快速傅里叶变换(FFT)

实验涉及的MATLAB子函数

1.fft

功能： 一维快速傅里叶变换(FFT)。

调用格式：

y＝fft(x)；利用FFT算法计算矢量x的离散傅里叶变换，当x为矩阵时，y为矩阵x每一列的FFT。当x的长度为2的幂次方时，则fft函数采用基2的FFT算法，否则采用稍慢的混合基算法。

y＝fft(x，n)；采用n点FFT。当x的长度小于n时，fft函数在x的尾部补零，以构成n点数据；当x的长度大于n时，fft函数会截断序列x。当x为矩阵时，fft函数按类似的方式处理列长度。

2.ifft

功能： 一维快速傅里叶逆变换(IFFT)。

调用格式：

y＝ifft(x)；用于计算矢量x的IFFT。当x为矩阵时，计算所得的y为矩阵x中每一列的IFFT。

y＝ifft(x，n)；采用n点IFFT。当length(x)<n时，在x中补零；当length(x)>n时，将x截断，使length(x)＝n。

3.fftshift

功能： 对fft的输出进行重新排列，将零频分量移到频谱的中心。

调用格式：

y＝fftshift(x)；对fft的输出进行重新排列，将零频分量移到频谱的中心。当x为向量时，fftshift(x)直接将x中的左右两半交换而产生y。

当x为矩阵时，fftshift(x)同时将x的左右、上下进行交换而产生y。

三、实验原理

1.用MATLAB提供的子函数进行快速傅里叶变换

从理论学习可知，DFT是唯一在时域和频域均为离散序列的变换方法，它适用于有限长序列。尽管这种变换方法是可以用于数值计算的，但如果只是简单的按照定义进行数据处理，当序列长度很大时，则将占用很大的内存空间，运算时间将很长。

快速傅里叶变换是用于DFT运算的高效运算方法的统称，FFT只是其中的一种。FFT主要有时域抽取算法和频域抽取算法，基本思想是将一个长度为N的序列分解成多个短序列，如基2算法、基4算法等，大大缩短了运算的时间。

MATLAB中提供了进行快速傅里叶变换(FFT)的子函数，用fft计算DFT，用ifft计算IDFT。

例10-1 已知一个长度为8点的时域离散信号，n1＝0，n2＝7，在n0＝4前为0，n0以后为1。对其进行FFT变换，作时域信号及DFT、IDFT的图形。

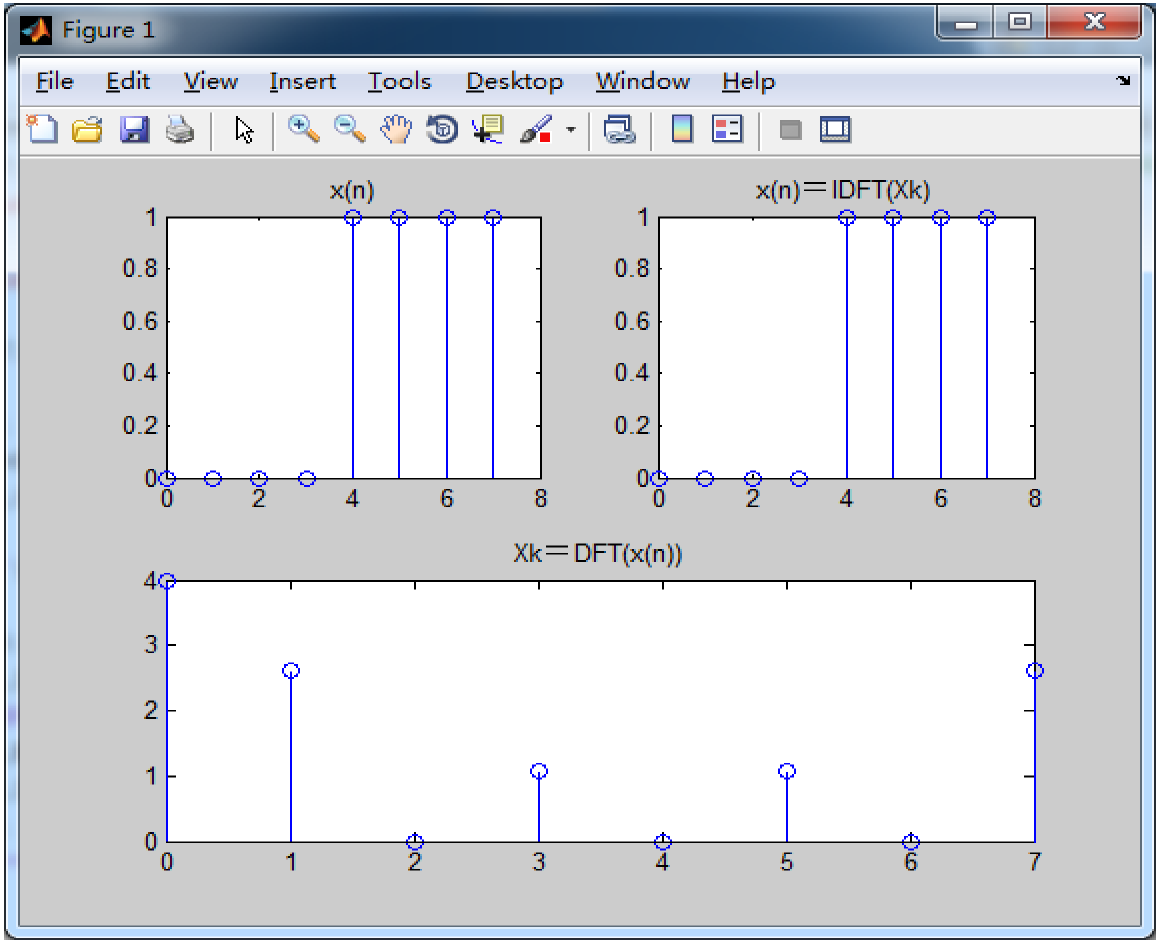
解 程序如下：

```
n1＝0；n2＝7；n0＝4；
n＝n1：n2；N＝length(n)；
xn＝［(n－n0)>＝0］；          %建立时域信号
subplot(2，2，1)；stem(n，xn)；
title('x(n)')；
```

```
k＝0：N－1；
Xk＝fft(xn，N)；%用FFT计算信号的DFT
subplot(2，1，2)；stem(k，abs(Xk))；
title(′Xk＝DFT(x(n))′)；
xn1＝ifft(Xk，N)；%用IFFT计算信号的IDFT
subplot(2，2，2)；stem(n，xn1)；
title(′x(n)＝IDFT(Xk)′)；
```

运行结果如图10-1所示。

图10-1 例10-1用FFT求有限长序列的傅里叶变换



例10-2 将例13-5已知的两个时域周期序列分别取主值，得到x1=［1，1，1，0，0，0］，x2=［0，1，2，3，0，0］，求时域循环卷积y(n)并用图形表示。

解 本例将例13-5使用DFT处理的计算，改为用FFT和IFFT进行循环卷积。

程序如下(作图程序部分省略)：

xn1=［0，1，2，3，0，0］；　　　%建立x1(n)序列

xn2=［1，1，1，0，0，0］；　　　%建立x2(n)序列

N=length(xn1)；

n＝0：N－1；k＝0：N－1；

Xk1＝fft(xn1，N)；%由x1(n)的FFT求X1(k)

Xk2＝fft(xn2，N)；%由x2(n)的FFT求

Yk＝Xk1.*Xk2；%Y(k)＝X1(k)X2(k)

yn＝ifft(Yk，N)；%由Y(k)的IFFT求y(n)

yn＝abs(yn)

运行结果如图10-5所示，与例10-5用DFT计算的结果一致。

2.用FFT计算有限长序列的频谱

1)基本概念

一个序号从n_1到n_2的时域有限长序列x(n)，它的频谱$X(e^{j\omega})$定义为它的离散傅里叶变换，且在奈奎斯特(Nyquist)频率范围内有界并连续。序列的长度为N，则N＝n2－n1＋1。计算x(n)的离散傅里叶变换(DFT)得到的是$X(e^{j\omega})$的N个样本点$X(e^{j\omega k})$。其中数字频率为

$$\omega_k = k(\frac{2\pi}{N}) = kd\omega$$

式中：dω为数字频率的分辨率；k取对应－(N－1)/2到(N－1)/2区间的整数。

在实际使用中，往往要求计算出信号以模拟频率为横坐标的频谱，此时对应的模拟频率为

$$\Omega_k = \omega_k / T_s = k\left(\frac{2\pi}{NT_s}\right) = k\left(\frac{2\pi}{L}\right) = kD$$

式中：D为模拟频率的分辨率或频率间隔；T_s为采样信号的周期，Ts＝1/Fs；定义信号时域长度L＝NT_s。

在使用FFT进行DFT的高效运算时，一般不直接用n从n1到n2的x(n)，而是取 $\tilde{x}(n)$ 的主值区间(n＝0，1，…，N－1)的数据，经FFT将产生N个数据，定位在k＝0，1，…，N－1的数字频率点上，即对应［0，2π］。如果要显示［−π，π］范围的频谱，则可以使用fftshift(X)进行位移。

2)频谱的显示及分辨率问题

例10-3 已知有限长序列x(n)＝［1，2，3，2，1］，其采样频率F_s＝10 Hz。请使用FFT计算其频谱。

解 MATLAB程序如下：

```
Fs＝10；
xn＝［1，2，3，2，1］；N＝length(xn)；
D＝2*pi*Fs/N；               %计算模拟频率分辨率
k＝floor(－(N－1)/2：(N－1)/2)；%频率显示范围对应［－π，π］
X＝fftshift(fft(xn，N))；%作FFT运算且移位π
```

subplot(1，2，1)；plot(k*D，abs(X)，′o：′)； %横轴化成模拟频率作幅度谱

title(′幅度频谱′)；xlabel(′rad/s′)；

subplot(1，2，2)；plot(k*D，angle(X)，′o：′)；%横轴化成模拟频率作相位谱

title(′相位频谱′)；xlabel(′rad/s′)；

程序运行结果：

absX＝

0.3820 2.6180 9.0000 2.6180 0.3820

angleX＝

－1.2566 2.5133 [KG*4/5]0 －2.5133 1.2566

运行结果如图10-2所示。

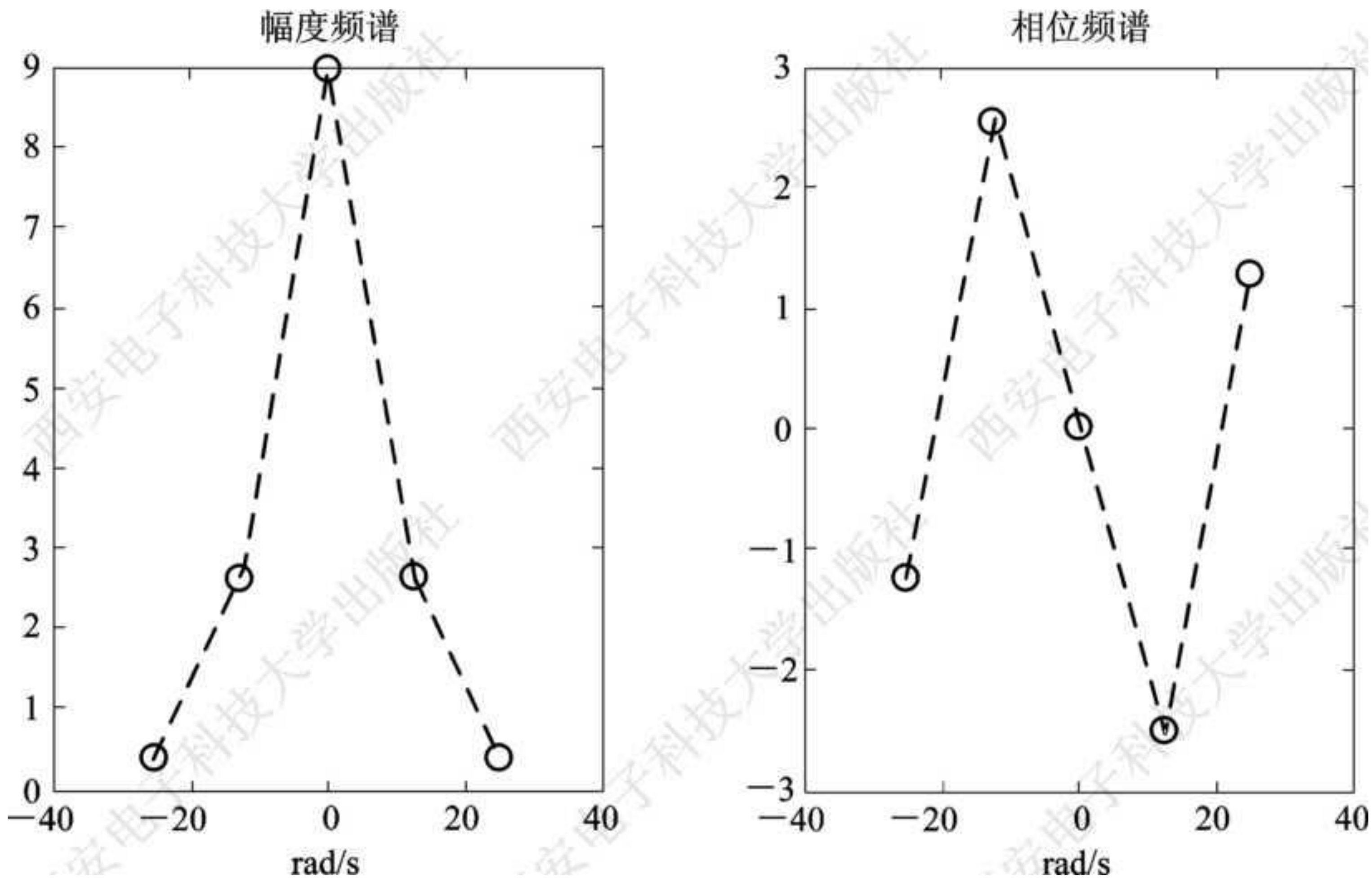


图10-2 例10-3有限长序列的频谱

由图10-2可知，当有限长序列的长度N＝5时，频谱的频率样本点数也为5，如图上用“　”表示的点位。频率点之间的间距非常大，即分辨率很低。即使使用了plot命令的插值功能，显示出的曲线仍是断断续续的，与真实曲线有较大的误差。

改变分辨率的基本方法是给输入序列补零，即增加频谱的密度。注意，这种方法只是改善了图形的视在分辨率，并不增加频谱的细节信息。

将上述有限长序列x(n)＝［1，2，3，2，1］末尾补0到N＝1000点，将程序改为：

Fs＝10；N＝1000；

xn＝［1，2，3，2，1］；Nx＝length(xn)；

xn＝［1，2，3，2，1，zeros(1，N－Nx－1)］；

D＝2*pi*Fs/N；　　　　%计算模拟频率分辨率

k＝floor(－(N－1)/2：(N－1)/2)；%频率显示范围对应［－π，π］

X＝fftshift(fft(xn，N))；%作FFT运算且移位π

subplot(1，2，1)；plot(k*D，abs(X))；%横轴化成模拟频率作幅度谱

title(′幅度频谱′)；xlabel(′rad/s′)；

subplot(1，2，2)；plot(k*D，angle(X))；%横轴化成模拟频率作相位谱

title(′相位频谱′)；xlabel(′rad/s′)；

此时程序执行的结果如图10-3所示。由图可以看出，图形的分辨率提高，曲线几乎是连续的频谱了。

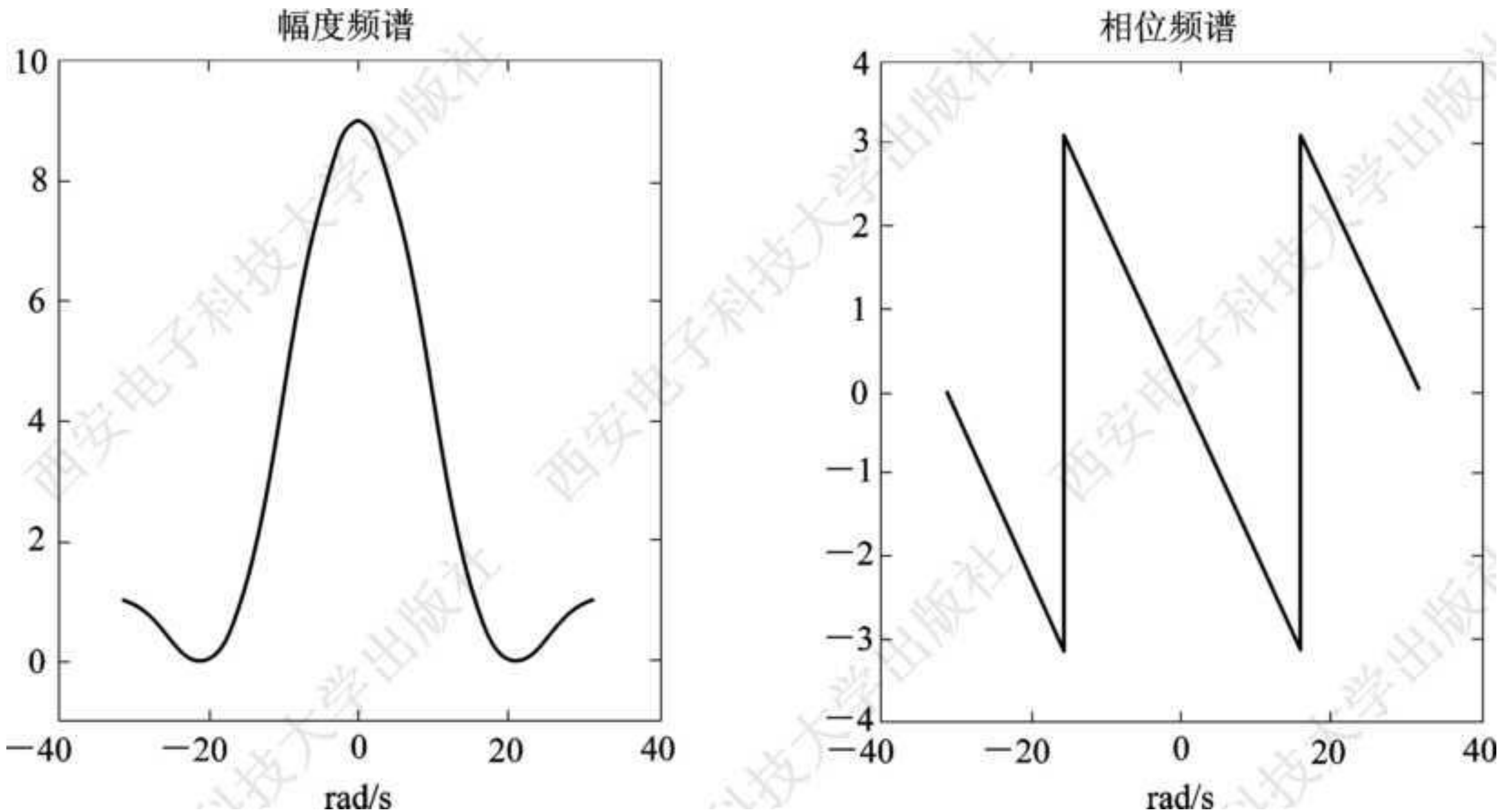


图10-3 将例10-2有限长序列末尾补0到N＝1000时的频谱

3)实偶序列如何补0

例10-4 已知一个矩形窗函数序列为

$$x(n)=\begin{cases}1 & |n|\le 5\\ 0 & |n|>5\end{cases}$$

采样周期T_s＝0.5 s，要求用FFT求其频谱。

解 由于该序列是一个实的偶序列，因而补0时需要仔细分析。假定按N＝32补0，则主值区域在n＝0～31，FFT的输入应为

$$Xn=[ones(1,6), zeros(1, N-11), ones(1,5)]$$

即原来n=[－5：－1]的前五个点移到n＝[27：31]中去了。

下面考虑分别用N=32，64，512，观察不同N值代入对频谱的影响。

程序如下，

Ts=0.5;C=[32,64,512]; %输入不同的N值

forr＝0：2；

N＝C(r＋1)；

xn＝［ones(1，6)，zeros(1，N－11)，ones(1，5)］；%建立x(n)

D＝2*pi/(N*Ts)；

k＝floor(－(N－1)/2：(N－1)/2)；

X＝fftshift(fft(xn，N))；

subplot(3，2，2*r＋1)；plot(k*D，abs(X))； %幅度频谱

subplot(3，2，2*r＋2)；stairs(k*D，angle(X))；%相位频谱

end

注意： 此处相位频谱使用了stairs，因为该相位频谱变化率比较陡峭。

程序执行结果如图10-4所示。

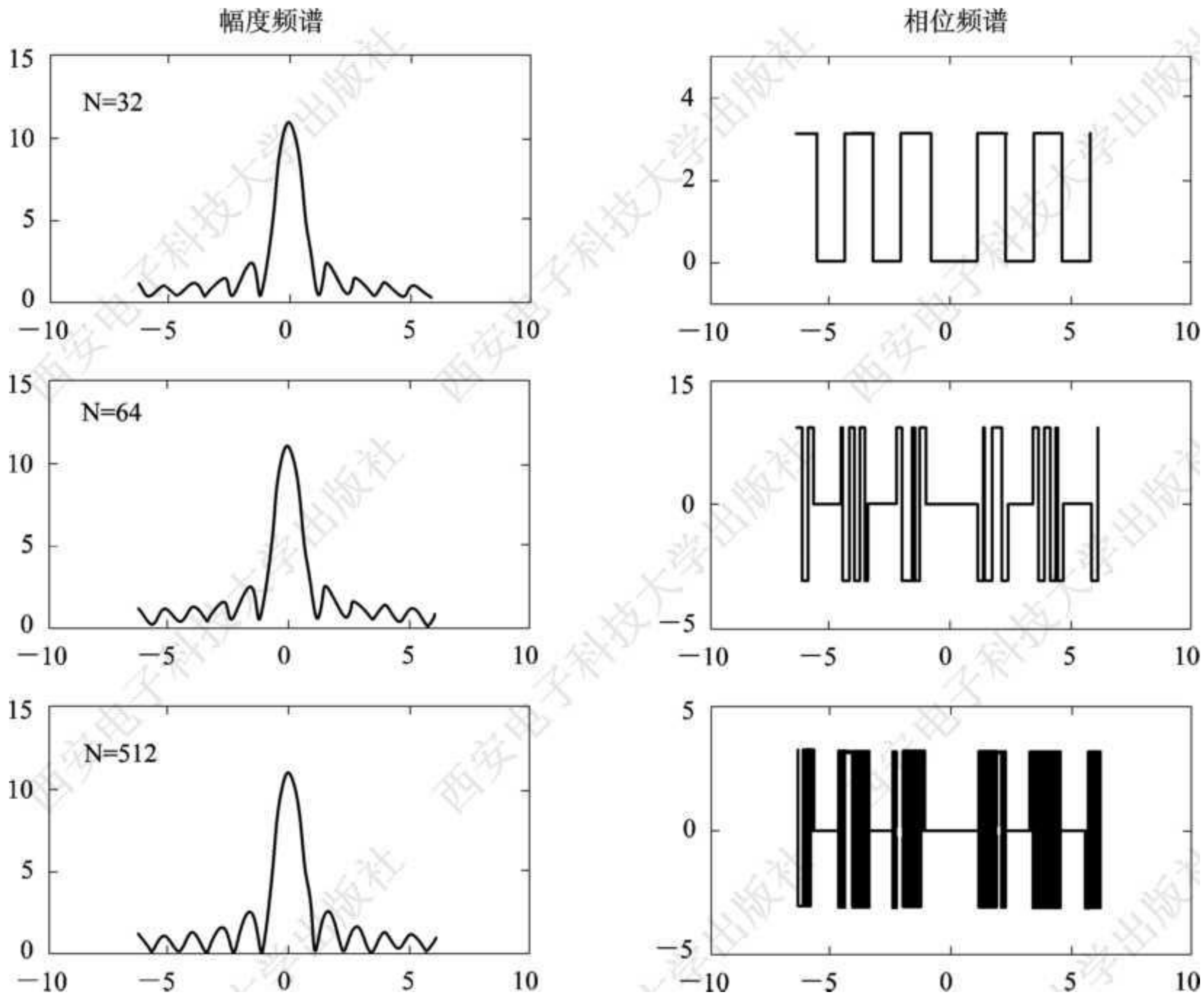


图10-4 将例10-4有限长序列补0到N＝32、64、512时的频谱

如果将x(n)的输入写成

xn＝［ones(1，11)，zeros(1，N－11)］；%建立x(n－5)相当于起点不是取自n＝0而是n＝－5，计算的是x(n－5)的频谱。幅度频谱不受影响，相位频谱引入一个线性相位－5ω，如图10-5所示。

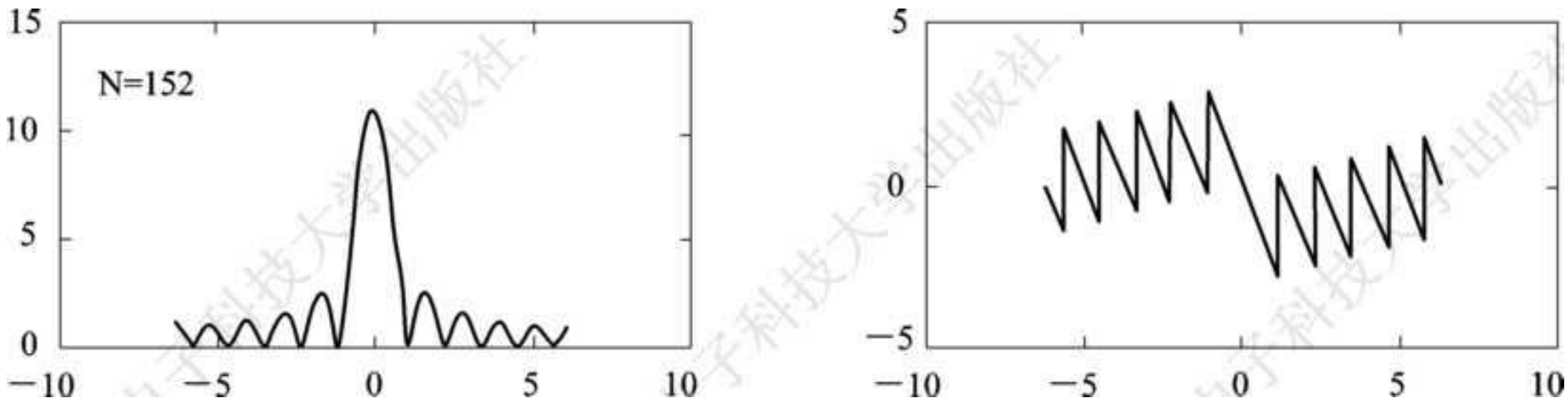


图10-5 将有限长位移序列x(n－5)补0到N＝512时的频谱

3.用FFT计算无限长序列的频谱

用FFT进行无限长序列的频谱计算，首先要将无限长序列截断成一个有限长序列。序列长度的取值对频谱有较大的影响，带来的问题是引起频谱的泄漏和波动。

例10-5 已知一个无限长序列为

$$x(n) = \begin{cases} e^{-0.5n} & n \geq 0 \\ 0 & n < 0 \end{cases}$$

采样频率F_s＝20 Hz，要求用FFT求其频谱。

解 MATLAB程序如下：

```
Fs＝20；C＝［8，16，128］；     %输入不同的N值
forr＝0：2；
N＝C(r＋1)；
n＝0：N－1；
xn＝exp(－0.5*n)；%建立x(n)
D＝2*pi*Fs/N；
k＝floor(－(N－1)/2：(N－1)/2)；
X＝fftshift(fft(xn，N))；
subplot(3，2，2*r＋1)；plot(k*D，abs(X))；
```

```
axis(［－80，80，0，3］)；
subplot(3，2，2*r＋2)；stairs(k*D，angle(X))；
axis(［－80，80，－1，1］)；
end
```

运行结果如图10-6所示。

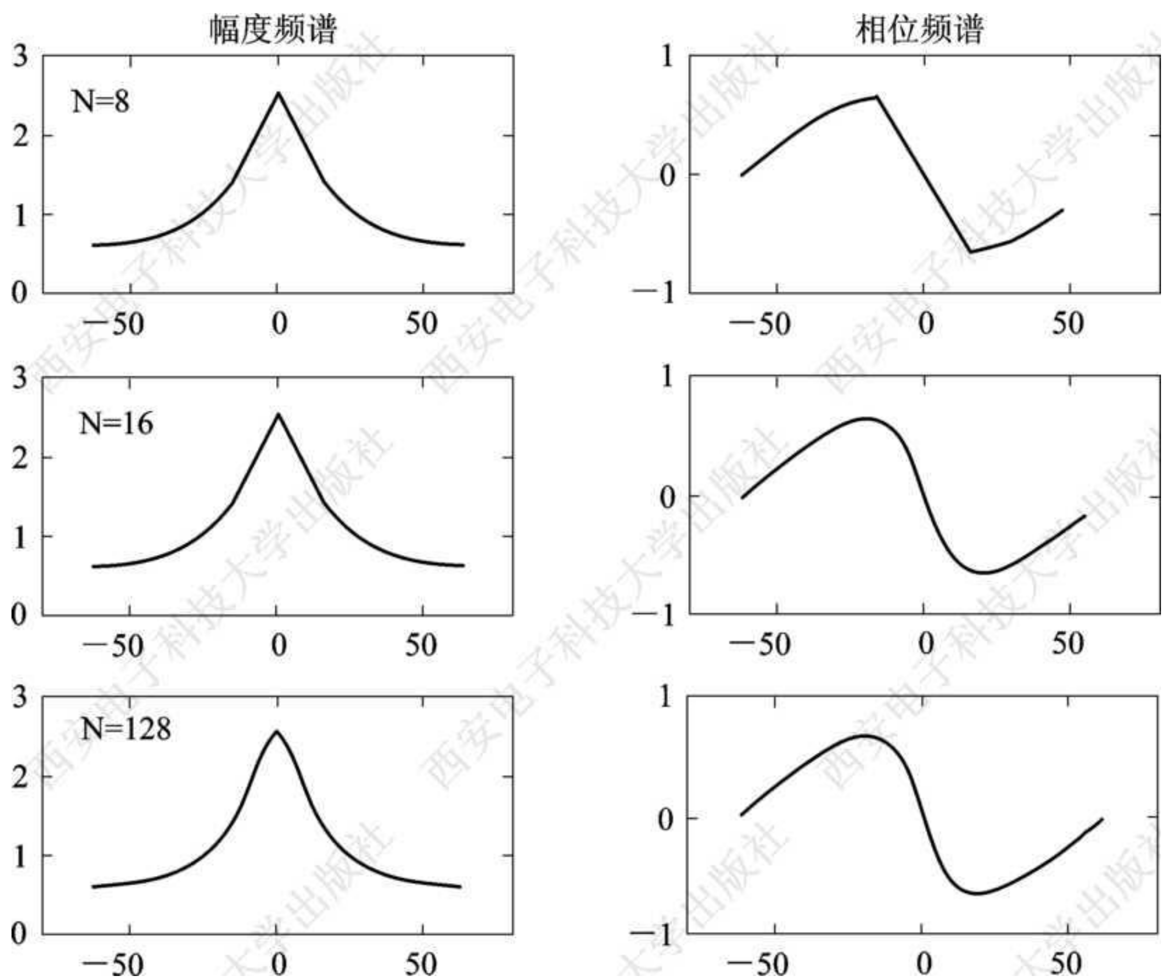


图10-6 将无限长序列截断为N＝8，16，128时的频谱

例10-6 用FFT计算下列连续时间信号的频谱，并观察选择不同的Ts和N值对频谱特性的影响。

$$x_a(t) = e - 0.01t(\sin 2t + \sin 2.1t + \sin 2.2t) \quad t \geq 0$$

解 该题选择了三个非常接近的正弦信号，为了将各频率成分区分出来，在满足奈奎斯特定理的条件下确定采样周期，选择三组数据，分别是T_s＝0.5 s、0.25 s和0.125 s；再确定N值，分别选择N＝256和2048。观察不同Ts和N的组合对频谱的影响。

程序如下：

T0＝［0.5，0.25，0.125，0.125］；　%输入不同的Ts值

N0＝［256，256，256，2048］；%输入不同的N值

forr＝1：4；

Ts＝T0(r)；N＝N0(r)；%赋Ts和N值

n＝0：N－1；

D＝2*pi/(Ts*N)；%计算模拟频率分辨率

xa＝[ZK(]exp(－0.01*n*Ts).*(sin(2*n*Ts)＋sin(2.1*n*Ts)＋sin(2.2*n*Ts))；

k＝floor(－(N－1)/2：(N－1)/2)；

```
Xa＝Ts*fftshift(fft(xa，N))；
［r，Xa(1)］%输出Xa(1)的数值，供误差计算用
subplot(2，2，r)；plot(k*D，abs(Xa)，′k′)；
axis(［1，3，1.1*min(abs(Xa))，1.1*max(abs(Xa))］)；
end
```

运行结果如图10-7所示。

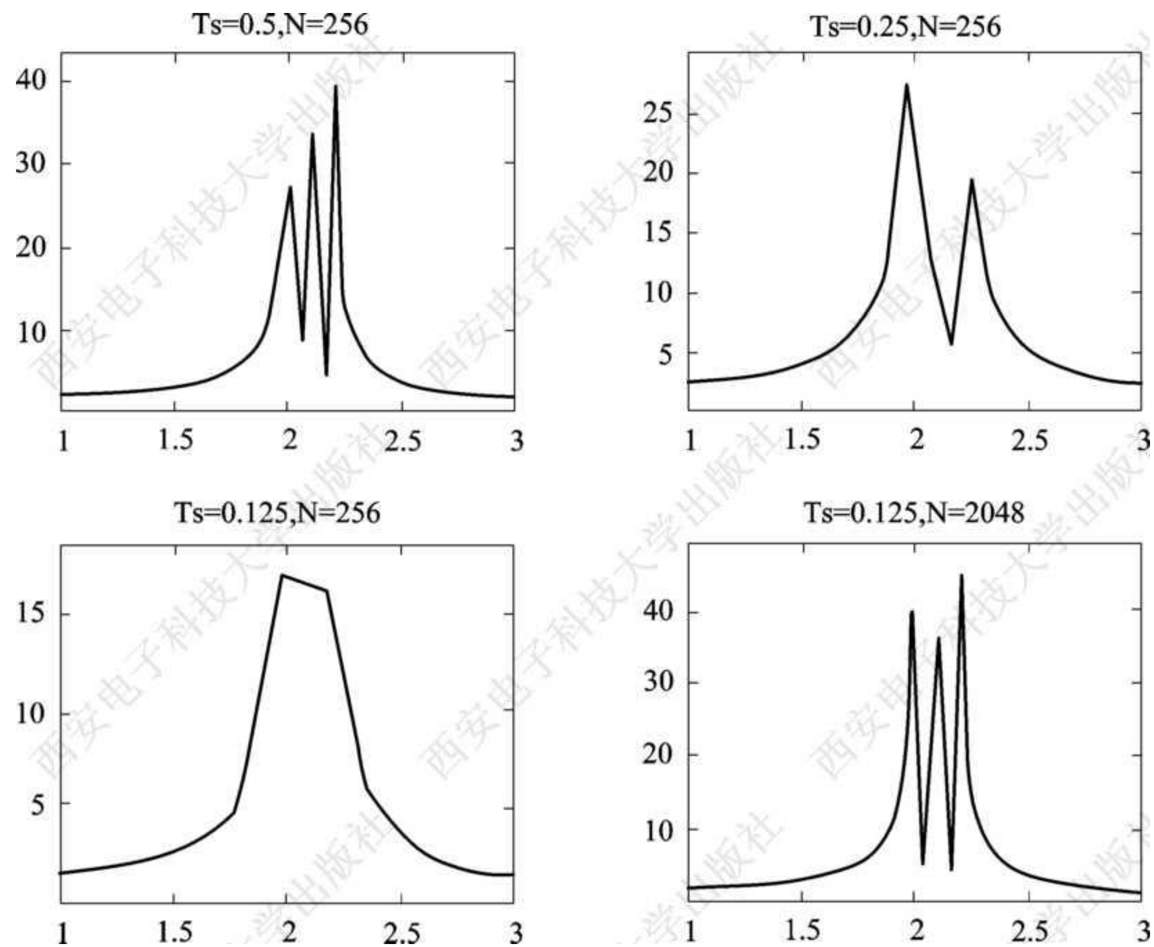


图10-7 用FFT计算三个很靠近的谐波分量的频谱图

由图10-7可以得出以下结论：

(1)N同样取256(如前三个图形)，当Ts越大时，时域信号的长度$L=NT_s$保留得越长，则分辨率越高，频谱特性误差越小；反之，则分辨率越低，频谱特性误差越大，甚至丢失某些信号分量。

(2)T_s相同(如后两个图形)，当N越大时，在［0，2π］范围内等间隔抽样点数越多，且时域信号的长度$L=NT_s$保留得越长，则分辨率越高，频谱特性误差越小；反之，当N越小时，在［0，2π］范围内等间隔抽样点数越少，则有可能漏掉某些重要的信号分量，称为栅栏效应。