第3章 离散傅里叶变换

第 2 章对序列的长度未加任何限制，既可是无限长序列又可是有限长序列。对于只能在 $0 \leqslant n \leqslant N-1$ 的有限范围内考虑的序列 $x(n)$，特引入离散傅里叶变换（Discrete Fourier Transform，DFT），它本身也是有限长度序列，而不是连续函数。离散傅里叶变换不仅在理论上有重要意义，而且有快速计算的方法——快速傅里叶变换（Fast Fourier Transform，FFT）。因而离散傅里叶变换在各种实现数字信号处理的方法中起着重要作用。

本章主要讨论傅里叶变换的 4 种形式；离散傅里叶级数（DFS）推导及主要性质；离散傅里叶变换的定义及性质；频率采样；用离散傅里叶变换对连续时间信号逼近的问题；加权技术与窗函数。

从理论上说，离散傅里叶变换是傅里叶变换的一种可能形式。为了更好地理解这点，并不致发生混淆，本书先研究傅里叶变换的各种可能形式。

视频讲解

3.1 4 种傅里叶变换

所谓傅里叶变换就是以时间为自变量的“信号”与以频率为自变量的“频谱”函数之间的某种变换关系。这种变换同样可以应用于其他有关物理或数学的各种问题中，并可以采用其他形式的变量。当自变量“时间”或“频率”取连续形式和离散形式的不同组合，就可以形成各种不同的傅里叶变换对，有些变换对是前文介绍过的，为前后一致，采用和前文相同的符号。讨论中所绘出的虚拟函数图只是为了清楚地说明各种特性，并不代表任何实际的变换对。

3.1.1 非周期连续时间信号的傅里叶变换

非周期连续时间信号 $x_a(t)$ 的傅里叶变换 $X_a(j\Omega)$ 可以表示为

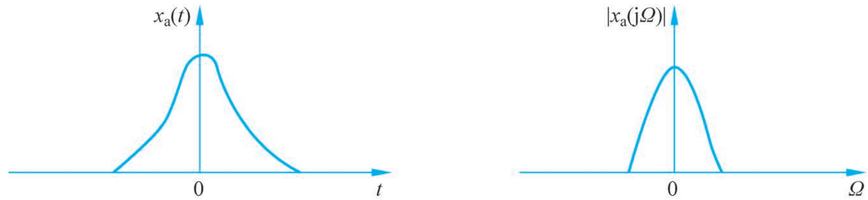
$$X_a(j\Omega) = \int_{-\infty}^{\infty} x_a(t) e^{-j\Omega t} dt \tag{3.1.1}$$

逆变换为

$$x_a(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X_a(j\Omega) e^{j\Omega t} d\Omega \tag{3.1.2}$$

式(3.1.1)及式(3.1.2)是大家所熟悉的非周期性连续时间信号及其频谱间的变换对，

其非周期连续时间函数及其傅里叶变换的形式如图 3.1.1 所示。可以看到：时域的连续函数造成频域的非周期谱，时域的非周期性造成频域的连续谱。结论是：一个非周期连续时间函数对应一个非周期连续频率变换函数。



(a) 非周期连续时间函数$x_a(t)$　　(b) $x_a(t)$的傅里叶变换——非周期连续频率函数$X_a(j\Omega)$

图 3.1.1　非周期连续时间函数及其傅里叶变换

3.1.2　周期连续时间信号的傅里叶变换

周期为 t_p 的周期性连续时间信号 $x_a(t)$的傅里叶变换是离散频率函数

$$X(m\Omega)=\frac{1}{t_p}\int_{-\frac{t_p}{2}}^{\frac{t_p}{2}} x_a(t)e^{-jm\Omega t}dt \tag{3.1.3}$$

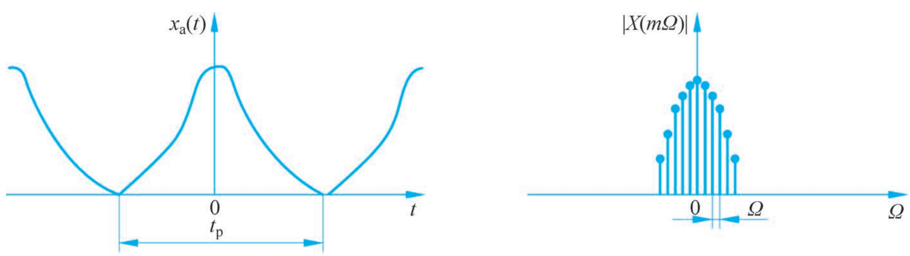
逆变换为

$$x_a(t)=\sum_{m=-\infty}^{\infty} X(m\Omega)e^{jm\Omega t} \tag{3.1.4}$$

式(3.1.3)及式(3.1.4)是被称为傅里叶级数的变换形式，式(3.1.3)所表示的积分是在 $x_a(t)$的一个周期内进行的。两相邻谱线分量之间的角频率增量与周期 t_p 之间的关系可表示为

$$\Omega=2\pi\frac{1}{t_p}=2\pi F\quad F=\frac{1}{t_p} \tag{3.1.5}$$

式(3.1.3) 及式(3.1.4)两个函数的特性如图 3.1.2 所示。可以看到：时域的连续函数造成频域的非周期谱；频域函数的离散(采样)造成了时域函数的周期。结论是：一个周期连续时间函数对应一个非周期离散频率变换函数。



(a) 周期连续时间函数$x_a(t)$　　(b) $x_a(t)$的傅里叶变换——非周期离散频率函数$X(m\Omega)$

图 3.1.2　周期连续时间函数及其傅里叶变换

3.1.3　非周期离散时间信号的傅里叶变换

非周期离散时间信号 $x(n)$的傅里叶变换 $X(e^{j\omega})$是第 2 章讨论的序列及其频域表示的

情况。其变换对为

$$X(\mathrm{e}^{\mathrm{j}\omega}) = \sum_{n=-\infty}^{\infty} x(n)\mathrm{e}^{-\mathrm{j}\omega n} \tag{3.1.6}$$

$$x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\mathrm{e}^{\mathrm{j}\omega})\mathrm{e}^{\mathrm{j}\omega n}\,\mathrm{d}\omega \tag{3.1.7}$$

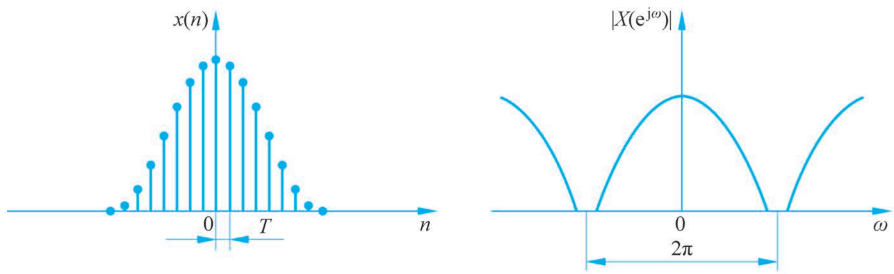
式(3.1.6)及式(3.1.7)用数字域频率 ω 来表示变换对,并且式(3.1.7)是在 $X(\mathrm{e}^{\mathrm{j}\omega})$的一个周期内求积分的。

采样频率 f_s 与采样周期 T 的关系是

$$f_s = \frac{1}{T}$$

采样的角频率为 $\Omega_s = \frac{2\pi}{T}$,而采样数字域频率 $\omega_s = 2\pi$。

式(3.1.6)及式(3.1.7)两个函数的特性如图 3.1.3 所示。该变换对说明,时域的采样,对应频域函数的周期延拓(其周期在数字域频率恰为 2π)。而时域函数的非周期对应频域函数的连续。



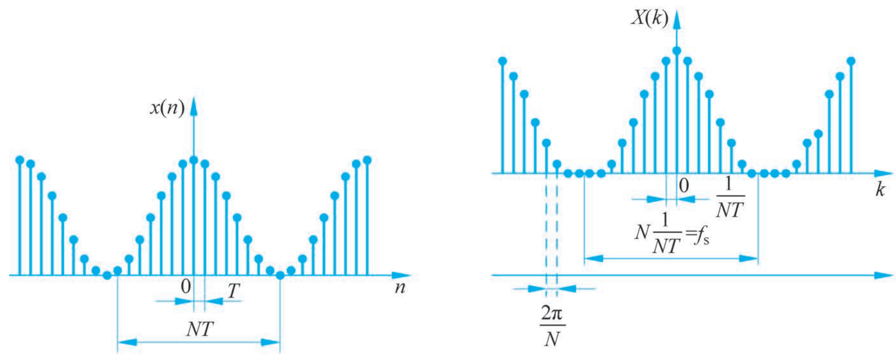
(a) 非周期离散时间信号$x(n)$　　(b) $x(n)$的傅里叶变换——周期连续频率函数$X(\mathrm{e}^{\mathrm{j}\omega})$

图 3.1.3　非周期离散时间信号及其傅里叶变换

3.1.4　周期离散时间信号的傅里叶变换

按照时间变量和频率变量是连续还是离散的不同组合,可以推断,存在时间变量和频率变量都是离散的情况。时域采样会得到频域的周期性函数。由于在非周期连续傅里叶变换中时间 t 及频率 f 是对称的,所以在一对傅里叶变换式中将 t 与 f 对调之后,计算关系同样成立。因此在频域采样,将使时域信号得到周期延拓(关于频率采样理论下文还要介绍)。这样,第 4 种傅里叶变换对实际上是周期的离散时间信号与周期的离散频率信号间的变换对,如图 3.1.4 所示。这就是本书将要分析的离散傅里叶级数变换。

总结以上 4 种傅里叶变换对的形式,可得如下结论:若一个函数在一个域内(时间域或频率域)是周期性的,则相应的在另一个域中的变换式必是采样的形式,即离散变量的函数;反之,如果在一个域中的函数是采样的,则在另一个域中必是周期性函数。在一个域中函数的周期必是另一个域中两个采样点间增量的倒数。



(a) 周期离散时间信号$\tilde{x}(n)$　　(b) $x(\tilde{n})$的傅里叶变换——周期离散频率函数$\tilde{X}(k)$

图 3.1.4 周期离散时间函数及其傅里叶变换

视频讲解

3.2 离散傅里叶级数

为了更好地理解离散傅里叶变换的概念，作为一种过渡，先简要地研究一下离散傅里叶级数。

3.2.1 离散傅里叶级数变换的推导

当用数字计算机对信号进行频谱分析时，要求信号必须以离散值作为输入，而计算机输出所得的频谱值，自然也是离散的。因此在 3.1 节中介绍的傅里叶变换的可能形式中，只有第 4 种形式对于数字信号处理有实用价值。前 3 种形式中或者信号是时间的连续函数，或者频谱是频率的连续函数，或者信号及频谱二者都是变量的连续函数，因此都不适合数字计算机进行计算。要使前 3 种形式能用数字计算机进行计算，必须针对每种形式的具体情况，或者在时域与频域上同时采样，或者在时域上采样，或者在频域上采样。信号在时域上采样导致频率的周期函数，而在频域上采样导致时域的周期函数，最后都将使原时间函数和频率函数二者都成为周期离散的函数，即由于采样的结果，前 3 种形式最后都能变为第 4 种形式——离散傅里叶级数形式。现在以第 3 种傅里叶级数形式（见图 3.1.3）为例来推导离散傅里叶级数变换。

为更清楚地表示式（3.1.6）所示的第 3 种形式傅里叶变换的周期性，下文在 $X(e^{j\omega})$上加表示周期性的上标"～"，并重写如下

$$\tilde{X}(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \tag{3.2.1}$$

设 $\tilde{x}(n)$的列长为 N，则式（3.2.1）可写为

$$\tilde{X}(e^{j\omega}) = \sum_{n=0}^{N-1} x(n)e^{-j\omega n}$$

再对 $\tilde{X}(e^{j\omega})$采样，使其成为周期性离散频率函数，并导致时域序列 $x(n)$周期化为 $\tilde{x}(n)$，如图 3.1.4 所示。图中时域采样间隔为 T，在一个周期内采样点数为 N。

由 3.2 节分析可知，在自变量为 t 及 f 的情况下，在一个域中对函数进行采样，两采样

点间增量的倒数，必是另一个域中函数的周期。当序列的周期为 NT 时，对频谱采样的谱间距是 $\frac{1}{NT}$。以数字频率表示时，谱间距是

$$\omega_{\mathrm{I}}=\frac{2\pi}{N} \tag{3.2.2}$$

因此，以数字频率 ω 为变量的 $X(\mathrm{e}^{\mathrm{j}\omega})$ 被离散化时，其变量 ω 成为

$$\omega=k\omega_{\mathrm{I}}=\frac{2\pi}{N}k \quad k=0,1,\cdots,N-1 \tag{3.2.3}$$

所以离散周期序列 $\tilde{x}(n)$ 的傅里叶级数可写成

$$\widetilde{X}(k)=\widetilde{X}(\mathrm{e}^{\mathrm{j}\omega})\Big|_{\omega=\frac{2\pi}{N}k}=\sum_{n=0}^{N-1}\tilde{x}(n)\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}kn}, \quad k=0,1,2,\cdots,N-1 \tag{3.2.4}$$

并将数字域频率简化为以 k 表示。上面两公式中，k 为整数，而且由于 $\widetilde{X}(\mathrm{e}^{\mathrm{j}\omega})$ 的周期是 2π，所以 k 只可取 $0\sim N-1$。这就是说，$\widetilde{X}(k)$ 有 N 个不同的值，$\widetilde{X}(k)$ 与 $\tilde{x}(n)$ 都是以 N 个采样值为一个周期的周期性函数。这一点在公式中看起来并不明显，事实上式(3.2.4)只能计算出 N 个独立的复数值。设 $k=r$，其中 r 是任意整数，则由式(3.2.4)有

$$\widetilde{X}(r)=\sum_{n=0}^{N-1}\tilde{x}(n)\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}rn}$$

又设，$k=r+N$，注意到

$$\mathrm{e}^{\frac{-\mathrm{j}2\pi n(r+N)}{N}}=\mathrm{e}^{\frac{-\mathrm{j}2\pi nr}{N}}\mathrm{e}^{-\mathrm{j}2\pi n}=\mathrm{e}^{-\mathrm{j}\frac{2\pi nr}{N}}$$

故有

$$\widetilde{X}(r+N)=\sum_{n=0}^{N-1}\tilde{x}(n)\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}nr}=\widetilde{X}(r)$$

因此式(3.2.4)的 $\widetilde{X}(k)$ 是以 N 个采样值为一个周期的周期性函数。

求离散傅里叶级数的逆变换，即从 $\widetilde{X}(k)$ 求 $\tilde{x}(n)$。将正变换式(3.2.4)两边同乘以 $\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}kr}$，并对一个周期求和，即

$$\sum_{k=0}^{N-1}\widetilde{X}(k)\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}kr}=\sum_{k=0}^{N-1}\left(\sum_{k=0}^{N-1}\tilde{x}(n)\mathrm{e}^{-\mathrm{j}2\pi kn/N}\right)\cdot\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}kr}=N\left[\sum_{n=0}^{N-1}\tilde{x}(n)\left(\frac{1}{N}\sum_{k=0}^{N-1}\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}kr(r-n)}\right)\right]$$

根据正交定理

$$\frac{1}{N}\sum_{k=0}^{N-1}\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}k(r-n)}=\begin{cases}1, & r=n\\ 0, & r\neq n\end{cases} \tag{3.2.5}$$

则得

$$\sum_{k=0}^{N-1}\widetilde{X}(k)\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}kr}=N\left[\sum_{n=0}^{N-1}\tilde{x}(n)\right]\Big|_{n=r}=N\tilde{x}(r)$$

以 n 替换 r 得

$$\tilde{x}(n)=\frac{1}{N}\sum_{k=0}^{N-1}\widetilde{X}(k)\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}kn} \tag{3.2.6}$$

与式(3.2.4)一样，式(3.2.6)所表达的也是一个以 N 为周期的周期序列

$$\tilde{x}(n+mN)=\frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}(k)\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}(n+mN)k}=\frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}(k)\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}kn}=\tilde{x}(n)$$

综上所述,离散周期序列的傅里叶级数变换对可表达为

$$\tilde{X}(k)=\mathrm{DFS}[\tilde{x}(n)]=\sum_{n=0}^{N-1}\tilde{x}(n)\mathrm{e}^{-\mathrm{j}\left(\frac{2\pi}{N}\right)kn} \tag{3.2.7}$$

$$\tilde{x}(n)=\mathrm{IDFS}[\tilde{X}(k)]=\frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}(k)\mathrm{e}^{\mathrm{j}\left(\frac{2\pi}{N}\right)nk} \tag{3.2.8}$$

其中,DFS[•]表示离散傅里叶级数正变换,IDFS[•]表示离散傅里叶级数逆变换。有时为了方便,令

$$W_N=\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}} \tag{3.2.9}$$

并称之为 W_N 因子,则式(3.2.7)及式(3.2.8)可表达为

$$\tilde{X}(k)=\mathrm{DFS}[\tilde{x}(n)]=\sum_{n=0}^{N-1}\tilde{x}(n)W_N^{kn} \tag{3.2.10}$$

$$\tilde{x}(n)=\mathrm{IDFS}[\tilde{X}(k)]=\frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}(k)W_N^{-kn} \tag{3.2.11}$$

3.2.2 傅里叶级数的主要性质

离散傅里叶级数的某些性质对其成功地应用于信号处理问题是极其重要的。下面简要地介绍一下这些重要性质。

1. 线性特性

若有周期皆为 N 的两个离散周期序列 $\tilde{x}_1(n)$和 $\tilde{x}_2(n)$线性组合成一个新的周期序列 $\tilde{x}_3(n)$

$$\tilde{x}_3(n)=a\tilde{x}_1(n)+b\tilde{x}_2(n) \tag{3.2.12}$$

则

$$\tilde{X}_3(k)=\mathrm{DFS}[a\tilde{x}_1(n)+b\tilde{x}_2(n)]=a\tilde{X}_1(k)+b\tilde{X}_2(k) \tag{3.2.13}$$

式中,a、b 为任意常数。线性特性可根据 DFS 的定义证明。

由于是线性组合,所以 $\tilde{x}_3(n)$的周期长度不变,仍为 N。$\tilde{X}_3(k)$也是周期为 N 的离散周期序列。

2. 序列移位

1) 时域移位

周期序列 $\tilde{x}(n)$左移 m 位后,得 $\tilde{x}(n+m)$,则

$$\mathrm{DFS}[\tilde{x}(n+m)]=W_N^{-mk}\tilde{X}(k) \tag{3.2.14}$$

证明:

$$\begin{aligned}\mathrm{DFS}[\tilde{x}(n+m)]&=\sum_{n=0}^{N-1}\tilde{x}(n+m)W_N^{nk} \qquad \text{换元 } i=n+m\\&=\sum_{i=m}^{N-1+m}\tilde{x}(i)W_N^{ki}W_N^{-mk}\\&=W_N^{-mk}\sum_{i=m}^{N-1+m}\tilde{x}(i)W_N^{ki}\end{aligned}$$

由于 $\tilde{x}(i)$及 W_N^{ki} 都是以 N 为周期的周期函数,因此对 i 求和时,下限从 m 至上限 $N-1+m$ 与下限从 0 至上限 $N-1$ 是相同的。因此

$$\sum_{i=m}^{N-1+m}\tilde{x}(i)W_N^{ki}=\sum_{i=0}^{N-1}\tilde{x}(i)W_N^{ki}=\widetilde{X}(k)$$

所以

$$\mathrm{DFS}[\tilde{x}(n+m)]=W_N^{-mk}\widetilde{X}(k)$$

显然,大于周期的任何移位(也即 $m\geqslant N$)和短于周期的移位在时域上不能区分。

2) 频域移位

当将 $\widetilde{X}(k)$左移 l 时,得 $\widetilde{X}(k+l)$,则

$$\mathrm{IDFS}[\widetilde{X}(k+l)]=W_N^{nl}\tilde{x}(n) \tag{3.2.15}$$

可用与上文类似的方法证明式(3.2.15)。

3. 周期卷积特性

1) 时域卷积

若 $\tilde{x}_1(n)$和 $\tilde{x}_2(n)$是周期为 N 的两个周期序列,并分别具有离散傅里叶级数 $\widetilde{X}_1(k)$和 $\widetilde{X}_2(k)$,则傅里叶级数 $\widetilde{X}_3(k)=\widetilde{X}_1(k)\cdot\widetilde{X}_2(k)$所对应的序列 $\tilde{x}_3(n)$为

$$\tilde{x}_3(n)=\sum_{m=0}^{N-1}\tilde{x}_1(m)\tilde{x}_2(n-m)=\sum_{m=0}^{N-1}\tilde{x}_2(m)\tilde{x}_1(n-m) \tag{3.2.16}$$

证明:

$$\begin{aligned}\tilde{x}_3(n)&=\mathrm{IDFS}[\widetilde{X}_3(k)]=\mathrm{IDFS}[\widetilde{X}_1(k)\cdot\widetilde{X}_2(k)]\\&=\frac{1}{N}\sum_{k=0}^{N-1}\widetilde{X}_1(k)\widetilde{X}_2(k)W_N^{-nk}\end{aligned}$$

将

$$\widetilde{X}_1(k)=\sum_{m=0}^{N-1}\tilde{x}_1(m)W_N^{mk}$$

代入,则

$$\begin{aligned}\tilde{x}_3(n)&=\frac{1}{N}\sum_{k=0}^{N-1}\sum_{m=0}^{N-1}\tilde{x}_1(m)\widetilde{X}_2(k)W_N^{-(n-m)k}\\&=\sum_{m=0}^{N-1}\tilde{x}_1(m)\left[\frac{1}{N}\sum_{k=0}^{N-1}\widetilde{X}_2(k)W_N^{-(n-m)k}\right]\\&=\sum_{m=0}^{N-1}\tilde{x}_1(m)\tilde{x}_2(n-m)\end{aligned}$$

只要将 $\tilde{x}_1(m)$、$\tilde{x}_2(n-m)$简单换元,就可证明

$$\tilde{x}_3(n)=\sum_{m=0}^{N-1}\tilde{x}_2(m)\tilde{x}_1(n-m) \tag{3.2.17}$$

式(3.2.17)是一个卷积公式,但不同于上文讨论过的线性卷积,二者的差别在于这里的卷积过程只限于一个周期以内,表现在式(3.2.17)中 m 取值取 $0\sim N-1$,所以称为周期卷积。

图 3.2.1 给出了两个周期为 $N=7$ 的序列 $\tilde{x}_1(n)$和 $\tilde{x}_2(n)$进行周期卷积的过程。

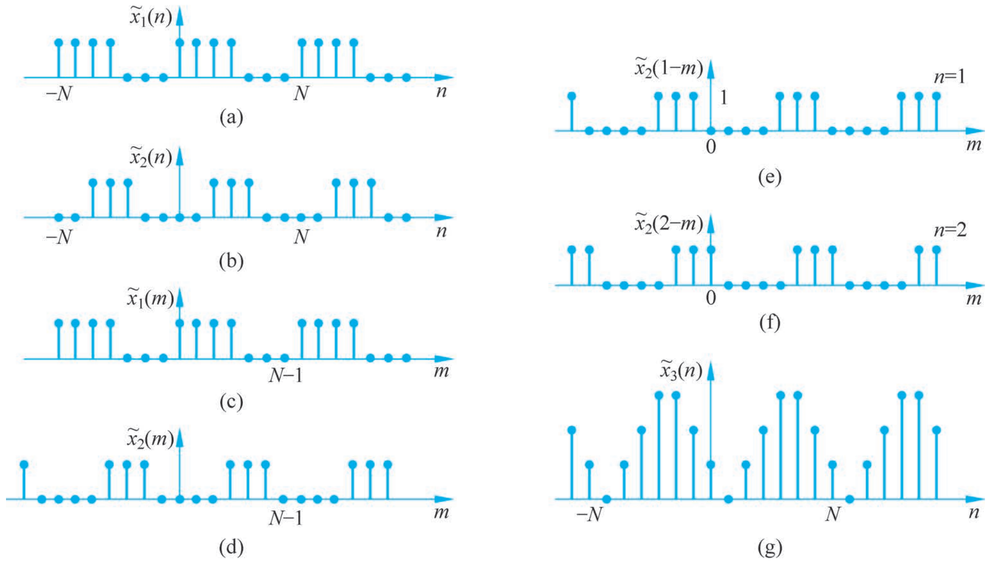


图 3.2.1 周期卷积

(1) 先在哑元坐标 m 上作 $\tilde{x}_1(m)$,$\tilde{x}_2(m)$(图中未画出);

(2) 将 $\tilde{x}_2(m)$以 $m=0$ 的垂直轴为轴作反卷得 $\tilde{x}_2(-m)$,并进行移位,例如图 3.2.1(d)、(e)、(f)所示的对应于 $n=0,1,2$ 时的 $\tilde{x}_2(n-m)$;

(3) 在一个周期内将 $\tilde{x}_2(n-m)$与 $\tilde{x}_1(m)$对应点逐点相乘后求和,就得到相应于 $n=0,1,2$ 时的 $\tilde{x}_3(n)$,继续下去就得到了整个周期内的全部 $\tilde{x}_3(n)$。

由于 $\tilde{x}_1(n)$,$\tilde{x}_2(n)$都是周期为 N 的周期序列,因此 $\tilde{x}_3(n)$也是周期为 N 的周期序列。

2) 频域卷积

对于周期序列的乘积,存在着频域的周期卷积。若

$$\tilde{x}_3(n)=\tilde{x}_1(n)\tilde{x}_2(n)$$

则

$$\begin{aligned}\tilde{X}_3(k)&=\mathrm{DFS}[\tilde{x}_3(n)]=\frac{1}{N}\sum_{l=0}^{N-1}\tilde{X}_1(l)\tilde{X}_2(k-l)\\&=\frac{1}{N}\sum_{l=0}^{N-1}\tilde{X}_2(l)\tilde{X}_1(k-l)\end{aligned} \tag{3.2.18}$$

根据 DFS 与 IDFS 变换的对称性不难证明式(3.2.18)。

【例 3.1】 计算 6 点的数字序列的 DFS 和卷积。

```
import numpy as np

N = 6
W6_1 = np.exp(-1j * 2 * np.pi / N * 1)
print("W6_1:", W6_1)
W6_2 = np.exp(-1j * 2 * np.pi / N * 2)
print("W6_2:", W6_2)

x = np.array([14, 12, 10, 8, 6, 10])
```

```
print("x:", x)
X = np.fft.fft(x)
print("X:", X)

# Define DFS function
def DFS(x, N, k):
    X = np.zeros(N, dtype = complex)
    for n in range(N):
        X[k] += x[n] * np.exp(-1j * 2 * np.pi * k * n / N)
    return X

X2 = DFS(x, 6, 0)
X3 = DFS(x, 6, 1)
print("X2:", X2)
print("X3:", X3)

x1 = np.array([1, 1, 1, 1, 0, 0, 0])
x2 = np.array([0, 0, 1, 1, 1, 0, 0])
x3 = np.convolve(x1, x2)
print("x3:", x3)
```

程序输出结果为

```
W6_1: (0.5000000000000001 - 0.8660254037844386j)
W6_2: (-0.4999999999999998 - 0.8660254037844387j)
x: [14 12 10 8 6 10]
X: [60. +0.00000000e+00j 9. -5.19615242e+00j 3. +1.73205081e+00j 0. +4.44089210e-16j
3. -1.73205081e+00j 9. +5.19615242e+00j]
X2: [60. +0.j 0. +0.j 0. +0.j 0. +0.j 0. +0.j 0. +0.j]
X3: [0. +0.j 9. -5.19615242j 0. +0.j 0. +0.j 0. +0.j 0. +0.j]
x3: [0 0 1 2 3 3 2 1 0 0 0 0 0]
```

视频讲解

3.3 离散傅里叶变换

3.3.1 DFT 只有 N 个独立的复值

离散傅里叶级数是周期序列，仍不便于计算机计算。离散傅里叶级数虽是周期序列却只有 N 个独立的复值，只要知道它一个周期的内容，其他的内容也就知道了。式(3.2.10)表明只要把一个周期内的 $\tilde{x}(n)$乘以对应的 W_N^{nk}，这里的 n 在$[0,N-1]$内取值，而 k 可在$(-\infty,\infty)$内取值，可得任意 k 对应的 $\widetilde{X}(k)$。式(3.2.11)表明，仅用 $\widetilde{X}(k)$的一个周期的值，即 k 只取$[0,N-1]$区间内的值，就可得任意 n，即 n 在$(-\infty,+\infty)$区间内取值时的 $\tilde{x}(n)$。如果同时限制式(3.2.10)中的 k 和式(3.2.11)中的 n 都只在区间$[0,N-1]$内取值，就得到了一个周期的 $x(n)$和一个周期的 $X(k)$间的对应关系。

$$X(k)=\mathrm{DFT}[x(n)]=\sum_{n=0}^{N-1}x(n)W_N^{kn},\quad 0\leqslant k\leqslant N-1 \tag{3.3.1}$$

$$x(n)=\mathrm{IDFT}[X(k)]=\frac{1}{N}\sum_{k=0}^{N-1}X(k)W_N^{-kn},\quad 0\leqslant n\leqslant N-1 \tag{3.3.2}$$

式(3.3.1)和式(3.3.2)即称为有限长序列的离散傅里叶变换对。式(3.3.1)称为离散傅里叶变换,简称 DFT。式(3.3.2)称为离散傅里叶逆变换,简称为 IDFT。

3.3.2　DFT 隐含周期性

若长度为 N 的有限长序列 $x(n)$是由对连续时间函数 $x(t)$采样得来的,则频域上已经意味着以 $\Omega_s=\frac{2\pi}{T}$(或 $\omega=2\pi$)为周期作周期延拓。再对频域作一次采样,则时间序列 $x(n)$按周期 N 延拓成为一个周期性时间序列 $\tilde{x}(n)$。因此,利用 DFT 对有限列长 N 的时间序列展开,相当于对此序列作周期性处理。离散傅里叶变换对式(3.3.1)及式(3.3.2),表面上看为非周期性的,但一定要想到它们隐含周期性。

3.3.3　DFT 是连续傅里叶变换的近似

离散傅里叶变换可以看成连续函数在时域、频域采样构成的变换。只要取出 $\tilde{x}(n)$的一个周期,乘以相应的内插函数就可恢复原连续函数。由下文分析可知,对频域也可取出 $\widetilde{X}(k)$的一个周期,乘以相应的内插函数,就可恢复原连续的频率函数。DFT 变换对可唯一地确定 $\widetilde{X}(k)$的一个周期 $X(k)$及 $\tilde{x}(n)$的一个周期 $x(n)$。$x(n)$及 $X(k)$都是长度为 N 的序列,都有 N 个独立复值,因而具有的信息是等量的,它们乘以相应的内插函数后,复原的连续函数也就完全确定了,因而离散傅里叶变换可以看作连续傅里叶变换的近似。$x(n)$及 $X(k)$都是有限长序列,便于用数字计算机计算,这样对连续函数的处理就可以用离散采样的处理代替。这就是要采用离散傅里叶变换的原因。

【例 3.2】　求 6 点的数字序列的 DFT 和 IDFT。

```
import matplotlib.pyplot as plt
import numpy as np
import math

# Number of sample points
N = 6
# sample spacing
T = 1.0 / 6.0
x = np.linspace(0.0, N * T, N, endpoint = False)
ys = x

# Build signal: Create a function to calculate the composite matrix M
def synthesis_matrix(N):
    ts = np.arange(N) / N
    fs = np.arange(N)
    args = np.outer(ts, fs)
    M = np.exp(1j * 2 * math.pi * args)
    return M

# Define DFT positive transformation
def dft(ys):
    N = len(ys)
    M = synthesis_matrix(N)
    amps = M.conj().transpose().dot(ys) # Calculate the weighted sum of frequency elements
```

```
    return amps

# Define inverse DFT transform
def idft(ys):
    N = len(ys)
    M = synthesis_matrix(N)
    amps = M.dot(ys) / N
    return amps

print(dft(ys))
print(idft(ys))
```

程序输出结果为

```
[ 2.5+0.00000000e+00j -0.5+8.66025404e-01j -0.5+2.88675135e-01j
 -0.5-3.06161700e-16j -0.5-2.88675135e-01j -0.5-8.66025404e-01j]
[ 0.41666667+0.00000000e+00j -0.08333333-1.44337567e-01j
 -0.08333333-4.81125224e-02j -0.08333333+5.10269500e-17j
 -0.08333333+4.81125224e-02j -0.08333333+1.44337567e-01j]
```

本程序中主要定义了 3 个函数：synthesis_matrix 是用于构建信号，给定一系列振幅和频率，合成一个输入矩阵，dft 和 idft 是根据 DFT 正逆变换的表达式对输入信号进行求值。

3.4 离散傅里叶变换的性质

在下文的讨论中，假定 $x_1(n)$与 $x_2(n)$都是列长为 N 的有限长序列，它们的离散傅里叶变换分别为

$$X_1(k)=\mathrm{DFT}[x_1(n)]$$
$$X_2(k)=\mathrm{DFT}[x_2(n)]$$

视频讲解

3.4.1 线性特性

若两个有限长序列 $x_1(n)$和 $x_2(n)$的线性组合为

$$x_3(n)=ax_1(n)+bx_2(n) \tag{3.4.1}$$

则 $x_3(n)$的离散傅里叶变换为 $X_1(k)$与 $X_2(k)$的线性组合

$$\begin{aligned}X_3(k)&=\mathrm{DFT}[ax_1(n)+bx_2(n)]\\&=a\mathrm{DFT}[x_1(n)]+b\mathrm{DFT}[x_2(n)]\\&=aX_1(k)+bX_2(k)\end{aligned} \tag{3.4.2}$$

式中 a、b 为任意常数。

注意，如果 $x_1(n)$列长为 N_1，$x_2(n)$列长为 N_2，则 $x_3(n)$的最大列长为 $N_3=\max[N_1,N_2]$。因而，离散傅里叶变换 $X_3(k)$必须按 $N=N_3$ 计算。例如，若 $N_1<N_2$，则 $X_1(k)$就是序列 $x_1(n)$增补 N_2-N_1 个零点后的离散傅里叶变换。

3.4.2 离散傅里叶逆变换的另一个公式

式(3.3.2)给出的离散傅里叶逆变换形式与正变换不同之处在于 W_N 因子用负指数，且有一个比例系数 $1/N$。离散傅里叶逆变换还有另一种形式，即

$$x(n)=\frac{1}{N}\left[\sum_{k=0}^{N-1}X^*(k)W_N^{nk}\right]^* \quad 0\leqslant n\leqslant N-1 \tag{3.4.3}$$

证明：

$$\frac{1}{N}\left[\sum_{k=0}^{N-1}X^*(k)W_N^{nk}\right]^*=\frac{1}{N}\left[\sum_{k=0}^{N-1}X(k)W_N^{-nk}\right]=x(n) \tag{3.4.4}$$

式(3.4.3)的逆变换公式中括号内 W_N 因子用正指数，与正变换公式中 W_N 指数一致。式(3.4.3)与正变换公式不同之处是用 $X(k)$的共轭 $X^*(k)$来作运算，并在作出括号内的变换运算后，再取共轭并乘以常数 $1/N$，得出时间序列 $x(n)$。这种逆变换形式在运算上与正变换一样，这样在电子计算机上，只要编一个程序就可以既用来计算离散傅里叶变换，又用来计算它的逆变换。

【例 3.3】 采用正变换求 6 点的数字序列的 DFT 和 IDFT。

```
import matplotlib.pyplot as plt
import numpy as np
import math

# Build signal: Create a function to calculate the composite matrix M
def synthesis_matrix(N):
    ts = np.arange(N) / N
    fs = np.arange(N)
    args = np.outer(ts, fs)
    M = np.exp(1j * 2 * math.pi * args)
    return M

# Define DFT positive transformation
def dft(x):
    N = len(x)
    M = synthesis_matrix(N)
    amps = M.conj().transpose().dot(x) # Calculate the weighted sum of frequency elements
    return amps

# Define inverse DFT transform
def idft(X):
    N = len(X)
    M = synthesis_matrix(N)
    amps = M.dot(X) / N
    return amps

N = 6
x = np.array([1, 2, 3, 4, 5, 6])
print("x:", x)

X = dft(x)
print("X:", X)

X1 = np.conj (X)
print("X1:", X1)

X2 = np.conj (dft(X1)) * (1 / N)
print("X2:", X2)
```

程序输出结果为

```
x: [1 2 3 4 5 6]
X: [21. + 0.00000000e + 00j  - 3. + 5.19615242e + 00j  - 3. + 1.73205081e + 00j
 - 3. - 2.20436424e - 15j  - 3. - 1.73205081e + 00j  - 3. - 5.19615242e + 00j]
X1: [21. - 0.00000000e + 00j  - 3. - 5.19615242e + 00j  - 3. - 1.73205081e + 00j
 - 3. + 2.20436424e - 15j  - 3. + 1.73205081e + 00j  - 3. + 5.19615242e + 00j]
X2: [1. - 4.29286236e - 15j 2. + 8.88178420e - 16j 3. - 1.11022302e - 15j
4. + 2.36847579e - 15j 5. + 3.91517295e - 15j 6. - 4.58892184e - 15j]
```

3.4.3 对称定理

若 $x(n)$的离散傅里叶变换为 $X(k)$，则当时间序列具有频谱序列的形状 $X(n)$时，其对应的离散傅里叶变换对如下

$$\frac{1}{N}X(n)\Leftrightarrow x(-k)=x(N-k) \tag{3.4.5}$$

式(3.4.5)说明 $X(n)$的对应频谱序列具有原来的时间序列 $x(n)$在时间上倒置的形状。

证明：

因为

$$x(-n)=x(N-n)=\frac{1}{N}\sum_{k=0}^{N-1}X(k)W_N^{-(N-n)k}=\frac{1}{N}\sum_{k=0}^{N-1}X(k)W_N^{nk}$$

所以

$$x(-k)=x(N-k)=\sum_{n=0}^{N-1}\left[\frac{X(n)}{N}\right]W_N^{kn}=\mathrm{DFT}\left[\frac{X(n)}{N}\right]$$

关于 $x(-n)=x(N-n)$，$X(-k)=X(N-k)$的说明见下文离散傅里叶变换的奇偶性及对称性。

3.4.4 反转定理

若 $x(n)$的离散傅里叶变换为 $X(k)$，则 $x(-n)$的离散傅里叶变换为 $X(-k)$。这可直接由离散傅里叶变换的定义得到证明。

3.4.5 序列的总和

列长为 N 的时间序列 $x(n)$中各采样值的总和等于其离散傅里叶变换 $X(k)$在 $k=0$ 时的值，即

$$X(k)\big|_{k=0}=\sum_{n=0}^{N-1}x(n)W_N^{nk}\big|_{k=0}=\sum_{n=0}^{N-1}x(n) \tag{3.4.6}$$

3.4.6 序列的始值

若序列的离散傅里叶变换为 $X(k)$，则对应的时间序列 $x(n)$的始值 $x(0)$为频谱序列各采样值 $X(k)$的总和除以 N，即

$$x(0)=\frac{1}{N}\sum_{k=0}^{N-1}X(k) \tag{3.4.7}$$

3.4.7 延长序列的离散傅里叶变换

把序列 $x(n)$,$0\leqslant n\leqslant N-1$,填充零值,人为地加长到 rN,得到

$$g(n),\quad 0\leqslant n\leqslant rN-1$$

式中 r 为正整数,而

$$g(n)=\begin{cases}x(n), & 0\leqslant n\leqslant N-1\\ 0, & N\leqslant n\leqslant rN-1\end{cases}\tag{3.4.8}$$

$g(n)$的离散傅里叶变换为

$$\begin{aligned}G(k)=\mathrm{DFT}[g(n)]&=\sum_{n=0}^{rN-1}g(n)\mathrm{e}^{-\mathrm{j}\frac{2\pi nk}{rN}}\\&=\sum_{n=0}^{N-1}x(n)\mathrm{e}^{-\mathrm{j}\frac{2\pi n\left(\frac{k}{r}\right)}{N}}\\&=X\left(\frac{k}{r}\right)\quad k=0,1,\cdots,rN-1\end{aligned}\tag{3.4.9}$$

这意味着,$g(n)$的频谱 $G(k)$与 $x(n)$的频谱 $X(k)$是相对应的,只不过 $G(k)$的频谱间隔比 $X(k)$的频谱间隔降低 k/r。也就是说,若把序列 $x(n)$填充零值而人为地加长以后再进行离散傅里叶变换,就可使得到的频谱更加细致。

若增加的长度并非 N 的整数倍,例如 $g(n)$的长度为 $L>N$,则由列长为 L 的序列 $g(n)$的离散傅里叶变换 $G(k)$,可得到序列 $x(n)$的 L 根谱线,它比由 $X(k)$得出的 N 根谱线要多。

【例 3.4】 计算对 16 点序列补零到 32 点和 64 点后的 DFT。

```
import numpy as np
import matplotlib.pyplot as plt

N1 = 16
n = np.arange(0, 16)
x = np.sin(2 * np.pi * n/N1) + np.cos(4 * np.pi * n/N1)

N2 = 32
n2 = np.arange(0, 32)
x2 = np.concatenate((x, np.zeros(16)))

N3 = 64
n3 = np.arange(0, 64)
x3 = np.concatenate((x, np.zeros(48)))

L1 = np.arange(0, 16)
dft_16 = np.fft.fft(x, 16)
L2 = np.arange(0, 32)
dft_32 = np.fft.fft(x2, 32)
L3 = np.arange(0, 64)
dft_64 = np.fft.fft(x3, 64)

nx = np.arange(0, 16)
K = 512
```

```
dw = 2 * np.pi/K
k = np.arange(0, 512)
X = np.dot(x, np.exp(1j * dw * nx.reshape(-1, 1) * k))

plt.subplot(4, 1, 1)
plt.plot(k * dw/(2 * np.pi), np.abs(X))
plt.ylim([0, 12])

plt.subplot(4, 1, 2)
plt.stem(L1/N1, np.abs(dft_16))
plt.ylim([0, 12])

plt.subplot(4, 1, 3)
plt.stem(L2/32, np.abs(dft_32))
plt.ylim([0, 12])

plt.subplot(4, 1, 4)
plt.stem(L3/64, np.abs(dft_64))
plt.ylim([0, 12])

plt.show()
```

延长序列 DFT 如图 3.4.1 所示。

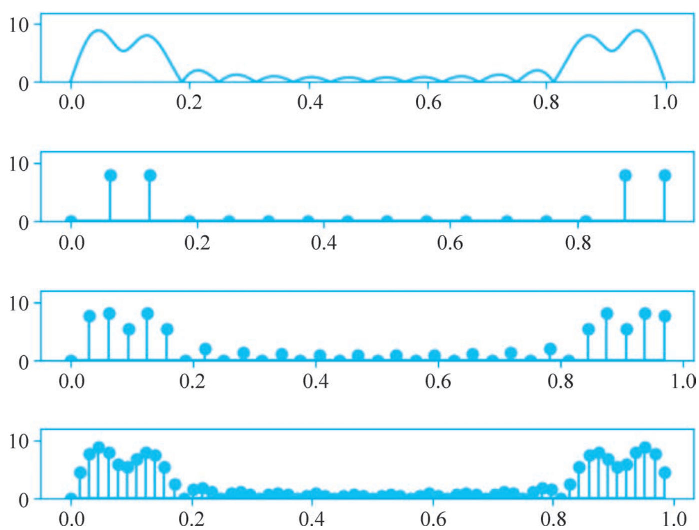


图 3.4.1 延长序列 DFT

视频讲解

3.4.8 序列的圆周移位

1. 圆周移位

一列长为 N 的有限长序列 $x(n)$，在区间$[0,N-1]$内取非零值，如图 3.4.2(a)所示。如令其沿坐标 n 右移 m 后仍在$[0,N-1]$区间内取值，如图 3.4.2(b)所示，则将发生信息损失。为避免这种情况，可将 $x(n)$以 N 为周期作周期延拓得

$$\tilde{x}(n)=x((n))_N$$

再把 $\tilde{x}(n)$偏移 m 得

$$\tilde{x}(n+m)=x((n+m))_N \tag{3.4.10}$$

上述过程如图 3.4.2(c),(d)所示。然后再对 $\tilde{x}(n+m)$仍在$[0,N-1]$区间上取值,即取移位周期序列 $\tilde{x}(n+m)$的主值序列 $x((n+m))_N R_N(n)$,如图 3.4.2(e)所示。

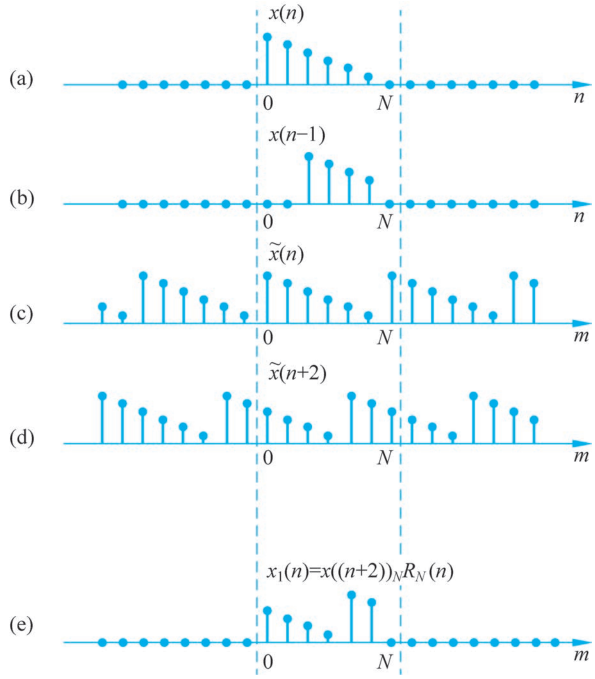
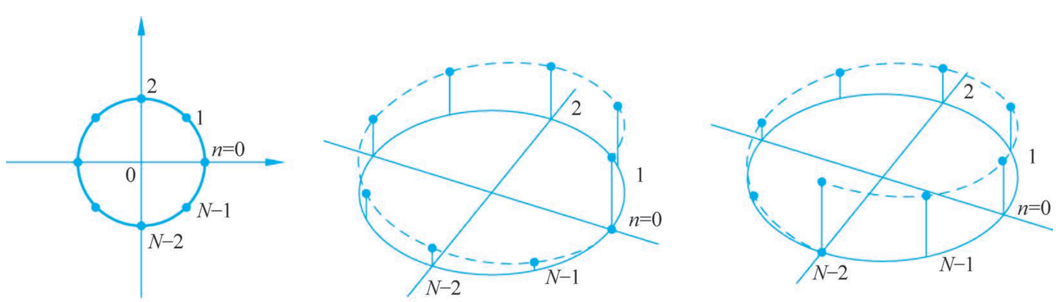


图 3.4.2　圆周移位过程图

这样,有限长序列 $x(n)$的圆周移位定义为

$$x_1(n)=x((n+m))_N R_N(n) \tag{3.4.11}$$

$x_1(n)$仍是一长度为 N 的有限长序列,它实际上 $x(n)$是向左移位,但 m 移到 n 负轴上的序列又从右边 $N-1$ 处循环回来。这可想象为将序列 $x(n)$排列在一个 N 等分的圆周上,且 $n=0$ 的点取为正水平方向。由于列长为 N,故相邻二点的间隔为 $2\pi/N$ 角度,然后将圆顺时针旋转 m 个 $2\pi/N$ 的角度,再从正水平方向开始依序取各点的序列值,即得 $x(n)$的圆周移位序列 $x_1(n)$,因此得名为"圆周移位",也称"循环移位"。其过程如图 3.4.3 所示。



(a) N等分的圆周　　(b) 将序列排列在N等分的圆周上　　(c) 令圆周旋转得$x(n)$的圆周移位

图 3.4.3　序列的圆周移位

2. 有限长序列圆周移位定理

1）时间移位定理

设 $\tilde{X}(k)$和 $\tilde{X}_1(k)$分别表示周期序列 $\tilde{x}(n)$和 $\tilde{x}_1(n)=\tilde{x}(n+m)$的离散傅里叶级数。据式(3.2.14)有

$$\tilde{X}_1(k)=W_N^{-km}\tilde{X}(k)$$

由式(3.3.8)可得序列 $x(n)$圆周移位所得的序列 $x_1(n)$的 DFT 为

$$X_1(k)=W_N^{-km}(k)\tilde{X}(k)R_N(k)=W_N^{-km}X(k) \tag{3.4.12}$$

式(3.4.12)表示时间移位定理。它说明：序列在时间上的时移将引起各根谱线产生相应的相移。

2）频率移位定理(也称调制定理)

对于频域，有限长序列 $X(k)$也可以认为是分布在一个 N 等分的圆周上。$X(k)$的圆周移位为 $X((k+l))_N R_N(k)$，利用 $x(n)$与 $X(k)$的对称特性可证明。

$$\mathrm{IDFT}[X((k+l))_N R_N(k)]=W_N^{nl}x(n) \tag{3.4.13}$$

式(3.4.13)即称为频率移位定理，有时也称为调制定理。此定理说明：若频谱序列在频率上作移位 l，则时间序列调制了 $l\Omega$ 的频率。这可以由下式看出

$$W_N^{nl}x(n)=\mathrm{e}^{-\mathrm{j}\left(\frac{2\pi}{N}\right)nl}x(n)=x(n)\mathrm{e}^{-\mathrm{j}\left(\frac{2\pi}{NT}\right)nlT}=x(n)\mathrm{e}^{-\mathrm{j}l\Omega_n T} \tag{3.4.14}$$

【例 3.5】 已知序列 $u(n)=[0\ 1\ 2\ 1\ 0]$，求：

(1) 循环右移 2 位；

(2) 循环左移 2 位；

(3) 序列反转；

(4) 序列反转循环右移 1 位。

```
import numpy as np
import matplotlib.pyplot as plt

u = np.array([0, 1, 2, 1, 0])

# 循环右移 2 位
u1 = np.roll(u, 2)
plt.subplot(5, 1, 1)
plt.stem(u)
plt.xlabel('序号')
plt.ylabel('u(n)')

plt.subplot(5, 1, 2)
plt.stem(u1)
plt.xlabel('序号')
plt.ylabel('u1')

# 循环左移 2 位
u2 = np.roll(u, -2)
plt.subplot(5, 1, 3)
plt.stem(u2)
```

```
plt.xlabel('序号')
plt.ylabel('u2')

# 序列反转
u3 = np.flip(u)
plt.subplot(5, 1, 4)
plt.stem(u3)
plt.xlabel('序号')
plt.ylabel('u3')

# 序列反转循环右移 1 位
u4 = np.roll(np.flip(u), 1)
plt.subplot(5, 1, 5)
plt.stem(u4)
plt.xlabel('序号')
plt.ylabel('u4')

# 调整子图之间的间距
plt.subplots_adjust(hspace = 1)

# 手动替换负号为正确显示负号
plt.rcParams['axes.unicode_minus'] = False
plt.xticks(fontproperties = 'SimHei')
plt.yticks(fontproperties = 'SimHei')

plt.tight_layout()
plt.show()
```

序列圆周移位如图 3.4.4 所示。

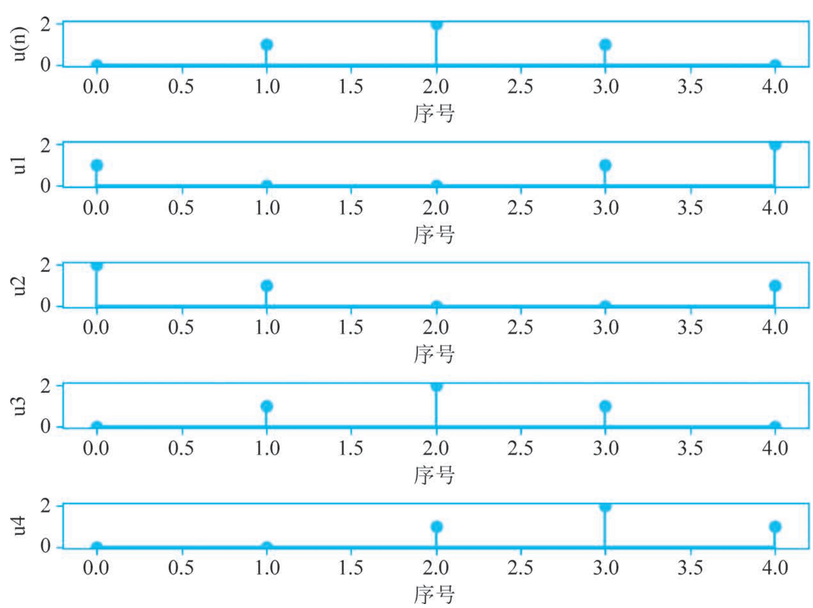


图 3.4.4　序列圆周移位

视频讲解

3.4.9 圆周卷积及其与有限长序列的线性卷积关系

圆周卷积也称循环卷积。圆周卷积定理：两个序列离散傅里叶变换的乘积等于此两个序列的圆周卷积的离散傅里叶变换，即若

$$X_3(k)=X_1(k)X_2(k)$$

则

$$\begin{aligned}x_3(n)=\mathrm{IDFT}[X_3(k)]&=\sum_{m=0}^{N-1}x_1(m)x_2((n-m))_N R_N(n)\\&=\sum_{m=0}^{N-1}x_2(m)x_1((n-m))_N R_N(n)\end{aligned}\tag{3.4.15}$$

证明：

由于 $X_1(k)$，$X_2(k)$都隐含周期性，所以其乘积 $X_3(k)$同样隐含周期性。为此要证明式(3.4.15)可以应用周期卷积的结果。

式(3.4.15)的卷积可以看作是周期序列 $\tilde{x}_1(n)$与 $\tilde{x}_2(n)$卷积后再取其主值序列，即将 $X_3(k)$周期延拓

$$\tilde{X}_3(k)=\tilde{X}_1(k)\tilde{X}_2(k)$$

则

$$\begin{aligned}\tilde{x}_3(n)=\mathrm{IDFS}[\tilde{X}_3(k)]&=\sum_{m=0}^{N-1}\tilde{x}_1(m)\tilde{x}_2(n-m)\\&=\sum_{m=0}^{N-1}x_1((m))_N x_2((n-m))_N\end{aligned}$$

因 $0\leqslant m\leqslant N-1$，$x_1((m))_N=x_1(m)$，因此

$$x_3(n)=\tilde{x}_3(n)R_N(n)=\left[\sum_{m=0}^{N-1}x_1(m)x_2((n-m))_N\right]R_N(n)\tag{3.4.16}$$

式(3.4.16)经过换元可得

$$x_3(n)=\sum_{m=0}^{N-1}x_2(m)x_1((n-m))_N R_N(n)$$

式(3.4.16)的卷积过程如图 3.4.5 所示，与图 3.2.1 的周期卷积相比，两者的卷积过程是一样的，只是这里只取卷积结果的主值序列。由于卷积过程只在主值区间内进行 $0\leqslant m\leqslant N-1$，因此 $x_2((n-m))_N R_N(n)$实际上就是 $x_2(m)$的圆周移位，所以上述卷积称为圆周卷积，习惯上常用⊛表示圆周卷积，以区别于线性卷积。式(3.4.16)的圆周卷积可表示为

$$x_3(n)=x_1(n)\circledast x_2(n)\tag{3.4.17}$$

必须指出，两列长为 N 的有限长序列的圆周卷积的结果也是一个列长为 N 的有限长序列，而不是像线性卷积那样列长为 $2N-1$。

由于 $x_3(n)$与 $X_3(k)$的对称特性，若

$$x_3(n)=x_1(n)x_2(n)$$

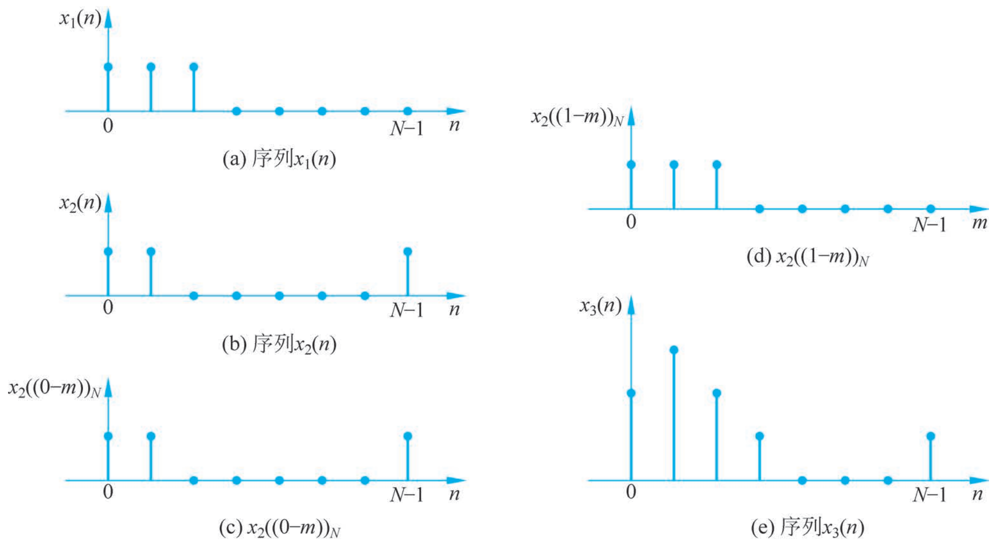


图 3.4.5 圆周卷积

同样可以证明

$$X_3(k)=\mathrm{DFT}[x_3(n)]=\frac{1}{N}\sum_{l=0}^{N-1}X_1(l)X_2((k-l))_N R_N(k)$$

$$=\frac{1}{N}\sum_{l=0}^{N-1}X_2(l)X_1((k-l))_N R_N(k) \tag{3.4.18}$$

式(3.4.18)称为频率卷积定理。

1. 圆周卷积和线性卷积的关系

实际问题求解线性卷积：例如，信号 $x(n)$通过系统 $h(n)$其输出为线性卷积 $y(n)=x(n)*h(n)$。下文将会讲到，圆周卷积由于可以采用快速傅里叶变换技术，所以比线性卷积运算速度快。如果 $x(n)$及 $h(n)$都是有限长序列，能否用圆周卷积来取代线性卷积而不失真呢？为此必须理解圆周卷积与线性卷积的关系，这是正确运用圆周卷积的关键。

假定 $x_1(n)$是列长为 N 的有限长序列，$x_2(n)$是列长为 M 的有限长序列，二者的线性卷积为

$$x(n)=x_1(n)*x_2(n)=\sum_{m=-\infty}^{\infty}x_1(m)x_2(n-m)$$

$x(n)$也是有限长序列，其列长可以这样来决定。

从 $x_1(m)$看，序号 m 的区间为

$$0\leqslant m\leqslant N-1$$

从 $x_2(n-m)$看，序号 $n-m$ 的区间为

$$0\leqslant n-m\leqslant M-1$$

将上两个不等式相加，得

$$0\leqslant n\leqslant N+M-2$$

在上述区间外不是 $x_1(m)=0$ 就是 $x_2(n-m)=0$，因而 $x(n)=0$。所以 $x(n)$是一个 n 取 $0\sim N+M-2$ 的序列，列长为 $N+M-1$(两序列长度之和减 1)。例如，图 3.4.6 中 $x_1(n)$

为 $N=3$ 的矩形序列，$x_2(n)$为 $M=5$ 的矩形序列，两者的线性卷积 $x(n)$是具有 $N+M-1=7$ 的采样值的序列。

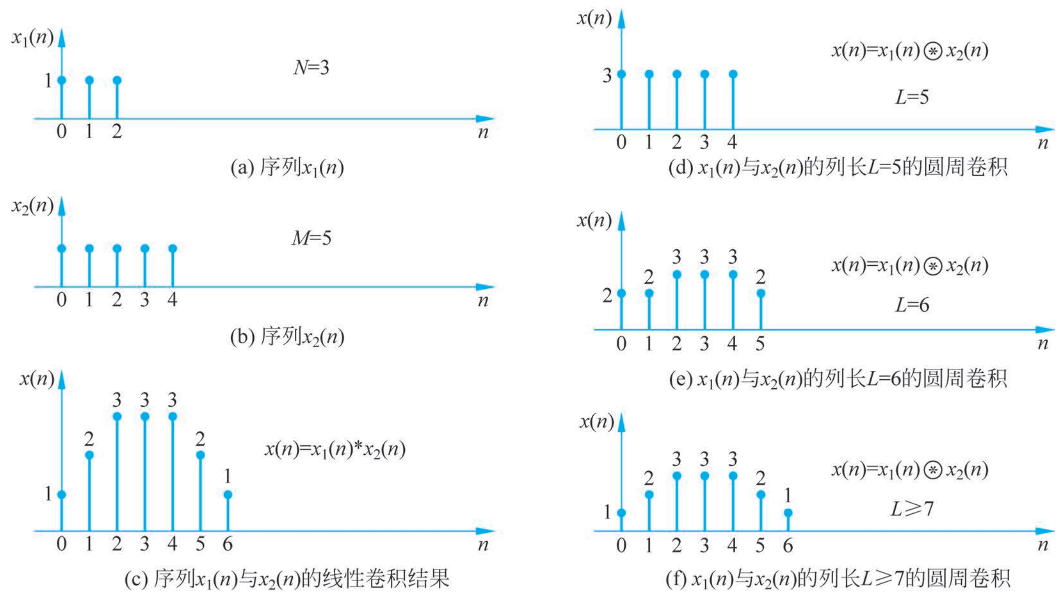


图 3.4.6 圆周卷积与线性卷积

对 $x_1(n)$和 $x_2(n)$进行列长为 L 的圆周卷积 $X_L(n)=x_1(n)⊛x_2(n)$的情况。把列长为 N 的序列 $x_1(n)$后面补上 $L-N$ 个零值点，把列长为 M 的序列 $x_2(n)$后面补上 $L-M$ 个零值点，均构成列长为 L 的有限长序列。为了进行圆周卷积，先将 $x_1(n)$及 $x_2(n)$进行周期延拓

$$\tilde{x}_1(n)=\sum_{q=-\infty}^{\infty} x_1(n+qL)$$

$$\tilde{x}_2(n)=\sum_{k=-\infty}^{\infty} x_2(n+qL)$$

则

$$\tilde{x}_2(-m)=\sum_{k=-\infty}^{\infty} x_2(-m+kL)$$

$$\tilde{x}_2(n-m)=\sum_{k=-\infty}^{\infty} x_2(n-m+kL)$$

$\tilde{x}_1(n)$及 $\tilde{x}_2(n)$的列长为 L 的周期卷积为

$$\begin{aligned}\tilde{x}_L(n)&=\sum_{m=0}^{L-1}\sum_{q=-\infty}^{\infty} x_1(m+qL)\sum_{k=-\infty}^{\infty} x_2(n-m+kL)\\&=\sum_{m=0}^{L-1}\sum_{k=-\infty}^{\infty} x_1(m)x_2(n-m+kL)\end{aligned}$$

交换求和次序，有

$$\tilde{x}_L(n)=\sum_{k=-\infty}^{\infty}\sum_{m=0}^{L-1} x_1(m)x_2(n+kL-m)=\sum_{k=-\infty}^{\infty} x(n+kL) \tag{3.4.19}$$

式(3.4.19)表明：$x_1(n)$和$x_2(n)$的周期卷积是$x_1(n)$和$x_2(n)$的线性卷积$x(n)$以L为周期的延拓。

由前文分析可知，线性卷积$x(n)$具有$N+M-1$个非零序列值。因此，如果周期卷积的周期$L<N+M-1$，则$x(n)$的周期延拓就必然有一部分非零序列值要交叠起来，发生混淆。只有当$L\geqslant N+M-1$时，才不会发生交叠，这时$x(n)$的周期延拓$\tilde{x}_1(n)$中的每一个周期L内，前$N+M-1$个序列值是序列$x(n)$的值，而剩下的$L-(N+M-1)$个点上序列值则是补充的零值。

圆周卷积就是周期卷积取主值序列，即

$$x_L(n)=x_1(n)\circledast x_2(n)=\tilde{x}_L(n)R_L(n)$$

因此

$$x_L(n)=\left[\sum_{k=-\infty}^{\infty}x(n+kL)\right]R_L(n)$$

所以要使圆周卷积与线性卷积相等，而不产生混淆的必要条件是

$$L\geqslant N+M-1 \tag{3.4.20}$$

符合式(3.4.20)的条件，则$x_L(n)=x(n)$，即

$$x_1(n)\circledast x_2(n)=x_1(n)*x_2(n)$$

图3.4.7(d)～图3.4.7(f)反映了当L取值不同时，线性卷积与圆周卷积的关系。图3.4.7(d)中$L=5$，这时混淆现象严重，使得$x_L(n)$与$x(n)$完全不一样，而图3.4.7(e)中

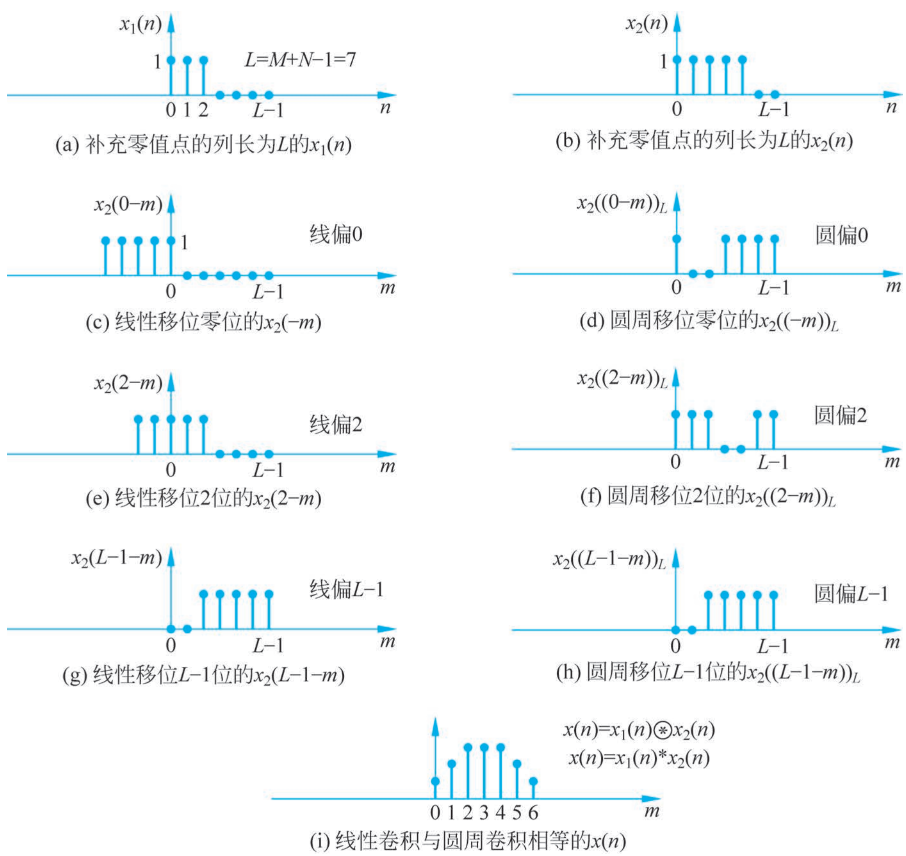


图3.4.7　圆周位移及线性位移比较

$L=6$,这时有两点($n=0,n=6$)发生混淆失真,在图3.4.7(f)中,满足条件$L\geqslant N+M-1=7$时,圆周卷积与线性卷积的结果相同。只要具体比较一下在这种条件下,圆周卷积与线性卷积的实际过程就会透彻地理解两者为什么相等。

如图3.4.7所示,L满足$L\geqslant N+M-1=7$的条件,列长为$N=3$的序列$x(n)$已补上$L-N=4$个零值点(图3.4.7(a));列长为$M=5$的序列$x_2(n)$已补充上$L-M=2$个零值点(图3.4.7(b)),则在L长的区间上对两个序列作圆周卷积时,由于$x_2(-m)$有$L-M$个零值点,所以它在$[0,L-1]$区间上作圆周移位时,在$[0,L-1]$区间上移位的结果和$x_2(-m)$的线性移位结果是一样的,如图3.4.7(c),(d),(e),(f),(g),(h)所示。另一方面,虽然在$[N,L-1]$区间上圆周移位结果和线性移位结果可能会不同,但由于$x_1(n)$在此区间均为零值点,所以也不会有输出。因而在区间$[0,L-1]$上的圆周卷积结果和$x_1(n),x_2(n)$的线性卷积结果实际是一样的。

综上所述可以清楚看出:有限长序列$x_1(n)$和$x_2(n)$的周期卷积是$x_1(n)$和$x_2(n)$的线性卷积以周期延拓。而圆周卷积是周期卷积的去主值序列,当满足条件$L\geqslant N+M-1$时,圆周卷积与线性卷积的结果相同。

【例3.6】 (1)求序列x1=[1,2,3]和序列x2=[2,3,1,2]的线性卷积。

```
import numpy as np
import matplotlib.pyplot as plt

def conv_m(x, h):
    nx = len(x)
    nh = len(h)
    ny = nx + nh - 1
    y = np.zeros(ny)                          # 初始化
    for n in np.arange(ny):
        y[n] = 0
        for m in np.arange(nh):
            k = n - m + 1
            if k >= 1 and k <= nx:
                y[n] = y[n] + h[m] * x[k - 1]    # 卷积

    plt.rcParams['font.sans-serif'] = ['SimHei']
    plt.rcParams['axes.unicode_minus'] = False      # 显示中文标签
    plt.stem(y)
    plt.title('线性卷积')
    plt.xlabel('n')
    plt.ylabel('y')
    plt.xlim(0, 8)
    plt.show()

conv_m(np.array([1, 2, 3]), np.array([2, 3, 1, 2]))
```

运行程序,输出结果如图 3.4.8 所示。

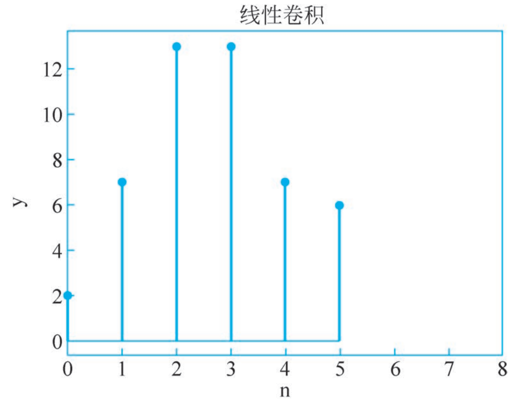


图 3.4.8 有限长序列的线性卷积

(2) 求序列 x1=[1, 2, 3]和序列 x2=[2, 3, 1, 2]在 N=5, 6, 7 点时的圆周卷积。

```
import numpy as np
import matplotlib.pyplot as plt

def conv_m(x, h):                                        # 线性卷积
    nx = len(x)
    nh = len(h)
    ny = nx + nh - 1
    y = np.zeros(ny)                                     # 初始化
    for n in np.arange(ny):
        y[n] = 0
        for m in np.arange(nh):
            k = n - m + 1
            if k >= 1 and k <= nx:
                y[n] = y[n] + h[m] * x[k - 1]            # 卷积
    return y

def cir_con(x1, x2, N):                                  # 圆周卷积
    nx1 = np.arange(0, len(x1))
    nx2 = np.arange(0, len(x2))
    x_1 = np.append(x1, np.zeros(N - len(x1)))
    h_1 = np.append(x2, np.zeros(N - len(x2)))
    y1 = conv_m(x_1, h_1)                                # 调用线性卷积函数
    z_1 = np.append(np.zeros(N), y1[0:N])
    z_2 = np.append(y1[N:2 * N], np.zeros(N))
    z = z_1[0:2 * N - 1] + z_2[0:2 * N - 1] + y1[0:2 * N - 1]
    y = z[0:N]
    ny = np.arange(0, N)

    plt.rcParams['font.sans-serif'] = ['SimHei']
    plt.rcParams['axes.unicode_minus'] = False           # 显示中文标签
    plt.stem(y)
    plt.title('圆周卷积')
    plt.xlabel('n')
    plt.ylabel('y')
    plt.xlim(0, 8)
```

```
    plt.show()

cir_con(np.array([1, 2, 3]), np.array([2, 3, 1, 2]), 5)
cir_con(np.array([1, 2, 3]), np.array([2, 3, 1, 2]), 6)
cir_con(np.array([1, 2, 3]), np.array([2, 3, 1, 2]), 7)
```

运行程序，结果如图 3.4.9 所示。

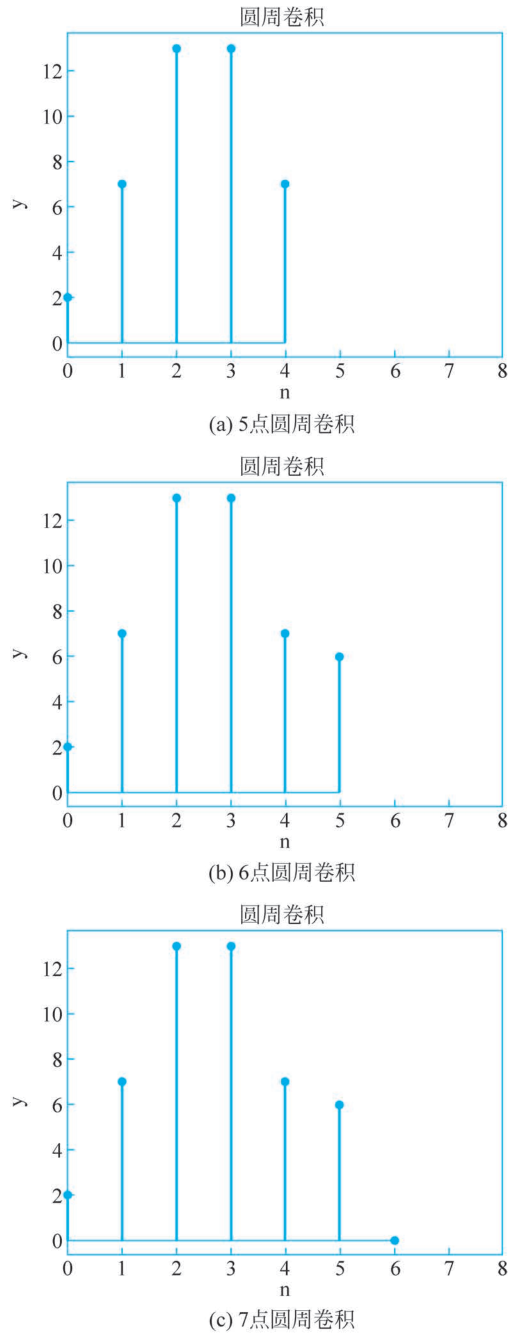


图 3.4.9　有限长序列的圆周卷积

2. 圆周卷积在信号处理中的应用——重叠相加法与重叠保留法

可用两种办法计算线性系统的输出序列。

(1) 按线性卷积来计算；

(2) 通过圆周卷积，用快速卷积法计算。

设输入 $x(n)$的列长为 N，系统的单位采样响应的列长为 M，则二者线性卷积后的输出序列 $y(n)$的列长为 $L=N+M-1$。由圆周卷积定理可知，两序列离散傅里叶变换的乘积等于它们圆周卷积的离散傅里叶变换。分别取 $x(n)$和 $h(n)$的 $L\geqslant N+M-1$ 点的离散傅里叶变换 $X(k)$和 $H(k)$，将两者相乘再求其逆变换即得两序列的圆周卷积，此时圆周卷积与线性卷积相等。这种方法称为快速卷积法，因为可以利用快速傅里叶变换迅速而有效地求出卷积，甚至当 $N+M-1$ 为 30 的时候，快速卷积法也比直接卷积法更有效。快速卷积法是一种重要的信号处理工具。

当输入序列 $x(n)$极长时，为了尽快得到输出，不允许等 $x(n)$全部集齐后再进行卷积，否则会使输出相对于输入有较长的延时。再者，如果 $N+M-1$ 太大，则需要大量存储单元。为此，可把 $x(n)$分段，分别求出每段的卷积，合在一起即得总的输出。这种分组进行卷积法可利用上文介绍的离散傅里叶变换实现，并可细分为重叠相加法、重叠保留法两种。下面针对图 3.4.10 所描绘的信号 $x(n)$和列长为 M 的单位采样响应 $h(n)$来解释这两种方法。

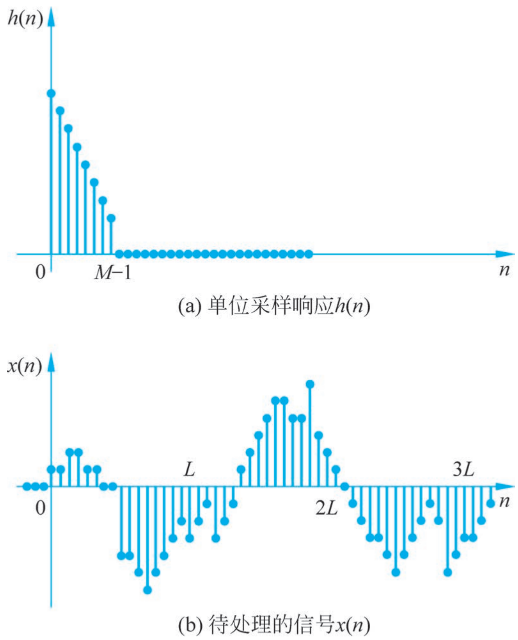


图 3.4.10　有限列长的单位采样响应及待处理的信号

1) 重叠相加法

$h(n)$的列长为 M，信号为 $x(n)$。将 $x(n)$分解为几段之和，每段长为 L 点，如图 3.4.10 所示。L 的选择是相当复杂的，一个好的经验是使 L 与 M 的数量级相同。

以 $x_k(n)$表示第 k 段 $x(n)$，于是输入信号 $x(n)$可表示成

$$x(n)=\sum_{k=0}^{\infty}x_k(n) \tag{3.4.21}$$

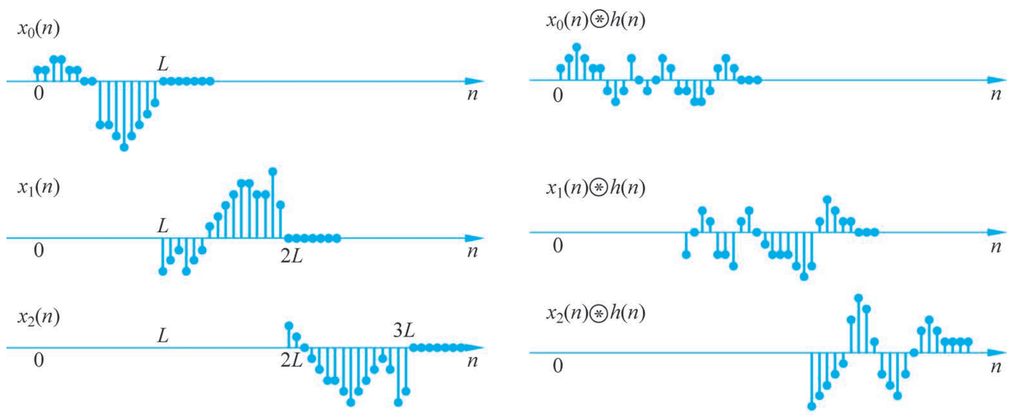
式中

$$x_k(n)=\begin{cases}x(n), & kL\leqslant n\leqslant(k+1)L-1\\0, & \text{其他}\end{cases}\tag{3.4.22}$$

$x(n)$与$h(n)$的线性卷积等于$x_k(n)$和$h(n)$卷积之和,即

$$x(n)*h(n)=\sum_{k=0}^{\infty}x_k(n)*h(n)\tag{3.4.23}$$

式(3.4.23)和式中的每一项$[x_k(n)*h(n)]$的列长为$L+M-1$,因此可将$h(n)$及$x_k(n)$两个序列补0,都加长到$L+M-1$点,以利用$L+M-1$的DFT,通过圆周卷积得到线性卷积$x_k(n)*h(n)$。为便于用基-2FFT运算,一般可取$L+M-1=2^v$。由于每一段的起点和前后相邻各段起点相隔L个点,而每一段信号$x_k(n)$和$h(n)$卷积后列长为$(L+M-1)$,所以式(3.4.23)在求和时,每段卷积的最后$M-1$个点必须和下一段的前$M-1$个点重叠,如图3.4.11所示。各输入段$x_k(n)$描绘于图3.4.11(a),相加各$x_k(n)$波形即可重新组成输入信号$x(n)$的波形。卷积后各段输出如图3.4.11(b)所示,而总的输出序列$y(n)=x(n)*h(n)$是通过相加图3.4.11(b)中各段构成的。重叠部分也要相加,故称为重叠相加法。重叠是由于每段输入序列$x_k(n)$与单位采样响应$h(n)$的线性卷积后的列长长于$x(n)$的分段长度造成的。



(a) 将$x(n)$分解为列长为L的不重复的几段　　(b) 每段$x_k(n)$和$h(n)$卷积的结果

图 3.4.11　重叠相加法图形

2) 重叠保留法

重叠保留法延长分段序列的办法不是补零,而是保留原来的输入序列值,且保留在每段的前端,如图3.4.12(b)中的虚线部分所示。这时如利用DFT实现$h(n)$和$x_k(n)$的圆周卷积,则其每段卷积结果的前$M-1$个点不等于线性卷积值,需舍去。

为了清楚地看出这点,研究$x(n)$中的任意一段列长为N的序列$x_i(n)$与列长为M的$h(n)$的圆周卷积情况

$$y'_i(n)=x_i(n)*h(n)=\sum_{m=0}^{N-1}x_i(m)h((n-m))_N R_N(n)\tag{3.4.24}$$

由于$h(n)$的列长为M,当在$0\leqslant n\leqslant M-2$范围内进行圆周位移时,$h((n-m))_N$将在$x_i(m)$的尾部出现有非零值,如图3.4.12(c)所示的$n=1$的情况。所以在这一部分$0\leqslant n\leqslant$

$M-2$ 的 $y_i'(n)$值中将混入 $x_i(m)$尾部与 $h((n-m))_N$ 的卷积值，从而使 $y_i'(n)$不同于线性卷积结果。但是当 n 从 $M-1$ 开始，直到 $N-1$ 点，则有 $h((n-m))_N=h(n-m)$，如图 3.4.12(d)，(e)所示。因此从 $n=M-1$ 点后，圆周卷积值完全与线性卷积值一样，$y_i'(n)$是正确的卷积值，因而每一段卷积运算结果的前 $M-1$ 个值需去掉，如图 3.4.12(f)所示。

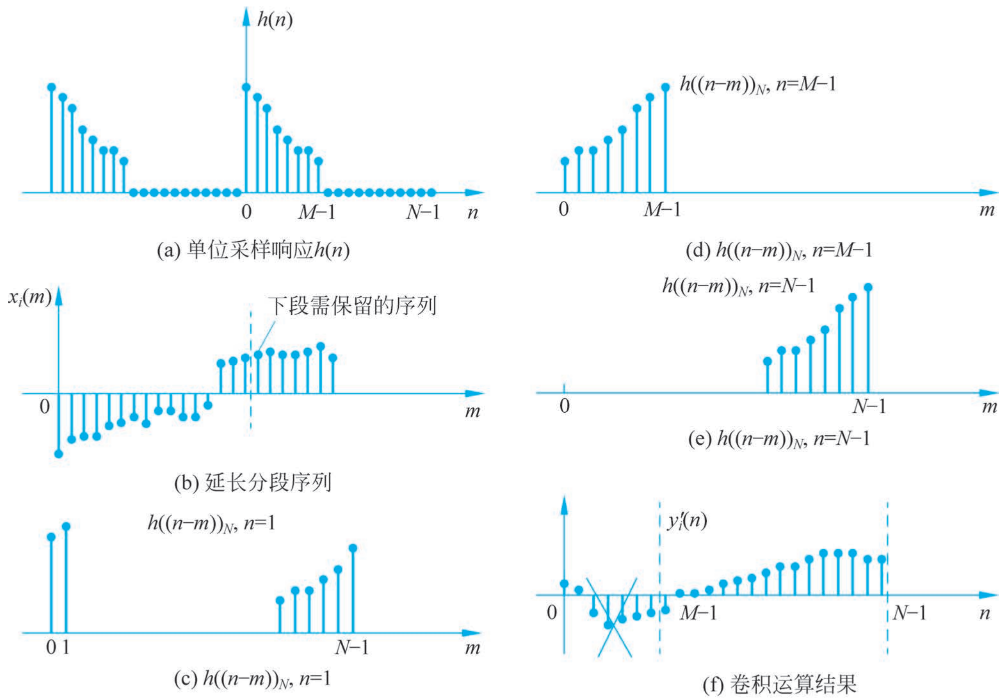


图 3.4.12 用保留信号代替补零后局部的混叠现象

为了不造成输出信号的遗漏，对 $x(n)$分段时，需使相邻两段有 $M-1$ 个点的重叠(对于第一段 $x(n)$由于没有前一段保留信号，则在其前填充 $M-1$ 个零值点)，为此将 $x_k(n)$定义为

$$x_k(n)=\begin{cases}x[n+k(N-(M-1))-M+1], & 0\leqslant n\leqslant N-1\\ 0, & \text{其他 } n\end{cases} \tag{3.4.25}$$

式中已规定将每段的时间原点放在该段的起始点，而不是 $x(n)$的原点。

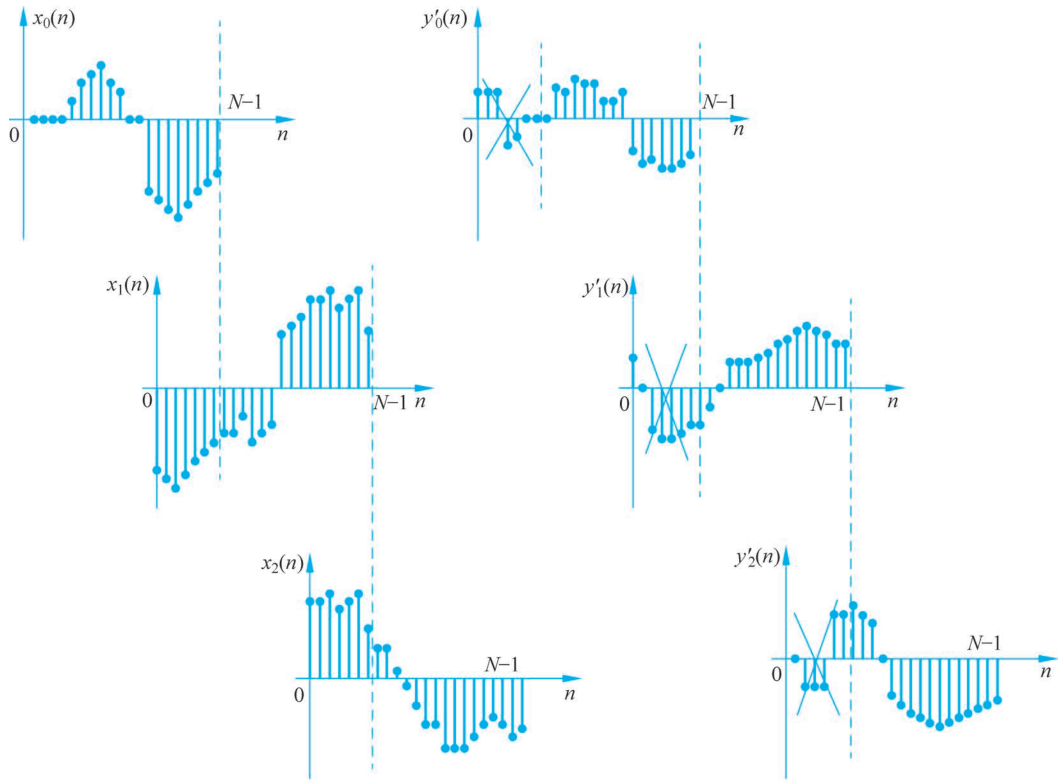
这种分段方法描绘于图 3.4.13(a)。每段和 $h(n)$的圆周卷积以 $y_k'(n)$表示，如图 3.4.13(b)所示。图中已标明每输出波形段开始的 $0\leqslant n\leqslant M-2$ 部分须舍去，把相邻各输出段留下的序列衔接起来，就构成了最终的没有遗漏而又正确的输出，即

$$y(n)=\sum_{k=0}^{\infty}y_k(n-k(N-M+1))$$

这里 n 为总的输出序列 $y(n)$的序号。而式中

$$y_k(n)=\begin{cases}y_k'(n), & M-1\leqslant n\leqslant N-1\\ 0, & \text{其他}\end{cases} \tag{3.4.26}$$

每段输出的时间原点放在 $y_k(n)$的起始点，而不是 $y(n)$的原点。



(a) $x(n)$分解为列长为N的不重复的几段　　(b) 每段与$h(n)$圆周卷积的结果，图中标出在形成线性卷积时每段要去掉的部分

图 3.4.13　重叠保留法示意图

重叠保留法的名称是因为每个相继的输入段均由 $N-M+1$ 个新点和前一段保留下来的 $M+1$ 个点所组成而得名的。

3.4.10　圆周相关定理

视频讲解

圆周相关也称循环相关。

若 $X_3(k)=X_1^*(k)X_2(k)$

则

$$x_3(n)=\mathrm{IDFT}[X_3(k)]=\sum_{i=0}^{N-1}x_1^*(l)x_2((l+n))_N R_N(n) \tag{3.4.27}$$

证明：

由于 $X_1(k)$和 $X_2(k)$都隐含周期性，所以其乘积 $X_3(k)$同样隐含周期性。将 $X_3(k)$周期延拓，则

$$\widetilde{X}_3(k)=\widetilde{X}_1^*(k)\widetilde{X}_2(k)$$

即

$$\tilde{x}_3(n)=\mathrm{IDFS}[\widetilde{X}_3(k)]=\frac{1}{N}\sum_{k=0}^{N-1}\widetilde{X}_1^*(k)\widetilde{X}_2(k)W_N^{-nk}$$

将

$$\tilde{X}_1^*(k)=\left[\sum_{l=0}^{N-1}\tilde{x}_1(l)W_N^{lk}\right]^*$$

代入，则

$$\begin{aligned}\tilde{x}_3(n)&=\frac{1}{N}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1}\tilde{x}_i^*(l)\tilde{X}_2(k)W_N^{-(n+l)k}\\&=\sum_{i=0}^{N-1}\tilde{x}_i^*(l)\left[\frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}_2(k)W_N^{-(n+l)k}\right]\\&=\sum_{i=0}^{N-1}\tilde{x}_1^*(l)\tilde{x}_2(n+l)\\&=\sum_{l=0}^{N-1}x_i^*((l))_N x_2((n+l))_N\end{aligned}$$

因 $0\leqslant l\leqslant N-1$，$x_1^*((l))_N=x_1^*(l)$故有

$$x_3(n)=\tilde{x}_3(n)R_N(n)=\left[\sum_{l=0}^{N-1}x_l^*(l)x_2((n+l))_N\right]R_N(n) \tag{3.4.28}$$

如果仅考虑 $x_1(l)$是实数，其共轭还是其本身，则得

$$x_3(n)=\left[\sum_{i=0}^{N-1}x_1(l)x_2((n+l))_N\right]R_N(n)$$

可以得到两个 N 点实序列 $x_1(n)$与 $x_2(n)$的 N 点圆周相关，而 $x_2((n+l))_N R_N(n)$正好是 $x_2(n)$的圆周移位，变量是 n，且无须折叠。若定义 $x_3(n)=\sum\limits_{m=-\infty}^{\infty}x_1(m)x_2(n+m)$为线性相关，则两列长分别为 N_1 及 N_2 的线性相关，应等于将这两序列补零到 $N=N_1+N_2-1$ 列长后的圆周相关。通过圆周相关定理可知，实信号的相关性，可借助于离散傅里叶正逆变换求得。当一个较短的序列与一个很长序列相关时，可类似圆周卷积的做法，采用分段相关的办法。

3.4.11　帕塞瓦尔定理

若 $X(k)=\mathrm{DFT}[x(n)]$，则

$$\sum_{n=0}^{N-1}x^2(n)=\frac{1}{N}\sum_{k=0}^{N-1}|X(k)|^2 \tag{3.4.29}$$

证明：

在圆周相关定理中，取 $x_2(l)=x_1(l)$，且 $n=0$，则

$$\sum_{i=0}^{N-1}x_1(l)x_2(l)=\frac{1}{N}\sum_{k=0}^{N-1}X^*(k)X(k)W_N^0$$

所以

$$\sum_{n=0}^{N-1}x^2(n)=\frac{1}{N}\sum_{k=0}^{N-1}|X(k)|^2$$

帕塞瓦尔定理建立了离散函数时域能量和频域能量之间的关系，又称为能量定理。

3.4.12 离散傅里叶变换的奇偶性及对称性

1. 周期性共轭对称和共轭反对称分量

第 2 章曾讨论过任一序列都可分解成共轭对称与共轭反对称两个分量之和。共轭对称分量为

$$x_e(n)=\frac{1}{2}[x(n)+x^*(-n)] \tag{3.4.30}$$

共轭反对称分量为

$$x_o(n)=\frac{1}{2}[x(n)-x^*(-n)] \tag{3.4.31}$$

在讨论有限长序列离散傅里叶变换对称性质时，一般不采用第 2 章给出的共轭对称和共轭反对称分量的定义。因为列长为 N 的序列 $X(k)$其离散傅里叶变换的列长也为 N，而其 $x_e(n)$及 $x_o(n)$的列长却都为$(2N-1)$(见图 3.4.14(g)～图 3.4.14(j))，难以由它们找出 DFT 的对称性。因为 DFT 的列长为 N。

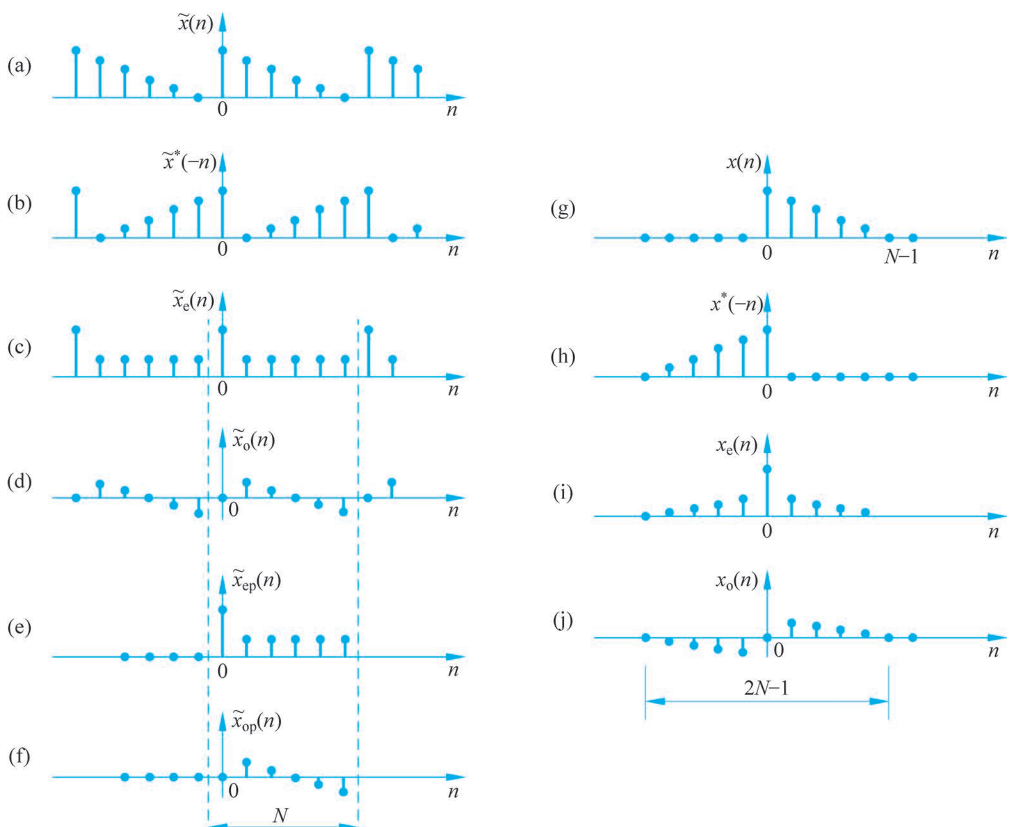


图 3.4.14 周期与非周期序列的共轭对称及共轭反对称分量以及周期性共轭对称及共轭反对称分量

对于周期为 N 的周期序列 $\tilde{x}(n)$，它的共轭对称分量和共轭反对称分量仍然都是周期性的，并且周期仍为 N。可将 $x(n)$分解成两个列长为 N 的有限长序列，其一对应于 $\tilde{x}(n)$的共轭对称分量的一个周期，记作 $x_{ep}(n)$，另一对应于 $\tilde{x}(n)$的共轭反对称分量的一个周

期，记作 $x_{op}(n)$。

为此，将 $x(n)$以 N 为周期延拓成周期序列

$$\tilde{x}(n)=x((n))_N$$

则周期序列 $\tilde{x}(n)$的共轭对称和共轭反对称分量分别为

$$\tilde{x}_e(n)=\frac{1}{2}[\tilde{x}(n)+\tilde{x}^*(-n)] \tag{3.4.32}$$

$$\tilde{x}_o(n)=\frac{1}{2}[\tilde{x}(n)-\tilde{x}^*(-n)] \tag{3.4.33}$$

取 $0\sim N-1$ 的一个周期值，则得

$$\begin{aligned}x_{ep}(n)&=\frac{1}{2}[x((n))_N+x^*((-n))_N]R_N(n)\\&=\frac{1}{2}[x(n)+x^*(N-n)]\end{aligned} \tag{3.4.34}$$

$$\begin{aligned}x_{op}(n)&=\frac{1}{2}[x((n))_N-x^*((-n))_N]R_N(n)\\&=\frac{1}{2}[x(n)-x^*(N-n)]\end{aligned} \tag{3.4.35}$$

又由式(3.4.32)及式(3.4.33)可得

$$\tilde{x}(n)=\tilde{x}_e(n)+\tilde{x}_o(n)$$

因此

$$\begin{aligned}x(n)&=\tilde{x}(n)R_N(n)\\&=[\tilde{x}_e(n)+\tilde{x}_o(n)]R_N(n)\\&=x_{ep}(n)+x_{op}(n)\end{aligned} \tag{3.4.36}$$

序列 $x_{ep}(n)$和 $x_{op}(n)$分别称为列长为 N 的序列 $x(n)$的周期性共轭对称分量和周期性共轭反对称分量。当 $x_{ep}(n)$和 $x_{op}(n)$为实序列时，分别称作周期性偶分量和周期性奇分量。这样的术语易使人误解，因为序列 $x_{ep}(n)$和 $x_{op}(n)$并不是周期序列，它们只是分别表示周期序列 $\tilde{x}_e(n)$和 $\tilde{x}_o(n)$的一个周期，其间关系如图 3.4.14(c)～图 3.4.14(f)所示。

显然，$x_{ep}(n)$和 $x_{op}(n)$与式(2.3.3)定义的 $x_e(n)$和式(2.3.4)定义的 $x_o(n)$并不等价。然而不难证明，$x_{ep}(n)$和 $x_{op}(n)$与列长为$(2N-1)$的 $x_e(n)$和 $x_o(n)$有下述关系

$$x_{ep}(n)=[x_e(n)+x_e(n-N)]R_N(n) \tag{3.4.37}$$

$$x_{op}(n)=[x_o(n)+x_o(n-N)]R_N(n) \tag{3.4.38}$$

以上讨论也完全适用于序列 $X(k)$，并得出类似的表达式。

最后还须指出：因为本书把有限长序列视为周期为 N 的时间序列中的一个周期，因此有

$$x(-n)=x(N-n) \tag{3.4.39}$$

同样地

$$X(-k)=X(N-k) \tag{3.4.40}$$

2. 奇偶序列的 DFT

1）奇序列的 DFT

当序列是奇对称时，即

$$x(n)=-x(-n)=-x(N-n) \tag{3.4.41}$$

则其离散傅里叶变换也是奇对称的，即

$$X(k)=-X(-k)=-X(N-k) \tag{3.4.42}$$

证明：

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{kn}=\begin{cases}\sum\limits_{n=0}^{N-1}[-x(-n)]W_N^{(-k)(-n)}=-X(-k)\\ \sum\limits_{n=0}^{N-1}[-x(N-n)]W_N^{(N-k)(N-n)}=-X(N-k)\end{cases}$$

式(3.4.42)证明中，使用了下述关系

$$W_N^{(N-k)(N-n)}=W_N^{N^2}\cdot W_N^{-kN}\cdot W_N^{-nN}\cdot W_N^{kn}=W_N^{kn}$$

2) 偶序列的 DFT

当序列是偶对称时，即

$$x(n)=x(-n)=x(N-n) \tag{3.4.43}$$

则其离散傅里叶变换也是偶对称的，即

$$X(k)=X(-k)=X(N-k) \tag{3.4.44}$$

可用与奇序列的 DFT 相似的方法证明。

3. 共轭复序列的 DFT

若 $x^*(n)$为 $x(n)$的共轭复序列，则

$$\mathrm{DFT}[x^*(n)]=X^*(N-k) \tag{3.4.45}$$

证明：

$$\mathrm{DFT}[x^*(n)]=\sum_{n=0}^{N-1}x^*(n)W_N^{nk}=\left[\sum_{n=0}^{N-1}x(n)W_N^{-nk}\right]^*,\quad 0\leqslant k\leqslant N-1$$

由于

$$W_N^{nN}=\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}nN}=\mathrm{e}^{-\mathrm{j}2\pi n}=1$$

故

$$\begin{aligned}\mathrm{DFT}[x^*(n)]&=\left[\sum_{n=0}^{N-1}x(n)W_N^{(N-k)n}\right]^*\\&=X^*((N-k))_N\\&=X^*(N-k)\end{aligned}$$

请注意，$k=0$ 时不能使用等式 $X^*((N-k))_N=X^*(N-k)$，因为这时 $X^*((N-0))_N=X^*(N-0)=X^*(N)$，而 $X(k)$只有 $0\leqslant k\leqslant N-1$ 范围内的 N 个值，所以已超出取值区间。式(3.4.45)的严格公式应该是

$$\mathrm{DFT}[x^*(n)]=X^*((N-k))_N \tag{3.4.46}$$

这样，当 $k=0$ 时有 $X^*((N-0))_N=X^*(0)$的正确结果。因为 $X(k)$可认为是分布在 N 等分的圆周上，它的末点即它的始点，也即 $X(N)=X(0)$。因此仍采用式(3.4.45)的习惯形式。在下文有关对称性的讨论中，凡是 $X(N)$都认为是 $X((N))_N=X(0)$。

4. 复序列的 DFT

若有限长序列 $x(n)$是一个复序列，设 $x_r(n)$及 $\mathrm{j}x_i(n)$分别表示 $x(n)$的实部与虚部，即

$$x(n)=x_r(n)+jx_i(n) \tag{3.4.47}$$

而

$$\begin{cases} x_r(n)=\frac{1}{2}[x(n)+x^*(n)] \\ jx_i(n)=\frac{1}{2}[x(n)-x^*(n)] \end{cases} \tag{3.4.48}$$

以 $X_{ep}(k)$及 $X_{op}(k)$分别表示实部及虚部序列的 DFT，则

$$\begin{aligned} X_{ep}(k)&=\mathrm{DFT}[x_r(n)] \\ &=\frac{1}{2}\mathrm{DFT}[x(n)+x^*(n)] \\ &=\frac{1}{2}[X(k)+X^*(N-k)] \end{aligned} \tag{3.4.49}$$

$$\begin{aligned} X_{op}(k)&=\mathrm{DFT}[jx_i(n)] \\ &=\frac{1}{2}\mathrm{DFT}[x(n)-x^*(n)] \\ &=\frac{1}{2}[X(k)-X^*(N-k)] \end{aligned} \tag{3.4.50}$$

根据线性特性

$$X(k)=X_{ep}(k)+X_{op}(k) \tag{3.4.51}$$

注意，这里 $X_{ep}(k)$与 $X_{op}(k)$均为复数，所以式(3.4.51)右端总的实部和总的虚部还是不能直接表示出来。

现在分析一下 $X_{ep}(k)$与 $X_{op}(k)$的一些对称特性。由式(3.4.49)得

$$\begin{aligned} X_{ep}^*(N-k)&=\frac{1}{2}[X(N-k)+X^*(N-N+k)]^* \\ &=\frac{1}{2}[X^*(N-k)+X(k)] \end{aligned} \tag{3.4.52}$$

与式(3.4.49)比较，可知

$$X_{ep}(k)=X_{ep}^*(N-k) \tag{3.4.53}$$

因此 $X_{ep}(k)$称为 $X(k)$的周期性共轭对称分量，其含义在上文已经说明。由此可认为 $X_{ep}(k)$是分布在 N 等分圆周上，则以 $k=0$ 为原点，$X_{ep}(k)$在左半圆上的序列与右半圆上的序列是共轭对称的。图 3.4.15 直观地表示了 $X_{ep}(k)$为实数时的示意图。若 $X_{ep}(k)$为复数，则其共轭对称的含义是模相等、幅角相反，即

$$\begin{cases} |X_{ep}(k)|=|X_{ep}(N-k)| \\ \arg[X_{ep}(k)]=-\arg[X_{ep}(N-k)] \end{cases} \tag{3.4.54}$$

或说实部相等、虚部相反。

根据式(3.4.50)可以推得

$$X_{op}(k)=-X_{op}^*(N-k) \tag{3.4.55}$$

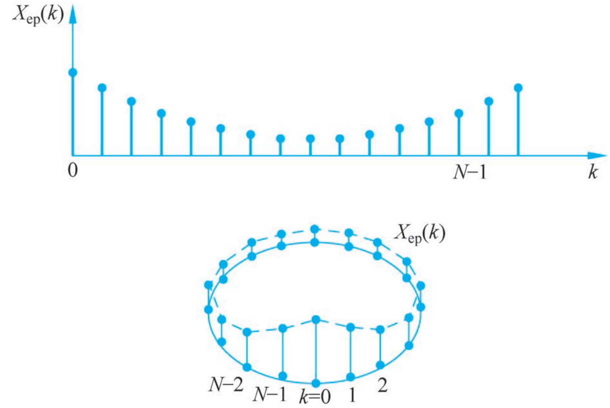


图 3.4.15　周期性共轭对称分量 $X_{ep}(k)$

$X_{op}(k)$称为 $X(k)$的周期性共轭反对称分量,它表示 $X_{op}(k)$分布在圆周上时,以 $k=0$ 为中心,$X_{op}(k)$在左半圆上的序列与 $X_{op}(k)$在右半圆上的序列是共轭反对称的。图 3.4.16 是 $X_{op}(k)$为实数时的示意图。当 $X_{op}(k)$为复数时,则应按式(3.4.55)的意义去理解共轭反对称,即实部相反、虚部相等。

$$\begin{cases} |X_{ep}(k)|=|X_{ep}(N-k)| \\ \arg[X_{ep}(k)]=-\arg[X_{ep}(N-k)] \end{cases} \tag{3.4.56}$$

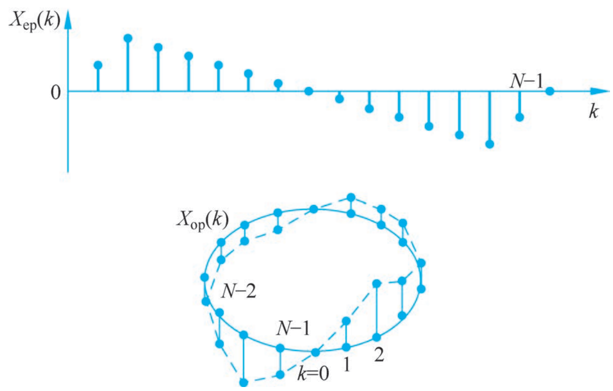


图 3.4.16　周期性共轭反对称分量 $X_{op}(k)$

通过上面的分析可知,对于一个复时间序列的 DFT 来说,序列的实部对应于 $X(k)$的周期性共轭对称分量 $X_{ep}(k)$;序列的虚部对应于 $X(k)$的周期性共轭反对称分量 $X_{op}(k)$。这种结果是有一定意义的。在此可以把两个实数序列 $x_1(n)$和 $x_2(n)$组合为单一的复数函数 $x(n)$,如式(3.4.47)所示。当算出复数表示的 $X(k)$后,即可利用式(3.4.49)和式(3.4.50)将 $X(k)$分成两个独立的分量 $X_{ep}(k)$和 $X_{op}(k)$,它们分别对应于 $x_1(n)$和 $x_2(n)$的离散傅里叶变换。因此利用这种方法,在一次计算中可得出两个独立信号的变换。

根据 $x_1(n)$与 $x_2(n)$的对称特性,同样可以找到 $X(k)$的实部、虚部与 $x(n)$的周期性共轭对称分量 $X_{ep}(k)$、周期性共轭反对称分量 $X_{op}(k)$的关系。请读者自行证明:$X_{ep}(k)$的 DFT 是 $X(k)$的实部 $\mathrm{Re}[X(k)]$,$X_{op}(k)$的 DFT 是 $X(k)$的虚部 $\mathrm{jIm}[X(k)]$。

5. 虚实序列的 DFT

1）虚序列的 DFT

当序列是纯虚序列，即 $x(n)=\mathrm{j}x_{\mathrm{i}}(n)$时，则由式(3.4.50)知其 $X(k)$只有周期性共轭反对称分量，即 $X(k)=x_{\mathrm{op}}(k)$，由式(3.4.54)可见，虚序列的离散傅里叶变换 $X(k)$的实部是奇对称的，虚部是偶对称的。

2）实序列的 DFT

若 $x(n)$是纯实数序列，即 $x(n)=x_{\mathrm{r}}(n)$时，由式(3.4.49)知其 $X(k)$只有周期性共轭对称分量，即 $X(k)=X_{\mathrm{ep}}(k)$，由式(3.4.54)知，其模是偶函数，而相角是奇函数，或者说，其实部是偶对称的，而虚部是奇对称的。

上述两种情况不论哪一种都只要知道一半数目的 $X(k)$，利用对称特性就可以得到另一半数目的 $X(k)$。在 DFT 中利用这个特点，可以提高运算效率。

联合考虑序列的奇偶特性及虚实特性，还可证明如下特性：

(1) 当序列是实的偶对称序列，则其离散傅里叶变换是实的偶对称的。

(2) 当序列是实的奇对称的序列，则其离散傅里叶变换是虚的奇对称的。

(3) 当序列是虚的偶对称的序列，则其离散傅里叶变换是虚的偶对称的。

(4) 当序列是虚的奇对称序列时，则其离散傅里叶变换是实的奇对称的。

在结束有关奇偶性及对称性的讨论时，为便于比较，将 DFT 的奇偶虚实特性列于表 3.4.1 中。

表 3.4.1 DFT 的奇偶虚实特性

$x(n)$	$X(k)$	$x(n)$	$X(k)$
偶序列	偶序列	实偶	实偶
奇序列	奇序列	实奇	虚奇
实	实部为偶，虚部为奇	虚偶	虚偶
虚	实部为奇，虚部为偶	虚奇	实奇

根据 $x(n)$与 $X(k)$变换关系的对称性，只要交换 $x(n)$和 $X(k)$列的标题，表中各项仍然是正确的。

应该指出，为了更有效地运用计算程序和更好地理解所得结果，对上述特性应深入地理解。

3.4.13 可将离散傅里叶变换看作一组滤波器

序列 $x(n)$的离散傅里叶变换

$$\begin{aligned}X(k)=\mathrm{DFT}[x(n)]&=\sum_{n=0}^{N-1}x(n)W_N^{nk}\\&=x(0)+x(1)W_N^{k}+x(2)W_N^{2k}+x(3)W_N^{3k}+\cdots+\\&\quad x(N-1)W_N^{(N-1)k}\end{aligned}\tag{3.4.57}$$

$X(k)$为序列 $x(n)$中 k 频率分量的大小，可看作如下的卷积形式

$$X(k)=\sum_{n=0}^{N-1}x(n)h_k(N-1-n) \tag{3.4.58}$$

令系统的单位采样响应为

$$h_k(n)=\begin{cases}W_N^{(N-1-n)k}, & n=0,1,2,\cdots,N-1\\ 0, & n<0\text{ 和 }n\geqslant N\end{cases} \tag{3.4.59}$$

输入序列为

$$0,0,0,x(0),x(1),x(2),\cdots,x(N-1),0,0,0,\cdots$$

此输入序列加入单位采样响应为 $h_k(n)$的滤波器后,在$(N-1)$时刻的输出为

$$\begin{aligned}y(N-1)&=\sum_{n=0}^{N-1}x(n)h_k(N-1-n)\\&=x(0)h_k(N-1)+x(1)h_k(N-2)+\\&\quad x(2)h_k(N-3)+\cdots+x(N-1)h_k(0)\\&=x(0)W_N^{(N-1-N+1)k}+x(1)W_N^{(N-1-N+2)k}+\\&\quad x(2)W_N^{(N-1-N+3)k}+\cdots+x(N-1)W_N^{(N-1)k}\\&=x(0)+x(1)W_N^k+x(2)W_N^{2k}+x(3)W_N^{3k}+\cdots+\\&\quad x(N-1)W_N^{(N-1)k}\end{aligned} \tag{3.4.60}$$

比较式(3.4.57)和式(3.4.60)可见,二者的结果是一样的。因此,求序列 $x(n)$的离散傅里叶变换 $X(k)$的过程相当于以序列 $x(n)$为输入,加到单位采样响应为 $h_k(n)$的滤波器,其在$(N-1)$时刻的输出就是 $X(k)$。由式(3.4.59)可见,$(N-1)$与 k 值有关,一定的 k 值对应一定的单位采样响应,取不同的 k 值可得不同的单位采样响应,分别对应不同滤波特性的数字滤波器。把 k 值取在 $0\leqslant k\leqslant N-1$ 区间,就可得 N 个不同滤波特性的数字滤波器。对 $x(n)$实施 N 个滤波器的滤波运算就可分别得到 N 个不同 k 值的 $X(k)$。所以离散傅里叶变换可看作一组滤波器。

下面分析这组滤波器的频率特性。为此先对 $h_k(n)$作 z 变换可得

$$\begin{aligned}H_k(z)&=\sum_{n=0}^{N-1}h_k(n)z^{-n}=\sum_{n=0}^{N-1}W_N^{(N-1-n)k}z^{-n}\\&=W_N^{(N-1)k}\sum_{n=0}^{N-1}(W_N^{-k}z^{-1})^n=W_N^{(N-1)k}\frac{1-z^{-N}W_N^{-Nk}}{1-z^{-1}W_N^{-k}}\end{aligned} \tag{3.4.61}$$

由于 $W_N=\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}}$,同时令 $z=\mathrm{e}^{\mathrm{j}\omega}$,代入式(3.4.61),得出滤波器的振幅频率特性为

$$|H_k(\mathrm{e}^{\mathrm{j}\omega})|=\left|\frac{1-(\mathrm{e}^{\mathrm{j}\omega})^{-N}(\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}k})^N}{1-(\mathrm{e}^{\mathrm{j}\omega})^{-1}(\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}k})^1}\right|=\left|\frac{\mathrm{e}^{\mathrm{j}\frac{N}{2}(\omega-\frac{2\pi}{N}k)}-\mathrm{e}^{-\mathrm{j}\frac{N}{2}(\omega-\frac{2\pi}{N}k)}}{\mathrm{e}^{\mathrm{j}\frac{1}{2}(\omega-\frac{2\pi}{N}k)}-\mathrm{e}^{-\mathrm{j}\frac{1}{2}(\omega-\frac{2\pi}{N}k)}}\right|$$

根据公式 $\sin x=\dfrac{\mathrm{e}^{\mathrm{j}x}-\mathrm{e}^{-\mathrm{j}x}}{2\mathrm{j}}$可得

$$|H_k(\mathrm{e}^{\mathrm{j}\omega})|=\left|\frac{\sin\frac{N}{2}\left(\omega-\frac{2\pi}{N}k\right)}{\sin\frac{1}{2}\left(\omega-\frac{2\pi}{N}k\right)}\right| \tag{3.4.62}$$

若令 $x=\frac{1}{2}\left(\omega-\frac{2\pi}{N}k\right)$，则这种振幅频率特性具有$\frac{\sin Nx}{\sin x}$的形式。在 $k=0$ 时，

$$H_0=\left|\frac{\sin\frac{N\omega}{2}}{\sin\frac{\omega}{2}}\right|$$

也是一种$\frac{\sin Nx}{\sin x}$的频率响应特性，当 $\omega=0$ 时，$H_0=N$，可作出其振幅特性如图 3.4.17 所示，第一副瓣电平约为-13.3dB。

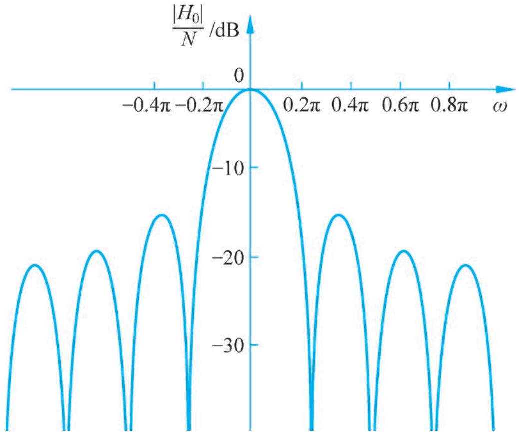


图 3.4.17　$k=0$ 时，H_0 的归一化振幅特性

若 $k\neq0$，而分别取 $1,2,3,\cdots,N-1$，则由式(3.4.62)可得出一组滤波器的振幅特性如图 3.4.18 所示。图中 $N=8$。各 H_k 的响应形式和 H_0 一样，最大幅值也为 N，差别在于主瓣移到了 $\omega=\frac{2\pi}{N}k$ 上。这些滤波器中心频率的间隔为 $\omega=\frac{2\pi}{N}$。

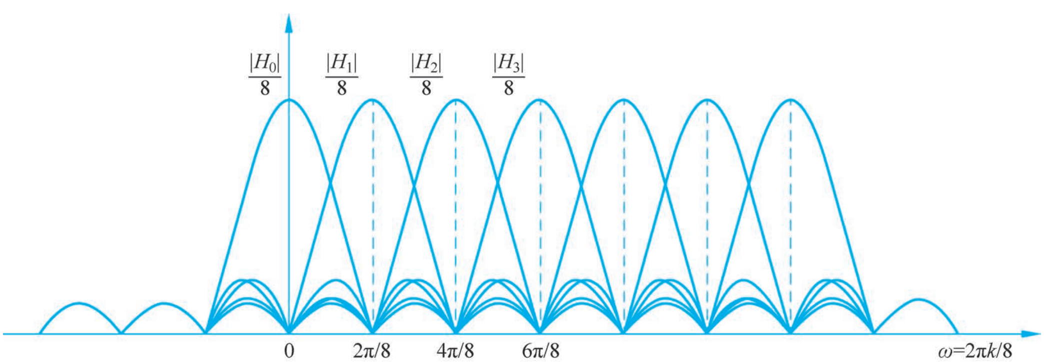


图 3.4.18　$N=8$ 时，不同 k 值的 H_k 的振幅特性

因此，离散傅里叶变换相当于用中心频率为 $\omega=\frac{2\pi}{N}k\,(k=0,1,2,\cdots,N-1)$，频率响应形式为$\frac{\sin Nx}{\sin x}$的 N 个滤波器对输入序列进行滤波。因为 $\omega=\Omega T=\frac{\Omega}{f_s}$，在 $\omega=\frac{2\pi}{N}$时，有 $\Omega=$

$\frac{2\pi}{N}f_s=\frac{2\pi}{NT}$。故当 $\omega=\frac{k2\pi}{N}$时这些滤波器在$(N-1)$时刻的输出即是频率分量为 $k\Omega$ 的傅里叶系数 $X(k)$。当输入序列 $x(n)$中所含频率分量不是 Ω 的整倍数,而是在$(k-1)\Omega$ 和 $k\Omega$ 之间时,输入序列经傅里叶变换后,在各个滤波器的输出端都可能有它的输出响应,而在第 k 个及第$(k-1)$个滤波器的输出响应中,将有一个或两个都具有较大值。所以利用离散傅里叶变换进行谱分析时,分辨能力决定于频谱谱线间隔 $\Omega=\frac{2\pi}{NT}=2\pi\frac{f_1}{N}$,并称$\frac{f_1}{N}$为频率分辨率或频率分辨单元。

离散傅里叶变换相当于一组滤波器,这一性质在脉冲多普勒雷达信号处理中得到了广泛的应用。

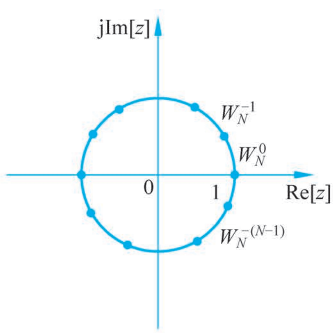
3.4.14 DFT 与 z 变换

列长为 N 的有限长序列 $x(n)$的 z 变换为 $X(z)$,而其离散傅里叶变换为 $X(k)$,$X(z)$和 $X(k)$二者均表示了同一有限长序列 $x(n)$的变换,它们之间的关系是:对 z 变换采样可得 DFT,而 DFT 的综合就是 z 变换。

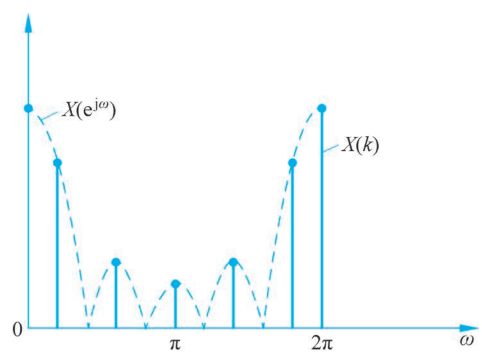
由前文的讨论可知,有限长序列 z 变换的收敛域是全部 z 平面,自然也包含了 z 平面的单位圆。如果在单位圆上等间隔取 N 个点,如图 3.4.19(a)所示,则在此 N 个点处的 z 变换值为

$$\begin{aligned}X(z_k)&=\sum_{n=0}^{N-1}x(n)z^{-n}\Big|_{z=z_k=\mathrm{e}^{\left(\mathrm{j}\frac{2\pi}{N}k\right)}=W_N^{-k}}\\&=\sum_{n=0}^{N-1}x(n)W_N^{nk}=\mathrm{DFT}[x(n)]=X(k)\end{aligned}\tag{3.4.63}$$

$z_k=W_N^{-k}=\mathrm{e}^{\mathrm{j}\left(\frac{2\pi}{N}\right)k}$ 是平面单位圆上幅角为 $\omega=\frac{2\pi}{N}k$ 的点,即 z 平面上单位圆 N 等分后的第 k 点。所以 $x(n)$DFT 的 N 个系数 $X(k)$也即 $x(n)$的 z 变换 $X(z)$单位圆上等距离的采样值,如图 3.4.19(b)所示。z 变换在单位圆上的值就是 $X(\mathrm{e}^{\mathrm{j}\omega})$。所以也可以说,$X(k)$是序列傅里叶变换 $X(\mathrm{e}^{\mathrm{j}\omega})$在相应点 $\omega=\frac{2\pi}{N}k=k\omega_N$ 上的采样值,其采样间隔为 $\omega=\frac{2\pi}{N}k$,即



(a) z平面单位圆上等间隔采样的各点



(b) $X(k)$是序列傅里叶变换$X(\mathrm{e}^{\mathrm{j}\omega})$的采样值

图 3.4.19 DFT 与 z 变换

$$\begin{cases} X(k)=X(\mathrm{e}^{\mathrm{j}k\omega_N}) \\ \omega_N=\dfrac{2\pi}{N} \end{cases}$$

由此可见，z 变换的采样可得 DFT。

3.5　频域采样

视频讲解

由上文讨论可知，采用 DFT 后实现了频域采样。对于任意一个频率特性能否用频率采样的办法去逼近呢？为此，首先应弄清它的限制，再研究经过频率采样后会有什么误差？如何消除误差？采样后所获得的频率特性怎样？

3.5.1　对 $X(z)$采样时采样点数的限制

对任一绝对可和的非周期序列 $x(n)$的 z 变换 $X(z)$在单位圆上，即对 $X(\mathrm{e}^{\mathrm{j}\omega})$进行等距采样，则由式(3.4.63)得

$$X(k)=X(z)\Big|_{z=W_N^{-k}}=\sum_{n=-\infty}^{\infty}x(n)W_N^{nk} \tag{3.5.1}$$

实现频域采样以后，信息有没有损失？能不能用序列的频谱 $X(\mathrm{e}^{\mathrm{j}\omega})$的采样值 $X(k)$恢复出原序列 $x(n)$？设频率采样后所得的 $X(k)$恢复出的有限长序列为 $x'(n)$，则

$$x'(n)=\mathrm{IDFT}[X(k)]$$

为了易于看清上述问题，本书先从周期序列 $\tilde{x}'(n)$开始研究

$$\tilde{x}'(n)=\mathrm{IDFS}[\widetilde{X}(k)]=\frac{1}{N}\sum_{k=0}^{N-1}\widetilde{X}(k)W_N^{-kn}=\frac{1}{N}\sum_{k=0}^{N-1}X(k)W_N^{-kn}$$

为探索频率采样后所恢复出的序列 $x'(n)$和原序列 $x(n)$之间的关系，将式(3.5.1)代入

$$\tilde{x}'(n)=\frac{1}{N}\sum_{k=0}^{N-1}\left[\sum_{m=-\infty}^{\infty}x(m)W_N^{mk}\right]W_N^{-kn}=\sum_{m=-\infty}^{\infty}x(m)\left[\frac{1}{N}\sum_{k=0}^{N-1}W_N^{(m-n)k}\right]$$

由于

$$\frac{1}{N}\sum_{k=0}^{N-1}W_N^{(m-n)k}=\begin{cases}1, & m=n+rN, r\ \text{为任意整数} \\ 0, & \text{其他}\ m\end{cases}$$

所以

$$\tilde{x}'(n)=\sum_{r=-\infty}^{\infty}x(n+rN) \tag{3.5.2}$$

式(3.5.2)说明 $\tilde{x}'(n)$是原序列 $x(n)$以 N 为周期的周期延拓序列。在第 2 章看到了时域的采样造成频域的周期延拓，现又证明了频域上的采样，同样也造成时域的周期延拓。这正是傅里叶变换中时域频域对称关系的反映。

如果 $x(n)$是列长为 M 的有限长序列，则当频域采样点数 $N<M$ 时，即频域的采样间隔不够密，$x(n)$的周期延拓就会出现某些序列交叠在一起，产生混叠现象。这样就不可能从 $x(n)$中提取一个周期不失真地恢复出原序列 $x(n)$来。因此对于列长为 M 的有限长序列 $x(n)$，频率采样不失真的条件是 $N\geqslant M$，这时有

$$x'(n)=\tilde{x}'(n)R_N(n)=\sum_{r=-\infty}^{\infty}x(n+rN)R_N(n)=x(n) \tag{3.5.3}$$

当 $x(n)$为无限长序列时,无论 N 取什么值,$x'(n)$都将不可能完全消除混叠误差,只能随着采样点 N 的增加而逐渐接近 $x(n)$。

3.5.2 $X(z)$的内插公式

由上文分析可知,列长为 N 的有限长序列 $x(n)$,可从单位圆上 $X(z)$的 N 个采样值 $X(k)$,即 $x(n)$的 DFT 恢复,因而这 N 个 $X(k)$也应该能完全表达整个 $X(z)$函数及频响 $X(\mathrm{e}^{\mathrm{j}\omega})$,也可以说 DFT 的综合就是 z 变换。这进一步揭示了 DFT 与 z 变换的关系。

$$\begin{cases} x(n)=\dfrac{1}{N}\displaystyle\sum_{k=0}^{N-1}X(k)W_N^{-nk} \\ X(z)=\displaystyle\sum_{n=0}^{N-1}x(n)z^{-n} \end{cases}$$

将 $x(n)$的表达式代入得

$$X(z)=\sum_{n=0}^{N-1}\left[\frac{1}{N}\sum_{k=0}^{N-1}X(k)W_N^{-nk}\right]z^{-n}=\frac{1}{N}\sum_{k=0}^{N-1}X(k)\sum_{n=0}^{N-1}(W_N^{-k}z^{-1})^n$$

$$=\frac{1}{N}\sum_{k=0}^{N-1}X(k)\frac{1-W_N^{-kN}z^{-N}}{1-W_N^{-k}z^{-1}}$$

因

$$W_N^{-kN}=\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}kN}=1$$

即

$$X(z)=\frac{1-z^{-N}}{N}\sum_{k=0}^{N-1}\frac{X(k)}{1-W_N^{-k}z^{-1}} \tag{3.5.4}$$

式(3.5.4)就是 $X(z)$的内插公式。在已知 $X(k)$时,可根据内插公式求得任意 z 点的 $X(z)$值,因此 $X(z)$的 N 个采样点的 $X(k)$值,包含了 z 变换的全部信息。式(3.5.4)可表示为式(3.5.5)的形式内插函数

$$X(z)=\sum_{k=0}^{N-1}X(k)\phi_k(z) \tag{3.5.5}$$

其中 $\phi_k(z)$为

$$\phi_k(z)=\frac{1}{N}\frac{1-z^{-N}}{1-W_N^{-k}z^{-1}} \tag{3.5.6}$$

将 $z=\mathrm{e}^{\mathrm{j}\omega}$ 代入,可以得到序列 $x(n)$的频响

$$X(\mathrm{e}^{\mathrm{j}\omega})=\sum_{k=0}^{N-1}X(k)\phi_k(\mathrm{e}^{\mathrm{j}\omega}) \tag{3.5.7}$$

$$\phi_k(\mathrm{e}^{\mathrm{j}\omega})=\frac{1}{N}\frac{1-\mathrm{e}^{-\mathrm{j}N\omega}}{1-\mathrm{e}^{-\mathrm{j}\left(\omega-k\frac{2\pi}{N}\right)}}=\frac{1}{N}\frac{\sin\left(\dfrac{\omega N}{2}\right)}{\sin\left[\left(\omega-\dfrac{k2\pi}{N}\right)\Big/2\right]}\cdot\mathrm{e}^{-\mathrm{j}\left(\frac{N\omega}{2}-\frac{\omega}{2}+\frac{k\pi}{N}\right)} \tag{3.5.8}$$

可将 $\phi_k(\mathrm{e}^{\mathrm{j}\omega})$ 表示为更明了的形式

$$\phi_k(\mathrm{e}^{\mathrm{j}\omega}) = \phi\left(\omega - k\frac{2\pi}{N}\right)$$

$$\phi(\omega) = \frac{1}{N}\frac{\sin\left(\frac{\omega N}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}\mathrm{e}^{-\mathrm{j}\omega\left(\frac{N-1}{2}\right)} \tag{3.5.9}$$

因此

$$X(\mathrm{e}^{\mathrm{j}\omega}) = \sum_{k=0}^{N-1} X(k)\phi\left(\omega - k\frac{2\pi}{N}\right) \tag{3.5.10}$$

内插函数 $\phi(\omega)$的振幅与相位特性如图 3.5.1 所示。当其变量 $\omega=0$(本采样点)时 $\phi(\omega)=1$,在其余采样点($\omega=i\frac{2\pi}{N}$,$i=1,2,\cdots,N-1$)上,$\phi(\omega)=0$。因而可知,有以下关系

$$\phi\left(\omega - k\frac{2\pi}{N}\right) = \begin{cases} 1, & \omega = k\frac{2\pi}{N} \\ 0, & \omega = i\frac{2\pi}{N}, i \neq k \end{cases}$$

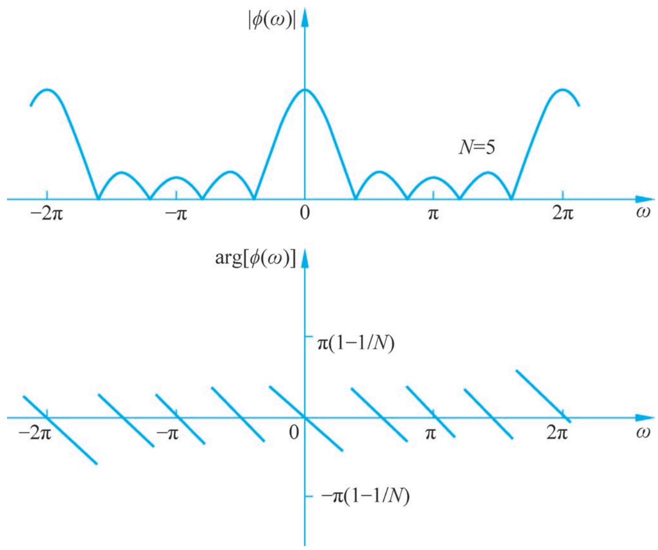


图 3.5.1　内插函数振幅特性与相位特性

即函数 $\phi\left(\omega-k\frac{2\pi}{N}\right)$在本采样点$\left(\omega=k\frac{2\pi}{N}\right)$上为 1,而在其他采样点$\left(\omega=i\frac{2\pi}{N},i\neq k\right)$上为 0。整个 $X(\mathrm{e}^{\mathrm{j}\omega})$正是 N 个 $\phi\left(\omega-k\frac{2\pi}{N}\right)$函数乘以加权 $X(k)$之和。明显可见,各个采样点处的 $X(\mathrm{e}^{\mathrm{j}\omega})$就等于各 $X(k)$的值,因为其余采样点的内插函数在这里都为零值。而采样点之间的 $X(\mathrm{e}^{\mathrm{j}\omega})$值则由各采样值乘以相应的内插函数延伸叠加形成。

内插函数的另一重要特点是具有如图 3.5.1 所示的线性相移特性。

综上所述,对于有限长序列的 $X(z)$及 $X(\mathrm{e}^{\mathrm{j}\omega})$可有时域序列 $x(n)$与频域序列 $X(k)$表示的两套表达式

$$
\begin{cases}
X(z)=\sum\limits_{n=0}^{N-1}x(n)z^{-n}\\
X(z)=\sum\limits_{k=0}^{N-1}X(k)\phi_k(z)
\end{cases}
$$

及

$$
\begin{cases}
X(\mathrm{e}^{\mathrm{j}\omega})=\sum\limits_{n=0}^{N-1}x(n)\mathrm{e}^{-\mathrm{j}\omega n}\\
X(\mathrm{e}^{\mathrm{j}\omega})=\sum\limits_{k=0}^{N-1}X(k)\phi_k\left(\omega-\dfrac{k2\pi}{N}\right)
\end{cases}
$$

时域采样定理说明，一个频带有限的信号，可对其进行时域采样而不丢失任何信息，因而，可用数字技术加工时域信号。频域采样理论说明，时间有限的信号（有限长序列），也可对其频域采样而不丢失任何信息。DFT 的理论使信号不仅在时域，而且在频域也离散化了，因而开辟了在频域采用数字技术处理的领域，而快速傅里叶变换（FFT）是计算 DFT 的快速有效算法。

视频讲解

3.6 用 DFT 对连续时间信号逼近的问题

DFT 的数学性质是确切的，但在许多信号处理的应用中，却很少作为一个最终目的被采用。对 DFT 感兴趣主要是因为它是连续傅里叶变换的一个近似。为了利用 DFT 对连续时间信号 $x_a(t)$进行傅里叶分析，需先对 $x_a(t)$进行采样，得到 $x(n)$，再对 $x(n)$进行 DFT 得 $X(k)$。$X(k)$是 $x(n)$的傅里叶变换 $X(\mathrm{e}^{\mathrm{j}\omega})$在频率区间$[0,2\pi]$上的 N 点等间隔采样。这里 $x(n)$和 $X(k)$均为有限长序列。但是，由傅里叶变换理论可知，若信号的持续时间为有限长，则其频谱无限宽；若信号的频谱为有限宽，则其持续时间无限长。严格地讲持续时间有限的带限信号是不存在的。为能满足 DFT 的变换条件，实际上对频谱很宽的信号，为防止时域采样后产生频谱混叠失真，可用前置滤波器滤除幅度较小的高频分量，使连续时间信号的带宽小于折叠频率。对于持续时间很长的信号，采样点数太多以致无法存贮和计算，只好截取为有限列长进行 DFT。从工程实际角度看，滤除幅度很小的高频分量和截去幅度很小部分的时间信号是允许的。由上述可见，用 DFT 对连续时间信号进行傅里叶分析必然是近似的，近似的准确程度严格地说是被分析波形的一个函数。特别要指出，两个变换之间的差异是因 DFT 需要对连续时间信号采样和截断为有限列长而产生的。这里主要有两个问题，即两种变换间相对数值的确定；以及在计算的变换与所需的变换之间造成误差的 3 种可能现象：混叠现象、栅栏效应和频谱泄漏。下面进行分别讨论。

3.6.1 计算的变换与所需变换间相对数值的确定

假设连续时间非周期性信号的变换定义为式(3.1.1)和式(3.1.2)，而周期信号的傅里叶级数的定义则为式(3.1.3)和式(3.1.4)。DFT 的计算可以按式(3.1.1)所定义的非周期信号的傅里叶变换，或式(3.1.3)所定义的周期信号的傅里叶系数的近似来进行。

如果采用 DFT 的基本定义式(3.3.1)去计算一个非周期性信号的傅里叶变换，则频谱

的正常幅度电平等于用 DFT 计算所得的频谱分量乘以 T。

如果已利用真实积分变换定义式(3.1.1)，或者如上所述，利用 DFT 乘以 T 来计算频谱函数，则可用包括因子 $1/N$ 在内的式(3.3.2)来近似计算式(3.1.2)所表示的时间函数。在此情况下，最后将各时间分量乘以 $NF=f$，即得正常幅度电平。所以从时间到频率，再从频率到时间的整个过程总共乘了 $T\cdot NF=1$ 也即幅度电平未受影响。

傅里叶级数系数的表示式(3.1.3)实际上是求被积式的平均值。如果采用式(3.3.1)的 DFT 公式来近似确定式(3.1.3)所表示的周期性函数的傅里叶级数系数时，频谱的正常幅度电平等于 DFT 所求出的频谱分量乘以 T，再除以 t_p(t_p 为周期性时间函数的有效周期)。由于$(T/t_p)=(1/N)$，因此，如果利用 DFT 的定义式(3.3.1)和式(3.3.2)来计算正规的傅里叶级数，则$(1/N)$因子最好放在 DFT 的正变换式中，而不是逆变换式中。

在许多实际问题中，各变换分量的实际电平往往无关紧要。确定信号特性的是频谱中各分量之间的相对电平，相对电平关系保持不变就可以了。

3.6.2　计算的变换与所需变换间的误差

1. 混叠现象

混叠现象在上文已详细讨论过。避免混叠现象的唯一办法是保证采样频率足够高。这就要求在确定采样频率之前，对频谱的性质要有所了解。下面着重讨论应用 DFT 时，为避免混叠所必须考虑的一些重要参数关系。

假设离散时间信号是从连续时间函数采样得出的，或者为了分析的目的选取相应的连续时间信号作为参考。并假设所处理的信号是基带信号，在采样前已利用低通模拟滤波器进行前置滤波，以避免高于折叠频率的分量出现。这样为避免混叠现象，要求采样频率

$$f_s \geqslant 2f_h \tag{3.6.1}$$

f_s 为信号的最高频率，而采样周期 T 必须满足

$$T \leqslant \frac{1}{2f_s} \tag{3.6.2}$$

设 F 表示频率分量间的增量，它就是前文提到的频率分辨率 $F=\dfrac{f_s}{N}$，t_p 为最小记录长度，也就是前文提到的周期性函数的有效周期。t_p 应按照所需的频率分辨率进行选择

$$t_p=\frac{1}{F} \tag{3.6.3}$$

式(3.6.2)和式(3.6.3)说明在高频容量与频率分辨率间存在着矛盾。增加高频容量，T 就必然减小，在采样点数 N 给定的情况下，记录长度必然缩短，从而降低了频率的分辨率。相反，要提高记录长度，分辨率就必须要增加，在采样点数 N 给定时，必然导致 T 的增加，因而减少了高频的容量。

在高频容量 f_h 与频率分辨率 F 两个参数中，保持其中一个不变而增加另一个的唯一办法，就是增加在一记录长度内的点数 N，如果 f_h 和 F 都已给定，则 N 必须满足

$$N \geqslant \frac{2f_h}{F} \tag{3.6.4}$$

这是未采用任何特殊数据处理(例如加窗处理)情况下,为实现基本的DFT算法所必须满足的最低条件。

2. 栅栏效应

栅栏效应是由于用DFT计算频谱只限制为基频的整数倍而不可能将频谱视为一个连续函数而产生的。就意义而言,栅栏效应表现为当用DFT计算整个频谱时,就好像通过一个"栅栏"来观看一个图景一样,只能在离散点的地方看到真实图景。如果不附加任何特殊处理,在两个离散的变换线之间若有一个特别大的频谱分量时,将无法检测出来。

减少栅栏效应的方法就是在原记录末端添加一些零值点来改变时间周期内的点数,并保持记录不变。从而在保持原有频谱连续形式不变的情况下,变更了谱线的位置。这样,原来看不到的频谱分量就能移动到可见的位置上。

必须注意,当在记录信号末端添加零点时,所用窗函数的宽度不能由于增加了零点而按较长的长度选择,而必须按照数据记录的实际长度来选择窗函数。关于窗函数的有关知识将在3.7节介绍。

3. 频谱泄漏

实际工作往往需要把信号的观察时间限制在一定的时间间隔之内。设有一个延伸到无限远处的离散时间信号 $x_1(n)$,其频谱为 $X_1(e^{j\omega})$(仅表示出周期频谱中周期的一部分)如图3.6.1所示。由于无法等待足够长的时间取用无限个数据,因此需要选择一段时间信号进行分析。

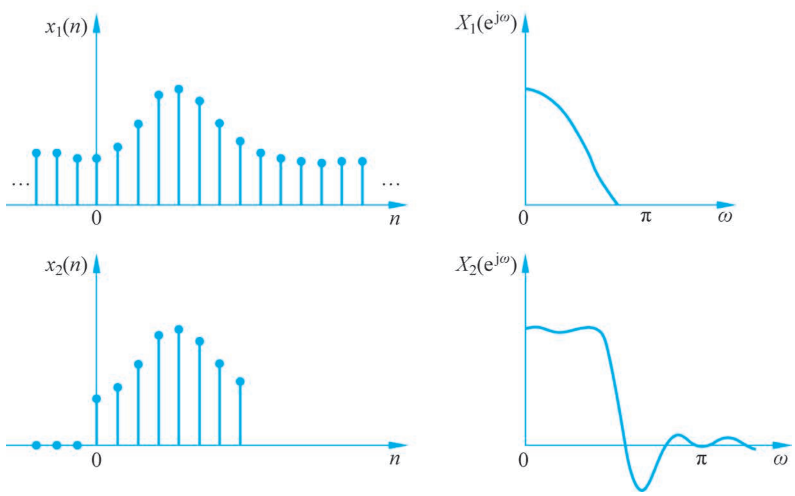


图3.6.1 信号截断时产生的频谱泄漏现象

取用有限个数据,即将信号截断的过程,就等于将信号乘以窗函数。如果窗函数是一个矩形窗函数,数据项数突然被截断,而窗内各项数据并不改变,如图3.6.1中 $x_2(n)$ 所示。时域的截断,在频域相当于所研究的波形的频谱 $X_1(e^{j\omega})$ 与矩形窗函数频谱周期卷积过程。这一卷积造成的失真频谱见图3.6.1中的 $X_2(e^{j\omega})$ 所示。

频谱分量从其正常频谱扩展开来,称为"泄漏"。如图3.6.2(a)所示的 $x_1'(t)$ 的频率 f_1 是 f_s/N 的整倍数,即

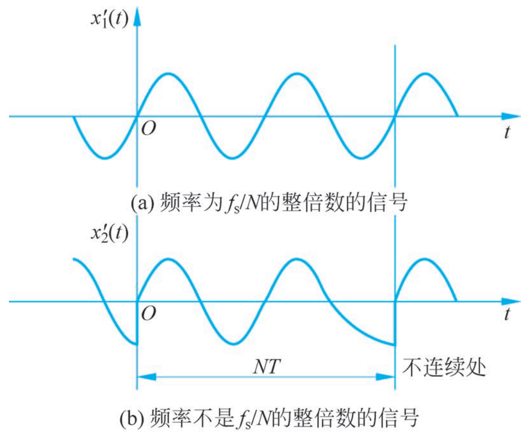


图 3.6.2 频率为 f_s/N 的整倍数及为 f_s/N 的非整倍数信号

$$f_1 = k\frac{f_s}{N} = \frac{k}{NT}$$

这说明在长度 NT 内信号有 k 整个周期。这时由 $x_1'(t)$构成的以 NT 为周期的周期信号是连续的。当 $x_2'(t)$的频率不是 f_s/N 的整倍数时，在 NT 的处理长度内，就不是恰好为信号周期的整数倍，由 $x_2'(t)$以 NT 为周期进行周期延拓所得到的周期信号就出现了不连续点，如图 3.6.2(b)所示，造成了频谱分量从其正常频谱扩展开来，在 DFT 等效滤波器组的各个副瓣将有大小不同的数值输出，再一次看到了频谱泄漏现象。应该指出，泄漏是不能与混叠完全分开的，因为泄漏将导致频谱的扩展，从而使频谱的最高频率超过折叠频率，造成混叠。因为无法取用无限个数据，所以在进行离散傅里叶变换时，时域中的截断是必须的，因此泄漏效应是离散傅里叶变换所固有的，必须设法进行抑制。

3.7 窗函数和加权

视频讲解

由上文分析可知，要抑制“泄漏”可以通过窗函数加权抑制 DFT 的等效滤波器的振幅特性的副瓣，或用窗函数加权使有限长度的输入信号周期延拓后在边界上尽量减少不连续程度的方法实现。在 FIR 数字滤波器设计中，为获得有限长单位采样响应，需要用窗函数截断无限长单位采样响应序列；在功率谱估计中也要用到窗函数加权问题。由此可见窗函数加权技术在数字信号处理中的重要地位。

在用窗函数加权时，有一个问题必须注意。一般文献上给出的窗函数 $w(n)$都是偶对称的时间序列，即

$$w(n) \quad n = -\frac{N}{2}, \cdots, -1, 0, 1, \cdots, +\frac{N}{2}$$

这里，N 是一个偶数，而窗函数 $w(n)$共有 $N+1$ 个采样值。但实际中常需要单边表示的窗函数，例如在 N 为偶数点的离散傅里叶变换时，处理区间为 $n=0\sim N-1$。因此，必须把偶对称表示的窗函数向右平移$\dfrac{N}{2}$点，让左端点与 $n=0$ 重合。如除去右端的一个采样值(一般为零值)，则窗函数 $w(n)$便在 $n=0,1,2,\cdots,N-1$ 点上定义，构成了适用于离散傅里

叶变换的单边窗函数序列。也可将被加权的序列向左平移$\frac{N}{2}$点,作用是一样的。位移$\frac{N}{2}$点只影响相位特性,并不影响振幅特性。本节主要研究加权的作用及窗函数的性能。

3.7.1 加权

现对离散傅里叶变换的等效滤波器的单位采样响应进行加权,即

$$h_k(n)=w(n)W_N^{(N-1-n)k}$$

式中,$w(n)$称为窗函数,用不同的窗函数对单位采样响应进行加权可以得到等效滤波器不同的振幅频率特性。

若 $w(n)=1,n=0,\cdots,N-1$,称为矩形窗。这时加权后的单位采样响应与等效滤波器原有的单位采样响应一样

$$h_k(n)=\begin{cases}W_N^{(N-1-n)k}, & n=0,1,2,\cdots,N-1\\0, & n<0\text{ 和 }n\geqslant N\end{cases}\tag{3.7.1}$$

其频率响应为

$$H_k(\mathrm{e}^{\mathrm{j}\omega})=\frac{\sin\left[\frac{N}{2}\left(\omega-\frac{2\pi}{N}k\right)\right]}{\sin\left[\frac{1}{2}\left(\omega-\frac{2\pi}{N}k\right)\right]}\mathrm{e}^{-\mathrm{j}\frac{N-1}{2}\left(\omega+\frac{2\pi}{N}k\right)}$$

其中,k 为零所对应的频率特性,由下式表示

$$H(\mathrm{e}^{\mathrm{j}\omega})=\frac{\sin\frac{N\omega}{2}}{\sin\frac{\omega}{2}}\mathrm{e}^{-\mathrm{j}\left(\frac{N-1}{2}\right)\omega}$$

上文已指出,这种频率特性的第一副瓣峰值比 $\omega=0$ 处的主瓣峰值低 13.3dB,约为主瓣幅度的 1/4～1/5。采用合适的窗函数进行加权,就能使对应的频率特性副瓣压降下来。例如,采用余弦加权

$$w(n)=\cos\pi\frac{n}{N},\quad n=-\frac{N}{2},\cdots,-1,0,1,\cdots,+\frac{N}{2}\tag{3.7.2}$$

为适用于离散傅里叶变换加权,需将偶对称的余弦窗移位$\frac{N}{2}$后采用正弦窗函数的形式,即

$$w(n)=\cos\left(\pi\frac{n}{N}-\frac{\pi}{2}\right)=\sin\pi\frac{n}{N}\tag{3.7.3}$$

这时对应的加权单位采样响应为

$$h_k(n)=\sin\left(\pi\frac{n}{N}\right)W_N^{(N-1-n)k}\quad n=0,1,2,\cdots,N-1\tag{3.7.4}$$

其频率特性可通过 z 变换求得

$$\begin{aligned}H_k(\mathrm{e}^{\mathrm{j}\omega})=H_k(z)\Big|_{z=\mathrm{e}^{\mathrm{j}\omega}}&=\sum_{n=0}^{N-1}\left[\sin\left(\pi\frac{n}{N}\right)W_N^{(N-1-n)k}\right]z^{-n}\\&=W^{(N-1)k}\sum_{n=0}^{N-1}\left[\sin\left(\pi\frac{n}{N}\right)W_N^{-kn}\right]z^{-n}\end{aligned}$$

暂不考虑求和号外的相移因子 $W_N^{(N-1)k}$，可求出

$$H_k(\mathrm{e}^{\mathrm{j}\omega})=\sum_{n=0}^{N-1}\left[\sin\left(\pi\frac{n}{N}\right)\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}kn}\right]z^{-n}$$

$$=\frac{1}{2\mathrm{j}}\sum_{n=0}^{N-1}\left[\mathrm{e}^{\mathrm{j}\left(\frac{\pi n}{N}\right)}-\mathrm{e}^{-\mathrm{j}\left(\frac{\pi n}{N}\right)}\right]\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}kn}\cdot\mathrm{e}^{-\mathrm{j}\omega n}$$

$$=\frac{1}{2\mathrm{j}}\sum_{n=0}^{N-1}\mathrm{e}^{\left\{-\mathrm{j}n\left[\omega-\left(k+\frac{1}{2}\right)\frac{2\pi}{N}\right]\right\}}-\frac{1}{2\mathrm{j}}\sum_{n=0}^{N-1}\mathrm{e}^{\left\{-\mathrm{j}n\left[\omega-\left(k-\frac{1}{2}\right)\frac{2\pi}{N}\right]\right\}}\tag{3.7.5}$$

把式(3.7.5)的最后两求和式总和起来，可得

$$H_k(\mathrm{e}^{\mathrm{j}\omega})=\frac{1}{2\mathrm{j}}\left\{\frac{\sin\frac{N}{2}\left[\omega-\left(k-\frac{1}{2}\right)\frac{2\pi}{N}\right]}{\sin\frac{1}{2}\left[\omega-\left(k+\frac{1}{2}\right)\frac{2\pi}{N}\right]}\mathrm{e}^{-\mathrm{j}\frac{N-1}{2}\left[\omega-\left(k+\frac{1}{2}\right)\frac{2\pi}{N}\right]}-\right.$$
$$\left.\frac{\sin\frac{N}{2}\left[\omega-\left(k-\frac{1}{2}\right)\frac{2\pi}{N}\right]}{\sin\frac{1}{2}\left[\omega-\left(k-\frac{1}{2}\right)\frac{2\pi}{N}\right]}\mathrm{e}^{-\mathrm{j}\frac{N-1}{2}\left[\omega-\left(k-\frac{1}{2}\right)\frac{2\pi}{N}\right]}\right\}$$

$$=\frac{1}{2\mathrm{j}}\left\{\frac{\sin\frac{N}{2}\left[\omega-\left(k+\frac{1}{2}\right)\frac{2\pi}{N}\right]}{\sin\frac{1}{2}\left[\omega-\left(k+\frac{1}{2}\right)\frac{2\pi}{N}\right]}-\right.$$
$$\left.\frac{\sin\frac{N}{2}\left[\omega-\left(k-\frac{1}{2}\right)\frac{2\pi}{N}\right]}{\sin\frac{1}{2}\left[\omega-\left(k-\frac{1}{2}\right)\frac{2\pi}{N}\right]}\mathrm{e}^{-\mathrm{j}\frac{(N-1)\pi}{N}}\right\}\mathrm{e}^{-\mathrm{j}\frac{N-1}{2}\left[\omega-\left(k+\frac{1}{2}\right)\frac{2\pi}{N}\right]}$$

由于 $\mathrm{e}^{-\mathrm{j}\frac{N-1}{N}\pi}=-\mathrm{e}^{\mathrm{j}\frac{\pi}{N}}$，故有

$$H_k(\mathrm{e}^{\mathrm{j}\omega})=\left\{\frac{1}{2}\frac{\sin\frac{N}{2}\left[\omega-\left(k+\frac{1}{2}\right)\frac{2\pi}{N}\right]}{\sin\frac{1}{2}\left[\omega-\left(k+\frac{1}{2}\right)\frac{2\pi}{N}\right]}\mathrm{e}^{-\mathrm{j}\frac{\pi}{2N}}+\frac{1}{2}\frac{\sin\frac{N}{2}\left[\omega-\left(k-\frac{1}{2}\right)\frac{2\pi}{N}\right]}{\sin\frac{1}{2}\left[\omega-\left(k-\frac{1}{2}\right)\frac{2\pi}{N}\right]}\mathrm{e}^{\mathrm{j}\frac{\pi}{2N}}\right\}\times$$
$$\mathrm{e}^{-\mathrm{j}\left\{\frac{N-1}{2}\left[\omega-\left(k+\frac{1}{2}\right)\frac{2\pi}{N}\right]+\frac{\pi}{2}-\frac{\pi}{2N}\right\}}\tag{3.7.6}$$

在式(3.7.6)的相位中还应加上文提到过的相位因子才是 $H_k(\mathrm{e}^{\mathrm{j}\omega})$的完整表达式。

由式(3.7.6)可知，余弦加权后的频率特性 $H_k(\mathrm{e}^{\mathrm{j}\omega})$是由两相邻的中心频率间隔为 $2\pi/N$ 的矩形窗加权的频率特性矢量合成的。由于 $N>1$，故括号内的两个振幅频率特性间的相位差 π/N 很小，因此矢量合成接近于代数相加，如图 3.7.1 所示。由图 3.7.1 可见，这时两个相邻的振幅频率特性的副瓣接近相减。因此通过选择合适的窗函数确实可以压低等效滤波器频率特性副瓣，达到抑制“频谱泄漏”的目的。由图 3.7.1 还可以看出，加权后还会引起频率特性的主瓣加宽，峰值响应降低的副作用。

也可以在离散傅里叶变换式中对输入序列 $x(n)$直接进行加权。这时，合适窗函数的加权作用是使被加权序列在边缘($n=0$ 和 $n=N-1$ 附近)比矩形窗函数圆滑而减小了陡峭的边缘所引起的副瓣分量。

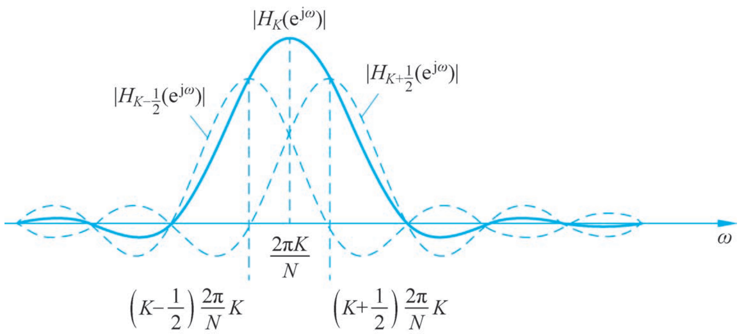


图 3.7.1 对离散傅里叶变换等效滤波器单位采样响应余弦加权后的频率响应

3.7.2 常用的窗函数

下面讨论几种常用的窗函数及其主要性能指标。各窗函数的幅度响应以分贝形式表示，定义为

$$W_{\mathrm{dB}}(\omega)=20\log_{10}\frac{|W(e^{j\omega})|}{W(e^{j0})} \tag{3.7.7}$$

$W(e^{j\omega})$为$w(n)$的傅里叶变换，$W(e^{j0})$为该变换的直流值。

1. 矩形窗(rectangular window)

$$w(n)=1,\quad n=-\frac{N}{2},\cdots,-1,0,1,\cdots,\frac{N}{2} \tag{3.7.8}$$

式中，N 为偶数。对序列的截断，实际上就是加矩形窗，其对应的频谱为

$$W(e^{j\omega})=\frac{\sin\frac{(N+1)}{2}\omega}{\sin\frac{\omega}{2}} \tag{3.7.9}$$

上述窗函数序列长为 $N+1$ 个点。例如取偶数点，即令

$$w(n)=1,\quad n=-\frac{N}{2},\cdots,-1,0,1,\cdots,\frac{N}{2} \tag{3.7.10}$$

对应的谱函数为

$$W(e^{j\omega})=W_R(e^{j\omega})=\sum_{n=-\frac{N}{2}}^{\left(\frac{N}{2}-1\right)}e^{-j\omega m}=e^{j\omega\frac{N}{2}}\sum_{n=0}^{N-1}e^{-j\omega n}=\frac{\sin\frac{N\omega}{2}}{\sin\frac{\omega}{2}}e^{j\frac{\omega}{2}} \tag{3.7.11}$$

以后本书就取式(3.7.11)所表示的函数为矩形窗频谱。

如果用幅度函数 $W(\omega)$与相位函数 $e^{-j\omega\alpha}$ 来表示窗函数的频谱 $W(e^{j\omega})$，即

$$W(e^{j\omega})=W(\omega)e^{-j\omega\alpha}$$

则上述窗函数的频谱幅度函数 $W(\omega)$具有$\frac{\sin Nx}{\sin x}$的形式，其频谱的主瓣宽度，以两个零交点之间的间隔计算，为 $2\times\frac{2\pi}{N}$，第一副瓣电平比主瓣峰值低 13dB 左右。图 3.7.2 给出了矩形窗函数及其对应的频谱函数。本节所述的所有窗函数及其频谱图，都是根据相应的窗函数

及其频谱函数表达式在计算机上绘成的。

单边表示的矩形窗函数为

$$w(n)=1,\quad n=0,1,2,\cdots,N-1 \tag{3.7.12}$$

它所对应的频谱函数为

$$W_{\mathrm{R}}(\mathrm{e}^{\mathrm{j}\omega})=\frac{\sin\frac{N\omega}{2}}{\sin\frac{\omega}{2}}\mathrm{e}^{-\mathrm{j}\left[\frac{N-1}{2}\omega\right]} \tag{3.7.13}$$

可以看出与式(3.7.11)相比,有一个序列移位$\frac{N}{2}$点所出现的相移因子 $\mathrm{e}^{-\mathrm{j}\frac{N}{2}\omega}$。

【例 3.7】 绘出矩形窗及其幅度响应。

```
import numpy as np
from scipy import signal, fft
import matplotlib.pyplot as plt
from matplotlib.ticker import MaxNLocator

# 计算矩形窗的 wn 值和幅度响应
N = 51                                          # 矩形窗长度
wn = signal.windows.boxcar(N)                   # 矩形窗的 wn 值
N0 = 2048
N1 = int(N0 / 2)
Ha = np.abs(fft.fft(wn, N0)) + 1e-10
Ha = Ha / np.max(Ha)
Ar = 20 * np.log10(Ha)
freq = np.linspace(0, 1, N1)

# 绘制矩形窗
fig, ax = plt.subplots()
ax.stem(wn, basefmt = "")
ax.set_title('矩形窗')
ax.set_xlabel('n')
ax.set_ylim([0, 1.5])
plt.rcParams['font.sans-serif'] = ['SimHei']    # 用来正常显示中文标签
fig.savefig('./win_rec1.png', dpi = 500)

# 绘制矩形窗的幅度响应
fig, ax = plt.subplots()
ax.plot(freq, Ar[:N1])
ax.grid()
ax.set_title('矩形窗的幅度响应')
ax.set_xlabel('k')
ax.set_xlabel(r'$ \omega / \pi $')
ax.set_ylabel(r'$ 20log_{10}|H(\omega)| $')
ax.set_xlim([0, 1])
ax.set_ylim([-100, 1])
ax.xaxis.set_major_locator(MaxNLocator(11))
ax.yaxis.set_major_locator(MaxNLocator(11))
plt.rcParams['axes.unicode_minus'] = False      # 用来显示负号
```

矩形窗的 $\omega(n)$和幅度响应绘制如图 3.7.2 所示。

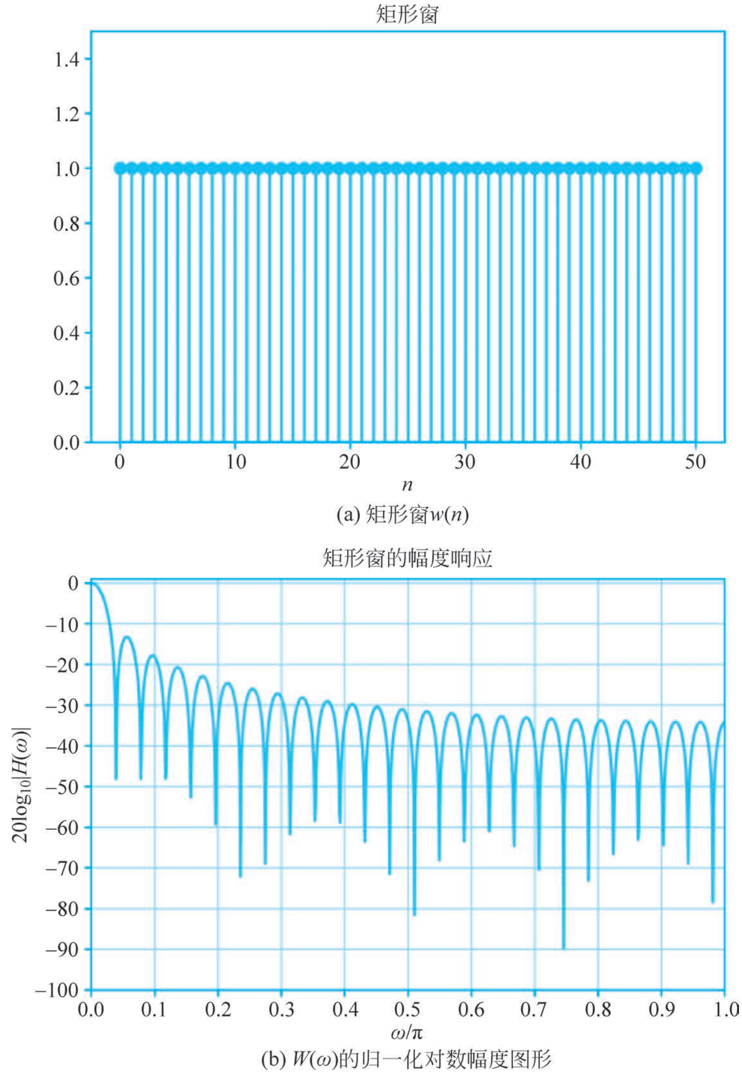


图 3.7.2 矩形窗及其频谱幅度宽度

2. 三角形窗(triangular window)

三角形窗又称巴特利特(Bartlett)窗。偶对称表达的三角形窗定义为

$$w(n)=1.0-\frac{|n|}{\frac{N}{2}},\quad n=-\frac{N}{2},\cdots,-1,0,1,\cdots,\frac{N}{2} \tag{3.7.14}$$

单边表示的三角形窗定义为

$$w(n)=\begin{cases}\dfrac{n}{\frac{N}{2}}, & n=0,1,2,\cdots,\dfrac{N}{2}\\[2ex] w(N-n), & n=\dfrac{N}{2},\dfrac{N}{2}+1,\cdots,N-1\end{cases} \tag{3.7.15}$$

它所对应的频谱函数为

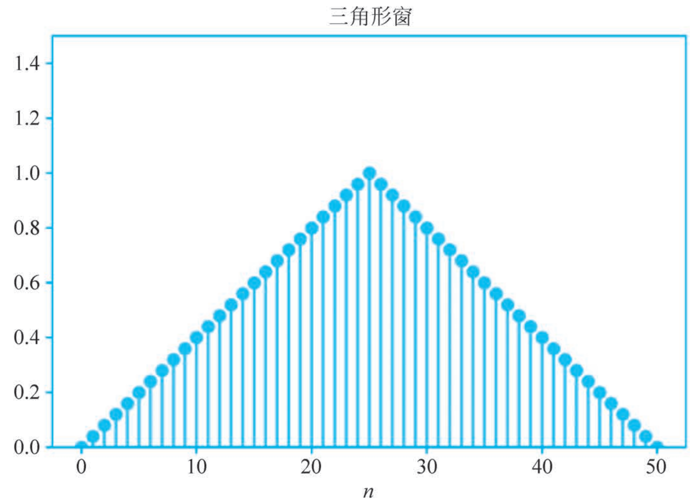
$$W(\mathrm{e}^{\mathrm{j}\omega})=\left[\frac{N}{2}\frac{\sin\left(\frac{N}{4}\omega\right)}{\sin\left(\frac{\omega}{2}\right)}\right]^2\mathrm{e}^{-\mathrm{j}\left[\left(\frac{N}{2}-1\right)\omega\right]} \tag{3.7.16}$$

其零交点之间的主瓣宽度是矩形窗的两倍，第一个副瓣电平比主瓣峰值低 26dB 左右。三角形窗是最简单的，频谱函数 $W(\mathrm{e}^{\mathrm{j}\omega})$ 为非负的一种窗函数。三角形窗序列及其频谱如图 3.7.3 所示。

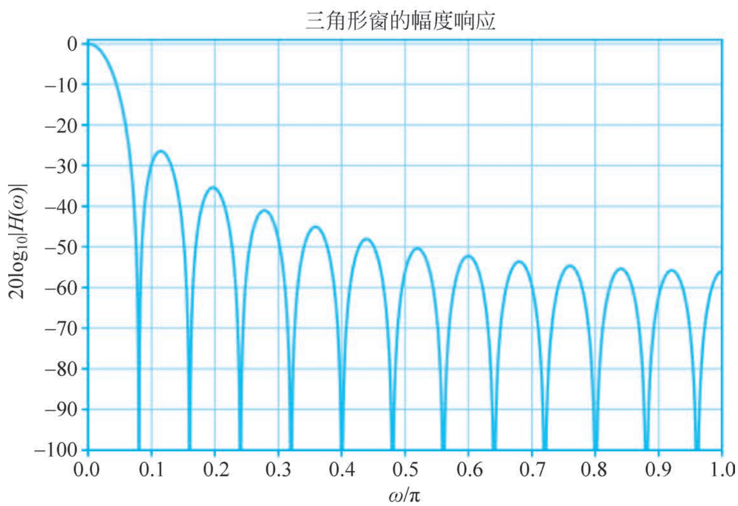
【例 3.8】 绘出三角形窗。

```
import numpy as np
from scipy import signal, fft
import matplotlib.pyplot as plt
from matplotlib.ticker import MaxNLocator

# 计算三角形窗的 wn 值和幅度响应
N = 51                                          # 三角形窗长度
wn = signal.windows.bartlett(N)                 # 三角形窗的 wn 值
N0 = 2048
N1 = int(N0 / 2)
Ha = np.abs(fft.fft(wn, N0)) + 1e-10
Ha = Ha / np.max(Ha)
Ar = 20 * np.log10(Ha)
freq = np.linspace(0, 1, N1)

# 绘制三角形窗
fig, ax = plt.subplots()
ax.stem(wn, basefmt = "")
ax.set_title('三角形窗')
ax.set_xlabel('n')
ax.set_ylim([0, 1.5])
plt.rcParams['font.sans-serif'] = ['SimHei']    # 用来正常显示中文标签
fig.savefig('./win_tri1.png', dpi = 500)

# 绘制三角形窗的幅度响应
fig, ax = plt.subplots()
ax.plot(freq, Ar[:N1])
ax.grid()
ax.set_title('三角形窗的幅度响应')
ax.set_xlabel('k')
ax.set_xlabel(r'$ \omega / \pi $')
ax.set_ylabel(r'$ 20log_{10}| H(\omega) | $')
ax.set_xlim([0, 1])
ax.set_ylim([-100, 1])
ax.xaxis.set_major_locator(MaxNLocator(11))
ax.yaxis.set_major_locator(MaxNLocator(11))
plt.rcParams['axes.unicode_minus'] = False      # 用来显示负号
```

三角形(巴特利特)窗的 $w(n)$ 和幅度响应绘制如图 3.7.3 所示。



(a) 三角形窗$w(n)$



(b) $W(\omega)$的归一化对数幅度图形

图 3.7.3　三角形窗及其频谱幅度函数

3. 汉宁窗(Hanning window)

汉宁窗也称余弦平方窗或升余弦窗,其偶对称表示式为

$$\begin{aligned}w(n)&=\cos^2\left(\frac{n}{N}\pi\right)\\&=\frac{1}{2}\left[1+\cos\left(\frac{2n}{N}\pi\right)\right]\\&=\frac{1}{2}+\frac{1}{2}\cos\left(\frac{2n}{N}\pi\right),\quad n=-\frac{N}{2},\cdots,-1,0,1,\cdots,+\frac{N}{2}\end{aligned}\tag{3.7.17}$$

单边表示为

$$w(n)=\sin^2\left(\frac{n}{N}\pi\right)=\frac{1}{2}\left[1-\cos\left(\frac{2n}{N}\pi\right)\right]\quad n=0,1,2,\cdots,N-1\tag{3.7.18}$$

由式(3.7.17),据欧拉公式可得

$$w(n)=\frac{1}{2}+\frac{1}{2}\left(\frac{1}{2}\mathrm{e}^{\mathrm{j}\frac{2n}{N}\pi}+\frac{1}{2}\mathrm{e}^{-\mathrm{j}\frac{2n}{N}\pi}\right)=\frac{1}{2}+\frac{1}{4}\mathrm{e}^{\mathrm{j}\frac{2n}{N}\pi}+\frac{1}{4}\mathrm{e}^{-\mathrm{j}\frac{2n}{N}\pi} \tag{3.7.19}$$

已知矩形窗的频谱函数是

$$w(n)=1\leftrightarrow W_{\mathrm{R}}(\mathrm{e}^{-\mathrm{j}\omega})=\frac{\sin\left(\frac{N\omega}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}\mathrm{e}^{\mathrm{j}\frac{\omega}{2}} \tag{3.7.20}$$

由变换的频移定理可得

$$\begin{cases} w(n)=1\cdot \mathrm{e}^{\mathrm{j}\frac{2\pi}{N}n}\leftrightarrow W_{\mathrm{R}}\left[\mathrm{e}^{\mathrm{j}\left(\omega-\frac{2\pi}{N}\right)}\right] \\ w(n)=1\cdot \mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}n}\leftrightarrow W_{\mathrm{R}}\left[\mathrm{e}^{\mathrm{j}\left(\omega+\frac{2\pi}{N}\right)}\right] \end{cases} \tag{3.7.21}$$

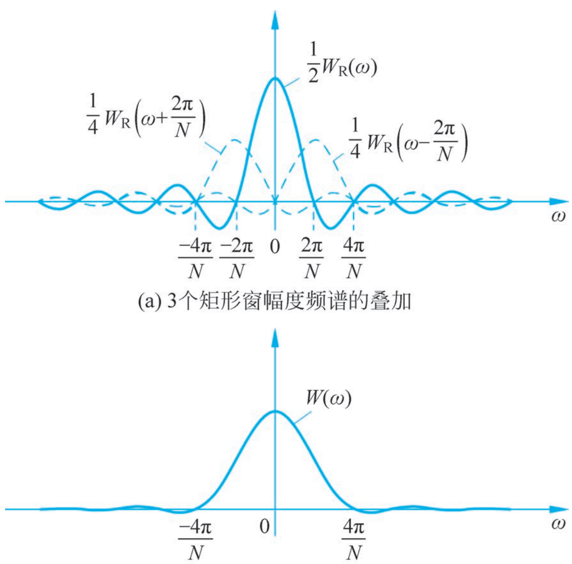
将频谱函数表达为

$$W(\mathrm{e}^{\mathrm{j}\omega})=W(\omega)\mathrm{e}^{-\mathrm{j}\omega\alpha}$$

这样，利用上述关系可以得到用矩形窗的频谱幅度函数 $W_{\mathrm{R}}(\omega)$来表示偶对称式汉宁窗的频谱函数

$$W(\omega)\approx\frac{1}{2}W_{\mathrm{R}}(\omega)+\frac{1}{4}W_{\mathrm{R}}\left(\omega-\frac{2\pi}{N}\right)+\frac{1}{4}W_{\mathrm{R}}\left(\omega+\frac{2\pi}{N}\right) \tag{3.7.22}$$

该频谱特性如图 3.7.4 所示，是由 3 个互有频移的不同幅值的矩形窗频谱幅度函数相加而成，这将使副瓣大为抵消，能量更有效地集中在主瓣内，但却使主瓣加宽了一倍。



(a) 3个矩形窗幅度频谱的叠加

(b) 汉宁窗谱$W(\omega)$主瓣

图 3.7.4　汉宁窗频谱

把 N 点偶对称表示的汉宁窗函数右移 $N/2$ 点，可得单边表示的汉宁窗，其对应的频谱函数为

$$\begin{aligned} W(\mathrm{e}^{\mathrm{j}\omega}) &=\frac{1}{2}W_{\mathrm{R}}(\omega)\mathrm{e}^{-\mathrm{j}\frac{N}{2}\omega}+\frac{1}{4}W_{\mathrm{R}}\left(\omega-\frac{2\pi}{N}\right)\mathrm{e}^{-\mathrm{j}\frac{N}{2}\left(\omega-\frac{2\pi}{N}\right)}+\frac{1}{4}W_{\mathrm{R}}\left(\omega+\frac{2\pi}{N}\right)\mathrm{e}^{-\mathrm{j}\frac{N}{2}\left(\omega+\frac{2\pi}{N}\right)} \\ &=\left[\frac{1}{2}W_{\mathrm{R}}(\omega)-\frac{1}{4}W_{\mathrm{R}}\left(\omega-\frac{2\pi}{N}\right)-\frac{1}{4}W_{\mathrm{R}}\left(\omega+\frac{2\pi}{N}\right)\right]\mathrm{e}^{-\mathrm{j}\frac{N}{2}\omega} \end{aligned} \tag{3.7.23}$$

注意,与偶对称表示的汉宁加权不同,式(3.7.23)中两个频移的矩形窗幅度函数前的符号是负的。在离散傅里叶变换中,直接实现对输入序列 $x(n)$ 的汉宁窗加权时,可不在时域进行 $x(n)$ 与 $w(n)$ 相乘,而在输出的频谱序列 $X(k)$ 上进行线性组合来实现。已知汉宁加权后的离散傅里叶变换 $X_W(k)$ 为

$$\begin{aligned}X_W(k)&=\sum_{n=0}^{N-1}w(n)x(n)\mathrm{e}^{-\mathrm{j}\left(\frac{2\pi}{N}\right)kn}\\&=\sum_{n=0}^{N-1}\left[\frac{1}{2}-\frac{1}{4}\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}n}-\frac{1}{4}\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}n}\right]x(n)\mathrm{e}^{-\mathrm{j}\left(\frac{2n}{N}\right)kn}\\&=\frac{1}{2}\sum_{n=0}^{N-1}x(n)\mathrm{e}^{-\mathrm{j}\left(\frac{2\pi}{N}\right)kn}-\frac{1}{4}\sum_{n=0}^{N-1}x(n)\mathrm{e}^{-\mathrm{j}\left(\frac{2n}{N}\right)(k-1)n}-\\&\quad\frac{1}{4}\sum_{n=0}^{N-1}x(n)\mathrm{e}^{-\mathrm{j}\left(\frac{2\pi}{N}\right)(k+1)n}\end{aligned}\tag{3.7.24}$$

由式(3.7.24)可知,汉宁窗加权的离散傅里叶变换输出 $X_W(k)$ 是矩形窗加权的离散傅里叶变换 $X(k)$ 的线性组合,即

$$\begin{aligned}X_W(k)&=\frac{1}{2}X(k)-\frac{1}{4}X(k-1)-\frac{1}{4}X(k+1)\\&=\frac{1}{2}\left\{X(k)-\frac{1}{2}[X(k-1)+X(k+1)]\right\}\end{aligned}\tag{3.7.25}$$

这种输出 $X(k)$ 的组合需要附加 $2N$ 次复加及 $2N$ 次右移(实现乘 1/2)操作来实现。汉宁窗加权的这些特点,在快速离散傅里叶变换运算中特别受到注意。其实,汉宁窗是一族窗函数中的一个,这族窗函数的偶对称表示式为

$$w(n)=\cos^{\alpha}\left(\frac{n\pi}{N}\right)\quad n=-\frac{N}{2},\cdots,-1,0,1,\cdots,\frac{N}{2}\tag{3.7.26}$$

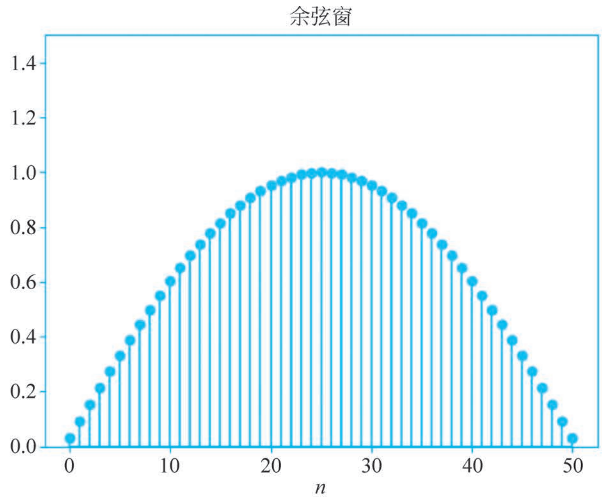
单边表示为

$$w(n)=\sin^{\alpha}\left(\frac{n\pi}{N}\right)\quad n=0,1,2,\cdots,N-1\tag{3.7.27}$$

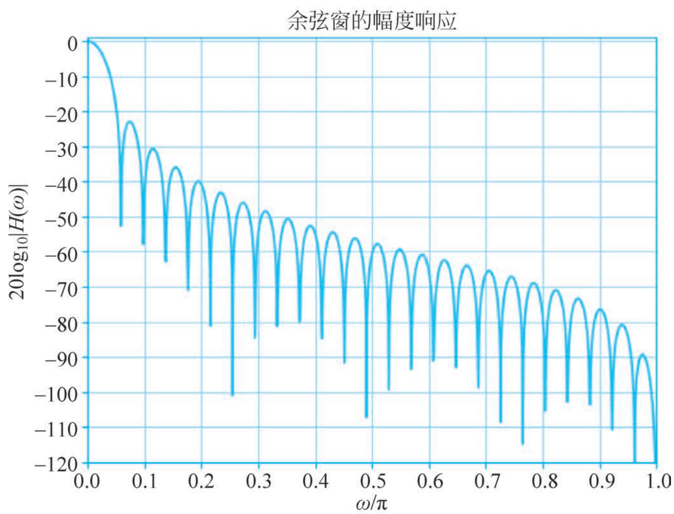
一般取 $\alpha=1,2,3,4$。当 $\alpha=1$ 时,是余弦窗。$\alpha=2$,是上文介绍的汉宁窗。当 α 越大时,窗函数 $\cos^{\alpha}(x)$ 序列越平滑,对应的频谱函数的副瓣电平就越下降,副瓣跌落越快速。但主瓣则变得越来越宽。图 3.7.5 给出了余弦窗及其频谱。图 3.7.6 给出了汉宁窗及其频谱。

【例 3.9】 绘出余弦窗及其频谱幅度函数。

```
import numpy as np
from scipy import signal, fft
import matplotlib.pyplot as plt
from matplotlib.ticker import MaxNLocator

# 计算余弦窗的 wn 值和幅度响应
N = 51                                              # 余弦窗长度
wn = signal.windows.cosine(N)                       # 余弦窗的 wn 值
N0 = 2048
N1 = int(N0 / 2)
Ha = np.abs(fft.fft(wn, N0)) + 1e-10
```



(a) 余弦窗$w(n)$



(b) $W(\omega)$的归一化对数幅度图形

图 3.7.5 余弦窗及其频谱幅度函数

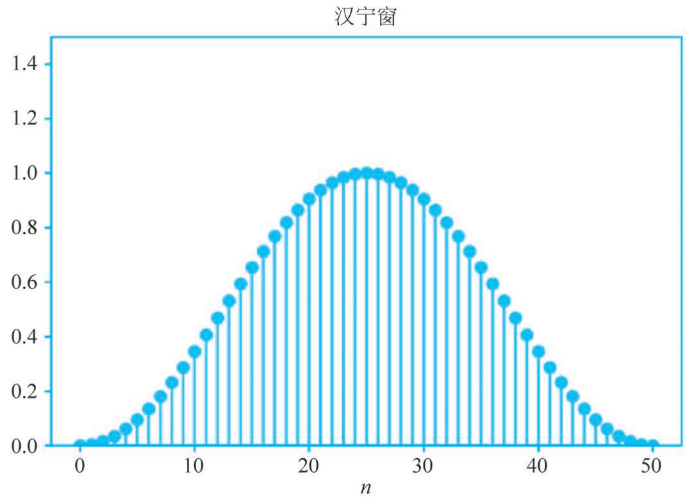
```
Ha = Ha / np.max(Ha)
Ar = 20 * np.log10(Ha)
freq = np.linspace(0, 1, N1)

# 绘制余弦窗
fig, ax = plt.subplots()
ax.stem(wn, basefmt = "")
ax.set_title('余弦窗')
ax.set_xlabel('n')
ax.set_ylim([0, 1.5])
plt.rcParams['font.sans-serif'] = ['SimHei']          # 用来正常显示中文标签
fig.savefig('./win_cos1.png', dpi = 500)

# 绘制余弦窗的幅度响应
fig, ax = plt.subplots()
ax.plot(freq, Ar[:N1])
```

```
ax.grid()
ax.set_title('余弦窗的幅度响应')
ax.set_xlabel('k')
ax.set_xlabel(r'$ \omega / \pi $')
ax.set_ylabel(r'$ 20log_{10}| H(\omega) | $')
ax.set_xlim([0, 1])
ax.set_ylim([-120, 1])
ax.xaxis.set_major_locator(MaxNLocator(11))
ax.yaxis.set_major_locator(MaxNLocator(13))
plt.rcParams['axes.unicode_minus'] = False                # 用来显示负号
```

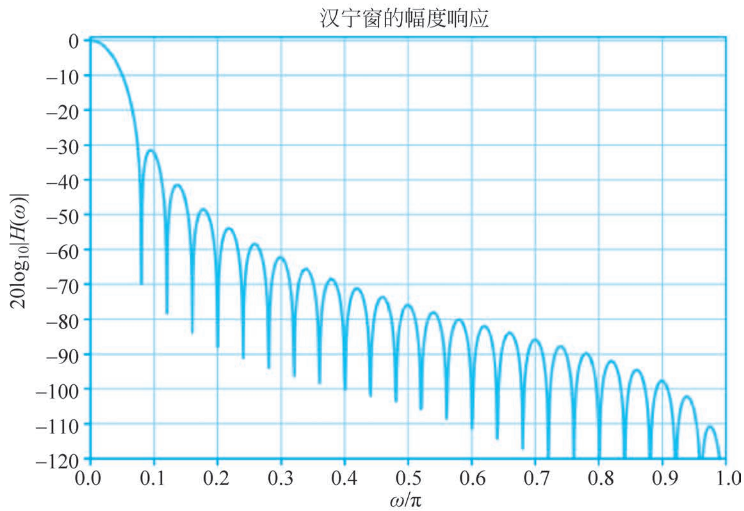
【例 3.10】 绘出汉宁窗及其幅度频谱函数。

```
import numpy as np
from scipy import signal, fft
import matplotlib.pyplot as plt
from matplotlib.ticker import MaxNLocator

# 计算汉宁窗的 wn 值和幅度响应
N = 51                                                    # 汉宁窗长度
wn = signal.windows.hann(N)                               # 汉宁窗的 wn 值
N0 = 2048
N1 = int(N0 / 2)
Ha = np.abs(fft.fft(wn, N0)) + 1e-10
Ha = Ha / np.max(Ha)
Ar = 20 * np.log10(Ha)
freq = np.linspace(0, 1, N1)

# 绘制汉宁窗
fig, ax = plt.subplots()
ax.stem(wn, basefmt="")
ax.set_title('汉宁窗')
ax.set_xlabel('n')
ax.set_ylim([0, 1.5])
plt.rcParams['font.sans-serif'] = ['SimHei']              # 用来正常显示中文标签
fig.savefig('./win_han1.png', dpi=500)

# 绘制汉宁窗的幅度响应
fig, ax = plt.subplots()
ax.plot(freq, Ar[:N1])
ax.grid()
ax.set_title('汉宁窗的幅度响应')
ax.set_xlabel('k')
ax.set_xlabel(r'$ \omega / \pi $')
ax.set_ylabel(r'$ 20log_{10}| H(\omega) | $')
ax.set_xlim([0, 1])
ax.set_ylim([-120, 1])
ax.xaxis.set_major_locator(MaxNLocator(11))
ax.yaxis.set_major_locator(MaxNLocator(13))
plt.rcParams['axes.unicode_minus'] = False                # 用来显示负号
```

汉宁窗的 $w(n)$ 和幅度响应绘制如下：



(a) 汉宁窗$w(n)$



(b) $W(\omega)$的归一化对数幅度图形

图 3.7.6　汉宁窗及其幅度频谱函数

4. 汉明窗(Hamming window)

汉明窗也称改进的升余弦窗。受汉宁窗启发,可设法调整相邻矩形窗频谱的大小以便更好地获取副瓣相消,为此可设加权窗函数为

$$w(n)=\alpha+(1-\alpha)\cos\left(\frac{2\pi}{N}n\right) \tag{3.7.28}$$

对应的频谱函数为

$$W(\omega)=\alpha W_R(\omega)+\frac{1}{2}(1-\alpha)\left[W_R\left(\omega-\frac{2\pi}{N}\right)+W_R\left(\omega+\frac{2\pi}{N}\right)\right] \tag{3.7.29}$$

改变可调整的比例系数 α 就可改变相邻 3 个矩形窗频谱幅度的大小,若选 $\alpha=0.54$,就是所谓“汉明窗”可使副瓣电平有显著的改善。

汉明窗的偶对称表示为

$$w(n)=0.54+0.46\cos\left(\frac{2\pi}{N}n\right),\quad n=-\frac{N}{2},\cdots,-1,0,1,\cdots,\frac{N}{2} \tag{3.7.30}$$

单边表示为

$$w(n)=0.54-0.46\cos\left(\frac{2\pi}{N}n\right),\quad n=0,1,2,\cdots,N-1 \tag{3.7.31}$$

所得到的频谱幅度函数为

$$W(\omega)=0.54W_R(\omega)+0.23\left[W_R\left(\omega-\frac{2\pi}{N}\right)+W_R\left(\omega+\frac{2\pi}{N}\right)\right] \tag{3.7.32}$$

结果达到99.96%的能量集中在主瓣内,在与汉宁窗相等的主瓣宽度下,获得了更好的副瓣抑制。若选 $\alpha=0.538\,56$,则副瓣电平是−43dB。图3.7.7给出了汉明窗及其频谱,可以看到,在第一副瓣处出现了很深的凹陷。汉明加权后的离散傅里叶变换也可用 $X(k)$ 来表达

$$X_w(k)=0.54X(k)-0.23[X(k-1)+X(k+1)] \tag{3.7.33}$$

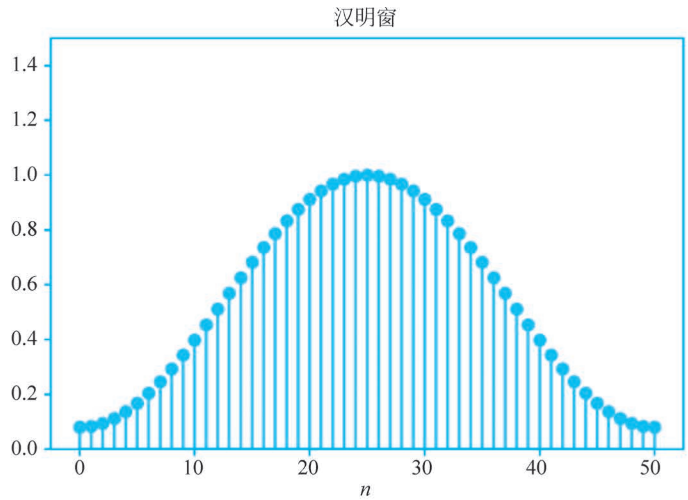
它不能像汉宁窗那样用简单的右移来代替与系数作乘法,而需做实数乘复数的乘法。

【例3.11】 绘出汉明窗及其幅度频谱函数。

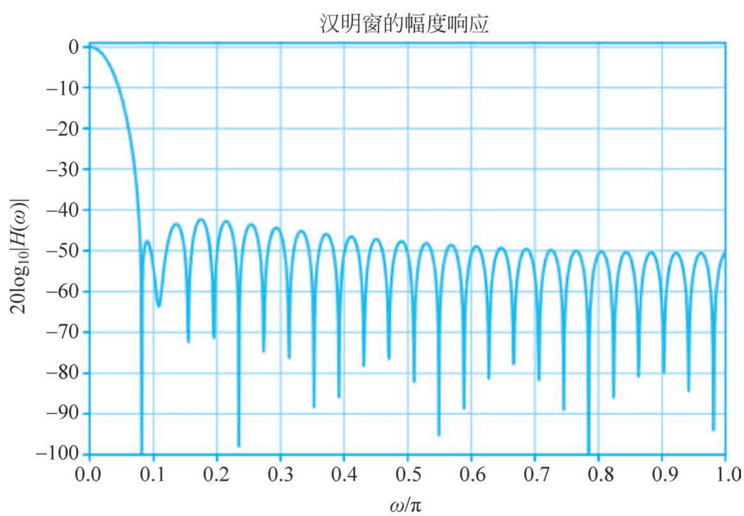
```
import numpy as np
from scipy import signal, fft
import matplotlib.pyplot as plt
from matplotlib.ticker import MaxNLocator

# 计算汉明窗的 wn 值和幅度响应
N = 51                                   # 汉明窗长度
wn = signal.windows.hamming(N)           # 汉明窗的 wn 值
N0 = 2048
N1 = int(N0 / 2)
Ha = np.abs(fft.fft(wn, N0)) + 1e-10
Ha = Ha / np.max(Ha)
Ar = 20 * np.log10(Ha)
freq = np.linspace(0, 1, N1)

# 绘制汉明窗
fig, ax = plt.subplots()
ax.stem(wn, basefmt="")
ax.set_title('汉明窗')
ax.set_xlabel('n')
ax.set_ylim([0, 1.5])
plt.rcParams['font.sans-serif'] = ['SimHei']       # 用来正常显示中文标签
fig.savefig('./win_ham1.png', dpi=500)

# 绘制汉明窗的幅度响应
fig, ax = plt.subplots()
ax.plot(freq, Ar[:N1])
ax.grid()
ax.set_title('汉明窗的幅度响应')
ax.set_xlabel('k')
ax.set_xlabel(r'$\omega / \pi $')
```

```
ax.set_ylabel(r'$ 20log_{10}|H(\omega)| $')
ax.set_xlim([0, 1])
ax.set_ylim([-100, 1])
ax.xaxis.set_major_locator(MaxNLocator(11))
ax.yaxis.set_major_locator(MaxNLocator(11))
plt.rcParams['axes.unicode_minus'] = False              # 用来显示负号
```

汉明窗的 $w(n)$ 和幅度响应绘制如图 3.7.7 所示。



(a) 汉明窗$w(n)$



(b) $W(\omega)$的归一化对数幅度图形

图 3.7.7　汉明窗及其幅度频谱函数

5. 布拉克曼窗(Blackman window)

布拉克曼窗也称二阶升余弦窗。汉宁、汉明加权都由 3 个中心频率不同的矩形窗频谱线性组合而成。布拉克曼利用更多的矩形窗频谱线性组合构成布拉克曼窗，其偶对称表示为

$$w(n)=\sum_{m=0}^{K-1} a_m \cos\left(\frac{2\pi}{N}mn\right),\quad n=-\frac{N}{2},\cdots,-1,0,1,\cdots,\frac{N}{2} \tag{3.7.34}$$

单边表示为

$$w(n)=\sum_{m=0}^{K-1}(-1)^m a_m \cos\left(\frac{2\pi}{N}mn\right),\quad n=0,1,2,\cdots,N-1 \tag{3.7.35}$$

单边表示的布拉克曼窗幅度频谱函数为

$$W(\omega)=\sum_{m=0}^{K-1}(-1)^m \frac{a_m}{2}\left[W_R\left(\omega-\frac{2\pi}{N}m\right)+W_R\left(\omega+\frac{2\pi}{N}m\right)\right] \tag{3.7.36}$$

这个窗函数中系数的选择应满足以下约束条件

$$\sum_{m=0}^{K-1} a_m = 1.0 \tag{3.7.37}$$

因此,汉宁、汉明窗是 a_0 和 a_1 不为零,而其他系数都为零的布拉克曼窗。假如布拉克曼窗有 K 个非零的系数 a_m,则其振幅频谱将由$(2K-1)$个中心频率不同的矩形窗频谱线性组合而成。显然,要使窗函数频谱的主瓣宽度窄,则 K 值不能选得很大。布拉克曼找到 $K=3$ 时在 $\omega=3.5(2\pi/N)$及 $\omega=4.5(2\pi/N)$处出现零点的窗序列系数 $a_m(m=0,1,2)$,其准确值及近似值如下式所示

$$\left.\begin{aligned} a_0&=\frac{7938}{10\,608}\approx 0.42\\ a_1&=\frac{9240}{18\,608}\approx 0.5\\ a_2&=\frac{1430}{18\,608}\approx 0.08 \end{aligned}\right\} \tag{3.7.38}$$

按照式(3.7.38)的系数近似得出的窗函数,称为布拉克曼窗,其偶对称表示为

$$\begin{gathered} w(n)=0.42+0.50\cos\left(\frac{2\pi}{N}n\right)+0.08\cos\left(\frac{2\pi}{N}2n\right)\\ n=-\frac{N}{2},\cdots,-1,0,1,\cdots,\frac{N}{2} \end{gathered} \tag{3.7.39}$$

单边表示为

$$\begin{gathered} w(n)=0.42-0.50\cos\left(\frac{2\pi}{N}n\right)+0.08\cos\left(\frac{2\pi}{N}2n\right)\\ n=0,1,2,\cdots,N-1 \end{gathered} \tag{3.7.40}$$

单边表示的频谱幅度函数为

$$\begin{aligned} W(\omega)=0.42W_R(\omega)+0.25\left[W_R\left(\omega-\frac{2\pi}{N}\right)+W_R\left(\omega+\frac{2\pi}{N}\right)\right]+\\ 0.04\left[W_R\left(\omega-\frac{4\pi}{N}\right)+W_R\left(\omega+\frac{4\pi}{N}\right)\right] \end{aligned} \tag{3.7.41}$$

这样可以得到更低的副瓣,但主瓣宽度却进一步加宽到矩形窗的 3 倍。

【例 3.12】 绘出布拉克曼窗及其幅度频谱函数。

```
import numpy as np
from scipy import signal, fft
import matplotlib.pyplot as plt
```

```
from matplotlib.ticker import MaxNLocator

# 计算布拉克曼窗的 wn 值和幅度响应
N = 51                                          # 布拉克曼窗长度
wn = signal.windows.blackman(N)                 # 布拉克曼窗的 wn 值
N0 = 2048
N1 = int(N0 / 2)
Ha = np.abs(fft.fft(wn, N0)) + 1e-10
Ha = Ha / np.max(Ha)
Ar = 20 * np.log10(Ha)
freq = np.linspace(0, 1, N1)

# 绘制布拉克曼窗
fig, ax = plt.subplots()
ax.stem(wn, basefmt="")
ax.set_title('布拉克曼窗')
ax.set_xlabel('n')
ax.set_ylim([0, 1.5])
plt.rcParams['font.sans-serif'] = ['SimHei']    # 用来正常显示中文标签
fig.savefig('./win_blac1.png', dpi=500)

# 绘制布拉克曼窗的幅度响应
fig, ax = plt.subplots()
ax.plot(freq, Ar[:N1])
ax.grid()
ax.set_title('布拉克曼窗的幅度响应')
ax.set_xlabel('k')
ax.set_xlabel(r'$ \omega / \pi $')
ax.set_ylabel(r'$ 20log_{10}|H(\omega)| $')
ax.set_xlim([0, 1])
ax.set_ylim([-150, 1])
ax.xaxis.set_major_locator(MaxNLocator(11))
ax.yaxis.set_major_locator(MaxNLocator(16))
plt.rcParams['axes.unicode_minus'] = False      # 用来显示负号
```

布拉克曼窗的 $w(n)$ 和幅度响应绘制如图 3.7.8 所示。

6. 最优化窗

最优化窗是指在某种优化准则下得出的窗函数。

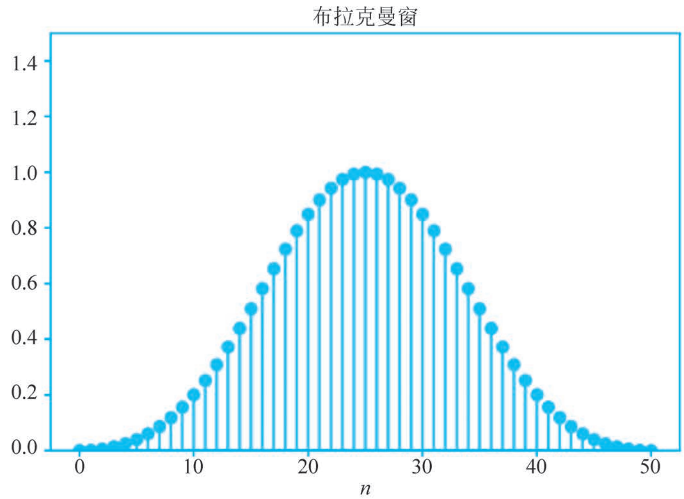
1）高斯窗（Gaussian window）

已经知道，任何一种信号时宽 T、频宽 W 之积 TW 满足下式

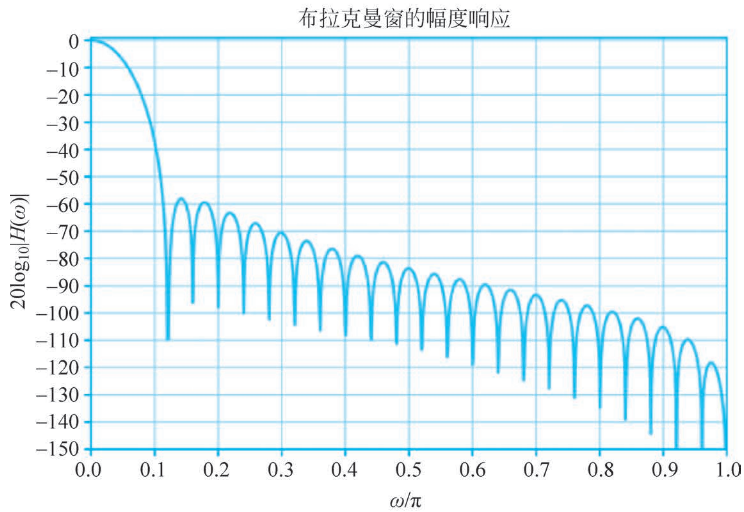
$$TW \geqslant \frac{1}{4\pi} \tag{3.7.42}$$

在最小时宽频宽积$\left(TW=\frac{1}{4\pi}\right)$的优化准则下得出的是高斯波形，因此，可选择高斯波形作窗函数。高斯波形是一直延伸到无穷大的，但是若在波形 3 倍均方值的地方进行截断，则误差很小。高斯窗如下式所示

$$w(n)=\mathrm{e}^{-\frac{1}{2}\left[a\frac{n}{N/2}\right]^2} \quad 0 \leqslant |n| \leqslant \frac{N-1}{2} \tag{3.7.43}$$



(a) 布拉克曼窗$w(n)$



(b) $W(\omega)$的归一化对数幅度图形

图 3.7.8 布拉克曼窗及其幅度频谱函数

有限列长的高斯形窗相当于高斯形窗与矩形窗相乘。因此,有限列长的高斯形窗的频谱函数应是高斯窗频谱与矩形窗频谱的卷积,即

$$W(\omega)=\frac{1}{2}\frac{\sqrt{2\pi}}{a}\mathrm{e}^{-\frac{1}{2}\left[\frac{\omega}{a}\right]^2} * W_R(\omega)$$

$$\approx\frac{N}{2}\frac{\sqrt{2\pi}}{a}\mathrm{e}^{-\frac{1}{2}\left[\frac{\omega}{a}\right]^2},\quad 当\ a>2.5,\omega\ 很小时 \tag{3.7.44}$$

a 是标准差值的倒数,是频谱宽度的一种度量。图 3.7.9 为 $a=3$ 时的高斯窗及其频谱。

【例 3.13】 绘出 $a=3$ 时的高斯窗及其频谱幅度函数。

```
import numpy as np
from scipy import signal, fft
```

```
import matplotlib.pyplot as plt
from matplotlib.ticker import MaxNLocator

# 计算高斯窗的 wn 值和幅度响应
N = 51                                         # 高斯窗长度
wn = signal.windows.gaussian(N, (N-1)/(2*3))  # 高斯窗的 wn 值
N0 = 2048
N1 = int(N0 / 2)
Ha = np.abs(fft.fft(wn, N0)) + 1e-10
Ha = Ha / np.max(Ha)
Ar = 20 * np.log10(Ha)
freq = np.linspace(0, 1, N1)

# 绘制高斯窗
fig, ax = plt.subplots()
ax.stem(wn, basefmt="")
ax.set_title('高斯窗(a = 3)')
ax.set_xlabel('n')
ax.set_ylim([0, 1.5])
plt.rcParams['font.sans-serif'] = ['SimHei']    # 用来正常显示中文标签
fig.savefig('./win_gauss1.png', dpi=500)

# 绘制高斯窗的幅度响应
fig, ax = plt.subplots()
ax.plot(freq, Ar[:N1])
ax.grid()
ax.set_title('高斯窗的幅度响应(a = 3)')
ax.set_xlabel('k')
ax.set_xlabel(r'$ \omega / \pi $')
ax.set_ylabel(r'$ 20log_{10}| H(\omega) | $')
ax.set_xlim([0, 1])
ax.set_ylim([-100, 1])
ax.xaxis.set_major_locator(MaxNLocator(11))
ax.yaxis.set_major_locator(MaxNLocator(11))
plt.rcParams['axes.unicode_minus'] = False      # 用来显示负号
```

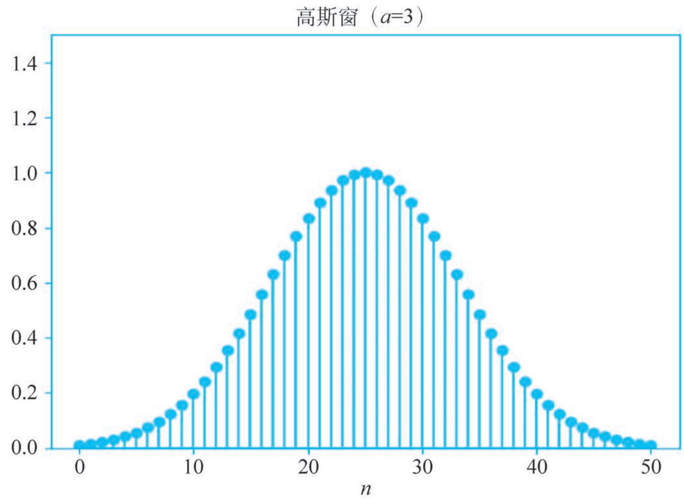
当 $N=51,\alpha=3,\sigma=\dfrac{N-1}{2\alpha}=\dfrac{50}{6}$时，高斯窗的 $w(n)$和幅度响应绘制如图 3.7.9 所示。

2）凯塞窗（Kaiser window）

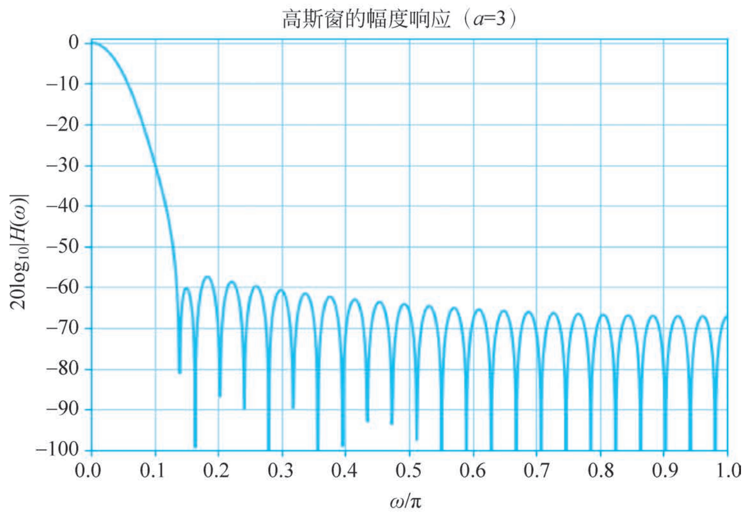
凯塞窗的最优化准则是：对于有限的信号能量，要求确定一个有限时宽 T 的信号波形，它使得频宽 W 内的能量最大。凯塞窗的定义如下

$$w(n)=\frac{I_0\left[\pi a\sqrt{1.0-\left(\frac{n}{N/2}\right)^2}\right]}{I_0(\pi a)},\quad 0\leqslant|n|\leqslant\frac{N}{2} \tag{3.7.45}$$

式中，$I_0(x)=1+\sum\limits_{k=1}^{\infty}\left[\frac{1}{k!}\left(\frac{x}{2}\right)^k\right]^2$ 是零阶第一类修正贝塞尔函数。凯塞窗的频谱幅度函数可近似为



(a) 高斯窗$w(n)$



(b) $W(\omega)$的归一化对数幅度图形

图 3.7.9 $a=3$ 时的高斯窗及其频谱幅度函数

$$W(\omega)\approx\frac{N}{I_0(a\pi)}\frac{\sinh\left[\sqrt{a^2\pi^2-\left(\frac{N\omega}{2}\right)^2}\right]}{\sqrt{a^2\pi^2-\left(\frac{N\omega}{2}\right)^2}}\tag{3.7.46}$$

上面公式中的 πa 是时宽、频宽积的一半。

【例 3.14】 绘出凯塞窗及其幅度响应。

```
import numpy as np
from scipy import signal, fft
import matplotlib.pyplot as plt
from matplotlib.ticker import MaxNLocator

# 计算凯塞窗的 wn 值和幅度响应
N = 51                                            # 凯塞窗长度
```

```
wn = signal.windows.kaiser(N, 14)            # 凯塞窗的 wn 值
N0 = 2048
N1 = int(N0 / 2)
Ha = np.abs(fft.fft(wn, N0)) + 1e-10
Ha = Ha / np.max(Ha)
Ar = 20 * np.log10(Ha)
freq = np.linspace(0, 1, N1)

# 绘制凯塞窗
fig, ax = plt.subplots()
ax.stem(wn, basefmt="")
ax.set_title('凯塞窗(πa = 14)')
ax.set_xlabel('n')
ax.set_ylim([0, 1.5])
plt.rcParams['font.sans-serif'] = ['SimHei']  # 用来正常显示中文标签
fig.savefig('./win_kaiser1.png', dpi=500)

# 绘制凯塞窗的幅度响应
fig, ax = plt.subplots()
ax.plot(freq, Ar[:N1])
ax.grid()
ax.set_title('凯塞窗的幅度响应(πa = 14)')
ax.set_xlabel('k')
ax.set_xlabel(r'$ \omega / \pi $')
ax.set_ylabel(r'$ 20log_{10}| H(\omega) | $')
ax.set_xlim([0, 1])
ax.set_ylim([-150, 1])
ax.xaxis.set_major_locator(MaxNLocator(11))
ax.yaxis.set_major_locator(MaxNLocator(16))
plt.rcParams['axes.unicode_minus'] = False     # 用来显示负号
```

当 $\pi a=14$ 时,凯塞窗的 $w(n)$ 和幅度响应绘制如图 3.7.10 所示。

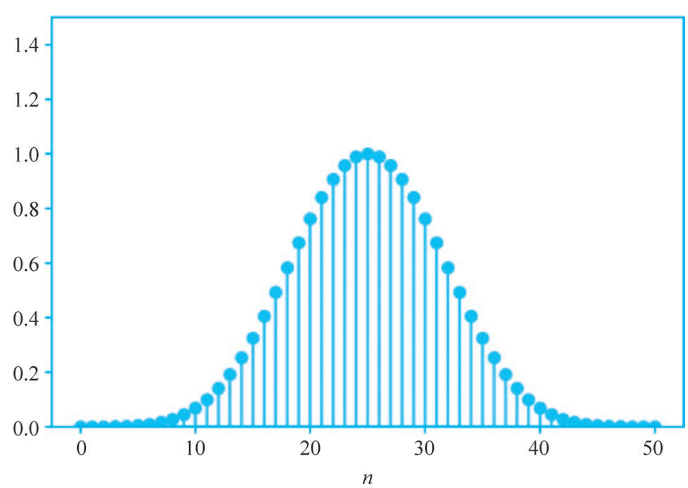


图 3.7.10　凯塞窗及其幅度响应($\pi a=14$)

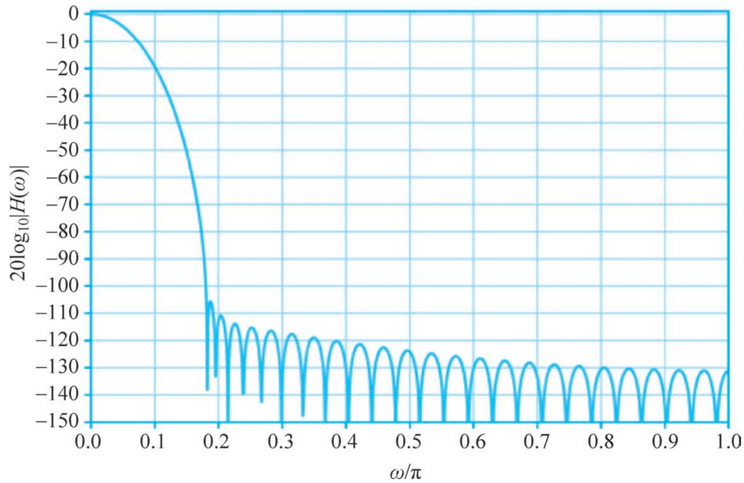


图 3.7.10 （续）

3.8 习题

1. 图 3.8.1 的序列 $\tilde{x}_1(n)$是具有周期为 4 的周期性序列。试确定傅里叶级数的系数 $\tilde{X}_1(k)$。

2. 在图 3.8.2 中表示了两个周期都为 6 的周期性序列，确定这两个序列的周期卷积的结果 $\tilde{x}_3(n)$，并画出草图。

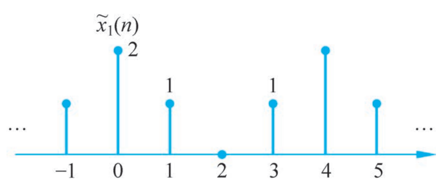


图 3.8.1 题 1 图

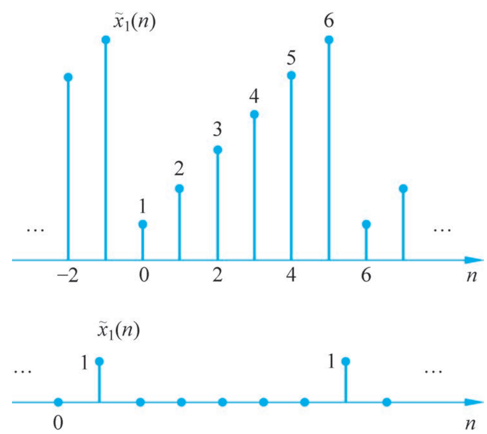


图 3.8.2 题 2 图

3. 在图 3.8.3 中表示了几个周期性序列 $x(n)$，这些序列可以按离散傅里叶级数表示为

$$\tilde{x}(n)=\frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}(k)\mathrm{e}^{\mathrm{j}\left(\frac{2\pi}{N}\right)kn}$$

(1) 对于哪些序列可以通过选择时间起点使得所有 $\tilde{X}(k)$是实数？

(2) 对于哪些序列可以通过选择时间起点使得除 $\tilde{X}(0)$外的所有 $\tilde{X}(k)$是虚数?

(3) 对于哪些序列能够做到 $\tilde{X}(k)=0, k=\pm 2, \pm 4, \cdots$?

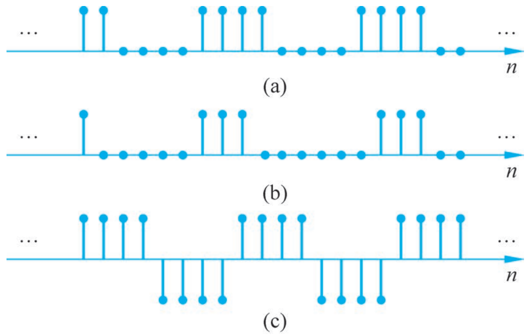


图 3.8.3　题 3 图

4. 试证明下面列出的周期序列离散傅里叶级数的对称特性。在证明中,可以利用离散傅里叶级数的定义及任何已证明的性质。例如在证明性质(3)时可以利用性质(1)和(2)。

序列	离散傅里叶级数
(1) $\tilde{x}^*(n)$	$\tilde{X}^*(-k)$
(2) $\tilde{x}^*(-n)$	$\tilde{X}^*(k)$
(3) $\mathrm{Re}[\tilde{x}(n)]$	$\tilde{X}_e(k)$
(4) $\mathrm{jIm}[\tilde{x}(n)]$	$\tilde{X}_o(k)$

根据以上证明的性质,证明对于实数周期序列 $\tilde{x}(n)$,离散傅里叶级数的下列对称特性成立:

(1) $\mathrm{Re}[\tilde{X}(k)]=\mathrm{Re}[\tilde{X}(-k)]$

(2) $\mathrm{jIm}[\tilde{X}(k)]=-\mathrm{Im}[\tilde{X}(k)]$

(3) $|\tilde{X}(k)|=|\tilde{X}(-k)|$

(4) $\arg[\tilde{X}(k)]=-\arg[\tilde{X}(-k)]$

5. 求下列序列的 DFT:

(1) $\{1 \quad 1 \quad -1 \quad -1\}$

(2) $\{1 \quad \mathrm{j} \quad -1 \quad -\mathrm{j}\}$

(3) $x(n)=cn, 0 \leqslant n \leqslant N-1$

(4) $x(n)=\sin\left(\dfrac{2\pi n}{N}\right), 0 \leqslant n \leqslant N-1$

6. 在图 3.8.4 中表示了有限长序列 $x(n)$,画出序列 $x_1(n)$和 $x_2(n)$的草图(注意: $x_1(n)$和 $x_2(n)$是 $x(n)$圆周移位的两个点)。

$$x_1(n)=x((n-2))_4 R_4(n)$$

$$x_2(n)=x((-n))_4 R_4(n)$$

7. 在图 3.8.5 中表示了两个有限长序列,试画出它们的 6 点圆周卷积。

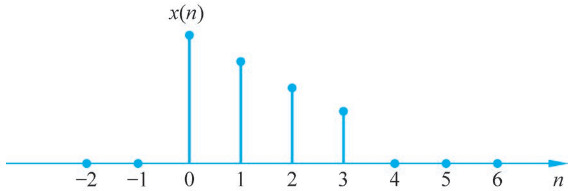


图 3.8.4　题 6 图

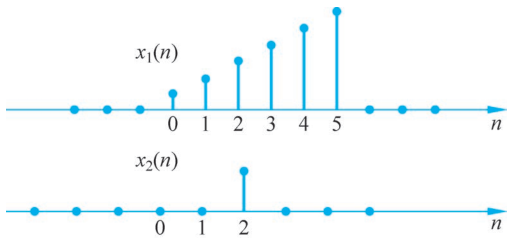


图 3.8.5　题 7 图

8. 有限长序列的 DFT 对应于序列在单位圆上的 z 变换的取样。例如一个 10 点序列 $x(n)$的 DFT 对应于图 3.8.6 表示的 10 个等间隔点上 $X(z)$的采样。希望找出在图 3.8.7 所示的围线上 $X(z)$的等间隔采样，即 $X(z)\Big|_{z=0.5e^{j\left[\left(\frac{2\pi k}{10}\right)+\left(\frac{\pi}{10}\right)\right]}}$。证明如何修改 $x(n)$以获得一个序列 $x_1(n)$致使 $x_1(n)$的 DFT 对应于所希望的 $X(z)$的采样。

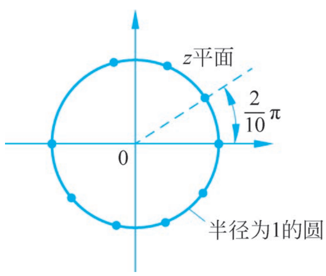


图 3.8.6　题 8 图 1

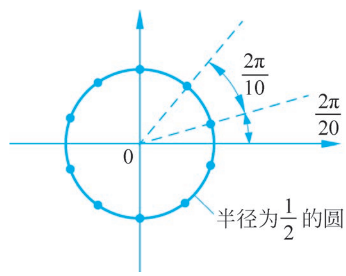


图 3.8.7　题 8 图 2

9. 列长为 8 的一个有限长序列具有 8 点离散傅里叶变换 $X(k)$，如图 3.8.8 所示。列长为 16 点的一个新的序列 $y(n)$定义为

$$y(n)=\begin{cases} x\left(\dfrac{n}{2}\right), & n \text{ 为偶数} \\ 0, & n \text{ 为奇数} \end{cases}$$

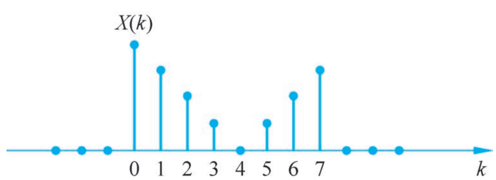


图 3.8.8　题 9 图 1

试从图 3.8.9 的几个图中选出相当于 $y(n)$的 16 点离散傅里叶变换序列图。

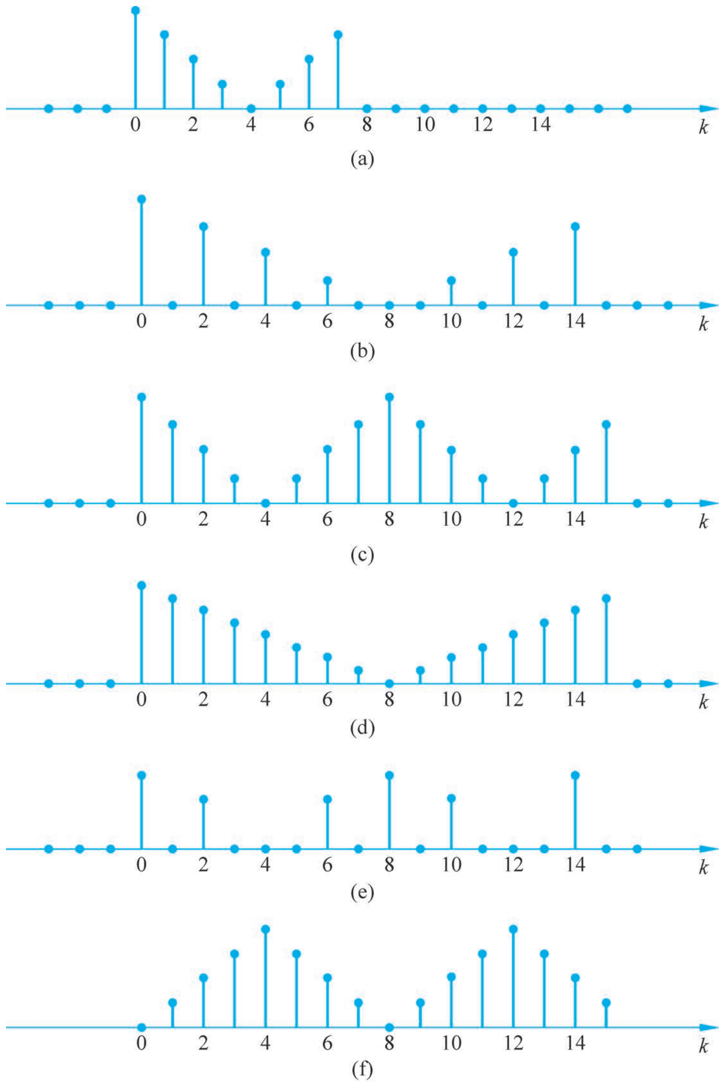


图 3.8.9 题 9 图 2

10. 图 3.8.10 表示一个 4 点序列 $x(n)$。

(1) 试绘出 $x(n)$同 $x(n)$线性卷积略图；

(2) 试绘出 $x(n)$同 $x(n)$4 点圆周卷积略图；

(3) 试绘出 $x(n)$同 $x(n)$10 点圆周卷积略图；

(4) 若 $x(n)$同 $x(n)$的某个 N 点圆周卷积与其线性卷积相同，试问此时 N 点的最小值是多少？

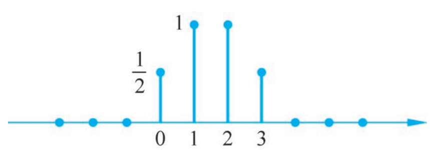


图 3.8.10 题 10 图