实验三 用FFT进行谱分析

实验四 利用FFT 实现快速卷积

相关课件：DSP_14.ppt　DSP_15.ppt　DSP_16.ppt

参考资料：

本节所涉及的内容，很多都是第3章(DFT)的内容，用FFT法来解决。

1. FFT是DFT的快速计算方法，因而DFT的所有应用，用FFT都能快速完成。一维FFT的格式为

* **y=fft(x)**

它利用FFT计算矢量x的DFT，当x为矩阵时，y为矩阵x的每一列的FFT，fft采用混合基算法。当x的长度length(x)为2^L(L为正整数)时，fft得到基-2 FFT算法，否则就采用运行稍慢的混合基算法(将n分解为素数)，若n为素数，则fft采用的是原始的DFT算法。

* **y=fft(x,n)**，则采用n点FFT，当x的长度小于n时，fft在x的尾部补零，补成n点数据；当x的长度大于n时，fft就会截断序列x，使之长度为n点长，当x是矩阵时，fft按类似的方法处理列长度。

* fft是用机器语言而不是用MATLAB指令写成的(即不存在.m文件)，因而它的执行速度很快。

* **ifft(x);ifft(x,n)**是计算IDFT的，其特点与fft相同，因此从运行速度上考虑，最好选为$n=2^L$。

2. 计算DTFT的矩阵表示法。当给定$x(n)$及n向量，并将频率离散化，规定计算分辨率为$d\omega=\frac{2\pi}{K}$，抽样频率向量序号为k，则DTFT的公式

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

可写成矩阵表示形式

$$[X(e^{j\omega_1}),X(e^{j\omega_2}),\cdots,X(e^{j\omega_K})]=[x(n_1),x(n_2),\cdots,x(n_N)]\begin{bmatrix}e^{-j\omega_1 n_1} & e^{-j\omega_2 n_1} & \cdots & e^{-j\omega_K n_1}\\ e^{-j\omega_1 n_2} & e^{-j\omega_2 n_2} & \cdots & e^{-j\omega_K n_2}\\ \vdots & \vdots & \ddots & \vdots\\ e^{-j\omega_1 n_N} & e^{-j\omega_2 n_N} & \cdots & e^{-j\omega_K n_N}\end{bmatrix}$$

将频率向量表示为$\boldsymbol{\omega}=[\omega_1,\omega_2,\cdots,\omega_K]$，$\omega_k=kd\omega=k\cdot\frac{2\pi}{K}$，$k=1,2,\cdots,K$

序列位置向量表示为$n=[n1:nN]$，则式中矩阵指数可写成$-jn'*\omega=-jn'*kd\omega$($n'$表示位置向量的转置)

MATLAB 中用 w 代替 ω，用 n′代替转置 n^{τ}，则 DTFT 也可表示为（x 用行向量表示）

```
X = x * exp( - j * dw * n' * k)
```

若 x，X 皆为列向量，则将上式的矩阵进行转置，可得另一种形式的表示

```
X = exp( - j * dw * k' * n) * x
```

【例 4.4】 已知 $x(n)=[2,1,-1,2,3]$，用矩阵表示法求 $x(e^{j\omega})=\mathrm{DTFT}[x(n)]$及 $x(n)$的 5 点 DFT，并将DFT 的$(0,2\pi)$范围移到与 DTFT 的$[-\pi,\pi]$重叠。

解

```
% x(n) = [2,1, - 1,2,3];
% 用矩阵表示法求 x(ejw) = DTFT[x(n)]及 5 点 DFT 并将 DFT 的(0,2pi)范围移到与 DTFT 的[ - pi,pi]
% 相重叠
clc;clear all
x = [2,1, - 1,2,3];nx = 0:4;K = 128;dw = 2 * pi/K;
k = floor(( - K/2 + 0.5):(K/2 - 0.5));
X = x * exp( - j * dw * nx' * k);
subplot(221);plot(k * dw,abs(X));hold on;xlabel('\omega');ylabel('幅度响应');
title('5 点序列的 DTFT 和 FFT');
Xd = fft([2,1, - 1,2,3]);plot([0:4] * 2 * pi/5,abs(Xd),'.');grid;
Xd1 = fftshift(Xd);
subplot(222);plot(k * dw,abs(X));hold on;xlabel('\omega');ylabel('幅度响应');title('FFT 移位后');
plot([ - 2:2] * 2 * pi/5,abs(Xd1),'.');grid;
subplot(223);plot(k * dw,angle(X));hold on;xlabel('\omega');ylabel('相位响应');
title('FFT 移位后');grid;
```

程序运行结果如图 4.31 所示。

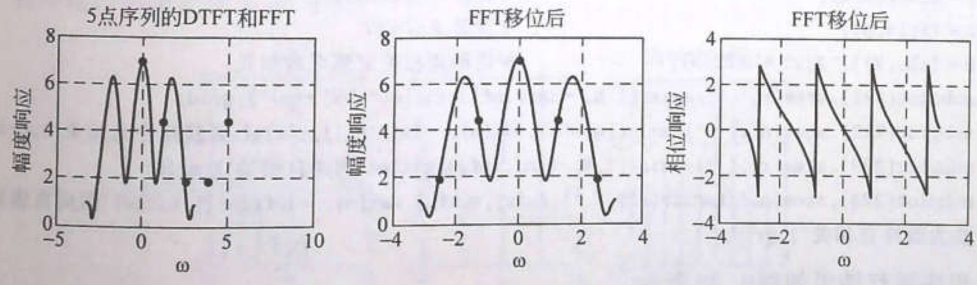

图 4.31　例 4.4 的图

4.11.1　用 FFT 计算有限长序列的线性卷积和线性相关

采用图 3.10 的框图，用 FFT 算法来求解两有限长序列的线性卷积及线性相关。

【例 4.5】 计算序列 $x(n)=[\underline{2},1,3,2,1,5,1]$与 $h(n)=[\underline{1},2,-1,-3]$的线性卷积。

解

```
clc;clear all
x = [2,1,3,2,1,5,1];h = [1,2, - 1, - 3];
N = length(x) + length(h) - 1;          % 卷积输出对应长度
n = 0:N - 1;
```

```
x = [x, zeros(1, N - length(x))];                         % 对 x 补零
h = [h, zeros(1, N - length(h))];                         % 对 h 补零
X = fft(x); H = fft(h);                                   % 求 x,h 的 FFT
Y = X. * H; y = ifft(Y);                                  % 两序列的 FFT 相乘并求 IFFT
subplot(221), stem(n, x, '.'); xlabel('n'); ylabel('x(n)'); title('序列 x(n)'); grid;
subplot(222), stem(n, h, '.'); xlabel('n'); ylabel('h(n)'); title('序列 h(n)'); grid;
subplot(223), stem(n, y, '.'); xlabel('n'); ylabel('y(n)'); title('序列 y(n)'); grid;
```

程序运行结果如图 4.32 所示。

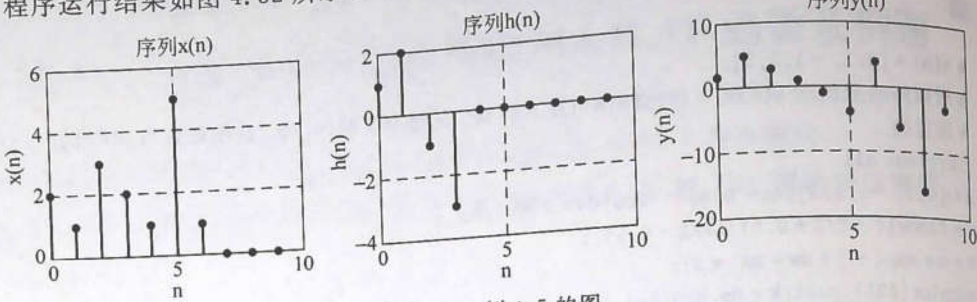

图 4.32 例 4.5 的图

【例 4.6】 计算 $x(n)=[\underline{2},1,3,2,1,5,1]$的线性自相关。

解

```
clc;clear all
x = [2,1,3,2,1,5,1];
N = 2 * length(x) - 1;                       % 自相关输出序列的长度
n = - length(x) + 1:length(x) - 1;           % 自相关序列有值范围
r = xcorr(x,x);                              % 直接计算自相关
X = fft(x,N);                                % 计算 N 点 FFT
R = (abs(X)).^2; r1 = ifft(R);               % 用圆周相关定理求自相关
subplot(221), stem(x,'.'); axis([1,N, - inf, inf]); title('序列 x(n)'); grid;
subplot(222), stem(n,r,'.'); axis([min(n),max(n), - inf, inf]); title('直接线性自相关'); grid;
subplot(223), stem(r1,'.'); axis([1,N, - inf, inf]); title('圆周自相关'); grid;
subplot(224), stem(n,fftshift(r1),'.'); axis([min(n),max(n), - inf, inf]); title('圆周自相关转
换为线性自相关'); grid;
```

程序运行结果如图 4.33 所示。

【例 4.7】 直接计算 $x(n)=[\underline{2},1,3,2,1,5,1]$与 $y(n)=[\underline{2},1,3,4]$的线性互相关 $r_{xy}(m)$，并利用 FFT 求此线性互相关。

解

```
clc;clear all
x = [2,1,3,2,1,5,1];                         % 输入序列
y = [2,1,3,4];                               % 输入序列
N = length(x) + length(y) - 1;               % 线性相关输出序列的长度
n = - length(y) + 1:length(x) - 1;           % 线性相关输出的位置向量
[rxy,mm] = xcorr(x,y);                       % 有位置向量的线性相关 rxy
X = fft(x,N); Y = fft(y,N);                  % 计算 x,y 的 FFT
Rxy1 = X. * conj(Y); rxy1 = ifft(Rxy1);      % 利用圆周相关定理计算相关
```

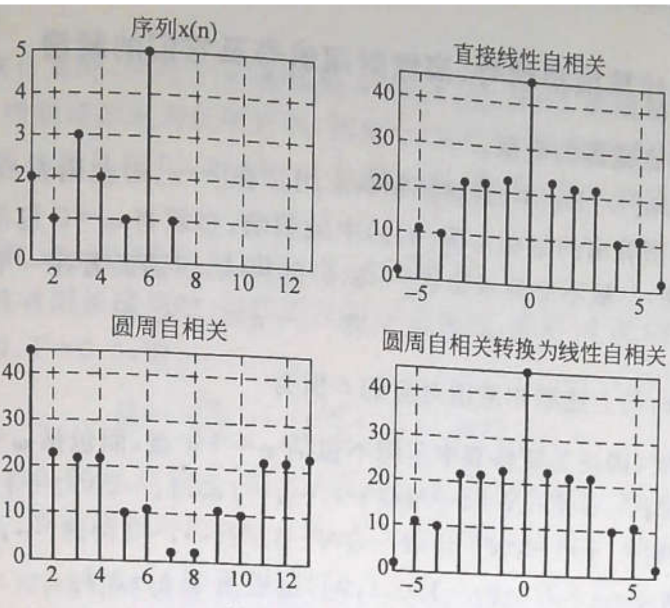

图 4.33　例 4.6 的图

```
subplot(231),stem(0:length(x) - 1,x,'.');grid;title('x(n)');
xlabel('n');
subplot(232),stem(0:length(y) - 1,y,'.');grid;title('y(n)');xlabel('n');
subplot(233),stem(mm,rxy,'.');grid;title('直接线性相关');xlabel('n');
axis([ - length(y) + 1,length(x) - 1, - inf,inf]);
subplot(234),stem(0:N - 1,rxy1,'.');grid;title('圆周相关');xlabel('n');
subplot(235),stem(n,[rxy1(length(x) + 1:N),rxy1(1:length(x))],'.');
grid;title('圆周相关转换为线性相关');axis([min(n),max(n), - inf,inf]);xlabel('n');
```

程序运行结果如图 4.34 所示。

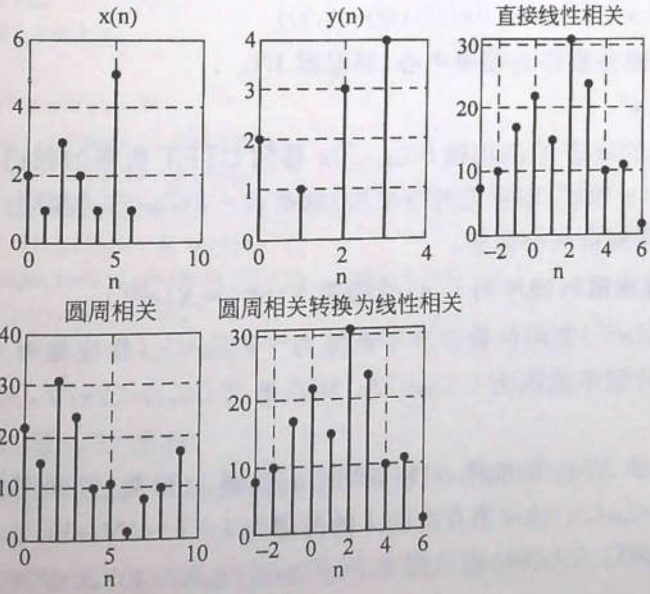

图 4.34　例 4.7 的图

4.11.2 用 FFT 计算模拟信号、离散时间信号及它们的频谱

1. 频率向量存在范围的表示。

如上面的讨论，在 DTFT 中，常将频率响应用 ω 在$[-\pi,\pi)$范围表示出来。而 DFT(FFT)则算出的是在 ω 为$[0,\pi)$中的频响，必须将 $\omega=0$ 包括在内，因而要去掉 $\omega=\pi$（考虑到对称性，一般不必显示 ω 在$[-\pi,0)$的频响）。因而若在一个周期中表示，则为

$$0\leqslant\omega<2\pi \quad 或 \quad -\pi\leqslant\omega<\pi$$

频域离散化 $\omega=\frac{2\pi}{N}k$，与上述频率范围对应的 k 值为

当 $-\pi\leqslant\omega<\pi$ 时（即奈奎斯特频率范围不包括 $\omega=+\pi$ 点，而包括 $\omega=-\pi$ 点）

N 为偶数时，$k=-N/2,-N/2+1,\cdots,-1,0,1,\cdots,N/2-1$

N 为奇数时，$k=-(N-1)/2,-(N-3)/2,\cdots,-1,0,1,\cdots,(N-1)/2$

例如，$N=6$，则 $k=[-3,-2,-1,0,1,2]$，这里由于 $k=3$ 与 $k=-3$ 处于同一位置，$\omega=\pm\pi$（由 z 平面单位圆上看 $\omega=\pm\pi$ 在同一位置），故将 $k=3$ 归于下一个周期内。

$N=7$，则 $k=[-3,-2,-1,0,1,2,3]$，这里由于 $k=3$ 及 $k=-3$ 都在 $\omega=\pm\pi$ 之内，所有抽样点都满足 $-\pi\leqslant\omega<\pi$。

* **在 MATLAB 语句中，将 N 为奇数与 N 为偶数两部分综合起来，可用以下语句表示 k 的范围**

* 对于 $-\pi\leqslant\omega<\pi$ 有 `k = floor((-N/2 + 0.5):(N/2 - 0.5))`

* 对于 $0\leqslant\omega<2\pi$ 有 `k = floor(0:(N - 0.5)) = [0:N - 1]`

其中 floor(x)表示 x 向下（向 $-\infty$ 方向）取整。

如果只取正半部分奈奎斯特频率，则有

* 对于 $0\leqslant\omega<\pi$ 有 `k = floor(0:(N/2 - 0.5))`

2. 将频谱的零频分量移到频谱中心，即重排 fft。

利用 fftshift(X)

它将 FFT 求出的频谱 X 的横轴 $0\leqslant\omega<2\pi$ 移到 DTFT 所需的频谱横轴 $-\pi\leqslant\omega<\pi$ 上。即将 ω 轴上 $\pi\leqslant\omega<2\pi$ 频段（即频谱的后半段）翻褶到 $-\pi\leqslant\omega<0$ 频段上去，当然同时要将位置向量的后半段也要翻褶到前面去。

3. 用 FFT 计算离散时间序列 $x(n)$ 的频谱 $X(e^{j\omega})=X(e^{j\Omega T})$。

(1) $X(e^{j\omega})=X(e^{j\Omega T})$ 常用的数字频率范围为 $-\pi\leqslant\omega<\pi$，相应地有 $-\pi/T\leqslant\Omega<\pi/T$，而 DFT 即 $X(k)$ 表示的频率范围为 $0\leqslant\omega<2\pi$，相应地有 $0\leqslant\Omega<2\pi/T$，其抽样值 k 的范围为 $0\leqslant k\leqslant N-1$。

若要用 $X(k)$ 表示 $X(e^{j\omega})$ 的抽样，可将频率中心移到 $\omega=0$ 处，因而用到 fftshift 语句。移位后其频率范围为 $-\pi\leqslant\omega<\pi$，抽样值存在的 k 的范围为 $k=\lfloor-(N-1)/2,(N-1)/2\rfloor$（$\lfloor\ \rfloor$ 表示向下取整），用 MATLAB 语句表示则为 $k=\text{floor}(-(N-1)/2,(N-1)/2)$（floor 表示向 $-\infty$ 方向取整）。

DFT 的抽样频率点为 $\omega_k=k(2\pi/N)=k\mathrm{d}\omega$，其中 $\mathrm{d}\omega=2\pi/N$，相应地有 $\Omega_k=\omega_k/T=k(2\pi/(NT))=kD$，其中 $D=2\pi/(NT)=(2\pi/N)f_s$，$f_s=1/T$ 为抽样频率，Ω_k 即为用角频率

表示的频率离散值。

在应用 **FFT 来计算时**，序列 $x(n)$ **必须是** $n=0,1,\cdots,N-1$ **的因果序列**，若不是因果序列，则必须用 N 点周期延拓成周期序列后，再取 $0\leqslant n\leqslant N-1$ 范围中的主值序列作为输入序列。**输出** $X(k)$ **也必然是在** $0\leqslant k\leqslant N-1$ **范围的值，若要处于对称的频率范围，则需采用 fftshift(X)。**

(2) 有限长序列 $x(n)$ **的频谱** $X(e^{j\Omega})=X(e^{j\omega/T})$ **计算。**

当用 FFT 计算有限长序列时，当序列长度 N 很短时，若用 N 点 FFT，则频域抽样也是 N 点，例如若 $N=9$，$T=0.4$，则

$$D=\frac{2\pi}{NT}=\frac{2\pi}{9\times 0.4}=1.7453\ \text{rad/s}$$

相邻两个角频率样本的间距 D 很大，若不用插值办法，很难恢复出 $X(e^{j\omega})$，为了提高计算分辨率，需减小 D 值。考虑到序列是有限长的，故只能采取将序列 $x(n)$ 补零值的办法使序列长度成为 L 点，$L>N$，这样可以使 $D_1=\dfrac{2\pi}{LT}<D=\dfrac{2\pi}{NT}$，提高了计算的分辨率。只要 L 选得合适就可较好地还原 $X(e^{j\omega})$。

【例 4.8】 已知有限长序列 $x(n)=\begin{cases}1, & -4\leqslant n\leqslant 4\\ 0, & \text{其他 } n\end{cases}$，这里 $x(n)$ 的长度为 $N=9$，求 $x(n)$ 的频谱(DTFT)。

解 若需作 L 点 DFT，则需将 $x(n)$ 补零值补到为 L 点长，然后作 L 点周期延拓，再取主值，然后对它作 DFT(FFT)。补零值越多，则计算的频率分辨率越高，也就是频谱上的取样点越多。这里选取抽样间隔为 $T=0.4$，取补零后 L 的长度分别为 9,16,32,512 来看对原频谱的逼近情况。

```
clc;clear;close all
c = [9,16,32,512];T = 0.4;
for i = 1:4
    L = c(i);D = 2 * pi/(L * T);
    x = [ones(1,5),zeros(1,L-9),ones(1,4)];
    k = floor( - (L-1)/2:(L-1)/2);
    X = fftshift(fft(x,L));
    subplot(2,2,i),plot(k * D,real(X));
    xlabel('\omega(rad)');ylabel('X(e^j^\omega)');grid;
    axis([min(k * D),max(k * D), -5,10]);
    Str = ['N= ',num2str(L)];title(Str);
end
```

程序运行结果如图 4.35 所示。

注：① 由于输入序列是实偶序列，故其 DFT 也为实偶序列，则 real(X)(实部)就是频率特性。

② plot 将输出序列顶点连线画出连续曲线即 $X(e^{j\Omega t})$，L 越大，则越趋近原频谱。$L=9,16$ 显然计算的分辨率不满足要求。

③ num2str(N) 表示变数值为字符串作为图的标注。

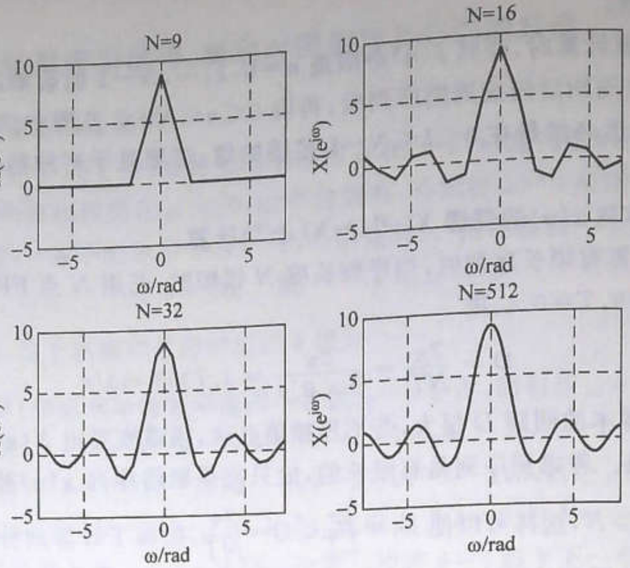

图 4.35　例 4.8 的图

(3) 无限长序列 $x(n)$ 的频谱 $X(e^{j\Omega})=X(e^{j\omega/T})$ 计算。

一个无限长序列用 FFT 求其频谱时，总是要将其截断成有限长序列，可以截断成任意长度，故不存在补零的问题。但是截断的结果会产生波纹以及泄漏。故只能逼近无限长序列的频谱，截断得越长，则逼近误差越小。

如何确定逼近误差，由于不知道精确的 $X(e^{j\omega})$ 是什么，因而无法和它相比较，只能用不断增加截断长度后，在相同的模拟角频率 Ω_i 上比较两次计算结果的误差，使得最大的绝对误差与最大的频谱绝对值的比值，即相对误差满足某个要求即可；不满足，则再增加截断长度。

为了方便比较，必须使每次截断长度等于前一次长度的两倍，这样，当抽样周期 T 选定后(使信号不产生混叠失真，这一要求在后面其他例子中讨论)便于比较。

当用 N_1 点长算出的频谱为 $X_1(j\Omega_1)$，其各频点为

$$\Omega_1(k_1)=k_1 2\pi/(N_1 T),\quad k_1=0,1,\cdots,N_1-1$$

当用 $N_2=2N_1$ 点长算出的频谱为 $X_2(j\Omega_2)$，其各频点为

$$\Omega_2(k_2)=k_2 2\pi/(N_2 T)=k_2 2\pi/(2N_1 T),\quad k_2=0,1,\cdots,2N_1-1$$

要求

$$\Omega_2(k_2)=\Omega_1(k_1)$$

则可解出 $k_2=2k_1$，即在 $\Omega_2(2k_1)=\Omega_1(k_1)$ 的这些点上比较 X_1 和 X_2 的幅度，从而求出相对误差的最大值，令其小于某一百分数，例如小于 1%。

由于幅频响应是偶对称函数，故只需在 $0\leqslant k\leqslant N_1/2-1$ 范围比较即可，编程如下。

【例 4.9】 $x(n)=(0.5)^n u(n)$，求此无限长序列的频谱。若需时域加倍长截断前后，同一频率处频谱的最大相对误差不大于 0.5%，试求截断长度为多少，画出其频谱。设抽样间隔为 $T=0.4$。

解

```
clc;clear
T = 0.4;r = 1;beta = 5e - 3;b = 0.01
```

```
while b>beta
    N1=2^r;n1=0:N1-1;                          % 判断结束循环的条件数据长度
    x1=0.5.^n1;X1=fft(x1);                     % 长度为 N1 的序列极其 FFT
    N2=2*N1;n2=0:N2-1;                         % 数据长度加倍为 N2
    x2=0.5.^n2;X2=fft(x2);                     % 长度为 N2 的序列极其 FFT
    k1=0:N1/2-1;k2=2*k1;                       % 确定两序列同一角频率的下标
    d=max(abs(X1(k1+1)-X2(k2+1)));             % 对应的同一频率点上的 FFT 的误差的最大绝对值
    X1m=max(abs(X1(k1+1)));                    % X1 幅度的最大值
    b=d/X1m;                                   % 最大相对误差的百分数
    r=r+1                                      % 序列长度加倍
end
k=floor(-(N2-1)/2:(N2-1)/2);D=2*pi/(N2*T);  % 那奎斯特频率范围,角频率间隔
subplot(121),plot(k*D,abs(fftshift(X2)));title('幅度谱');
xlabel('模拟角频率(rad/s)');ylabel('幅度');grid
subplot(122),plot(k*D,angle(fftshift(X2)));title('相位谱');
xlabel('模拟角频率(rad/s)');ylabel('相角');grid

N2=16
```

程序运行结果如图 4.36 所示。

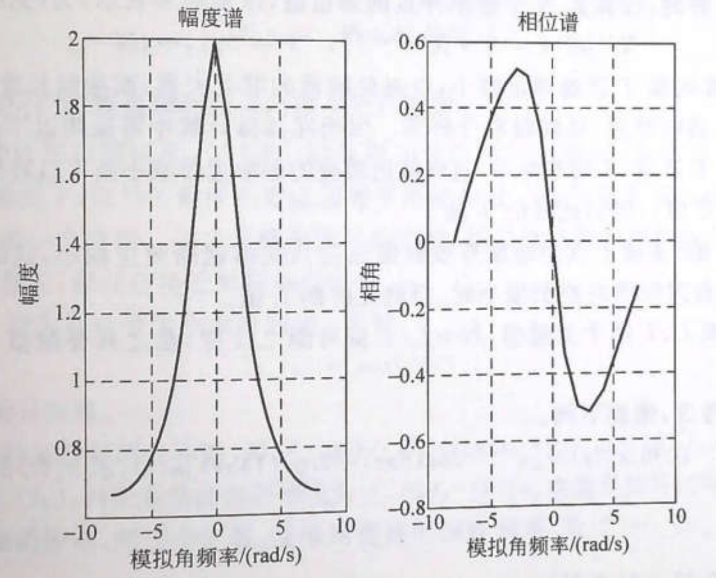

图 4.36　例 4.9 的图

4. 用 FFT 计算连续时间非周期信号 $x_a(t)$ 的频谱 $X_a(\mathrm{j}\Omega)$。

利用 FFT 计算连续时间非周期信号的频谱 $X(\mathrm{j}\Omega)$。将(3.6.12)式重写如下

$$X_a(\mathrm{j}\Omega)=\int_{-\infty}^{\infty}x_a(t)\mathrm{e}^{-\mathrm{j}\Omega t}\,\mathrm{d}t$$

用 DFT(FFT)计算时,得到的频谱公式可见(3.6.17)式,重写如下

$$X_a(\mathrm{j}k\Omega_0)=X_a(\mathrm{j}\Omega)\mid_{\Omega=k\Omega_0}=T\cdot\mathrm{DFT}[x(n)],\quad |\Omega|\leqslant\pi/T$$

T 为抽样间隔,$T=1/f_s$,f_s 为抽样频率,当频域抽样足够密时,则 $T\cdot\mathrm{DFT}[x(n)]$ 的包络足以逼近 $X(\mathrm{j}\Omega)$。

按 3.6.5 节对模拟信号用 DFT(FFT)作谱分析时参数选择的讨论,可知

(1) 有效数据的截断长度 T_0 的选择。 T_0 首先是影响频率分辨率 F_0，一般来说，$F_0=\frac{1}{T_0}$；反映在 DFT 上，则用截段长度为 $L=NT$ 表示 T_0，其中 T 为抽样间隔，N 为抽样点数。若频率分辨率用角频率 Ω_0 表示，且用 D 代替 Ω_0，则第 k 个抽样的角频率为

$$\Omega_k = k\frac{2\pi}{NT} = kD,\quad D = 2\pi/(NT) = 2\pi/L,\quad L = NT$$

截断的另一个影响是引起吉布斯效应，即产生波纹和肩峰，产生频谱的泄漏，同时也会产生一定的频谱混叠。

(2) 抽样间隔 $T\left(\text{即抽样频率 } f_s=\frac{1}{T}\right)$ 的选择。 T 要足够小使得频率混叠尽量小。例如使占信号绝大部分能量[(98～99)%]的最高频率 f_h 满足

$$f_h < f_s/2$$

或判断在 $\Omega=\pm\pi/T$ 处的频谱分量与最大频谱分量的比值小于某一百分数，例如小于1%。

(3) 抽样点数 N。

用 FFT 计算时，计算其 N 个频率样点的频谱值(以角频率表示)为(见(3.6.17)式)

$$X_a(\mathrm{j}kD) = T\cdot X(\mathrm{e}^{\mathrm{j}2\pi k/N}),\quad k=0,1,\cdots,N-1$$

综上 抽样间隔 T 要选择足够小，以避免频谱的混叠失真，而截断长度 $L=NT$ 要选择足够大，以减小截断效应，提高频率分辨率。因而在**具体计算中可采用以下步骤**：

① 先选定 L 不变，T 逐步减小，直到频谱混叠的影响逐步减小到可以容忍时(研究 $-\pi/T$ 或 π/T 处的频谱值)，得到此时的 T 值。

② 增加 L 值(实质上就是增加有效数据长度)，使得截断效应减小，这时采用上面已求的最小 T 值。直到频谱的差别很小时，得到此时的 L 值。

所以只需赋 L、T 两个变量值，$N=L/T$ 就可随之改变，总之只要赋值 N、L、T 中两个变量即可。

基于以上考虑，编程下例。

【例 4.10】 已知 $x_a(t)=[\mathrm{e}^{-0.02t}\cos(6\pi t)+2\cos(14\pi t)]u(t)$，求 $x_a(t)$ 的频谱。

解

```
clear,clc
T0 = [0.05,0.02,0.01,0.01];                            % 4 种抽样间隔
L0 = [10,10,10,20];                                    % 4 种信号记录长度,N = L0(i)/T0(i)
for i = 1:4
    T = T0(i);N = L0(i)/T0(i);                         % 按顺序选用 T 和 L
    D = 2 * pi/(N * T);                                % 频率分辨率
    n = 0:N - 1;x = exp( - 0.02 * n * T). * cos(6 * pi * n * T) + 2 * cos(14 * pi * n * T);   % 序列
    k = floor( - (N - 1)/2:(N - 1)/2);                 % 奈奎斯特下标向量
    X = T * fftshift(fft(x));                          % 求 x 的 FFT 并移到对称位置
    [i,X(i)]                                           % 检查四次循环在奈奎斯特频率处的幅度
    subplot(2,2,i),plot(k * D,abs(X));
    xlabel('模拟角频率(rad/s)');ylabel('幅度');grid
    axis([min(k * D),max(k * D),0,inf]);               % 坐标范围
    str = ['T = ',num2str(T),' N = ',num2str(N)];title(str);  % 标题显示抽样间隔及 FFT 点数 N
end
```

程序运行结果如图 4.37 所示。

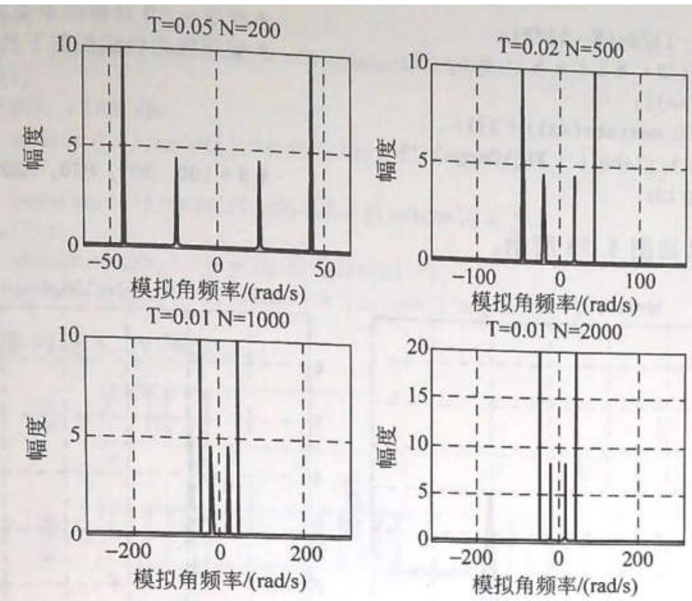

图 4.37　例 4.10 的图

5. 用 FFT(DFT)计算连续时间周期信号 $\tilde{x}(t)$ 频谱。

在例 3.17 中已用实例证明，用 DFT 研究周期性信号的频谱时，在满足 $f_0=f_s/N$(或 $f_0=Mf_s/N$)的情况下，信号的截断长度必须等于周期信号的整数倍周期长度，且至少应为抽样后周期序列的一个周期。如很难确定信号的周期，则只能作为非周期性信号处理，但是要使截断长度尽量长，以便更接近周期的整数倍。

【例 4.11】　设周期信号为全部时间轴 t 上的

$$x_a(t)=\cos(10t)$$

试用 DFT 法分析其频谱。

解　此函数的频谱是两个冲激，即 $X_a(\mathrm{j}\Omega)=\pi[\delta(\Omega-10)+\delta(\Omega+10)]$，$x_a(t)$的最高信号频率为 $f_h=10/(2\pi)$，因而最低抽样频率应为 $f_s\geqslant 2f_h=10/\pi$，也就是抽样时间间隔 $T=1/f_s\leqslant\pi/10$，由于截断的影响，频谱会扩散，故采用更高的抽样频率，取 $T=0.05$。由于 $x_a(t)$抽样后的 $x(n)$为

$$x(n)=\cos(10nT)=\cos(0.5n)=\cos(\omega_0 n)$$

则 $2\pi/\omega_0$ 不能用有理数$\left(\dfrac{N}{k}\right)$($N,k$ 为互素的整数)表示，故抽样后序列不是周期性的序列，因而截断长度不可能恰好是周期的整数倍，所以不可能得到准确的频谱峰值，只能取较大的截断值 $L=NT$，即使抽样值 N 更大一些，数据更多一些，使频谱更接近真实情况。以下用 N 作为自选参数，取 $N=100,200,800,1200$ 来计算。

MATLAB 程序代码为

```
clc;clear;close all
N = input('N = ');T = 0.05;n = 1:N;          % 原始数据
D = 2 * pi/(N * T);                          % 计算分辨率
xa = cos(10 * n * T);                        % 有限长余弦序列
Xa = T * fftshift(fft(xa,N));Xa(1)           % 求 x(n)的 DFT,移到对称位置
```

```
k = floor( - (N - 1)/2:(N - 1)/2);            % 对于 w = 0 对称的奈奎斯特频率下标向量
TITLE = sprintf('N = %i,L = %i',N,N * T);     % 变数值为格式控制下的字符串
plot(k * D,abs(Xa));
axis([ - 20,20,0,max(abs(Xa)) + 2]);
xlabel('\Omega');ylabel('| X(j\Omega)|')      % N = 100, 200, 800, 1200 分别执行
title(TITLE);grid;
```

程序运行结果如图 4.38 所示。

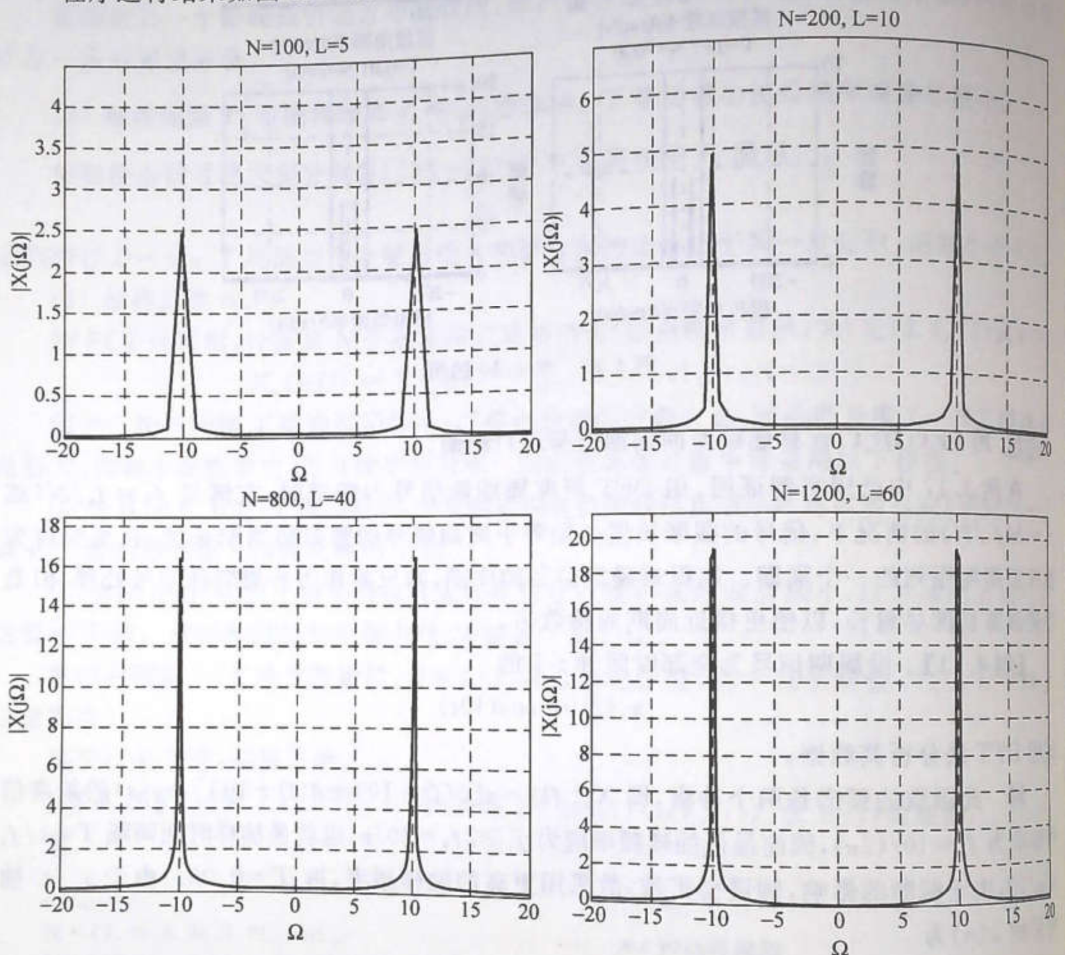

图 4.38 例 4.11 的图

【例 4.12】 采用例 3.2 中的周期序列，即$\tilde{x}(n)=\begin{cases}1, & 0\leqslant n\leqslant 4\\ 0, & 5\leqslant n\leqslant N-1=9\end{cases}$，求此周期序列的 DFS，并作图。

解 利用 1.5 节中的周期序列形成方法——模运算法，并利用例 3.2 中已求出的 $\tilde{X}(k)=\text{DFS}[\tilde{x}(n)]$ 的表达式 $\tilde{X}(k)=\text{DFS}[\tilde{x}(n)]=\mathrm{e}^{-\mathrm{j}2\pi k/5}\sin(\pi k/2)/\sin(\pi k/10)$。

```
clc;clear;close all
x = [ones(1,5),zeros(1,5)];
N1 = 10;N = length(x);m = 3;
nx = 0:m * N - 1;                             % 取三个周期画图
xn = x(mod(nx,N) + 1);                        % 形成周期序列
```

```
k=[0:29]+eps;
Xk=exp(-j*2*pi*k/5).*sin(pi*k/2)./sin(pi*k/10);
magXk=abs(Xk);
angXk=angle(Xk).*180./pi;
subplot(411);stem(x,'.');axis([0,29,0,1.1]);
ylabel('x(n)');
subplot(412);stem(xn,'.');axis([0,29,0,1.1]);xlabel('n')
ylabel('x10(n)');
subplot(413);stem(k,magXk,'.');ylabel('|X10(k)|');
subplot(414);stem(k,angXk,'.');xlabel('k');ylabel('argX10(k)');
```

程序运行结果如图 4.39 所示。

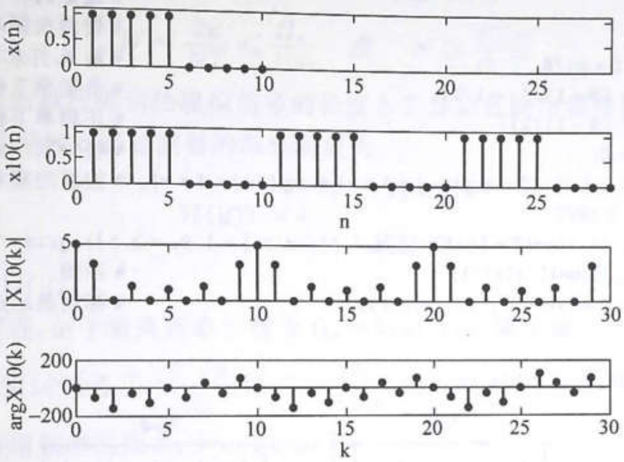

图 4.39　例 4.12 的图

6. 用 IFFT，从频谱 $X(e^{j\Omega})=X(e^{j\omega/T})$ 计算离散时间序列 $x(n)$。

$$x(n)=\frac{1}{N}\sum_{k=0}^{N-1}X(k)W_N^{-nk}=\frac{1}{N}\sum_{k=0}^{N-1}X(k)e^{j\frac{2\pi}{N}nk}=\text{IFFT}[X(k)]$$

其中 $X(k)=X(e^{j\omega})|_{\omega=2\pi k/N}=X(e^{j\Omega T})/_{\Omega=2\pi k/(NT)}$

利用 IDFT(IFFT)时要注意几点：

(1) 如果用 Ω 作变量，则必须给出 $T=1/f_s$，其中 T 为抽样间隔(sec)，f_s 为抽样频率(Hz)，必须满足 $f_s\geqslant 2f_m$，f_m 为信号的最高频率，即要求

$$T\leqslant\frac{1}{2f_m}=\frac{\pi}{\Omega_m}$$

(2) 若 $x(n)$ 为实序列，则 $X(e^{j\omega})$ 必须满足共轭对称关系，即实部(模)偶对称，虚部(相角)奇对称，若表现为 $X(k)$，则需满足圆周共轭对称的关系，因此若只给出 $X(e^{j\omega})$ 的表达式又看不出它的对称关系式，则必须构造一个在 $0\leqslant\omega<\pi$ 和 $-\pi\leqslant\omega<0$ 范围的圆周共轭对称的抽样值，即将 $k<0$ 部分平移到 $k\geqslant 0$ 部分的右端，以便形成 $n=0:N-1$ 的序列。

即：**若设 $k_l=\text{floor}(-(N-1)/2:-1/2)$，$k_h=\text{floor}(0:(N-1)/2)$**

则取　$k=[k_h,k_l]$

这样移动后频谱的抽样序列一定满足圆周共轭对称关系。

(3) 由于有运算误差，可能算出值有微小的虚部，需将其舍掉，但若虚部值很大，则可能是所给频谱不满足共轭对称性。

(4) 频域抽样点数为 N，当 N 的选取不当，可能引起时域的混叠失真，这在 3.5.1 节已有分析，故要检验结果的正确性。即选 N 必须满足 $N \geqslant M$（M 是序列的长度点数）。

【例 4.13】 若 $X(e^{j\omega})=3+e^{-j\omega}-2e^{-j3\omega}+e^{-j4\omega}$，求 $x(n)=\mathrm{IDTFT}[X(e^{j\omega})]$。

由于此例一看就知：$x(n)=\{\underline{3},1,0,-2,1\}$，但又无法给出整个频段的频谱，故用数值分析法来求解它。

解 由于 $X(e^{j\omega})$ 是由 ω 作变量，则可令 $T=1$，且必须将 ω 在 $[0,2\pi]$ 间分成 N 等份。

MATLAB 程序如下：

```
clc;clear;close all
T = 1;c = [4,8];                                              % 设定两种 N 值
for i = 1:2                                                   % 做两次循环运算
    N = c(i);D = 2 * pi/N;                                    % 取 N,并求模拟频率分辨率
    kl = floor( - (N-1)/2: - 1/2);                            % 负频率下标向量
    kh = floor(0:(N-1)/2);                                    % 正频率下标向量
    w = [kh,kl] * D;                                          % 将负频率移到正频率的右方
    X = 3 + exp( - j * w) - 2 * exp( - j * 3 * w) + exp( - j * 4 * w); % 按新的频率排序,输入数字频谱
    x = ifft(X); % IFFT
    subplot(1,2,i),stem(T * [0:N-1],x,'.');axis([ - 1,9, - 3,5]);grid
                                                              % 画图
    xlabel('n');ylabel('x(n)');
    str = ['N = ',num2str(N)];title(str);                     % 标题显示点数
end
```

程序运行结果如图 4.40 所示。

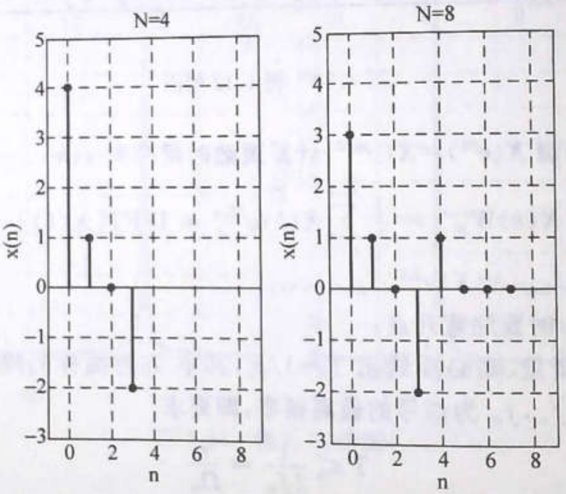

图 4.40 例 4.13 的图

讨论：当频域抽样点数大于或等于序列长度时，频域抽样后，时域周期延拓不会产生混叠现象，故 $N=4$ 时，小于时域序列长度（长度为 5），产生时域混叠现象；当 $N=8>5$ 时，产生的时域序列是正确的，只不过在尾部多出三个零值点。

7. 用 FFT，从频谱 $X(j\Omega)$ 计算连续时间信号 $x_a(t)$。

从(3.6.18)式知

$$x(t)\big|_{t=nT}=x(n)\approx\frac{1}{T}\mathrm{IDFT}[X(jk\Omega_0)]$$

因而用 FFT(DFT)从频谱 $X(j\Omega)$ 计算 $x(t)$ 时，可以利用上一个例子，只是要把 X 换成 X/T，即将模拟信号的频谱换成数字信号的频谱，并且要用 plot 代替 stem，因为 plot 可以自动直接内插将 $x(n)$ 转换成连续时间信号 $x(t)$，但在程序应用上有两个问题要讨论：

(1) **T 的选择**。为了使 plot 的直线内插能真正接近等于 $x(t)$，必须使所选 T 值比满足抽样定理所需 $T<\frac{\pi}{\Omega_m}$ 要小得多，例如取 $T=0.1\frac{\pi}{\Omega_m}$。

(2) **N 的选择**。T 减小，相应的 N 也要加大，N 的选择主要考虑在频谱 $X(j\Omega)$ 有值(不等于零)的频率范围内样值数要足够多，例如超过 100 个使得频域抽样点间步长很小，频域抽样值更为精确地反映频谱特性，若以 Ω(角频率)考虑，则有

$$D=\frac{2\pi}{NT}\leqslant\frac{\Omega_m}{100}\quad 或\quad N\geqslant\frac{200\pi}{\Omega_m T}$$

N 的选择其次要考虑最后得到的模拟信号的长度 NT 足以反映出模拟信号的变化。

【例 4.14】 设理想低通滤波器的频率响应为

$$H(j\Omega)=\begin{cases}1, & |\Omega|<5\\ 0, & 其他\ \Omega\end{cases}$$

求 $h(t)=\mathrm{IDTFT}[H(j\Omega)]$。

解 按抽样定理，由于最高频率分量为 $\Omega_m=5\mathrm{rad/s}$，则满足抽样定理的时域抽样间隔 $\frac{\pi}{\Omega_m}=0.6283$，按上面讨论选 $T=\frac{0.1\pi}{\Omega_m}$，则 $f_s=\frac{\Omega_m}{0.1\pi}$，$\Omega_s=2\pi f_s=20\Omega_m=100$。

故 奈奎斯特角频率范围为 $[-\pi/T:\pi/T)=[-50:50)$

因而 $N\geqslant\frac{200\pi}{\Omega_m T}=\frac{200\pi}{0.1\pi}=2000$

则角频率分辨率为 $D=\frac{2\pi}{NT}=0.05$，于是在有值频带内可有 $\Omega_m/D=100$

所对应的边界频率的下标为 $M=\mathrm{floor}(\Omega_m/D)=100$

```
clc;clear all
wc = 5;T = 0.1 * pi/wc;                            % 选 T 为抽样间隔的 1/10
N = 100 * 2 * pi/wc/T;                             % 输入的样点数
D = 2 * pi/(N * T);w = [0:N-1] * D;                % 模拟角频率的分辨率及频率向量
M = floor(wc/D);                                   % 有效频段的边界下标
H = [ones(1,M+1),zeros(1,N-2*M-1),ones(1,M)];      % 按新的频率排序后的频谱
h = ifft(H/T);                                     % 用 IFFT 求 h(n)/T
plot([0:N-1] * T,h) ;xlabel('t');ylabel('h(t)');   % 用 plot 直线样值画出单位抽样响应
```

程序运行结果如图 4.41 所示。

4.11.3 线性调频 z 变换(CZT)算法

CZT 算法的分析见 4.8 节，它是在更大范围的 z 平面上(既可是单位圆，也可不是单位圆，既可是整个单位圆，也可是单位圆上部分频段；更可以是 z 平面任意螺旋线上)来计算 $X(z_k)$ 的值。

MATLAB 信号处理工具箱中，计算 CZT 的函数为 czt，调用格式为

```
y = czt(x,m,w,a)
```

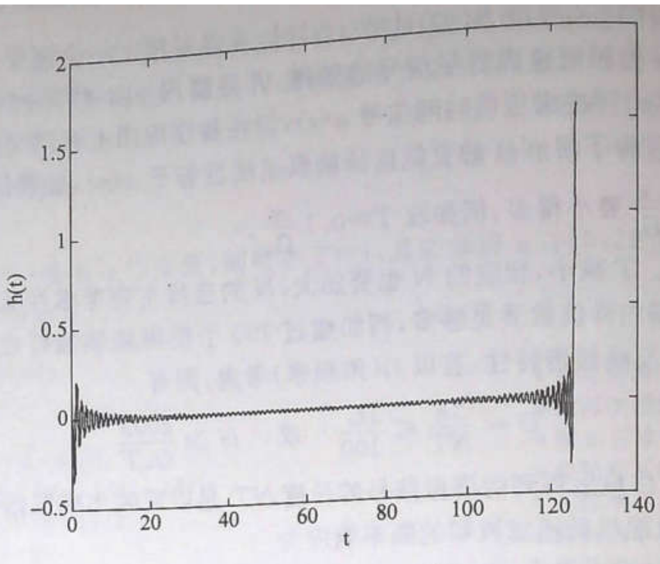

图 4.41　例 4.14 的图

它可以计算信号 x 的 m 点线性调频 z 变换，m、w、a 皆为标量，此函数沿 z 平面指定的弧线计算 z 变换的抽样值，这个弧线由 m、w 和 a 来确定，m 是变换的长度，w 是弧线上各相邻点之间的复数比值，a 是复数的起点。各量的定义可见 4.8 节中的(4.8.1)式～(4.8.3)式。

CZT 的定义见(4.8.4)式，利用 MATLAB 函数时，数组从 1 开始计算，故可改写为

$$X(z_k)=\sum_{n=1}^{N}x(n)z_k^{-(n-1)},\quad z_k=aw^{-(k-1)},\quad 1\leqslant k\leqslant M$$

以下举例说明，若螺线取为单位圆时，如何用 CZT 来将部分频段上的频谱加以细化。

【例 4.15】 采用 CZT 算法研究窗函数线性相位 FIR 滤波器过渡带中更详细的频率特性。设 $f_p=100\text{Hz}$，$f_{st}=200\text{Hz}$，抽样频率 $f_s=1000\text{Hz}$。

解

```
clc;clear all
h = fir1(40,150/500);
fs = 1000;f1 = 100;f2 = 200;                          % 模拟频率 Hz
m = 1024;
w = exp( - j * 2 * pi * (f2 - f1)/(m * fs));          % z 平面弧线
a = exp(j * 2 * pi * f1/fs);                          % 弧线的起点
H = fft(h,1000);                                      % 用 FFT 求滤波器频率响应抽样值
H1 = czt(h,m,w,a);                                    % 利用 CZT 研究过渡带内的频率特性
fH = (0:length(H) - 1)' * 1000/length(H);
fH1 = ((0:length(H1) - 1)' * (f2 - f1)/length(H1)) + f1;
figure(1)
subplot(211)
plot(fH(1:500),abs(H(1:500)));axis([1,500,0,1.2]);
title('FFT');xlabel('f');ylabel('H(jf)');grid
subplot(212)
plot(fH1,abs(H1));axis([f1,f2,0,1.2]);
title('czt');xlabel('f');ylabel('H(jf)');grid;
```

程序运行结果如图 4.42 所示。

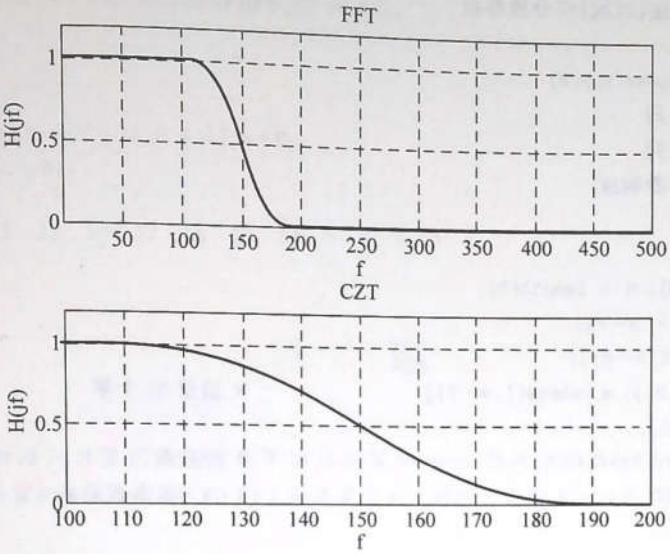

图 4.42 例 4.15 的图

4.11.4 重叠保留法与重叠相加法

1. 采用 FFT 的分段卷积法——重叠相加法实现线性卷积的 MATLAB 程序可参照 4.9.1 节讨论及图 4.27，它是输入分段不重叠，输出将重叠部分相加。MATLAB 信号处理工具箱中，函数 **fftfilt.m** 就是实现这一运算的，其中在求各段的线性卷积时采用 FFT 算法，故求卷积的速度很快，由于求卷积就是计算滤波器输出，故有上述命名(fftfilt)。它也是适宜输入序列很长，而单位冲激响应很短的情况，有两种格式：

```
y = fftfilt(b,x)
y = fftfilt(b,x,N)
```

这里，b 是系统单位冲激响应(FIR 滤波器)，x 是输入的长序列。对第二种表示法，所采用的 FFT 长度为 $n_{fft}=2$ ^nextpow2(N)，N 必须大于 b 的长度，而数据分段长度为 $N_1=n_{fft}-$ length(b)+1。其中 nextpow2(N) 函数则是找出大于 N 而又与 N 最接近的 2 的幂指数。例如 N=31，则 nextpow2(N)=5，又如 N=40，则 nextpow2(N)=6。

对第一种表示法，fftfilt 会选择 FFT 的长度及数据段长度，以保证合适的执行时间。

这一程序中，b 既可代表单位冲激响应(有限长)，又可代表 FIR 滤波器系统函数(以 z^{-1} 表示，即 z 的负幂表示)的系数，所以这一程序又是利用重叠相加法基于 FFT 的 FIR 滤波。

这种方法没有采用 FFT，故计算效率不高。改成 FFT 运算法是很方便的。

2. 采用时域作重叠保留法(分段圆周卷积)实现线性卷积的 MATLAB 程序。可参照 4.9.2 节及图 4.30，程序(输入分段有重叠，输出去掉重叠部分)ovelpsav.m 如下：

```
function [y] = ovelpsav(x,h,N)
% 用重叠保留法(时域)作分段卷积
% ------------------------------------------------------------------
% [y] = ovelpsav(x,h,N)
% y = 输出序列
% x = 输入序列
% h = 单位冲激响应
% N = 分段长
%
Lx = length(x); M = length(h);
M1 = M-1; L = N-M1;
h = [h zeros(1,N-M)];
x = [zeros(1,M1), x, zeros(1,N-1)]                    % 前置 M1 个零
%K=ceil(Lx/L);                                         % 分段数
K = floor((Lx+M1)/L)+1;                               % 与后续各段卷积
y = zeros(K,N);
for k=0:K-1
    xk = x(k*L+1:k*L+N);
    y(k+1,:) = circonvt(xk,h,N);
end                                                    % 去掉前 M1 个样值
y=(:,M:N)';                                            % 装成输出
Y=(y(:))';
```

3. 采用频域方法(利用 FFT)作重叠保留法(分段)实现线性卷积的 MATLAB 程序,重叠保留法的 FFT 算法分析见 4.9.2 节,尤其是参照图 4.30,程序 hsolpsav.m 如下:

```
function [y] = hsolpsav(x,h,N)
% 用 FFT 的重叠保留法作分段卷积
%
% [y] = hsolpsav(x,h,N)
% y = 输出序列
% x = 输入序列
% h = 脉冲响应
% N = 分段长(必须是二的幂次)
%
N = 2^(ceil(log10(N)/log10(2)));
Lx = length(x); M = length(h);
M1 = M-1; L = N-M1;
H = fft(h,N);
K=ceil(Lx/L);                                          % 分段的数目
x = [zeros(1,M1), x, zeros(1,N-1)];                    % 前置 M1 个零点
y= zeros(K,N);
for k=0:K-1
    Xk = fft(x(k*L+1:k*L+N));
    y(k+1,:) = real(ifft(Xk.*H));
end
y = y(:,M:N)'; y = (y(:))';                            % 去掉前面 M1 个值
```

【例 4.16】 已知 $x(n)=13-n, 0\leqslant n\leqslant 12, h(n)=\{2,-1,1\}$，用重叠保留法（采用 FFT 计算）。求 $y(n)=x(n)*h(n)$，设分段长度 $N=6$。

解

```
clc;clear all;
n=[0:12];x=13-n;h=[2,-1,1];N=6;
y=hsolpsav(x,h,N);

y=26  11  23  21  19  17  15  13  11  9  7  5  3  1  1
```

7. 圆周卷积。用圆周卷积计算线性卷积。

（1）时域方法。

* 圆周卷积的公式见(3.4.46)式，若用圆周卷积矩阵表示，则采用(3.4.48)式。下面采用圆周移位关系的表达式来实现圆周卷积，即利用开发出的 **circonvt.m** 函数来实现圆周卷积。

$$y(n)=\left[\sum_{m=0}^{L-1} x_1(m)\, x_2((n-m))_L\right] R_L(n) \quad (3.4.46)$$

```
function y = circonvt(x1,x2,N)
% 在 x1 和 x2(时域)之间的 N 点圆周卷积
% ----------------------------------------
% [y] = circonvt(x1,x2,N)
% y = 包含圆周卷积的输出序列
% x1 = 长度 N1 <= N 的输入序列
% x2 = 长度 N2 <= N 的输入序列
% N = 圆周缓冲器的大小
% 方法 y(n) = sum(x1(m) * x2((n-m)mod N))
```

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(L-1) \end{bmatrix} = \begin{bmatrix} x_2(0) & x_2(L-1) & \cdots & x_2(1) \\ x_2(1) & x_2(0) & \cdots & x_2(2) \\ \vdots & \vdots & \ddots & \vdots \\ x_2(L-1) & x_2(L-2) & \cdots & x_2(0) \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_1(1) \\ \vdots \\ x_1(L-1) \end{bmatrix} \quad (3.4.48)$$

```
% Check for length of x1
if length(x1)> N
        error('N 必须 >= x1 的长度')
end

% Check for length of x2
if length(x2)> N
        error('N 必须 >= x2 的长度')
end

x1 = [x1 zeros(1,N - length(x1))];
x2 = [x2 zeros(1,N - length(x2))];

m = [0:1:N - 1];
x2 = x2(mod(- m,N) + 1);
H = zeros(N,N);

for n = 1:1:N
    H(n,:) = cirshift(x2,n - 1,N);
end

y = x1 * H';
```

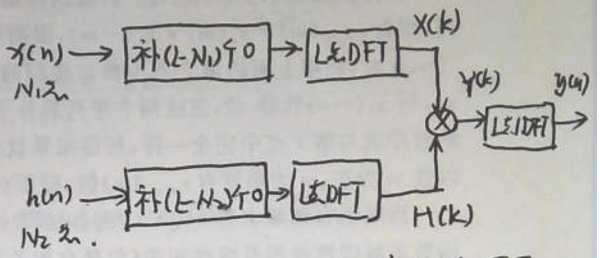

图3.10. 利用DFT计算两有限长L点序列的圆周卷积和框图.

【例 3.24】 已知 $x_1(n)=[2,4,3,1]$，$x_2(n)=[2,1,3]$，求 4 点圆周卷积 $y(n)=x_1(n)④x_2(n)$。

解：

```
clc;clear all
x1 = [2  4  3  1];x2 = [2  1  3];
y = circonvt(x1,x2,4)

y = 14     13     16     17
```

* 用圆周卷积计算线性卷积。

显然，当 $N \geqslant N_1+N_2-1$ 时，由 circonvt(x1,x2,N) 所得到的结果就等于线性卷积结果，即

$$x_1(n)⑥x_2(n) = x_1(n) * x_2(n) = [4,10,16,17,10,3]$$

(2) 在频域中计算来实现圆周卷积及线性卷积。

* 实际上就是利用图 3.10 的框图来求解，它既可以求圆周卷积，也可以求线性卷积。求圆周卷积时，只需将短序列长度补零值点，补到和长序列的长度相等即可。利用 DFT 矩阵 dftmtx(L)及 IDFT 矩阵 conj(dftmtx(L)/L)即可求解。

* 当然最方便的办法是采用以下几条简单语句即可（利用 FFT（快速傅里叶变换）算法）

```
X1 = fft(x1,N);   X2 = fft(x2,N);
Y = X1 * X2;      y = ifft(Y,N);
```

显然，当 N 满足 $N \geqslant N_1+N_2-1$ 时，这样求出的圆周卷积结果就代表线性卷积结果。