自带得 `fft` 函数结果进行比较后，可以看出结果具有正确性。

同时，同基 2 时域抽选法相比，基 4 所需的复乘次数降低，而复加次数则有所上升，因为复乘有着较大的运算开销，因此复乘次数的减少有利于改进 FFT 的计算效率