

中南大学

自动化学院本科生实验报告

数字信号处理

课程实验报告

专业 轨道交通与控制 班级 交控 1802 姓名 张浩波 学号 8212180912

预定时间 星期 节次 实际实验时间 星期 五 (2020.07.03) 节次

地点 台号 授课教师 陈志文 指导教师 陈志文

实验名称 用FFT作谱分析

一、实验原理、目的与要求

1. 实验目的

- (1) 进一步加深DFT算法原理和基本性质的理解(FFT作为DFT的一种快速算法,其运算结果必然满足DFT的基本性质)。
- (2) 熟悉FFT算法原理和FFT子程序的应用。
- (3) 学习用FFT对连续信号和时域离散信号进行谱分析的办法,了解可能出现的误差及其原因,以便在实际中正确应用FFT。

2. 实验原理

(1) DFT基本定义与性质

DFT是一种时域离散,频域也离散的离散傅里叶变换(DFT),DFT是针对有限长序列的,实际上,它可以被认为是取周期序列的离散傅里叶级数(DFS)的一个周期的对应关系来加以定义和研究的,这是一种具有隐含周期性的变换。

在介绍DFT定义前，我们先考察一下离散周期序列及其主值周期的概念，通常的，用 $\tilde{x}(n)$ 表示一个周期性的离散序列，设其周期为 N ，则有

$$\tilde{x}(n) \equiv \tilde{x}(n + rN), \forall r \in \mathbb{Z} \quad (1)$$

取 $\tilde{x}(n)$ 在区间 $\{0, 1, \dots, N-1\}$ 上的值，构造出的有限长序列称作 $\tilde{x}(n)$ 的主值序列，记 $x(n)$ 。对于那个区间 $\{0, 1, \dots, N-1\}$ （或写作 $0 \leq n \leq N-1$ ），称为主值区间，两者间的基本关系式为

$$\begin{aligned} x(n) &= \tilde{x}(n) \cdot R_N(n) \\ \tilde{x}(n) &= x(n)_N \end{aligned} \quad (2)$$

其中 $x(n)_N$ 表模余关系，即

$$x(n)_N = x(n \bmod N) \quad (3)$$

为了引出DFT的基本定义，我们先介绍DFS，即离散傅里叶级数，像连续傅里叶级数（CFS）那样，离散傅里叶级数的作用对象同样是具周期性的，但因其是离散的而与CFS有本质区别。它（DFS）把离散序列 $\tilde{x}(n)$ 表成各 k 次谐波（ $e^{j\frac{2\pi}{N}k}$ ）的线性组合

$$DFS[\tilde{x}(n)] = \tilde{X}(k) = \sum_{n=0}^{N-1} \tilde{x}(n) e^{-j\frac{2\pi}{N}nk} \quad (4)$$

其逆变换是

$$IDFS[\tilde{X}(k)] = \tilde{x}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}(k) e^{j\frac{2\pi}{N}nk} \quad (5)$$

为了计算和记号上的方便，常常用 $W_N = e^{-j\frac{2\pi}{N}}$ 进行替换，下面将经常性使用这个记号。

之所以 $\tilde{X}(k)$ 有 \sim 标志是因为其同样为周期序列，且周期亦为 N ，这容易由(4)式得到。重要的是， $\tilde{X}(k)$ 成为周期序列表明了DFS的谐波分量只有 N 个独立分量（对应于 k 取 $0, 1, \dots, N-1$ 时的基频序列 $\{e^{j\frac{2\pi}{N}k}\}$ ）。所以说， $\tilde{x}(n)$ 和 $\tilde{X}(k)$ 都是离散周期序列，只需要讨论在一个周期（最方便的一般是主值周期）的 N 个序列值就可以了。

离散傅里叶变换（DFT）与离散傅里叶级数（DFS）最主要的区别就是两者的研究对象有所不同，DFT研究的是有限长序列，或可认为是DFS研究的周期序列的主值序列，两者在计算上没有本质差别，DFT的正变换与逆变换分别为

$$\begin{aligned} DFT[x(n)] &= X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}, k = 0, 1, \dots, N-1 \\ IDFT[X(k)] &= x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-nk}, n = 0, 1, \dots, N-1 \end{aligned} \quad (6)$$

如果用先前的DFS表示这一关系，可以写成

$$x(n) = IDFT[X(k)] = IDFS\left[X((k))_N\right]R_N(n)$$

此外，我们可以从**频域抽样**的角度来理解DFT，再次注意到其正变换，即

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk} = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi}{N}nk} = \sum_{n=0}^{N-1} x(n)e^{-j\omega n}\bigg|_{\omega=\frac{2\pi}{N}k} \quad (7)$$

说明DFT的频谱相当于是对序列 $x(n)$ 的傅里叶变换的 N 点等距离采样，抽样间隔 $\frac{2\pi}{N}$ ，所以有下面的式子

$$X(k) = DFT[x(n)] = \mathfrak{X}[x(n)]\bigg|_{z=e^{j\frac{2\pi}{N}k}} = DTFT[x(n)]\bigg|_{\omega=\frac{2\pi}{N}k}, k = 0, 1, \dots, N-1$$

从这一点可以看出，我们常用的对序列补零操作，由于不会对 $x(n)$ 的傅里叶变换产生影响，所以说由此计算得到的 $DFT[x(n)]$ 的整体包络不会改变。但随着补零的个数增加， $X(k)$ 的序列长度增加，意味着频域采样变得更密。在后面我们会看增加补零的点数可以减小栅栏效应，也有利于利用FFT算法进行计算。

下面来讨论一下DFT的主要性质，由于DFT处理的是有限长序列，且隐含周期性，有一些不同于一般傅里叶变换的性质。这里主要考察圆周移位性质，圆周共轭对称性质，圆周翻褶性质和圆周卷积性质。

序列 $x(n)$ 的圆周移位，可以看作是先将 $x(n)$ 以周期 N 作周期延拓得 $\tilde{x}(n)$ ，随后进行移位，再对其取主值区间，即

$$x_m(n) = x((n+m))_N R_N(n) \quad (8)$$

移位后再进行DFT运算，有

$$X_m(k) = DFT[x_m(n)] = W_N^{-mk} DFT[x(n)] \quad (10)$$

然后我们来考虑圆周共轭对称性质，一个有限长序列 $x(n)$ 总可以写成一个圆周共轭偶对称序列 $x_{ep}(n)$ 和圆周共轭奇对称 $x_{op}(n)$ 的和。在 N 点长的区间内， $x_{ep}(n)$ 和 $x_{op}(n)$ 各自满足

$$\begin{aligned} x_{ep}(n) &= x_{ep}^*(N-n), n = 0, 1, \dots, N-1 \\ x_{op}(n) &= -x_{op}^*(N-n), n = 0, 1, \dots, N-1 \end{aligned} \quad (11)$$

上式出现了 $N-n$ 结构而非简单的 $-n$ ，可以限制 $x_{ep}(n)$ 和 $x_{op}(n)$ 总是定义在 $\{0, 1, \dots, N-1\}$ 上，如果我们将 $x(n)$ 绕在一个圆周上，那么 $x(N-n)$ 之于 $x(n)$ ，就像是反着（逆时针）来读原来的序列，这里的对称性，就可以称之为圆周对称性了。

前面说到，任意有限长序列 $x(n)$ 总可以写成一个圆周共轭偶对称序列 $x_{ep}(n)$ 和圆周共轭奇对称 $x_{op}(n)$ 的和，也就是 $x(n) = x_{ep}(n) + x_{op}(n)$ 。那么接下来的工作，就是完成对 $x_{ep}(n)$ 和 $x_{op}(n)$ 的求解了，对 $x(n)$ 取共轭并结合(11)式，有

$$\begin{aligned} x_{ep}(n) &= \frac{1}{2}[x(n) + x^*(N-n)] \\ x_{op}(n) &= \frac{1}{2}[x(n) - x^*(N-n)] \end{aligned} \quad (12)$$

讨论DFT运算的圆周共轭对称性，基本结论由下两个式子给出，其中 \downarrow 指出了对应的DFT变换对

$$\begin{array}{ccc} x(n) & = & Re[x(n)] + jIm[x(n)] \\ \downarrow & & \downarrow \quad \downarrow \\ X(k) & = & X_{ep}(k) + X_{op}(k) \end{array} \quad (13)$$

$$\begin{array}{ccc} x(n) = x_{ep}(n) + x_{op}(n) & & \\ \uparrow & \uparrow & \uparrow \\ X(k) = Re[X(k)] + jIm[X(k)] & & \end{array} \quad (14)$$

这就是圆周共轭对称性在DFT中的体现，利用对称性，我们常常可以简化计算，验证分析等。例如，如果 $x(n)$ 是实序列，那么 $X(k)$ 一定是圆周共轭偶对称的。

还需要补充的是，如果 $X(k)$ 圆周共轭偶对称，可进一步讨论 $Re[X(k)]$ 是圆周偶对称的， $Im[X(k)]$ 是共轭奇对称的；如果 $X(k)$ 圆周共轭奇对称，则 $Re[X(k)]$ 是圆周奇对称， $Im[X(k)]$ 是圆周偶对称。

关于圆周卷积，同样可以理解为两有限长序列（长度匹配到 N ）在整圆周上作卷积，圆周卷积的定义为

$$x_1(n) \textcircled{*} x_2(n) = \left[\sum_{m=0}^{L-1} x_1(m) x_2((n-m))_L \right] \cdot R_L(n), L \geq \max(N_1, N_2) \quad (15)$$

作用到频域上，有

$$DFT[x_1(n) \textcircled{*} x_2(n)] = DFT[x_1(n)] \cdot DFT[x_2(n)] \quad (16)$$

(2) DFT谱分析及相关误差分析

我们来看一下如何利用DFT对离散序列，以及连续信号（经抽样）进行频谱分析。

首先我们来看最简单的情形——**对有限长序列作频谱分析**由于DFT是直接于有限长序列 $x(n)$ 的，所以按照矩阵方法或者是快速算法FFT直接求解即可，这里要求的频谱通常是 $x(n)$ 的傅里叶变换（DTFT），根据前面(7)式所表现的，即DFT相当于对DTFT频谱的 N 点等距离抽样，选取适当的点数即可体现出频谱 $X(e^{j\omega})$ 特征。在实践中， N 通常选2的整数次幂，多的补零以代替，若计算分辨率不够增加一倍点数重试即可。

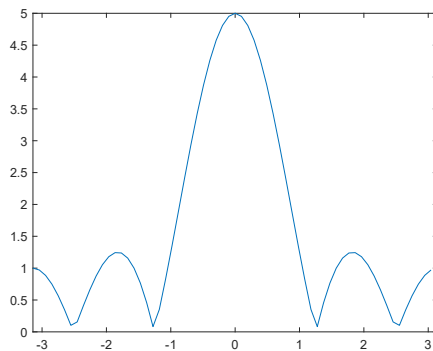
下面来看对**无限长序列频谱分析**的基本方法，计算机不可能处理无限长数据，所以必须有截断(截断可以理解为加窗的特例——加矩形窗)，这将带来频谱泄露。一般地，我们把原序列记为 $x(n)$ ，取 N 点长后的截断序列为 $x_N(n)$ ，有

$$x_N(n) = x(n) R_N(n) \quad (17)$$

作傅里叶变换（DTFT），利用卷积定理有

$$X_N(e^{j\omega}) = \frac{1}{2\pi} X(e^{j\omega}) \cdot R_N(e^{j\omega}) \quad (18)$$

其中 $R_N(e^{j\omega}) = e^{-j\frac{N-1}{2}\omega} \sin\left(\frac{\omega N}{2}\right) / \sin\left(\frac{\omega}{2}\right)$ ，其幅度谱如下图所示，其中主瓣宽度为 $4\pi/N$ ，两侧各有许多旁瓣， N 越大，旁瓣越多，主瓣越狭窄。



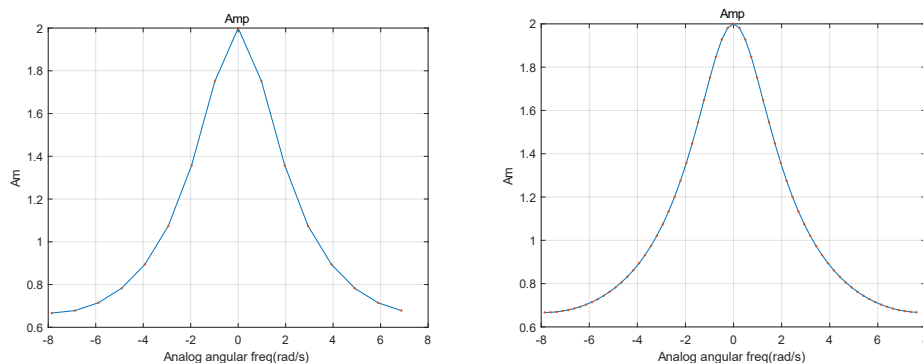
当矩形窗于其发生作用后，可以想见 $X_N(e^{j\omega})$ 与 $X(e^{j\omega})$ 并不会完全一致，这将首先造成频谱泄露，意味着对于原信号中离得很近两信号，与 $R_N(e^{j\omega})$ 卷积后将不能够分辨开。而 N 越大则主瓣宽度 $\frac{4\pi}{N}$ 越小，展宽得越窄，泄露越少。

另一方面，这还会造成谱间串扰，由于旁瓣的存在，它便有可能使原信号中的两频率分量，强信号的旁瓣掩盖住了弱信号的主瓣，使其被忽视，这就造成了频率分辨率的降低。对于模拟信号的抽样来讲，这时就要适当增大抽样频率 f_s 。

为了减轻截断效应，通常有两种做法，一种是可以采取缓变的窗函数，这样旁瓣幅度小，干扰量小，但主瓣宽度会增大，也会降低频率分辨率，需要予以权衡；还可以增加截断点数 N （也即增加截断长度 $T_0 = N/f_s$ ），降低主瓣宽度。

在操作中，为了确定逼近误差的程度，也采用下面的办法，就是分别取定两序列 $x_1(n)$ 与 $x_2(n)$ ，它们的截断点数分别为 N_1, N_2 ，为了方便操作有 $N_2 = 2N_1$ 。然后，我们计算前后两次最大的绝对误差与最大的频谱绝对值的比值，与事先规定的相对误差 ϵ 作比较，若符合要求则取这次的运算结果。

下面两幅图是书上 P256 页 例 4.9 的图例，分别取相对误差 $\beta = .5\%$ 和 $.005\%$ 。通常来看，截断点数越大，截断长度越大，我们能获得的精度就越高。



最后我们来看一下利用DFT对模拟信号频谱的分析，首先回顾模拟信号抽样的一些基本概念，抽样频率是 f_s ，它与抽样间隔 T 互为倒数；频率分辨率指长度为 N 的信号序列中所对应的连续谱中能分辨的

两个频率分量峰值的最小频率间距 F_0 ，它与总的的数据长度 T_0 互为倒数，若不做补零处理，有

$$T_0 = NT = \frac{N}{f_s} \quad (19)$$

需要注意的是，通过补零的办法可以增加DFT计算的频率分辨率，但由于没有增加有效长度，并不能提高频率分辨率。

随后，我们看一下用DFT对模拟信号频谱的分析基础，对于模拟信号 $x(t)$ 的傅里叶变换（CFT），即

$$X(j\Omega) = \int_{-\infty}^{+\infty} x(t)e^{-j\Omega t} dt \quad (20)$$

我们用矩形和逼近整个积分，也就是

$$X(j\Omega) \approx \sum_{n=-\infty}^{\infty} x(nT)e^{-j\Omega nT} \cdot T \quad (21)$$

抽样间隔 T 越小，(21)右边就越对 $X(j\Omega)$ 逼近。由于DFT是对有限长序列做处理，所以还需要截断的步骤，也就是

$$X(j\Omega) \approx T \sum_{n=0}^{N-1} x(nT)e^{-j\Omega nT} \quad (22)$$

其中 N 是截断点数，截断长度，或者说总的的数据长度 $T_0 = NT$ ，最小频率间距为 $F_0 = 1/T_0$

最后，我们要对频域离散化，或者说是频域上抽样，以 $k\Omega_0, k = 0, 1, \dots, N-1$ 代之，则有

$$X(jk\Omega_0) \approx T \sum_{n=0}^{N-1} x(nT)e^{-jk\Omega_0 nT} \stackrel{\Omega_0=2\pi F_0=\frac{2\pi}{NT}}{=} T \sum_{n=0}^{N-1} x(nT)e^{-j\frac{2\pi}{N}nk} = T \cdot DFT[x(n)] \quad (23)$$

当频域抽样足够密的时候，我们就可以利用 $T \cdot DFT[x(n)]$ 的包络逼近 $X(j\Omega)$

在实践中，可能出现下列误差现象

首先是频谱的混叠失真，由奈奎斯特抽样定理，抽样频率要高于信号最高频率的二倍，即 $f_s \geq 2f_h$ ，实际中不容易清晰知道 f_h 确切值甚至 f_h 不可求，所以通常要求 $f \leq f_s/2$ 的范围内包含98%以上的信号能量（尤其对于无限带宽情形），且在 $f_s = (3 \sim 6)f_h$ 范围内选取抽样频率。

前面说到的截断产生的频谱泄露，也要求我们要更高的频率分辨率，由于 $F_0 = N/f_s$ ，若固定 N ，则增大 f_s 同样也是减弱频谱泄露的办法。

其次可能出现的由截断产生的频谱泄露情况，前面已经介绍过了，还有一种情况值得注意，那就是栅栏效应，在 N 点抽样值任意相邻两点之间的频率点上的频谱值是不知道的，这就导致了获取得到频谱的片面性。要减小栅栏效应，可以在 T_0 不变的条件下同时增加 f_s 和 N ，还可以通过补零的办法增加序列点数。这两种做法都没有改变 $x(n)$ 的有效数据，但是使频域抽样变密，从而能够看到更多的频率上的频谱。

需要说明，对于周期信号而言，必须使抽样后的序列仍是周期的，且截断数据长度 T_0 应该是周期序列周期的整数倍，否则就会产生频谱泄露现象，本实验中第五个序列即是对周期信号进行抽样，并利用

DFT进行分析, 应格外注意这一点。

在实际操作中, 我们可以采取逐步试错的办法调整相关参数 T, T_0, N 的配置, 一般可以这样做:

- ① 先固定数据长度 $T_0 = NT$ 不变, 逐步减小 T , 观察频谱混叠的影响是否足够小
- ② 第一步确定 T (也就是确定 f_s 后), 增加 T_0 (也就是增大有效序列长度 N), 使得截断效应减小, 增大频率分辨率, 观看此时频谱差别是否很小
- ③ 最后, 看栅栏效应是否明显, 且在上述两步, N 应时常通过补零操作使其为2的整数幂。若有必要, 最后再进行一次补零。

$$\text{Review: } X(e^{j\omega}) \approx T \cdot \text{DFT}[x(n)]$$

$$\Omega = \frac{2\pi}{NT} \cdot k$$

(3) FFT快速算法

现在来看一下计算DFT的快速算法FFT。FFT有DIT (基于时间) 和DIF(基于频率)两种。基于时间的抽选法的每一步分解都是按照输入序列在时间上的次序是输入奇序列还是偶序列来分解为两个更短的子序列, 所以称为“按时间抽选法” (DIT); 而按频率的抽选法则是对频率按上述规则划分。DIT和DIF在各级蝶形的结构上有本质的区别。

下面我们来看基2时间抽选法的整个过程, 如前所述, 我们对 $x(n)$ 按 n 的奇偶来分成两部分, 其中

$$\begin{aligned} x_1(r) &= x(2r), \quad r = 0, 1, \dots, N/2 - 1 \\ x_2(r) &= x(2r + 1), r = 0, 1, \dots, N/2 - 1 \end{aligned} \quad (24)$$

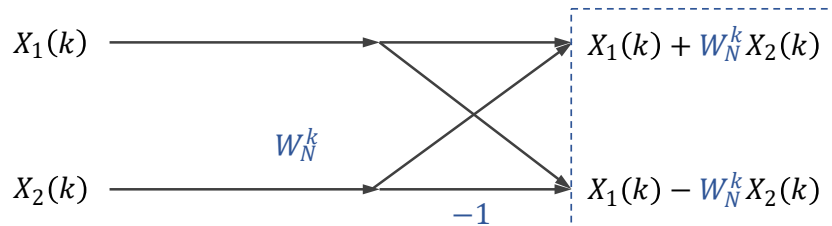
要求 N 是2的整数幂, 也就有 $N = 2^P$, 这样 $x_1(r)$ 与 $x_2(r)$ 就都是 $N/2$ 点长序列。随后, 我们求 $X(k)$, 拆分成两部分 (分别是关于 $X_1(k)$ 和 $X_2(k)$ 来看), 有

$$\begin{aligned} X(k) &= \text{DFT}[x(n)] = \sum_{n=0}^{N-1} x(n)W_N^{nk} \\ &= \sum_{\substack{n \text{ 是偶数} \\ n=0}}^{N-1} x(n)W_N^{nk} + \sum_{\substack{n \text{ 是奇数} \\ n=0}}^{N-1} x(n)W_N^{nk} \\ &= \sum_{r=0}^{\frac{N}{2}-1} x(2r)W_N^{2rk} + \sum_{r=0}^{\frac{N}{2}-1} x(2r+1)W_N^{(2r+1)k} \\ &= X_1(k) + W_N^k X_2(k) \end{aligned} \quad (25)$$

需要注意的是, $X_1(k)$ 和 $X_2(k)$ 都是 $N/2$ 点长序列, 而 $X(k)$ 则为 N 点长序列, 对于 $X(k)$ 的后一部分, 我们取 $k + N/2, k = 0, 1, \dots, N/2 - 1$ 代入, 并考虑性质 $W_N^{\frac{N}{2}+k} = W_N^{\frac{N}{2}}W_N^k = -W_N^k$, 有

$$X(k) = X_1(k) - W_N^k X_2(k), k = 0, 1, \dots, \frac{N}{2} - 1 \quad (26)$$

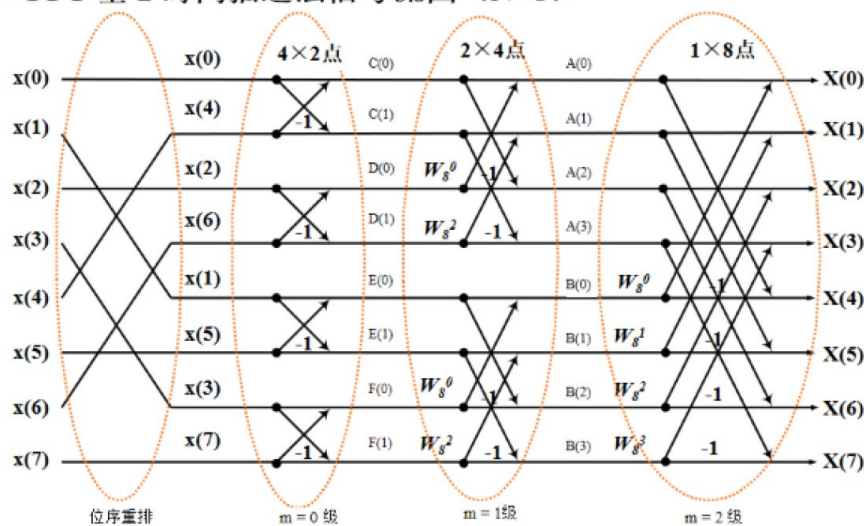
(25)与(26)表明的, 可以用下面的蝶形运算代替



可以观察到，一个蝶形结构中包含两次复加，一次复乘。对于 $N/2$ 点长的 $X_1(k)$ 与 $X_2(k)$ ，我们可以进一步拆分成 $N/4$ 点长序列的蝶形运算，最终直到要进行 2 点 DFT 不能再拆分为止。对于 2 点 DFT，由于 $W_2^0 = 0, W_2^1 = -1$ ，所以事实上只需进行两次复加即可。

另外，由于每次蝶形运算需要按奇偶划分，所以最开始应完成位序的重排，以 $N = 8$ 为例，完整的 FFT 算法如下图所示

■ FFT 基 2 时间抽选法信号流图 ($N=8$)



整得来看， $N = 2^P$ 点长序列 $x(n)$ FFT（基 2 DIT）快速算法共有 P 级蝶形结构，每级都有 $N/2$ 个蝶形，每个蝶形有 2 次复加，1 次复乘。所以每级的运算量是 $N/2 * 2 = N$ 次复加， $N/2$ 次复乘（第 0 级除外，因为 2 点 DFT 事实上无需复数加法）

所以 DIT - FFT 的总运算量为

$$\begin{aligned} \text{复乘次数: } & \frac{N}{2} \log_2 N \\ \text{复加次数: } & N \log_2 N \end{aligned} \quad (27)$$

FFT 快速算法的时间复杂度是 $O(n \log n)$ 级别的，当序列长度很大时，可以有效提高计算效率。

我们再简单介绍一下 DIF 的基本原理，类似地，我们把 $X(k)$ 按奇偶划分，有

$$\begin{aligned} X(2r) &= \sum_{n=0}^{\frac{N}{2}-1} x_1(n) W_N^{nr}, r = 0, 1, \dots, \frac{N}{2} - 1 \\ X(2r+1) &= \sum_{n=0}^{\frac{N}{2}-1} x_2(n) W_N^{nr}, r = 0, 1, \dots, \frac{N}{2} - 1 \end{aligned} \quad (28)$$

DIT-FFT和DIF-FFT的运算量是相同的, 当 $N = 2^P$ 时都为 P 级蝶形结构, 每级都有 $N/2$ 个蝶形结, 如式(27)所示。

二、实验仪器设备(标注实验设备名称及设备号)

装有 Matlab R2018a 的计算机一台

三、实验线路示图、内容步骤

(1) 设计序列发生程序

这一部分主要是完成实验书给出的 x_1 至 x_6 序列, 是较为简单的。 x_6 是一个连续信号, 所以这个过程就是抽样, 可以先构造句柄

```
f=@(t) cos(8*pi*t) + cos(16*pi*t) + cos(20*pi*t);
```

随后进行抽样即得

```
x_6 = f(n*T);
```

其余离散序列生成就较为简单, 这里就不再赘述了

(2) 设计DFT分析程序

这里的关键之处在于函数 `fft_anlysis` 的设计, 整体的输入输出为

```
function [k, x_n, mag_X_k, ang_X_k, re_X_k, im_X_k] = fft_anlysis(x_n, N, anylsis)
```

x_n , N 分别是分析参数, 对应于要分析的序列, 以及计算点数 N (若长度不够则补零), `anlysis` 是一个布尔型变量, 传入 `1` 代表在内部即完成绘图分析, 传入 `0` 表示不进行绘图, 只进行相关计算。

在函数内部, 调用内置函数 `fft` 完成相关计算, 这里简单介绍一下 Matlab 自带函数 `fft` 的用法, 函数 `fft` 接受一个向量 x , 计算它的DFT变换列, 若传入矩阵 x , 则完成对各列的DFT计算。若 x 的长度是2的整数幂, 则进行基2 FFT算法, 否则进行稍慢的混合基算法, 若其长度是素数, 则 `fft` 采取原始的DFT算法 (即构造矩阵 W 进行DFT运算)

这里的关键步骤是对 x_n 利用FFT进行运算

```
X_k = fftshift(fft(x_n));
```

注意到这里还有 `fftshift` 的使用, 实际上 `fftshift` 是将计算完成的FFT频谱 X_k 的后半部分, 也就是对应于数字频率 $\pi < \omega \leq 2\pi$ 的部分移到 $-\pi < \omega \leq 2\pi$ 。从结果上看, 是将频谱由 $[0, 2\pi)$ 移动到 $[-\pi, \pi)$ 上去, 其实借助了 X_k 序列隐含的周期性。在这里, 为了让序列 X_k 前后对应, 在其末尾 (也就是对应于 $\omega = \pi$) 的地方, 补充上首位元素 (对应于 $\omega = -\pi$) 使其成为 $N + 1$ 点序列, 更方便观看其圆周对称性。

```
X_k(N+1) = X_k(1);
```

当然，对应的数字频率序列也有补充一位，即

```
k = floor(-(N-1)/2:(N+1)/2);  
k = k .* (2 * pi / N);
```

(在一般情形下，k 的写法是 $\text{floor}(-(N-1)/2:(N-1)/2)$)

同时我们要对序列 k 乘上 $2\pi/N$ 以其频率抽样间隔为 $2\pi/N$ 。这样就完成了数字域上的一一对应。

当我们要将数字域变换到模拟域时，需要利用关系

$$\omega = \Omega/f_s$$

所以可以看出在 x_6 的频谱分析中，有下面这步

```
plot(k*Fs, mag_x, '--'); % ...
```

(3) 设计实验二，实验三比较程序

实验二要求我们由 $x(n) = x_4(n) + x_5(n)$ 的 DFT 序列得到分别的 $DFT[x_4(n)]$ 与 $DFT[x_5(n)]$ ，这里需要利用其圆周共轭对称性质，通过观察，可以发现，当 $N = 16$ 时，有

$$x_4(n) = x_4(N-n), x_5(n) = -x_5(N-n)$$

如果我们将 $x(n)$ 拆分成圆周共轭偶序列与圆周共轭奇序列，就有 $x_{ep}(n) = x_4(n), x_{op}(n) = x_5(n)$ 。不过，对 $N = 8$ 的情形，由于 $x_4(n)$ 与 $x_5(n)$ 都是圆周共轭偶序列，就无从分别开了。

按照 (14) 式中性质，我们可以得到 $DFT[x_4(n)] = \text{Re}[X(k)], DFT[x_5(n)] = j\text{Im}[X(k)]$ 其中 $X(k) = DFT[x(n)] = DFT[x_4(n) + x_5(n)]$

利用前面的 `fft_anlysis` 函数计算封装好的 X_k ，然后作如下比较 $\text{Re}[X(k)] = DFT[x_4(n)], j\text{Im}[X(k)] = DFT[x_5(n)]$ 是否成立，为了比较上的方便（也是引用上的方便，取模比较）

```
subplot(2, 2, 1)  
plot(k, re_X_k)  
title(' |Re[X(k)] | ')  
subplot(2, 2, 2)  
plot(k, abs(im_X_k))  
title(' |Im[X(k)] | ')  
subplot(2, 2, 3)  
plot(k, mag_x4)  
title(' |X_(ep)(k)| (|X_4(k)|) ')  
subplot(2, 2, 4)  
plot(k, mag_x5)  
title(' |X_(op)(k)| (|X_5(k)|) ')
```

对于实验三，构造 $x(n) = x_4(n) + jx_5(n)$ ，由 (13) 式可得 $DFT[x_4(n)] = X_{ep}(k), DFT[jx_5(n)] = X_{op}(k)$ ，这里的关键在于对给定序列圆周共轭偶序列和奇序列的求解。

这里构造函数 `circevod(x)` 用来计算序列 x 的圆周共轭偶序列和奇序列，其他比较地方与前相同。

```
xep = .5*(x +conj(x(mod(-n, N)+1)));
xop = .5*(x -conj(x(mod(-n, N)+1)));
```

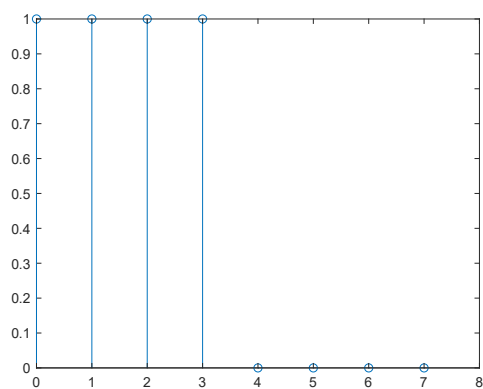
四、实验数据记录及数据处理

图片地址见 <https://github.com/pacria/MatlabExp/tree/master/MatlabExp/DSP/exp2/checkedSVG>

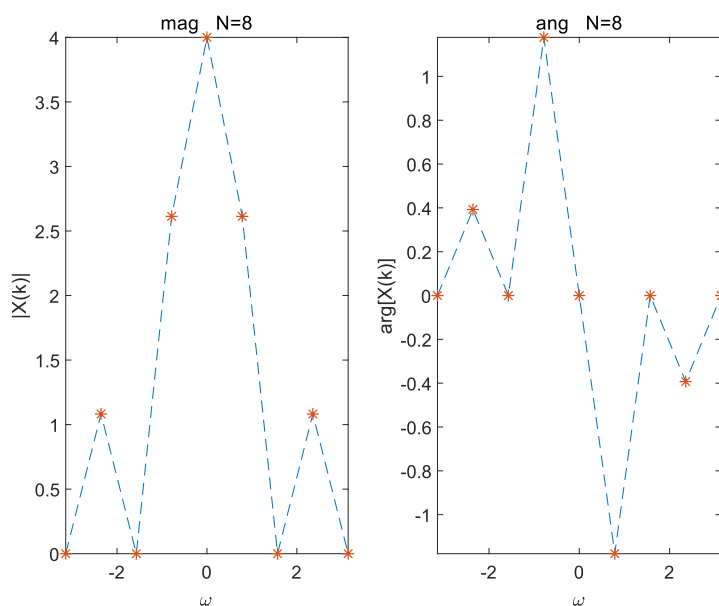
(2) EXP1

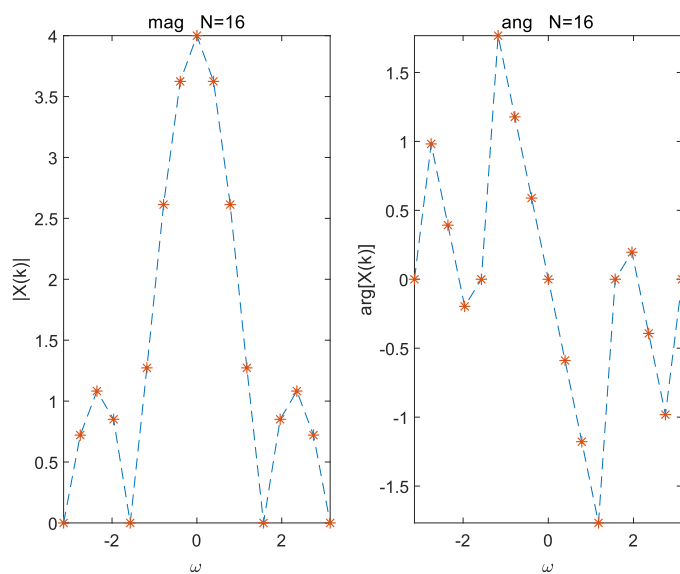
实验一要求对给出的各信号做谱分析, x_1 至 x_5 均为离散序列, 分析方法相同; x_6 是连续信号, 需要进行抽样。

先考虑 $x_1 = R_4(n)$, 作图为

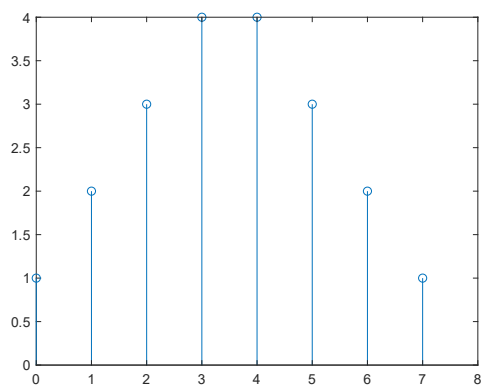


直接给出其 n 点DFT为 $X(k) = W_N^{\frac{3k}{2}} \sin\left(\frac{4\pi k}{N}\right) / \sin\left(\frac{\pi k}{N}\right)$, 当 $n = 8$ 和 $n = 16$ 时, 其DFT为

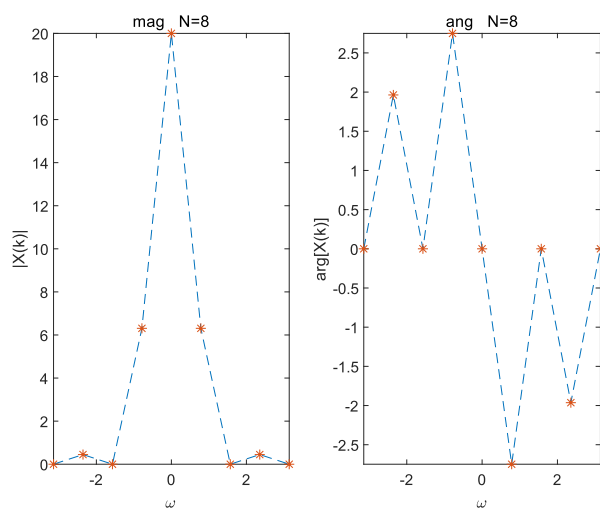


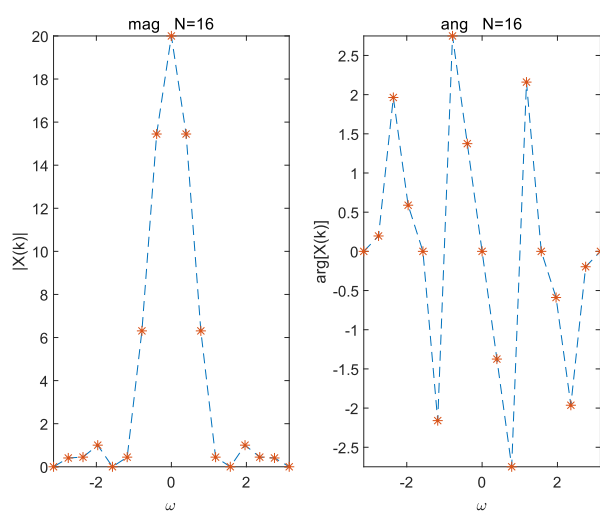


对于 $x_2(n) = \{1, 2, 3, 4, 4, 3, 2, 1\}$, 同样的方法作图得

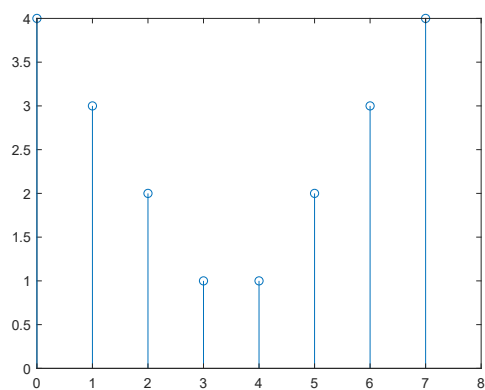


其8点DFT和其16点DFT分别为

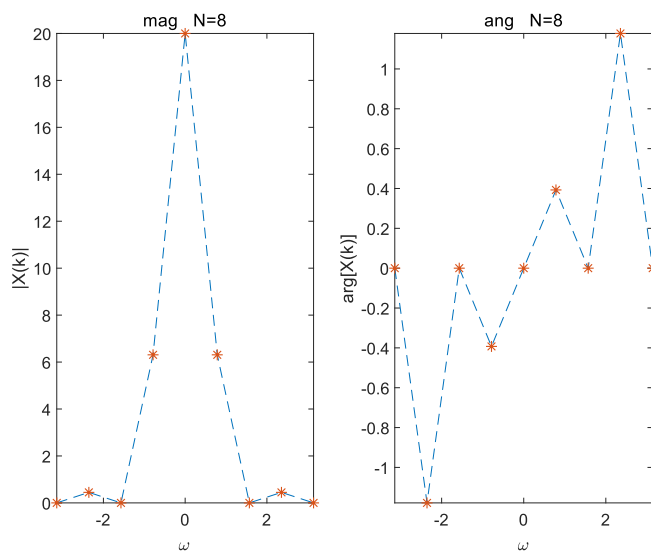


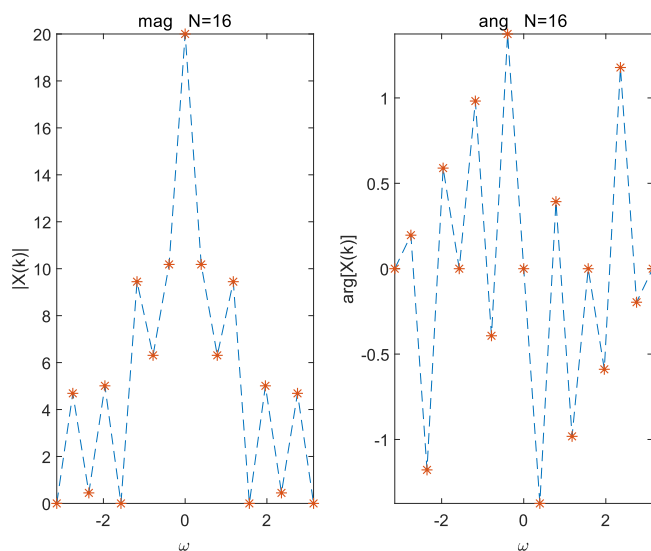


对于 $x_3(n) = \{4, 3, 2, 1, 1, 2, 3, 4\}$, 作图结果为



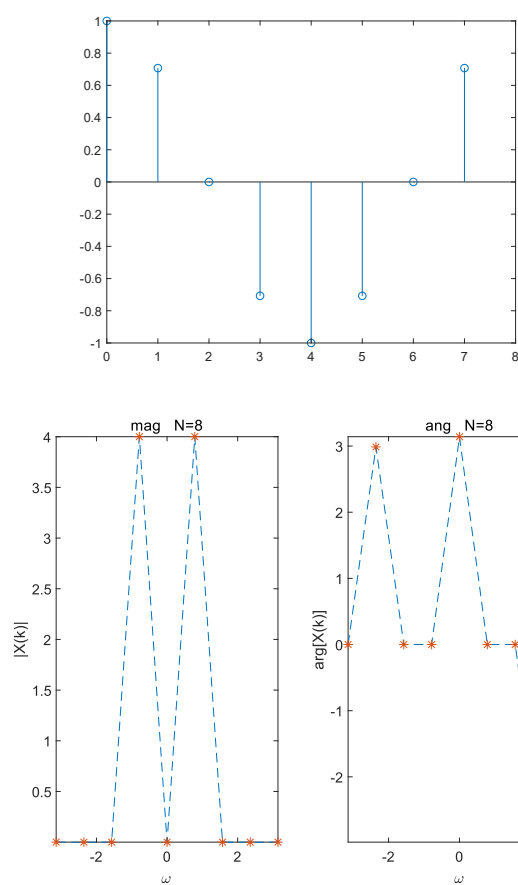
其8点DFT和其16点DFT分别为



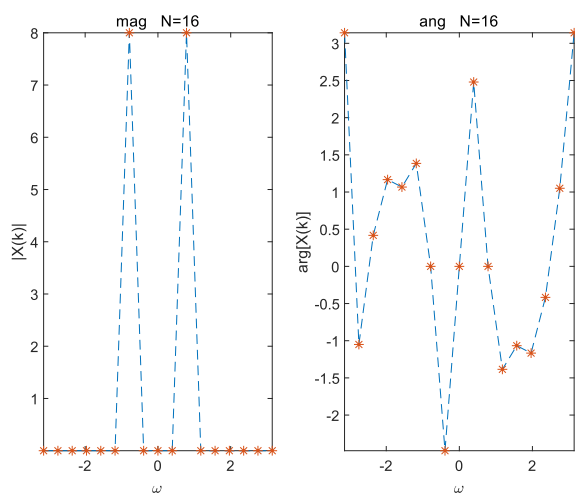
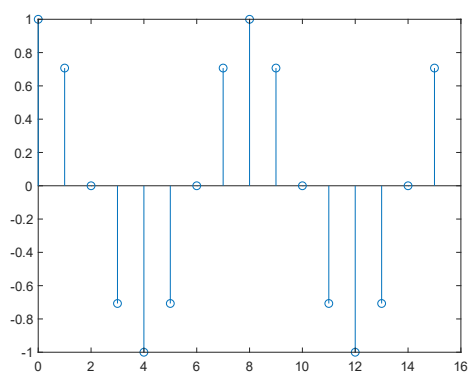


对于 $x_4(n) = \cos(n\pi/4)$ ，这是一个周期为8的离散序列，其8点DFT和其16点DFT由于其长度都是周期的整数倍，所以都可以注意到在 $2\pi k/N = \pi/4$ 处有峰值，无频谱泄露。

下面是取 $N = 8$ 的情形

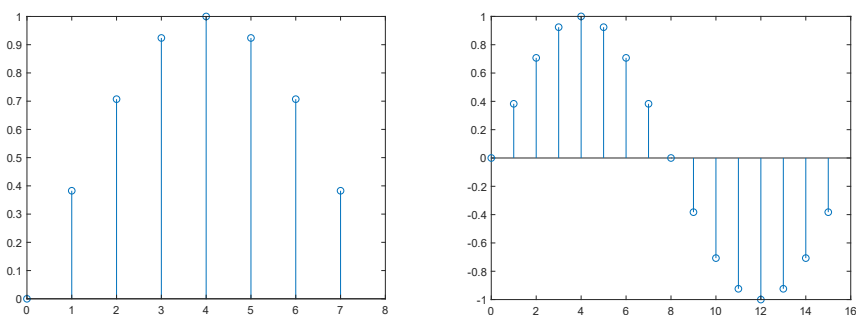


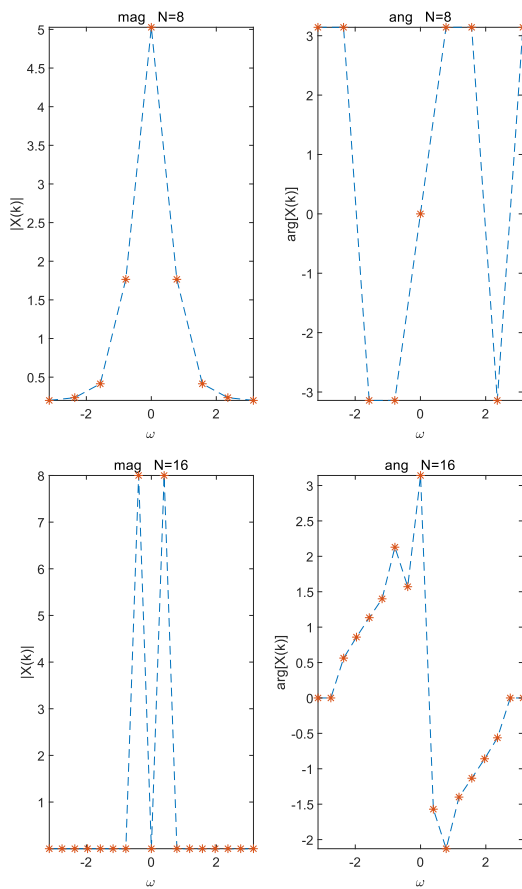
下面是取 $N = 16$ 的情形



对于 $x_5(n) = \sin(n\pi/8)$ ，这是一个周期为16的离散序列，其8点DFT由于不能包含整主值周期，所以会发生频谱泄露，而当 N 取16则无次情形发生。

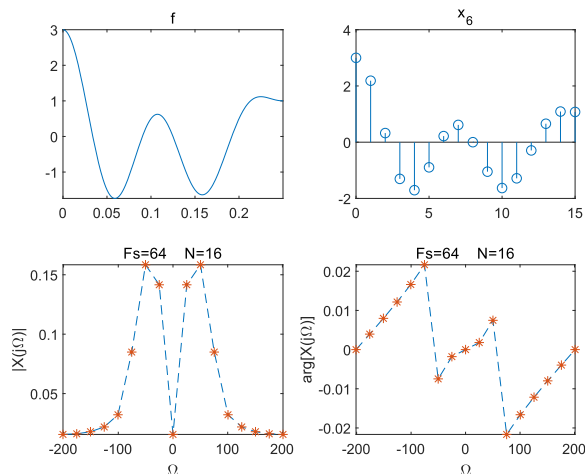
当 $N = 8$ 和当 $N = 16$ 时，作图为

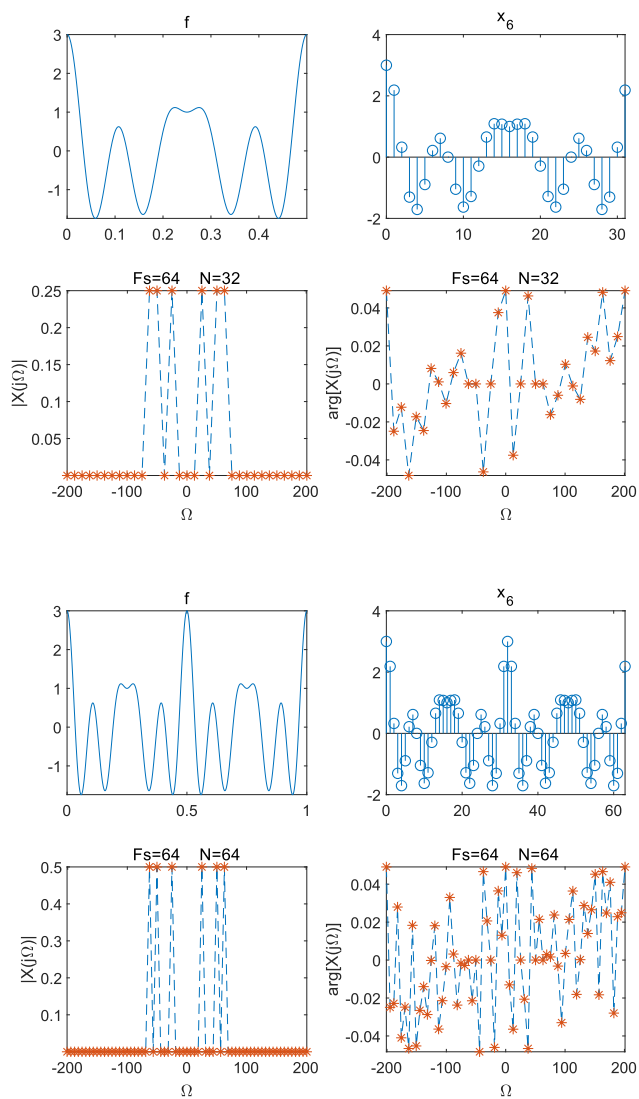




对于 x_6 序列，要先对 $f(t) = \cos 8\pi t + \cos 16\pi t + \cos 20\pi t$ 进行抽样，首先考虑到 $f(t)$ 的最高频率 $f_h = 10\text{Hz}$ 所以 $f_s \geq 20\text{Hz}$ 且根据实际，应在 $f_s = (3\sim 6)f_h$ 里取，所以取 $f_s = 64\text{Hz}$ 没有问题，通过后续结果看也是合适的。

取 $t = nT$ 抽样后，序列 $x_6 = \cos n\pi/8 + \cos n\pi/4 + \cos 5n\pi/16$ 的周期为 $N = 32$.前面 分析误差时说到，要让取值点数时周期数的整数倍才可行，不会发生频谱泄露，所以 $N = 16$ 会发生频谱泄露， $N = 32$ 与 $N = 64$ 较为合适，可以观察到三个峰值



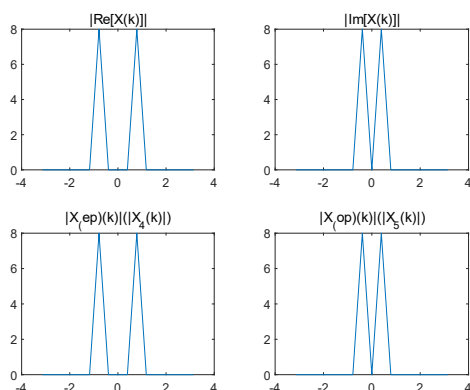


(3) EXP2

记 $x(n) = x_4(n) + x_5(n)$, 验证等式

$$DFT[x_4(n)] = \text{Re}[X(k)], DFT[x_5(n)] = j\text{Im}[X(k)]$$

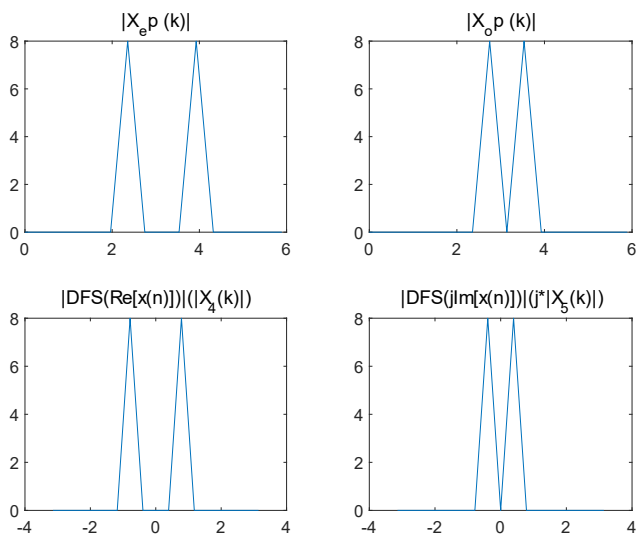
作图直观来看:



EXP3

记 $x(n) = x_4(n) + jx_5(n)$, 验证等式

$$DFT[x_4(n)] = X_{ep}(k), DFT[jx_5(n)] = X_{op}(k)$$



结果符合我们的预期。

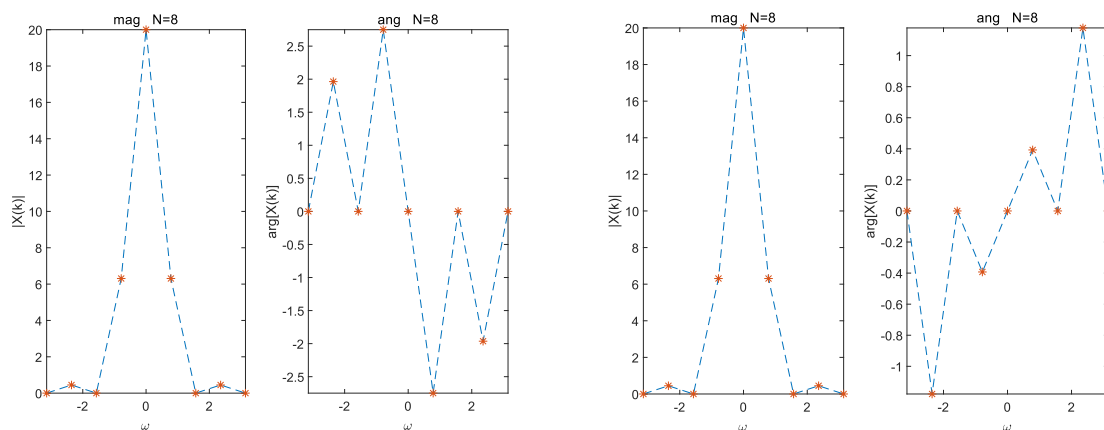
五、实验结论探讨及分析

通过本次实验, 我们回顾了有关DFT的基本性质, 尤其是与圆周对称相关的内容, 包括圆周共轭对称性, 圆周翻褶性质, 圆周移位性质等。掌握了DFT谱分析的基本办法, 包括在对离散序列与连续信号傅里叶变换频谱的逼近。了解了主要包括混叠失真, 频谱泄露, 谱间串扰, 栅栏效应等一系列可能出现的误差, 学会了相关参数的基本调节做法。还有DFT的快速算法——FFT的基本原理的掌握。

思考题:

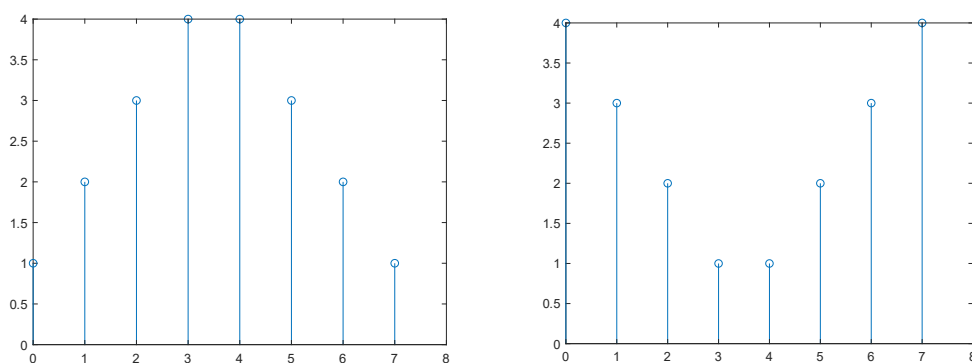
- (1) 当 $N = 8$ 时, $x_2(n)$ 与 $x_3(n)$ 的幅频特性是否相同? 当 $N = 16$ 时又是怎样的?

对于这个问题, 我们可以首先进行观察, 对比查看一下其DFT变换结果



从图像上直观看来, 应该有 $|X_2(k)| = |X_3(k)|$, 如果查看工作区变量, 可以同样验证该说法。事实上, 利用DFT的圆周移位定理, 我们可以得到该结论。

这是因为 $x_2(n)$ 与 $x_3(n)$ 事实上有圆周移位关系, 当 $N = 8$ 时, 注意有 $x_2((n-4))_N = x_3((n))_N$



再同时进行DFT, 得到

$$W_N^{4k} X_2(k) = X_3(k) \quad (30)$$

再同时取模, 就有 $|X_2(k)| \cdot |W_N^{4k}| = |X_3(k)|$ 。此时 $|W_N^{4k}| = |W_8^{4k}| = |W_2^k| = |\pm 1| = 1$, 就有 $|X_2(k)| = |X_3(k)|$ 成立。

当 $N = 16$, 不同的是由于补零操作, 原有的圆周移位性质就失去了, 也即是不再有 $x_2((n-4))_N = x_3((n))_N$ 关系了, 所以我们看到此时不存在 $|X_2(k)| \equiv |X_3(k)|$

(2) 如果周期信号的周期预先不知道，如何进行频谱分析？

实际操作中，我们只能进行多次尝试，以逼近原模拟信号的频谱，这里的关键参数是抽样频率 f_s (或者抽样间隔 T)，以及数据长度 T_0 (或有效点数 N)，按照在实验原理中表明的做法，我们可以有如下步骤

① 固定 N ，调整 T ，使得没有明显的频谱混叠现象为止

② 确定抽样频率后，调整 N ，也就是增大数据长度 T_0 以减小截断带来的不利影响，直至频谱的差别很小时

通过多次尝试，我们可以得到误差允许内的频谱（截断的影响，参考本报告 P4 页说明的一个例子）

六、附录

代码地址见 <https://github.com/pacria/MatlabExp/tree/master/MatlabExp/DSP/exp2>

图片地址见 <https://github.com/pacria/MatlabExp/tree/master/MatlabExp/DSP/exp2/checkedSVG>

函数定义区

```
function [k, x_n, mag_X_k, ang_X_k, re_X_k, im_X_k, X_k] = fft_anlysis(x_n, N, anylisis)
M = length(x_n);
x_n = [x_n zeros(1, N-M)];

X_k = fftshift(fft(x_n));
X_k(N+1) = X_k(1); % 多补充一位以利于体现对称性（在  $n=(N+1)/2$  处补上  $n=-(N-1)/2$  的值 相当于将
序列在圆周上\pi处排列的值重复一遍）

k = floor(-(N-1)/2:(N+1)/2);
k = k .* (2 * pi / N);

mag_X_k = abs(X_k);
ang_X_k = angle(X_k);

re_X_k = real(X_k);
im_X_k = imag(X_k);

if anylisis
    subplot(1, 2, 1)
    plot(k, mag_X_k, '--');
    axis([-pi, pi, -inf, inf]);
    xlabel('\omega')
    ylabel('|X(k)|')
    title(['mag ', 'N=', num2str(N)])
    hold on
```

```
plot(k, mag_X_k, '*');
hold off

subplot(1, 2, 2)
plot(k, ang_X_k, '--');
axis([-pi, pi, -inf, inf]);
xlabel('\omega')
ylabel('arg[X(k)]')
title(['ang ', 'N=', num2str(N)])
hold on
plot(k, ang_X_k, '*');
hold off

end
end
function [xep, xop] = circevod(x)
    N = length(x); n = 0:N-1;

    xep = .5*(x + x(mod(-n, N)+1));
    xop = .5*(x - conj(x(mod(-n, N)+1)));
end
```

预置序列区

```
clc;
clear
N1 = 8;
N2 = 16;
N = N2;
n = 0:N-1;
x1 = ones(1, 4);
x2 = [1:4 4:-1:1];
x3 = [4:-1:1 1:4];
x4 = cos(pi/4 * n);
x5 = sin(pi/8 * n);
```

EXP1-ANALYSIS

x_n Below:

```
x_n = x3;
[k, x_n, ~] = fft_anlysis(x_n, N, 1);
figure
stem(n, x_n);
axis([0 N -inf inf])
```

$f(t)$ Below:

```
% 对 x_6 进行谱分析
clc;
clear
```

```
f = @(t) cos(8*pi*t) + cos(16*pi*t) + cos(20*pi*t);
Fs = 64;
T = 1/Fs;
N = 64; % 32, 64
n = 0:N-1;
x_6 = f(n*T);
[k, x_6, mag_x, ang_x, ~] = fft_anlysis(x_6, N, 0);
mag_x = T*mag_x; ang_x = T*ang_x;
subplot(2, 2, 1)
fplot(f, [0, N*T])
title('f')
subplot(2, 2, 2)
stem(n, x_6)
title('x_6')
subplot(2, 2, 3)
plot(k*Fs, mag_x, '--');
axis([-Fs*pi, Fs*pi, -inf, inf])
xlabel('\Omega')
ylabel('|X(j\Omega)|')
title(['Fs=', num2str(Fs), '    N=', num2str(N)])
hold on
plot(k*Fs, mag_x, '*')
hold off
subplot(2, 2, 4)
plot(k*Fs, ang_x, '--');
axis([-Fs*pi, Fs*pi, -inf, inf])
xlabel('\Omega')
ylabel('arg[X(j\Omega)]')
title(['Fs=', num2str(Fs), '    N=', num2str(N)])
hold on
plot(k*Fs, ang_x, '*')
hold off
```

EXP2-ANALYSIS

```
% EXP-2
N = 16;
x_n = x4+x5;
[k, x_n, mag_X_k, ang_X_k, re_X_k, im_X_k] = fft_anlysis(x_n, N, 0);
[k, x4a, mag_x4, ~] = fft_anlysis(x4, N, 0);
[k, x5a, mag_x5, ~] = fft_anlysis(x5, N, 0);
subplot(2, 2, 1)
plot(k, re_X_k)
title('|Re[X(k)]|')
subplot(2, 2, 2)
plot(k, abs(im_X_k))
title('|Im[X(k)]|')
```

```

subplot(2, 2, 3)
plot(k, mag_x4)
title('|Xep(k)|(|X4(k)|)')
subplot(2, 2, 4)
plot(k, mag_x5)
title('|Xop(k)|(|X5(k)|)')

```

EXP3-ANALYSIS

```

N = 16;
n = 0:N-1;
D = 2*pi/N;
n = D*n;
x_n = x4+1j* x5;
X_k = fft(x_n);
[k, x4a, mag_x4, ~] = fft_anylsis(x4, N, 0);
[k, x5a, mag_x5, ~] = fft_anylsis(x5, N, 0);
[xep, xop] = circevod(X_k);
xep = fftshift(xep);
xop = fftshift(xop);
subplot(2, 2, 1)
plot(n, abs(xep))
title('|Xep (k)|')
subplot(2, 2, 2)
plot(n, abs(xop))
title('|Xop (k)|')
subplot(2, 2, 3)
plot(k, mag_x4)
title('|DFS(Re[x(n)])|(|X4(k)|)')
subplot(2, 2, 4)
plot(k, mag_x5)
title('|DFS(jIm[x(n)])|(j*|X5(k)|)')

```