信号分析与处理

——MATLAB语言及应用

黄文梅　熊桂林　杨　勇　编著

国防科技大学出版社
·长沙·

内 容 简 介

本书紧密结合信号分析和处理的基本知识，详细介绍了 MATLAB 语言的应用和编程技术。主要内容有：信号与系统基础；信号的傅里叶变换和 z 变换；模拟滤波器和数字滤波器设计；滤波器的实现及分析；随机信号相关分析和功率谱估计及其应用；MATLAB 信号处理交互式图形用户界面等。

本书将基本理论和 MATLAB 语言有机结合起来，并有丰富的应用实例和习题，可以帮助学生在学习基础知识和掌握计算机工程应用能力两方面都起到事半功倍的效果。

本书可作为自动控制、机电工程、机械设计与自动化、通信工程、电力等专业本科生、研究生有关课程的教材或参考书，也可作为有关工程技术人员、研究人员学习和运用 MATLAB 语言的自学用书。

图书在版编目(CIP)数据

信号分析与处理：MATLAB 语言及应用/黄文梅等编著. —长沙：国防科技大学出版社，2000.2

ISBN 7-81024-601-1

Ⅰ.信… Ⅱ.黄… Ⅲ.①信号分析②信号处理③MATLAB 语言 Ⅳ.TN911

中国版本图书馆 CIP 数据核字(1999)第 72310 号

国防科技大学出版社出版发行

电话：(0731)4555621 邮政编码：410073

E-mail：gfkdcbs@public.cs.hn.cn

责任编辑：何 晋 责任校对：文 慧

新华书店总店北京发行所经销

国防科技大学印刷厂印装

*

787×1092 1/16 印张：19.25 字数：445 千

2000 年 2 月第 1 版第 1 次印刷 印数：1—4000 册

*

定价：23.00 元

前 言

MATLAB 是一种面向科学和工程计算的高级语言，现已成为国际公认的最优秀的科技界应用软件，在世界范围内广为流行和使用。该软件的特点是，强大的计算功能、计算结果和编程可视化及极高的编程效率。这是其他语言无与伦比之处。MATLAB 包含的几十个工具箱，覆盖了通信、自动控制、信号处理、图像处理、财经、化工、生命科学等科学技术领域，汲取了当今世界这些领域的最新研究成果，已经成为从事科学研究和工程设计不可缺少的工具软件。今天，在欧美高等院校里，MATLAB 已成为大学生、硕士生、博士生、教师的必备的基本技能，广泛应用于科学研究、工程计算、教学、撰写论文等。国内不少高校也正在推广应用 MATLAB 软件，用户愈来愈多。

本书专为大学自动控制、机械电子、机械制造与自动化、电力电气、通信工程等专业大学本科生、研究生、教师、科技工作者编写。本书力图将信号分析与处理、信号与系统、数字信号处理、工程测试等有关教学内容和 MATLAB 语言紧密、有机地结合起来，使学生在学习基础理论知识的同时学会应用 MATLAB；在学习应用 MATLAB 的同时，加深对基础知识的理解，增强学生的计算机应用能力，提高教学效果。本书编入大量 MATLAB 应用程序，每章也有一定的数量习题，供读者能尽快掌握 MATLAB 的编程技巧。

本书采用 MATLAB 5 的最新版本编写，和 MATLAB 以前的版本相比，MATLAB 5 包含功能更强大的编程工具，内容更丰富的工具箱，编程语言更简捷、更友好的图形界面和可视化编程。

本书和已经出版的《系统分析与仿真——MATLAB 语言及应用》是一套丛书，内容相互联系，涉及自动控制和信号处理领域内主要研究课题和最新研究动向。

本书共分七章和附录。第一、三章由熊桂林编写，第五、七章和附录由杨勇和周力行编写，第二、四、六章由黄文梅编写，唐亚利编写了各章习题。全书由黄文梅汇总整理。本书由湖南大学林丞教授主审。

由于时间仓促，不当之处在所难免，恳请读者不吝指教。

编者

1999 年 9 月

目　录

第一章　信号与系统基础

1.1　连续时间信号

连续时间信号是指在所讨论的时间间隔内，对于任意时间值(除若干个不连续点之外)都可给出确定的函数值。一些基本连续信号的表达式和波形有：

一、指数信号

$$f(t) = Ke^{at} \tag{1.1-1}$$

式中，a 是实数。a>0，信号幅值随 t 增加而增大，为增值函数；a<0，幅值随 t 增加而减少，为衰减函数。实际中，常遇到的信号为衰减指数信号，如图 1.1 所示。

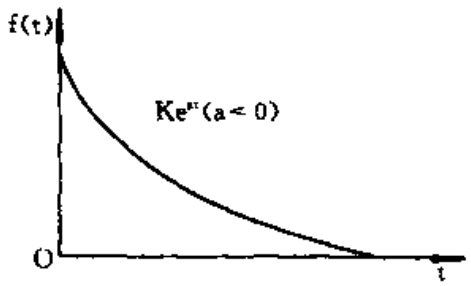


图 1.1　衰减指数信号

二、正弦信号

$$f(t) = K\sin(\omega t + \theta) \tag{1.1-2}$$

式中，K 为振幅，ω 为角频率，单位为弧度/秒，θ 为初相位，单位为弧度，其波形如图 1.2 所示。

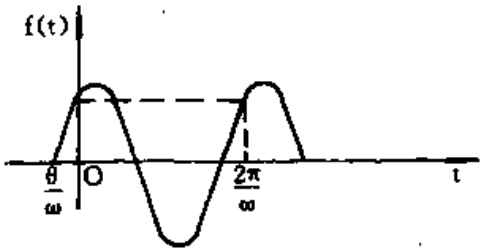


图 1.2　正弦信号

余弦信号和正弦信号仅在相位上相差 π/2。

正弦信号和余弦信号常借用复指数信号来表示，由欧拉公式可知

$$e^{j\omega t} = \cos\omega t + j\sin\omega t \tag{1.1-3}$$

$$e^{-j\omega t} = \cos\omega t - j\sin\omega t$$

三、单位阶跃信号

$$u(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geqslant 0 \end{cases} \tag{1.1-4}$$

信号的波形如图 1.3 所示。

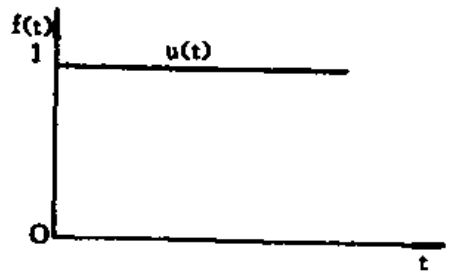


图 1.3　单位阶跃信号

信号在 t = 0 时发生跳变。

四、单位斜坡信号

$$r(t) = \begin{cases} 0, & t < 0 \\ t, & t \geqslant 0 \end{cases} \tag{1.1-5}$$

信号的波形如图 1.4 所示。

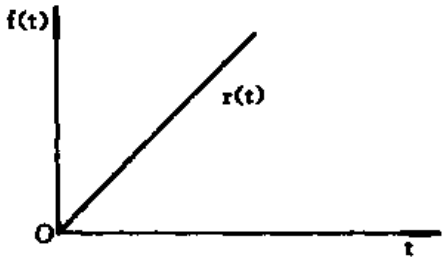


图 1.4　斜坡信号

五、正负号信号

$$\mathrm{sgn}(t) = \begin{cases} 1, & t > 0 \\ -1, & t < 0 \end{cases} \tag{1.1-6}$$

信号的波形如图 1.5 所示。

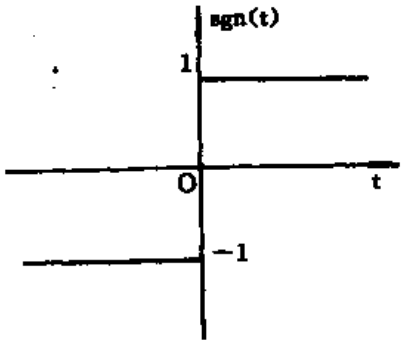


图 1.5　正负号信号

六、脉冲信号

单位脉冲函数可定义为：在 ε 时间内，某一方波 $S_\varepsilon(t)$ 的面积为 1，即满足下式

$$S_\varepsilon(t)=\begin{cases}\dfrac{1}{\varepsilon}, & 0\leqslant t\leqslant \varepsilon\\ 0, & t>\varepsilon, t<0\end{cases} \tag{1.1-7}$$

当 $\varepsilon\to 0$ 时，方波的极限就称为单位脉冲函数，常记作 $\delta(t)$，又称为 δ 函数。用 δ 函数所描述的信号为 δ 信号，如图 1.6 所示。

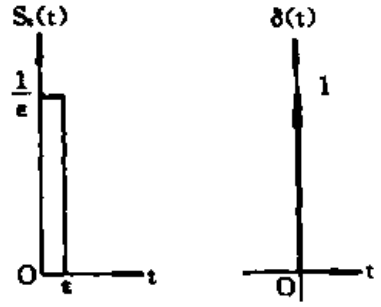


图 1.6 δ 函数数

从函数的极限角度看，

$$\delta(t)=\begin{cases}\infty, & t=0\\ 0, & t\neq 0\end{cases} \tag{1.1-8}$$

从面积的角度看，

$$\int_{-\infty}^{\infty}\delta(t)dt=\lim_{\varepsilon\to 0}\int_{-\infty}^{\infty}S_\varepsilon(t)dt=1 \tag{1.1-9}$$

δ 函数的重要性质有：

(1) 采样特性

$$f(t)\delta(t)=f(0)\delta(t) \tag{1.1-10}$$

或

$$f(t)\delta(t-t_0)=f(t_0)\delta(t-t_0)$$

(2) 积分特性

$$\int_{-\infty}^{t}\delta(\tau)dt=u(t) \tag{1.1-11}$$

$$\int_{-\infty}^{\infty}f(t)\delta(t)=f(0) \tag{1.1-12}$$

或

$$\int_{-\infty}^{\infty}f(t)\delta(t-t_0)dt=f(t_0)$$

(3) 卷积特性

$$f(t)*\delta(t)=\int_{-\infty}^{\infty}f(\tau)\delta(t-\tau)d\tau=f(t) \tag{1.1-13}$$

(4) δ 函数的变换

拉氏变换

$$\mathscr{L}[\delta(t)]=\int_{0}^{\infty}\delta(t)e^{-st}dt=1 \tag{1.1-14}$$

傅氏变换

$$\mathscr{F}[\delta(t)]=\int_{-\infty}^{\infty}\delta(t)e^{-j2\pi ft}dt=1 \tag{1.1-15}$$

任何信号 f(t)可以在时域内近似分解成为具有不同延时的矩形脉冲信号分量的叠加，如图 1.7 所示。当脉宽 $\Delta\tau \to 0$ 时，信号 f(t)可认为是由无数脉冲信号 δ(t)的叠加。即

$$f(t) = \int_{-\infty}^{\infty} f(\tau)\delta(t-\tau)d\tau \tag{1.1-16}$$

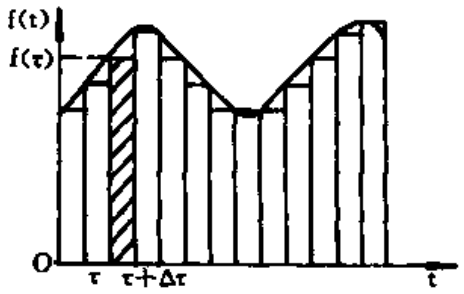


图 1.7　信号分解为脉冲信号叠加

七、sinc 信号

由下面函数式描述：

$$\mathrm{sinc}(t) = \frac{\sin t}{t} \tag{1.1-17}$$

其信号波形如图 1.8 所示。

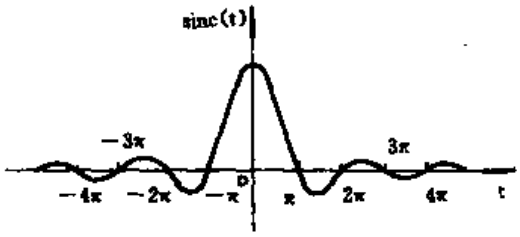


图 1.8　sinc 函数

注意到，该函数是一个偶函数，在 t 的正、负两个方向振幅都是逐渐衰减的，当 $t = \pm\pi, \pm 2\pi, \cdots, \pm n\pi$ 时，函数值等于 0。

sinc(t)函数具有以下性质：

$$\int_0^{\infty} \mathrm{sinc}(t)dt = \frac{\pi}{2} \tag{1.1-18}$$

$$\int_{-\infty}^{\infty} \mathrm{sinc}(t)dt = \pi \tag{1.1-19}$$

八、复指数信号

如果指数信号的指数因子为一复数，则称之为复指数信号，其表达式为

$$f(t) = e^{st} \tag{1.1-20}$$

式中 $s = \sigma + j\omega$.

借助欧拉公式，式(1.1-20)变为

$$f(t) = e^{(\sigma+j\omega)t} = e^{\sigma t}(\cos\omega t + j\sin\omega t) \tag{1.1-21}$$

其结果表明，一个复指数信号可分解为实部和虚部两部分。其中，实部包含余弦信号，虚部

包含正弦信号。指数因子实数 σ 表征了正弦、余弦函数振幅随时间变化的情况，而指数因子虚部 ω 则表示正弦和余弦函数的角频率。实际工程中并不能产生复指数信号，但可利用复指数信号来描述各种基本信号，因此它在信号分析中起了十分重要的作用。

1.2 离散时间信号

如前所述，离散时间信号定义为一时间函数，它只在某些离散的瞬时给出函数值，而在其他处无定义。因此，它是时间上不连续按一定先后次序排列的一组数的集合，故称为时间序列，简称序列，通常表示为

$$\{x(n)\} \quad -\infty < n < +\infty \tag{1.2-1}$$

或具体地写为

$$\{x(n)\} = \{x(-\infty),\cdots,x(-1),x(0),x(1),\cdots,x(\infty)\}$$

x(n)仅对整数 n 才有定义。序列值 x(n)与位置 n 有关，正如连续信号 x(t)与时间 t 有关一样。

离散序列{x(n)}可由连续时间信号 x(t)在 nT 时刻采样而得，T 为采样周期。

用序列(1.2-1)描述的离散时间信号可用图 1.9 表示。

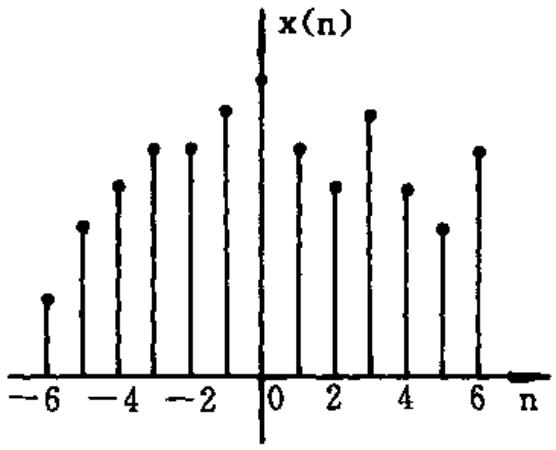


图 1.9 离散时间信号图形表示

式(1.2-1)表示的序列为无限长序列，而实际应用中，序列长度是有限的，为有限长序列。式(1.2-1)中，$n_1 \leqslant n \leqslant n_2$，$n_1$ 和 n_2 均为整数。

MATLAB 是用向量表示序列的。由于 MATLAB 矢量的第一个元素位置是 x(1)，因此为了清楚表示序列{x(n)}要用两个向量，其中一个向量 n 表示序列元素的位置，而另一个向量 x 表示序列值，如

```
n = [-3  -2  -1  0  1  2  3  4];
x = [0  1  2  3  2  1  0  -1];
```

一般情况下，序列值是从 n=0 开始的，因此一个长度为 N 的有限序列的 MATLAB 表示为

```
x = [x(0)  x(1)  …  x(N-1)]
```

对于多通道信号可用多列向量构成的矩阵来表示，矩阵的每个列向量表示一个信号，矩阵每一行表示某一采样时刻各信号值。如三个信号 x，2x 和 x/π 可表示为

```
y=[x 2x x/pi]
```

和连续时间信号类似，离散时间信号的基本形式有

一、单位采样序列

$$\delta(n)=\begin{cases}1, & n=0\\ 0, & n\neq 0\end{cases} \tag{1.2-2}$$

或

$$\delta(n-n_0)=\begin{cases}1, & n=n_0\\ 0, & n\neq n_0\end{cases} \tag{1.2-3}$$

【例 1.1】 用 MATLAB 编写生成单位脉冲序列函数的程序，n∈[−10,10]。

程序清单如下：

```
%MATLAB PROGRAM 1-1
%create a Delta Sequence
n0=0;
n1=-10;
n2=10;

n=[n1:n2];
nc=length(n);
x=zeros(1,nc);
for i=1:nc
   if n(i)==n0
     x(i)=1;
   end
end
stem(n,x)
xlabel('n');ylabel('x(n)');title('Delta Sequence');
grid
```

或采用更简单程序如下：

```
%MATLAB PROGRAM 1-2
%create a Delta Sequence
n0=0;
n1=-10;
n2=10;

n=[n1:n2];
x=[(n-n0)==0];
stem(n,x)
```

```
xlabel('n');ylabel('x(n)');title('Delta Sequence');
grid
```

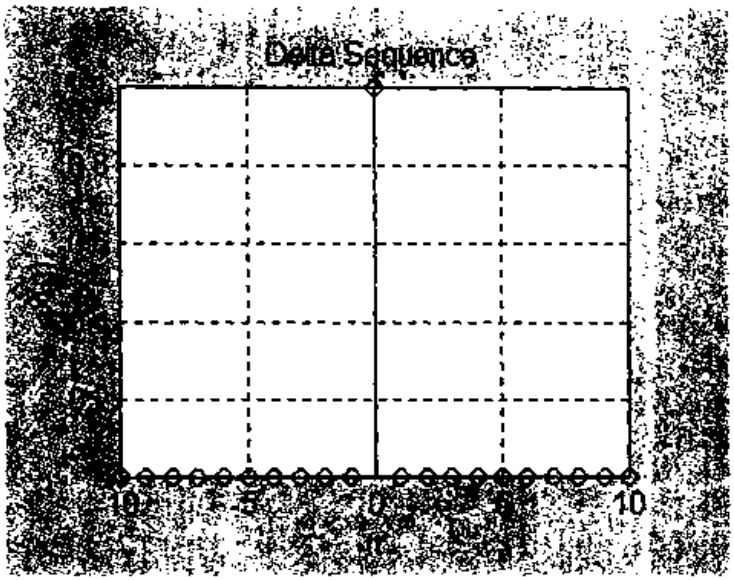


图 1.10　单位采样序列

二、单位阶跃序列

$$u(n)=\begin{cases}1, & u\geqslant 0\\ 0, & u<0\end{cases} \tag{1.2-4}$$

【例 1.2】　用 MATLAB 编写生成单位阶跃序列的程序，$n\in[-10,10]$。

程序清单如下：

```
%MATLAB PROGRAM 1-3
%create unit Step Sequence
n0=0;
n1=-10;
n2=10;

n=[n1:n2];
x=[(n-n0)>=0];
stem(n,x)
xlabel('n');ylabel('x(n)');title('Step Sequence');
grid
```

三、单位斜坡序列

$$x(n)=\begin{cases}0, & n<0\\ n, & n\geqslant 0\end{cases} \tag{1.2-5}$$

【例 1.3】　用 MATLAB 编写生成斜坡序列的程序，$n\in[0,10]$。

```
%MATLAB PROGRAM 1-4
%Create Ramp Sequence
n1=0;
```

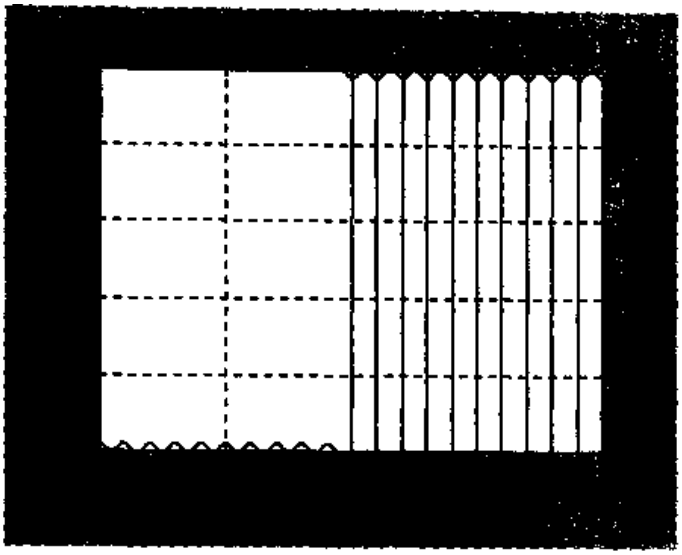


图 1.11 单位阶跃序列

```
n2=1;
n=[n1:0.1:n2];
x=n;
stem(n,x)
xlabel('n');ylabel('x(n)');title('Ramp Sequence');
grid
```

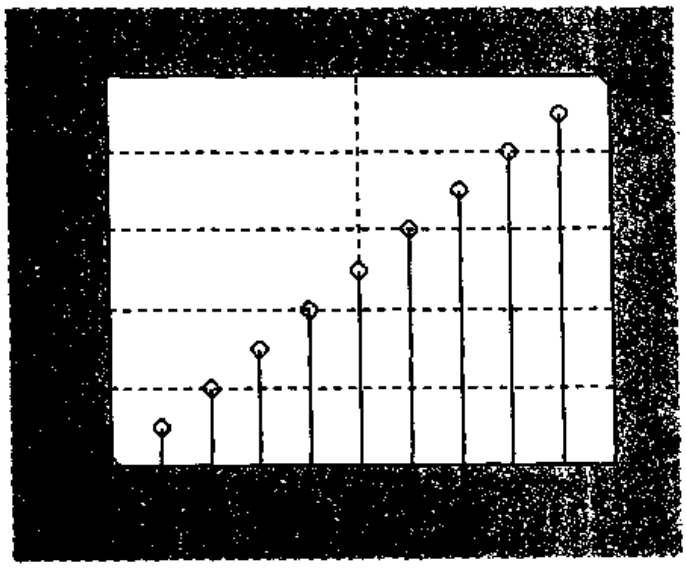


图 1.12 斜坡序列

四、正余弦序列

$$x(n) = K\sin(\omega_0 n + \theta_0), \qquad -\infty < n < +\infty \tag{1.2-6}$$

式中，ω_0 为数字角频率，弧度；θ_0 为数字初相位，弧度。

【例 1.4】 用 MATLAB 编写生成正弦序列 $x(n)=2\sin(0.5\pi n+\frac{\pi}{4})$ 的程序如下：

```
%MATLAB PROGRAM 1-5
%Create Sine Sequence
```

```
n=[0:100];
x=2*sin(0.05*pi*n+pi/4);
stem(n,x)
xlabel('n');ylabel('x(n)');title('Sine Sequence');
grid
```

一个 $x(n)=2\sin(0.5\pi n+\pi/4)$ 的正弦序列波形如图 1.13 所示。

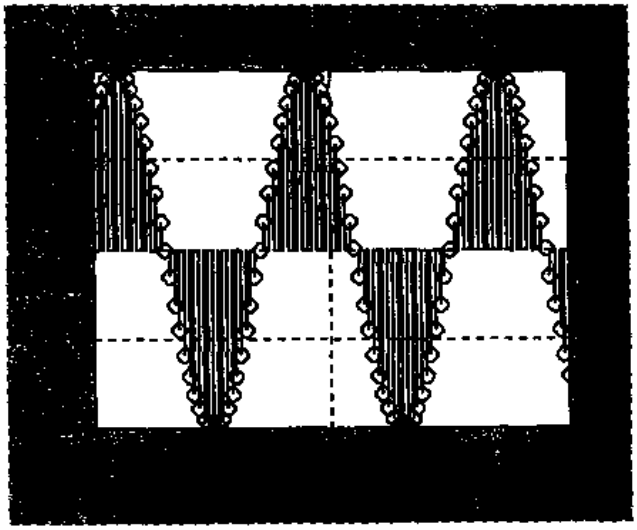


图 1.13 正弦序列

值得注意的是，正弦序列可能是周期序列，也可能不是周期序列。当 $\omega_0/2\pi$ 为有理分式时，即

$$\frac{\omega_0}{2\pi}=\frac{P}{q}$$

式中，P，q 均为整数。

该正弦序列为周期序列，且周期为 q，即 $\sin(\omega_0(n+q))=\sin\omega_0 n$。

五、实指数序列

$$x(n)=a^n \qquad -\infty<n<+\infty \tag{1.2-7}$$

式中，a 为实数。

【例 1.5】 用 MATLAB 编写生成实指数序列 0.5^n 的程序如下：

```
%MATLAB PROGRAM 1-6
%Create Real power Sequence
n=[0:10];
x=(0.5).^n;
stem(n,x)
xlabel('n');ylabel('x(n)');title('Real power Sequence');
grid
```

一个 $x(n)=0.5^n$ 的实指数序列波形如图 1.14 所示。

六、复指数序列

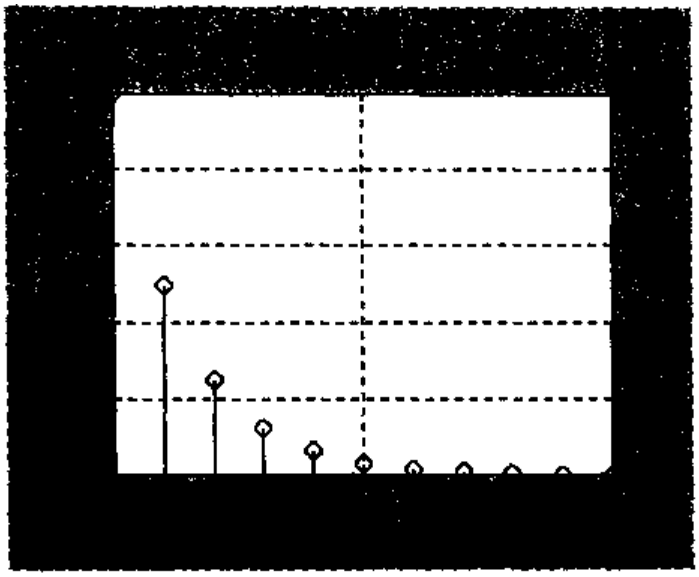


图 1.14 实指数序列

$$x(n) = e^{(\sigma+j\omega_0)n} \qquad -\infty < n < \infty \tag{1.2-8}$$

式(1.2-8)可写为

$$x(n) = Re(n) + jI_m(n) \tag{1.2-9}$$

式中,实部 $Re(n)=e^{\sigma}\cos\omega_0 n$,虚部 $I_m(n)=e^{\sigma}\sin\omega_0 n$.

或

$$x(n) = |x(n)|\angle x(n) \tag{1.2-10}$$

式中,模 $|x(n)|=e^{\sigma}$;幅角 $\angle x(n)=\omega_0 n$.

【例 1.6】 用 MATLAB 编写生成复指数序列 $x(n)=e^{(-0.1+j0.3)n}$的程序,$-10<n<10$。

```
%MATLAB PROGRAM 1-7
%Create Complex power Sequence
clf
n=[-10:10];
alpha=-0.1+0.3*j;
x=exp(alpha*n);

Real_x=real(x);
Image_x=imag(x);
Mag_x=abs(x);
Phase_x=(180/pi)*angle(x);

subplot(221)
stem(n,Real_x);title('Real Part');xlabel('n');
grid
subplot(222)
stem(n,Image_x);title('Imaginary Part');xlabel('n');
```

```
grid
subplot(223)
stem(n,Mag _ x);title('Magnitude ');xlabel('n');
grid
subplot(224)
stem(n,Phase _ x);title('Phase');xlabel('n');
grid
```

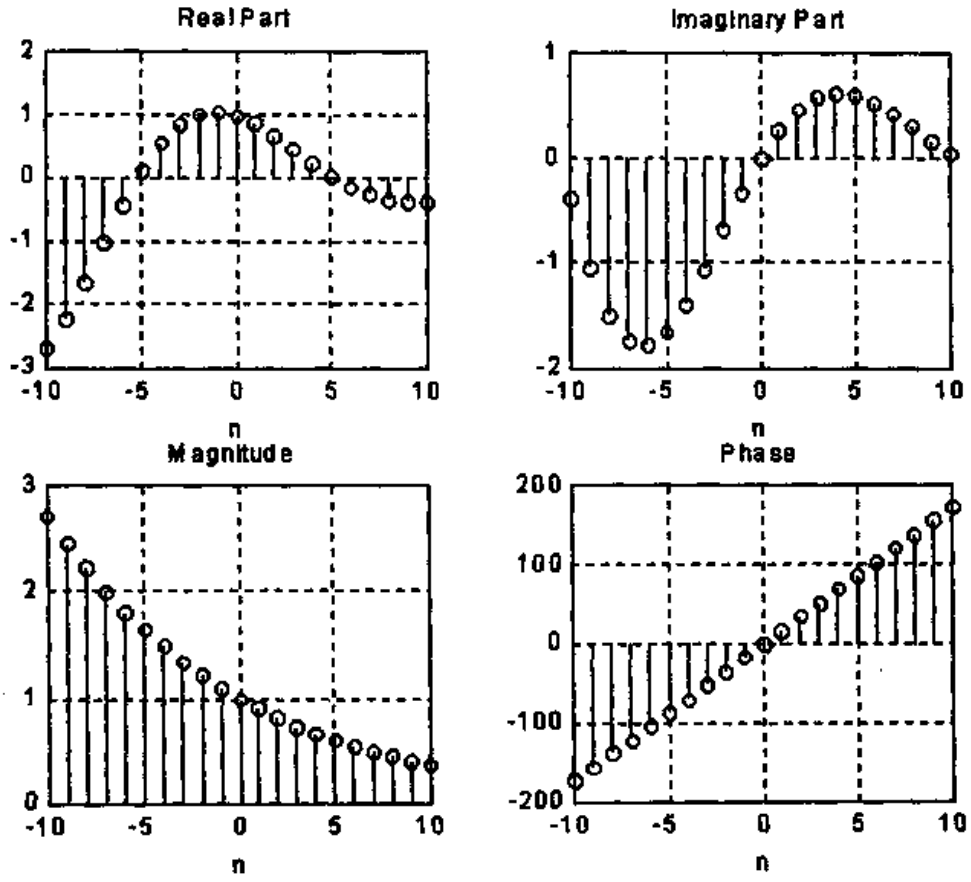


图 1.15　复指数序列

七、随机序列

MATLAB 提供两个产生随机序列的函数：

rand(1,n)用于产生[0,1]上均匀分布随机序列，长度为 n。

randn(1,n)用于产生均值为 0、方差为 1 的高斯随机序列，即白噪声序列，长度为 n。

【例 1.7】 用 MATLAB 编写程序，生成长度为 10 的随机序列。

```
%MATLAB PROGRAM 1-8
%Create a Random Sequence
n=[1:10];
x=rand(1,10);
subplot(221)
stem(n,x);
```

```
xlabel('n');ylabel('x(n)');title('Random Sequence');
grid

x=randn(1,10);
subplot(222)
stem(n,x);
xlabel('n');ylabel('x(n)');title('Random Sequence');
grid
```

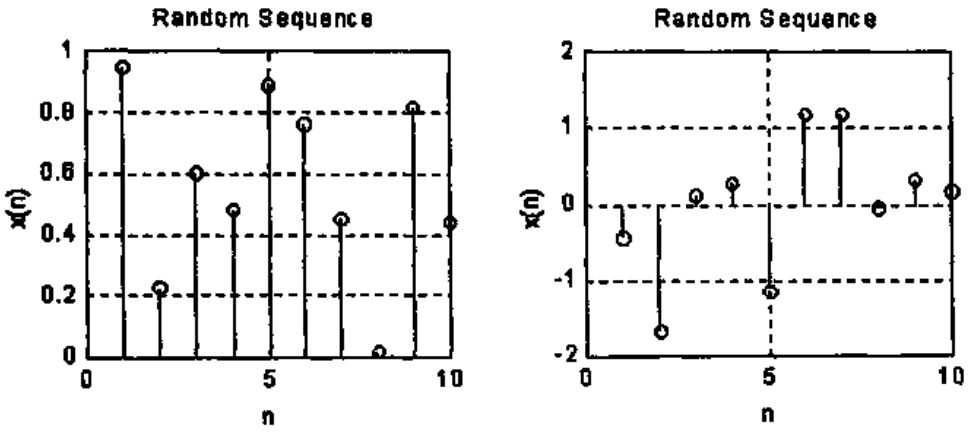


图 1.16 随机序列

八、周期序列

若 x(n)=x(n+T),则 x(n)为周期序列。T 为周期。

用 MATLAB 产生一个信号 x(n)的 P 个周期的周期信号 xp(n),用 MATLAB 语言可直接产生

$$xp = \underbrace{[x(n),x(n),\cdots,x(n)]}_{P个};$$

1.3 信号的运算

1.3.1 连续时间信号

连续时间信号的运算通常有信号的时移、信号倒置、信号加、信号乘、信号微分和积分等。通过信号运算,可由基本信号生成各种复杂信号。

一、信号的时移

$$y(t) = x(t+\tau) \qquad (\tau > 0) \tag{1.3-1}$$

式中,x(t)为原信号,y(t)为新信号,y(t)信号波形相对于 x(t)信号波形向左平移 τ。

若 y(t)=x(t−τ),则 y(t)相对于 x(t)向右平移 τ。

二、信号倒置(折叠)

$$y(t) = x(-t) \tag{1.3-2}$$

y(t)相对于x(t)以纵坐标为对称轴。

三、信号时间尺寸改变

$$y(t) = x(at) \tag{1.3-3}$$

式中，a为时间尺寸变换系数。

a>1，y(t)波形在时间域内被“压缩”成1/a；0<a<1，y(t)波形在时间域内被“扩展”a倍。

四、信号加

$$y(t) = x_1(t) \pm x_2(t) \tag{1.3-4}$$

五、信号微分和积分

$$y(t) = \frac{dx(t)}{dt} \tag{1.3-5}$$

$$y(t) = \int_{-\infty}^{t} f(\tau)d\tau \tag{1.3-6}$$

1.3.2 离散时间信号

离散时间信号一般可用数的序列表示，和连续时间信号类似，序列也可进行运算。这里介绍序列的基本运算和它的MATLAB实现。为了便于以后的应用，这里基本运算以M—文件函数形式给出。

一、信号加

$$x(n) = x_1(n) + x_2(n) \tag{1.3-7}$$

序列加的条件是$x_1(n)$和$x_2(n)$具有相长度，且在相同的采样位置上相加，否则需进行转换。

用MATLAB实现信号加的函数sigadd( )程序清单如下：

```
function [y,n]=sigadd(x1,n1,x2,n2)
% Implements y(n)=x1(n)+x2(n)
n=min(min(n1),min(n2)):max(max(n1),max(n2));
y1=zeros(1,length(n));
y2=y1;
y1(find((n>=min(n1))&(n<=max(n1))==1))=x1;
y2(find((n>=min(n2))&(n<=max(n2))==1))=x2;
y=y1+y2;
```

二、信号乘

$$y(n) = x_1(n) \cdot x_2(n) \tag{1.3-8}$$

序列乘的条件是$x_1(n)$和$x_2(n)$具有相同长度，且在相同的采样位置上相乘，否则需进行转换。

用MATLAB编写的实现序列乘的函数sigmult( )程序清单如下：

```
function [y,n]=sigmult(x1,n1,x2,n2)
%Implements y(n)=x1(n)*x2(n)
```

```
n=min(min(n1),min(n2)):max(max(n1),max(n2));
y1=zeros(1,length(n));
y2=y1;
y1(find((n>=min(n1))&(n<=max(n1))==1))=x1;
y2(find((n>=min(n2))&(n<=max(n2))==1))=x2;
y=y1. * y2;
```

三、信号移位

$$y(n) = x(n + n_0) \tag{1.3-9}$$

序列 y(n)相对于序列 x(n)左移 n_0 个采样周期。

MATLAB 编写的实现序列移位函数 sigshift( )程序清单如下：

```
function [y,n]=sigshift(x,m,n0)
%Implements y(m+k)=x(m)
n=m+n0;
y=x;
```

四、序列折叠

$$y(n) = x(-n) \tag{1.3-10}$$

即序列 x(n)每一项对 n=0 的纵坐标折叠，y(n)和 x(n)相对于 n=0 的纵坐标轴对称。

用 MATLAB 编写的序列折叠函数 sigfold( )程序清单如下：

```
function [y,n]=sigfold(x,m,n0)
%Implements y(n)=x(-n)
y=fliplr(x);
n=-fliplr(n);
```

五、序列的奇偶性

任何一个序列 x(n)都可以分解为偶分量 $x_e(n)$和奇分量 $x_0(n)$两部分之和。

$$x(n) = x_e(n) + x_0(n) \tag{1.3-11}$$

偶分量的定义是

$$x_e(n) = x_e(-n) \tag{1.3-12}$$

奇分量的定义是

$$x_0(n) = -x_0(-n) \tag{1.3-13}$$

式(1.3-11)中，

$$x_e(n) = \frac{1}{2}(x(n) + x(-n)) \tag{1.3-14}$$

$$x_0(n) = \frac{1}{2}(x(n) - x(-n)) \tag{1.3-15}$$

用 MATLAB 编写的函数 sigevenodd( )用于将序列分解成偶序列和奇序列两部分，程序清单如下：

```
function [xeven,xodd,m]=sigevenodd(x,n)
%Real signal decomposition into even and odd parts
```

```
if (imag(x)~=0)
  error('x is not a real sequence');
end

m=fliplr(n);
m1=min([m,n]);m2=max([m,n]);m=m1:m2;
nm=n(1)-m(1);n1=1:length(n);
x1=zeros(1,length(m));
x1(n1+nm)=x;x=x1;
xeven=0.5*(x+fliplr(x));
xodd =0.5*(x-fliplr(x));
```

【例 1.8】 设单位阶跃序列

$$u(n)=\begin{cases}0, & -10\leqslant n<0\\ 1, & 0\leqslant n\leqslant 10\end{cases}$$

将其分解为偶分量和奇分量。

用 MATLAB 编写序列分解的程序如下：

```
%MATLAB PROGRAM 1-9
%Create unit Step Sequence
clf
n0=0;
n1=-10;
n2=10;

n=[n1:n2];
x=[(n-n0)>=0];

subplot(221)
stem(n,x)
xlabel('n');ylabel('x(n)');title('Step Sequence');
grid

%Decomposition of the Sequence
[xeven,xodd,m]=sigevenodd(x,n);

subplot(222)
stem(m,xeven);
xlabel('m');ylabel('x even(n)');title('Even Part');
grid
```

```
subplot(223)
stem(m,xodd);
xlabel('m');ylabel('x odd(n)');title('Odd Part');
grid
```

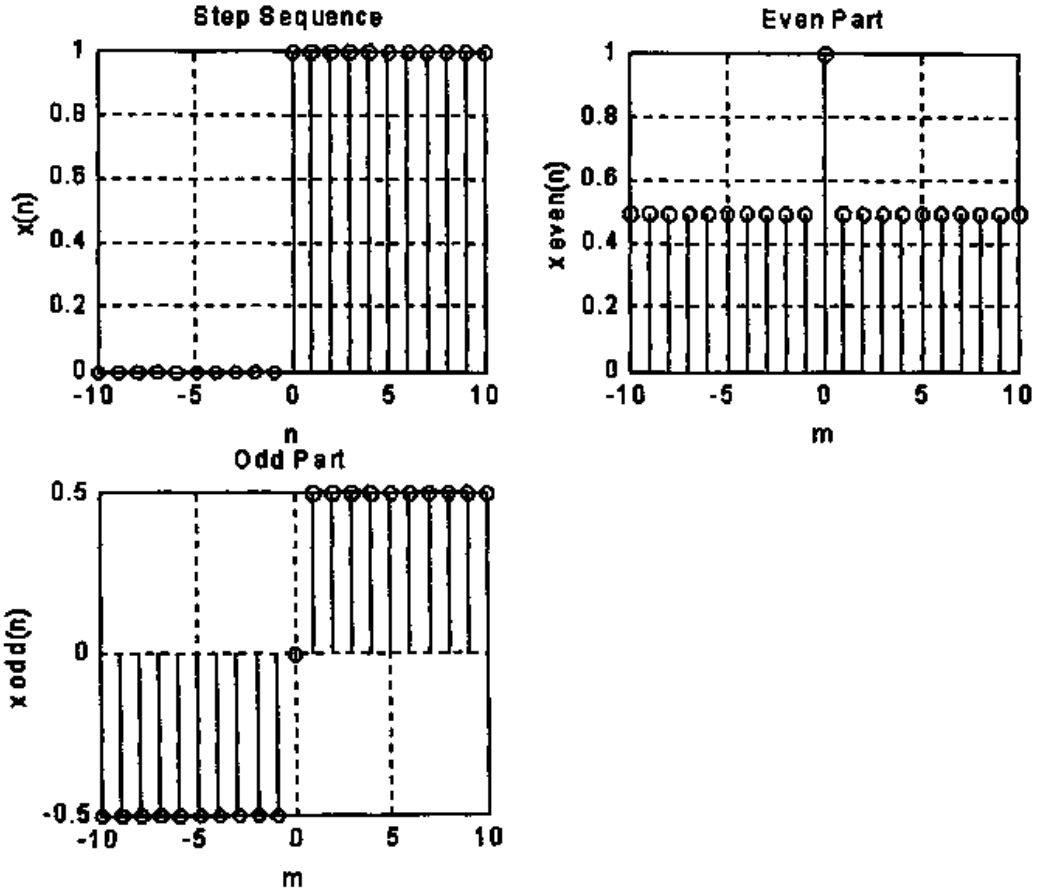


图 1.17　阶跃序列的奇偶分解

1.4　MATLAB 信号生成函数

MATLAB 5 信号处理工具箱提供一些特殊信号波形生成函数(见附录),这些函数在信号处理中起了十分重要作用,这里仅介绍常用的 SQUARE、SAWTOOTH、SINC、DIRIC、RECTPULS 和 PULSTRAN。

一、SQUARE

功能:产生周期为 2π,幅值为±1 的方波信号。

调用格式:

```
x=square(t)
x=square(t, duty)
```

其中,t 为时间向量,duty 为正幅值部分占周期的百分数。

二、SAWTOOTH

功能:产生锯齿波或三角波。

调用格式：

x=sawtooth(t)

x=sawtooth(t, width)

sawtooth(t)用于产生时间向量为t，周期为2π，宽度为width的三角波。这里width为0和1之间的数。width=0.5时，产生标准正三角波。

【例1.9】 (1) 产生幅值为±1的周期性方波，周期为2s，脉宽为1s；(2) 产生幅值为±1的周期性方波，周期为2s，脉宽为0.2s；(3) 产生幅值为±1的周期锯齿波，周期为2s；(4) 产生幅值为±1的周期正三角波，周期为2s.

用MATLAB编写信号生成程序如下：

```
%MATLAB PROGRAM 1-10
clf
x=[0:0.01:10];
y=square(pi*x);
subplot(221)
plot(x,y);
axis([0,10,-2,2]);
title('Square');xlabel('x');ylabel('y');

y=square(pi*x,20);
subplot(222)
plot(x,y);
axis([0,10,-2,2]);
title('Square');xlabel('x');ylabel('y');

x=[0:0.01:10];
y=sawtooth(pi*x);
subplot(223)
plot(x,y);
axis([0,10,-2,2]);
title('Sawtooth');xlabel('x');ylabel('y');

y=sawtooth(pi*x,0.5);
subplot(224)
plot(x,y);
axis([0,10,-2,2]);
title('Sawtooth');xlabel('x');ylabel('y');
```

生成信号波形如图1.18所示。

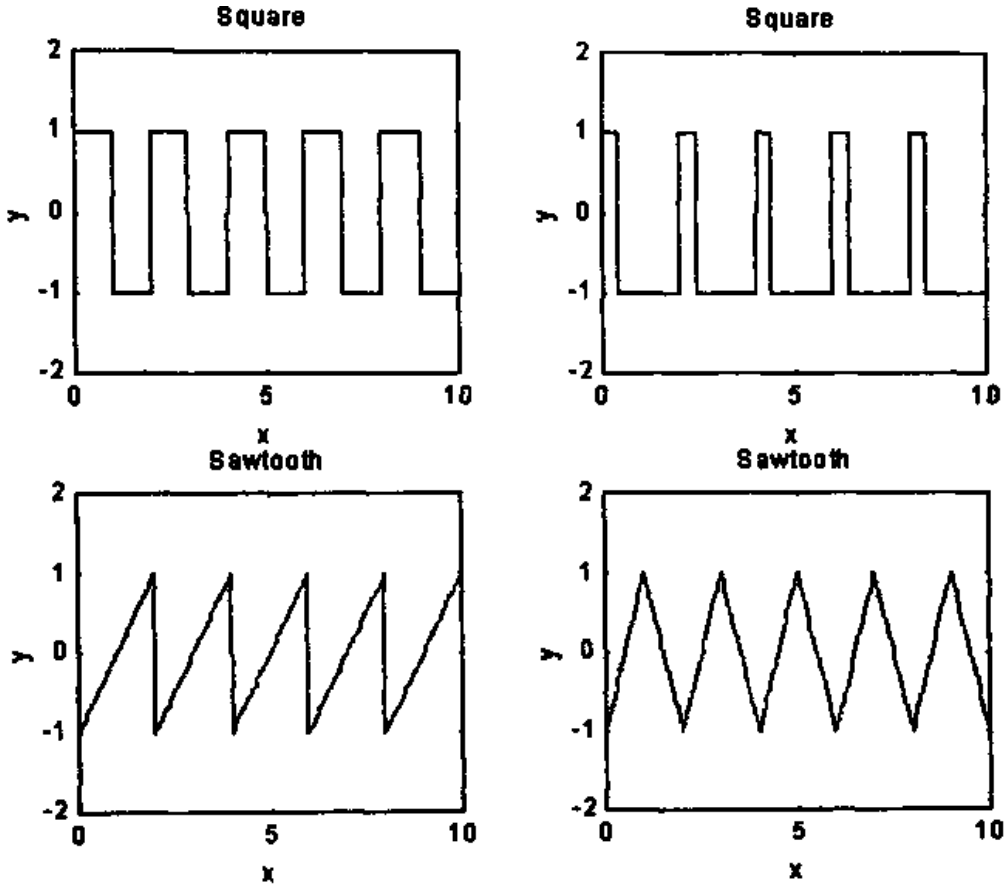


图 1-18　周期方波、锯齿波和三角波

三、SINC

功能：生成 sinc 函数波形。

调用格式：

$$y = \text{sinc}(x)$$

其中，x 为一向量。

函数 sinc(x)的周期为 2π，并随 x 的增加而做衰减振荡，且为偶函数，在 nπ 处的值为零。调用时应注意此特点。

四、DIRIC

功能：产生 Dirichlet 或 sinc 周期函数。

调用格式：

$$y = \text{diric}(x, n)$$

其中，x 为一向量，n 为正整数。

Dirichlet 是周期 Sinc 函数，表达式为

$$\text{diric}(x) = \begin{cases} -1^{k(n-1)}, & x = 2\pi k, \quad k = 0, \pm 1, \pm 2, \cdots \\ \dfrac{\sin(nx/2)}{n\sin(x/2)}, & \text{其他} \end{cases}$$

n 为非零整数。当 n 为奇数时，周期为 2π；当 n 为偶数时，周期为 4π。

【例 1.10】　生成 sinc(x)函数波形和不同 n 的 diric(x, n)波形曲线。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 1-11
clf
x=[-2:0.1:2];
y=sinc(pi*x);
subplot(221)
plot(x,y);
title('Sinc');xlabel('x');ylabel('y');
grid

y=diric(pi*x,5);
subplot(222)
plot(x,y);
title('Dirichlet n=5');xlabel('x');ylabel('y');
grid

y=diric(pi*x,7);
subplot(223)
plot(x,y);
title('Dirichlet n=7');xlabel('x');ylabel('y');
grid

y=diric(pi*x,8);
subplot(224)
plot(x,y);
title('Dirichlet n=8');xlabel('x');ylabel('y');
grid
```

由程序绘出的函数曲线如图 1.19 所示。

五、RECTPULS

功能：在采样点上产生非周期的、单位高度矩形信号。

调用格式：

```
y = rectpuls(t)
y = rectpuls(t,w)
```

其中，t 为时间向量，w 为矩形脉宽。

调用后，以 t = 0 为中心产生宽度为 w，高度为 1 的矩形信号。

六、PULSTRAN

功能：产生脉冲串信号。

调用格式：

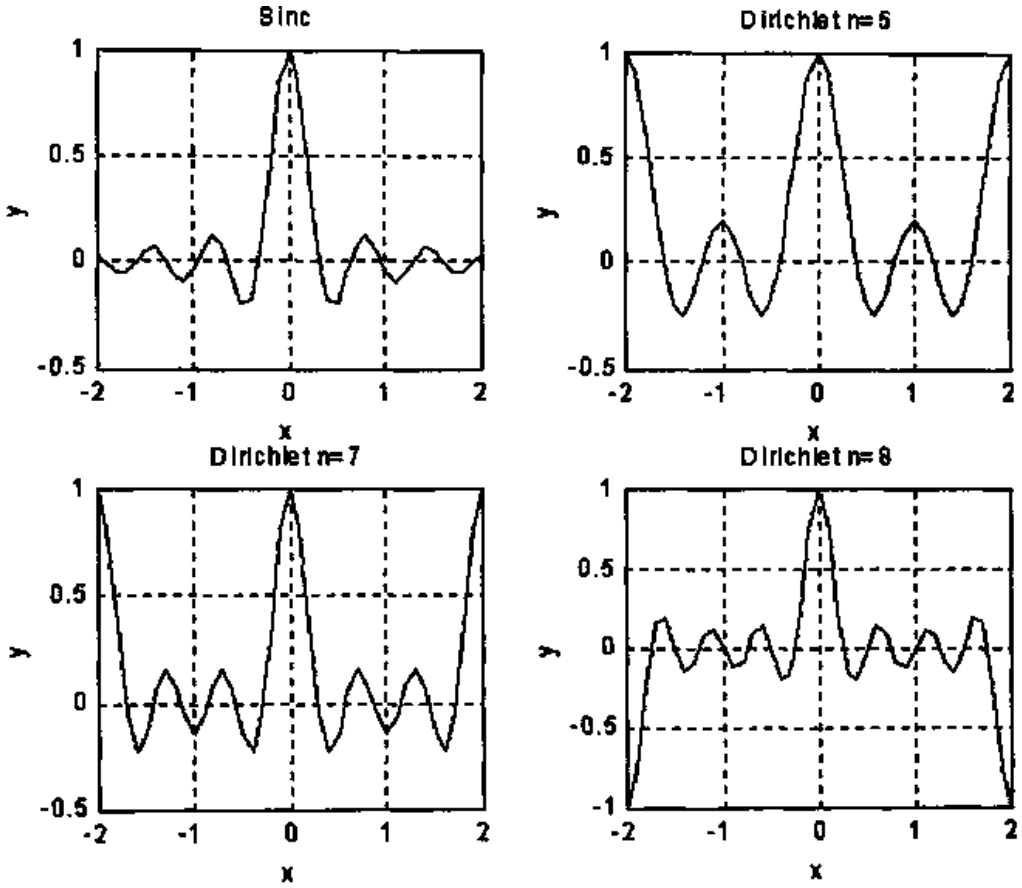


图 1.19　sinc 和 dirichlet 函数曲线

$$y = \text{pulstran}(t, d, 'func', P1, P2)$$

其中，t 为时间向量，d 为脉冲串位置向量(采样点向量)，P1，P2 为脉冲有关参数设置，fun 为脉冲类型函数，MATLAB 提供三种脉冲类型：

- gauspuls 高斯调制正弦脉冲
- rectpuls 非周期矩形脉冲
- tripuls 非周期三角形脉冲

函数 pulstran 执行下面运算

$$y = \text{func}(t - d(1)) + \text{func}(t - d(2)) + \cdots$$

调用后，在向量 d 所指定的位置产生脉冲串。

【例 1.11】 产生锯齿波脉冲串和矩形波脉冲串，脉宽为 0.1s，脉冲重复频率为 3Hz，采样频率 1kHz，信号时间长度为 1s。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 1-12
clf
x=[-2:0.1:2];
y=rectpuls(x);
subplot(221)
plot(x,y);
```

```
title('Rectpuls width=defaul');xlabel('x');ylabel('y');
axis([-2,2,0,1.2]);

x=[-2:0.1:2];
y=rectpuls(x,2);
subplot(222)
plot(x,y);
title('Rectpuls width=2');xlabel('x');ylabel('y');
axis([-2,2,0,1.2]);

t=[0:0.001:2];
d=[0:1/4:1];
y=pulstran(t,d,'tripuls',0.1);
subplot(223)
plot(t,y);
title('Pulstran-tripuls');xlabel('t');ylabel('y');
axis([0,2,0,1.2]);

t=[0:0.001:2];
d=[0:1/4:1];
y=pulstran(t,d,'rectpuls',0.1);
subplot(224)
plot(t,y);
title('Pulstran-rectpuls');xlabel('t');ylabel('y');
axis([0,2,0,1.2]);
```

绘制的波形曲线如图 1.20 所示。

1.5 连续时间系统

由传感器获得的信号一般不能直接使用，往往需要经过处理和加工。信号可通过模拟信号处理装置或模拟滤波器等实现信号变换；或输入计算机，用计算机对信号进行运算、变换、滤波等数字处理过程。无论是模拟信号处理系统，还是数字信号处理系统均涉及信号输入和输出的关系，即必须研究信号处理系统的特性。在模拟信号处理系统中，输入和输出信号都是连续时间信号，处理系统为连续时间系统；在数字信号处理系统中，输入和输出信号是离散的时间序列，处理系统为离散时间系统。本节简要回顾连续时间系统特性。

1.5.1 线性连续时间系统描述

在信号处理中，广泛应用的连续时间系统是线性时不变系统(LTI)。

连续时间系统常用微分方程来描述，可表示为

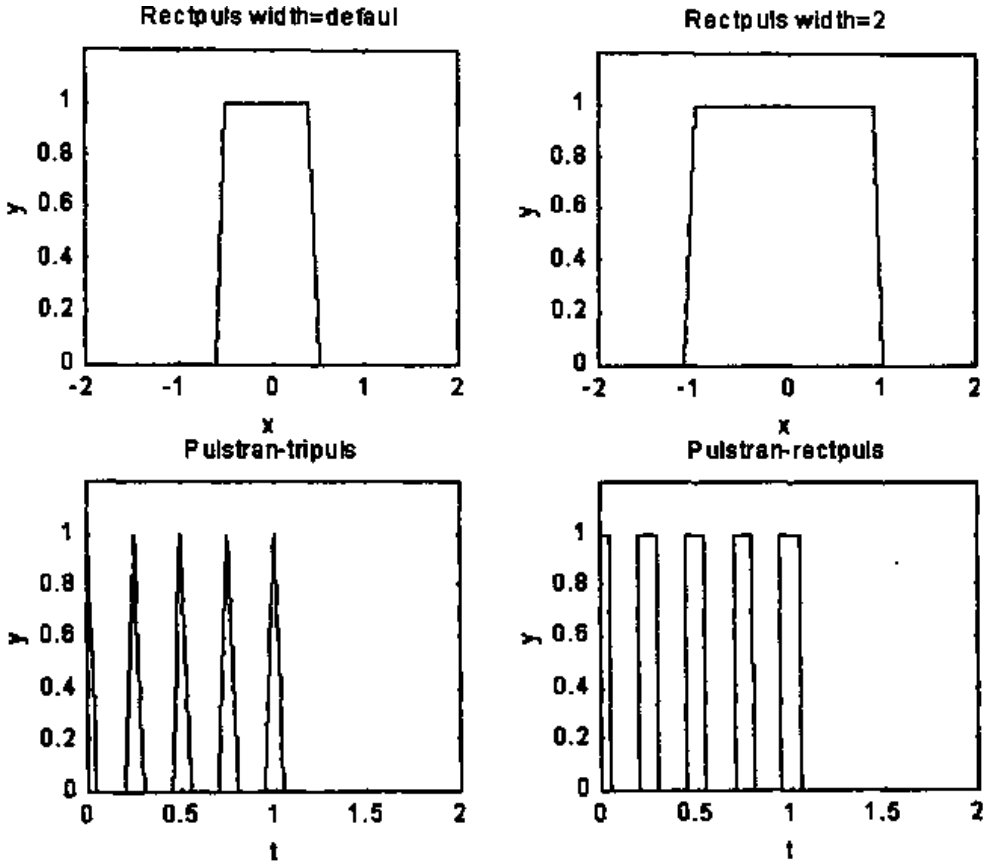


图 1.20 矩形波和非周期脉冲信号串

$$a(1)y^{(na)}(t) + a(2)y^{(na-1)}(t) + \cdots + a(na+1)y(t)$$
$$= b(1)x^{(nb)}(t) + b(2)x^{(nb-1)}(t) + \cdots + b(nb+1)x(t) \qquad (1.5\text{-}1)$$

式中， y(t)和 x(t)分别为系统的输入与输出；$y^{(i)}(t)$ i=1，2，…，na 是系统输出量的各阶导数；$x^{(i)}(t)$ i=1，2，…，nb 是系统输入量的各阶导数；a(1)，…，a(na+1)，b(1)，…，b(nb+1)为常系数。

方程两边进行 Laplace 变换可得

$$H(s) = \frac{Y(s)}{X(s)} = \frac{b(1)s^{(nb)} + b(2)s^{(nb-1)} + \cdots + b(nb+1)}{a(1)s^{(na)} + a(2)s^{(na-1)} + \cdots + a(na+1)} \qquad (1.5\text{-}2)$$

H(s)称为连续时间系统传递函数(或称系统函数)。

线性系统满足叠加原理。若系统输入 $x_i(t)$，(i=1，2，…，n)，系统输入为 $y_i(t)$(i=1，2，…，n)；当系统输入 $x(t) = \sum_{i=1}^{n} a_i x_i(t)$ 时，则系统输出 y(t)可用下式表示

$$y(t) = \sum_{i=1}^{n} a_i y_i(t)$$

1.5.2 脉冲响应函数和系统的时间响应

如图 1.21 所示，把系统输入可分解为由许多 $x(t_i)\Delta t$ 窄条叠加而成。若系统的单位脉冲响应函数为 h(t)，则由输入引起的响应为

$$y(t) \approx \sum_{0}^{t} [x(t_i)\Delta t]h(t - t_i) \qquad (1.5\text{-}3)$$

当 $\Delta t \to 0$，系统的输出

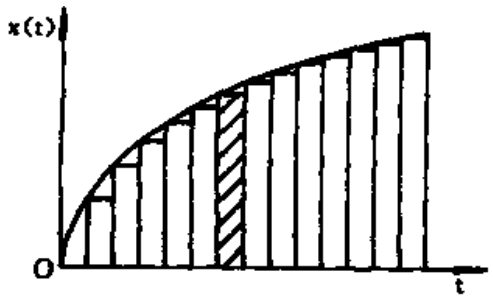


图 1.21 输入信号的分解

$$y(t) = \int_0^t x(t_i)h(t - t_i)dt \tag{1.5-4}$$

或

$$y(t) = x(t) * h(t) \tag{1.5-5}$$

式(1.5-4)说明，线性连续时间系统对任意输入 x(t)的响应 y(t)是输入信号 x(t)与该系统的单位脉冲响应函数 h(t)的卷积。

由控制理论可知，若一个线性系统传递函数为 H(s)，则该系统的单位脉冲响应函数 h(t)可由下式计算

$$h(t) = \mathscr{L}^{-1}[G(s)] \tag{1.5-6}$$

1.5.3 系统的频率响应

线性系统有频率保持特性。如果线性系统的输入为某一频率的正弦信号，如 $x(t) = x_0e^{j\omega t}$，则系统的稳态输出也是同频率的正弦信号，只是相位和幅值上有所变化，即 $y(t) = y_0e^{j(\omega t+\varphi)}$。

由式(1.5-5)，根据卷积定理可得

$$Y(\omega) = X(\omega)H(\omega) \tag{1.5-7}$$

式中，H(ω)是 h(t)的傅里叶变换，一般是一个复数，写为 H(jω)，则

$$H(j\omega) = \frac{Y(\omega)}{X(\omega)} \tag{1.5-8}$$

H(jω)称为系统的频率响应，它是系统在频域内对信号传递特性的描述，即输入信号的各频率成分通过系统进行加工处理，输出信号出现新的特性。

系统的频率响应 H(jω)是一个复数，它可写成如下形式

$$H(j\omega) = Re(\omega) + jI_m(\omega) \tag{1.5-9}$$

式中，Re(ω)为 H(jω)的实部，$I_m(\omega)$为 H(jω)的虚部。

或

$$H(j\omega) = |H(j\omega)|\exp(j\arg[H(j\omega)]) \tag{1.5-10}$$

式中，|H(jω)|为幅频响应，arg[H(jω)]为相频响应。

1.6 离散时间系统

1.6.1 线性离散系统描述

在数字信号处理中，输入信号序列通过离散时间系统进行处理，且主要采用线性时不变系统。离散时间系统常用差分方程来描述，可表示为

$$a(1)y(n)+a(2)y(n-1)+\cdots+a(na+1)y(n-na)$$
$$=b(1)x(n)+b(2)x(n-1)+\cdots+b(nb+1)x(n-nb) \qquad (1.6\text{-}1)$$

式中，y(n)，x(n)分别是系统的输出和输入，y(n)延迟阶数为 na，x(n)延迟阶数为 nb，a(1)，…，a(na+1)；b(1)，…，b(nb+1)为常系数。

方程两边进行 Z 变换可得

$$H(z)=\frac{Y(z)}{X(z)}=\frac{b(1)+b(2)z^{-1}+\cdots+b(nb+1)z^{-nb}}{a(1)+a(2)z^{-1}+\cdots+a(na+1)z^{-na}} \qquad (1.6\text{-}2)$$

H(z)称为脉冲传递函数或 Z 传递函数(离散系统的系统函数)。和连续时间系统一样，线性离散系统也满足叠加原理。

1.6.2 脉冲响应序列和系统时间响应

若系统对单位脉冲序列 δ(n)的响应为 h(n)，对于任意输入 x(n)可表示为

$$x(n)=\sum_{k=-\infty}^{\infty}x(k)\delta(n-k) \qquad (1.6\text{-}3)$$

则线性时不变系统的输出应为

$$y(n)=T[\sum_{k=-\infty}^{\infty}x(k)\delta(n-k)]=\sum_{k=-\infty}^{\infty}x(k)h(n-k)$$

或

$$y(n)=x(n)*h(n) \qquad (1.6\text{-}4)$$

式(1.6-4)说明，线性离散时间系统对任意输入 x(n)的响应 y(n)是输入序列 x(n)和该系统的单位采样响应序列 h(n)的卷积。

若已知系统的传递函数 H(z)，则系统的单位脉冲响应序列

$$h(n)=\mathcal{Z}^{-1}[H(z)] \qquad (1.6\text{-}5)$$

1.6.3 系统频率响应

设输入序列 $x(n)=e^{j\omega n}$，则由式(1.6-4)系统输出应为

$$y(n)=\sum_{k=-\infty}^{\infty}h(k)e^{j(n-k)\omega}=e^{j\omega n}\sum_{k=-\infty}^{\infty}h(k)e^{-j\omega k}$$

令

$$H(e^{j\omega})=\sum_{n=-\infty}^{\infty}h(n)e^{-j\omega n}$$

则

$$Y(e^{j\omega})=e^{j\omega n}H(e^{j\omega})$$

对任意输入 x(n)，同样可得

$$Y(e^{j\omega})=X(e^{j\omega})H(e^{j\omega})$$

$$H(e^{j\omega})=\frac{Y(e^{j\omega})}{X(e^{j\omega})} \qquad (1.6\text{-}6)$$

上式与(1.5-8)式形式上是相同的，$H(e^{j\omega})$为系统的频率响应，它表示线性离散时间系统的特性和连续时间系统相同，系统频率特性 $H(e^{j\omega})$是一个复数，它可写成以下形式

$$H(e^{j\omega})=\mathrm{Re}(e^{j\omega})+\mathrm{Im}(e^{j\omega})=|H(e^{j\omega})|\exp(j\arg[H(e^{j\omega})]) \qquad (1.6\text{-}7)$$

式中，$\mathrm{Re}(e^{j\omega})$和 $\mathrm{Im}(e^{j\omega})$分别为实频响应和虚频响应，$|H(e^{j\omega})|$和 $\arg[H(e^{j\omega})]$分别为的幅频响应和相频响应。

若令 $z=e^{j\omega}$，式(1.6-6)变为

$$H(z)=\frac{Y(z)}{X(z)} \tag{1.6-8}$$

式中,H(z)为系统 z 的传递函数。

1.7 系统的 MATLAB 描述和转换

MATLAB 的信号处理工具箱提供了几种线性时不变系统的模型,我们可以灵活地选择合适的模型以满足不同应用场合的需要。信号处理工具箱提供的系统模型常用于描述滤波器,因此在表示形式上和控制工具箱提供的系统模型有所不同,应注意区分。

1.7.1 离散时间系统

MATLAB 信号处理工具箱提供的离散时间系统模型用于数字滤波器,这些模型包括传递函数、零极点增益形式、状态空间形式、部分分式、二阶级联形式、格状结构和卷积矩阵。

一、传递函数形式

在 MATLAB 信号处理工具箱中,数字滤波器的 z 传递函数具有以下形式:

$$H(z)=\frac{b(1)+b(2)z^{-1}+\cdots+b(nb+1)z^{-nb}}{a(1)+a(2)z^{-1}+\cdots+a(na+1)z^{-na}} \tag{1.7-1}$$

这种形式称为 DSP(Digital Signal Processing)形式,它和控制工具箱的 z 传递函数模型形式不同。

这一模型用 z 传递函数分子和分母多项式的系数构成两个向量来确定,即

$$\text{num}=[b(1)\ b(2)\ \cdots\ b(nb+1)]$$
$$\text{den}=[a(1)\ a(2)\ \cdots\ a(na+1)]$$

式中,nb 和 na 分别为分子 num 和分母 den 的阶数。

在数字信号处理中,要定义一个 DSP 形式传递函数可利用函数 FILT。调用格式为

```
sys = filt(num, den)
sys = filt(num, den, Ts)
```

该函数在 MATLAB 控制系统工具箱内。

二、零极点增益形式

在 MATLAB 信号处理工具箱中,离散时间系统(数字滤波器)的零极点增益模型表达式为

$$H(z)=\frac{q(z)}{p(z)}=K\frac{(z-q(1))(z-q(2))\cdots(z-q(n))}{(z-p(1))(z-p(2))\cdots(z-p(n))} \tag{1.7-2}$$

这一模型用零点向量 q(z)和极点向量 p(z)和增益 K 来确定,即

$$q(z)=[q(1),\ q(2),\ \cdots,\ q(n)]$$
$$p(z)=[p(1),\ p(2),\ \cdots,\ p(n)]$$
$$K$$

MATLAB 函数 poly 和 roots 可实现传递函数形式和零极点增益形式的转换。

【例 1.12】 已知 IIR 滤波器传递函数为

$$H(z)=\frac{2+3z^{-1}+4z^{-2}}{1+3z^{-1}+3z^{-2}+z^{-3}}$$

试确定其零极点表达式。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 1-13
b=[2 3 4];
a=[1 3 3 1];
% transfer function
f=filt(b,a)
% Convert to zero-pole-gain
q=roots(b)
p=roots(a)
k=b(1)/a(1)
% Convert back to transfer function
bb=k * poly(q)
aa=poly(p)

≫smp113
Transfer function:
  2 + 3 z^ -1 + 4 z^ -2
-----------------------
1 + 3 z^ -1 + 3 z^ -2 + z^ -3

Sampling time: unspecified
q =
  -0.7500 + 1.1990i
  -0.7500 - 1.1990i

p =
  -1.0000
  -1.0000 + 0.0000i
  -1.0000 - 0.0000i

k =
     2
bb =
     2.0000     3.0000     4.0000
aa =
     1.0000     3.0000     3.0000     1.0000
```

三、状态空间形式

离散时间系统(数字滤波器)的状态空间表达式为

$$x(n+1) = Ax(n) + Bu(n) \qquad (1.7\text{-}3)$$
$$y(n) = Cx(n) + Du(n)$$

式中,u 为输入,y 为输出,x 为状态向量。

在 MATLAB 信号处理工具箱中,用矩阵 A、B、C、D 表示数字滤波器模型(和控制系统工具箱一样)。

MATLAB 信号处理工具箱还提供传递函数、零极点和状态空间三种模型之间的相互转换函数,它们是 ZP2FT、ZP2SS、TF2SS、TF2ZP、SS2TF 和 SS2ZP。

```
[Z, P, K] = tf2zp(Num, den)
[num, den] = zp2tf(Z, P, K)
[A, B, C, D] = tf2ss(num, den)
[num, den] = ss2tf(A, B, C, D, iu)
[A, B, C, D] = zp2ss(Z, P, K)
[Z, P, K] = ss2zp(A, B, C, D, iu)
```

【例 1.13】 已知滤波器的传递函数

$$H(z) = \frac{2 + 3z^{-1}}{1 + 0.4z^{-1} + z^{-2}}$$

试求该滤波器的零极点和状态空间表达式。

用 MATLAB 编写程序如下:

```
%MATLAB PROGRAM 1-14
%Input parameters
num=[2  3];
den=[1  0.4  1];
f=filt(num,den)
disp(' Convert to zero-pole-gain ')
[z,p,k]=tf2zp(num,den)
disp(' Convert to state space')
[A,B,C,D]=tf2ss(num,den)

≫smp114
Transfer function:
    2 + 3 z^ -1
-----------------
1 + 0.4 z^ -1 + z^ -2

Sampling time: unspecified
Convert to zero-pole-gain
```

```
z =
   -1.5000
p =
   -0.2000 + 0.9798i
   -0.2000 - 0.9798i
k =
    2
Convert to state space
A =
   -0.4000   -1.0000
    1.0000         0
B =
    1
    0
C =
    2    3
D =
    0
```

四、部分分式展开式

离散时间系统(数字滤波器)的传递函数 H(z)也可用下面部分分式展开式来描述：

(1) 若 H(z)不包含重极点

$$H(z)=\frac{b(z)}{a(z)}=\frac{r(1)}{1-p(1)z^{-1}}+\frac{r(2)}{1-p(2)z^{-1}}+\cdots+\frac{r(n)}{1-p(n)z^{-1}}+k(1)+k(2)z^{-1}+\cdots+k(m-n+1)z^{-(m-n)} \quad (1.7\text{-}4)$$

(2) 若 H(z)包含重极点，如 p(j)有 Sr 个重极点，H(z)部分分式展开式中包含相应的形式为：

$$\frac{r(j)}{1-p(j)z^{-1}}+\frac{r(j+1)}{(1-p(j)z^{-1})^2}+\cdots+\frac{r(j+Sr-1)}{(1-p(j)z^{-1})^{Sr}} \quad (1.7\text{-}5)$$

在 MATLAB 信号处理工具箱中，函数 RESIDUEZ( )用来把传递函数形式转变为部分分式展开式。

调用格式为

```
[r, p, k] = residuez(b, a)
```

其中， b，a 分别为 z 传递函数的分子、分母；p 为传递函数极点向量；r 为相应极点的留数向量；k 为传递函数商向量。若原传递函数为真分数，此项为空向量。

RESIDUEZ 也可用于将部分分式展开式转换为传递函数形式，调用格式为

```
[b, a] = residuz(r, p, k)
```

【例 1.14】 已知系统传递函数为

$$H(z)=\frac{-4+8z^{-1}}{1+6z^{-1}+8z^{-2}}$$

求其部分分式展开式。

用 MATLAB5 编写程序如下：

```
%MATLAB PROGRAM 1-15
% Input Parameters of model
b=[-4 8];
a=[1 6 8];
%transform partial--fraction expansion
[r,p,k]=residuez(b,a)

≫smp115
r =
    -12
     8
p =
    -4
    -2
k =
    [ ]
```

即
$$H(z)=-\frac{12}{1+4z^{-1}}+\frac{8}{1+2z^{-1}}$$

MATLAB 信号处理工具箱中引入系统的部分分式展开式，利用公式直接求传递函数 H(z)的 z 反变换。

如上例中，H(z)的反变换为 h(n)，

$$h(n)=-12(-4)^n+8(-2)^n,\quad n=0,1,2,\cdots$$

五、二阶因子级联形式

任何传递函数 H(z)可写为两阶因子级联形式，即

$$H(z)=\prod_{k=1}^{L}H_k(z)=\prod_{k=1}^{L}\frac{b_{0k}+b_{1k}zh^{-1}+b_{2k}z^{-2}}{a_{0k}+a_{1k}z^{-1}+a_{2k}z^{-2}} \tag{1.7-6}$$

即 H(z)可化成多个两阶因式相乘，L 为两阶因式的个数。这样，一个离散时间系统可用一个 L×6 的数组 SOS 表示，每一行表示一个两阶因子，分子三个系数和分母三个系数。

$$SOS=\begin{bmatrix} b_{01} & b_{11} & b_{21} & a_{01} & a_{11} & a_{21} \\ b_{02} & b_{12} & b_{22} & a_{02} & a_{12} & a_{22} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{0L} & b_{1L} & b_{2L} & a_{0L} & a_{1L} & a_{2L} \end{bmatrix} \tag{1.7-7}$$

这种形式用于滤波器的二阶环节级联结构中。

MATLAB 信号处理工具箱中，提供了一般模型和 SOS 形式的相互转换函数 ZP2SOS、SOS2ZP、SS2SOS、SOS2SS 和 SOS2TF。

函数调用格式如下：

```
sos = zp2sos(z, p, k)
[z, p, k] = sos2zp(sos)
sos = ss2sos(A, B, C, D)
[A, B, C, D] = sos2ss(sos)
[num, den] = sos2tf(sos)
```

【例 1.15】 已知滤波器的二阶级形式为

$$SOS = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & -1 \\ -2 & 3 & 1 & 1 & 10 & 1 \end{bmatrix}$$

确定其状态空间表达式和零极点形式。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 1-16
% Input Parameters of model
sos=[1 1 1 1 0 -1;-2 3 1 1 10 1];
[num,den]=sos2tf(sos);
f=filt(num,den)
disp('convert to zero-pole form')
[z,p,k]=sos2zp(sos)

≫smp116

Transfer function:
-2 + z^ -1 + 2 z^ -2 + 4 z^ -3 + z^ -4
---------------------------------------
  1 + 10 z^ -1 - 10 z^ -3 - z^ -4

Sampling time: unspecified
convert to zero-pole form

z =
  -0.5000 + 0.8660i
  -0.5000 - 0.8660i
   1.7808
  -0.2808
p =
  -1.0000
   1.0000
  -9.8990
```

```
    -0.1010
k =
    -2
```

六、格状结构

滤波器的实现通常采用格状结构。

对于 N 阶全零点或全极点滤波器采用格型结构。可以用多项式 a(n)，n = 1，2，…，N 来描述，也可通过格型结构反射系数 k(n)，n = 1，2，…，N 来描述。

对于一般 IIR 滤波器采用格型/梯形结构，它既包含零点，也包含极点。因此，可通过传递函数分子多项式 b 和分母多项式 a 来描述，也可以通过和分母多项式 a 对应的格型反射系数 k(n)及分子多项式 b 对应的梯形系数 v(n)来描述。

总之，对于格状结构滤波器的描述可用系数 k(n)和 c(n)来描述。

MATLAB 信号处理工具箱提供了滤波器格状结构转换函数 LATCFILT、POLY2RC、RC2POLY、LATC2TF 和 TF2LATC。请阅读第五章有关内容。

【例 1.16】 把下面传递函数形式的 IIR 滤波器转变为格型/梯形结构。

$$H(z)=\frac{1+2z^{-1}+2z^{-2}+z^{-3}}{1+\frac{13}{24}z^{-1}+\frac{5}{8}z^{-2}+\frac{1}{3}z^{-3}}$$

用 MATLAB5 编写程序如下：

```
%MATLAB PROGRAM 1-17
% Input Parameters of model
num=[1 2 2 1];
den=[1 13/24 5/8 1/3];
%Transfer function to lattice filter conversion
[k,v]=tf2latc(num,den)

≫smp117
k =
  0.2500
  0.5000
  0.3333

v =
  -0.2695
   0.8281
   1.4583
   1.0000
```

七、卷积矩阵

若有两个向量 h 和 x，满足下面等式的矩阵 C 称为卷积矩阵

$$C \cdot x = h * x \tag{1.7-8}$$

由式(1.6-5)可见，当滤波器的单位采样响应序列为 h(n)，滤波器输入为 x(n)，滤波器输出信号为

$$y(n) = h * x = C \cdot x \tag{1.7-9}$$

可见，卷积矩阵是数字滤波器的另一种描述形式。

MATLAB 信号处理工具箱提供了计算卷积矩阵函数 CONVMTX。调用格式为

C = convmtx(h, n)

式中，h 为 m×1 列向量；n 为 x 列向量维数；c 为(m+n−1)×n 阶卷积矩阵。

【例 1.17】 已知 h = [1 2 3 2]，x 为 4×1 随机序列，求其卷积矩阵。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 1-18
% Input Parameters of model
h=[1 2 3 2];
n=4;
randn('seed',0);
x=randn(n,1);
% Find convolution matrix
C=convmtx(h',n)
y1=C*x
y2=conv(h,x)

≫smp118

C =

     1     0     0     0
     2     1     0     0
     3     2     1     0
     2     3     2     1
     0     2     3     2
     0     0     2     3
     0     0     0     2

y1 =
    1.1650
    2.9567
    4.8236
    4.7122
```

```
    2.1821
    1.2050
    0.7032

y2 =
    1.1650
    2.9567
    4.8236
    4.7122
    2.1821
    1.2050
    0.7032
```

程序运行结果，证实式(1.7-9)所定义的卷积矩阵的概念。

1.7.2 连续时间系统

连续时间系统模型用来描述模拟滤波器。连续时间系统可以用下面几种形式来描述：

(1) 状态空间表达式

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{1.7-10}$$

(2) 传递函数

$$H(s) = \frac{b(s)}{a(s)} = \frac{b(1)s^{nb} + b(2)s^{nb-1} + \cdots + b(nb+1)}{a(1)s^{na} + a(2)s^{na-1} + \cdots + a(na+1)} \tag{1.7-11}$$

(3) 零极点增益

$$H(s) = \frac{z(s)}{p(s)} = k\frac{(s-z(1))(s-z(2))\cdots(s-z(m))}{(s-p(1))(s-p(2))\cdots(s-p(n))} \tag{1.7-12}$$

(4) 部分分式展开式

部分分式展开式和式(1.4-23)、(1.4-24)形式类似。

在离散时间系统中所述的有关 MATLAB 函数也适用于连续时间系统。

1.8 卷积及时域响应的求解

由式(1.5-5)和式(1.6-4)可知，卷积运算可用于计算线性系统的时间响应，因此卷积运算在信号处理中是十分重要的。

对于序列卷积

$$y(n) = x(n) * h(n) = \sum_{k=0}^{N} x(k)h(n-k) \tag{1.7-13}$$

MATLAB 提供卷积计算的函数有 CONV、CONV2 和 CONVN。

函数 CONV 用于计算向量卷积和多项式乘。调用格式为

```
c = conv(a, b)
```

其中，a，b 为两个序列，c= a * b.

若向量 a 的长度为 na，向量 b 的长度为 nb，则向量 c 的长度 nc = na+nb−1。

函数 CONV2 用于两维卷积，CONVN 用于 n 维卷积运算。

求离散系统的时间响应，除了用卷积运算外，还可以用 MATLAB 函数 filter。函数 filter 调用格式见第五章。

应用函数 filter 求系统对任意输入 x 的响应 y 的方法有两种：一是直接利用 filter 函数求 x 的响应 y；二是利用卷积理论，先用 filter 求系统对单位脉冲响应 h(n)，再应用卷积 y(n) = conv(x(n), h(n))。下面给出一个例子说明两种方法结果相同，读者可加深对卷积定理的理解。

【例 1.18】 已知滤波器的传递函数

$$H(z) = \frac{0.15}{1 - 0.8z^{-1}}$$

输入信号为 x(t) = 2sin(0.05πt)+w(t)，w(t)为随机信号，幅值为 0.2。试绘出滤波器的输出信号波形。

用两种方法求滤波器输出，并用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 1-19
% Model of the filter
b=0.15;
a=[1 -0.8];
% Create input of filter
n=[0:100];
x=2*sin(0.05*pi*n)+0.2*randn(1,101);
%Calculate response according to convolution
imp=[1;zeros(100,1)];
h=filter(b,a,imp);
yc=conv(h,x);
y=yc(1:101);
%Calculate response according to matlab function
y1=filter(b,a,x);
plot(n,x,'r',n,y1,'b',n,y,'m');
xlabel('n');ylabel('x y yc');title('Time response');
grid
```

滤波器输入、输出波形如图 1.22 所示。

由图 1.18 可见，用卷积原理计算的输出和用函数 filter 计算的输出完全吻合。本例中，输入 x 是 1×101 维向量，h 也是 1×101 维向量，利用函数 conv 求得 yc 为 1×202 维向量，只有前 1×101 维向量可用来描述响应 y.

函数 conv 用于计算两个矢量的卷积，这里假定两个序列 a 和 b 都是从 n=0 开始的。如果两个序列从某一负值开始，则不能直接采用 conv 函数计算卷积。

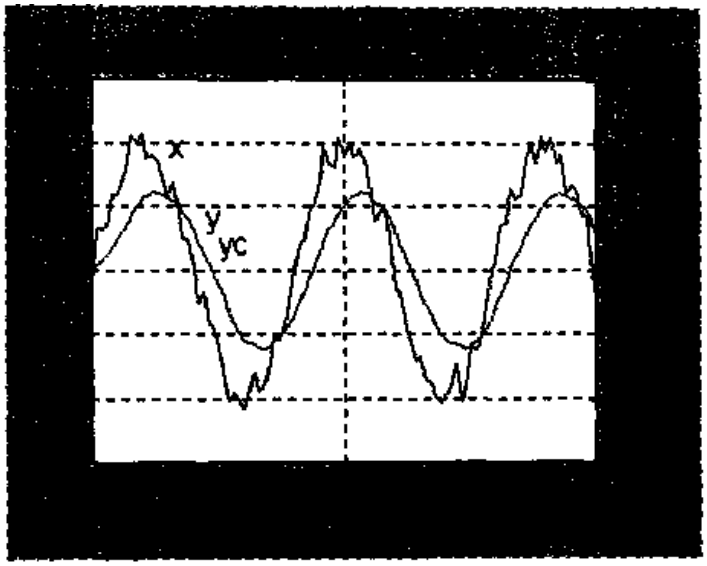


图 1.22　滤波器输入和输出曲线

习　题

1.1　用 MATLAB 绘制下列各连续时间信号的波形图。

(1) $te^{-t}u(t)$

(2) $(1+\cos\pi t)[u(t)-u(t-2)]$

(3) $\dfrac{\sin\alpha(t-t_0)}{\alpha(t-t_0)}$

(4) $\sin 2\pi f_1 t+\cos 2\pi f_2 t+\omega(t)$

其中，$f_1=25\text{Hz}$，$f_2=100\text{Hz}$，$\omega(t)$为白噪声信号。

1.2　绘制下列各时间序列的波形图。

(1) $x(n)=\left(\dfrac{1}{2}\right)^n u(n)$

(2) $x(n)=2^n u(n)$

(3) $x(n)=\left(-\dfrac{1}{2}\right)^n u(n)$

(4) $x(n)=(-2)^n u(n)$

(5) $x(n)=2^{n-1}u(n-1)$

(6) $x(n)=\left(\dfrac{1}{2}\right)^{n-1} u(n)$

1.3　绘制下列各序列的波形图并检查其周期性。

(1) $x(n)=\sin\left(\dfrac{n}{10}\pi-\dfrac{\pi}{4}\right)$

(2) $x(n)=0.5n+\cos\left(\dfrac{5\pi}{7}n\right)$

(3) $x(n)=e^{j\left(\frac{n}{8}-\pi\right)}$

1.4　用 MATLAB 生成复指数序列

$$x(n)=e^{(0.1-j0.3)n}$$

绘制序列的实部图、虚部图、幅值图、相角图。

1.5 绘制序列 $h(n)=2\delta(n)-\delta(n-1)$ 波形图。

1.6 已知序列 $n=\{-4, -3, -2, -1, 0, 1\}$，$x(n)=\{-6, -4, -2, 0, 2, 4\}$，试产生并绘制下面样本序列：

(1) $x_1(n)=2x(n-2)+3x(n)+4x(n+2)$

(2) $x_2(n)=2x(2-n)+3x(-n)+4x(-2-n)$

(3) $x_3(n)=e^{-0.1n}x(n-1)n$

1.7 (1) 已知序列 $x_1(n)=\{1, 2, 3, 4, 5\}$，$x_2(n)=\{2, 3, 4\}$，求 $x_1(n)\cdot x_2(n)$。

$$\underset{n=0}{\uparrow} \qquad\qquad \underset{n=0}{\uparrow}$$

(2) 已知序列 $x_1(n)=\{1, 2, 3, 4, 5\}$，$x_2(n)=\{2, 3, 4\}$，求 $x_1(n)\cdot x_2(n)$。

$$\underset{n=0}{\uparrow} \qquad\qquad \underset{n=0}{\uparrow}$$

1.8 用 MATLAB 函数产生周期为 1s，幅值为 ± 2 的方波和正三角波，并绘制其波形图。

1.9 用 MATLAB 函数产生一个矩形脉冲串，连续时间为 2s，脉冲个数为 10，并绘制其波形图。

1.10 已知低通数字滤波器系统函数

$$H(z)=\frac{0.0186+0.0743z^{-1}+0.1114z^{-2}+0.0743z^{-3}+0.0186z^{-4}}{1-1.5704z^{-1}+1.2756z^{-2}-0.4844z^{-3}+0.0762z^{-4}}$$

(1) 试求部分分式展开形式；

(2) 两阶因式级联形式。

1.11 计算下面序列的卷积：

(1) $x_1(n)=\left(\frac{1}{2}\right)^n u(n)$，$x_2(n)=u(n)$

(2) $x_1(n)=u(n-1)-u(n-2)$，$x_2(n)=u(n+2)+u(n+3)$

(3) $x_1(n)=\left(\frac{1}{2}\right)^n u(n)$，$x_2(n)=\left(\frac{1}{3}\right)^n u(n)$

1.12 已知滤波器传递函数

$$H(z)=\frac{0.1311+0.2622z^{-1}+0.1311z^{-2}}{1-0.7478z^{-1}+0.2722z^{-2}}$$

(1) 求滤波器的单位脉冲响应 $h(n)$，并绘制响应曲线；

(2) 系统输入为 $x(n)=\left(\frac{1}{2}\right)^n u(n)+0.1\sin(\pi n)$ 时，利用卷积原理，求滤波器输出并绘制输出曲线；

(3) 系统输入为 $x(n)=\left(\frac{1}{2}\right)^n u(n)+\sin(0.5\pi n)$，直接利用函数 filter 求滤波器输出并绘制输出曲线；

(4) 比较(2)和(3)的结果，分析输出曲线不同的原因。

第二章　信号变换和调制

傅里叶变换和 z 变换是信号系统分析中两个基本的数学工具。通过信号的傅里叶变换，可实现信号的频谱分析，掌握信号频域特征，达到提取信号中有用信息的目的。作为信号分析的有力工具，傅里叶变换已广泛地应用于通讯工程、自动测量和控制、机械振动研究、语言处理、图像处理等一系列工程技术领域。随着计算机技术的发展，离散傅氏变换(DFT)和快速傅氏变换(FFT)技术也在不断完善和发展。

在连续时间信号和系统中，通过拉普拉斯变换将时域的微分方程转换为复频域的代数方程，而且拉氏变换被看成是傅里叶变换的推广。在离散时间信号和系统中，z 变换把离散系统时域方程——差分方程转变为较简单的代数方程，z 变换也可看成是离散傅里叶变换的扩展，z 变换成为分析离散信号和系统的重要工具。

本章首先简要介绍连续的傅里叶变换，然后重点介绍离散的傅里叶变换(DFT)和快速傅里叶变换(FFT)及其 MATLAB 实现。接着介绍 z 变换和它的应用。

信号调制和解调是对信号按一定方式变换的过程，在信号传输、工程测试和通信中得到了广泛应用。本章也结合 MATLAB 函数作简要介绍。

2.1　连续信号的傅里叶变换

2.1.1　周期信号的频谱分析——傅里叶级数

周期信号是按一定时间间隔(周期)不断重复的信号。

设周期信号 x(t)，重复周期 T，角频率 $\omega_0=2\pi/T=2\pi f$，且满足狄里赫里条件，都可以展开成傅里叶级数，傅里叶级数表达式有三角形式和指数形式两种。

三角形式的傅里叶级数表达式为：

$$x(t)=a_0+\sum_{n=1}^{\infty}(a_n\cos n\omega_0 t+b_n\sin n\omega_0 t),\qquad n=1,2,3,\cdots \tag{2.1-1}$$

式中

$$a_0=\frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}}x(t)dt$$

$$a_n=\frac{2}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}}x(t)\cos n\omega_0 t dt$$

$$b_n=\frac{2}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}}x(t)\sin n\omega_0 t dt$$

或

$$x(t)=a_0+\sum_{n=1}^{\infty}A_n\sin(n\omega_0 t+\varphi_n) \tag{2.1-2}$$

式中

$$A_n=\sqrt{a_n^2+b_n^2}$$

$$\varphi_n=\arctan\frac{a_n}{b_n}$$

指数形式的傅里叶级数为

$$x(t)=\sum_{n=-\infty}^{\infty}C_n e^{jn\omega_0 t},\qquad n=0,\pm1,\pm2,\cdots \tag{2.1-3}$$

式中

$$C_n=\frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}}x(t)e^{-jn\omega_0 t}dt$$

在一般情况下，C_n 为复数，可以写成

$$C_n=C_{nR}+jC_{nj}=|C_n|e^{j\varphi_n}$$

C_n 与 C_{-n} 共轭。

周期信号的频谱具有三个特点：

(1) 离散性，即谱线是离散的；

(2) 谱波性，即谱线只出现在基波频率的整数倍上；

(3) 收敛性，即谱波的幅度随谱波次数的增高而减小。

2.1.2 非周期信号的频谱分析——傅里叶变换

非周期信号的频谱是连续的，也具有收敛特性。用周期信号的傅里叶级数通过极限的方法导出非周期信号的频谱表达式称为傅里叶变换。

傅里叶变换式：

$$F(\omega)=\mathscr{F}[f(t)]=\int_{-\infty}^{\infty}f(t)e^{-j\omega t}dt \tag{2.1-4}$$

傅里叶反变换式：

$$f(t)=\mathscr{F}^{-1}[F(\omega)]=\frac{1}{2\pi}\int_{-\infty}^{\infty}F(\omega)e^{j\omega t}dt \tag{2.1-5}$$

$F(\omega)$ 是 $f(t)$ 的频谱密度函数，一般也为复数，故可写为

$$F(\omega)=Re(\omega)+jIm(\omega)=|F(\omega)|e^{j\varphi(\omega)} \tag{2.1-6}$$

式中，$|F(\omega)|$ 称为 $f(t)$ 的幅值谱密度(常称幅值谱)，$\varphi(\omega)$ 称为 $f(t)$ 的相位谱密度(常称相位谱)。

应该指出的是，周期信号的幅值谱描述各谐振分量的幅值，而非周期信号的幅值谱和相位谱描述幅值谱密度(单位：振幅/频率)和相位谱密度(单位：弧度/频率)。

图 2.1 绘出了几种典型信号的频谱。

(1) 矩形脉冲信号

$$f(t)=\begin{cases}E, & |t|\leqslant\dfrac{\tau}{2}\\[2ex] 0, & |t|>\dfrac{\tau}{2}\end{cases}$$

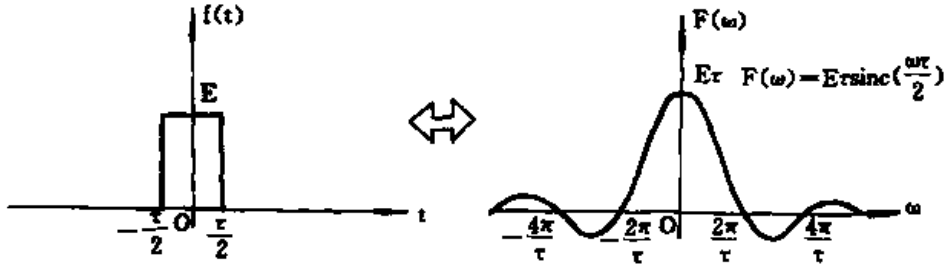
(2) 周期矩形信号

重复为 T_1，$\omega_1=2\pi f_1=\dfrac{2\pi}{T_1}$，在一个周期内$\left(-\dfrac{\tau}{2}\leqslant t\leqslant\dfrac{\tau}{2}\right)$的表达式为

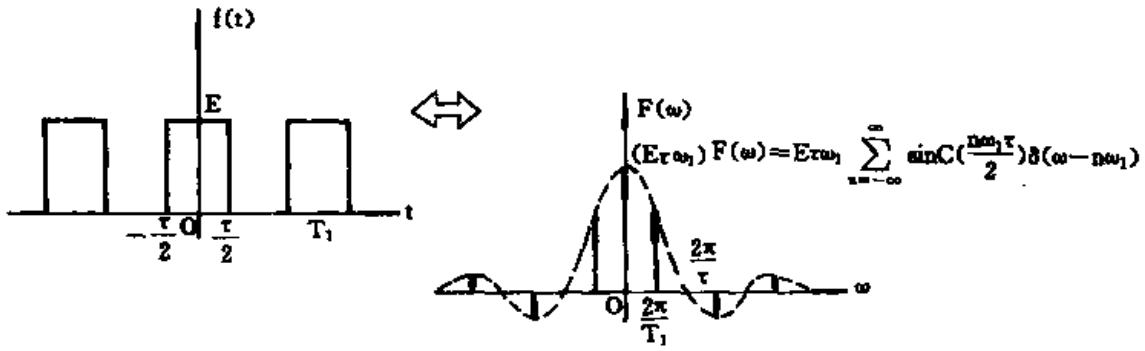
$$f(t)=\begin{cases}E, & |t|\leqslant\dfrac{\tau}{2}\\[2ex] 0, & |t|>\dfrac{\tau}{2}\end{cases}$$

(3) 周期脉冲信号

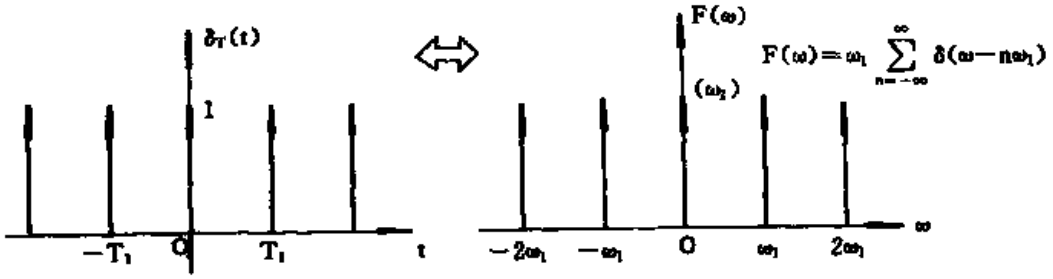
$$\delta_T(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_1), \qquad n = 0, \pm 1, \pm 2, \cdots$$



(a)矩形脉冲信号



(b)周期矩形信号



(c)周期脉冲信号

图 2.1 几种典型信号频谱

式中，T_1 为周期，$\omega_1 = 2\pi/T_1$.

2.1.3 傅里叶变换的基本性质

傅里叶变换具以有下基本性质：

$$f(t)\leftrightarrow F(\omega),\ f_1(t)\leftrightarrow F_1(\omega),\ f_2(t)\leftrightarrow F_2(\omega)$$

(1) 线性特性

$$a_1f_1(t)+a_2f_2(t)\leftrightarrow a_1F_1(\omega)+a_2F_2(\omega)$$

(2) 时移特性：

$$f(t\pm t_0)\leftrightarrow F(\omega)e^{\pm j\omega t_0}$$

(3) 频移特性

$$f(t)e^{\mp j\omega_0 t}\leftrightarrow F(\omega\pm\omega_0)$$

(4) 时间尺度变换特性

$$f(t)\leftrightarrow\frac{1}{|a|}F\left(\frac{\omega}{a}\right)$$

(5) 对称特性

$$F(t)\leftrightarrow 2\pi f(-\omega)$$

(6) 时域微分

$$f^{(n)}(t)\leftrightarrow(j\omega)^nF(\omega)$$

(7) 频域微分

$$F^{(n)}(\omega)\leftrightarrow(jt)^nf(t)$$

(8) 积分特性

$$\int_{-\infty}^{t}f(\tau)d\tau\leftrightarrow\frac{F(\omega)}{j\omega}+\pi F(0)\delta(\omega)$$

(9) 时域卷积

$$f_1(t)*f_2(t)\leftrightarrow F_1(\omega)\cdot F_2(\omega)$$

(10) 频域卷积

$$f_1(t)\cdot f_2(t)\leftrightarrow\frac{1}{2\pi}(F_1(\omega)*F_2(\omega))$$

2.1.4 采样信号的傅里叶变换

一、时域采样

时域采样利用周期脉冲序列 p(t)从连续信号 f(t)获得一系列的离散值。这种方法所得到的离散信号称为采样信号。

采样信号的傅里叶变换可由以下方法求出。

令连续信号 f(t)的傅里叶变换为 $F(\omega)=\mathscr{F}[f(t)]$，采样的脉冲序列 p(t)为单位脉冲序列，采样周期 T_s 为

$$p(t)=\delta_T(t)=\sum_{n=-\infty}^{\infty}\delta(t-nT_s)$$

采样信号

$$f_s=f(t)\delta_T(t)$$

由傅里叶变换公式可求得采样信号 $f_s(t)$的频谱为

$$F_s(\omega) = \frac{1}{T_s}\sum_{n=-\infty}^{\infty} F(\omega - n\omega_s) \tag{2.1-7}$$

图 2.2 为采样信号的频谱图。

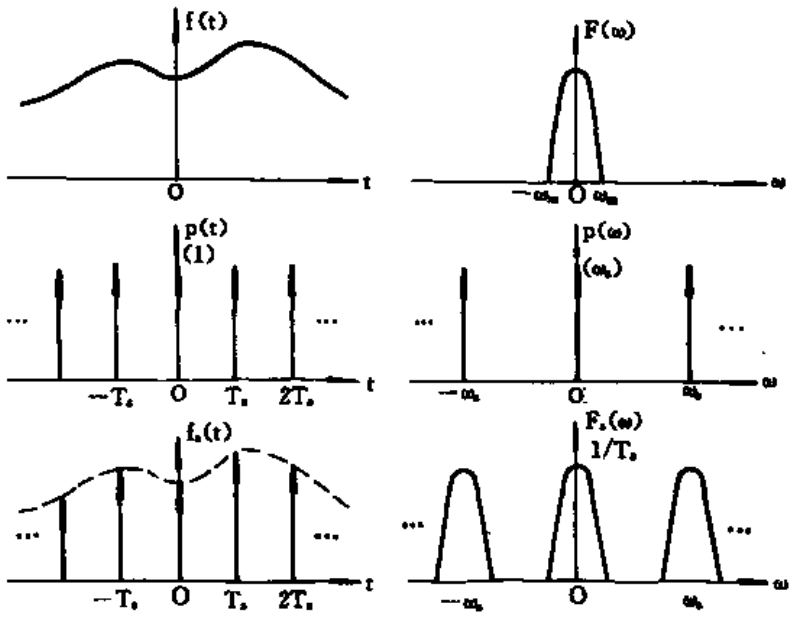


图 2.2　采样信号频谱

由上可见，连续信号经时域采样之后为离散信号，新的离散信号的频谱是周期性的，周期为 T_s。若模拟信号最高频率为 ω_m，只要满足采样频率 $\omega_s > 2\omega_m$，$F_s(\omega)$不会产生频率混叠，这就是时域采样定理。通常把最低允许的采样频率 $\omega_s = 2\omega_m$ 称为奈奎斯特频率，把最大允许的采样间隔 $T_s = \pi/\omega_s = 1/2(f_m)$称为奈奎斯特间隔。

信号在时域被采样后，它的频谱 $F_s(\omega)$是连续信号的频谱 $F(\omega)$以采样周期 T_s 为间隔周期地重复而得到。借助无混叠频谱，我们可以利用矩形窗口函数将原信号的频谱无失真地取出，并恢复原模拟信号 $f(t)$。

二、频域采样

已知连续频谱函数 $F(\omega)$，对应的时间函数为 $f(t)$。若 $F(\omega)$与频域中被间隔为 ω_1 的脉冲序列 $\delta_s(\omega)$所采样，采样后的频谱函数 $F_1(\omega)$为

$$F_1(\omega) = F(\omega) \cdot \delta_s(\omega) = F(\omega)\sum_{n=-\infty}^{\infty}\delta(\omega - n\omega_1)$$

由傅里叶逆变换可求得 $F_1(\omega)$所对应的时间函数

$$f_1(t) = \frac{1}{\omega}\sum_{n=-\infty}^{\infty} f(t - nT_1) \tag{2.1-8}$$

图 2.3 为采样频谱函数对应的时间函数 $f_1(t)$波形。

可见，若 $f(t)$的频谱 $F(\omega)$被间隔为 ω_1 的脉冲序列在频域中采样，则在时域中得到一个周期信号，且为 $f(t)$以 $T_1 = 2\pi/\omega_1$ 为周期而重复构成。周期信号的频谱是离散的。

由上可见，只要频域采样间隔不大于 $1/(2t_m)$，则在时域中波形不会产生混叠，这就是频域采样定理。用矩形窗函数作为叠通信号就可以无失真地恢复其原信号 $f(t)$。

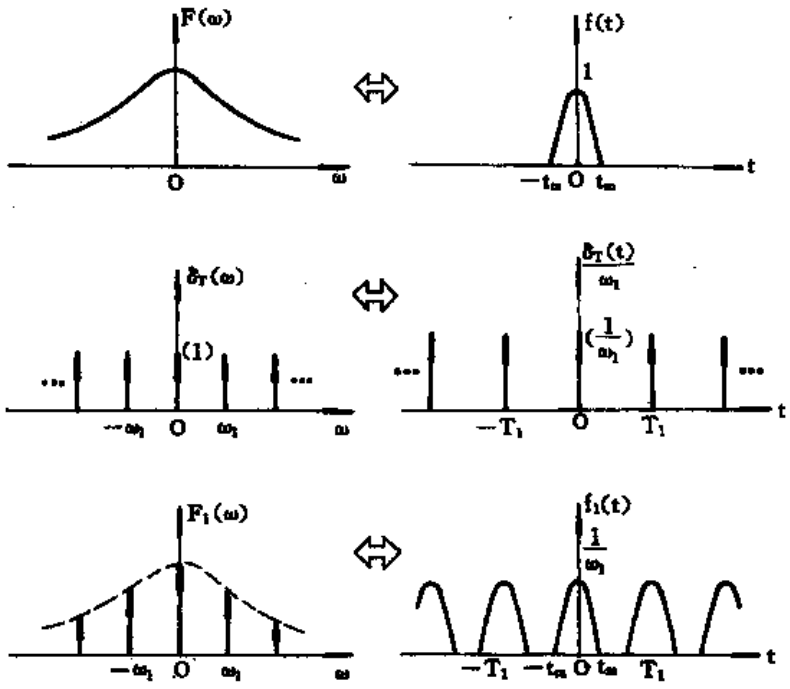


图 2.3　频率采样所对应的信号波形

2.2　离散信号的傅里叶变换

2.2.1　周期序列——离散傅里叶级数

第一章已述，离散信号可用序列 x(n)表示。若周期序列

$$\tilde{x}(n) = \tilde{x}(n + kN) \tag{2.2-1}$$

也可表示为

$$\tilde{x}(n) = \frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}(k)e^{j\frac{2\pi}{N}kn}, \qquad n = 0, \pm 1, \cdots \tag{2.2-2}$$

式中，

$$\tilde{X}(k) = \sum_{n=0}^{N-1}\tilde{x}(n)e^{-j\frac{2\pi}{N}nk}, \qquad k = 0, \pm 1, \cdots \tag{2.2-3}$$

式(2.2-2)和(2.2-3)称为周期序列的离散傅里叶级数表示。

令 $W_N \triangleq e^{-j\frac{2\pi}{N}}$，则

$$\tilde{X}(k) = DFS[\tilde{x}(n)] = \sum_{n=0}^{N-1}\tilde{x}(n)W_n^{nk}, \qquad k = 0, \pm 1, \pm 2\cdots \tag{2.2-4}$$

$$\tilde{x}(k) = IDFS[\tilde{X}(k)] = \frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}(k)W_n^{-nk}, \qquad n = 0, \pm 1, \pm 2\cdots \tag{2.2-5}$$

离散傅里叶级数系数 $\tilde{X}(k)$在频域上仍然是一个周期为 N 的周期序列。

$$\tilde{X}(k) = \tilde{X}(k + N) \tag{2.2-6}$$

【例 2.1】 求下面周期序列的 DFS 表达式

$$\tilde{x}(n) = \{0,1,2,3,0,1,2,3,0,1,2,3,0,1,2,3,\cdots\}$$

解　上述序列基本周期 N=4，因此

$$W_4 = e^{-j\frac{2\pi}{4}} = e^{-j\frac{\pi}{2}} = \cos\frac{\pi}{2} - j\sin\frac{\pi}{2} = -j$$

$$\tilde{X}(k) = \sum_{0}^{3}\tilde{x}(n)W_4^{nk}, \quad k = 0, \pm 1, \pm 2, \cdots$$

$$\tilde{X}(0) = \sum_{n=0}^{3}\tilde{x}(n)W_4^{0\cdot n} = \sum_{n=0}^{3}\tilde{x}(n)$$
$$= \tilde{x}(0) + \tilde{x}(1) + \tilde{x}(2) + \tilde{x}(3) = 6$$

$$\tilde{X}(1) = \sum_{n=0}^{3}\tilde{x}(n)W_4^{1\cdot n} = \sum_{n=0}^{3}\tilde{x}(n)(-j)^n$$
$$= 0 + 1\cdot(-j) + 2\cdot(-j)^2 + 3\cdot(-j)^3 = -2 + 2j$$

$$\tilde{X}(2) = \sum_{n=0}^{3}\tilde{x}(n)W_4^{2\cdot n} = \sum_{n=0}^{3}\tilde{x}(n)(-j)^{2n} = 2$$

$$\tilde{X}(3) = \sum_{n=0}^{3}\tilde{x}(n)W_4^{3\cdot n} = \sum_{n=0}^{3}\tilde{x}(n)(-j)^{3n} = -2 - 2j$$

考察式(2.2-5)，并用 MATLAB 实现计算，可采用下面方法。由于 $\tilde{x}(n)$ 和 $\tilde{X}(k)$ 均为周期函数，周期为 N，可设 $\tilde{x}$ 和 $\tilde{X}$ 代表序列 $\tilde{x}(n)$ 和 $\tilde{X}(k)$ 的主值区间序列(简称主值序列)，则由式(2.2-4)和(2.2-5)可写为

$$\tilde{X} = W_N\tilde{x} \tag{2.2-7}$$

$$\tilde{x} = \frac{1}{N}W_N^*\tilde{X} \tag{2.2-8}$$

式中，

$$W_N \triangleq [W_N^{kn} \quad 0 \leqslant (k,n) \leqslant N-1]$$
$$= \begin{matrix} k \\ \downarrow \end{matrix}\begin{bmatrix} 1 & 1 & n\rightarrow & 1 \\ 1 & W_N^1 & \cdots & W_N^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{N-1} & \cdots & W_N^{(N-1)^2} \end{bmatrix} \tag{2.2-9}$$
$$W_N^* \triangleq [W_N^{-kn} \quad 0 \leqslant (k,n) \leqslant N-1]$$

矩阵 W_N 为方阵，叫做 DFS 矩阵。

用 MATLAT 编写函数 dfs 实现 DFS 计算，程序如下：

```
function [Xk]=dfs(xn,N)
% To calculate coefficients of DFS
n=[0:1:N-1];
k=n;
WN=exp(-j*2*pi/N);
nk=n'*k;
WNnk=WN.^nk;
Xk=xn*WNnk;
```

函数 idfs 实现 IDFS 计算，程序如下：

```
%MATLAB PROGRAM
function [xn]=idfs(Xk,N)
% To calculate IDFS
n=[0:1:N-1];
k=n;
WN=exp(-j*2*pi/N);
nk=n'*k;
WNnk=WN.^(-nk);
xn=Xk*WNnk/N;
```

【例 2.2】 (1)例 2.1 的周期序列 $\tilde{x}(n)$,求其 $\tilde{X}(k)$主值序列;(2)利用(1)计算结果,求 $\tilde{x}(n)$的主值序列。

用 MATLAB 编写程序计算 $\tilde{X}(k)$如下:

```
%MATLAB PROGRAM 2-1
xn=[0 1 2 3];
N=4;
Xk=dfs(xn,N)'

≫ smp201

Xk =
      6.0000
  -2.0000 - 2.0000i
  -2.0000 + 0.0000i
  -2.0000 + 2.0000i

Xk =
      6.0000
  -2.0000 - 2.0000i
  -2.0000 + 0.0000i
  -2.0000 + 2.0000i
```

运行结果和例 2.1 相同。用 MATLAB 编写程序计算 $\tilde{x}(n)$,程序如下:

```
%MATLAB PROGRAM 2-2
Xk =[6.0000 -2.0000 + 2.0000i -2.0000 -2.0000 - 2.0000i];
N=4;
xn=idfs(Xk,N)'

≫ smp202
xn =
```

```
        0
   1.0000 - 0.0000i
   2.0000 + 0.0000i
   3.0000 + 0.0000i
```

【例 2.3】 周期方波序列

$$\tilde{x}(n)=\begin{cases}1, & mN\leqslant n\leqslant mN+L-1,\\ 0, & mN+L\leqslant n\leqslant (m+1)N-1,\end{cases}\quad m=0,\pm 1,\pm 2\cdots$$

其中,N 是基本周期,L/N 是占空比,绘出下列情况下的|X(k)|曲线:

(a) L=5,N=20 ; (b) L=5,N=40;

(c) L=5,N=40 ; (d) L=7,N=60。

用 MATLAB 编写程序如下(利用 dfs 函数):

```
%MATLAB PROGRAM 2-3
clf
L=5;N=20;k=[-N/2:N/2];
xn=[ones(1,L),zeros(1,N-L)];
Xk=dfs(xn,N);
magXk=abs([Xk(N/2+1:N) Xk(1:N/2+1)]);
subplot(221);
stem(k,magXk);
axis([-N/2,N/2,0,5.5]);
xlabel('k');ylabel('|X(k)|');
title('DFS of Square Wave L=5 N=20');

L=5;N=40;k=[-N/2:N/2];
xn=[ones(1,L),zeros(1,N-L)];
Xk=dfs(xn,N);
magXk=abs([Xk(N/2+1:N) Xk(1:N/2+1)]);
subplot(222);
stem(k,magXk);
axis([-N/2,N/2,0,5.5]);
xlabel('k');ylabel('|X(k)|');
title('DFS of Square Wave L=5 N=40');

L=5;N=60;k=[-N/2:N/2];
xn=[ones(1,L),zeros(1,N-L)];
Xk=dfs(xn,N);
magXk=abs([Xk(N/2+1:N) Xk(1:N/2+1)]);
subplot(223);
```

```
stem(k,magXk);
axis([-N/2,N/2,0,5.5]);
xlabel('k');ylabel('|X(k)|');
title('DFS of Square Wave L=5 N=60');

L=6;N=70;k=[-N/2:N/2];
xn=[ones(1,L),zeros(1,N-L)];
Xk=dfs(xn,N);
magXk=abs([Xk(N/2+1:N) Xk(1:N/2+1)]);
subplot(224);
stem(k,magXk);
axis([-N/2,N/2,0,7.5]);
xlabel('k');ylabel('|X(k)|');
title('DFS of Square Wave L=6 N=70');
```

程序运行结果如图 2.4 所示。

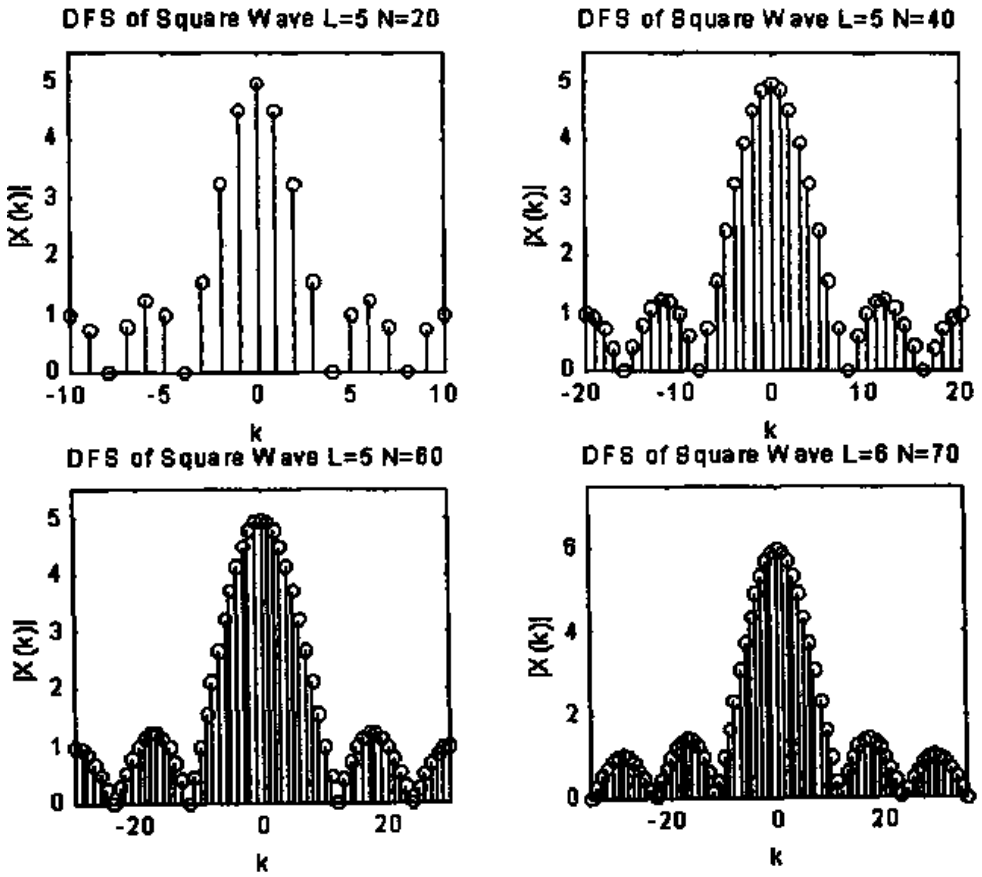


图 2.4 周期方波的傅里叶级数

2.2.2 有限长序列——离散傅里叶变换

有限长序列 x(n)表达式为

$$x(n)=\begin{cases}x(n), & 0\leqslant n\leqslant N-1\\ 0, & n\text{ 为其他值}\end{cases} \tag{2.2-10}$$

x(n)是非周期序列,但可理解为周期序列 $\tilde{x}(n)$的主值序列。由离散傅里叶级数 DFS 和 IDFS 很容易引出有限长序列的离散傅里叶正、逆变换关系式为

$$X(k)=DFT[x(n)]=\sum_{n=0}^{N-1}x(n)W^{nk},\qquad 0\leqslant k\leqslant N-1 \tag{2.2-11}$$

$$x(n)=IDFT[X(k)]=\frac{1}{N}\sum_{k=0}^{N-1}X(k)W^{-nk},\qquad 0\leqslant n\leqslant N-1 \tag{2.2-12}$$

DFT[·]表示离散傅里叶正变换,IDFT[·]表示离散傅里叶逆变换。

由式(2.2-11)和式(2.2-12)可知,长度为 N 的有限长序列 x(n)的离散傅里叶变换 X(k)仍然是一个 N 长的频域有限长度序列。x(n)和 X(k)有唯一确定的对应关系。

若把有限长序列 x(n)看成周期序列 $\tilde{x}(n)$的主值序列,相应离散傅里叶变换 X(k)即为相应周期序列的傅里叶级数 $\tilde{X}(k)$的主值序列,即

$$x(n)=\tilde{x}(n)\cdot R_N(n)=IDFS[\tilde{X}(k)]\cdot R_N(n) \tag{2.2-13}$$

$$X(k)=\tilde{X}(k)\cdot R_N(n)=DFS[\tilde{x}(n)]\cdot R_N(n) \tag{2.2-14}$$

式中,$R_N(n)$表示长度为 N 点的矩形序列

$$R_N(n)=\begin{cases}1, & 0\leqslant n\leqslant N-1\\ 0, & \text{其他 } n\end{cases} \tag{2.2-15}$$

式(2.2-13)和(2.2-14)表明,离散傅里叶变换和离散傅里叶级数有着固有的内在关系。

显然,在 0≤k≤N−1 上 DFT 和 DFS 结果相同;在 0≤n≤N−1 上 IDFT 和 IDFS 结果相同。

离散傅里叶变换 X(k)是一个 N 长度的序列,所对应的离散频率 ω 在 0～2π 之间,且频率间隔相等,为 2π/N.

MATLAB 中,以列向量 x 和 X 分别表示序列 x(n),X(k),由式(2.2-11)和(2.2-12),

$$X=W_Nx \tag{2.2-16}$$

$$x=\frac{1}{N}W_N^*X \tag{2.2-17}$$

式中,W_N 和 W_N^* 与式(2.2-7)和(2.2-8)相同,只是这里称为 DFT 矩阵。

可用函数 dfs 和 idfs 实现傅里叶变换和傅里叶逆变换,并将函数分别更名为 dft 和 idft.

函数 dft 用来求序列的 DFT,调用格式为

$$[X_k]=dft(x_n,N)$$

x_n 为有限长序列,N 为序列 x_n 长度,X_k 为序列 x_n 的 DFT

函数 idft 用来求 IDFT,调用格式为

$$[x_n]=idft(X_k,N)$$

X_k 为有限长序列,N 为序列 X_k 长度,x_n 为序列 X_k 的 IDFT。

【例 2.4】 已知序列 x(n)=cos(0.48πn)+cos(0.52πn), 0≤n≤100,试绘制 x(n)及

它的离散傅里叶变换|X(k)|图。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 2-4
clf
N=100;
n=0:N-1;
xn=cos(0.48*pi*n)+cos(0.52*pi*n);
Xk=dft(xn,N);
magXk=abs(Xk);phaXk=angle(Xk);
subplot(221)
plot(n,xn);
xlabel('n');ylabel('x(n)');
title('x(n) N=100');

subplot(222);
k=0:length(magXk)-1;
plot(k,magXk);
xlabel('k');ylabel('|X(k)|');
title('X(k) N=100');
```

程序运行结果如图 2.5 所示。

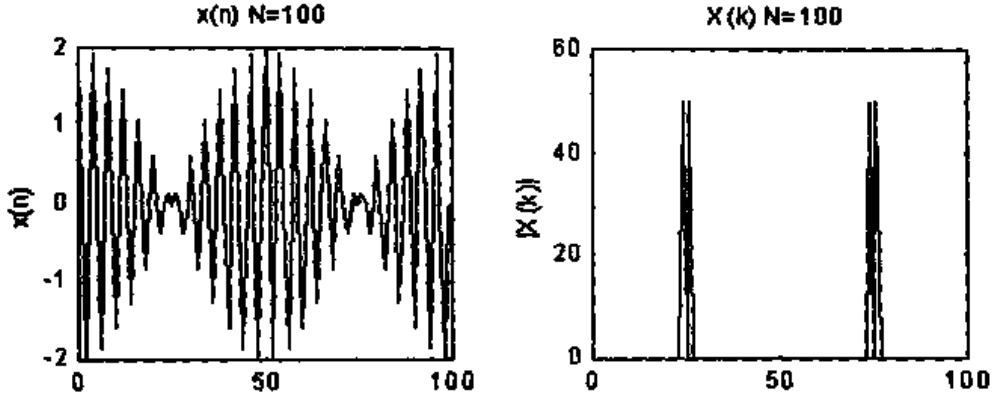


图 2.5 复合余弦信号的傅里叶变换

若序列 x(n)长度为 N，可通过增零形成长度为 N_1 的有限长序列 $x_1(n)$，则有

$$X(k) = DFT[x(n)] = \sum_{n=0}^{N-1} x(n)W^{nk}$$

$$X_1(k) = DFT[x_1(n)] = \sum_{n=0}^{N_1-1} x_1(n)W^{-nk}$$

式中，$N_1 > N$.

显然，$X_1(k)$和 X(k)具有相同特征，但二者是不相同的。$X_1(k)$的长度为 N_1，而 X(k)的长度为 N。这一点由例 2.5 可清楚地看出。因此，一个有限长序列可进行大于其序列长度的任意点数的离散的傅里叶变换，具体点数可根据实际需要选定。

【例 2.5】 对于例 2.4 序列，通过增零补长至长度 120，绘制 x(n)和它的离散傅里叶变换|X(k)|图。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 2-5
nn=100;
Nn=120;
n=[0:nn-1];
n1=[0:Nn-1];

xn=cos(0.48*pi*n)+cos(0.52*pi*n);
xn=[xn zeros(1,Nn-nn)];
Xk=dft(xn,Nn);
magXk=abs(Xk);phaXk=angle(Xk);
subplot(221)
plot(n1,xn);
axis([0 120 -2 2])
xlabel('n');ylabel('x(n)');
title('x(n) N=120');

subplot(222);
k=0:length(magXk)-1;
plot(k,magXk);
axis([0 120 0 50])
xlabel('k');ylabel('|X(k)|');
title('X(k) N=120');
```

程序运行结果如图 2.6 所示

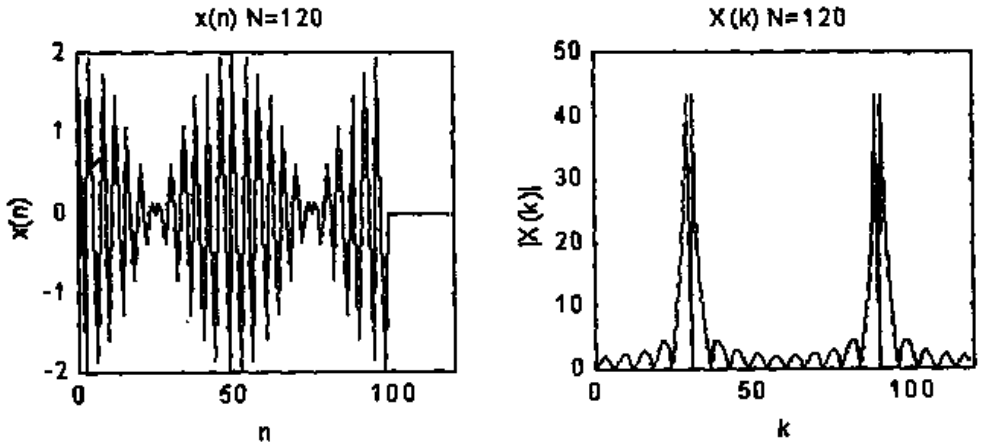


图 2.6 增零补长的复合余弦的傅氏变换

2.2.3 离散傅里叶变换的性质

以下讨论假定 x(n)和 y(n)都是长度 N 的有限长序列，且

$$X(k) = DFT[x(n)]$$

$$Y(k) = DFT[y(n)]$$

一、线性特性

$$DFT[ax(n) + by(n)] = aX(k) + bY(k) \tag{2.2-18}$$

二、时移特性

为了研究有限长序列时移特性，引入"圆周移位"概念。有限长序列 x(n)位于 $0 \leqslant n \leqslant N-1$ 区间，向右时移 m 位，序列 x(n−m)仍为有限长，位于 $m \leqslant n \leqslant N+m-1$ 区间，给 DTT 带来不便。但由圆周移位(或称循环移位)定义，有限序列 x(n)的圆周移位序列为

$$f(n) = x[(n-m)]_N R_N \tag{2.2-19}$$

式(2.2-19)表明 f(n)仍是长度 N 的有限序列，x(n)右移 m 位时，超出 N−1 以外的 m 个采样值又从左边依次填补了空位，相当于序列 x(n)排列在 N 等分的圆周上，N 个采样点首尾相接。序列 x(n)右移 m 位，相当于 x(n)在圆周上逆时针旋转 m 位，故称为圆周移位或循环移位。

圆周移位特性是向右圆移 m 位，若

$$DFT[x(n)] = X(K)$$

$$y(n) = x((n-m)_N R_N(n)$$

则

$$Y(k) = DFT[y(n)] = X(k)W_N^{mk} \tag{2.2-18}$$

式(2.2-18)表明，序列在时域上圆周移位，在频域上将产生附加相移，幅频特性保持不变。序列向左移位具有相同的特性。

【例 2.6】 已知序列 $x(n)=10(0.8)^n(0 \leqslant n \leqslant 10)$，序列圆周向右移位 m=3，绘制

(1)原序列波形及傅氏变换幅值图；

(2)圆周移位序列波形及傅氏变换幅值图。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 2-6
n=[0:10];
M=6;
N=11;
%The Original Sequence
x=10*0.8.^n;
y=cirshift(x,M,N);

subplot(221)
stem(n,x);
title('Original Sequence');
xlabel('n');ylabel('x(n)');
```

```
subplot(222)
stem(n,y);
title('Circlar Shift Sequence');
xlabel('n');ylabel('y(n)');

Xk=dft(x,N);
magXk=abs(Xk);phaXk=angle(Xk);
subplot(223);
stem(n,magXk);
title('Original Sequence');
xlabel('k');ylabel('|X(k)|');

Yk=dft(y,N);
magYk=abs(Yk);phaYk=angle(Yk);
subplot(224);
stem(n,magYk);
title('Circlar Shift Sequence');
xlabel('k');ylabel('|Y(k)|');
```

程序运行结果如图 2.7 所示。

该程序中调用了两个自编函数 cirshift 和 sigmod. 若周期序列由有限长序列 x(n)产生,周期为 N. 函数 sigmod 用来找出周期序列任意位置 n 所对应的主值有限序列 x(n)中的位置 m.

```
function m=sigmod(n,N);
m=rem(n,N);
m=m+N;
m=rem(m,N);
```

其中,n 为周期序列的元素位置序号;N 为序列周期;m 为主值序列 x(n)的元素位置序号。

【例 2.7】 序列 x(n)= [5 4 3 2 1 0],由它构成周期序列 $\tilde{x}(n)$,求 $\tilde{x}(16)$幅值。

用 MATLAB 编写程序如下:

```
%MATLAB PROGRAM 2-7
x=[5 4 3 2 1 0];
N=6;
n=16;
xp=[x x x x x];
nx=[0:length(xp)-1];
```

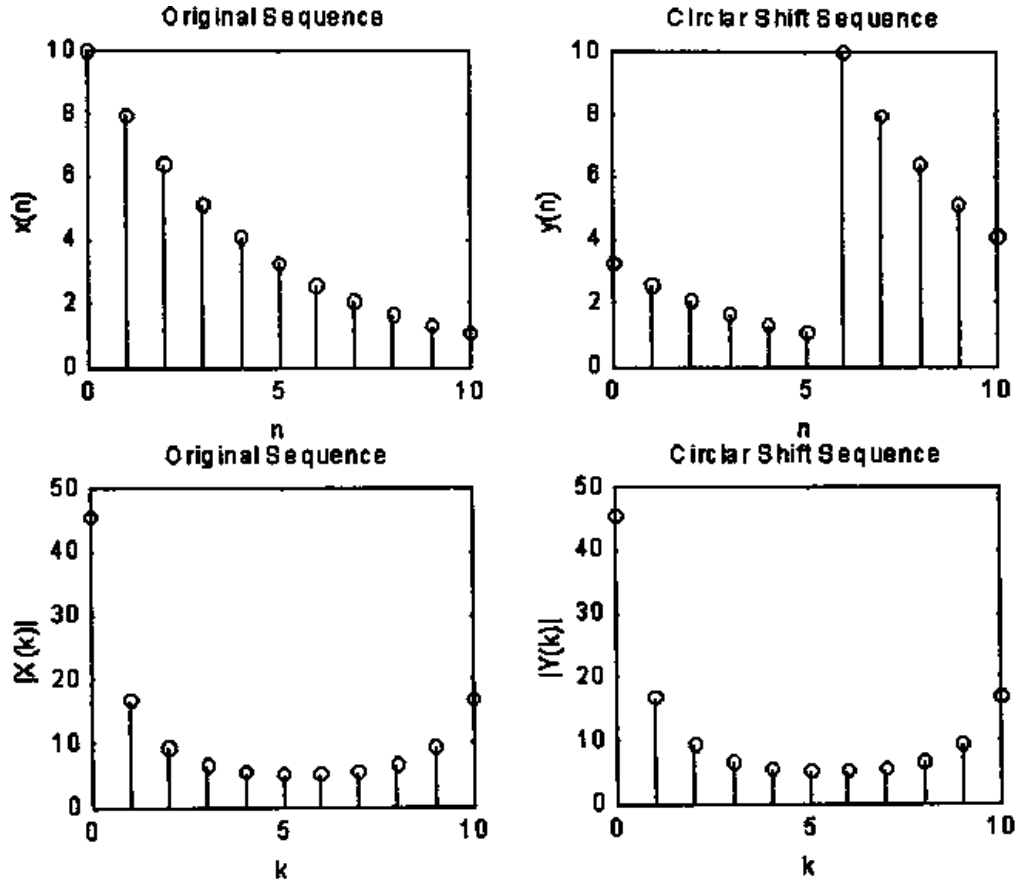


图 2.7　序列圆周移位及傅氏变换

```
subplot(221)
stem(nx,xp);
xlabel('n');ylabel('x(n)');
m=sigmod(n,N)
xv=x(m)
```

程序运行结果：

$$m = 4 \qquad xv = 2$$

绘制的周期序列如图 2.8 所示。

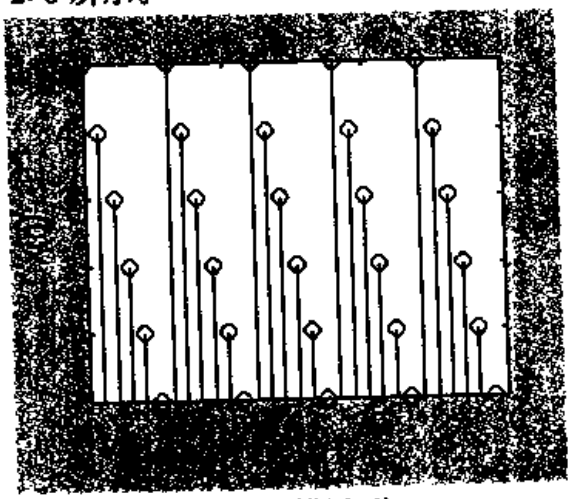


图 2.8　周期序列

由图可见，周期序列序号 16 位置处值为 2，相当于原序列 x(4)的值。

函数 cirshift 用来产生序列的圆周移位，程序清单如下：

```
function y=cirshift(x,m,N);
%x:Input sequence
%m:Shift number
%N:Show length
if length(x)>N
    error('N must be greater then length(x)');
end
x=[x zeros(1,N-length(x))];
n=[0:N-1];
n=sigmod(n-m,N);
y=x(n+1);
```

三、频移特性

若

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

$$Y(k) = X((k-l))_N R_N(k)$$

则

$$y(n) = \mathrm{IDFT}[Y(k)] = x(n)W^{-ln} \tag{2.2-21}$$

四、奇偶虚实特性

设 x(n)为实序列，DFT[x(n)]=X(k)，令

$$X(k) = X_r(k) + jX_i(k)$$

式中，$X_r(k)$是 X(k)的实部，$X_i(k)$是 X(k)的虚部。

可以证明，$X_r(k)$是偶函数，$X_i(k)$是奇函数，即

$$X_r(k) = X((-k))_N R_N(k)$$

$$X_i(k) = -X((-k))_N R_N(k) \tag{2.2-22}$$

【例 2.8】 已知序列 $x(n)=10(0.8)^n$，序列长度 N=21，绘出 x(n)傅里叶变换的实部和虚部并验证其奇偶性。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 2-8
n=[0:20];
N=length(n);
%The original sequence
x=10*0.8.^n;
subplot(221);
stem(n,x);
title(' sequence x(n)');
xlabel('n');ylabel('x(n)');
```

```
Xk=dft(x,N);
Xkreal=real(Xk);
Xkimage=imag(Xk);

subplot(222);
stem(n,Xkreal);
xlabel('k');ylabel('XkReal(k)');
title('Real Part X(k)');

subplot(223);
stem(n,Xkimage);
hold on
plot(n,zeros(1,N))
xlabel('k');ylabel('XkImg(k)');
title('Image Part X(k)');
hold off
```

程序运行结果可得图 2.9。由图 2.9 可见，对于实序列 x(n)，其 DFT 的实部是偶对称，而虚部为奇对称。

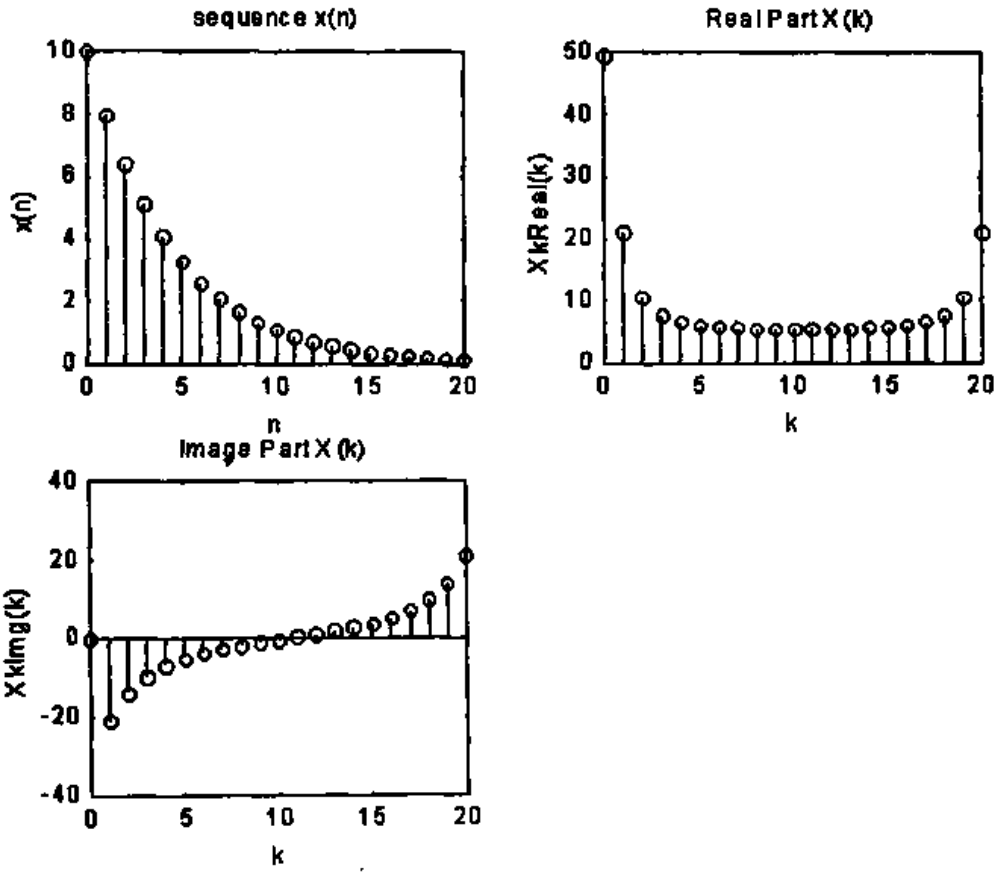


图 2.9　DFT 的奇偶虚实特性

在 1.3.2 节中已提及，任何一个序列 x(n)可以分解为偶分量 $x_e(n)$和奇分量 $x_o(n)$，则有

$$\begin{aligned} DFT(x_e(n)) &= X_r(k) \\ DFT(x_o(n)) &= X_i(k) \end{aligned} \tag{2.2-23}$$

这里，为了实现对序列的偶分量和奇分量的 DFT，$x_e(n)$和 $x_o(n)$定义为：

$$x_e(n) \triangleq \frac{1}{2}[x(n) + x(-n)]$$

$$x_o(n) \triangleq \frac{1}{2}[x(n) - x(-n)]$$

x(n)的分解可借助下面函数 circevod，程序清单如下：

```
function [xev,xod]=circevod(x)
if any(imag(x)~=0)
    error('The sequence is not real.')
end
N=length(x);
n=0:(N-1);
xev=0.5*(x+x(sigmod(-n,N)+1));
xod=0.5*(x-x(sigmod(-n,N)+1));
```

其中，x 为原序列；xev 为序列的偶分量；xod 为序列的奇分量。

【例 2.9】 将例 2.8 中的序列分解为奇、偶序列，分别绘出其 DFT，并与图 2.9 作比较。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 2-9
n=[0:20];
N=length(n);
%The original sequence
x=10*0.8.^n;
[xev,xod]=circevod(x);
subplot(221);
stem(n,xev);
title(' Even Part ');
xlabel('n');ylabel('xev(n)');

subplot(222);
stem(n,xod);
title(' Odd Part ');
xlabel('n');ylabel('xod(n)');
hold on
```

```
plot(n,zeros(1,N))
hold off

Xkev=dft(xev,N);
Xkod=dft(xod,N);

subplot(223);
stem(n,real(Xkev));
xlabel('k');ylabel('Xkev(k)');
title('DFT of Even part');

subplot(224);
stem(n,imag(Xkod));
xlabel('k');ylabel('Xkod(k)');
title('DFT of Odd part');
hold on
plot(n,zeros(1,N))
hold off
```

程序运行结果可得图 2.10。

比较图 2.9 和图 2.10，证实了式(2.2-23)的正确性。

五、时域圆周卷积

若 $Y(k)=X(k)H(k)$，则

$$y(n) = \mathrm{IDFT}[Y(k)] = \sum_{m=0}^{N-1} x(m)h((n-m))_N R_N(n) \tag{2.2-24}$$

此卷积过程只在 $0\leqslant m\leqslant N-1$ 区间进行，$h((n-m))_N$ 实际上是 $h(m)$ 的圆周移位，所以称圆周卷积。为了区别，把一般卷积(仅作平移)称为线性卷积，如式(1.7-13)。圆周卷积的符号以⊛表示。因此

$$\begin{aligned} x(n)\circledast h(n) &= \sum_{m=0}^{N-1} x(m)h((n-m))_N R_N(n) \\ &= \sum_{m=0}^{N-1} h(m)x((n-m))_N R_N(n) \end{aligned} \tag{2.2-25}$$

$$\mathrm{DFT}[x(n)\circledast h(n)] = X(k)\cdot H(k)$$

因此

$$x(n)\circledast h(n) = \mathrm{IDFT}[X(k)H(k)] \tag{2.2-26}$$

圆周卷积的计算方法可分为时域方法和频域方法两种。基于式(2.2-25)计算圆周卷积称为时域方法，编制函数 circonvt 可实现圆周卷积运算。

函数 circonvt 的程序清单如下：

```
function y=circonvt(x1,x2,N)
```

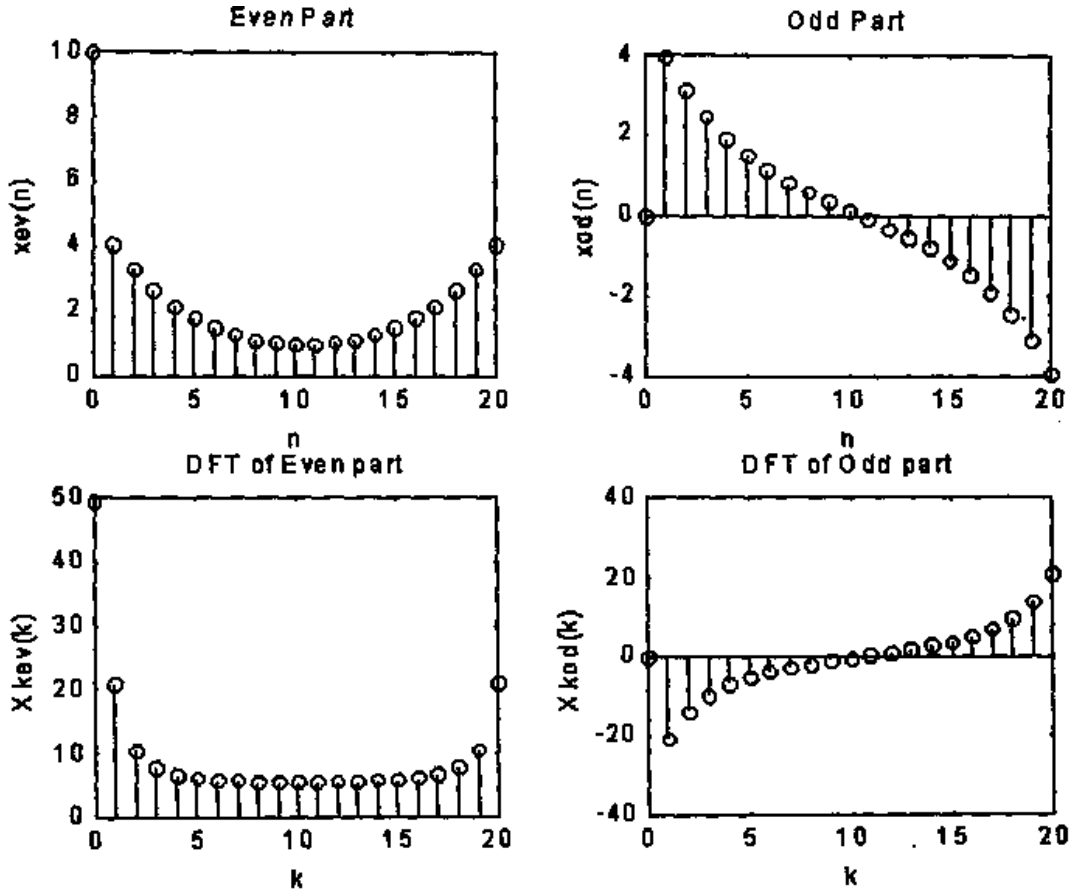


图 2.10　序列的奇偶部分的 DFT

```
%To compute y=x1⊛x2
if length(x1)>N
    error('Length(x1)is not greater than N');
end

if length(x2)>N
    error('Length(x2)is not greater than N');
end

x1=[x1,zeros(1,N-length(x1))];
x2=[x2,zeros(1,N-length(x2))];
m=[0:N-1];
x2=x2(mod(-m,N)+1);
H=zeros(N,N);
for n=1:N;
    H(n,:)=cirshift(x2,n-1,N);
end
y=x1*H';
```

函数调用格式

$$y = \text{circonvt}(x1,x2,N)$$

其中,N 为圆周卷积长度; x1,x2 为长度小于或等于 N 的有限长序列。

$$y = x1 \textcircled{N} x2$$

【例 2.10】 已知 x1=[1 2 2],x2=[1 2 3 4],试计算(1) x1④x2; (2) x1⑥x2; (3) x1⑦x2; (4) x1 * x2。

用 MATLAB 编写计算程序如下:

```
%MATLAB PROGRAM 2-10
x1=[1 2 2];
x2=[1 2 3 4];
disp('Circular Convolution N=5:')

N=5;
y=circonvt(x1,x2,N)
disp('Circular Convolution N=6:')
N=6;
y=circonvt(x1,x2,N)
disp('Circular Convolution N=7:')
N=7;
y=circonvt(x1,x2,N)
disp('Linear Convolution:')
yc=conv(x1,x2)

≫ smp210
Circular Convolution N=5:
y =
     9     4     9    14    14
Circular Convolution N=6:
y =
     1     4     9    14    14     8
Circular Convolution N=7:
y =
     1     4     9    14    14     8     0
Linear Convolution:
yc =
     1     4     9    14    14     8
```

计算结果表明,两个序列的 N 点的圆周卷积是一个长度为 N 的序列。6 点圆周卷积和利用 conv 函数计算的线卷积相同。值得注意的是,两个序列的线卷积序列长度为两序列长度之和减 1,本例中,线卷积序列长度为 6。同时,还会发现 5 点圆周卷积的第一项为

6点圆周卷积的第一项和最后一项之和。

基于式(2.2-26)计算圆周卷积称为频域方法。

【例2.11】 按频域方法计算例2.10的圆周卷积。

用MATLAB编写程序实现这种卷积方法,程序清单如下:

```
%MATLAB PROGRAM 2-11
x1=[1 2 2];
x2=[1 2 3 4];
N=7
x1=[x1,zeros(1,N-length(x1))];
x2=[x2,zeros(1,N-length(x2))];
Xk1=dft(x1,N);
Xk2=dft(x2,N);
Yk=Xk1. * Xk2;
y=idft(Yk,N)
```

程序中,N为圆周卷积长度。当N分别取5,6,7时,程序运行结果所得圆周卷积y(n)和例2.10的时域计算结果完全相同。

由以上所述,对于有限长序列存在两种卷积方法:线性卷积和圆周卷积。由例2.10和例2.11的运算结果可清楚地得出下面结论:

若x(n)是长度为nx的有限长序列,h(n)是长度为nh的有限长序列,要使N点圆周卷积

$$y_c(n) = x(n) \textcircled{N} h(n)$$

和线性卷积

$$y_l(n) = x(n) * h(n)$$

相等且不发生混迭失真的主要条件是

$$N \geqslant nx + nh - 1 \tag{2.2-27}$$

六、频域圆周卷积

若y(n)=x(n)h(n),则

$$\begin{aligned} Y(k) &= DFT[y(n)] \\ &= \frac{1}{N}\sum_{l=0}^{N-1} X(l)H((k-1))_N R_N(k) \\ &= \frac{1}{N}\sum_{l=0}^{N-1} H(l)X((k-1))_N R_N(k) \end{aligned} \tag{2.2-28}$$

七、帕斯瓦尔(parseval)定理

若DFT[x(n)]=X(k),则

$$\sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N}\sum_{k=0}^{N-1} |X(k)|^2 \tag{2.2-29}$$

如果x(n)为实序列,则有

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

式(2.2-29)用来计算频域的能量。

2.3 z变换

上一节讨论了离散时间信号的傅里叶变换。但傅氏变换存在缺点——离散信号的傅氏变换不总是存在的，它必须满足均匀收敛的条件，即

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

而工程应用中许多有用信号不满足上述条件，如单位阶跃信号u(n)，它的傅氏变换不存在，因此必须采用z变换来分析信号和系统。

在离散信号与系统的理论研究中，z变换是一种重要的数学工具。它把离散系统的数学模型——差分方程转化为简单的代数方程，使求解过程得以简化。特别是计算机采样、数据处理、控制技术的发展使z变换占有更加重要地位。

本节介绍z变换的定义、性质及z变换与傅氏变换的关系。

2.3.1 定义

z变换有单边和双边之分。

序列x(n)的单边z变换定义为

$$X(z)=Z[x(n)]=x(0)+\frac{x(1)}{z}+\frac{x(2)}{z^2}+\cdots=\sum_{n=0}^{\infty}x(n)z^{-n}\qquad(2.3\text{-}1)$$

序列x(n)的双边z变换定义为

$$X(z)=z[x(n)]=\sum_{n=-\infty}^{\infty}x(n)z^{-n}\qquad(2.3\text{-}2)$$

式中，$z=re^{j\omega}$是一个复变量。X(z)是一个以z为自变量的函数。

单边z变换的重要特点是可以考虑起始条件，在离散系统分析中，它既可以求零输入响应，也可以获得零状态响应。单边z变换更容易收敛，因此在实际中应用较多。

双边z变换，序列n从$-\infty$开始，所以无法考虑初始条件，但对于数字信号处理，一般无需考虑初值。双边z变换，序列n可以在$-\infty$至$+\infty$内变化，便于与傅里叶变换直接挂钩，因此在信号处理理论中采用双边z变换较多。

2.3.2 z变换的收敛域

对于任何序列x(n)，能保证式(2.3-1)或式(2.3-2)所表示的级数收敛的所有z值的集合称为z变换的收敛域(Region of Convergence，简称ROC)。讨论z变换ROC的意义在于给出z变换函数表达式同时，必须说明它的收敛域后，才能单独地确定它所对应的序列。

z变换的收敛域可表示为

$$R_{x-}<|z|<R_{x+}\qquad(2.3\text{-}3)$$

其中，R_{x-}和R_{x+}是为收敛半径，正数，收敛域大小和序列x(n)的性质有关。

(1) 有限长序列

$$x(n)=\begin{cases}x(n), & N_1\leqslant n\leqslant N_2\\ 0, & \text{其他}\end{cases}$$

z 变换 ROC 为 $0<|z|<\infty$ 的 z 平面上处处收敛。

(2) 右边序列

序列是一个有始无终序列，其 z 变换为

$$X(z)=\sum_{n=n_1}^{\infty}x(n)z^{-n}$$

它的 ROC 为半径 R_{x-} 的圆外部分。

(3) 左边序列

序列是一个无始有终序列，其 z 变换为

$$X(z)=\sum_{n=-\infty}^{n_2}x(n)z^{-n}$$

它的 ROC 为半径为 R_{x+} 的圆内部分。

(4) 双边序列

序列从 $n=-\infty$ 延伸到 $n=+\infty$ 的序列，其 z 变换为

表 1.1

序列	z 变换	收敛域
$\delta(n)$	1	$0\leqslant\|z\|\leqslant\infty$
$\delta(n-k)$	z^{-k}	$0\leqslant z\leqslant\infty$
$u(n)$	$\frac{z}{z-1}=\frac{1}{1-z^{-1}}$	$\|z\|>1$
$R_N(n)$	$\frac{1-z^{-N}}{1-z^{-1}}$	$\|z\|>0$
$nu(n)$	$\frac{z}{(z-1)^2}=\frac{z^{-1}}{(1-z^{-1})^2}$	$\|z\|>1$
$a^nu(n)$	$\frac{z}{z-a}=\frac{1}{1-az^{-1}}$	$\|z\|>\|a\|$
$-a^nu(-n-1)$	$\frac{z}{z-a}=\frac{z}{1-az^{-1}}$	$\|z\|<\|a\|$
$e^{-j\omega_0 n}u(n)$	$\frac{z}{z-e^{-j\omega_0}}=\frac{1}{1-e^{-j\omega_0}z^{-1}}$	$1<\|z\|\leqslant\infty$
$\sin(\omega_0 n)u(n)$	$\frac{z\sin\omega_0}{z^2-2z\cos\omega_0+1}$	$1<\|z\|\leqslant\infty$
$\cos(\omega_0 n)u(n)$	$\frac{z(z-\cos\omega_0)}{z^2-2z\cos\omega_0+1}$	$1<\|z\|\leqslant\infty$

$$X(z)=\sum_{n=-\infty}^{\infty}x(n)z^{-n}=\sum_{n=0}^{\infty}x(n)z^{-n}+\sum_{n=-\infty}^{-1}x(n)z^{-n}$$

显然，它是由左边序列和右边序列的 z 变换叠加。因此，它的 ROC 为

$$R_{x1}<|z|<R_{x2}$$

z 变换可写成下面形式

$$x(z)=\frac{P(z)}{Q(z)}$$

其分母 Q(z)的根为 x(z)的极点，使 x(z)→∞，因此 x(z)在收敛域内不会出现极点，收敛域常以极点为边界。

表 1.1 给出常用序列的 z 变换及其收敛域，便于读者查阅。

2.3.3 逆 z 变换

若已知序列 x(n)的 z 变换为

$$X(z)=\mathcal{Z}[x(n)]$$

则 X(z)的逆 z 变换记为

$$x(n)=\mathcal{Z}^{-1}[X(z)]$$

求逆 z 变换的方法通常有三种：围线积分法（留数法）、幂级数展开法（长除法）及部分分式展开法。用 MATLAB 实现逆 z 变换最方便的方法是部分分式展开法。

若给定 X(z)具有下面形式：

$$X(z)=\frac{b_0+b_1z^{-1}+\cdots+b_mz^{-m}}{1+a_1z^{-1}+\cdots+a_nz^{-n}},\qquad R_{x-}<|z|<R_{x+}$$

这里，分母和分子均按 z^{-1}升幂排列。

用 MATLAB 函数 residuez 将其分解为式(1.7-4)或(1.7-5)形式，它们包含 z 变换的最基本形式，由 z 变换基本公式求出 x(n)，即

$$\mathcal{Z}^{-1}\left[\frac{1}{1-az^{-1}}\right]=\begin{cases}a^nu(n), & |z_k|\leqslant R_{x-}\\ -a^nu(-n-1), & |z_k|\geqslant R_{x+}\end{cases}\qquad(2.3\text{-}4)$$

【例 2.12】 已知 $X(z)=\dfrac{z}{3z^2-4z+1}$，求在下列收敛域下的逆 z 变换。

(1) $1<|z|<\infty$；　(2) $0<|z|<\frac{1}{3}$；　(c) $\frac{1}{3}<|z|<1$

为了利用 MATLAB 求其逆 z 变换，将 x(z)改写成下面形式

$$X(z)=\frac{z^{-1}}{3-4z^{-1}+z^{-2}}$$

用 MATLAB 编写程序求上式的部分分式

```
% MATLAB PROGRAM 2-12
num=[0 1];
den=[3 -4 1];
[r,p,k]=residuez(num,den)
```

```
≫ smp212
r =
    0.5000
   -0.5000
p =
    1.0000
    0.3333
k =
    []
```

则 $X(z)=\dfrac{\frac{1}{2}}{1-z^{-1}}-\dfrac{\frac{1}{2}}{1-\frac{1}{3}z^{-1}}$

(1) $1<|z|<\infty$，$R_{x-}=1$，两个极点 $z_1=1, z_2=\frac{1}{3}$，$|z_1|=1<R_{x-}$，$|z|=\frac{1}{3}<R_{x-}$

$$x(n)=Z^{-1}[X(z)]=\frac{1}{2}u(n)-\frac{1}{2}\left(\frac{1}{3}\right)^n u(n)$$

这是一个右边序列。

(2) $0<|z|<\frac{1}{3}$，$R_{x+}=\frac{1}{3}$，$|z_1|=1>R_{x+}$，$|z_2|=\frac{1}{3}>R_{x+}$

$$x(n)=\frac{1}{2}[-u(-n-1)]-\frac{1}{2}\left[-\left(\frac{1}{3}\right)^n u(-n-1)\right]$$

$$=\frac{1}{2}\left(\frac{1}{3}\right)^n u(-n-1)-\frac{1}{2}u(-n-1)$$

这是一个左边序列。

(3) $\frac{1}{3}<|z|<1$，$R_{x-}=\frac{1}{3}$，$R_{x+}=1$，$|z_1|=1>R_{x+}$，$|z_2|=\frac{1}{3}<R_{x-}$

$$x(n)=-\frac{1}{2}u(-n-1)-\frac{1}{2}\left(\frac{1}{3}\right)^n u(n)$$

这是一个双边序列。

【例 2.13】 已知 $X(z)=\dfrac{z+1}{3z^2-4z+1}$，求 $|z|>1$ 的逆 z 变换。

已知 $X(z)=\dfrac{z^{-1}+z^{-2}}{3-4z^{-1}+z^{-2}}$，$|z|>1$

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 2-13
num=[0 1 1];
den=[3 -4 1];
[r,p,k]=residuez(num,den)
```

```
≫ smp213
r =
```

```
    1.0000
   -2.0000
p =
    1.0000
    0.3333
k =
    1
```

由此可得

$$x(z) = 1 + \frac{1}{1 - z^{-1}} - \frac{1}{1 - \frac{1}{3}z^{-1}}$$

由表 1 得到逆 z 变换

$$x(n) = \delta(n) + u(n) - 2\left(\frac{1}{3}\right)^n u(n)$$

2.3.4 z 变换基本性质

z 变换具有下面基本性质

(1) 线性性质

$$X(z) = \mathcal{Z}[x(n)],\ R_{x-} < |z| < R_{x+}$$
$$y(z) = \mathcal{Z}[y(n)],\ R_{y-} < |z| < R_{y+}$$
$$\mathcal{Z}[ax(n) + by(n)] = aX(z) + bY(z)$$
$$\max(R_{x-}, R_{y-}) < |z| < \min(R_{x-}, R_{y-})$$

(2) 序列位移

$$\mathcal{Z}[x(n - n_0)] = z^{-n_0}X(z),\quad R_{x-} < |z| < R_{x+}$$

(3) 序列指数加权(z 域尺度变换)

$$\mathcal{Z}[a^n x(n)] = X\left(\frac{z}{a}\right),\ R_{x-} < \left|\frac{z}{a}\right| < R_{x+}$$

(4) 序列线性加权(z 域微分)

$$\mathcal{Z}[nx(n)] = -z\frac{d}{dz}X(z),\ R_{x-} < |z| < R_{x+}$$

(5) 初值定理

$$x(0) = \lim_{z\to\infty} X(z),\ x(n) \text{ 为因果序列},\ |z| > R_{x-}$$

(6) 终值定理

$x(\infty) = \lim_{z\to 1}(z - 1)X(z)$，$x(n)$ 为因果序列，且当 $|z| \geqslant$ 时$(z - 1)X(z)$ 收敛

(7) 复序列的共轭

$$\mathcal{Z}[x^*(n)] = X^*(z^*),\ R_{x-} < |z| < R_{x+}$$

(8) 时域卷积

$$\mathcal{Z}[x(n) * y(n)] = X(z)Y(z)$$
$$\max(R_{x-}, R_{y-}) < |z| < \min(R_{x+}, R_{y+})$$

(9) z 域卷积(序列相乘)

$$\mathcal{Z}[x(n)y(n)] = \frac{1}{2\pi j}\oint_{c1} X\left(\frac{z}{v}\right)H(v)v^{-1}dv$$

$$R_{x-}R_{y-} < |z| < R_{x+}R_{y+}$$

时域卷积性质在许多方面得到应用，且很容易用 MATLAB 函数实现。下面给出几个例子，说明基本性质的应用。

【例 2.14】 已知序列 x(n)={2,3,4}，y(n)={3,4,5,6}，求 h(n)=x(n) * y(n)。

本题求 x(n) * y(n)方法有二：利用 MATLAB 函数 conv；利用 z 变换性质 x(n) * y(n)=IDFT(X(z)Y(z))。

$$X(z) = 2 + 3z^{-1} + 4z^{-2}$$

$$y(z) = 3 + 4z^{-1} + 5z^{-2} + 6z^{-3}$$

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 2-14
xn=[2 3 4];Xz=[2 3 4];
yn=[3 4 5 6];Yz=[3 4 5 6];
%Convolution
xnCyn=conv(xn,yn)
%polynomial multiplication XzmYz=Xz * Yz
XzMYz=conv(Xz,Yz)

≫ smp214
xnCyn =
     6    17    34    43    38    24
XzMYz =
     6    17    34    43    38    24
```

由上可知，用函数 conv 计算结果

$$h(n) = x(n) * y(n) = \{6 \quad 17 \quad 34 \quad 43 \quad 38 \quad 24\}$$

用 z 变换性质求得

$$H(z) = 6 + 17z^{-1} + 34z^{-2} + 38z^{-3} + 24z^{-1}$$

由此，其相应的序列

$$h(n) = \{6 \quad 17 \quad 34 \quad 43 \quad 38 \quad 24\}$$

两种方法计算结果相同。

实际上在 MATLAB 中，两个多项式乘积是通过两个向量卷积获得的。

【例 2.14】中，序列 x(n)和 y(n)起点均为 n=0。若两个序列的位置不同，不能利用 conv 函数求卷积，只能用自编函数 sigconv 求卷积。

【例 2.15】 已知 $X_1(z) = z + 2 + 3z^{-1}$，$X_2(z) = 2z^2 + 4z + 3 + 5z^{-1}$，求 $X_3(z) = X_1(z)X_2(z)$。

已知 $x_1(n)=[1, 2, 3]$，$n_1=[-1 \quad 0 \quad 1]$

$x_2(n)=[2 \quad 4 \quad 3 \quad 5]$，$n_2=[-2 \quad -1 \quad 0 \quad 1]$

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 2-15
X1=[2 4 8];
n1=[ 0 1 2];
X2=[1 3 5 7];
n2=[-2 -1 0 1];
[X3,n3]=sigconv(X1,n1,X2,n2)

≫ smp215
X3 =
     2     10     30     58     68     56
n3 =
    -2     -1      0      1      2      3
```

由结果可知

$$X_3(z) = 2z^2 + 10z + 30 + 56z^{-1} + 68z^{-2} + 56z^{-3}$$

所对应的时间序列

$$n = \{-2 \quad -1 \quad 0 \quad 1 \quad 2 \quad 3\}$$

$$x_3(n) = \{2 \quad 10 \quad 30 \quad 56 \quad 68 \quad 56\}$$

2.3.5 利用 z 变换解差分方程

一个线性时不变系统的差分方程的一般形式为

$$\sum_{k=0}^{N} a_k y(n-k) = \sum_{r=0}^{M} b_r x(n-r) \tag{2.3-4}$$

解差分方程就是求系统响应 y(n)，方法有时域方法(递推算法)和 z 变换方法。

当系统初始状态为零时，由输入引起的响应计算：

$$Y(z) = H(z)X(z)$$

式中，H(z)称为系统 z 传递函数，且具有下面形式

$$H(z) = \frac{\sum_{r=0}^{M} b_r z^{-r}}{\sum_{k=0}^{N} a_k z^{-k}} = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^M}{a_0 + a_1 z^{-1} + \cdots + a_N z^N} \tag{2.3-5}$$

$$y_1(n) = \mathcal{Z}^{-1}[Y(z)]$$

这里，$y_1(n)$是仅由输入引起的响应，称零状态响应。

当系统输入为零，由初始状态引起的响应计算：

$$Y(z) = H(z)X_{1c}(z)$$

$$y_2(n) = \mathcal{Z}^{-1}[y(z)]$$

这里，$y_2(n)$是仅由初始状态引起的响应，称为零输入响应。

系统的全响应为

$$y(n) = y_1(n) + y_2(n) \tag{2.3-6}$$

借助函数 FILTER 和 FILTIC(详见第五章)可以求解差分方程全解,即离散时间系统在输入和初始状态作用下的响应。应指出的是,式(2.3-4)和式(2.3-5)均具有 DSP 形式,适用于 MATLAB 信号处理工具箱的函数。而在控制系统工具箱中介绍的离散时间系统的模型则不采用 DSP 形式,因此 MATLAB 控制系统工具箱的时间响应分析函数,如 STEP、LSIM 等在 MATLAB 信号处理系统中均不能采用。

【例 2.16】 求解系统差分方程

$$y(n) = x(n) - 5x(n-1) + 8x(n-3)$$

方程两边 z 变换得

$$H(z) = 1 - 5z^{-1} + 8z^{-3}$$

用 MATLAB 编程求解如下:

```
%MATLAB PROGRAM 2-16
b=[1 -5 0 8];
N=30;
n=0:N-1;
x=0.8.^n;
y=filter(b,1,x);
stem(n,y);
grid
```

程序运行结果如图 2.11 所示。

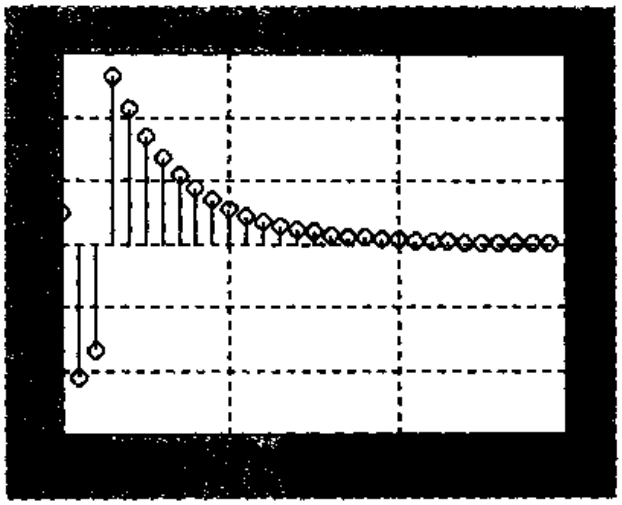


图 2.11 例 2.16 的解

【例 2.17】 求 $y(n)-\frac{3}{2}y(n-1)+\frac{1}{2}(n-2)=x(n)$,在 $x(n)=\left(\frac{1}{4}\right)^n u(n)$ 和初始条件 $y(-1)=4$, $y(-2)=10$ 情况下的解。

先将方程两边进行 z 变换,可得

$$H(z) = \frac{y(z)}{x(z)} = \frac{1}{1 - \frac{3}{2}z^{-1} + \frac{1}{2}z^{-2}}$$

初始条件 $y_0=[4 \quad 10]$

输入 $x(n)=\left(\frac{1}{4}\right)^n u(n)$

用 MATLAB 编程如下:

```
%MATLAB PROGRAM 2-17
n=[0:7]';
x=0.25.^n;
y0=[4 10];
num=1;
den=[1 -3/2 1/2];
Zi=filtic(num,den,y0)
[y,Zf]=filter(num,den,x,Zi);
```

```
y


≫ smp217
Zi =
     1     -2
y =
    2.0000
    1.2500
    0.9375
    0.7969
    0.7305
    0.6982
    0.6824
    0.6745
```

【例 2.18】 求解差分方程 $y(n)-0.4y(n-1)-0.45y(n-2)=0.45x(n)+0.4x(n-1)-x(n-2)$，其中，$x(n)=0.8^n u(n)$，初始状态 $y(-1)=0$，$y(-2)=1$，$x(-1)=1$，$x(-2)=2$。

将方程两边进行 z 变换得

$$H(z)=\frac{0.45+0.4z^{-1}-z^{-2}}{1-0.4z^{-1}-0.45z^{-2}}$$

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 2-18
num=[0.45 0.4 -1];
den=[1 -0.4 -0.45];

x0=[1 2];
y0=[0 1];
N=50;
n=[0: N-1]';
x=0.8.^n
Zi=filtic(num,den,y0,x0);
[y,Zf]=filter(num,den,x,Zi);
plot(n,x,'r-',n,y,'b--');
title('Response')
xlabel('n');ylabel('x(n)-y(n)');
legend('Input x','Output y',1);
grid
```

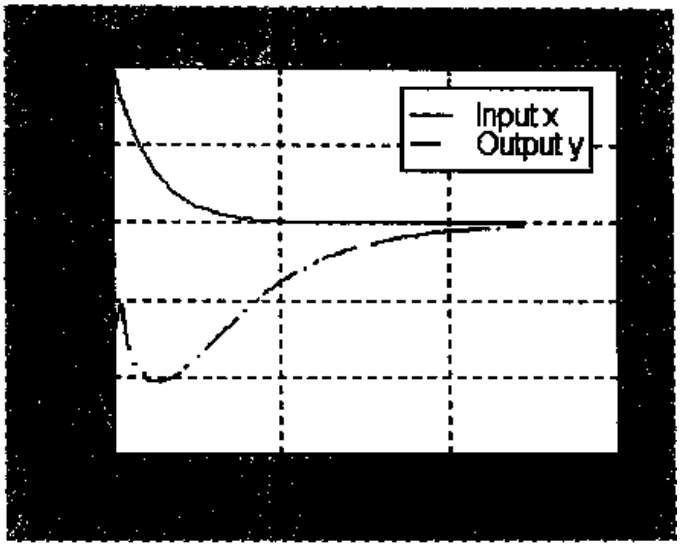


图 2.12 差分方程解

程序运行结果如图 2.12。y 曲线为其解。

2.4 离散傅里叶变换和 z 变换

2.4.1 z 域采样——由 z 变换到离散傅里叶变换

若有限序列 x(n)的长度为 N,其 z 变换为

$$X(z)=\sum_{n=0}^{N-1}x(n)z^{-n} \tag{2.4-1}$$

令 $z=e^{j(\frac{2\pi}{N})k}$,代入式(2.4-1),则有

$$X(z)\big|_{z=e^{j(\frac{2\pi}{N})k}}=\sum_{n=0}^{N-1}x(n)e^{-j(\frac{2\pi}{N})nk}$$

即

$$X(k)=X(z)\big|_{z=e^{j(\frac{2\pi}{N})k}} \tag{2.4-2}$$

而 $z=e^{j(\frac{2\pi}{N})k}(k=0,1,\cdots,N-1)$,表示在 z 平面上单位圆上 N 个等分上的第 k 个样点,因此式(2.4-2)表明,有限长序列 x(n)的离散傅里叶变换 X(k)是它的 z 变换在单位圆上 N 个等分点上的采样值。常称频域采样。

值得注意的是 X(k)=DFT(x(n)),而 $X(z)=\mathcal{Z}[x(n)]$,$X(k)\neq X(z)\big|_{z=k}$.

2.4.2 z 域重构——由离散傅里叶变换到 z 变换

对于有限长序列 x(n),若已知它的傅里叶变换 X(k),如何确定该序列的 z 变换 X(z),从而不失真地恢复序列 x(n)。

由 2.1 节可知,在满足频域采样定理条件下,有

$$X(z)=\sum_{k=0}^{N-1}X(k)\left[\frac{1}{N}\frac{1-z^{-N}}{1-W^{-k}z^{-1}}\right] \tag{2.4-3}$$

式中

$$W=e^{-j(\frac{2\pi}{N})}$$

或

$$X(z)=\sum_{k=0}^{N-1}X(k)\left[\frac{1}{N}\frac{1-z^{-N}}{1-e^{j(\frac{2\pi}{N})k}z^{-1}}\right]$$

这就是由 z 平面单位圆上采样点值 X(k)确定 X(z)的表达式,也称"内插公式"。或写为

$$X(z)=\sum_{k=0}^{N-1}X(k)\phi_k(z) \tag{2.4-4}$$

式中

$$\phi_k(z)=\frac{1}{N}\frac{1-z^{-N}}{1-W^{-k}z^{-1}}$$

式(2.4-2)描述由傅里叶变换 X(k)至 z 变换 X(z)的转换关系。

当然,已知有限长序列 x(n)的傅里叶变换 X(k),也可以由傅里叶逆变换求得 $\tilde{x}(n)$,$\tilde{x}(n)$为周期序列。由频域采样定理可以推论:

对于 x(n)是有限长序列,序列长度为 N,M 点频域采样 X(k)的逆变换 $\tilde{x}(n)$不失真地代表序列 x(n)特征的条件是

$$M\geqslant N \tag{2.4-5}$$

若 x(n)是无限长序列,则随着频率采样点数 M 的增加,逐渐逼近 x(n)。下面给出一个例子,说明 DFT 和 ZT 的关系并证实频率采样定理的正确性。

【例 2.19】 已知序列的 $x(n)=(0.9)^nu(n)$,①求序列的 z 变换;②当频率采样点数 M=5,10,20,25 时,序列的傅里叶变换 X(k)及其逆变换 $\tilde{x}(n)$;③比较 M 对 $\tilde{x}(n)$的影响。

由 z 变换公式

$$X(z)=\frac{1}{1-0.9z^{-1}}=\frac{z}{z-0.9},\qquad |z|>0.9$$

$$X(k)=X(z)|_{z=e^{j\frac{2\pi}{N}k}}$$

并由逆变换计算 x(n)，即 $\tilde{x}(n)$=IDFT(X(k)).

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 2-19
clf
a=0.8;
Nsum=40;
n=0:Nsum-1;
for i=1:4
switch i
case 1
    N=5;
case 2;
    N=10;
case 3
    N=20;
case 4
    N=40;
end
k=0:N-1;
zk=exp(j*2*pi*k/N);
Xk=zk./(zk-a);
xnN=real(idft(Xk,N));
xn=xnN'*ones(1,Nsum/N);
xn=(xn(:))';
gsptext=['22'int2str(i)];
subplot(gsptext)
stem(n,xn);axis([0,40,-0.1,1.5]);
xtext=['IDFT N='int2str(N)];
xlabel('n');ylabel('x(n)');title(xtext);
hold on
plot(n,zeros(1,length(n)));
hold off
end
```

程序运行结果如图 2.13 所示。

由图 2.13 可见，当频域采样点数 N 小于 10 时，引起较大的时域混叠，恢复的 x(n)会

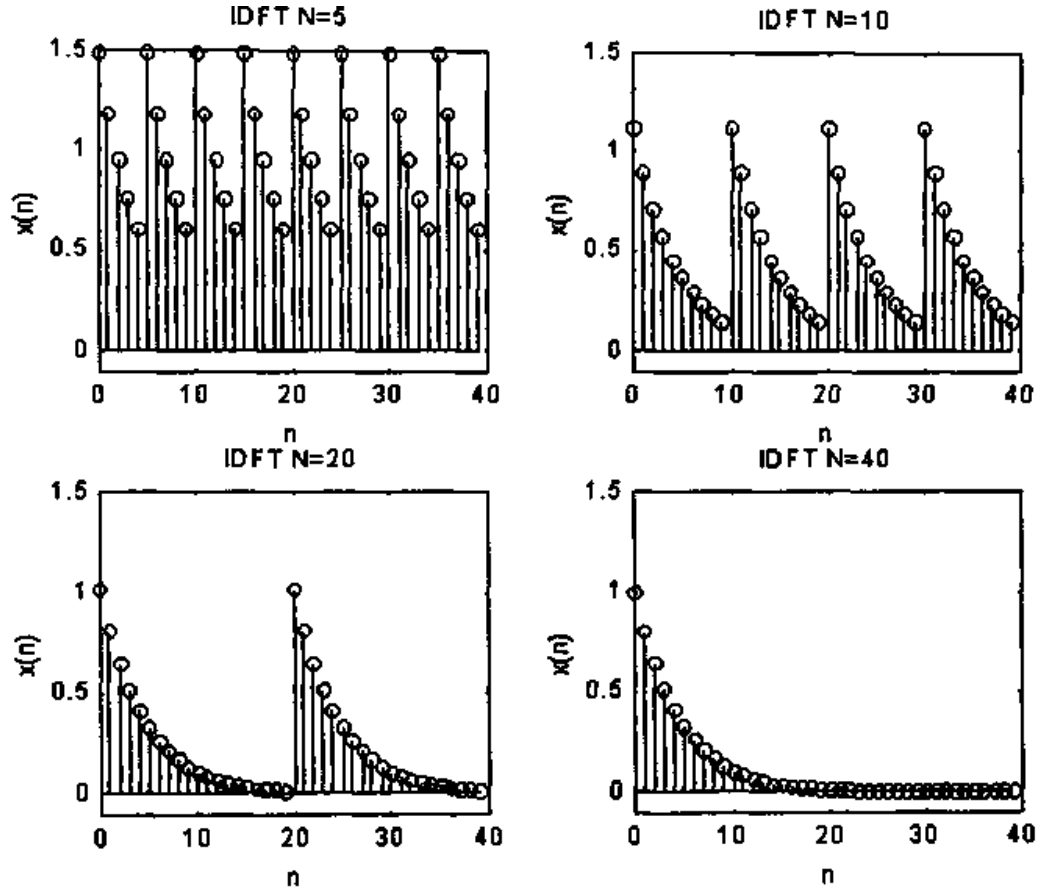


图 2.13　不同频域采样的时域混叠

产生较大失真。当 N 较大时，不会引起明显的混叠。

2.5　离散时间系统的频率响应

在 1.5 节中讨论了离散时间系统响应及时域响应求解的 MATLAB 实现。本节重点讨论离散时间系统的频率响应及求解的 MATLAB 实现。

在第一章已讨论过，任何离散时间系统可以由 z 传递函数来描述。

$$H(z)=\frac{Y(z)}{X(z)}=\frac{num(z)}{den(z)} \tag{2.5-1}$$

令 $z=e^{j\omega}$，可得

$$H(e^{j\omega})=\frac{Y(j\omega)}{X(j\omega)}=\frac{num(j\omega)}{den(j\omega)} \tag{2.5-2}$$

式中，$H(e^{j\omega})$为离散时间系统的频率响应，若已知离散时间系统的 z 的传递函数，求系统频率特性，可以利用下式

$$H(e^{j\omega_k})=H(z)|_{z=e^{j\omega_k}} \tag{2.5-3}$$

式中，k=0,1,…,k 分布在 0～π 之间。$|H(e^{j\omega})|$为系统幅频响应，$\angle H(e^{j\omega})$为系统相频响应。

系统的频率响应 $H(e^{j\omega})$以及相应的幅频响应 $|H(e^{j\omega})|$和相频响应 $\angle H(e^{j\omega})$都是数字角频率的函数，而且是以 2π 为周期的周期函数。

【例 2.20】　已知系统差分方程为

$$y(n) = x(n) + 2x(n-1) + x(n-2) - 0.5y(n-1) - 0.25y(n-2)$$

绘制系统的幅值响应和相位响应曲线。

对系统差分方程两边进行 z 变换可得系统传递函数为

$$H(z) = \frac{1 + 2z^{-1} + z^{-2}}{1 + 0.5z^{-1} + 0.25z^{-2}}$$

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 2-20
clf
b=[1 2 1];
a=[1 0.5 0.25 ];
m=0:length(b)-1;
n=0:length(a)-1;
K=512;
k=1:K;
w=pi*k/K;
num=b*exp(-j*m'*w);
den=a*exp(-j*n'*w);
H=num./den;
magH=20*log(abs(H));
phaH=angle(H);
subplot(221)
plot(w/pi,magH);
xlabel('w (pi)');ylabel('|H(jw)| dB');
title('Magnitude');
grid

subplot(222)
plot(w/pi,phaH*180/pi);
xlabel('w (pi)');ylabel('<H(jw) degree');
title('Phase')
grid
```

程序运行结果绘制幅值响应和相位响应如图 2.14 所示。

离散时间系统频率特性另一种求解方法是直接利用 MATLAB 信号处理工具箱函数 freqz.

函数 freqz 用于计算数字滤波器频率响应，也适用于类似离散时间系统。该函数调用格式见 5.2.2 节。

【例 2.21】 对上例中的系统，试用函数 freqz 绘制系统频率响应图并与例 2.20 的结果进行比较。

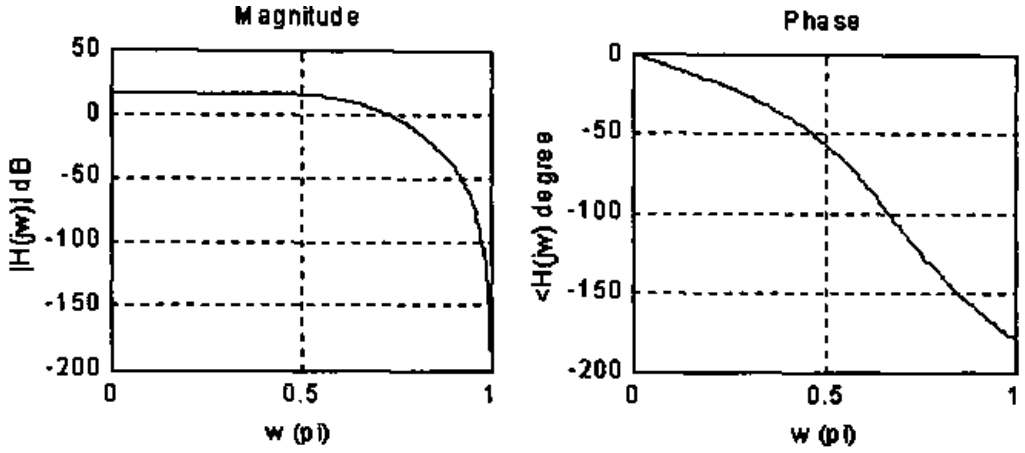


图 2.14　频率响应图

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 2-21
clf
b=[1 2 1];
a=[1 0.5 0.25 ];
K=512;
figure(1)
freqz(b,a,K)

[H,w]=freqz(b,a,K);
magH=20 * log(abs(H));
phaH=angle(H);
figure(2)
subplot(221)
plot(w/pi,magH);
xlabel('w (pi)');ylabel('|H(jw)|');
title('Magnitude');
grid

subplot(222)
plot(w/pi,phaH * 180/pi);
xlabel('w (pi)');ylabel('<H(jw) degree');
title('Phase')
grid
```

程序运行结果如图 2.15 所示。图中出现“Normalized freqency”(标准化频率)。关于 MATLAB 的“标准化频率”的定义的说明如下：

在 MATLAB 信号处理工具箱中，定义：单位频率为 Nyquist 频率。Nyquist 频率为采样频率的一半。信号处理工具箱采用的“标准化频率”为实际频率和 Nyquist 频率之比。如

一个系统采样频率为1000Hz，实际频率值为300Hz，转换为标准化频率为300/500=0.6。若已知标准化频率，将其转换为实际频率则乘以Nyquist频率(采样频率一半)。如一个系统采样频率为100Hz，标准化频率为0.8，则实际频率为0.8×50=40Hz，也可进一步转化为圆频率 $\omega=2\pi f$.

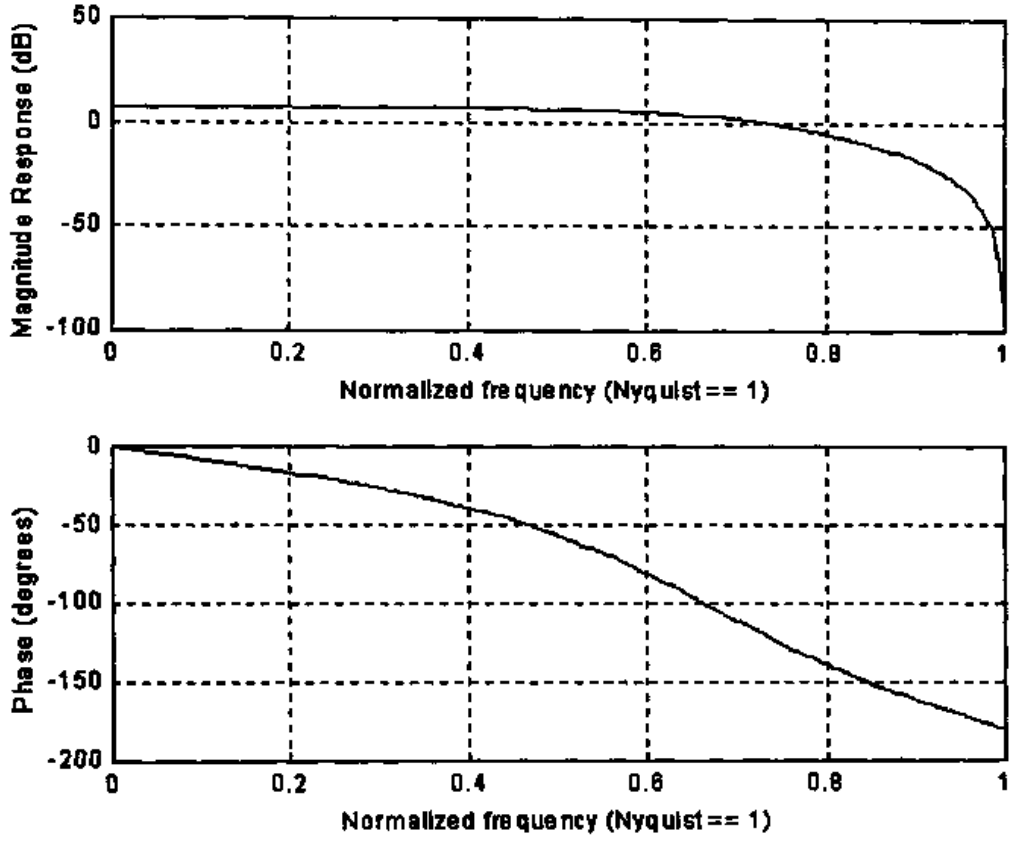


图 2.15 系统频率响应

由图 2.15 可知，用函数 freqz 绘制的频率响应图和图 2.14 的结果完全相同。

2.6 快速傅里叶变换(FFT)

2.6.1 基-2 FFT 算法

离散傅里叶变换(DFT)是利用计算机对信号进行频谱分析的理论依据。而快速傅里叶变换(FFT)是快速计算 DFT 的算法，使 DFT 这一理论得到广泛应用。

由式(2.2-11)离散傅里叶变换为

$$\mathrm{DFT}(x(n))=X(k)=\sum_{n=0}^{N-1}x(n)e^{-j\frac{2\pi}{N}nk},\quad k=0,1,\cdots,N-1$$

由 $W_N=e^{-j\frac{2\pi}{N}}$ 可得

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{nk},\qquad k=0,1,\cdots,N-1 \tag{2.6-1}$$

或写成矩形式

$$\begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & \cdots & W^0 \\ W^0 & W^{1\times1} & W^{2\times1} & \cdots & W^{(N-1)\times1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ W^0 & W^{1\times(N-1)} & W^{2\times(N-1)} & \cdots & W^{(N-1)\times(N-1)} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix} \tag{2.6-2}$$

考察上式，注意到下面特性：

(1)W^{nk}的周期性

$$W^{nk} = W^{n(k+N)} = W^{k(n+N)} \tag{2.6-3}$$

(2)W^{nk}的对称性

$$W^{(nk+\frac{N}{2})} = -W^{nk} \tag{2.6-4}$$

经周期性和对称性简化之后，上面矩阵[W]中许多元素相同，而使 DFT 运算过程大大简化。这就是 FFT 的基本思想。

FFT 的算法有许多种形式，但基本上可以分为时间抽取法和频率抽取法两大类。

一、时间抽取基-2FFT 算法

设序列长度 N 是 2 的整数幂次方，$N=2^M$。这样我们首先将序列 x(n)分解成两组：偶数项一组，奇数项一组，得到 2 个 N/2 的子序列，即

$$x_1(r) = x(2r)$$

$$x_2(r) = x(2r+1), \quad r = 0,1,\cdots,\frac{N}{2}-1$$

对两个子序列分别进行 DFT，可得

$$X_1(k) = DFT[x_1(r)]$$
$$X_2(k) = DFT[x_2(r)]$$

对于原序列 x(n)有

$$X(k) = DFT[x(n)], \qquad k = 0,1,\cdots,N-1$$

由 W_N^k 的性质可得

$$X(k) = X_1(k) + W_N^k X_2(k)$$

$$X\left(k+\frac{N}{2}\right) = X_1(k) - W_N^k X_2(k), \qquad k = 0,1,\cdots,\frac{N}{2}-1$$

研究表明，通过奇偶分解后 DFT 的计算工作量几乎节省一半。

由于 $N/2=2^{M-1}$仍然是偶数，可类似地将 N/2 点序列再分解为两个点 N/4 序列，……，继续这样分解过程，直到最后是 2 点序列的 DFT。由于每次分解均将序列从时域上按偶数奇数提取，所以称为时间抽取，且每次都是一分为二，所以称基数为 2(基-2)的算法。

时间抽取基-2 的 FFT 算法比普通的 DFT 算法运算量大大减小。例如：

一个 N 长序列，直接用 DFT 方法需要

复数乘法 N^2 次；复数加法 N(N−1)次。

而采用 FFT 则只需要

复数乘法 $\frac{N}{2}\log_2 N$ 次；复数加法 $N\log_2 N$ 次

二、频率抽取基-2FFT 算法

对于 $N=2^M$ 序列，频率抽取不是按偶、奇分解，而是按前后对半分解，并按一定规则组成两个点 N/2 序列，它们的 DFT 分别是 $X_1(r)$和 $X_2(r)$，它们分别对应于原序列 N 点序列的 DFT 的偶数部分和奇数部分。和时间抽取一样，由于 N/2 仍是偶数，继续上面的分解过程，直至最后剩下 2 点的 DFT。这种分解方法，由于每次都是按输出 X(k)在频域上顺序属于偶数还是奇数分成两组，故称基数为 2(基-2)的频率抽取法。和时间抽取基-2FFT算法一样，频率抽取基-2FFT 算法运算工作量比普通 DFT 算法大大减小。

由上可知，基-2FFT 算法均要求序列长度 N 是 2 的整数幂次方，即

$$N = 2^m$$

式中，m 为整数。

由此，N 通常取 128，256，1024，2048 等。

关于 FFT 算法请读者参考信号处理教材有关章节，这里不再重复。

2.6.2 基-2IFFT 算法

由式(2.2-12)，离散傅里叶逆变换为

$$\begin{aligned} x(n) = \mathrm{IDFT}[X(k)] &= \frac{1}{N}\sum_{k=0}^{N-1} X(k)e^{j\frac{2\pi}{N}nk} \\ &= \frac{1}{N}\sum_{k=0}^{N-1} X(k)W^{-nk}, \quad k = 0,1,\cdots,N-1 \end{aligned} \tag{2.6-5}$$

$$\begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & \cdots & W^0 \\ W^0 & W^{-1\times 1} & W^{-1\times 2} & \cdots & W^{-1\times(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ W^0 & W^{-(N-1)\times 1} & W^{-(N-1)\times 2} & \cdots & W^{-(N-1)\times(N-1)} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix} \tag{2.6-6}$$

比较式(2.6-6)和(2.6-2)可见，IDFT 算法和 DFT 的算法完全相似，快速傅里叶变换的算法完全适用于快速傅里叶逆变换(IFFT)。

关于 IFFT 算法，请读者阅读信号处理教材有关章节。

2.6.3 FFT 的 MATLAB 实现

在 MATLAB 的信号处理工具箱中函数 FFT 和 IFFT 用于快速傅里叶变换和逆变换。下面介绍这些函数。

函数 FFT 用于序列快速傅里叶变换。

函数的一种调用格式为

```
y = fft(x)
```

其中，x 是序列，y 是序列的 FFT，x 可以为一向量或矩阵，若 x 为一向量，y 是 x 的 FFT，且和 x 相同长度。若 x 为一矩阵，则 y 是对矩阵的每一列向量进行 FFT。

如果 x 长度是 2 的幂次方，函数 fft 执行高速基-2FFT 算法；否则 fft 执行一种混合基的离散傅里叶变换算法，计算速度较慢。

函数 FFT 的另一种调用格式为

```
y = fft(x,N)
```

式中，x，y 意义同前，N 为正整数。

函数执行 N 点的 FFT。若 x 为向量且长度小于 N，则函数将 x 补零至长度 N。若向量 x 的长度大于 N，则函数截短 x 使之长度为 N。若 x 为矩阵，按相同方法对 x 进行处理。

经函数 fft 求得的序列 y 一般是复序列，通常要求其幅值和相位。MATLAB 提供求复数的幅值和相位函数：ABS，ANGLE，这些函数一般和 FFT 同时使用。

函数 abs(x)用于计算复向量 x 的幅值，函数 angle(x)用于计算复向量 x 的相角，介于 $-\pi$和 π 之间，以弧度表示。

函数 unwrap(p)用于展开弧度相位角 p，当相位角绝对变化超过 π 时，函数把它扩展至 2π.

用 MATLAB 工具箱函数 fft 进行频谱分析时需要注意：

(1) 函数 fft 返回值 y 的数据结构对称性

若已知序列 x=[4,3,2,6,7,8,9,0]，求 X(k)=DFT[x(n)]。

利用函数 fft 计算，用 MATLAB 编程。

```
%MATLAB PROGRAM 2-22
N=8;
n=0:N-1;
xn=[4 3 2 6 7 8 9 0]';
Xk=fft(xn)
```

```
≫  smp222
Xk =
     39.0000
  -10.7782 + 6.2929i
    0.0000 - 5.0000i
    4.7782 - 7.7071i
    5.0000
    4.7782 + 7.7071i
    0.0000 + 5.0000i
  -10.7782 - 6.2929i
```

由程序运行所得结果可见，X(k)和 x(n)的维数相同，共有 8 个元素。X(k)的第一行元素对应频率值为 0，第五行元素对应频率为 Nyquist 频率，即标准频率值为 1。因此第一行至第五行对应的标准频率为 0～1。而第五行至第八行对应的是负频率，其 X(k)值是以 Nyquist 频率为轴对称。(注：通常表示为 Nyquist 频率外扩展，标以正值。)

一般而言，对于 N 点的 x(n)序列的 FFT 是 N 点的复数序列，其点 n=N/2+1 对应 Nyquist 频率，作谱分析时仅取序列 X(k)的前一半，即前 N/2 点即可。X(k)的后一半序列和前一半是对称的。

(2) 频率计算

若 N 点序列 x(n)(n=0,1,…,N−1)是在采样频率 f_s(Hz)下获得的。它的 FFT 也是

N 点序列，即 X(k)(k=0,1,2,…,N−1)，则第 k 点所对应实际频率值为 $f=k*f_s/N$.

(3) 作 FFT 分析时，幅值大小与 FFT 选择点数有关，但不影响分析结果。

【例 2.23】 已知信号由 15Hz 幅值 0.5 的正弦信号和 40Hz 幅值 2 的正弦信号组成，数据采样频率为 100Hz，试分别绘制 N=128 点 DFT 的幅频图和 N=1024 点 DFT 幅频图。

由题意，信号可写为 $x=0.5\sin(2\pi f_1 t)+2\sin(2\pi f_2 t)$，其中，$f_1=15Hz$，$f_2=40Hz$。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 2-23
clf
fs=100;
N=128;
n=0:N-1;
t=n/fs;
x=0.5*sin(2*pi*15*t)+2*sin(2*pi*40*t);
y=fft(x,N);
mag=abs(y);
f=(0:length(y)-1)'*fs/length(y);
subplot(221)
plot(f,mag);
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('N=128')
grid

subplot(222)
plot(f(1:N/2),mag(1:N/2));
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('N=128')
grid

fs=100;
N=1024;
n=0:N-1;
t=n/fs;
x=0.5*sin(2*pi*15*t)+2*sin(2*pi*40*t);
y=fft(x,N);
mag=abs(y);
```

```
f=(0:length(y)-1)' * fs/length(y);
subplot(223)
plot(f,mag);
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('N=1024')
grid

subplot(224)
plot(f(1:N/2),mag(1:N/2));
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('N=1024')
grid
```

程序运行结果如图 2.16 所示。

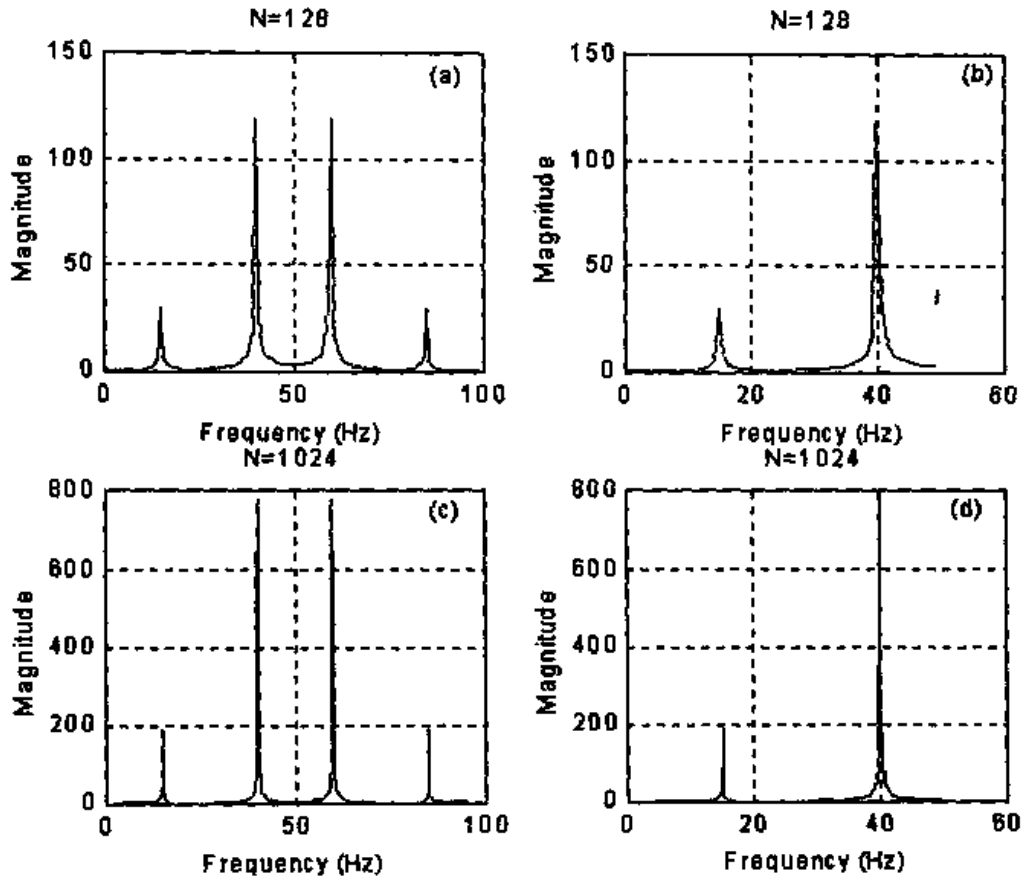


图 2.16 幅频谱图

图 2.16(a)、(b)为 N=128 点幅频谱图，(c)、(d)为 N=1024 点幅频谱图。由于采样频率 f_s=100Hz，故 Nyquist 频率为 $f_s/2$=50Hz。(a)、(c)是 0～100Hz 频谱图，(b)、(d)是 0～50Hz频谱图。由(a)或(c)可见，整个频谱图是以 Nyquist 频率为轴对称的。因此利用 fft 对信号作谱分析，只要考察 0～Nyquist 频率(采样频率一半)范围的幅频特性。比较(a)和

(c)或(b)和(d)可见，幅值大小与 fft 选用点数 N 有关，但只要点数 N 足够不影响研究结果。从图 2.16 幅频谱图可见，信号中包括 15Hz 和 40Hz 的正弦分量。

【例 2.24】 已知带有测量噪声信号 $x(t)=\sin(2\pi f_1 t)+\sin(2\pi f_2 t)+2\omega(t)$，其中，$f_1=50Hz$，$f_2=120Hz$，$\omega(t)$为均值为零的随机信号，采样频率 1000Hz，数据点数 N=1024。试绘制信号的频谱图和无噪声信号的频谱图。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 2-24
clf;
fs=1000;
N=1024;
n=0:N-1;
t=n/fs;
f1=50;f2=120;
%Signal with Noise
x=sin(2*pi*f1*t)+sin(2*pi*f2*t);
x=x+2*randn(1,length(t));
y=fft(x,N);
mag=abs(y);
f=(0:length(y)-1)'*fs/length(y);
subplot(211)
plot(f(1:N/2),mag(1:N/2));
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('N=1024 With Noise')
grid
%Signal without Noise
x=sin(2*pi*f1*t)+sin(2*pi*f2*t);
y=fft(x,N);
mag=abs(y);
f=(0:length(y)-1)'*fs/length(y);
subplot(212)
plot(f(1:N/2),mag(1:N/2));
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('N=1024 Without Noise')
grid
```

程序运行结果如图 2.17 所示。本例中，由于采样频率为 1000Hz，选择频谱图频率范围为 0～500Hz。由图 2.17 可见，通过 FFT 分析可将周期性信号从噪声中提取出来，这一

点在工程上应用十分广泛。

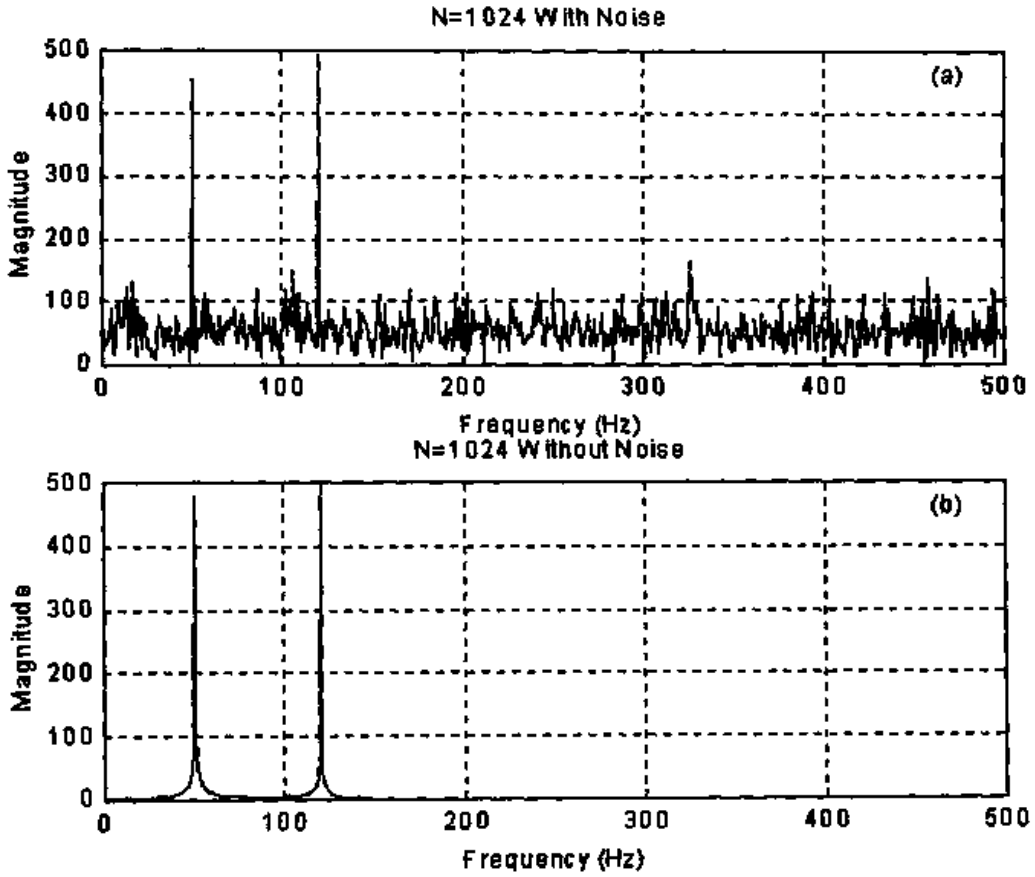


图 2.17 带噪声信号的幅频谱图

下面的例子进一步说明序列 x(n)长度 Ndate 和 FFT 的长度 Nfft 对信号频谱的影响。

【例 2.25】 已知信号 $x(t)=0.5\sin(2\pi f_1 t)+2\sin(2\pi f_2 t)$，其中，$f_1=15Hz$，$f_2=40Hz$，采样频率为 100Hz，在下列情况下绘制其幅频谱。

(1) Ndate=32, Nfft=32;

(2) Ndate=32, Nfft=128;

(3) Ndate=136, Nfft=128;

(4) Ndate=136, Nfft=512。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 2-25
clf
fs=100;
%Length of Data
Ndata=32;
%Length of FFT
N=32;
n=0:Ndata-1;
t=n/fs;
x=0.5*sin(2*pi*15*t)+2*sin(2*pi*40*t);
```

```
y=fft(x,N);
mag=abs(y);
f=(0:length(y)-1)' * fs/length(y);
subplot(221)
plot(f(1:N/2),mag(1:N/2));
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('Ndata=32 Nfft=32')
grid
```

程序运行结果如图 2.18 所示。

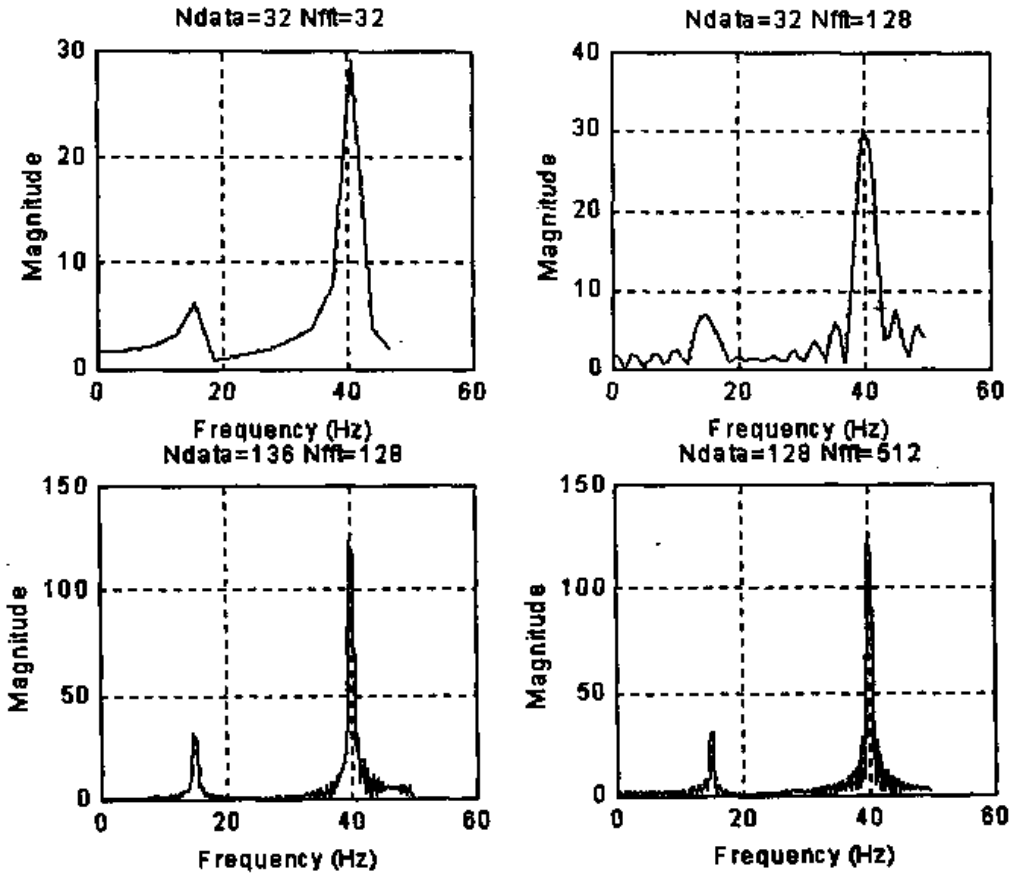


图 2.18　不同序列和 DFT 长度时的频谱

比较图 2.18 和图 2.16 可见，对信号作频谱分析时数据样本应有足够长度，一般应取 DFT 的点数和数据样本相同，这样频谱图具有较高质量，减小因补零或截断而产生的影响。

函数 IFFT 用于一维快速傅里叶逆变换。函数调用格式为

$$y = \text{ifft}(x)$$

$$y = \text{ifft}(x,N)$$

其中，x 是某序列的傅里叶变换 X(k)；y 是 IDFT[X(k)]，返回一个复数序列；n 为采用的 IDFT 的点数。

当 N 小于 x 长度时，对 x 进行截断；当 N 大于 x 长度时，对 x 进行补零。

【例 2.26】 对信号 x(t)=sin(2π * 40t)+ sin(2π * 15t)进行 DFT,对其结果进行 IDFT,并将 IDFT 的结果和原信号进行比较。

```
%MATLAB PROGRAM 2-26
clf
fs=100;
%Length of FFT
N=128;
n=0:N-1;
t=n/fs;
x=sin(2 * pi * 40 * t)+sin(2 * pi * 15 * t);        % 生成原始信号
subplot(221)
plot(t,x);
xlabel('t (s)');
ylabel('x');
title('Original signal')
grid

y=fft(x,N);
mag=abs(y);                                          % 原始信号 FFT
f=(0:length(y)-1)' * fs/length(y);
subplot(222)
plot(f(1:N/2),mag(1:N/2));
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('FFT to original signal')
grid

xifft=ifft(y);
magx=real(xifft);                                    % 用 IFFT 恢复原始信号
ti=[0:length(xifft)-1]/fs;
subplot(223)
plot(ti,magx);
xlabel('t (s)');
ylabel('x');
title('Signal from IFFT')
grid

yif=fft(xifft,N);                                    % 恢复的信号 FFT
```

```
mag=abs(yif);
f=(0:length(y)-1)' * fs/length(y);
subplot(224)
plot(f(1:N/2),mag(1:N/2));
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('FFT to signal from IFFT')
grid
```

程序运行结果如图 2.19 所示。结果显示,原信号和由函数 ifft 恢复信号的时域和频域特性完全相同。

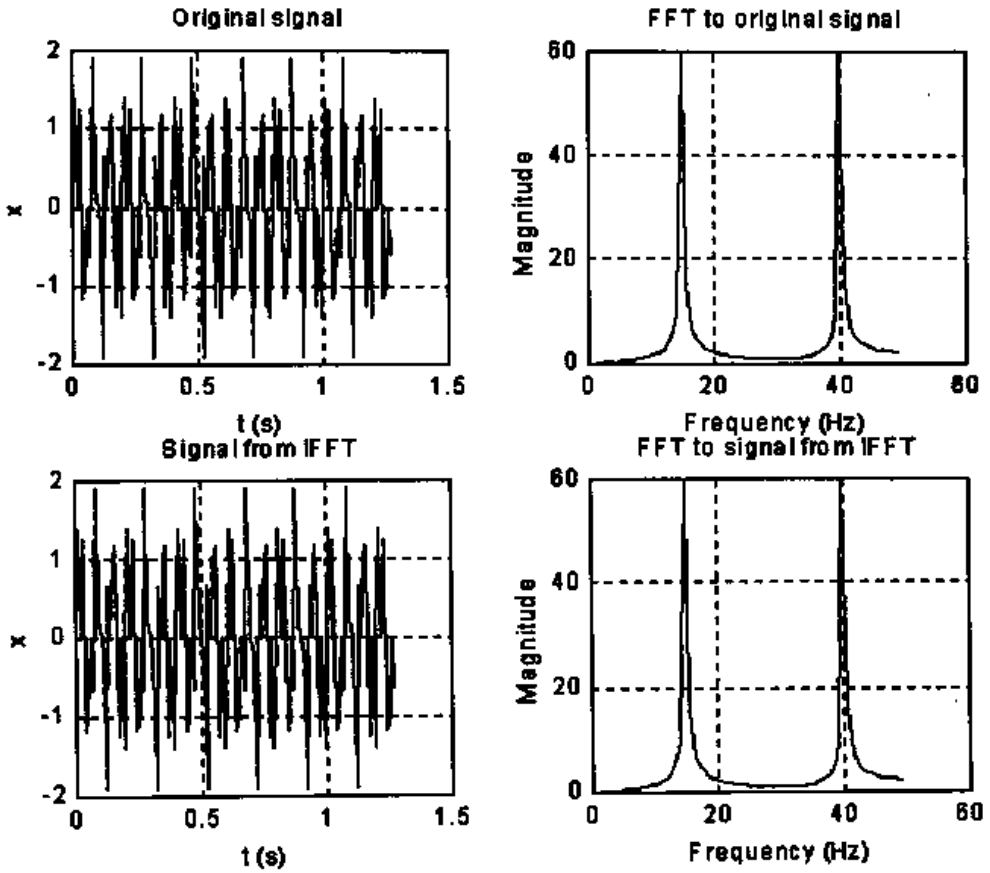


图 2.19 原始信号和恢复信号比较

2.6.4 线性卷积的 FFT 算法

前面已讲过,MATLAB 实现卷积的函数为 CONV,对于 N 值较小的向量,这是十分有效的。对于 N 值较大的向量卷积可用 FFT 加快计算速度。

由 DFT 性质可知,若 $DFT[x_1(n)]=X_1(k)$,$DFT[x_2(n)]=X_2(k)$,则

$$x_1(n) * x_2(n) = IDFT[X_1(k) \cdot X_2(k)] = IDFT[DFT[x_1(n)] \cdot DFT[x_2(n)]]$$

若 DFT 和 IDFT 均采用 FFT 和 IFFT 算法,可提高卷积速度。

因此,计算 $x_1(n)$和 $x_2(n)$的线性卷积的 FFT 算法可由下面步骤实现:

(1) 计算 $X_1(k)=FFT[x_1(n)]$;

(2) 计算 $X_2(k)=FFT[x_2(n)]$;

(3) 计算 $Y(k)=X_1(k)\cdot X_2(k)$;

(4) 计算 $x_1(n)*x_2(n)=IFFT[Y(k)]$。

【例 2.27】 用函数 conv 和 FFT 计算长为 1000 序列的卷积，比较其计算时间。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 2-27
clf;
L=5000;
N=L*2-1;
n=1:L;
x1=0.5*n;
x2=2*n;
%Compute convotion x1*x2 using function CONV
t0=clock;
yc=conv(x1,x2);
tc=etime(clock,t0)
%Compute convotion x1*x2 using FFT
t0=clock;
yf=ifft(fft(x1,N).*fft(x2,N));
tf=etime(clock,t0)
n1=0:length(yf)-1;
plot(n1,yc,'r',n1,abs(yf),'b')
```

程序运行结果表明：两种方法计算结果相同。但用函数 conv 计算卷积需 6.1500s，而用 FFT 计算卷积只需 0.9900s。

2.6.5 特殊变换

MATLAB 信号处理工具箱除了提供 DFT 函数外，还提供以下特殊变换函数。

一、Chirp-Z 变换

正如前面所述，DFT(FFT)计算有限长度 N 的序列的 X(k)值，也是在 z 平面单位圆上 N 个等间隔点上的频率采样值。在许多情况下，需要的不是单位圆上全部频谱，而只是某一频带内频谱。或者序列谱不在单位圆上，z 变换不是沿单位圆，而是沿 z 平面某一给定的螺旋线上等角度间隔采样。这时，不能直接利用 DFT。Chirp-Z 变换就是计算序列在 Z 平面上沿给定螺旋线进行的 z 变换。通过 Chirp-Z 变换间接对信号进行分析。Chirp-Z 变换在语音信号分析及其他高分辨率窄带谱分析中得到广泛应用。关于 Chirp-Z 变换(CZT)的基本原理和方法请读者查阅数字信号处理教材有关章节。

MATLAB 信号处理工具箱提供用于计算 Chirp-Z 变换函数 CZT。

函数 CZT 用于计算信号 x 的 Chirp-Z 变换。函数调用格式为

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

其中，x 为信号向量，也可为矩阵；m 为变换长度，w 为沿 z 平面螺旋线点之间比例；a 为螺旋线开始点，复数。这样，z 平面上的螺旋线为

```
z = a * (w. ^ -(0:m-1))
```

函数的另一种调用格式为

```
y = czt(x)
```

函数采用缺省值：

- m=length(x)
- w=exp(j*2*pi/m)
- a=1

这样相当于执行 DFT(x)。

【例 2.28】 用 FFT 和 CZT 分析一个低通滤波器的频率响应并比较结果。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 2-28
%Chirp-z Transform
%Create a lowpass FIR filter
h=fir1(30,125/500,boxcar(31));
%Input frequency and Create CZT parameter
Fs=1000;f1=100;f2=150;      % in Hertz
m=1024;
w=exp(-j*2*pi*(f2-f1)/(m*Fs));
a=exp(j*2*pi*f1/Fs);
%Compute DFT and CZT
y=fft(h,1000);
z=czt(h,m,w,a);
%Create frequency and compare the results
subplot(221)                                    % 绘制 DFT 频谱图
fy=(0:length(y)-1)'*1000/length(y);
plot(fy(1:500),abs(y(1:500)));
xlabel('Freqeucy (Hz)');ylabel('Magnitude')
title('FFT');grid;
axis([1 500 0 1.2]);
subplot(222)                                    % 绘制 CZT 频谱图
fz=((0:length(z)-1)'*(f2-f1)/length(z))+f1;
plot(fz,abs(z));
xlabel('Freqeucy (Hz)');ylabel('Magnitude')
title('CZT');grid;
axis([f1 f2 0 1.2]);

%Plot the z-plane spiral contour                % 绘制 Z 平面上的螺旋线
subplot(223)
```

```
zspiral=a*(w.^(-(0:m)));
zplane([],z.');
```

程序运行结果如图 2.20 所示。结果显示采用 CZT 使在频带 100Hz 至 150Hz 之间频谱分析有更高的频率分辨率。

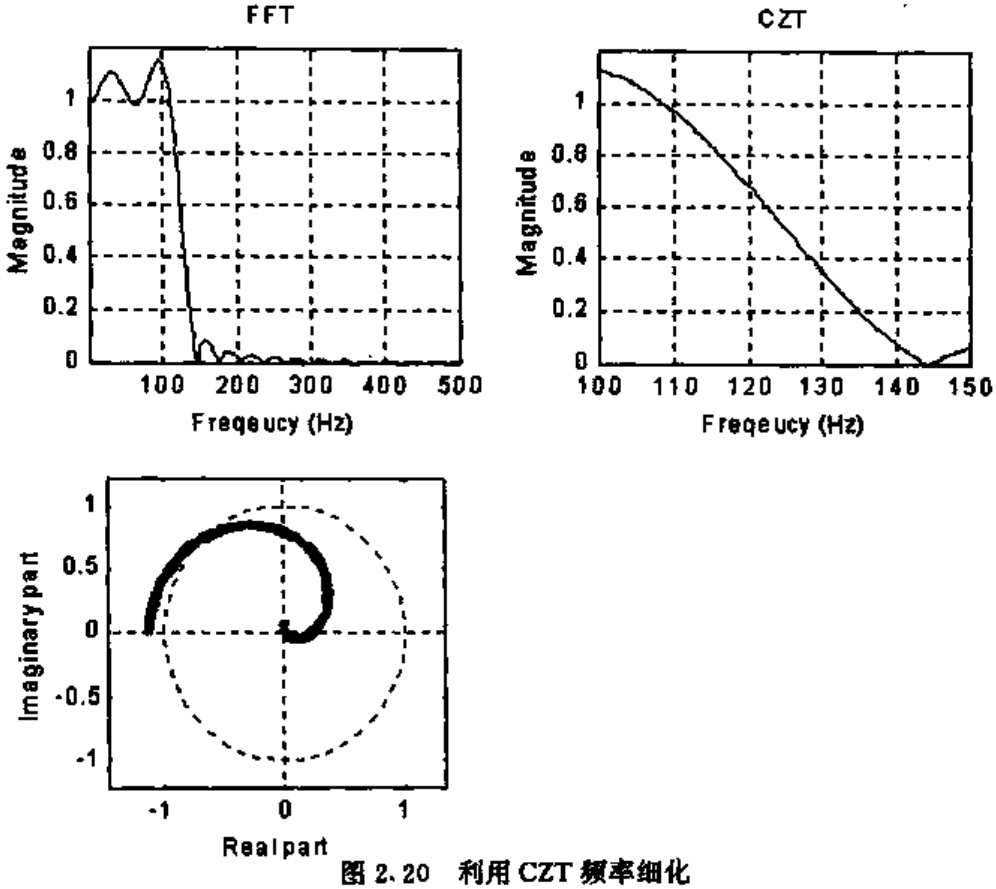


图 2.20　利用 CZT 频率细化

二、离散余弦变换

序列 x(n)的离散余弦变换(DCT)为

$$Y(k)=\sum_{n=0}^{N}w(n)x(n)\cos\frac{\pi k(2n+1)}{2N},\quad k=0,1,2,\cdots,N-1$$

式中

$$w(n)=\begin{cases}\dfrac{1}{\sqrt{N}}, & n=0;\\[2ex] \sqrt{\dfrac{2}{N}}, & \text{其他。}\end{cases}$$

离散余弦逆变换(IDCT)为

$$x(n)=\sum_{k=0}^{N}w(k)Y(k)\cos\frac{\pi k(2n+1)}{2N},\quad n=0,1,2,\cdots,N-1 \qquad (2.6\text{-}7)$$

式中

$$w(k)=\begin{cases}\dfrac{1}{\sqrt{N}}, & k=0;\\[2ex] \sqrt{\dfrac{2}{N}}, & \text{其他。}\end{cases}$$

和 DFT 一样,MATLAB 提供离散余弦变换函数 DCT 和逆变换函数 IDCT。

函数 DCT 用于序列的离散余弦变换。调用格式为

$$y = dct(x)$$

其中，x 为信号时间序列，向量或矩阵；y 为 x 的离散余弦变换，和 x 同维向量或矩阵。

函数的另一种调用方式为

$$y = dct(x,N)$$

其中，N 为 DCT 变换长度。函数根据序列 x 长度，补零或截断，使 x 长度变为 N。

序列离散余弦变换具有较好的能量密实性质。我们可以利用很少的 DCT 系数精确地重构原信号序列。这里首先要解决的问题是需要多少个 DCT 系数代表序列，才能使 IDCT 能够较精确地恢复原信号序列。下面给出一个例子。

【例 2.29】 已知信号 $x(t)=\sin(2\pi ft)$，$f=15Hz$，采样频率为 1000Hz，试确定表示该信号 99%能量的 DCT 系数的个数。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 2-29
t=[0:1/999:1];
x=sin(2*pi*15*t);
%DCT to the sequency and reconstruct the signal
y=dct(x);
[yy,ind]=sort(abs(y));
ind=fliplr(ind);
i=1;
while(norm([y(ind(1:i)) zeros(1,100-1)])/norm(y)<0.99)
   i=i+1;
end
i
```

程序运行结果：i=15

下面例子说明，如果用 15 个 DFT 元素重构信号，考察重构信号精度与所用元素数量关系。

【例 2.30】 用 DFT 技术重构上例的信号，并比较所用 DFT 元素数目 N 对重构信号的精度影响。（试取 N=4,7,16）

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 2-30
clf
%Create sinusoidal sequence,sampling frequency 1000 Hz
t=[0:1/999:1];
x=sin(2*pi*15*t);
%DCT to the sequency and reconstruct the signal
y=dct(x);
y2=find(abs(y)<1.05);
```

```
disp('Number of the points for reconstruct signal:')
Nidct=length(y)-length(y2)
y(y2)=zeros(size(y2));
z=idct(y);
%Output and compare
subplot(221)
plot(t,x)
axis([0 1 -2 2]);grid
title('Original signal');
subplot(222)
plot(t,z);
axis([0 1 -2 2]);grid
title('Reconstruct signal N=16')
```

程序运行结果如图 2.21 所示。

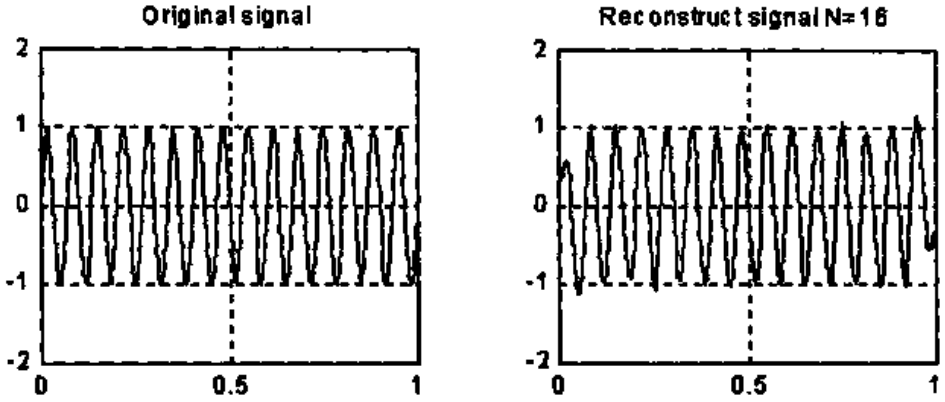


图 2.21 利用 DCT 重构信号

由图 2.21 可见，重构信号只需用 15 个以上 DCT 序列元素，就能较好地用 IDCT 将信号恢复；若所用元素较少，重构的信号精确度则较低。

三、希尔伯特(Hilbert)变换

希尔伯特变换的定义是：在区间$-\infty<t<\infty$内给出的实函数 x(t)，它的希尔伯特变换为

$$H[x(t)]=\frac{1}{\pi}\int_{-\infty}^{\infty}\frac{x(\tau)}{t-\tau}d\tau$$

MATLAB 信号处理工具箱提供变换函数 HILBERT。函数 HILBERT 用于计算 Hilbert 变换。调用格式为

```
y = hilbert(x)
```

其中，x 为实数序列；y 为复螺旋线序列，常称为解析信号。

解析信号的实部是原始实序列，而虚部是实际的 Hilbert 变换，是原实序列 90°相移而产生的序列。若原序列 x 是正弦，则 y 的虚部是余弦。Hilbert 变换序列 y 具有和原序列相同的幅值和频率成分，也包含原序列的相位信息。

【例 2.31】 已知原始信号 x=sin(2πft)，f=60Hz，求该信号的 Hilbert 变换。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 2-31
%Hilbert Transform
t=0:1/1023:1;
x=sin(2 * pi * 60 * t);
y=hilbert(x);
plot(t(1:50),real(y(1:50)),'r—');
hold on
plot(t(1:50),imag(y(1:50)),'b——');
legend('portion of data','Hilbert transform',4)
grid
```

程序运行结果如图 2.22 所示。由图可见，Hilbert 变换序列相对原信号序列滞后 90°。

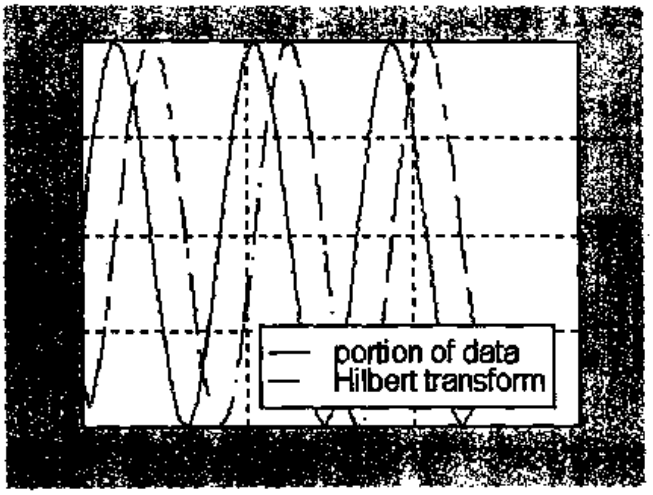


图 2.22 Hilbert 变换

Hilbert 变换在计算原时间序列瞬态属性方面十分有用。

2.7 信号调制和解调

信号调制是用一个低频信号对高频信号的某一特征参数进行控制。低频信号称为信息信号(Message Signal)，高频信号称为载波(Carrier)。调制后的信号称为调制信号(Modulated Signal)。调制在时域上是使载波的某一特征参数随信息信号的变化而变化的过程，在频域上是一个移频过程。

调幅的类型有多种。一般载波信号用高频余弦，若调制的特征参数为幅值称为幅值调制；若调制的特征参数为频率称频率调制；若调制的特征参数为相位称为相位调制。

解调就是将已调制波恢复为原低频信息信号的过程，它是信号调制的逆过程。

MATLAB 信号处理工具箱提供信号调制和解调工具函数 MODULATE 和 DEMOD。这两个函数适用于通信仿真和信号的调制和解调处理。

函数 MODULATE 用于数字信号调制，调用格式为

$$y = \text{modulate}(x, F_c, F_s, 'method', opt)$$

$$[y,t] = \text{modulate}(x, F_c, F_s)$$

其中，x 为信息信号，F_c 为载波频率，F_s 为信号采样频率，method 为期望的调制方法，opt 是补足选择项，某些调制方法才要求此项，y 为调制信号，t 为函数计算时间间隔向量。

关于 method 有几下 8 种选择：

(1) amdsb—sc 或 am　幅值调制、双侧带、抑制载波

该方法执行算法为

$$y = x. * \cos(2 * pi * F_s * t)$$

(2) amdsb-tc　幅值调制、双侧带、传输载波

该方法执行算法为

$$y = (x\text{-}opt). * \cos(2 * pi * F_c * t)$$

若 opt 缺省，则 opt=min(min(x))

(3) amssb　幅值调制，单侧带

该方法执行运算为

$$y = x. * \cos(2 * pi * F_c * t) + \text{img}(\text{hilbert}(x)). * \sin(2 * pi * F_c * t)$$

式中，hilbert(x)是 x 赫尔伯特变换。

(4) fm　频率调制

该方法执行运算为

$$y = \cos(2 * pi * F_c * t + opt * \text{cumsum}(x))$$

式中，cumsum(x)是 x 积分的矩形近似，opt 为频率调制常数，缺省时 opt=pi/(max(max(x)))。

(5) pwm　脉宽调制，从输入 x 中产生脉宽调制信号

如 y=modulate(x,F_c,F_s,'pwm','centered')!!!

x 为脉冲序列，其元素的幅值在 0 和 1 之间，且定义了宽度占一个周期的比例。产生调制信号的脉冲中心在每个周期开始处。y 长度为 length(x) * F_s/F_c.

(6) ptm　脉冲时间调制，从输入 x 产生一个脉冲时间调制信号 y

输入 x 为脉冲序列，其元素必须在 0 和 1 之间，并定义每个脉冲的开始时刻在一个周期中的位置。

opt 为 0～1 之间的标量，用以定义调制信号 y 每个脉冲长度占一个周期的比例，缺省值为 0.1，y 长度为 length(x) * F_s/F_c.

(7) qam　正交幅值调制，用于从信号 x 和 opt 产生正交幅值调制。

函数执行运算为

$$y = x. * \cos(2 * pi * F_c * t) + opt. * \sin(2 * pi * F_c * t)$$

opt 必须具有与 x 相同的尺寸

函数 DEMOD 用于信号的解调处理，调用格式为

$$x = \text{demod}(y, F_c, F_s, 'method')$$

$$x = \text{demod}(y, F_c, F_s, 'method', opt)$$

x = demod(y, F_c, F_s,'centered')

[x1,x2] = demod(y, F_c, F_s,'qam')

其中,各项意义与函数 modulate 相同。

【例 2.32】 已知信号为频率 45Hz 正弦信号,载波频率为 200Hz,试用 MATLAB 函数产生幅值调制信号,并比较它们的时域、频域特征。

用 MATLAB 编写程序如下:

```
%MATLAB PROGRAM 2-32
%Amplitude Modulation
%Create Original signal
%==========
Fs=1000;
Fc=200;
N=1000;
n=0:N-1;
t=n/Fs;
x=sin(2*pi*45*t);
subplot(221)
plot(t,x);
xlabel('t(s)');ylabel('x');
title('Original Signal')
axis([0 0.1 -1 1])
%Estimate PSD for original signal x
%==================
Nfft=1024;
window=hamming(512);
noverlap=256;
dflag='none';
[Pxx,f]=psd(x,Nfft,Fs,window,noverlap,dflag);
subplot(222)
plot(f,Pxx)
xlabel('Frequency (Hz)');
ylabel('Power Spectrum(X)');
title('Original Signal')
grid
%Create Modulated signal y
%==========
y=modulate(x,Fc,Fs,'am');
subplot(223)
```

```
plot(t,y)
xlabel('t(s)');ylabel('y');
axis([0 0.1 -1 1]);
title('Modulated Signal')
%Estimate PSD for modulated signal y
%==================
[Pxx,f]=psd(y,1024,F,,window,noverlap,dflag);
subplot(224)
plot(f,Pxx)
xlabel('Frequency (Hz)');
ylabel('Power Spectrum(y)');
title('Modulated Signal')
grid
```

程序运行结果如图 2.23 所示。比较原信号和调制信号时域曲线可见两种完全不同，而在功率谱图上显示，调制信号相对原信号有一个频移，频移大小为载波频率 200Hz。

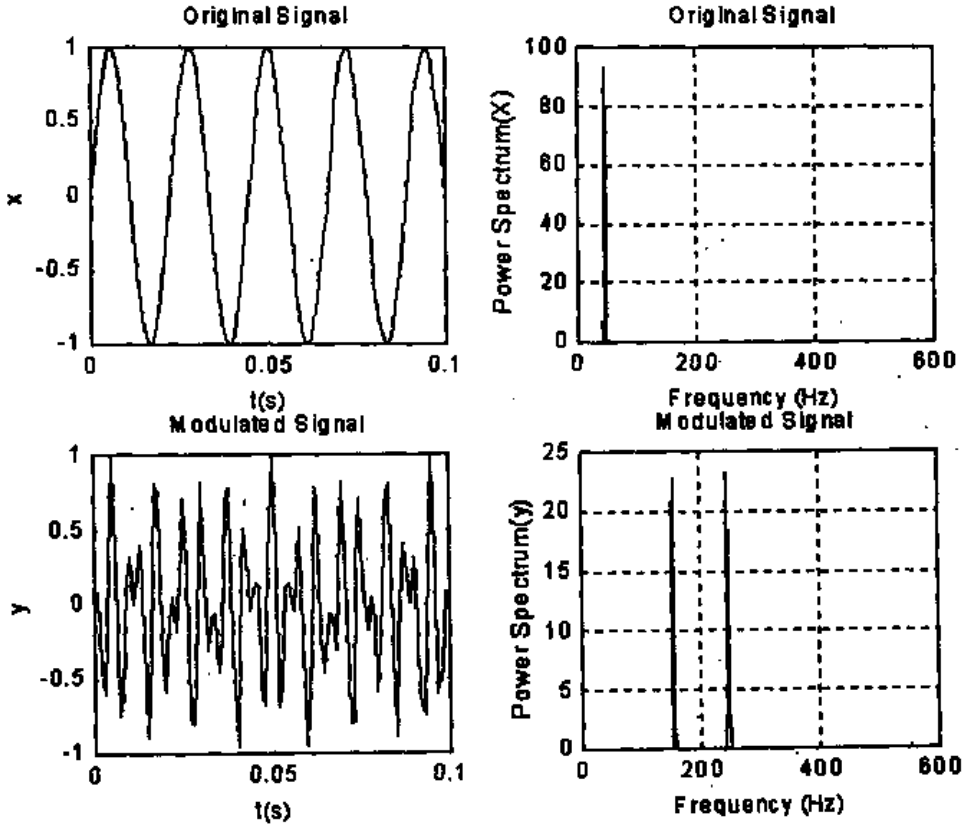


图 2.23 幅值调制

【例 2.33】 已知信号为频率 45Hz 正弦信号，载波频率为 200Hz，试用 MATLAB 函数产生频率调制信号，并比较它们的时域频域特征。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 2-33
%Frequency Modulation
%Create Original signal
%==========
Fs=1000;
Fc=200;
N=1000;
n=0:N-1;
t=n/Fs;
x=sin(2*pi*45*t);
subplot(221)
plot(t,x);
xlabel('t(s)');ylabel('x');
title('Original Signal')
axis([0 0.05 -1 1])
%Estimate PSD for original signal x
%==================
Nfft=1024;
window=hamming(512);
noverlap=256;
dflag='none';
[Pxx,f]=psd(x,Nfft,Fs,window,noverlap,dflag);
subplot(222)
plot(f,Pxx)
xlabel('Frequency (Hz)');
ylabel('Power Spectrum(X)');
title('Original Signal')
grid

%Create Modulated signal y
%==========
y=modulate(x,Fc,Fs,'fm');
subplot(223)
plot(t,y)
xlabel('t(s)');ylabel('y');
axis([0 0.05 -1 1]);
title('Modulated Signal')
%Estimate PSD for modulated signal y
```

```
%==================
[Pxx,f]=psd(y,1024,F,,window,noverlap,dflag);
subplot(224)
plot(f,Pxx)
xlabel('Frequency (Hz)');
ylabel('Power Spectrum(y)');
title('Modulated Signal')
grid
```

程序运行结果如图 2.24 所示。由图可见，调制信号的频率随信息信号的幅值变化。信

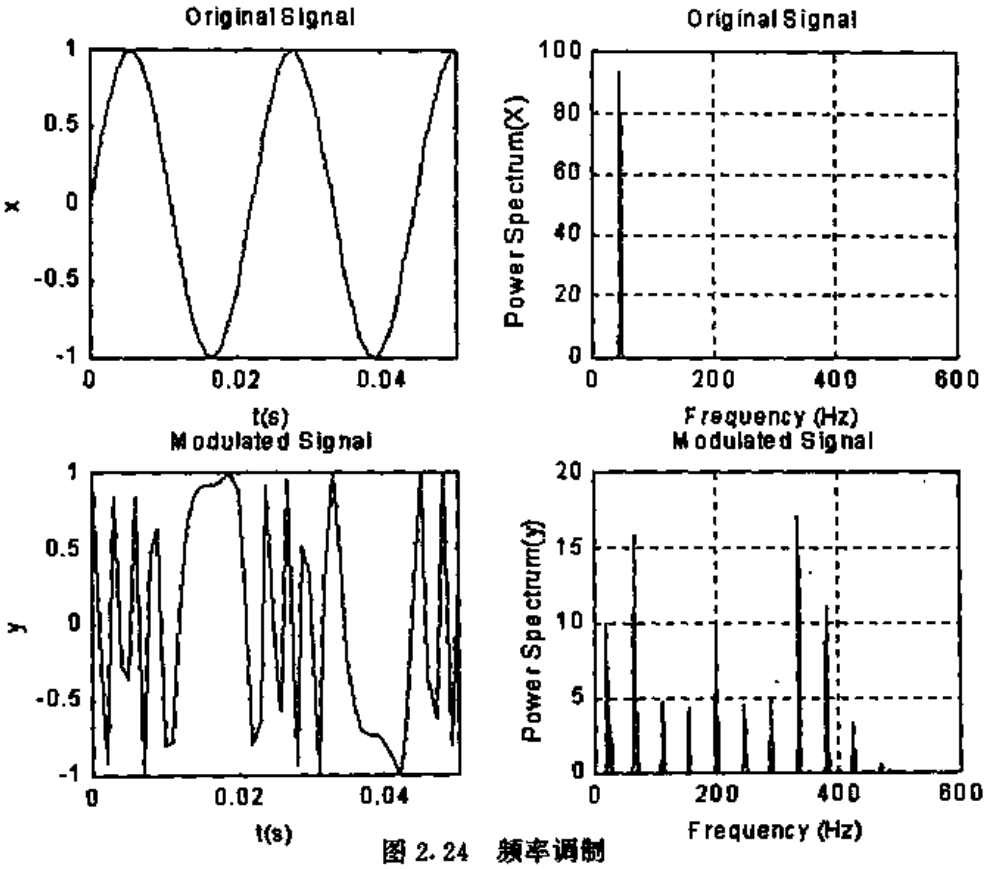


图 2.24 频率调制

息信号是单一的频率成分，而调制信号的频率成分很多。

习 题

2.1 求下列周期序列的 DFT：

(1) $\tilde{x}(n)=\{0, 0, 1, 1, 1\}$, N=5

(2) $\tilde{x}(n)=\{0, 0, 2, 0, 0\}$, N=5

(3) $\tilde{x}(n)=\{1, j, j, 1\}$, N=4

2.2 求下列有限长序列的 DFT 并绘制其幅值曲线：

(1) $x(n)=\delta(n)$, $\quad -10\leqslant n\leqslant 10$

(2) $x(n)=0.4^n$, $-10\leqslant n\leqslant 10$

(3) $x(n)=nR_{10}(n)$, $-20\leqslant n\leqslant 20$

(4) $x(n)=2\cos(0.2\pi n)[u(n)-u(n-10)]$, $0\leqslant n\leqslant 50$

2.3 已知有限长序列 x(n)的 DFF 为

$$X(k)=\begin{cases}\frac{N}{2}e^{j\theta}, & k=m,m\text{ 为正整数}\\ \frac{N}{2}e^{-j\theta}, & k=N-m,\ 0<m<\frac{N}{2}\\ 0, & \text{其他}\end{cases}$$

求 x(n)=IDFT[X(k)],并绘出 x(n)和|X(k)|图形。

2.4 已知序列 x(n)={1, 2, 3, 4, 5}, y(n)={1, 1, 1,1, 1}, 试求:

(1) 它们的圆周卷积(N=5);

(2) 它们的线性卷积;

(3) 比较圆周卷积和线性卷积结果;

(4) 如何用圆周卷积来求线性卷积而不失真。

2.5 已知两个有限长序列

$$x_1(n)=\cos\left(\frac{2\pi}{N}n\right)R_N(n)$$

$$x_2(n)=\sin\left(\frac{2\pi}{N}n\right)R_N(n)$$

试用直接卷积法和基于 DFT 的频域方法分别求 $x_1(n)$和 $x_2(n)$的圆周卷积

$$y(n)=x_1(n)\textcircled{N}x_2(n)$$

2.6 求以下序列的 z 变换及收敛域:

(1) $x(n)=2^nu(n)-2\left(\frac{1}{2}\right)^nu(n)$

(2) $x(n)=\frac{1}{2}\delta(n)+\delta(n-1)-\frac{1}{3}\delta(n-2)$

(3) $x(n)=3\left(\frac{1}{2}\right)^nu(n)+5\left(\frac{1}{4}\right)^nu(-n-1)$

(4) $x(n)=2\delta(n-2)+3u(n-3)$

2.7 利用部分分式展开法求下列逆 z 变换:

(1) $X(z)=\frac{2+3z^{-1}}{1-z^{-1}+0.81z^{-2}}$, $|z|>0.9$

(2) $X(z)=\frac{1-4z^{-1}+5z^{-2}}{1-6z^{-1}+11z^{-2}-6z^{-3}}$

(a) $1<|z|<3$ (b) $|z|>3$ (c) $|z|<1$

2.8 利用卷积定理和 IDFT 求下面序列的卷积 $y(n)=x(n)*h(n)$

(1) $x(n)=a^nu(n)$, $h(n)=\delta(n-2)$

(2) X(z)=[1, 2, 3, 4], H(z)=[4, 5, 6]

2.9 利用 z 变换求解下列差分方程

(1) $y(n)=\frac{1}{2}y(n-1)+x(n)$, $n\geqslant 0$

其中，$x(n)=(0.5)^n u(n)$，$y(-1)=\frac{1}{4}$

(2) $y(n)=0.5y(n-1)+0.25y(n-2)+x(n)$，　　$n\geqslant 0$；

$y(-1)=1$，$y(-2)=2$，$x(n)=(0.8)^n u(n)$

2.10　对于一个线性时不变系统，其差分方程为

$$y(n)=x(n)+0.5x(n-1)-0.5y(n-1)+0.25y(n-2)$$

试求：

(1) 系统的脉冲响应表达式；

(2) 系统传递函数表达式；

(3) 输入 $x(n)=2(0.9)^n u(n)$ 时的输出 $y(n)$；

(4) 系统零极点图。

2.11　设有限序列 $x(n)=(0.5)^n u(n)$，试求：

(1) 序列 $x(n)$ 的 z 变换表达式 $X(z)$；

(2) 求 $X(z)$ 在单位圆上 $N=20$ 等分上频率采样值 $X(k)$；

(3) 求 $IDFT[X(k)]=\tilde{x}(n)$；

(4) 改变 N，观察 N 对 $\tilde{x}(n)$ 的影响。

2.12　已知线性时不变系统

$$y(n)=y(n-1)+y(n-2)+x(n-1)$$

试绘制系统的幅频响应和相频响应曲线

2.13　设周期信号为

$$x(t)=0.75+3.4\cos 2\pi ft+2.7\cos 4\pi ft+1.5\sin 3.5\pi ft+2.5\sin 7\pi ft$$

其中，$f=\frac{25}{16}$Hz，取采样频率 Fs=100Hz，试绘出信号在下列不同长度 N 的序列 $x(n)$ 的幅频谱图。

(1) N=24；　　(2) N=128；　　(3) N=512；　　(4) N=1024。

2.14　已知信号为 $x(t)=2\sin(4\pi t)+5\cos(8\pi t)+0.2w(t)$，其中，$w(t)$ 为白噪声。用 FFT 绘制信号的幅频谱图。

2.15　已知序列 $x_1(n)=0.5n$，$x_2(n)=0.5n$，序列长度 N=5000，用 FFT 求卷积 $x_1(n)*x_2(n)$。

2.16　已知一个频率为 5Hz 正弦信号，用频率为 200Hz 的余弦信号作为载波信号，用 MATLAB 编程绘制调幅信号波形及频谱图。

第三章　模拟滤波器设计

滤波是信号处理的一种最基本而重要的技术，利用滤波从复杂的信号中提取所需要的信号，抑制不需要的部分。所谓滤波器是具有一定传输特性的信号处理装置。

根据滤波器所处理的信号不同，滤波器可分为模拟滤波器和数字滤波器两类。本章介绍模拟原型滤波器的设计基本原理、方法和几种常用的模拟滤波器。

3.1　模拟滤波器设计原理

3.1.1　信号无失真传输条件

所谓信号无失真传输是指输入信号通过系统后，输出信号的幅值和输入信号呈比例，允许有一定的延时，但没有波形上的畸变。据此可得，系统的频率响应 $H(\omega)$ 满足下面特性

$$|H(j\omega)| = k \tag{3.1-1}$$

$$\varphi(\omega) = \angle H(j\omega) = -\omega t_d \tag{3.1-2}$$

式中，k，t_d 均为常数。

即信号无失真输传输的条件是：系统的幅频响应 $|H(j\omega)|$ 应为常数，相频响应 $\angle H(j\omega)$ 应与频率 ω 成正比。或者说，滤波器应具有无限宽的定值幅频与线性相频。由式(3.1-2)定义群延迟(Group Delay)为信号通过系统的延迟时间，即

$$t_d = -\frac{d\varphi(\omega)}{d\omega} \tag{3.1-3}$$

群延迟为相频特性曲线的斜率。对于信号无失真传输，t_d 为常数，故群延迟为常数；否则它是频率 ω 的函数。

3.1.2　理想滤波器特性

滤波器是一个选频装置。理想滤波器应能无失真传输有用信号，而又能完全抑制无用信号。有用信号和无用信号往往占有不同频带。信号能通过滤波器的频带称为通带(Passband)，信号被抑制的频带称为阻带(Stopband)。理想滤波器频率特性可写为

$$H(j\omega) = \begin{cases} Ke^{j\omega t_d}, & \text{在通带内} \\ 0, & \text{在阻带内} \end{cases} \tag{3.1-4}$$

但理想滤波器是物理不可实现系统。实际滤波器的频率特性只能“逼近”理想滤波器，图 3.1 所示为一通低滤波器的幅频特性。

可见，在通带内不是完全平直的，而是呈波纹变化；在阻带内，幅频特性也不为零，而是衰减至某个值；在通带和阻带之间存在一个过渡带，而不是突然下降。通常，实际设计要求的滤波器的技术指标包括通带波纹 R_p(Passband ripple)(dB)、阻带衰减 R_s(Stopoband attenuation)(dB)、通带边界频率 ω_p、阻带边界频率 ω_s、过渡带宽 $(\omega_s-\omega_p)$。

3.1.3　滤波器传递函数设计

模拟滤波器的技术指标可由平方幅值响应函数 $A(\omega^2) = |H(j\omega)|^2$ 形式给出，而

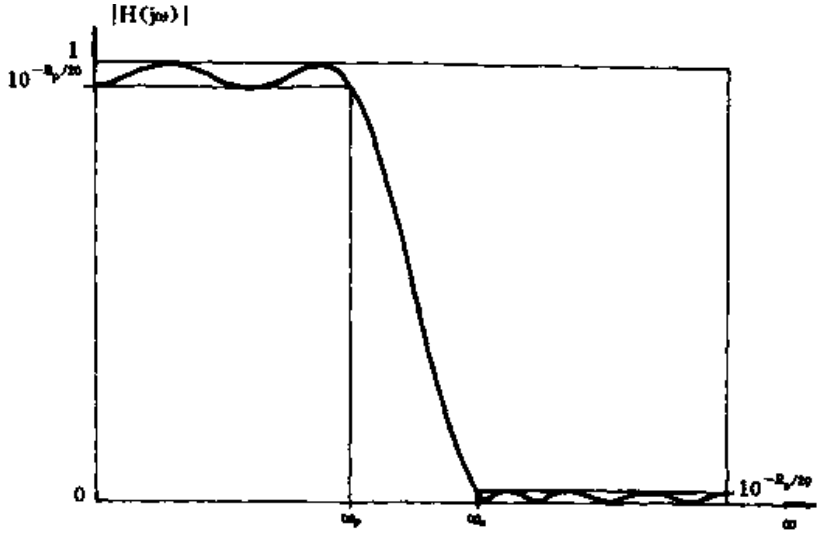


图 3.1　实际滤波器特性

$|H(jw)|^2$和传递函数存在下面关系：

$$A(\omega^2) = |H(j\omega)|^2 = H(s)H(-s)|_{s=j\omega}$$

即

$$A(\omega^2)|_{\omega^2=-s^2} = H(s)H(-s) \tag{3.1-5}$$

由式(3.1-5)可知，当给定模拟滤波的技术指标后，由 $A(\omega^2)= |H(j\omega)|^2$ 求出 $A(-s^2)$，再适当地分配零极点可求出 H(s)。为了使滤波器稳定，H(s)的极点必须落在 s 平面左半平面，而零点选择可任取 $A(-s^2)$的一半零点。但如果要求 H(s)具有最小相位，零点也应选在 s 左半平面。

3.2　模拟原型滤波器

本节介绍常用的低通模拟原型滤波器的主要特点及其 MATLAB 实现，包括巴特沃斯滤波器、切比雪夫 I 型滤波器、切比雪夫 I 型滤波器、椭圆滤波器、巴塞尔滤波器。各类模拟滤波器及数字滤波器可通过这些低通模拟原型滤波器变换获得。

3.2.1　巴特沃斯滤波器

巴特沃斯(Butterworth)模拟低通滤波器原型的平方幅频响应函数为

$$|H(j\omega)|^2 = A(\omega^2)\frac{1}{1+(\omega/\omega_c)^{2N}} \tag{3.2-1}$$

式中，ω_c 为低通滤波器的截止频率(Cutoff frequency)，N 为滤波器的阶数。

Butterworth 低通滤波器特点是：通带内具有最大平坦的幅频特性，且随频率增大平滑单调下降；阶数 N 愈高，特性愈接近矩形，过渡带宽愈窄。传递函数无零点，极点等距离分布在以 $|s|=\omega_c$ 为半径的圆周上。MATLAB 信号处理工具箱提供 Butterworth 模拟低通滤波器原型设计函数 BUTTAP，函数调用格式为

```
[Z, P, K]= buttap(n)
```

其中，n 为 Butterworth 滤波器阶数；Z，P，K 分别为滤波器零点、极点和增益。

滤波器传递函数具有下面形式：

$$H(s) = \frac{Z(s)}{P(s)} = \frac{K}{(S-P(1))(S-P(2))\cdots(S-P(n))}$$

【例 3.1】 绘制 Butterworth 低通模拟原型滤波器的平方幅频响应曲线，阶数分别为 2,5,10,20。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 3-1
%Buterworth analog lowpass filter prototype
n=0:0.01:2;
for i=1:4
   switch i
   case 1
       N=2;
   case 2
       N=5;
   case 3
       N=10;
   case 4
       N=20;
end
[z,p,k]=buttap(N);
[b,a]=zp2tf(z,p,k);
[H,w]=freqs(b,a,n);
magH2=(abs(H)).^2;
hold on
plot(w,magH2);
axis([0 2 0 1]);
end
xlabel('w/wc');
ylabel('|H(jw)|^2');
title('Buterworth analog filter prototype');
grid
```

程序运行结果如图 3.2 所示。

3.2.2 切比雪夫滤波器

切比雪夫(Chebyshev)滤波器有 Chebyshev Ⅰ型和 Chebyshev Ⅱ型两类。

一、Chebyshev Ⅰ型

Chebyshev Ⅰ型模拟低通滤波器的平方幅值响应函数为

$$|H(j\omega)|^2 = A(\omega^2) = \frac{1}{1+\varepsilon^2 C_N^2\left(\frac{\omega}{\omega_c}\right)} \tag{3.2-2}$$

其中，ε 为小于 1 的正数，表示通带内幅值波纹情况；ω_c 为截止频率，N 为 Chebyshev 多项

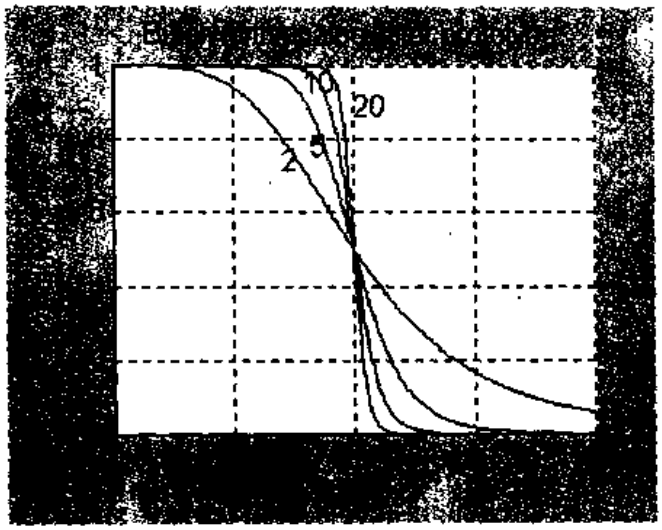


图 3.2 Butterworth 低通滤波器原型平方幅频图

式阶数，$C_N(\frac{\omega}{\omega_c})$为 Chebyshev 多项式，定义为

$$C_N(x)=\begin{cases}\cos(N\cos^{-1}(x)) & |x|\leqslant 1\\ \cosh(N\cosh^{-1}(x)) & x\geqslant 1\end{cases} \tag{3.2-3}$$

Chebyshev I 型滤波器特点是：通带内具有等波纹起伏特性，而在阻带内则单调下降且具有更大衰减特性；阶数 N 愈高，特性愈接近矩形。传递函数没有零点，极点分布在一个椭圆上。

MATLAB 信号处理工具箱函数 CHEB1AP 设计 N 阶的 Chebyshev I 型模拟低通滤波器原型。调用格式为

$$[Z, P, K]=\text{cheb1ap}(N, R_p)$$

其中，N 为滤波器阶数；R_p 为通带波纹，dB；Z，P，K 为滤波器零点、极点和增益。

滤波器传递函数具有下面形式

$$H(s)=\frac{Z(s)}{P(s)}=\frac{K}{(s-p(1))(s-p(2))\cdots(s-p(n))}$$

【例 3.2】 绘制 Chebyshev I 型模拟低通滤波器的平方幅频响应曲线，阶数分别为 2，4，6，8。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 3-2
clf
%Chebyshev type I analog lowpass filter prototype
n=0:0.01:2;
for i=1:4
   switch i
   case 1
      N=2;
   case 2
```

```
    N=4;
  case 3
    N=6;
  case 4
    N=8;
  end
Rp=1;
[z,p,k]=cheb1ap(N,Rp);
[b,a]=zp2tf(z,p,k);
[H,w]=freqs(b,a,n);
magH2=(abs(H)).^2;
%Output
posplot=['22'num2str(i)];
subplot(posplot)
plot(w,magH2);
axis([0 2 0 1]);
xlabel('w/wc');
ylabel('Chebyshev I |H(jw)|^2');
title(['N='num2str(N)]);
grid
end
```

程序运行结果如图3.3所示。

二、Chebyshev I 型

Chebyshev I 型模拟低通滤波器平方幅值响应函数为

$$|H(j\omega)|^2 = A(\omega^2) = \frac{1}{\left[1+\varepsilon^2 C_N^2\left(\frac{\omega}{\omega_c}\right)\right]^{-1}} \tag{3.2-4}$$

式中各项意义同式(3.2-2)。

Chebyshev I 型滤波器特点是：阻带内具有等波纹起伏特性，而在通带内是单调、平滑的。阶数 N 愈高，特性曲线愈接近矩形。传递函数既有极点，又有零点。

MATLAB 信号处理工具箱提供函数 CHEB2AP 设计 N 阶的 Chebyshev I 型模拟低通滤波器原型，调用格式为

$$[Z, P, K]=\text{cheb2ap}(N, R_s)$$

其中，N 为滤波器阶数；R_s 为阻带波纹，dB；Z，P，K 为滤波器零极点和增益。

滤波器的传递函数具有下面形式：

$$H(s) = \frac{z(s)}{p(s)} = k\,\frac{(s-z(1))(s-z(2))\cdots(s-z(n))}{(s-p(1))(s-p(2))\cdots(s-p(n))}$$

【例3.3】 绘制 Chebyshev I 型模拟低通滤波器的平方幅频响应曲线，阶数分别为2，4，6，8。

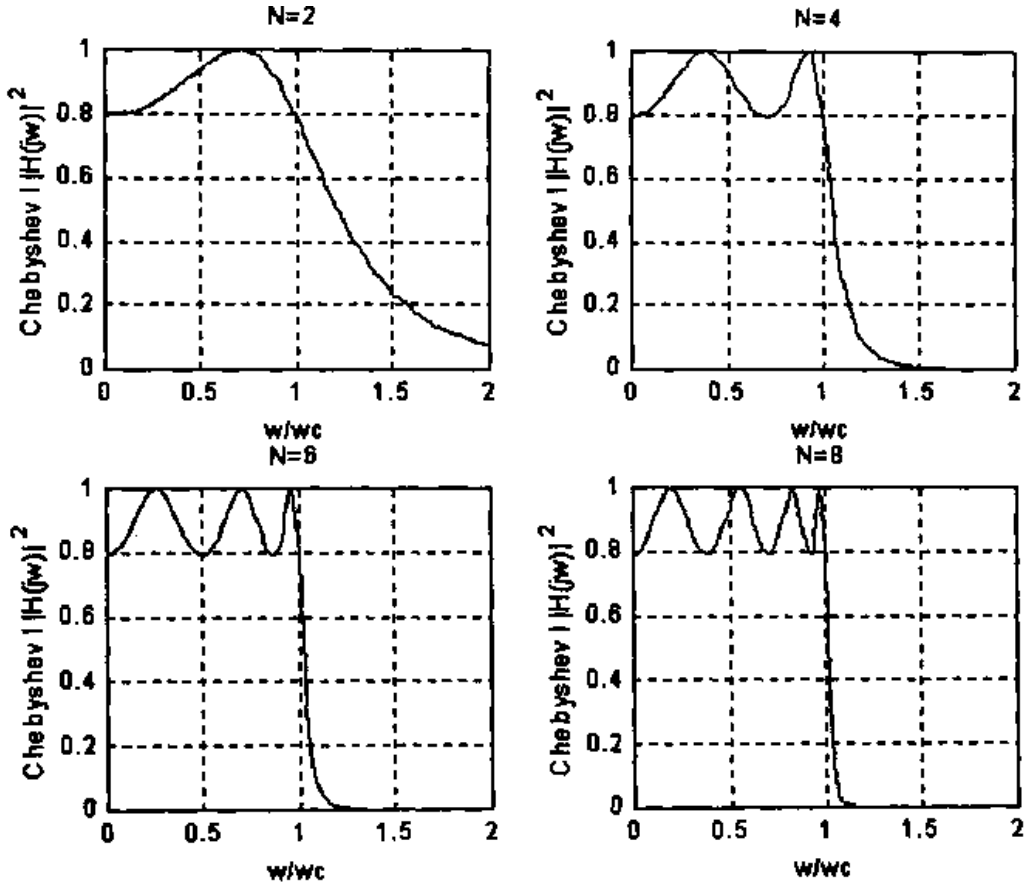


图3.3 Chebyshev I 型模拟原型滤波器平方幅频图

用 MATLAB 编写程序并绘制的响应曲线如图3.4所示。

3.2.3 椭圆滤波器

椭圆的(Elliptic)模拟低通滤波器原型的平方幅值响应函数为

$$|H(j\omega)|^2 = A(\omega^2) = \frac{1}{1+\mu^2 E_N^2(\omega/\omega_c)} \tag{3.2-5}$$

式中，μ 为小于1的正数，表示波纹情况；ω_c 为截止频率；$E_N\left(\frac{\omega}{\omega_c}\right)$ 为椭圆函数，定义为

当 N 为偶数(N=2m)时，

$$E_N(\Omega) = \prod_{k=1}^{m} \frac{\Omega_k^2 - \Omega^2}{1 - \Omega^2\Omega_k^2} \tag{3.2-6}$$

当 N 为奇数(N=2m+1)时，

$$E_N(\Omega) = \Omega\prod_{k=1}^{m} \frac{\Omega_k^2 - \Omega^2}{1 - \Omega^2\Omega_k^2}$$

其中

$$\Omega = \frac{\omega}{\omega_c}$$

椭圆模拟滤波器特点是：在通带和阻带内均具有等波纹起伏特性。和以上滤波器原型相比，相同的性能指标所需的阶数最小。但相频响应具有明显的非线性。

MATLAB 信号处理工具箱函数 ELLIPAP 用于设计 N 阶椭圆低通滤波器原型。调用格式为

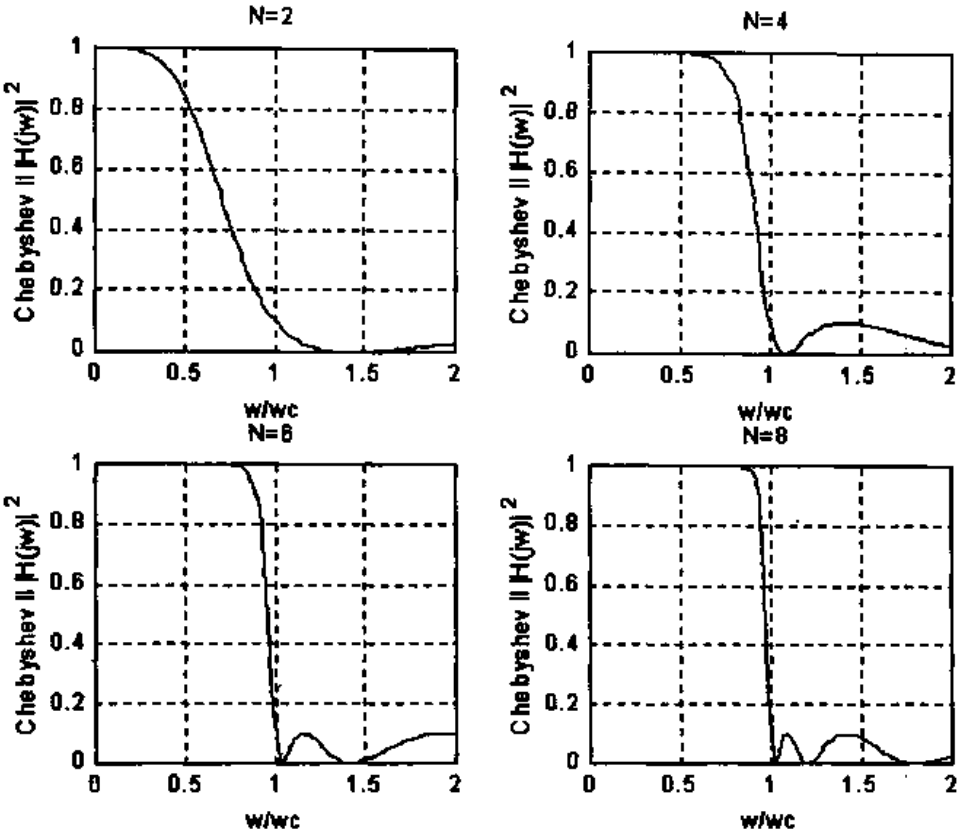


图3.4 Chebyshev I 型模拟滤波器原型平方幅频图

$$[z, p, k]=\text{ellipap}(N, R_p, R_s)$$

其中，N 为滤波器阶数；R_p 为通带波纹，dB；R_s 为阻带衰减，dB；z，p，k 分别为滤波器传递函数零点、极点和增益。

滤波器传递函数具有下面形式：

$$H(s)=\frac{z(s)}{p(s)}=k\frac{(s-z(1))(s-z(2))\cdots(s-z(n))}{(s-p(1))(s-p(2))\cdots(s-p(n))}$$

当阶数 N 为偶数，z 和 p 的长度为 N；当阶数 N 为奇数，z 向量长度为 N−1。

【例3.4】 绘制椭圆模拟低通滤波器原型平方幅频响应曲线，阶数分别为2，4，6，8。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 3-3
clf
%Elliptic type I analog lowpass filter prototype
n=0:0.01:2;
for i=1:4
   switch i
    case 1
        N=2;
    case 2
```

```
    N=3;
  case 3
    N=4;
  case 4
    N=5;
  end
Rp=1;
Rs=10;
[z,p,k]=ellipap(N,Rp,Rs);
[b,a]=zp2tf(z,p,k);
[H,w]=freqs(b,a,n);
magH2=(abs(H)).^2;
%Output
posplot=['22'num2str(i)];
subplot(posplot)
plot(w,magH2);
axis([0 2 0 1]);
xlabel('w/wc');
ylabel('Ellipse |H(jw)|^2');
title(['N='num2str(N)]);
grid
end
```

程序运行结果如图3.5所示。显然，在低阶情况下，过渡带就已经很窄。

3.2.4 贝塞尔滤波器

贝塞尔(Bessel)模拟低通滤波器原型的特点是在零频时具有最平坦的群延迟，并在整个通带内群延迟几乎不变。在零频时的群延迟为$\left\{\frac{(2N)!}{2^N N!}\right\}^{1/N}$。由于这一特点，贝塞尔模拟滤波器通带内保持信号形状不变。但数字贝塞尔滤波器没有平坦特性，因此 MATLAB 信号处理工具箱只有模拟巴塞尔滤波器设计函数。

函数 BESSELAP 用于设计贝塞尔模拟低通滤波器原型，调用格式为

$$[z,\ p,\ k]=\text{besselap}(N)$$

其中，N 为滤波器阶数，小于25；z，p，k 为滤波器零点、极点和增益。

滤波器传递函数具有下面形式

$$H(s)=\frac{k}{(s-p(1))(s-p(2))\cdots(s-p(n))}$$

滤波器无零点。

【例3.5】 绘制5阶和10阶的贝塞尔低通滤波器原型的平方幅频和相频图。

用 MATLAB 编写程序如下：

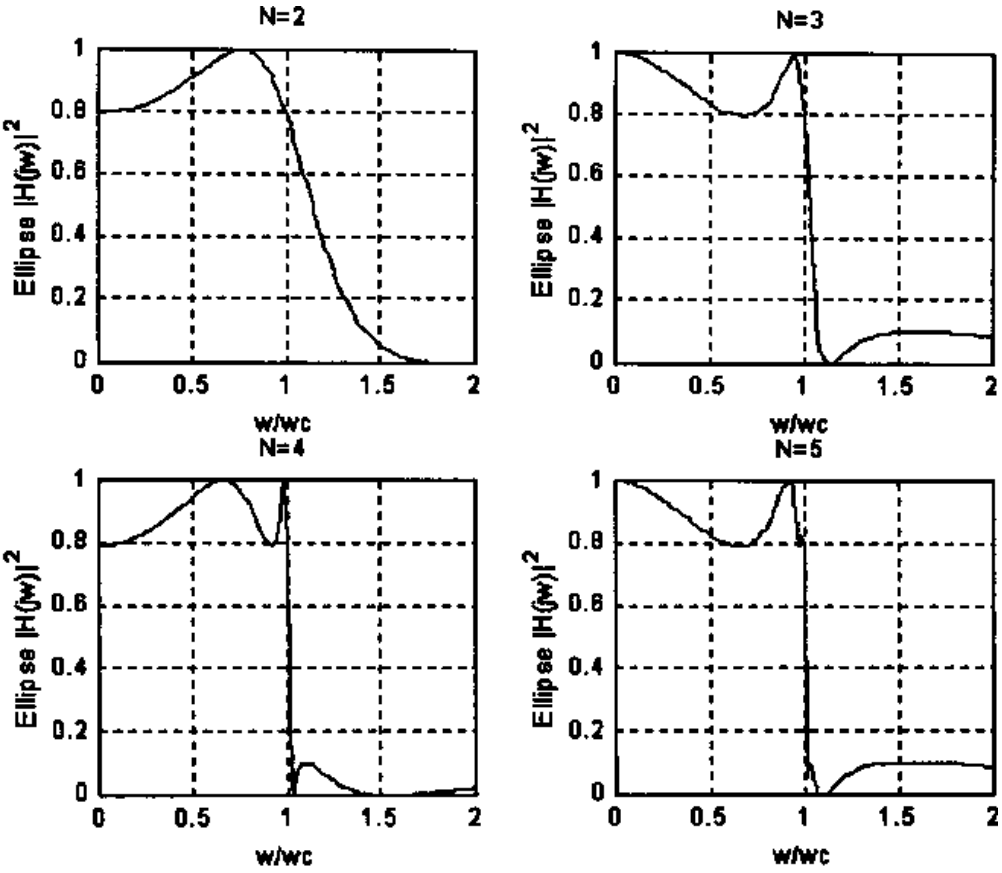


图3.5 椭圆滤波器原型平方幅频图

```
%MATLAB PROGRAM 3-4
clf
%Bessel analog lowpass filter prototype
n=0:0.01:2;
for i=1:2
    switch i
    case 1
        pos=1;
        N=5;
    case 2
       pos=3;
       N=10;
    end
 [z,p,k]=besselap(N);
 [b,a]=zp2tf(z,p,k);
 [H,w]=freqs(b,a,n);
 magH2=(abs(H)).^2;
 phaH=unwrap(angle(H));
 phaH=phaH*180/pi;
```

```
%Output
posplot=['22'num2str(pos)];
subplot(posplot)
plot(w,magH2);
axis([0 2 0 1]);
xlabel('w/wc');
ylabel('Bessel |H(jw)|^ 2');
title(['N='num2str(N)]);
grid
posplot=['22'num2str(pos+1)];
subplot(posplot)
plot(w,phaH);
xlabel('w/wc');
ylabel('Bessel Phase <H(jw)');
title(['N='num2str(N)]);
grid
end
```

程序运行结果如图3.6所示。显然，在 $\omega/\omega_c<1$ 内，相频曲线几乎为线性。

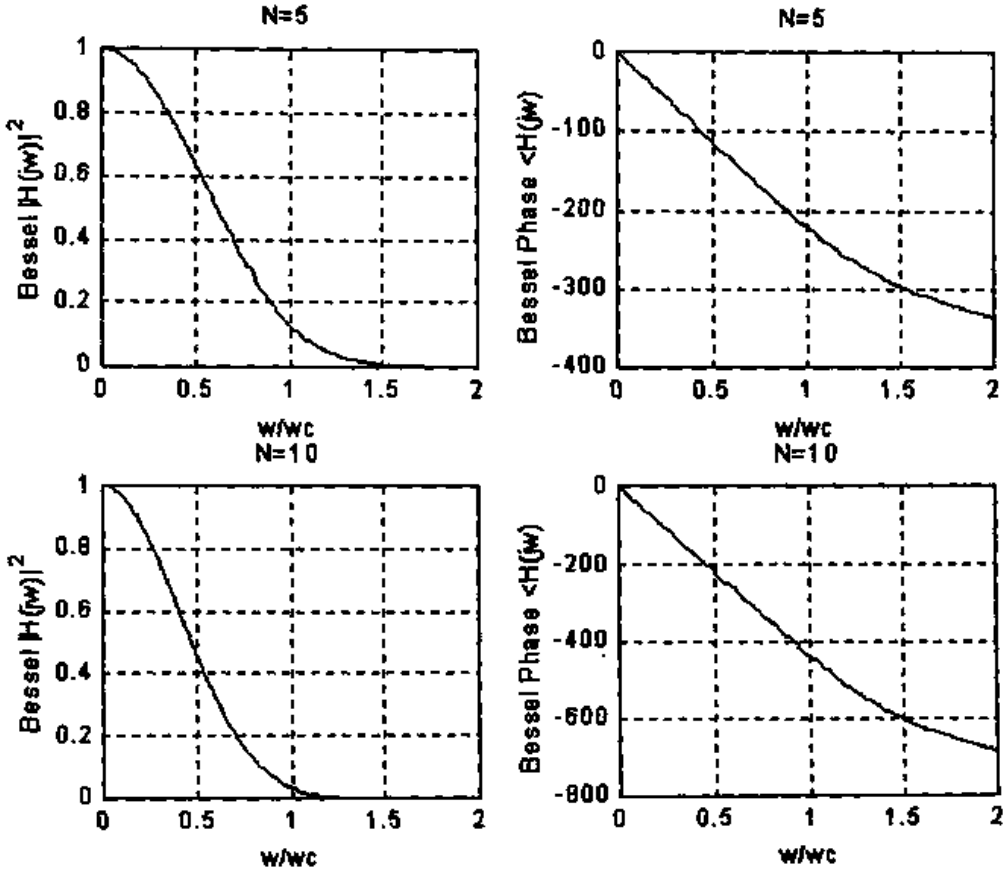


图3.6 贝塞尔模拟低通滤波器原型幅相频图

3.3 频率变换

所谓频率变换是指各类滤波器(低通、高通、带通、带阻)和低通滤波器原型的传递函数中频率自变量之间的变换关系。通过频率变换,我们可以从模拟低通滤波器原型获得模拟的低通滤波器、高通滤波器、带通滤波器和带阻滤波器,再借助 S 域至 z 域转换又可以设计各类数字 IIR 滤波器。这是滤波器设计的重要方法之一。

3.3.1 频率变换工具函数

MATLAB 信号处理工具箱有 LP2LP、LP2HP、LP2BP 和 LP2BS 四个频率变换函数。

(1) 函数 LP2LP 用于实现用低通模拟原型滤波器至低通滤波器的频率转换。调用格式为

$$[bt, at] = lp2lp(b, a, \omega_0)$$

$$[At, Bt, Ct, Dt] = lp2lp(A, B, C, D, \omega_0)$$

式中,b, a 或 A,B,C,D 为模拟原型滤波器,其模型有下面形式:

$$\frac{b(s)}{a(s)} = \frac{b(1)s^{nn} + \cdots b(nn)s + b(nn+1)}{a(1)s^{nd} + \cdots a(nd)s + a(nd+1)}$$

或

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

ω_0为低通滤波器的期望截止频率(rad/s)。

函数返回低通滤波器的模型,用传递函数形式(bt, at)或状态空间形式(At, Bt, Ct, Dt)。

该函数执行下面变换

$$H(s) = H(p)|_{p=\omega_0 s} \tag{3.3-1}$$

式中,H(p)为低通原型滤波器传递函数,H(s)为低通滤波器传递函数。

(2) 函数 LP2HP 用于实现由低通模拟原型滤波器至高通滤波器的频率转换。

调用格式为

$$[bt, at] = lp2hp(b, a, \omega_0)$$

$$[At, Bt, Ct, Dt] = lpzlp(A, B, C, D, \omega_0)$$

其中,ω_0为高通滤波器的期望截止频率(rad/s)。

该函数执行下面变换

$$H(s) = H(p)|_{p=\frac{\omega_0}{s}} \tag{3.3-2}$$

(3) 函数 LP2BP 用于实现由低通模拟原型滤波器至带通滤波器的频率变换。

调用格式为

$$[bt, at] = lpzbp(b, a, \omega_0, B_w)$$

$$[At, Bt, Ct, Dt] = lp2bp(A, B, C, D, \omega_0, B_w)$$

其中,ω_0为带通滤波器的中心频率(rad/s), B_w 为带通滤波器带宽(rad/s),而

$$\omega_0 = \sqrt{\omega_1 \omega_2} \tag{3.3-3}$$

$$B_w = \omega_2 - \omega_1 \tag{3.3-4}$$

其中，ω_1为带通滤波器下边界频率，ω_2为带通滤波器上边界频率。

该函数执行下面变换运算

$$H(s)=H(p)|p=\frac{\omega_0}{B_w}\frac{\left(\frac{s}{\omega_0}\right)^2+1}{\frac{s}{\omega_0}} \tag{3.3-5}$$

(4) 函数 LP2BS 用于实现由低通模拟原型滤波器至带阻滤波器的频率变换。

调用格式为

[bt, at] = lp2bs(b, a, ω_0, B_w)

[At, Bt, Ct, Dt] = lp2bs(A, B, C, D, ω_0, B_w)

其中，ω_0为带阻滤波器中心频率(rad/s)，B_w 为带阻滤波器带宽(rad/s)，而

$$\omega_0=\sqrt{\omega_1\omega_2}$$

$$B_w=\omega_2-\omega_1$$

其中，ω_1为带阻滤波器下边界频率，ω_2为带阻滤波器上边界频率。

该函数执行下面变换运算

$$H(s)=H(p)|p=\frac{\omega_0}{B_w}\frac{\frac{s}{\omega_0}}{\left(\frac{s}{\omega_0}\right)^2+1} \tag{3.3-6}$$

其余同函数 LP2LP。

注意，输出的带阻和带通模拟滤波器是滤波器原型阶数的2倍。

3.3.2 模拟滤波器设计方法

利用频率变换设计模拟滤波器的步骤为：

(1) 给定模拟滤波器的性能指标，如截止频率 ω_0(对于低通和高通)或上、下边界频率 ω_1，ω_2；波纹特性；带阻衰减等。

(2) 确定滤波器阶数。

(3) 设计模拟低通原型滤波器。MATLAB 信号处理工具箱的滤波器原型设计函数有 buttap, cheb1ap, cheb2ap, ellipap, besselap。

(4) 按频率变换设计模拟滤波器(低通、高通、带通、带阻)。MATLAB 信号处理工具箱的频率变换函数有：lp2lp, lp2hp, lp2bp, lp2bs。

【例3.6】 设计一个5阶的 chebshev Ⅰ型带通滤波器，通带滤纹3dB，下边界频率200π rad/s，上边界频率1000π rad/s，并绘制幅频响应图。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 3-5
%Desired performents
N=5;
Rp=3;
w1=200 * pi;
```

```
w2=1000 * pi;
%Chebyshev analog lowpass filter prototype
[z,p,k]=cheb1ap(N,Rp);
[b,a]=zp2tf(z,p,k);
%Find the center frequency and Bandwidth
Wo=sqrt(w1 * w2);
Bw=w2-w1;
%Design analog bandpass filter
disp('Designed analog bandpass filter:')
[bt,at]=lp2bp(b,a,Wo,Bw);
[h,w]=freqs(bt,at);
semilogy(w/pi,abs(h));
xlabel('Frequency (pi)')
grid on
```

程序运行结果给出带通滤波器传递函数，它的幅频响应如图3.7。

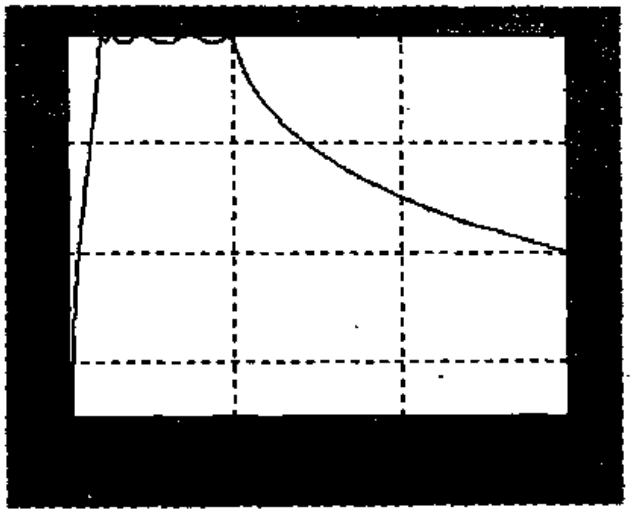


图3.7 模拟带通波波器幅频图

3.4 模拟滤波器最小阶数的选择

前面所述的模拟滤波器设计中，一个重要的步骤是确定滤波器的阶数，因为它是决定滤波器品质的主要参数之一。在满足性能指标的前题下，阶数应该尽可能小，以满足易于实现的要求。而在阶数和滤波器性能之间存在一定的函数关系，通过这一函数关系可以求出满足滤波器性能指标的最低阶数。下面首先介绍常用低通滤波器最小阶数的确定原理及 MATLAB 实现；接着介绍 MATLAB 信号处理工具箱中用来计算最小阶数和截止频率的工具函数。

3.4.1 巴特沃斯低通模拟滤波器

若给出模拟低通滤波器的设计性能指标要求：通带边界频率 ω_p，阻带边界频率 ω_s，通

带波纹 R_p(dB),阻带衰减 R_s(dB),要确定 Butterworth,低通波波器最小阶数 N 及截止频率 ω_c(−3dB)。ω_p,ω_s,R_p,R_s 的意义如图3.1所示。

由图3.1可知,当 $\omega=\omega_p$ 时,

$$|H(j\omega)| = 10^{-\frac{R_p}{20}}$$

即

$$R_p = -10\lg|H(j\omega)|^2$$

以截止频率 ω_c(幅值下降3dB)为1,化 ω 为相对 ω_c 的相对频率 $\frac{\omega}{\omega_c}$,则由上式可写为

$$R_p = -10\log_{10}\left[\frac{1}{1+\left(\frac{\omega_p}{\omega_c}\right)^{2N}}\right] \tag{3.4-1}$$

同理,当 $\omega=\omega_s$ 时,$|H(j\omega)|=10^{-\frac{R_s}{20}}$,

$$R_s = -10\log_{10}\left[1+\frac{1}{\left(\frac{\omega_s}{\omega_c}\right)^{2N}}\right] \tag{3.4-2}$$

由此可解:

$$N = \frac{\log_{10}[(10^{R_p/10}-1)(10^{R_s/10}-1)]}{2\log_{10}\left(\frac{\omega_p}{\omega_s}\right)} \tag{3.4-3}$$

N 应向上取整

$$\omega_c = \frac{\omega_p}{\sqrt[2N]{(10^{R_p/10}-1)}} \tag{3.4-4}$$

或

$$\omega_c = \frac{\omega_s}{\sqrt[2N]{(10^{R_s/10}-1)}} \tag{3.4-5}$$

由式(3.4-3)和式(3.4-4)或式(3.4-5),用 MATLAB 编程计算滤波器最小阶数 N 和截止频率 ω_c.

【例3.7】 设计一个模拟 Butterworth 滤波器。设计指标:通带边界频率 $w_p=200\pi$,阻带边界频率 $w_s=300\pi$,通带波纹1dB,阻带宽衰减16dB,试确定最小阶数 N 和截止频率 w_c.

用 MATLAB 编程如下:

```
%MATLAB PROGRAM 3-6
%Butterworth lowpass filter
%Desired performents
wp=200 * pi;
ws=300 * pi;
Rp=1;
Rs=16;
%Compute Order N
N1=log10((10^ (Rp/10)-1)/(10^ (Rs/10)-1))/(2 * log10(wp/ws));
```

```
N=ceil(N1)
%Compute cutoff frequency to meet wp
Wcp=wp/((10^ (Rp/10)-1)^ (1/(2*N)))
%Compute cutoff frequency to meet ws
Wcs=ws/((10^ (Rs/10)-1)^ (1/(2*N)))

≫smp306
N =
     7
Wcp =
  691.9839
Wcs =
  725.7292
```

由程序运行结果可知，滤波器最小阶数为7，在 ω_p 处满足设计指标的截止频率 ω_{cp} 为691.9839 rad/s，在 ω_s 处满足设计指标的截止频率 w_{cs} 为725.7292rad/s，截止频率 ω_c 在 ω_s 和 ω_p 之间(R_p<3dB)。

3.4.2 切比雪夫低通模拟滤波器

和 Butterworth 低通模拟滤波器设计一样，若给定设计性能指标要求：ω_c，ω_s，R_p，R_p，确定 Chebyshev 低通模拟滤波器最小阶数 N 和截止频率 ω_c(−3dB 频率)。

一、Chebyshev Ⅰ型

由式(3.2-2)可得

$$\begin{aligned} \varepsilon &= \sqrt{10^{R_p/10}-1} \\ A &= 10^{R_s/20} \end{aligned} \tag{3.4-6}$$

故阶数 N 可由下式求得

$$N=\frac{\log10(g+\sqrt{g^2+1})}{\log10\left(\frac{\omega_s}{\omega_p}+\sqrt{\left(\frac{\omega_s}{\omega_p}\right)^2-1}\right)} \tag{3.4-7}$$

式中，

$$\begin{aligned} g&=\sqrt{(A^2-1)/\varepsilon^2} \\ &\text{截止频率 } \omega_c=\omega_p \end{aligned} \tag{3.4-8}$$

由式(3.4-7)和式(3.4-8)，用 MATLAT 编程计算滤波器最小阶数 N 和截止频率 ω_c.

【例3.8】 设计一个模拟低通 chebyshev 1 型滤波器，设计指标同例3.7，试确定最小阶数 N 和截止频率 ω_c.

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 3-7
%Chebyshev I lowpass filter
%Desired performents
```

```
wp=200 * pi;
ws=300 * pi;
Rp=1;
Rs=16;
%Compute Order N
ebs=sqrt(10^ (Rp/10)-1);
A=10^ (Rs/20);
Wc=wp
Wr=ws/wp;
g=sqrt(A * A-1)/ebs;
N1=log10(g+sqrt(g * g-1))/log10(Wr+sqrt(Wr * Wr-1));
N=ceil(N1)

≫smp307
Wc =
  628.3185
N =
    4
```

由程序运行结果，滤波器最小阶数为4，−3dB 截止频率 ωc 为628.3185rad/s.

二、Chebyshev Ⅰ型

由式(3.2-4)和图3.3可知，Chebyshev Ⅰ型通带内是平滑的，而阻带具有等波纹起伏特性。因此，在阶数 N 的计算公式上是相同的，而 −3dB 截止频率 ω_c 则不同。下面例子用于计算 Chebyshev Ⅰ型低通滤波器阶数 N 和截止频率 ω_c.

【例3.9】 设计一个模拟低通 Chebyshev Ⅰ型滤波器。设计指标同【例3.7】，试确定最小阶数 N 和截止频率 ω_c.

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 3-8
%Chebyshev I I lowpass filter
%Desired performents
wp=200 * pi;
ws=300 * pi;
Rp=1;
Rs=16;
%Compute Order N
ebs=sqrt(10^ (Rp/10)-1);
A=10^ (Rs/20);
Wr=ws/wp;
g=sqrt(A * A-1)/ebs;
```

```
N1=log10(g+sqrt(g*g-1))/log10(Wr+sqrt(Wr*Wr-1));
N=ceil(N1)
%Compute cutoff frequency
new_wp=1/cosh(1/N*acosh(sqrt((10^(.1*abs(Rs))-1)/(10^(.1*abs(Rp))-1))));
Wn=wp/new_wp

≫smp308
N =
     4
Wn =
  839.8376
```

程序运行结果：滤波器最小阶数为4，截止频率为 $\omega_c=839.8\text{rad/s}$.

3.4.3 椭圆低通模拟滤波器

由式(3.2-5)，滤波器的阶数可由下式确定

$$\mu=\sqrt{10^{R_p/10}-1}$$

$$A=10^{R_s/20}$$

$$N=\frac{K(k)K(\sqrt{1-k_1^2})}{K(k_1)K(\sqrt{1-k^2})} \tag{3.4-9}$$

$$\omega_{ln}=\omega_p \tag{3.4-10}$$

式中

$$k=\frac{\omega_p}{\omega_s}\qquad k_1=\frac{\mu}{\sqrt{A^2-1}}$$

$$K(x)=\int_0^{\frac{\pi}{2}}\frac{d\theta}{\sqrt{1-x^2\sin^2\theta}} \tag{3.4-11}$$

由式(3.4-9)和式(3.2-10)，计算滤波器的最小阶数 N 和截止频率 w_c.

【例3.9】 设计一个模拟低通椭圆滤波器。设计指标同例3.7，试确定最小阶数 N 和截止频率 ω_c.

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 3-9
%Elliptic analog lowpass filter
%Desired performents
wp=200*pi;
ws=300*pi;
Rp=1;
Rs=16;
%Compute Order N
```

```
ebs=sqrt(10^ (Rp/10)-1);
A=10^ (Rs/20);
Wc=wp
k=wp/ws;
k1=ebs/sqrt(A * A-1);
KK1=ellipke(k);
KK2=ellipke(sqrt(1-k1^ 2));
KK3=ellipke(k1);
KK4=ellipke(sqrt(1-k^ 2));
N1=KK1 * KK2/(KK3 * KK4);
N=ceil(N1)

≫smp309
Wc =
  628.3185
N =
     3
```

程序运行结果，滤波器的最小阶数 N 为3，截止频率为 ω_c=628.3185rad/s

3.4.4 最小阶数选择函数

MATLAB 信号处理工具箱提供各种滤波器最小阶数选择工具函数，这些函数和滤波器设计函数连接起来可方便成功地设计各类滤波器。用来选择模拟滤波器阶数的函数有：

(1) Butterworth 模拟滤波器阶数选择函数 Buttord，调用格式为

$$[n,\ \omega_n]=\text{buttord}(\omega_p,\ \omega_s,\ R_p,\ R_s,\ 's')$$

(2) Chebyshev Ⅰ模拟滤波器阶数选择函数 CHEB2ORD，调用格式为

$$[n,\ \omega_n]=\text{cheb1ord}(\omega_p,\ \omega_s,\ R_p,\ R_s,\ 's')$$

(3) Chebyshev Ⅱ型模拟滤波器阶数选择函数 CHEBZORD，调用格式为

$$[n,\ \omega_n]=\text{cheb2ord}(\omega_p,\ \omega_s,\ R_p,\ R_s,\ 's')$$

(4) 椭圆模拟滤波器阶数选择函数 ELLIPORD，调用格式为

$$[n,\ \omega_n]=\text{ellipord}(\omega_p,\ \omega_s,\ R_p,\ R_s,\ 's')$$

其中，ω_p 为通带边界频率，rad/s；ω_s 为阻带边界频率，rad/s；R_p 为通带波动，dB，此值是滤波器在通带0～ω_p 之间允许最大允许幅值损失；R_s 为阻带衰减，dB，此值是滤波幅值从通带至阻带下降的分贝数；'s'表示模拟滤波器(缺省时该函数适用于数字滤波器)；函数返回值 n 模拟滤波器最小阶数；ω_n 为模拟滤波器的截止频率(－3dB 频率)，rad/s，这四个函数适用于高通、带通、带阻滤波器。

对于高通模拟滤波器，$\omega_p>\omega_s$. 对于带通和带阻滤波器存在两个过渡带，ω_p 和 ω_s 均应为两个元素的向量，分别表示两个过渡带的边界频率。这时返回值 ω_n 也为两个元素的行

向量。

【例3.10】 设计一个模拟低通 Butterworth 滤波器和 Chebyshev 滤波器，满足：通带边界频率 $\omega_p=200\pi$，阻带边界频率 $\omega_s=300\pi$，通带波纹 $R_p=1$dB，阻带衰减 $R_s=16$dB，试确定其阶数 N 和 −3dB 截止频率 ω_c.

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 3-10
%Buttworth Lowpass Filter
Wp=200 * pi;
Ws=300 * pi;
Rp=1;
Rs=16;
disp('Buttworth Lowpass Filter')
[N,Wc]=buttord(Wp,Ws,Rp,Rs,'s')

%Chebyshev I I type Lowpass Filter
Wp=200 * pi;
Ws=300 * pi;
Rp=1;
Rs=16;
disp('Chebyshev I I type Lowpass Filter')
[N,Wc]=cheb2ord(Wp,Ws,Rp,Rs,'s')

≫smp310
Buttworth Lowpass Filter
N =
     7
Wc =
  725.7292
Chebyshev II type Lowpass Filter
N =
     4
Wc =
  839.8376
```

程序运行结果分别和例3.7、例3.9完全相同。

3.5 模拟滤波器设计函数

在3.3.2节中已经介绍了模拟滤波器的设计步骤，设计过程中要分步调用 MATLAB 有关工具函数：首先调用 buttap，cheblap 等函数产生模拟低通原型滤波器，然后调用

lp2lp、lp2bp 等频率转换函数，将低通转换为高通、带通、带阻等各类模拟滤波器，编程比较麻烦。

MATLAB 信号处理工具箱还提供模拟滤波器的完全设计函数：butter，cheby 1，cheby 2，ellip，besself。用户只需调用一次设计函数就可自动完成全部设计过程，编程十分简单。这些工具函数用于模拟滤波器设计，也适用于数字滤波器。本节只讨论这些函数在模拟滤波器设计中的应用，关于数字滤波器的设计将在第四章中讨论。

3.5.1 巴特沃斯模拟滤波器

函数 BUTTER 用于 Butterworth 滤波器设计，调用格式

```
[b, a] = butter(n, ωn, 's')
[b, a] = butter(n, ωn, 'ftype', 's')
[z, p, k] = butter(…)
[A, B, C, D] = butter(…)
```

其中，n 为滤波器阶数；ω_n 为滤波器截止频率，rad/s($\omega_n>0$)；'s'为模拟滤波器，缺省时为数字滤波。

'ftype'滤波器类型：

'hign'为高通滤波器，截止频率 ω_n；

'stop'为带阻滤波器，$\omega_n=[\omega_1,\omega_2](\omega_1<\omega_2)$；

'ftype'缺省时为低通或带通滤波器。

低通、高通滤波器时，ω_n 为截止频率；带通或带阻滤波器时，$\omega_n=[\omega_1, \omega_2](\omega_1<\omega_2)$。a，b 分别为滤波器的传递函数分子和分母多项式系数向量；z，p，k 分别为滤波器的零极点和增益；A，B，C，D 为滤波器状态空间表达式矩阵。

滤波器传递函数具有下面形式

$$H(s)=\frac{B(s)}{A(s)}=\frac{b(1)s^n+b(2)s^{n-1}+\cdots+b(n+1)}{a(1)s^n+a(2)s^{n-1}+\cdots+a(n+1)}$$

状态空间表达式

$$\dot{x}=Ax+Bu$$
$$y=Cx+Du$$

若滤波器为带通或带阻型，则滤波器的阶数为2n，否则阶数为 n.

该函数和函数 buttord 连接使用，可方便而精确地设计最小阶数的 butterworth 模拟滤波器。

【例3.11】 设计一个 Butterworth 模拟带通滤波器，设计指标为：逼近频率1000～2000Hz，两侧过渡带宽500Hz，通带波纹1dB，阻带衰减大于100dB。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 3-11
%Design a Butterworth Analog bandpass filter
%Desired performents of the filter
wp=[1000 2000]*2*pi;
ws=[500 2500]*2*pi;
```

```
Rp=1;
Rs=100;

%Compute Oder and Cutoff frequency
[N,Wn]=buttord(wp,ws,Rp,Rs,'s');
N
Fc=Wn/(2*pi)
%Compute the filter
[b,a]=butter(N,Wn,'s');
%Output
w=linspace(1,3000,1000)*2*pi;
H=freqs(b,a,w);
magH=abs(H);
phaH=unwrap(angle(H));
plot(w/(2*pi),20*log10(magH));
xlabel('Frequency (Hz)');
ylabel('Magnidute (dB)');
title('Butterworth Analog bandpass filter')
grid on
```

程序运行结果：滤波器为23阶，两个截止频率：$w_{c1}=989.9$Hz，$w_{c2}=2020.4$Hz

滤波器幅频特性如图3.8。

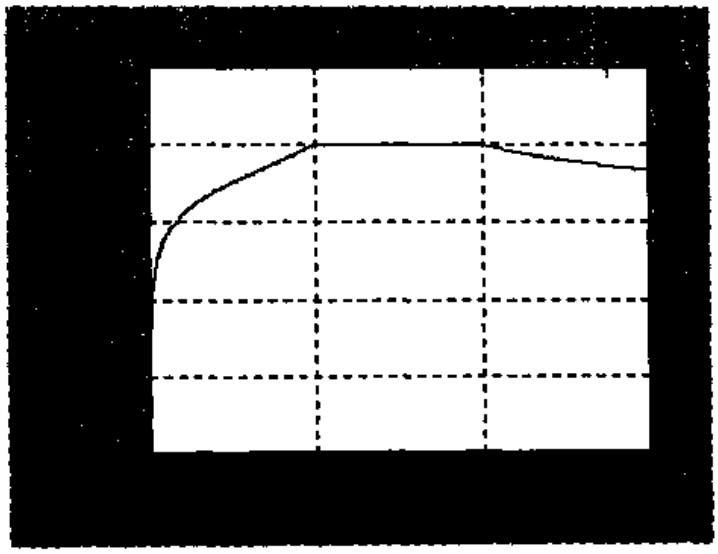


图3.8 Butterworth 带通滤波器幅频特性

3.5.2 切比雪夫模拟滤波器

一、切比雪夫Ⅰ型模拟滤波器

函数 cheby 1用于 chebyshev Ⅰ型模拟滤波器的设计。调用格式为

[b, a] = cheby 1 (n, R_p, ω_n, 's')

[b, a] = cheby 1(n, R_p, ω_n, 'ftype', 's')

[z, p, k] = cheby1(…)

[A, B, C, D] = cheby1(…)

其中，R_p 为通带波纹(dB)，其余各项意义与函数 cheby 1相同。

【例3.12】 设计一个 chebyshev Ⅰ型模拟带阻滤波器，设计指标为：阻带频率1000Hz～2000Hz，两侧过渡带宽500Hz，通带波纹1dB，阻带衰减大于50dB。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 3-12
%Design a Chebyshev I Analog bandstop filter
%Desired performents of the filter
wp=[500 2500]*2*pi;
ws=[1000 2000]*2*pi;
Rp=1;
Rs=50;

%Compute Oder and Cutoff frequency
[N,Wn]=cheb1ord(wp,ws,Rp,Rs,'s');
N
%Compute the filter
[b,a]=cheby1(N,Rp,Wn,'stop','s');
%Output
w=linspace(1,3000,1000)*2*pi;
H=freqs(b,a,w);
magH=abs(H);
phaH=unwrap(angle(H));
plot(w/(2*pi),20*log10(magH));
xlabel('Frequency (Hz)');
ylabel('Magnidute (dB)');
title('Chebyshev I Analog bandstop filter')
grid on
```

运行结果，滤波器阶数为7。滤波器的幅频特性如图3.9所示。

二、切比雪夫Ⅰ型模拟滤波器

函数 cheby2用于 chebyshev Ⅱ型模拟滤波器的设计。调用格式为

[b, a] = cheby 2 (n, R_s, ω_n, 's')

[b, a] = cheby 2(n, R_s, ω_n, 'ftype', 's')

[z, p, k] = cheby 2(…)

[A, B, C, D] = cheby 2(…)

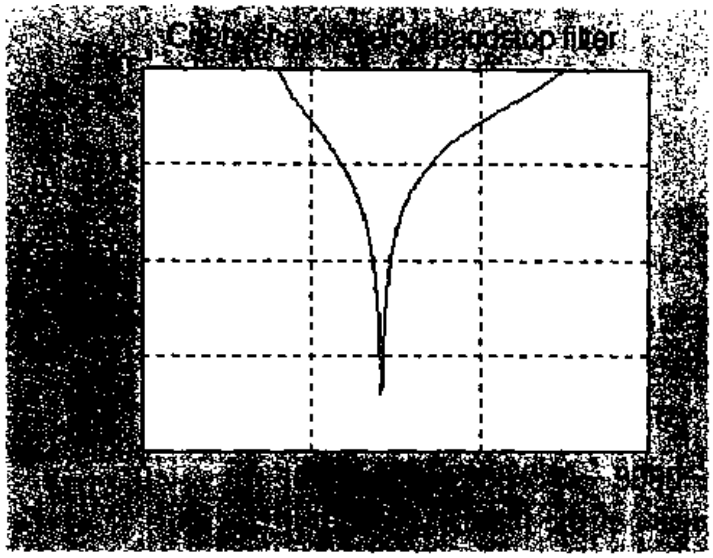


图3.9 chebyshev Ⅰ型带阻滤波器幅频特性

其中，R_s 为阻带衰减(dB)，其余各项意义与函数 butter 相同。

【例3.13】 设计一个 chebyshev Ⅱ型带阻滤波器设计指标同例3.12。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 3-13
%Design a Chebyshev II Analog bandstop filter
%Desired performents of the filter
wp=[500 2500]*2*pi;
ws=[1000 2000]*2*pi;
Rp=1;
Rs=50;

%Compute Oder and Cutoff frequency
[N,Wn]=cheb2ord(wp,ws,Rp,Rs,'s');
N
%Compute the filter
[b,a]=cheby2(N,Rs,Wn,'stop','s');
%Output
w=linspace(1,3000,1000)*2*pi;
H=freqs(b,a,w);
magH=abs(H);
phaH=unwrap(angle(H));
plot(w/(2*pi),20*log10(magH));
xlabel('Frequency (Hz)');
ylabel('Magnidute (dB)');
title('Chebyshev II Analog bandstop filter')
```

grid on

运行结果：滤波器阶数为7，滤波器幅频特性如图3.10所示。

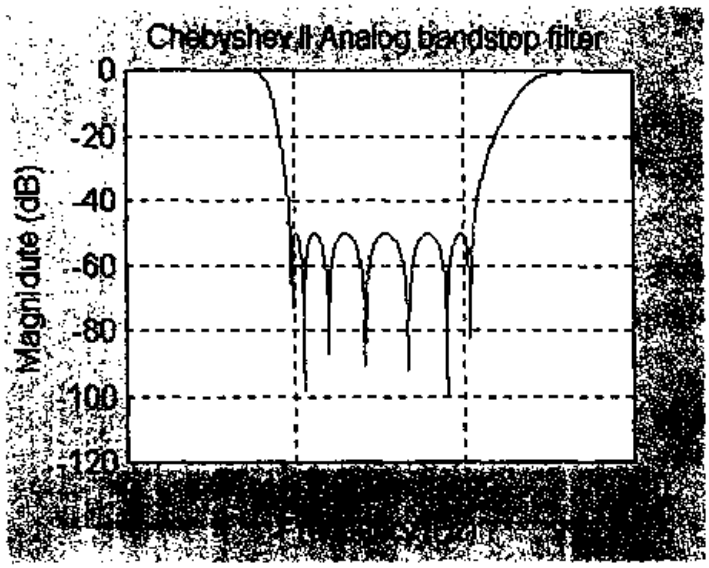


图3.10 chebyshev Ⅰ型带阻滤波器幅频特性

3.5.3 椭圆模拟滤波器

函数 ELLIP 用于椭圆模拟滤波器设计。调用格式为

$[b, a] = \text{ellip}(n, R_p, R_s, \omega_n, 's')$

$[b, a] = \text{ellip 2}(n, R_p, R_s, \omega_n, 'ftype', 's')$

$[z, p, k] = \text{ellip 2}(\cdots)$

$[A,B,C,D] = \text{ellip 2}(\cdots)$

其中，R_p 为通带波纹(dB)，R_s 为阻带衰减(dB)，其余各项意义和函数 butter 相同。

【例3.14】 设计一个高通椭圆模拟滤波器。设计性能指标要求：通带边界频率 ω_p=1500Hz，阻带边界频率 ω_s=1000Hz，通带波纹 R_p=1dB，阻带衰减 R_s=20dB。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 3-14
%Design a Elliptic Analog highpass filter
%Desired performents of the filter
wp=1500 * 2 * pi;
ws=1000 * 2 * pi;
Rp=1;
Rs=20;

%Compute Oder and Cutoff frequency
[N,Wn]=ellipord(wp,ws,Rp,Rs,'s');
N
%Compute the filter
[b,a]=ellip(N,Rp,Rs,Wn,'high','s');
```

```
%Output
w=linspace(1,3000,1000)*2*pi;
H=freqs(b,a,w);
magH=abs(H);
phaH=unwrap(angle(H));
plot(w/(2*pi),20*log10(magH));
xlabel('Frequency (Hz)');
ylabel('Magnidute (dB)');
title('Elliptic highpass filter')
grid on
```

运行结果，滤波器为3阶。滤波器的幅频特性如图3.11所示。

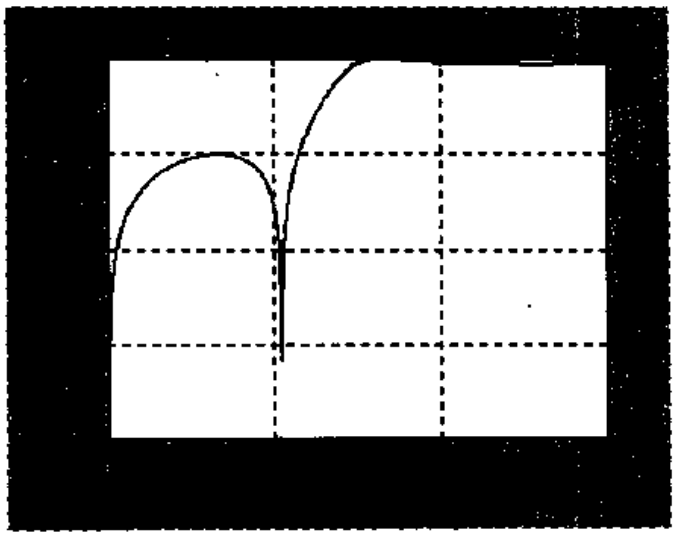


图3.11 椭圆高通模拟滤波器

3.5.4 贝塞尔模拟滤波器

函数 BESSELF 用于设计贝塞尔(Bessel)模拟滤波器。该函数不适于数字滤波器。函数调用格式为

$$[b, a] = \text{besself}(n, \omega_n)$$
$$[b, a] = \text{besself}(n, \omega_n, 'ftype')$$
$$[z, p, k] = \text{besself}(\cdots)$$
$$[A, B, C, D] = \text{besself}(\cdots)$$

其中，n 为滤波器阶数，ω_n 为滤波器截止频率(−3dB 频率)，其中各项意义及用法同函数 butt。

【例3.15】 设计一个5阶巴塞尔低通模拟滤波器，截止频率为1000rad/s，绘制滤波器的频率特性图。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 3-15
clf
```

```
%Design a Bessel Analog lowpass filter
%Desired performents of the filter
N=5;
Wn=1000;
%Compute the filter
[b,a]=besself(N,Wn);
%Output
freqs(b,a);
```

绘制的巴塞尔低通滤波器如图3.12所示。

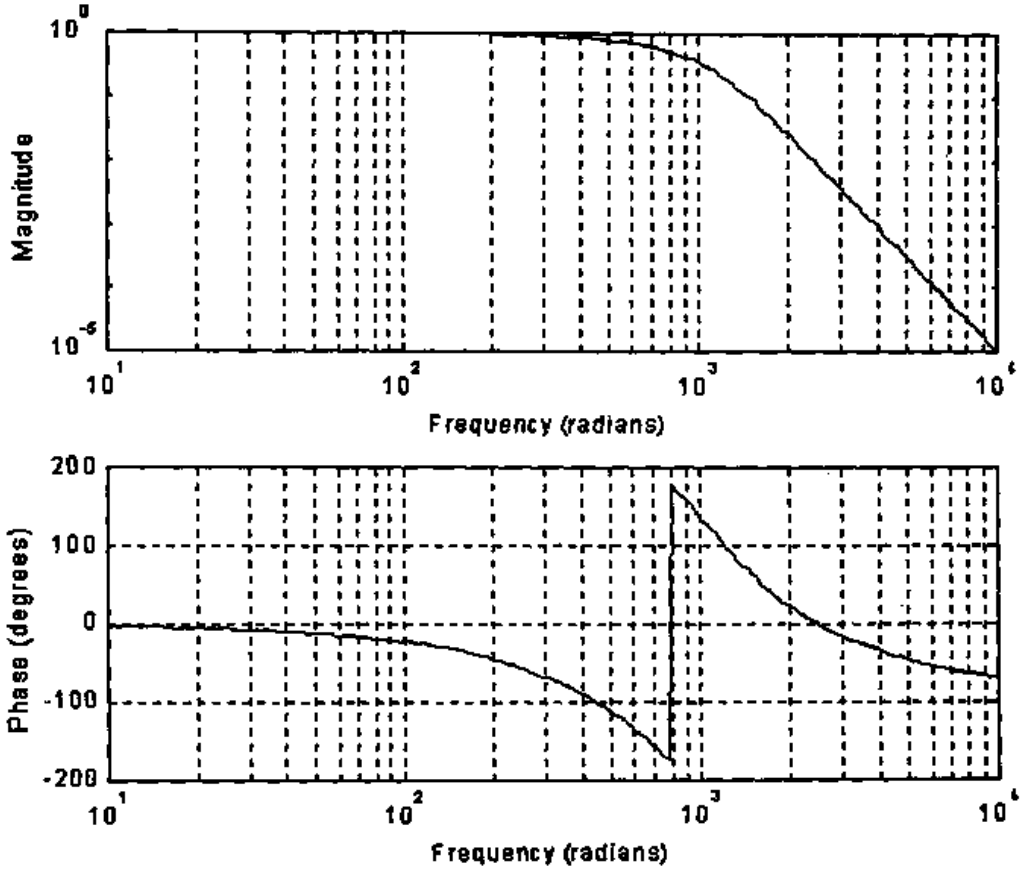


图3.12 Bessel 低通滤波器频率特性

习 题

3.1 模拟滤波器的可分为哪几种类型?设计时要求的主要性能指标有哪些?

3.2 设计一个巴特沃斯模拟低通滤波器,要求满足下面指标:截止频率(-3dB 频率)ω_c=1000Hz,通带波纹小于3dB,阻带始点频率 ω_s=1500Hz,阻带衰减大于35dB。

(1) 确定滤波器的最小阶次;

(2) 滤波器的传递函数;

(3) 绘制滤波器的平方幅频和相频特性图;

(4) 绘制滤波器的零极点图；

(5) 当输入信号为 $x(t)=\sin(2\pi f_1 t)+\cos(2\pi f_2 t)$，$f_1=500Hz$，$f_2=2000Hz$，绘制滤波器输出信号曲线；

(6) 绘制滤波器的单位脉冲响应曲线。

3.3 确定三阶切比雪夫 I 型低通模拟滤波器的系统函数，性能指标为：通带波纹为1dB，截止频率为 $\omega_c=2rad/s$。

3.4 一个低通模拟滤波器要求满足下面条件：边界频率为500Hz 和750Hz，通带波纹为2dB，阻带衰减40dB，试求满足此要求的巴特沃斯滤波器和切比雪夫 I 型滤波器的最小阶数。

3.5 设计一个高通滤波器，边界频率为100Hz 和150Hz，通带波纹不大于2dB，阻带衰减不小于30dB，试设计满足此要求的巴特沃斯、切比雪夫和椭圆滤波器，并绘制它们的平方幅频响应图。

3.6 设计带通巴特沃斯模拟滤波器，要求满足下面特性指标：中心频率1kHz，−3dB带宽为200Hz。通带波纹1dB 以下，阻带衰减40dB 以上。确定滤波器阶数、系统函数，并绘制其幅频特性图、相频特性图和零极点图。

3.7 设计一个巴塞尔低通模拟滤波器，性能指标要求同习题3.4，绘制滤波器的平方幅频图和相频图。

第四章　数字滤波器设计

数字滤波是数字信号处理技术的重要内容。和模拟滤波器一样，数字滤波器的主要功能是对数字信号进行处理，最常见的处理是保留数字信号中的有用成分，去除信号中无用成分。

本章重点介绍 IIR 和 FIR 两种类型的数字滤波器的设计方法及 MATLAB 实现。

4.1　概述

4.1.1　数字滤波器的工作原理

数字滤波器是具有一定传输特性的数字信号处理装置。与模拟滤波器不同，它的输入和输出均为离散的数字信号，借助于数字器件或一定的数值计算方法，对输入信号进行处理，改变输入信号的波形或频谱，达到保留信号中有用成分去除无用成分的目的。如果在数字信号处理系统(DSP)前后加上 A/D 和 D/A 转换，它可以用于处理模拟信号，作用相当于模拟滤波器，如图 4.1 所示。

图 4.1　数字滤波器

本章讨论的数字滤波器，它是一个离散时间系统，输入 x(n)是一个时间序列，输出 y(n)也是一个时间序列。若数字滤波器的系统函数为 H(z)，其脉冲响应序列 h(n)，则在时域内，存在下面关系

$$y(n) = x(n) * h(n) \tag{4.1-1}$$

在 z 域内，输入和输出存在下面关系

$$Y(z) = H(z)X(z) \tag{4.1-2}$$

式中，X(z)、Y(z)分别为输入 x(n)和输出 y(n)的 z 变换。

同样，在频域内，输入和输出存在下面关系

$$Y(j\omega) = H(j\omega)X(j\omega) \tag{4.1-3}$$

式中，H(jω)为数字滤波器的频率特性；x(jω)和 y(jω)分别为 x(n)和 y(n)的频谱；ω 为数字角频率，rad。

为了讨论方便起见，本章内，ω 表示数字角频率，单位：弧度(rad)，而 Ω 表示模拟角频率，单位：弧度/秒(rad/s)。数字角频率 Ω 和模拟角频率存在以下映射关系

$$\omega = \Omega T \tag{4.1-4}$$

式中，T 为采样周期，单位：秒(s)，数字角频率 Ω 在 0～π 范围内。

由式(4.1-1)至(4.1-3)可知，一个合适的数字滤波器系统函数 H(z)可以改变输入 x(n)的频率特性。经数字滤波器处理后的信号 y(n)，保留信号 x(n)中有用频率成分，去除无用的频率成分。

4.1.2 数字滤波器分类

按时域特性，数字滤波器可以分为无限冲激响应数字滤波器(Infinite Impulse Response Digital Filter，简称 IIR 滤波器)和有限冲激响应数字滤波器(Finite Impulse Response Digital Filter，简称 FIR 滤波器)两类。

IIR 滤波器的系统函数为

$$H(z)=\frac{Y(z)}{X(z)}=\sum_{n=0}^{\infty}h(n)z^{-n}=\frac{\sum_{r=0}^{M}b_r z^{-r}}{1+\sum_{k=1}^{N}a_k z^{-k}} \tag{4.1-5}$$

该滤波器的脉冲响应 h(n)从 n=0～∞均有值。

FIR 滤波器的系统函数为

$$H(z)=\frac{Y(z)}{X(z)}=\sum_{n=0}^{N-1}h(n)z^{-n} \tag{4.1-6}$$

该滤波器的脉冲响应 h(n)在 n=0,1,…，N−1 的有限个 N 点上有值。注意到式(4.1-5)，当分母 a_k 全为零时，H(z)具有全零点形式，此时，式(4.1-5)变为式(4.1-6)，滤波器变为 FIR 形式。

按频域特性来分，数字滤波器和模拟滤波器一样可分为低通、高通、带通和带阻等。数字滤波器是一个离散时间系统，在频率特性具有周期性，因此我们讨论的频率范围仅在 ω=0～π 范围内，相应的标准化频率在 0～1 之间。

和模拟滤波器一样，理想数字滤波器的频率特性 $H_d(j\omega)$在通带内必须满足

$$\begin{cases}|H_d(j\omega)|=K\\ \angle H_d(j\omega)=-\alpha\omega\end{cases} \tag{4.1-7}$$

式中，K，α 均为常数。

和模拟滤波器一样，数字滤波器的设计目的是使滤波器的频率特性达到所给定的性能指标。

4.2 IIR 数字滤波器的设计方法

IIR 滤波器是一种数字滤波器，滤波器的系统函数如式(4.1-5)所示，由于它的脉冲响应序列 h(n)是无限长的，故称无限冲激响应滤波器。IIR 滤波器的设计就是根据滤波器某些性能指标要求，设计滤波器的分子和分母多项式。它和 FIR 滤波器相比优点是在满足相同性能指标要求条件下，IIR 滤波器的阶数要明显低于 FIR 滤波器。但 IIR 滤波器的相位是非线性的；在 MATLAB 内，数据处理通常是离线的，即在滤波前整个的数据序列可以利用，这样可采用非因果、零相位滤波逼近方法。

IIR 滤波器设计方法可分为三种：模拟滤波器变换(经典设计法)、直接设计法、参数模型设计法、最大平滑滤波器设计。表 4.1 给出了 IIR 数字滤波器设计主要方法及

MATLAB 工具函数。

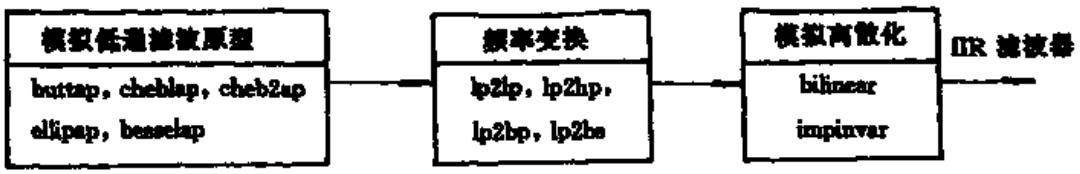
表 4.1

设计方法	说　明	工具函数
经典设计法	首先设计满足性能要求的模拟滤波器；再离散化变换为数字滤波器。	完全设计函数： butter，cheby1，cheby2，ellip，besself(模拟) 阶数估计函数： buttord，cheb1ord，cheb2ord，ellipord 低通模拟原型滤波器函数： buttap，cheb1ap，cheb2ap，ellipord 频率转换函数： lp2lp，lp2bp，lp2hp，lp2bs 滤波器离散化函数： bilinear，impinvar
直接设计法	在离散域内用最小二乘法逼近给定的幅频特性	yulewalk
参数模型设计法(见第五章)	采用参数模型逼近给定时域或频率响应求得滤波器	时域建模函数： lpc，prony，stmcb 频域建模函数 invfreqs(模拟)，invfreqz(数字)
最大平滑滤波器(一般化巴特沃斯滤波器)	设计一般化低通模拟滤波器，其零点多于极点	maxflat

4.3 IIR 滤波器经典设计

4.3.1 模拟滤波器变换方法

基于模拟滤波器变换原理，首先是先根据滤波器的技术指标设计出相应的模拟滤波器，然后再将设计好的模拟滤波器变换成满足给定技术指标的数字滤波器。这就是 IIR 数字滤波器设计的经典法。经典法可设计出低通、高通、带通、带阻等各类 IIR 滤波器。在 MATLAB 中，经典法设计 IIR 数字滤波器采用下面的主要步骤：



由上可见，经典设计法是利用模拟滤波器的设计成果。第一步和第二步都在第三章模拟滤波器设计中已经详细讨论过，第二步完成后，一个达到期望性能指标的模拟滤波器

(低通、高通、带通和带阻等)已经设计出来。第三步离散化主要任务就是把模拟滤波器变换成数字滤波器,即把模拟滤波器的系统函数 H(s)映射成数字滤波器的系统函数 H(z)。这样,数字滤波器的设计工作全部完成。

实现系统传递函数 s 域至 z 域映射有脉冲响应不变法和双线性映射法两种方法。下面分别介绍这两种方法设计要点及 MATLAB 实现。

一、脉冲响应不变法

所谓脉冲响应不变法就是使数字滤波器的脉冲响应序列 h(n)等于模拟滤波器的脉冲响应 $h_a(t)$的采样值,即

$$h(n) = h_a(t)\big|_{t=nT} = h_a(nT) \tag{4.3-1}$$

式中,T 为采样周期

因此,数字滤波器的系统函数 H(z)可由下式求得:

$$H(z) = \mathcal{Z}[h(n)] = \mathcal{Z}[h_a(nT)] \tag{4.3-2}$$

如果,已经获得了满足性能要求的模拟滤波器的系统函数 $H_a(s)$,求与之对应的数字滤波器的系统函数 H(z)的方法是:

(1) 求模拟滤波器的单位脉冲响应 h_a

$$h_a(t) = \mathcal{L}^{-1}[H_a(s)] \tag{4.3-3}$$

(2) 由式(4.3-1)求模拟滤波器采样值,即数字滤波器脉冲响应序列 h(n)。

(3) 由式(4.3-2)求得数字滤波器的系统函数 H(z)。

由上述方法推论出更直接地由模拟滤波器系统函数 $H_a(s)$求出数字滤波器系统函数 H(z)的步骤是:

(1) 利用部分分式展开式,把模拟滤波器系统函数 $H_a(s)$展开成下面形式:

$$H_a(s) = \sum_{k=1}^{N} \frac{R_k}{s - p_k} \tag{4.3-4}$$

(2) 将模拟极点 p_k 变换为数字极点 e^{P_kT},式(4.3-4)变换为数字滤波器系统函数

$$H(z) = \sum_{k=1}^{N} \frac{R_k}{1 - e^{P_kT}z^{-1}} \tag{4.3-5}$$

由式(4.3-4)和式(4.3-5)可见,在脉冲响应不变法中,$H_a(s)$和 H(z)的变换关系为:$H_a(s)$在 s 平面上有一个极点 $s=s_i$,则 H(z)在 z 平面上相应也有一极点 $z_i=e^{S_iT}$。

值得注意的是,不能将关系 $z=e^{ST}$直接代入 $H_a(s)$来获取 H(z),因为两者的零点不存在这种关系。

按上面方法可用 MATLAB 编程由滤波器的模拟系统函数 $H_a(s)$获得数字系统函数 H(z)。同时,MATLAB 信号处理工具箱又提供了专用的工具函数。函数 IMPINVAR 基于脉冲响应不变法实现由模拟滤波器至数字滤波器的变换,调用格式为

```
[bz, az]= impinvar(b, a, Fs)
[bz, az]= impinvar(b, a)
```

其中,b, a 分别为模拟滤波器的分子和分母多项式系数向量;F_s 为采样频率,Hz,缺省值 F_s=1Hz;bz,az 分别为数字滤波器分子和分母多项式系数向量。

【例 4.1】 用脉冲响应不变法将模拟滤波器 $H_a(s)=\frac{3s+2}{2s^2+3s+1}$ 变换为数字滤波器 H(z)，采样周期 $T_s=0.1s$。

下面的 MATLAB 程序首先采用直接编程法计算 H(z)，再利用函数 impinvar 计算 H(z)，比较其结果，程序清单如下：

```
%MATLAB PROGRAM 4-1
%Impulse invariance method
%From analog to digital filter conversion
%Model of analog filter
b=[3 2];
a=[2 3 1];
Ts=0.1;
[R,Ps,K]=residue(b,a);
Pz=exp(Ps * Ts);
disp('Use direct principle:')
[bz,az]=residue(R,Pz,K)
disp('Use function "IMPINVAR":')
[bz1,az1]=impinvar(b,a,1/Ts)

≫ smp401
Use direct principle:
bz =
    1.5000    -1.4036
az =
    1.0000    -1.8561    0.8607
Use function "IMPINVAR":
bz1 =
    0.3000    -0.2807
az1 =
    2.0000    -3.7121    1.7214
```

在应用脉冲响应不变法设计时注意它的特点。脉冲响应不变法是数字角频率 ω 和模拟角频率 Ω 满足 $\omega=\Omega T$ 线性变换关系，如果模拟滤波器频响是带限的话，通过变换所得的数字滤波器的频响可以非常接近于模拟滤波器的频响。但由于数字滤波器的频响是模拟滤波器频响的周期延拓，因此存在混叠效应而造成的频响失真，因此这种方法原则上只适用于带限滤波器，如低通和带通滤波器。对于高通、带阻等滤波器，由于它们高频部分不衰减，势必产生严重的混叠失真。

二、双线性变换法

由于 s 平面和 z 平面的单值双线性映射关系为

$$s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} \tag{4.3-6}$$

$$z = \frac{1+\frac{T}{2}s}{1-\frac{T}{2}s} \tag{4.3-7}$$

式中，T 为采样周期。

因此，若已知模拟滤波器的系统函数 $H_a(s)$，将式(4.3-6)关系代入 $H_a(s)$即可得到数字滤波器的系统函数 H(z)，即

$$H(z) = H_a(s)\Big|_{s=\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}} \tag{4.3-8}$$

在双线性变换中，模拟角频率 Ω 和数字角频率 ω 存在下面关系

$$\Omega = \frac{2}{T}\tan\frac{\omega}{2} \tag{4.3-9}$$

或

$$\omega = 2\arctan\frac{\Omega T}{2} \tag{4.3-10}$$

式中，Ω 的单位为弧度/秒，ω 的单位为弧度，在 0～π 之间。

可见，Ω 和 ω 之间的变换关系是非线性的。

按上面方法可用 MATLAB 编程实现模拟滤波器 $H_a(s)$至数字滤波器 H(z)的变换。MATLAB 信号处理工具箱提供了实现双线性变换的工具函数。

函数 BILINEAR 采用双线性变换法实现模拟 s 域至数字 z 域的映射，直接用于模拟滤波器变换为数字滤波器。对于不同形式模拟滤波器模型，函数调用格式不同。

(1) 零极点增益形式

$$[z_d, P_d, K_d] = \text{bilinear}(z, p, k, F_s)$$

$$[z_d, P_d, K_d] = \text{bilinear}(z, p, k, F_s, F_p)$$

式中，z，p 分别为模拟滤波器零、极点列向量；k 为模拟滤波器增益；F_s 为采样频率，Hz；F_p 为预畸变频率，Hz，函数选择项；z_d，P_d，K_d 为数字滤波器的零、极点增益。

若 F_p 缺省时，函数按下式实现 s 平面至 z 平面映射：

$$H(z) = H_a(s)\Big|_{s=2f_s\frac{z-1}{z+1}}$$

s 平面虚轴 jΩ(Ω=－∞→∞)映射至 z 平面单位圆 $e^{j\omega}$(ω=－π→π)，按下式计算：

$$\omega = 2\arctan\frac{\Omega}{2f_s}$$

若函数输入项预畸变频率 F_p 给定后，函数按下式实现 s 平面至 z 平面映射：

$$H(z) = H_a(s)\Big|_{s=\frac{2\pi f_p}{\tan(\pi\frac{f_p}{f_s})}\cdot\frac{z-1}{z+1}}$$

s 平面虚轴 jΩ(ω=－∞→∞)映射至 z 平面单位圆 $e^{j\omega}$(ω=－π→π)，按下式计算：

$$\omega = 2\arctan\left[\frac{\Omega\tan\left(\pi\frac{f_p}{f_s}\right)}{2\pi f_p}\right]$$

预畸变频率 F_p(Prewarped frequeney)是一个“匹配”频率。该频率点上，频率响应在双

线性变换前后可精确地匹配。

(2) 传递函数形式

$$[\text{numd}, \text{dend}]=\text{bilinear}(\text{num}, \text{den}, F_s)$$

$$[\text{numd}, \text{dend}]=\text{bilinear}(\text{num}, \text{den}, F_s, F_p)$$

其中,num, den 分别为模拟滤波器传递函数分子和分母多项式系数向量,模拟滤波器传递函数具有下面形式

$$\frac{\text{num}(s)}{\text{den}(s)}=\frac{\text{num}(1)s^{nn}+\cdots+\text{num}(nn)s+\text{num}(nn+1)}{\text{den}(1)s^{nd}+\cdots+\text{den}(nd)s+\text{den}(nd+1)}$$

numd 和 dend 分别为数字滤波器传递函数分子和分母多项式系数向量。F_s 和 F_p 意义同上述。

(3) 状态空间形式

$$[A_d, B_d, C_d, D_d]=\text{bilinear}(A, B, C, D, F_s)$$

$$[A_d, B_d, C_d, D_d]=\text{bilinear}(A, B, C, D, F_s, F_p)$$

其中,模拟滤波器为

$$\begin{cases}\dot{x}=Ax+Bu\\ y=Cx+Du\end{cases}$$

数字滤波器为

$$\begin{cases}x[n+1]=A_dx[n]+B_du[n]\\ y[n]=C_dx[n]+D_du[n]\end{cases}$$

函数实现从模拟滤波器至数字滤波器的变换。

【例 4.2】 用双线性变换法将模拟滤波器 $H_a(s)=\frac{3s+2}{2s^2+3s+1}$ 变换为数字滤波器 H(z),采样频率 T=0.1s。

用 MATLAB 编程如下:

```
%MATLAB PROGRAM 4-2
%Bilinear transformation
%Model of analog filter
b=[3 2];
a=[2 3 1];
Ts=0.1;
%From analog to digital filter conversion
[bz1,az1]=bilinear(b,a,1/Ts)
```

```
≫ smp402
bz1 =
    0.0720    0.0046   -0.0674
az1 =
    1.0000   -1.8560    0.8606
```

双线性变换法克服了脉冲响应不变法的频谱混叠问题，其幅值逼近程度好，且也可适用于高通、带阻等各种滤波器设计。s 域和 z 域对应关系也简单。缺点是，如式(4.3-9)和式(4.3-10)所示的频率变换非线性导致数字滤波器与模拟滤波器在幅度与频率的对应关系上发生畸变。但一般滤波器的幅频响应具有分段常数特点，故变换后，这一特点仍保留，影响不大。由数字边界频率计算模拟边界频率时，不是按线性关系而是按式(4.2-9)，这就是所谓"预畸变"。经预畸变换可保证所要设计的模拟边界频率正好映射在所要求的数字边界频率上。在数字滤波器设计时，在双线性变换中应用预畸变方法实现频率性能指标转换。

4.3.2 经典设计法

IIR 数字滤波器经典设计法的一般步骤是：

(1) 根据给定的性能指标和方法不同，首先对设计性能指标中的频率指标，如边界频率进行转换，转换后的频率指标作为模拟滤波器原型设计性能指标。

(2) 估计模拟滤波器最小阶数和边界频率，利用 MATLAB 工具函数 buttord、cheb1ord、cheb2ord、ellipord 等。

(3) 设计模拟低通滤波器原型，利用 MATLAB 工具函数 buttap、cheb1ap、cheb2ap、ellipap 等。

(4) 由模拟低通原型经频率变换获得模拟滤波器(低通、高通、带通、带阻等)，利用 MATLAB 工具函数 lp2lp、lp2hp、lp2bp、lp2bs。

(5) 将模拟滤波器离散化获得 IIR 数字滤波器，利用 MATLAB 工具函数 bilinear、impinvar。

后面三个步骤及工具函数调用已在第三章作了详细阐述，这里介绍第一步：关于设计性能指标的转换。

设计 IIR 滤波器时，给出的性能指标通常分数字指标和模拟指标两种。

数字性能指标给出通带截止频率 ω_p，阻带起始频率 ω_s，通带波纹 R_p，阻带衰减 R_s 等。数字频率 ω_c 和 ω_s 的取值范围为 0～π，单位：弧度，而 MATLAB 工具函数常采用标准化频率，ω_c 和 ω_s 的取值范围为 0～1。

模拟性能指标给出通带截止频率 Ω_p，阻带起始频率 Ω_s，通带波纹 R_p，阻带衰减 R_s 等。模拟频率 Ω_c 和 Ω_s 的单位均为弧度/秒。

MATLAB 信号处理工具箱中，设计性能指标的转换应根据不同设计方法进行不同处理：

(1) 脉冲响应不变法，可按下图对设计性能中的频率指标作处理。

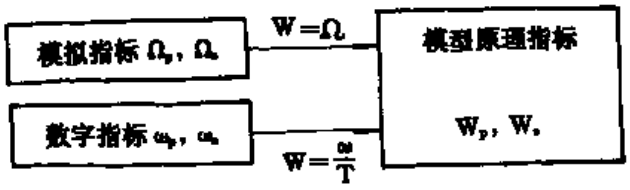


图 4.2 脉冲响应不变法性能指标转换

【例 4.3】 用脉冲响应不变法设计一个 Butterworth 低通数字滤波器，使其特性逼近一个低通 Butterworth 低通模拟滤波器的下列性能指标：

通带截止频率 $\Omega_p = 2\pi \times 2k$ rad/s，阻带边界频率 $\Omega_s = 2\pi \times 4k$ rad/s，通带波纹 R_p 小于 3dB，阻带衰减大于 15dB，采样频率 $F_s = 20000$Hz。

由于此例给出模拟指标，采用的是脉冲响应不变法设计，因此所给出的性能指标可直接作为模拟原型指标使用。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 4-3
%Design a Butterworth digital lowpass filter
%Desired performents of the filter
wp=2000 * 2 * pi;
ws=3000 * 2 * pi;
Rp=3;
Rs=15;
Fs=10000;
Nn=128;
%Compute Oder and Cutoff frequency
[N,Wn]=buttord(wp,ws,Rp,Rs,'s')
%Compute the analog filter
[z,p,k]=buttap(N);
[Bap,Aap]=zp2tf(z,p,k);
[b,a]=lp2lp(Bap,Aap,Wn);
%Impulse invariance method
%for analog to digital filter conversion.
[bz,az]=impinvar(b,a,Fs)
freqz(bz,az,Nn,Fs)

≫ smp403

N =
     5
Wn =
  1.3387e+004

bz =
    0.0000    0.0703    0.2923    0.1253    0.0053         0
az =
    1.0000   -1.1280    0.9642   -0.4467    0.1166   -0.0131
```

【例 4.4】 用脉冲响应不变法设计 Butterworth 低通数字滤波器，要求通带频率为 $0\leqslant\omega\leqslant0.2\pi$，通带波纹小于 1dB，阻带在 $0.3\pi\leqslant\omega\leqslant\pi$ 内，幅度衰减大于 15dB，采样周期 $T_s=0.01s$。

该题给出的通带边界频率 $\omega_p=0.2\pi$，阻带边界频率 $\omega_s=0.3\pi$ 均为数字频率，因此设计时，首先将其换算成模拟频率 Ω_p 和 Ω_s 作为模拟滤波器原形频率指标 W_p 和 W_s。

$$W_p=\Omega_p=\frac{\omega_p}{T_s}$$

$$W_s=\Omega_s=\frac{\omega_s}{T_s}$$

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 4-4
%Design a Butterworth digital lowpass filter
%Desired performents of the filter
wp=0.2*pi;
ws=0.3*pi;
Rp=1;
Rs=15;
Ts=0.01;
Nn=128;
%Convert digital frequency to analog frequency
Wp=wp/Ts;
Ws=ws/Ts;
%Compute Oder and Cutoff frequency
[N,Wn]=buttord(Wp,Ws,Rp,Rs,'s')
%Compute the analog filter
[z,p,k]=buttap(N);
[Bap,Aap]=zp2tf(z,p,k);
[b,a]=lp2lp(Bap,Aap,Wn);
%Impulse invariance method
%for analog to digital filter conversion.
[bz,az]=impinvar(b,a,1/Ts)
freqz(bz,az,Nn,1/Ts)

≫ smp404
N =
     6
Wn =
    70.8654
```

bz =

0.0000　　0.0007　　0.0105　　0.0167　　0.0042　　0.0001　　0

az =

1.0000　　−3.3443　　5.0183　　−4.2190　　2.0725　　−0.5600　　0.0647

(2) 双线性变换法，MATLAB 信号处理工具箱按图 4.3 对性能指标中的频率指标进行处理。

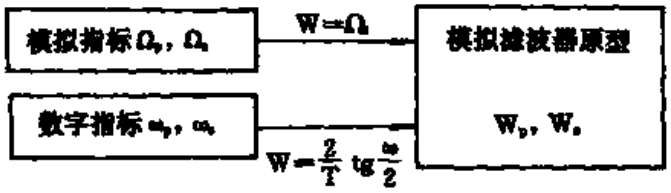


图 4.3　双线性变换法性能指标转换

【例 4.5】 用双线变换设计一个 Butterworth 低通滤波器，其性能指标同【例 4.3】。该题所给性能指标为模拟指标，用 MATLAB 编程如下：

```
%MATLAB PROGRAM 4-5
%Design a Butterworth digital lowpass filter
%Desired performents of the filter
wp=2000 * 2 * pi;
ws=3000 * 2 * pi;
Rp=3;
Rs=15;
Fs=10000;
Ts=1/Fs;
Nn=128;
%Compute Oder and Cutoff frequency
[N,Wn]=buttord(wp,ws,Rp,Rs,'s')
%Compute the analog filter
[z,p,k]=buttap(N);
[Bap,Aap]=zp2tf(z,p,k);
[b,a]=lp2lp(Bap,Aap,Wn);
%Bilinear transformation
%for analog to digital filter conversion.
[bz,az]=bilinear(b,a,Fs)
freqz(bz,az,Nn,Fs)

≫ smp405
N =
```

```
    5
Wn =
  1.3387e+004
bz =
    0.0171    0.0854    0.1708    0.1708    0.0854    0.0171
az =
    1.0000   -1.2271    1.1622   -0.5176    0.1450   -0.0159
```

【例 4.6】 用双线性变换法设计一个 Butterworth 低通滤波器，其性能指标同例 4.4。该题所给性能指标为数字指标，用 MATLAB 编程如下：

```
%MATLAB PROGRAM 4-6
%Design a Butterworth digital lowpass filter
%Desired performents of the filter
wp=0.2*pi;
ws=0.3*pi;
Rp=1;
Rs=15;
Ts=0.01;
Nn=128;
%Convert digital frequency to analog frequency
Wp=(2/Ts)*tan(wp/2);
Ws=(2/Ts)*tan(ws/2);
%Compute Oder and Cutoff frequency
[N,Wn]=buttord(Wp,Ws,Rp,Rs,'s')
%Compute the analog filter
[z,p,k]=buttap(N);
[Bap,Aap]=zp2tf(z,p,k);
[b,a]=lp2lp(Bap,Aap,Wn);
%Bilinear transformation
%for analog to digital filter conversion.
[bz,az]=bilinear(b,a,1/Ts)
freqz(bz,az,Nn,1/Ts)

≫ smp406
N =
    6
Wn =
    76.6229
bz =
```

0.0007　　0.0044　　0.0111　　0.0148　　0.0111　　0.0044　　0.0007

az =

1.0000　　−3.1836　　4.6222　　−3.7795　　1.8136　　−0.4800　　0.0544

4.3.3 IIR 滤波器完全设计函数

以上介绍了 IIR 数字滤波器设计原理和基本方法步骤，并给出了一些例子说明如用 MATLAB 编程分步实现这些设计，但必须多次调用 MATLAB 有关基本工具函数。

实际上，MATLAB 信号处理工具箱还提供了 IIR 滤波器设计的完全工具函数，用户只要调用这些工具函数即可一次性完成设计，而不需要调用那些基本工具函数分步实现。

IIR 滤波器设计的完全工具函数有 Butter、cheby1、cheby2、ellip。这些工具函数既可用于设计模拟滤波器，也适用于数字滤波器。关于这些函数在模拟滤波器设计中的应用已在第三章讨论过，这里介绍这些函数在 IIR 滤波器设计中的应用。在这两类滤波器设计中，这些工具函数调用格式基本相同，只是在频率处理上有所不同。

在 MATLAB 滤波器设计工具函数中，数字频率采用标准化频率，取值范围在 0～1 之间，标准化频率 1 对应的数字频率为 π，对应的模拟频率为采样频率的一半。在应用 MATLAB 工具函数设计数字滤波器时应注意这一特点。

一、巴特沃斯数字滤波器

函数 BUTTER 也可用于 Butterworth 数字滤波器，调用格式为

[b, a]= butter (n, ω_n)

[b, a]= butter (n, ωn,'ftype')

[z, p, k]= butter (…)

[A, B, C, D]= butter (…)

其中，n 为滤波器阶数；ω_n 为滤波器截止频率，0～1；'ftype'为滤波器类型：

- 'high'为高通滤波器，截止频率 ω_n
- 'stop'为带阻滤波器，截止频率 $\omega_n=[\omega_1,\omega_2]$，$(\omega_1<\omega_2)$
- 'ftype'缺省时为低通和带通滤波器，低通滤波器时，ω_n 为截止频率；带通滤波器时，$\omega_n=[\omega_1,\omega_2]$，$(\omega_1<\omega_2)$。

b，a 分别为滤波器传递函数分子和分母多项式系数向量；z，p，k 分别为滤波器的零点、极点和增益；A，B，C，D 分别为滤波器状态空间表达式。

数字滤波器传递函数具有下面形式

$$H(z)=\frac{B(z)}{A(z)}=\frac{b(1)+b_2z^{-1}+\cdots+b(n+1)z^{-n}}{1+a(2)z^{-1}+\cdots+a(n+1)z^{-n}}$$

状态空间表达式为

$$\begin{cases}X[n+1]=Ax[x]+Bu[n]\\Y[n+1]=Cx[x]+Du[n]\end{cases}$$

函数 butter 用于 IIR 数字滤波器设计时，采用双线性变换法和频率的预畸变处理。将模拟滤波器离散化转换为数字滤波器，同时保证模拟滤波器和数字滤波器在 ω_n 或 ω_1，ω_2 处具有相同的幅频响应。

设计时应注意模拟频率和 MATLAB 数字频率(标准化频率)的转换。

【例 4.7】设计一个 Butteworth 高通数字滤波器，通带边界频率为 300Hz，阻带边界频率 200Hz，通带波纹小于 1dB，阻带衰减大于 20dB，采样频率为 1000Hz。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 4-7
%Design a Butterworth digital lowpass filter
%Desired performents of the filter
Fs=1000;
wp=300 * 2/Fs;
ws=200 * 2/Fs;
Rp=1;
Rs=15;
Nn=128;
%Compute Oder and Cutoff frequency
[N,Wn]=buttord(wp,ws,Rp,Rs)
%Design digital filter conversion.
[b,a]=butter(N,Wn,'high');
%Output
freqz(b,a,Nn,Fs)
```

```
≫ smp407
N =
     4
Wn =
    0.5344
```

所设计的 Butterworth 高通数字滤波器的频率特性如图 4.4 所示。

二、切比雪夫数字滤波器

切比雪夫滤波器分为切比雪夫 I 型和切比雪夫 I 型两种，MATLAB 信号处理工具箱均有相应的设计工具函数。

函数 CHEBY1 也可用于 Chebyshev I 型数字滤波器设计，调用格式为

$$[b, a]= \text{cheby1}(n, R_p, \omega_n)$$

$$[b, a]= \text{cheby1}(n, R_p, \omega_n, \text{'ftype'})$$

$$[z, p, k]= \text{cheby1}(\cdots)$$

$$[A, B, C, D]= \text{cheby1}(\cdots)$$

式中，R_p 为通带波纹，dB；ω_n 为截止频率，在 0～1 之间，在该频率处滤波器的幅值响应为 $-R_p$；其余参数同函数 butter。

【例 4.8】 设计一个带通切比雪夫 I 型数字滤波器，通带为 100Hz～200Hz，过渡带宽均为 50Hz，通带波纹小于 1dB，阻带衰减 30Hz，采样频率 $F_s=1000$Hz。

用 MATLAB 编程如下：

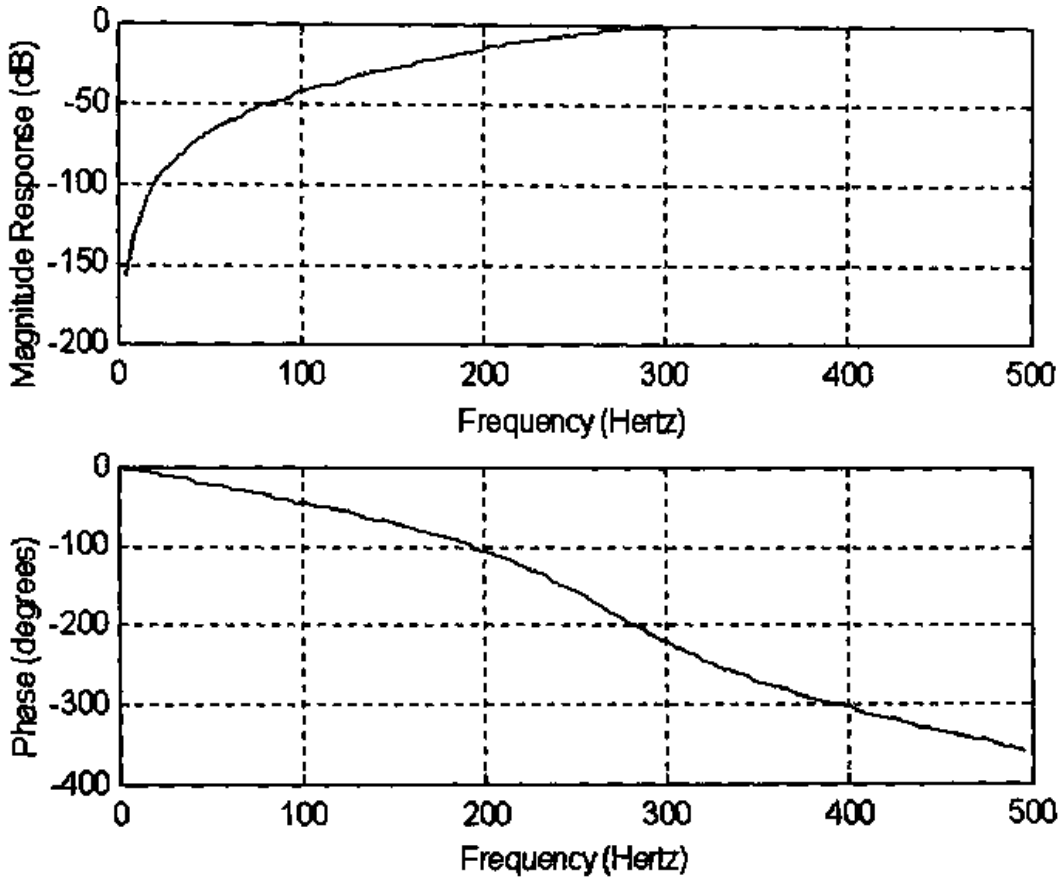


图 4.4 Butterworth 高通数字滤波器频率特性

```
%MATLAB PROGRAM 4-8
%Design a Chebyshev I type digital bandpass filter
%Desired performents of the filter
Fs=1000;
wp=[100 200]*2/Fs;
ws=[50 250]*2/Fs;
Rp=1;
Rs=30;
Nn=128;
%Compute Oder and Cutoff frequency
[N,Wn]=cheb1ord(wp,ws,Rp,Rs)
%Design digital filter conversion.
[b,a]=cheby1(N,Rp,Wn);
%Output
freqz(b,a,Nn,Fs)
≫ smp408
N =
    4
```

Wn =

0.2000　　0.4000

所设计的 IIR 滤波器的频率特性如图 4.5 所示。

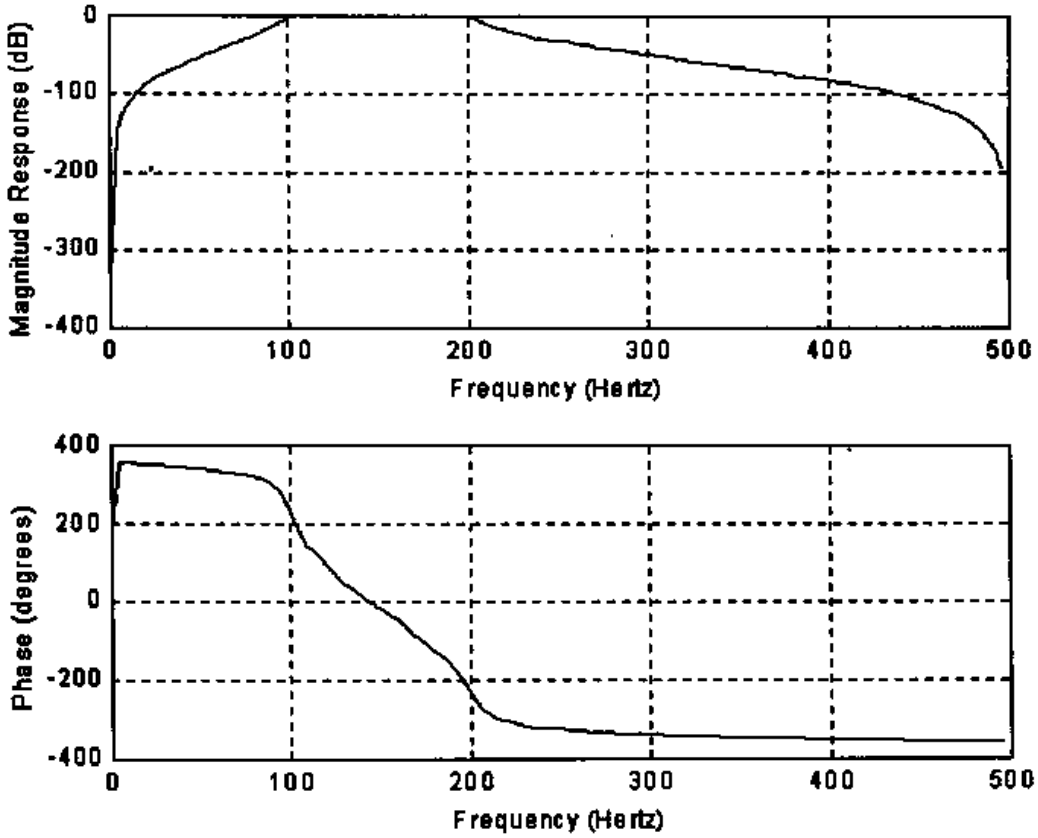


图 4.5　Chebyshev I 型带通数字滤波器频率特性

函数 CHEBY2 也可用于 Chebyshev I 型数字滤波器设计，调用格式为

[b, a]= cheby2(n, R_s, ω_n)

[b, a]= cheby2 (n, R_s, ω_n, 'ftype')

[z, p, k]= cheby2 (…)

[A, B, C, D]= cheby2 (…)

式中，R_s 为阻带衰减，单位：dB；ω_n 为截止频率，即阻带开始频率，在该频率下，滤波器的幅值响应为 $-R_s$，取值范围 0～1。其余各项意义与函数 butter 相同。

【例 4.9】 设计一个带通切比雪夫 I 型滤波器，设计参数同例 4.8。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 4-9
%Design a Chebyshev II type digital bandpass filter
%Desired performents of the filter
Fs=1000;
wp=[100 200]*2/Fs;
ws=[50 250]*2/Fs;
Rp=1;
Rs=30;
```

```
Nn=128;
%Compute Oder and Cutoff frequency
[N,Wn]=cheb2ord(wp,ws,Rp,Rs)
%Design digital filter conversion.
[b,a]=cheby2(N,Rs,Wn);
%Output
freqz(b,a,Nn,Fs)

≫ smp409

N =
     4
Wn =
    0.1515    0.4913
```

所设计的 IIR 滤波器的频率特性如图 4.6 所示。

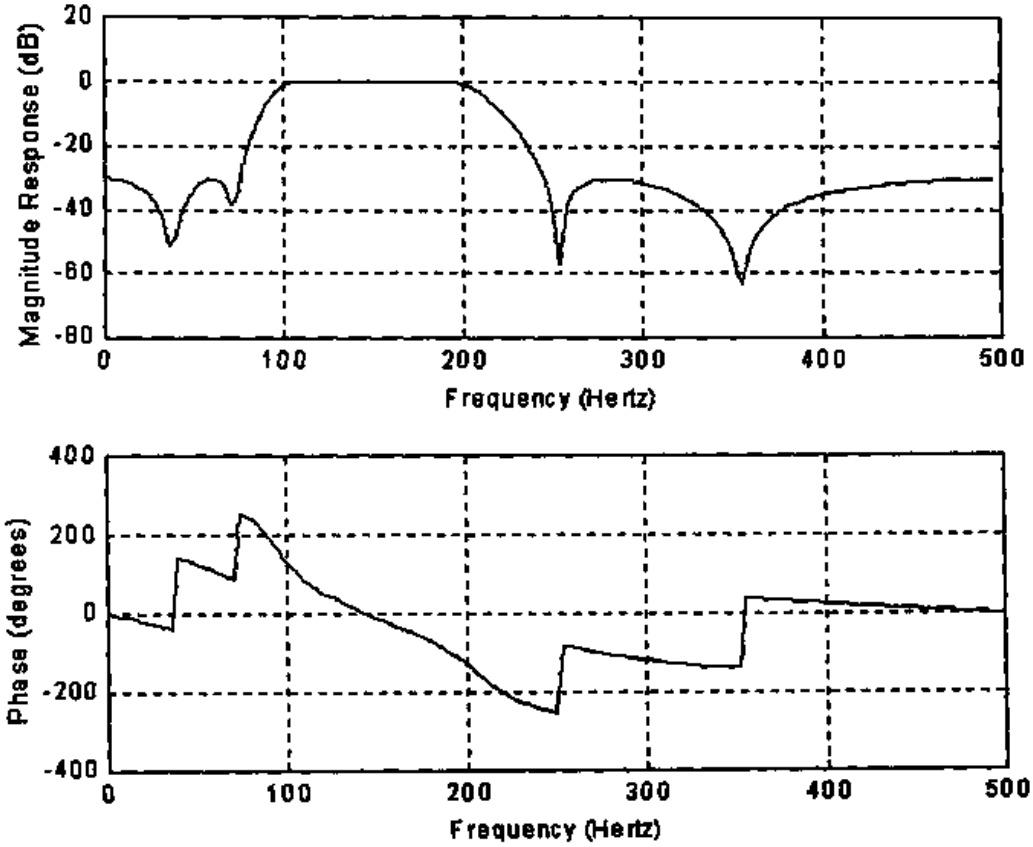


图 4.6 chebyshev Ⅰ型带通数字滤波器频率特性

三、椭圆数字滤波器

函数 ELLIP 也可用于椭圆数字滤波器设计，调用格式

[b, a]= ellip (n, R_p, R_s, ω_n)

[b, a]= ellip (n, R_p, R_s, ω_n, 'ftype')

[z, p, k]= ellip (…)

[A, B, C, D]= ellip (…)

其中，R_p 为通带波纹，单位：dB；R_s 为阻带衰减，单位：dB；其余各项意义同函数 butter。

【例 4.10】 设计一个带通椭圆数字滤波器，通带频率从 100Hz～200Hz，通带波纹小于 3dB，阻带衰减为 30dB，两边过渡带宽 50Hz，采样频率 1000Hz。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 4-10
%Design a Elliptic digital bandpass filter
%Desired performents of the filter
Fs=1000;
wp=[100 200]*2/Fs;
ws=[50 250]*2/Fs;
Rp=3;
Rs=30;
Nn=516;
%Compute Oder and Cutoff frequency
[N,Wn]=ellipord(wp,ws,Rp,Rs)
%Design digital filter conversion.
[b,a]=ellip(N,Rp,Rs,Wn);
%Output
freqz(b,a,Nn,Fs)
```

```
≫ smp410
N =
     3
Wn =
    0.2000    0.4000
```

所设计的 IIR 滤波器频率特性如图 4.7 所示。

以上设计程序中，均调用确定滤波器的最小阶数函数 buttord、cheb1ord、cheb2ord 和 ellipord。利用这些工具函数不仅可确定满足设计指标的滤波器最小阶数，而且可根据性能指标精确计算截止频率 ω_n。若设计给定阶数滤波器，且对截止频率精度无特殊要求，则可直接调用滤波器完全设计函数 butter、cheby1、cheby2 和 ellip。

4.4 IIR 滤波器直接设计

IIR 数字滤波器的经典设计法只限于几种标准的低通、高通、带通、带阻滤波器，而对于具有任意形状或多频带滤波器的设计是无能为力的。

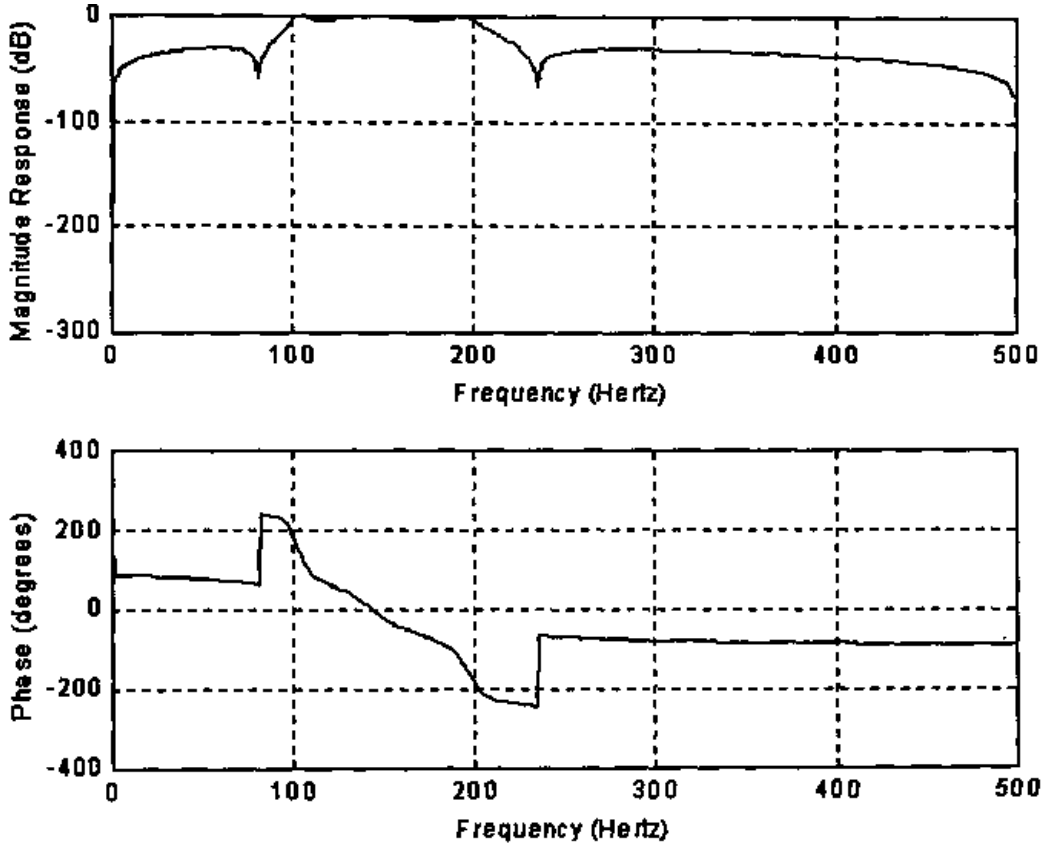


图 4.7 椭圆带通数字滤波器频率特性

如果所设计 IIR 滤波器幅频特性比较复杂，可采用最小二乘法拟合给定的幅频响应，使设计的滤波器幅频特性逼近期望的频率特性，这种方法称为 IIR 滤波器的直接设计法。

MATLAB 信号处理工具箱函数 YULEWALK 采用直接法设计 IIR 数字滤波器。函数调用格式为

$$[b, a]=\text{yulewalk}\ (n, f, m)$$

其中，n 为滤波器阶数；f 为给定的频率点向量，标准化频率取值范围 0～1，f 的第 1 个频率点必须是 0，最后一个频率点必须是 1，f 向量的频率点必须是递增的；m 为和频率向量 f 对应的理想幅值响应向量，m 和 f 必须是相同维数向量；b，a 分别是所设计的滤波器分子和分母多项式系数向量。IIR 滤波器的传递函数具有下面形式

$$H(z)=\frac{B(z)}{A(z)}=\frac{b(1)+b_2z^{-1}+\cdots+b(n+1)z^{-n}}{a(1)+a(2)z^{-2}+\cdots+a(n+1)z^{-n}}$$

在定义频率响应时，应避免通带至阻带的过渡段形状过分尖锐。为了获得一个最好的设计，常常需要调整过渡带的斜率。

函数 YULEWALK 首先计算给定幅频响应傅里叶逆变换和相关系数，再采用修正的 Yule-Walker 方程计算滤波器传递函数分母多项式系数。

函数 YULEWALK 采取下面步骤计算分子多项式：

(1) 计算与分子多项式相应的幅值平方响应的辅助式；

(2) 由辅助分子式和分母多项式计算完全的频率响应；

(3) 计算滤波器的脉冲响应；

(4) 采用最小二乘法拟合脉冲响应最终求得滤波器的分子多项式系数。

函数 YULEWALK 允许用户自由定义滤波器的频率向量 f 和幅值向量 m，因此该函数可设出具有任意形状的幅频响应的滤波器，包括多频带滤器。应该注意的是该函数不能设计给定相位指标的滤波器。

【例 4.11】 用直接法设计一个多频带数字滤波器，幅频响应值如下：

```
f=[0  0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1.0]
m=[0  0  1  1  0  0  1  1  0  0  0]
```

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 4-11
%Direct IIR digital tilter design
Oder=10;
f=0:0.1:1;
m=[0 0 1 1 0 0 1 1 1 0 0];
[b,a]=yulewalk(Oder,f,m);
[h,w]=freqz(b,a,128);
axes('position',[0.2 0.2 0.4 0.4]);
plot(f,m,'b-',w/pi,abs(h),'m--')
xlabel('Frequency(pi)');
ylabel('Magnitude');
title('Direct IIR design-Yuliwalk')
legend('Desired','Actual',1)
grid
```

运行结果，绘制多带滤波器幅频响应的期望曲线和实际曲线如图 4.8 所示。

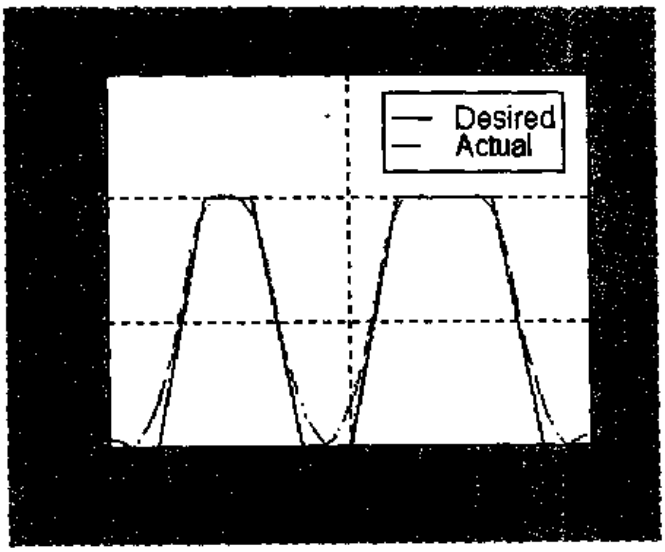


图 4.8 直接法设计的多频带滤波器

4.5 最大平滑 IIR 滤波器设计

最大平滑数字滤波器也称通用巴特沃斯数字滤波器。

在 IIR 数字滤波器经典设计法中谈到巴特沃斯数字滤波器的设计问题。用经典法设计的巴特沃斯滤波器的系统函数分子阶数和分母阶数相同。所谓最大平滑 IIR 滤波器是指所设计的巴特沃斯低通数字滤波器分母和分子阶数不但可以不同，而且分子阶数也可以高于分母，是巴特沃斯低通数字滤波器的更一般化形式。

这种滤波器的频率特性更加平滑。在滤波器实现方面，若零点的实现比极点容易的话，采用通用形式的巴特沃斯滤波器会更加经济。MATLAB 信号处理工具箱函数 MAXFLAT 用于最大平滑数字滤波器——通用巴特沃斯数字低滤波器设计，函数调用格式有多种。

$$[b, a] = \text{maxflat}(nb, na, \omega_n)$$

其中，nb 和 na 分别为滤波器分子和分母多项式阶数；ω_n 为滤波器－3dB 截止频率，0～1。

$$b = \text{maxflat}(nb, 'sym', \omega_n)$$

其中，nb 为滤波器分子多项式阶数；'sym'表示对称型 FIR 巴特沃斯滤波器，ω_n 为滤波器－3dB 截止频率，0～1。该调用格式用于设计对称型 FIR 巴特沃斯滤波器。

$$[b, a, b1, b2] = \text{maxflat}(nb, na, \omega_n)$$

其中，b 为滤波器分子多项式系数向量；b1 为多项式系数向量，包含全部 $z=-1$ 的零点；b2 为多项式系数向量，包含除－1 外其余全部零点；$b = b1 * b2$.

$$[b, a, b1, b2] = \text{maxflat}(nb, na, \omega_n, 'design\text{-}flag')$$

其中，'design-flag'为监测设计过程标志：

- trace　表格显示设计过程；
- plot　绘制滤波器的幅频图、群延迟和零极点图；
- 执行上面两种监测方式。

【例 4.12】 设计通用巴特沃斯低通数字滤波器，其系统函数分子阶数为 10，分母阶数为 2，截止频率为 0.6π。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 4-12
%Maximally flat digital filter design
% Generalized Lowpass Butterworth
nb=6;
na=3;
wn=0.6;
[b,a]=maxflat(nb,na,wn,'plot')
```

```
≫ smp412
b =
    0.1743    0.6709    0.9423    0.5465    0.0800   -0.0189    0.0019
```

a =

1.0000　　0.7049　　0.5713　　0.1206

滤波器的幅频特性图、零极点图及群延迟图如图 4.9 所示。

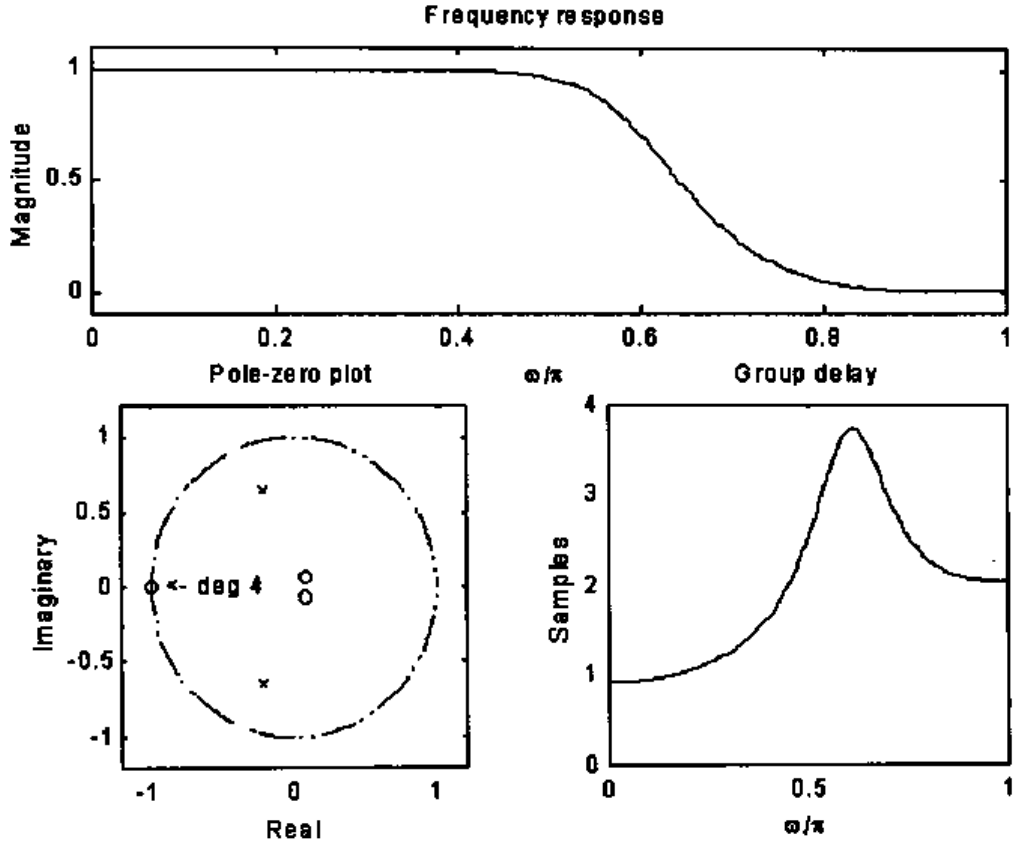


图 4.9　最大平滑巴特沃斯低通数字滤波器特性

4.6　FIR 数字滤波器的线性相位特性和设计方法

前面讨论的 IIR 滤波器设计只能保证其幅频响应满足性能指标，相位特性无法考虑且往往是非线性的。FIR 滤波器的突出优点是，在保证满足滤波器幅频响应要求的同时，还可以获得严格线性相位特性，这对于高保真的信号处理，如语音处理、数据处理和测试等是十分重要的。和 IIR 滤波器相比，FIR 滤波器的主要缺点是，达到相同性能指标所需滤波器阶数要高得多，延迟也要大得多。

FIR 数字滤波器的系统函数如式(4.1-6)所示，即

$$H(z)=\frac{Y(z)}{X(z)}=\sum_{n=0}^{N-1}b(n)z^{-n} \tag{4.6-1}$$

由此式可得系统的差分方程

$$\begin{aligned}y(n)&=b(0)x(n)+b(1)x(n-1)+\cdots+b(N-1)x[n-(N-1)]\\&=\sum_{m=0}^{N-1}b(m)x(n-m)\\&=b(n)*x(n)\end{aligned} \tag{4.6-2}$$

由式(4.6-2)，FIR 数字滤波器又称为卷积滤波器。FIR 滤波器的频率响应表达式为

$$H(e^{jw}) = \sum_{n=0}^{N-1} b(n)e^{-jnw} \tag{4.6-3}$$

信号通过FIR数字滤波器不失真条件如式(4.1-7)所示。即滤波器在通带内具有恒定的幅频响应和线性相位特性，可以证明，对于FIR数字滤波器，在保证在逼近平直幅频特性同时，还能获得严格的线性相位特性，其条件是使FIR滤波器的系数满足下列中心对称条件，即

$$b(n) = b(N-1-n)$$

或

$$b(n) = -b(N-1-n) \tag{4.6-4}$$

若FIR数字滤波器的单位脉冲响应序列为h(n)，由式(4.6-1)可见，它就滤波器系数向量b(n)。线性相位FIR滤波器的脉冲响应也满足中心对称条件。

根据中心对称性和滤波器阶数N的奇偶性，FIR线性相位滤波器具有内在的频率响应限制，如表4.2所示。

表4.2

线性相位滤波器类型	滤波器阶数N	系数对称性	频率响应H(ω) ω=0	频率响应H(ω) ω=1(Nyquist频率)
Ⅰ型	偶数	偶数：b(k)=b(N+2-k)，k=1,2,…,N+1	无限制	无限制
Ⅱ型	奇数		无限制	H(1)=0
Ⅲ型	偶数	奇数：b(k)=-b(N+2-k)，k=1,2,…,N+1	H(0)=0	H(1)=0
Ⅳ型	奇数		H(0)=0	无限制

线性相位FIR滤波器的相位滞后和群延迟在整个频带上是相等且不变的。对于一个N阶线性相位FIR滤波器，群延迟是N/2，滤波后的信号简单地延迟N/2时间步长，这一特性使通带频率内信号通过滤波器后仍保持原有波形而无相位失真。

MATLAB信号处理工具箱提供的FIR数字滤波器设计函数用于设计表4.2中四类具有线性相位滤波器。采用的设计方法和主要工具函数如表4.3所示。

表4.3

设计方法	说明	工具函数
窗设计法	理想滤波器加窗处理	fir1(单带) fir2(多带) Kaiserord
最优化设计	最小平方误差最小化逼近理想幅频响应或Park-McClellan算法产生等波纹滤波器	firls remez remezord
约束的最小二乘逼近	在满足最大误差限制条件下使整个频带最小平方误差最小化	fircls fircls1
任意响应设计	具有任意响应的FIR滤波器，包括复响应和非线性相位	cremez
升余弦函数	具有光滑、正弦过渡带的低通滤波器设计	firrcos

下面介绍FIR数字滤波器设计方法工具函数调用格式及应用。

4.7 FIR 滤波器的窗函数设计

4.7.1 窗函数法基本原理

FIR 滤波器设计的主要任务是根据给定的性能指标确定滤波器的系数 b，即系统单位脉冲序列 h(n)，它是一个有限长序列。

假设 $H_d(e^{j\omega})$是所要求的理想频率响应，则

$$H_d(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h_d(n)e^{-j\omega n} \tag{4.7-1}$$

式中，$h_d(n)$是对应的单位脉冲响应序列，而滤波器的频率响应和单位脉冲响应序列是傅里叶变换对，则

$$h_d(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega})e^{j\omega n}d\omega \tag{4.7-2}$$

求得序列 $h_d(n)$后，可得到 $H_d(z)$

$$H_d(z) = \sum_{n=-\infty}^{\infty} h_d(n)z^{-n} \tag{4.7-3}$$

注意到，这里 $h_d(n)$为无限长序列，因此 $H_d(z)$是物理不可实现的。

为了使系统变为物理可实现的，且使 FIR 滤波器实际频率响应尽可能逼近理想的频率响应，采用窗函数将无限脉冲响应 $h_d(n)$截取一段 h(n)来近似表示 $h_d(n)$，可得

$$h(n) = h_d(n)W(n) \tag{4.7-4}$$

由此可得

$$H(z) = \sum_{n=0}^{N-1} h(n)z^{-n} \tag{4.7-5}$$

式中，N 为窗口宽度，H(z)是物理可实现系统。

为了获得线性相位 FIR 数字滤波器 h(n)又必须满足中心对称条件，序列 h(n)应有延迟 α，且

$$\alpha = \frac{N-1}{2} \tag{4.7-6}$$

这种方法的基本原理是用一定宽度窗函数截取无限脉冲响应序列获得有限长的脉冲响应序列，从而得到 FIR 滤波器，故称为 FIR 滤波器的窗函数设计法。

经过加窗处理后所得滤波器实际的频率响应 $H(e^{j\omega})$能否很好逼近理想频率响应 $H_d(e^{j\omega})$呢？

图 4.10 给出了一个例子。理想低通滤波器幅频响应如图(a)所示，一个矩形窗的幅频响应如图(b)所示。由式(4.7-4)和复卷积定理可知，FIR 滤波器频率响应 $H(e^{j\omega})$是由 $H_d(e^{j\omega})$和窗函数频响 $W(e^{j\omega})$的圆周卷积获得，即

$$H(e^{j\omega}) = H_d(e^{j\omega})\circledast W(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} W(e^{j\theta})H_d(e^{j(\omega-\theta)})d\theta \tag{4.7-7}$$

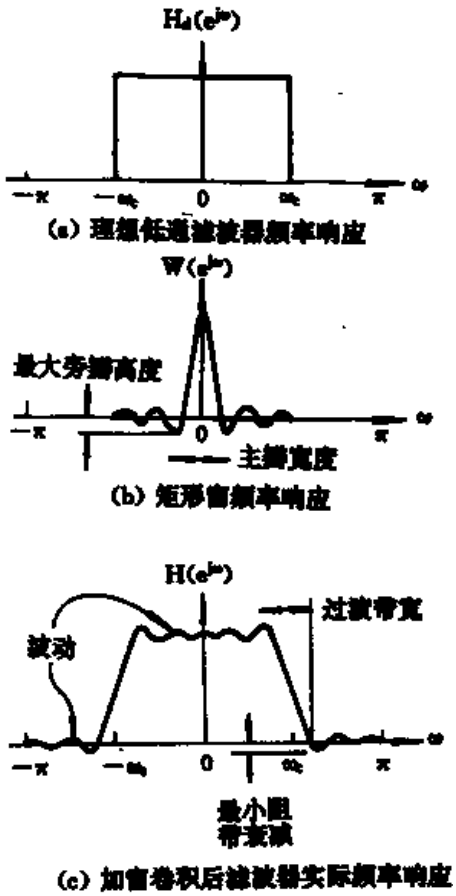


图 4.10 FIR 滤波器理想和实际频率响应

图(c)表示卷积运算后，实际幅频特性 $H(e^{j\omega})$ 曲线。

由图 4.10 可清楚看出，加窗后使实际频响偏离理想频响，主要影响有三个方面：

(1) 理想幅频特性陡直边沿处形成过渡带，过渡带宽度取决于窗函数频响应主瓣宽度；

(2) 过渡带两侧形成肩峰和波纹，这是窗函数频响的旁瓣引起的，旁瓣相对值愈大，肩峰愈大，旁瓣愈多，波纹愈多；

(3) 增加窗函数宽度 N，窗函数频响的主瓣宽度减小，但不改变旁瓣的相对值。

关于窗函数的特性将在第六章详细讨论。

为了改善 FIR 滤波器性能，要求窗函数的主瓣宽度尽可能窄，以获得较窄的过渡带；旁瓣相对值尽可能小，数量尽可能少，以获得通带波纹小、平稳而阻带衰减大，使滤波器实际频响 $H(e^{j\omega})$ 更好地逼近理想频响 $H_d(e^{j\omega})$。

为此，用窗函数法设计 FIR 数字滤波器时，要根据给定的滤波器性能指标选择窗口宽度 N 和窗函数 w(n)。各种窗函数性能如表 4.4 所示，供设计时参考。

表 4.4

窗函数	主瓣宽	第一旁瓣相对主瓣衰减(dB)	所构成低通滤波器阻带最小衰减(dB)
矩形窗	$\frac{4\pi}{N}$	−13	21
汉宁窗	$\frac{8\pi}{N}$	−31	44
哈明窗	$\frac{8\pi}{N}$	−41	53
巴特洛特窗	$\frac{8\pi}{N}$	−25	25
勃莱克曼窗	$\frac{12\pi}{N}$	−57	74
三角窗	$\frac{8\pi}{N}$	−25	25
凯塞窗	可调整	可调整	可调整
切比雪夫窗	可调整	可调整	可调整

基于窗函数的 FIR 数字滤波器设计的算法十分简单，其主要步骤为：

(1) 由式(4.7-2)对滤波器理想特性 $H_d(e^{j\omega})$ 进行傅里叶逆变换获得理想滤波器的单位脉冲响应 $h_d(n)$。一般假设理想低通滤波器的截止频率为 ω_c，其幅频特性满足

$$|H_d(e^{j\omega})| = \begin{cases} 1, & 0 \leqslant \omega \leqslant \omega_0 \\ 0, & \omega_0 < \omega < \pi \end{cases}$$

或式(4.7-2)可写为

$$h_d(n)=\frac{1}{2\pi}\int_{-\omega_c}^{\omega_c}e^{j\omega n}d\omega=\frac{\sin[\omega_c(n-\alpha)]}{\pi(n-\alpha)} \tag{4.7-8}$$

(2) 由性能指标确定窗函数 W(n)和窗口长度 N，由过渡带宽近似于窗函数主瓣宽求得窗口长度 N。

(3) 由式(4.7-4)求得实际滤波器的单位脉冲响应 h(n)

$$h(n)=h_d(n)\cdot W(n)$$

h(n)即为所设计 FIR 滤波器系数向量 b(n)。

(4) 检验滤波器性能。

【例 4.13】 用窗函数法设计一个线性相位 FIR 低通滤波器，并满足性能指标：通带边界频率 $\omega_p=0.5\pi$，阻带边界频率 $\omega_s=0.66\pi$，阻带衰减不小于 40dB，通带波纹不大于 3dB。

由题意，阻带衰减不小于 40dB，选取汉宁窗。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 4-13
%Window-based FIR filter design
%Perfoment parameter
wp=0.5*pi;
ws=0.66*pi;
%Width of transition band
wdelta=ws-wp;
%Length of the filter
N=ceil(8*pi/wdelta)
if rem(N,2)==0
    N=N+1;
end
%Length of the window
Nw=N;
%Cutoff Frequency of the filter
wc=(wp+ws)/2;
%Compute impulse response of ideal filter
n=0:N-1;
alpha=(N-1)/2;
m=n-alpha+0.00001;
hd=sin(wc*m)./(pi*m);
%Compute time response of the Hanning Window
win=hanning(Nw);
%Compute acture impulse response of the filter
h=hd.*win';
```

```
b=h;
freqz(b,1,512)
```

程序运行即得所设计 FIR 线性相位滤波器频率特性。如图 4.11 所示。

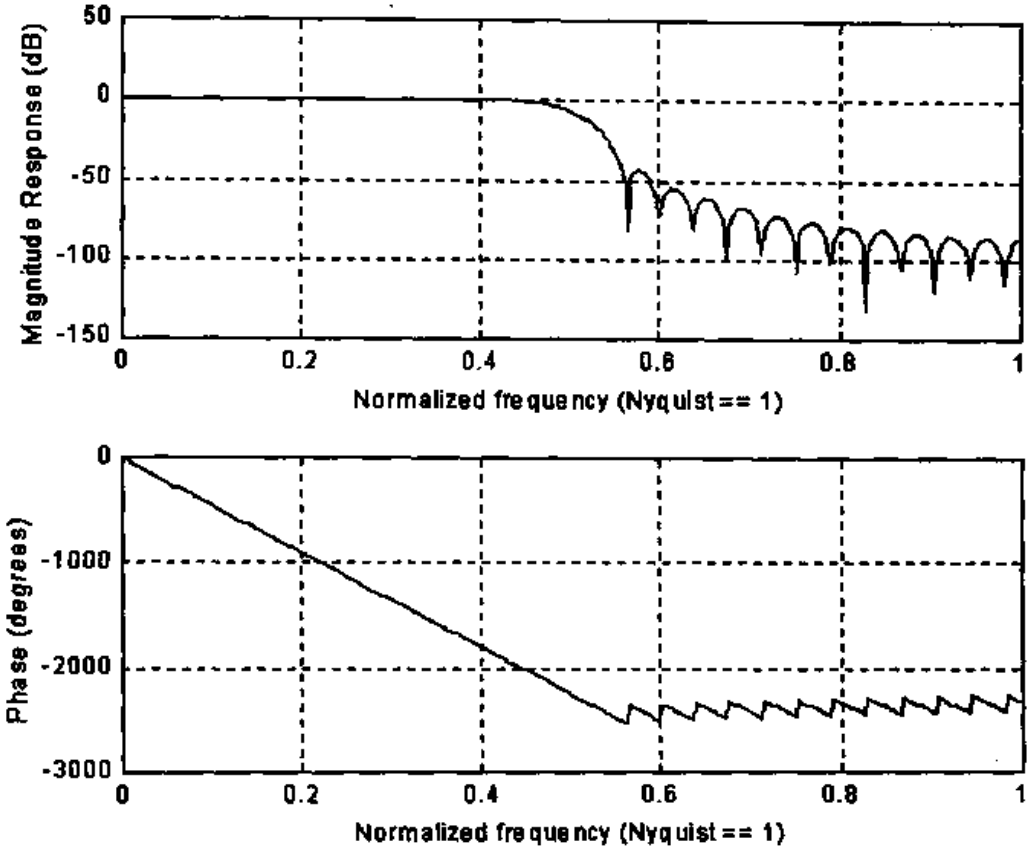


图 4.11 FIR 滤波器频率响应

4.7.2 标准型 FIR 滤波器

MATLAB 信号处理工具箱提供基于上述原理设计标准型 FIR 滤波器的工具函数。

函数 FIR1 是采用经典窗函数法设计线性相位 FIR 数字滤波器，且具有标准低通、带通、高通和带阻等类型。函数调用格式为

```
b=fir1(n, ωn)
b=fir1(n, ωn, 'ftype')
b=fir1(n, ωn, window)
b=fir1(n, ωn, 'ftype', window)
```

其中，n 为 FIR 滤波器的阶数，对于高通、带阻滤波器 n 取偶数；ω_n 为滤波器截止频率，0～1；对于带通、带阻滤波器，$\omega_n=[\omega_1,\omega_2]$，且 $\omega_1<\omega_2$；对于多带滤波器 $\omega_n=[\omega_1,\omega_2,\omega_3,\omega_4]$，频带分段为 $0<\omega<\omega_1$，$\omega_1<\omega<\omega_2$，$\omega_2<\omega<\omega_3$，…

'ftype'为滤波器类型：

- 缺省时为低通或带通滤波器；
- 'high'为高通滤波器；
- 'stop'为带阻滤波器；
- 'DC—1'使多带的第一频带为通频带；

• 'DC—0'使多带的第一频带为阻频带。

window 为窗函数,列向量,其长度为 n+1。

缺省时,自动取 Hamming 窗。MATLAB 提供的窗函数有 boxcar, hanning, Hamming, Bartlett, Blackmann Kaiser, Chebwin,调用格式见第六章。

b 为 FIR 滤波器系数向量,长度(n+1)。FIR 滤波器具有下列形式:

$$b(z) = b(1) + b(2)z^{-1} + \cdots + b(n+1)z^{-n}$$

用函数 fir1 设计的 FIR 滤波器的群延迟为 n/2。

【例 4.14】 用窗函数设计一个线性相位 FIR 低通滤波器,技术性能指标同例 4.13。

这里采用 FIR 数字滤波器设计工具函数 fir1,并用 MATLAB 编程如下:

```
%MATLAB PROGRAM 4-14
%Window-based FIR filter design
%Perfoment parameter
wp=0.5*pi;
ws=0.66*pi;
%Width of transition band
wdelta=ws-wp;
%Length of the filter
N=ceil(8*pi/wdelta)
%Cutoff Frequency of the filter
Wn=(0.5+0.66)*pi/2;
%Using 'FIR1' Design the filter
b=fir1(N,Wn/pi,hanning(N+1));
freqz(b,1,512)
```

程序运行得所设计的 FIR 线性相位滤波器频率特性和图 4.11 完全相同。该程序比例 4.13 的程序要简单得多。函数 fir1 自动实现 FIR 滤波器的全部设计过程。

由图 4.11 可见,用函数 fir1 设计的 FIR 数字滤波器在通带内具有很好的线性相位特性。

【例 4.15】 设计一个 24 阶 FIR 带通滤波器,通带频率为 $0.35 \leqslant \omega \leqslant 0.65$。

利用基于窗函数 FIR 滤波器设计工具函数 fir1 编写程序如下:

```
%MATLAB PROGRAM 4-15
%Window-based FIR filter design
%Perfoment parameter
wp=[0.35 0.65];
N=24;
%Using 'FIR1' Design the filter
b=fir1(2*N,Wn);
freqz(b,1,512)
```

程序运行得所设计 FIR 带通滤滤器频率特性,如图 4.12 所示。

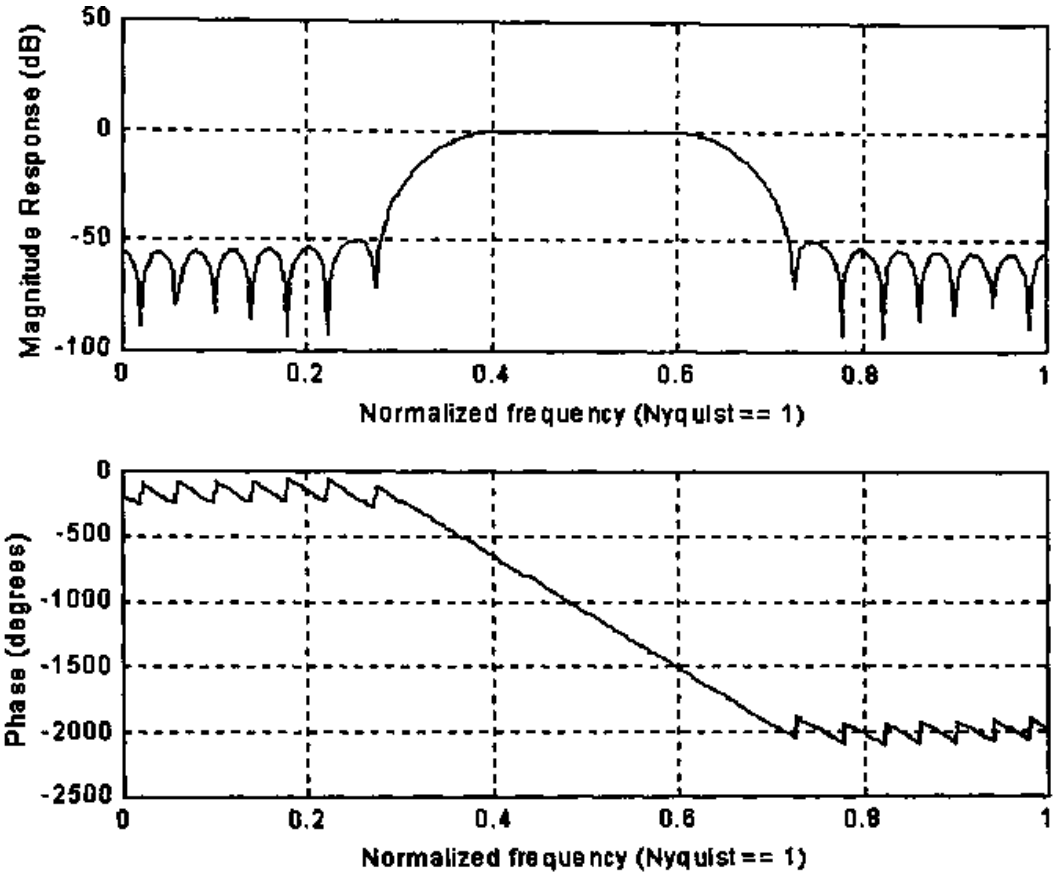


图 4.12 FIR 带通滤波器频率特性

4.7.3 多频带 FIR 滤波器

除了设计标准型的 FIR 滤波器外，MATLAB 信号处理工具箱还提供另一种基于窗函数滤波器设计的工具函数 FIR2，用于设计具有任意形状频率响应的 FIR 数字滤波器，其调用格式为

```
b=fir2 (n, f, m)
b=fir2 (n, f, m, window)
b=fir2 (n, f, m, npt)
b=fir2 (n, f, m, npt, window)
b=fir2 (n, f, m, npt, lap)
b=fir2 (n, f, m, npt, lap, window)
```

其中，n 为滤波器阶数；f 和 m 分别为滤波器期望幅频响应的频率向量和幅值向量，取值在 0～1 之间，m 与 f 相同长度；window 为窗函数，列向量，长度必须是(n+1)，缺省时自动取 Hamming 窗；npt 为对频率响应进行内插点数，缺省时为 512；lap 定义一个区域尺寸，函数 fir2 在重复频率点周围建立这个区域并提供光滑、陡峭的过渡频率响应，缺省值为 25；b 为 FIR 滤波器系数向量，长度为 n+1。滤波器具有下面形式：

$$b(z) = b(1) + b(2)z^{-1} + \cdots + b(n+1)z^{-n}$$

【例 4.16】 用窗函数法设计一个多频带的 FIR 滤波器，理想频率响应同例 4.11，滤波器阶数分别为 10 和 50，比较理想和实际滤波器的幅频响应。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 4-16
clf
%FIR filter design using the window method
%FIR with arbitrary filter shape.
%Desired frequency response
f=0:0.1:1;
m=[0 0 1 1 0 0 1 1 1 0 0];
%Design 10th Oder FIR filter
Oder=10;
b=fir2(Oder,f,m,hamming(Oder+1));
[h,w]=freqz(b,1,128);
subplot(221)
plot(f,m,w/pi,abs(h))
title('Oder=10')
xlabel('Normanized Frequency');
ylabel('Magnitude')
grid
%Design 50th Oder FIR filter
Oder=50;
b=fir2(Oder,f,m,hamming(Oder+1));
[h,w]=freqz(b,1,128);
subplot(222)
plot(f,m,w/pi,abs(h))
title('Oder=50')
xlabel('Normanized Frequency');
ylabel('Magnitude')
grid
```

由图 4.13 知，只有取 50 阶 FIR 滤波器时实际滤波器的幅频响应才逼近理想幅频响应。和例 4.11 的结果比较，相同性能的 FIR 滤波器阶数要比 IIR 滤波器高得多。

4.8 最优 FIR 滤波器设计

MATLAB 信号处理工具箱提供了比基于窗函数法的 FIR 滤波器设计工具函数 fir1 和 fir2 更为通用的函数 FIRLS 和 REMEZ。它们采用不同的优化方法设计最优的标准的和多频带 FIR 数字滤波器。

函数 firls 是函数 fir1 和 fir2 的扩展，其基本设计准则是利用最小二乘法使期望的频率响应和实际的频率响应之间的整体误差最小。

函数 remez 实现 Parks-McClellan 算法，这种算法利用 Remez 交换算法和 Chebyshev

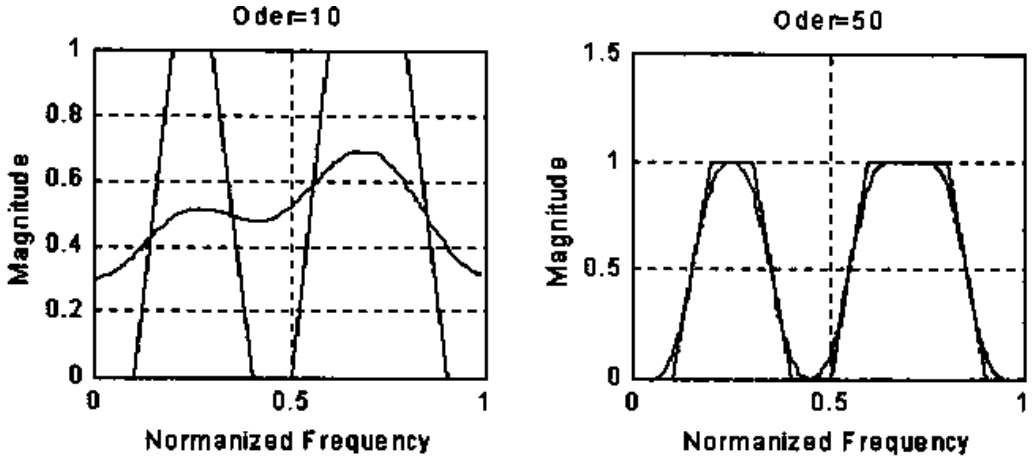


图 4.13　多频带 FIR 滤波器幅频响应

近似理论来设计滤波器，使实际频率响应拟合期望频率响应达到最优。从实际和理想频响之间最大误差最小化的观点来看，函数 remez 设计的滤波器是最优的，因此，有时称之为最小滤波器。在频域内，滤波器呈现等波纹特点，因此也称为等波纹滤波器。Parks-McClellan 滤波器设计方法是 FIR 滤波设计中最流行的和应用最广的。

函数 firls 和 remez 的调用语法规则相同的，不同的只要优化算法不同。

4.8.1　基本型式最优滤波器

函数 firls 和 remez 的基本格式用于设计 I 型和 II 型线性相位 FIR 滤波器。由表 4.2 可知，这是偶对称滤波器，I 型和 II 型的区别在于滤波器阶数是偶数还是奇数。

函数 firls 的基本调用格式为

$$b=firls(n, f, a)$$

其中，n 为滤波器阶数；f 为滤波器期望频率特性的频率向量标准化频率，0～1，递增向量，允许定义重复频率点；a 为滤波器期望频率特性的幅值向量，向量 a 和 f 必须是同长度且为偶数；b 为函数返回的滤波器系数，长度 n+1，且具有偶对称关系

$$b(k) = b(n + 2 - k)$$

函数 remez 的基本调用格式为

$$b = remez(n, f, a)$$

其中各项意义同以上函数 firls 所述。

下面例子的目的在于比较函数 firls 和 remez 设计的 FIR 滤波器。

【例 4.17】　分别用函数 firls 和函数 remez 设计一个 20 阶 FIR 低通滤波器，通带边界频率为 0.4π，幅值为 1，阻带边界频率为 0.5π，幅值为 0。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 4-17
%Multiband FIR filter design with transition band
%Desired freqency responce
clf
n=20;
```

```
f=[0 0.4 0.5 1];
a=[1 1 0 0];
%Design FIR filter by FIRLS
b=firls(n,f,a);
[h,w]=freqz(b);
%Design FIR filter by REMEZ
bb=remez(n,f,a);
[hh,w]=freqz(bb);
%Output
axes('position',[0.2 0.2 0.5 0.5]);
plot(w/pi,abs(h),'b-',w/pi,abs(hh),'r--');
xlabel('Normanized Frequency');
ylabel('Magnitude')
legend('firls','remez')
grid on
```

程序运行结果如图 4.14 所示，比较两种方法所设计的滤波器幅频响应可见，用函数 firls 设计的滤波器在整个频率范围内(包括通带和阻带)均具有较好的频响，但理想频响和实际频响的误差在带区内不均匀，且在边界频率处误差较大；而函数 remez 设计的滤波器在通带内，且有等波纹特性，在边界频率 0.4π 和 0.5π 处及过渡带内更接近于理想频响。

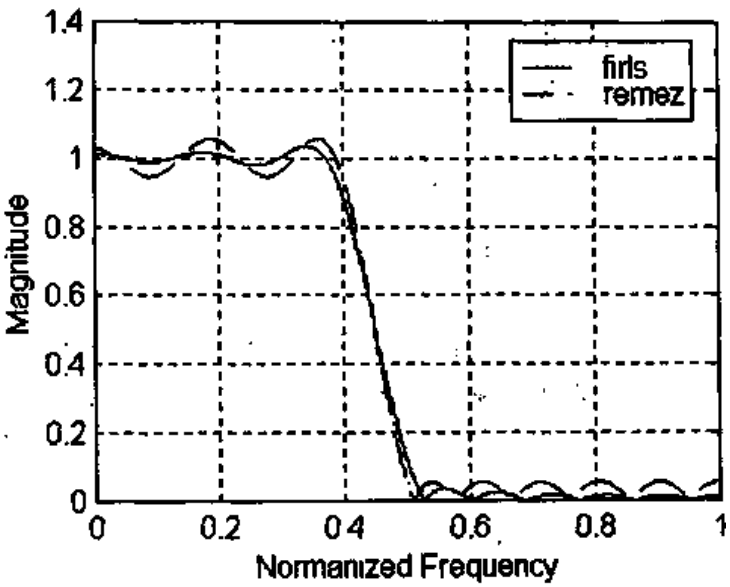


图 4.14 最优基本 FIR 滤波器设计

函数 firls 和 remez 可用于设计低通、高通、带通和带阻等一般类型的滤波器，这可由函数中给定的理想幅频响应的频率向量 f 和幅值向量 a 确定。

如设计一个带通滤波器，理想幅频响应向量对的给定形式如

```
f=[0   0.3   0.4   0.7   0.8   1];
a=[0   0     1     1     0     0];
```

这里，理想滤波器幅频响应定义为

- 阻带频率：0～0.3，0.8～1
- 通带频率：0.4～0.7
- 过渡带：0.3～0.4，0.7～0.8

设计一个高通滤波器，理想幅频响应向量对的给定形式如

f=[0 0.7 0.8 1];
a=[0 0 1 1];

这里,理想滤波器幅频响应定义为

- 阻带频率:0～0.7
- 通带频率:0.8～1.0
- 过渡带:0.7～0.8

设计一个带阻滤波器,理想幅频响应向量对的给出形式如

f=[0 0.3 0.4 0.5 0.8 1];
a=[1 1 0 0 1 1];

这里,理想滤波器幅频响应定义为

- 阻带频率:0.4～0.5
- 通带频率:0～0.3, 0.8～1.0
- 过渡带:0.3～0.4, 0.5～0.8

此外,函数 firls 和 remez 还可设计多频带滤波器。下面给出一个例子。

【例 4.18】 用函数 firls 设计一个 50 阶多通带滤波器,滤波器理想幅频响应对为

f=[0 0.1 0.15 0.25 0.3 0.4 0.45 0.55 0.6 0.7 0.75 0.85 0.9 1];
a=[1 1 0 0 1 1 0 0 1 1 0 0 1 1];

绘制理想和实际幅频图。

用 MATLAB 编写程序如下:

```
%MATLAB PROGRAM 4-18
%Multiband FIR filter design with transition band
%Desired freqency responce
clf
n=50;
f=[0 0.1 0.15 0.25 0.3 0.4 0.45 0.55 0.6 0.7 0.75 0.85 0.9 1];
a=[1 1 0 0 1 1 0 0 1 1 0 0 1 1 ];
%Design FIR filter by FIRLS
b=firls(n,f,a);
[h,w]=freqz(b);
%Design FIR filter by REMEZ
bb=remez(n,f,a);
[hh,w]=freqz(bb);
%Output
axes('position',[0.2 0.2 0.5 0.5]);
plot(w/pi,abs(h),'b.',w/pi,abs(hh),'r--',f,a,'m');
xlabel('Normanized Frequency');
ylabel('Magnitude')
legend('firls','remez','desired')
```

grid on

程序运行结果如图 4.15 所示，可见实际滤波器的频响在通带和阻带内均有波纹，减小波纹需增加滤波器的阶数。

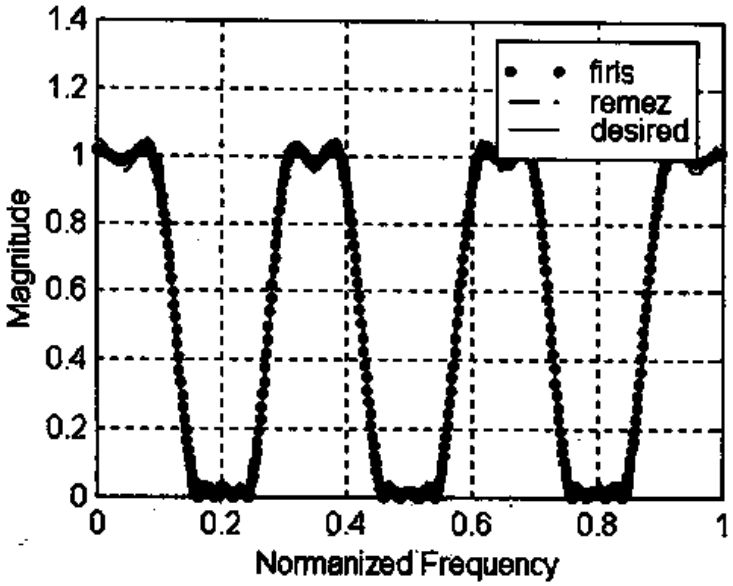


图 4.15 多频带 FIR 滤波器

函数 firls 和 remez 还能设计具有任意线性过渡带连接阻带和通带，使过渡带具有更广阔平滑的过渡区间。如理想幅频响应按如下频响对给出

```
f=[0  0.4  0.42  0.48  0.5  1];
a=[1  1  0.8  0.2  0  0];
```

这里，过渡带 0.4～0.6 之间给定了多个响应值设计出的具有线性过渡带 FIR 滤波器的频响，如图 4.16 所示。

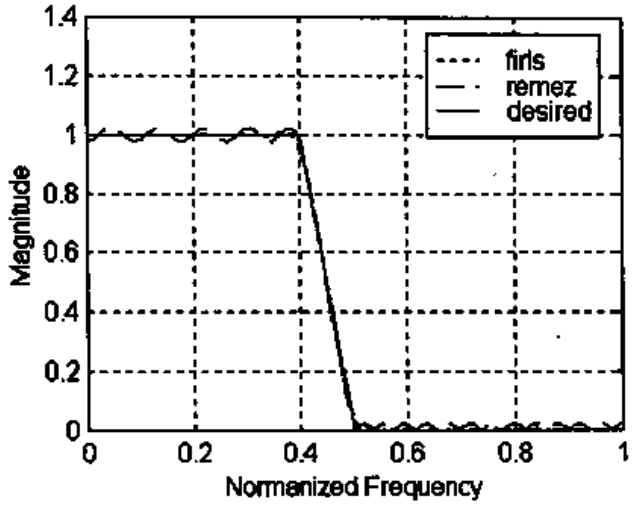


图 4.16 具有线性过渡带的 FIR 滤波器

4.8.2 加权最优滤波器

函数 firls 和 remez 输入参数设置权向量 w，在不同频段设置不同权值，使不同频段的误差最小化得到不同程度重视。具有权向量输入的函数 firls 和 remez 实现滤波器每个频

率段加权处理。

调用格式为：

```
b=firls (n, f, a, w)
b=remez (n, f, a, w)
```

其中，w 为权向量，为 f 和 a 向量长度的一半，一个频带必须对应一个权值。

如设计一个低通等波纹滤波器，通带边界频率 0.4，阻带边界频率 0.5，阻带波纹约为通带波纹的 1/10。

滤波器的理想幅频向量对及权向量应设置为

```
f = [0   0.4   0.5   1];
a = [1    1     0    0];
w = [1   10];
b = remez (n, f, a, w);
```

或

```
b = firls (n, f, a, w);
```

4.8.3 反对称 FIR 滤波器——赫尔伯特变换器

函数 firls 和 remez 可用于设计Ⅲ型和Ⅳ型线性相位滤波器。由表 4.2 可见，这类滤波器属于奇对称滤波器，Ⅲ型和Ⅳ型的区别在于滤波器阶数是偶数还是奇数。函数调用格式为

```
b=firls (n, f, a, 'h')
b=firls (n, f, a, w, 'h')
```

或

```
b=remez (n, f, a, 'h')
b=remez (n, f, a, w, 'h')
```

式中，'h'为选择项，表示设计的滤波器是奇对称(Ⅲ型和Ⅳ型)线性相位滤波器，滤波器可实现信号的赫尔伯特(Hilberst)变换，故又称赫尔伯特变换器。

【例 4.19】 利用函数 remez 和 firls 设计一个高通反对称线性相位滤波器，并绘制其频率特性图。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 4-19
%Multiband FIR filter design with transition band
%Desired freqency responce
clf
n=21;
f=[0 0.1 0.2 1];
a=[0 0     1     1];
%Design FIR filter by FIRLS
b=firls(n,f,a,'h');
[h,w]=freqz(b);
%Design FIR filter by REMEZ
```

```
bb=remez(n,f,a,'h');
[hh,w]=freqz(bb);
%Output
axes('position',[0.2 0.2 0.5 0.5]);
plot(w/pi,abs(h),'b',w/pi,abs(hh),'r--',f,a,'m:');
xlabel('Normanized Frequency');
ylabel('Magnitude')
legend('firls','remez','desired',4)
grid on
```

滤波器(赫尔伯特变换器)幅频特性如图 4.17 所示。

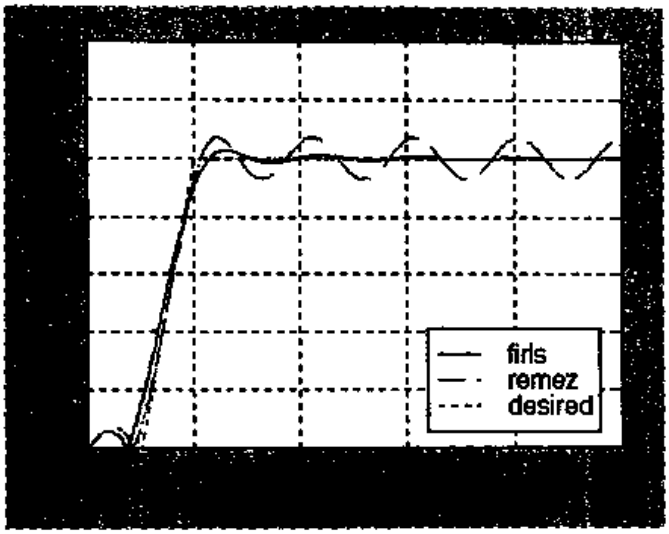


图 4.17 赫尔伯特变换器幅频图

4.8.4 微分 FIR 滤波器——微分器

对信号在时域的微分等价于信号的傅里叶变换和一个虚单位斜坡函数的乘积。也就是说,信号的微分相当于让该信号通过一个频响为 $H(\omega)=j\omega$ 的滤波器。函数 firls 和 remez 可用于设计这种具有微分作用的 FIR 滤波器,调用格式为

```
b=firls (n, f, a, 'd')
b=firls (n, f, a, w, 'd')
```

或

```
b=remez (n, f, a, 'd')
b=remez (n, f, a, w, 'd')
```

其中,'d'为选择项,表示设计的滤波器是Ⅱ型和Ⅳ型滤波器,具有微分器作用。

【例 4.20】 利用函数 remez 和 firls 设计一个 FIR 微分器,频率在 0~0.9 范围内,绘制其频率特性图。

用 MATLAB 编程如下:

```
%MATLAB PROGRAM 4-20
%Multiband FIR filter design with transition band
```

```
%Desired freqency responce
clf
n=30;
f=[0 0.9];
a=[0 0.9];
f1=[f 1];
a1=[a 0];
%Design FIR filter by FIRLS
b=firls(n,f,a,'d');
[h,w]=freqz(b);
%Design FIR filter by REMEZ
bb=remez(n,f,a,'d');
[hh,w]=freqz(bb);
%Output
axes('position',[0.2 0.2 0.5 0.5]);
plot(w/pi,abs(h),'b',w/pi,abs(hh),'r--',f1,a1,'m:');
xlabel('Normanized Frequency');
ylabel('Magnitude')
legend('firls','remez','desired',2)
grid on
```

FIR 微分器幅频特性如图 4.18 所示。

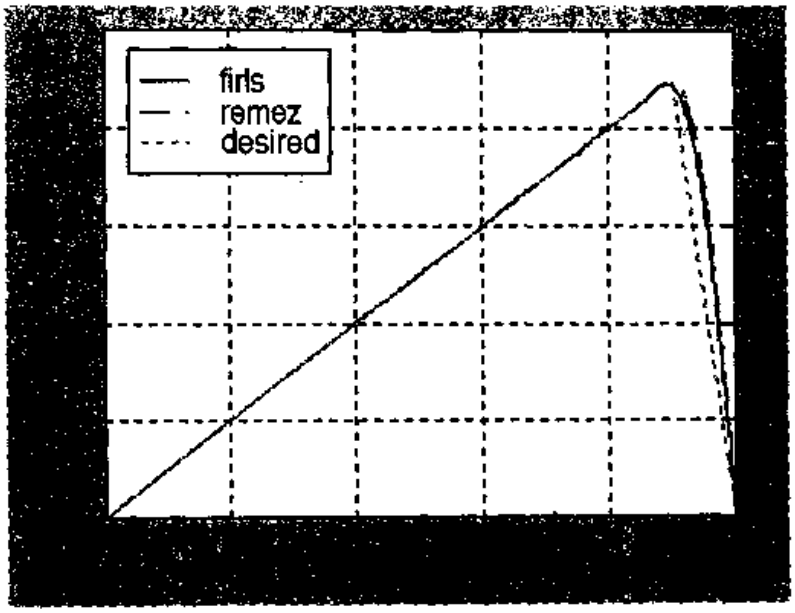


图 4.18 FIR 微分器幅频图

4.9 约束最小二乘 FIR 滤波器设计

FIR 滤波器的约束最小二乘法(CLS)的关键特点是在给定滤波器幅频响应最大允许

波纹的上下阈值约束条件下，使实际滤波器的幅频响应在整个频率范围内最小误差平方最小化。约束最小二乘法对幅频响应的过渡带没有明确定义，只需定义截止频率(对于高通、低通、带通或带阻等滤波器)或通带和阻带边界频率(对于多频带滤波器)作为期望频响。在给定约束的条件下，最小二乘法作用于期望幅频响应任何不连续区域。这种方法的附带作用使用户定义任意小的峰值减轻吉布斯(Gibbs)现象。

MATLAB 信号处理工具箱有两个函数实现这种设计技术，如表 4.5 所示。

表 4.5

函数	说明
fircls	约束最小二乘法多频带滤波器设计
fircls1	约束最小二乘法低通和高通线性相位滤波器设计

4.9.1 CLS 多频带滤波器

MATLAB 信号处理工具箱提供用约束最小二乘法(CLS)设计线性相位多频带、分段常数 FIR 滤波器工具函数 FIRCLS，调用格式为：

```
b=fircls (n, f, a, up, lo)
b=fircls (n, f, a, up, lo, 'flag')
```

其中，n 为 FIR 滤波器阶数；f 和 a 分别为给定的滤波器期望幅频特性的频率向量和幅值向量，f 为标准化频率，在 0～1 之间，且第一个值必须是 0，最后一个值必须是 1，a 的长度应为 length(f)－1；up 和 lo 分别为每个频带上边界和下边界频率，均为向量，且长度应为等于 a 的长度；b 为返回的 FIR 滤波器系数向量，长度为(n＋1)；'flag'为选择项，用于监视滤波器设计。

- trace　文字跟踪显示；
- plot　绘制滤波器幅频图、群延迟、零极点图；
- both　上述两种方式同时采用。

【例 4.21】　利用 CLS 法设计一个多频带滤波器且满足下面要求：

频率 0～0.3π；幅值为 0，允许变化范围：[－0.005，0.005]；

频率 0.3π～0.5π；幅值为 0.5，允许变化范围：[0.49，0.51]；

频率 0.5π～0.7π；幅值为 0，允许变化范围：[－0.03，0.03]；

频率 0.7π～0.9π；幅值为 1，允许变化范围：[0.98，1.02]；

频率 0.9π～1π；幅值为 0，允许变化范围：[－0.05，0.05]。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 4-21
%Linear-phase FIR filter design
%by constrained least-squares.
n=129;
f=[0 0.3 0.5 0.7 0.9 1];
a=[0 0.5 0 1 0];
f1=[0 0.3 0.3 0.5 0.5 0.7 0.7 0.9 0.9 1];
```

```
a1=[0 0  0.5 0.5 0  0  1  1  0  0];
up=[0.005 0.51 0.03 1.02 0.05];
lo=[-0.005 0.49 -0.03 0.98 -0.05];
b=fircls(n,f,a,up,lo);
[h,w]=freqz(b);
%Output
axes('position',[0.2 0.2 0.5 0.5]);
plot(w/pi,abs(h),'b')
hold on
plot(f1,a1,'m:');
xlabel('Normanized Frequency');
ylabel('Magnitude')
legend('fircls','desired',2)
grid on
```

设计的滤波器幅频特性如图 4.19 所示。

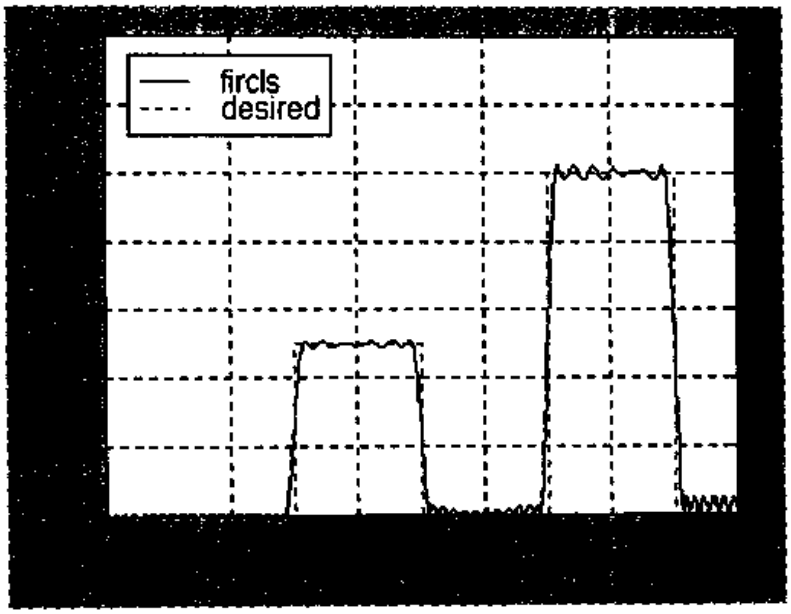


图 4.19 CLS 多频带滤波器幅频图

4.9.2 CLS 低通和高通滤波器

MATLAB 信号处理工具箱还提供用约束最小二乘法(CLS)设计基本的线性相位的低通和高通滤波器的工具函数 FIRCLS1。

设计低通滤波器的调用格式为：

$$b=\text{fircls1}(n,\ \omega_0,\ d_p,\ d_s)$$

$$b=\text{fircls1}(n,\ \omega_0,\ d_p,\ d_s,\ \omega_t)$$

$$b=\text{fircls1}(n,\ \omega_0,\ d_p,\ d_s,\ \omega_t,\ \text{'flag'})$$

其中，n 为滤波器阶数；ω_0 为滤波器截止频率，标准化频率，0～1；d_p 为通带离幅值 1 的最

大偏差；d_s 为阻带离幅值 0 的最大偏差；'flag'为设计监测标志，和函数 fircls 相同；b 为返回的滤波器系数向量，长度(n+1)；ω_t 为一个定义频率，确保设计的滤波器满足通带或阻带的边界要求。

- 若 $0<\omega_t<\omega_0<1$，则在 $0<\omega<\omega_t$ 范围内，滤波器幅值允差在 d_p 内
- 若 $0<\omega_0<\omega_t<1$，则在 $\omega_t<\omega<1$ 范围内，滤波器幅值允差在 d_s 内

设计高通滤波器调用格式为：

```
b=fircls1(n, ω0, dp, ds, 'high')
b=fircls1(n, ω0, dp, ds, ωt, 'high')
b=fircls1(n, ω0, dp, ds, …, 'high', 'flag')
```

其中，'high'表示设计高通滤波器；其余各项意义同上述。

【例 4.22】 用 CLS 法设计一个 25 阶低通滤波器，截止频率为 0.3π。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 4-22
%Basis linear-phase FIR lower filter design
%by Constrained Least-Squares.
n=55;
wo=0.3;
dp=0.02;
ds=0.008;
b=fircls1(n,wo,dp,ds,'plot');
```

滤波器实际幅频图及各频带幅值偏差如图 4.20 所示。可见，偏差均在给定范围之内。

4.9.3 加权 CLS 滤波器

采用 CLS 法设计低通和高通滤波器时，还可以对通带和阻带的误差最小化执行加权处理。方法是调用函数 FIRCLS1 时，定义这个权值为两带误差比例系数。函数调用格式为：

```
b=fircls1(n, ω0, dp, ds, ωp, ωs, K)
b=fircls1(n, ω0, dp, ds, ωp, ωs, K 'high')
b=fircls1(n, ω0, dp, ds, …, 'high', 'flag')
```

其中，ω_p 为最小二乘权函数通带边界频率 0～1；ω_s 为最小二乘权函数阻带边界频率 0～1；这里，对于低通滤波器，$\omega_p<\omega_0<\omega_t$，对于高通滤波器，$\omega_s<\omega_0<\omega_p$。K 为通带内误差平方和与阻带误差平方和之比。

$$K=\frac{\int_0^{\omega_p}|A(\omega)-D(\omega)|^2d\omega}{\int_0^{\omega_s}|A(\omega)-D(\omega)|^2d\omega}$$

其余各项同上述。

【例 4.23】 用加权 CLS 法设计一个 55 阶低通 FIR 滤波器，截止频率为 0.3π。通带允许最大波纹 0.02，阻带允许最大波纹 0.008。加权要求：权函数的通带边界为 0.28π；权

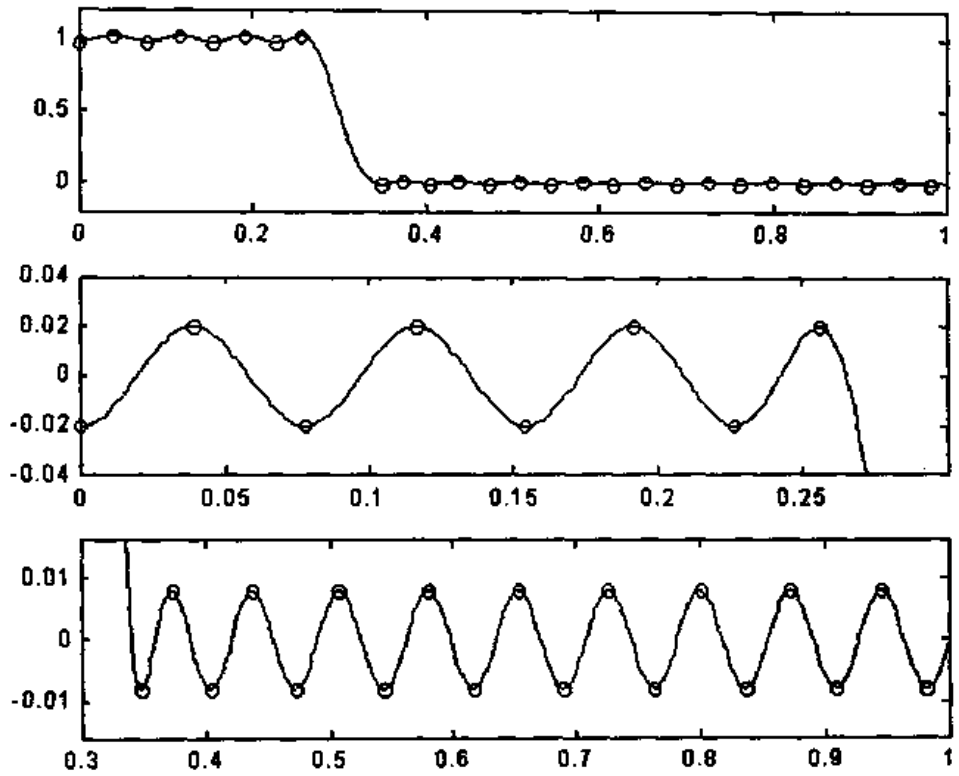


图 4.20 CLS 低通滤波器的幅频及偏差图

函数的阻带边界为 0.32π,阻带内最小误差平方和为通带的 1/10。

用 MATLAB 编写设计程序:

```
%MATLAB PROGRAM 4-23
%linear-phase FIR lower filter design
%by CLS with a weighted function
n=55;
wo=0.3;
dp=0.02;
ds=0.008;
wp=0.28;
ws=0.32;
k=10;
b=fircls1(n,wo,dp,ds,wp,ws,k,'plot');
```

滤波器实际幅频图及各频带幅值偏差如图 4.21 所示。和图 4.20 相比较,可见加权 CLS 滤波器的阻带频率特性更佳。

4.10 任意响应 FIR 滤波器设计

MATLAB 信号处理工具箱还有一个通用 FIR 滤波器设计工具函数 CREMEZ。和其他滤波器设计函数不同,该函数可设计任意复响应和非线性相位等波纹 FIR 滤波器,因此为用户提供十分通用和功能强大的滤波器设计工具。函数利用扩展的 Remez 交换算法

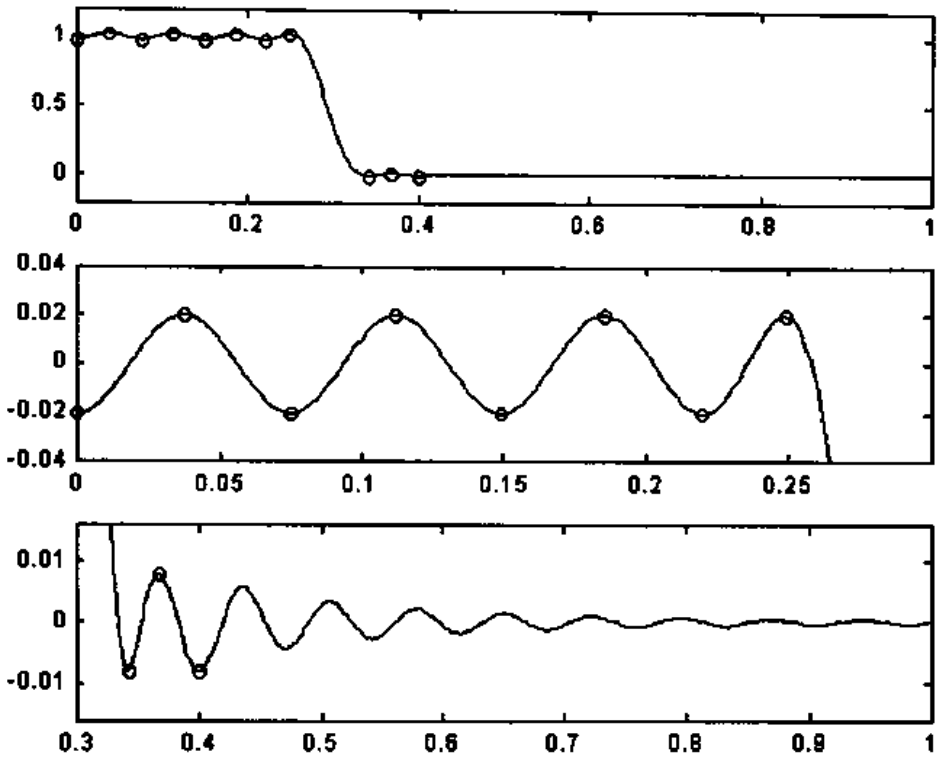


图 4.21　加权 CLS 滤波器幅频及偏差图

作为初始估计来优化 Chebyshev 误差；若这种方法失败，该算法转换为递增—递减算法使最优解收敛。

4.10.1　多频带复响应滤波器设计

函数 CREMEZ 的最基本调用格式为：

b=cremez (n, f, 'frep')

b=cremez (n, f, {'frep', P_1, P_2…}, w)

b=cremez (n, f, a, w)

其中，n 为滤波器阶数；f 为滤波器多频带边界频率向量，取值可在－1～1 之间；'frep'为滤波器预先定义的频率响应函数：

(1) lowpass (低通)、highpass (高通)、bandpass (带通)、bandstop (带通)等用于设计标准型式滤波器。

如　　b = cremez (n, f, 'lowpass',…)

(2) multiband 用于设计多频带线性相位任意幅频响应滤波器。

如　　b = cremez (n, f, {'multiband', a}, …)

这里，a 为在边界频带向量 f 点上的期望幅值。

(3) differentiator 用于设计线性相位微分器。

如　　b = cremez (n, f, {'differentiator', F_s}, …)

这里，F_s 为采样频率，决定微分器响应斜率。

(4) hilbfilt 用于设计线性相位赫尔伯特变换滤波器，设计时，零频必须在过渡带上。

如　　b = cremez (n, f, 'hilbfilt',…)

w 为权向量，对每个频段上幅值拟合度加权，非负值，w 向量长度为频率向量 f 的一半。

【例 4.24】 利用函数 cremez 设计一个 38 阶任意响应多频带滤波器。

滤波器的边界频率向量：

$$f = [-1, -0.5, -0.4, 0.3, 0.4, 0.8]$$

各频段的幅值响应：

$$a = [5, 1, 2, 2, 2, 1]$$

各频段的最优化权向量：

$$w = [1, 10, 5]$$

用 MATLAB 编写滤波器设计程序如下：

```
%MATLAB PROGRAM 4-24
%Complex and nonlinear phase equiripple FIR filter design.
clf
n=38;
f=[-1 -0.5 -0.4 0.3 0.4 0.8];
a=[5 1 2 2 2 1];
w=[1 10 5];
%Design FIR filter by CREMEZ
b=cremez(n,f,{'multiband',a},w);
[h,wf]=freqz(b,1,512,'whole' );
%Output
axes('position',[0.2 0.2 0.5 0.5]);
plot(wf/pi-1,fftshift(abs(h)));
xlabel('Normanized Frequency');
ylabel('Magnitude Response')
legend('cremez')
grid on
```

设计的滤波器幅频特性如图 4.22 所示。

这种多带滤波器系数是复数，因为滤波器在频域内是不对称的，滤波器脉冲响应如图 4.23 所示。

4.10.2 复响应滤波器群延迟设置

用函数 cremez 设计的复响应滤波器的群延迟可通过参数 d 复位，以便滤波器的响应在单位采样间隔内的群延迟为(n/2+d)，d 可为正，也可为负。正的 d 产生较小的群延迟；负的 d 产生较大的群延迟。此时，函数调有格式为：

对于低通、高通、带通、带阻等滤波器：

```
b=cremez (n, f, {'lowpass', d}, …)
```

对于多频带任意响应滤波器：

```
b=cremez (n, f, {'multiband', a, d}, …)
```

对于线性相位微分器

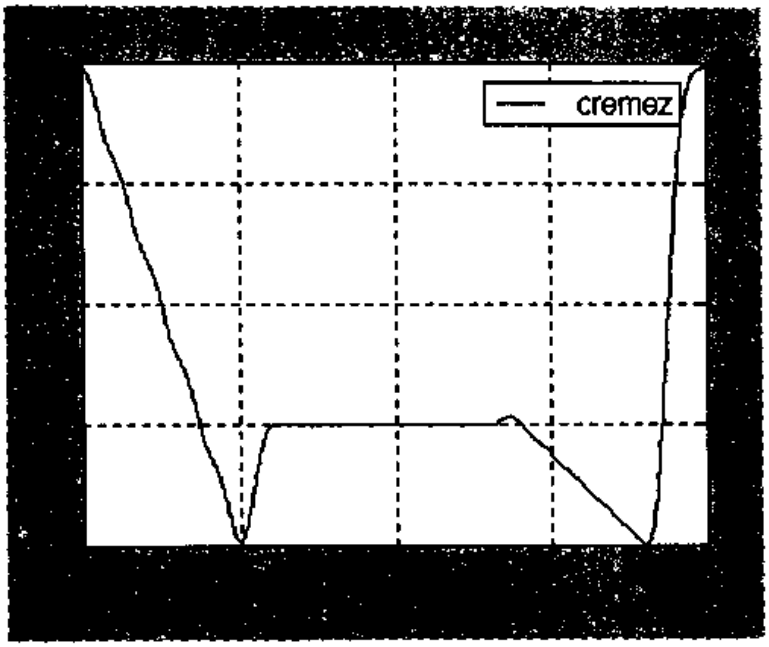


图 4.22　任意复响应滤波器幅频图

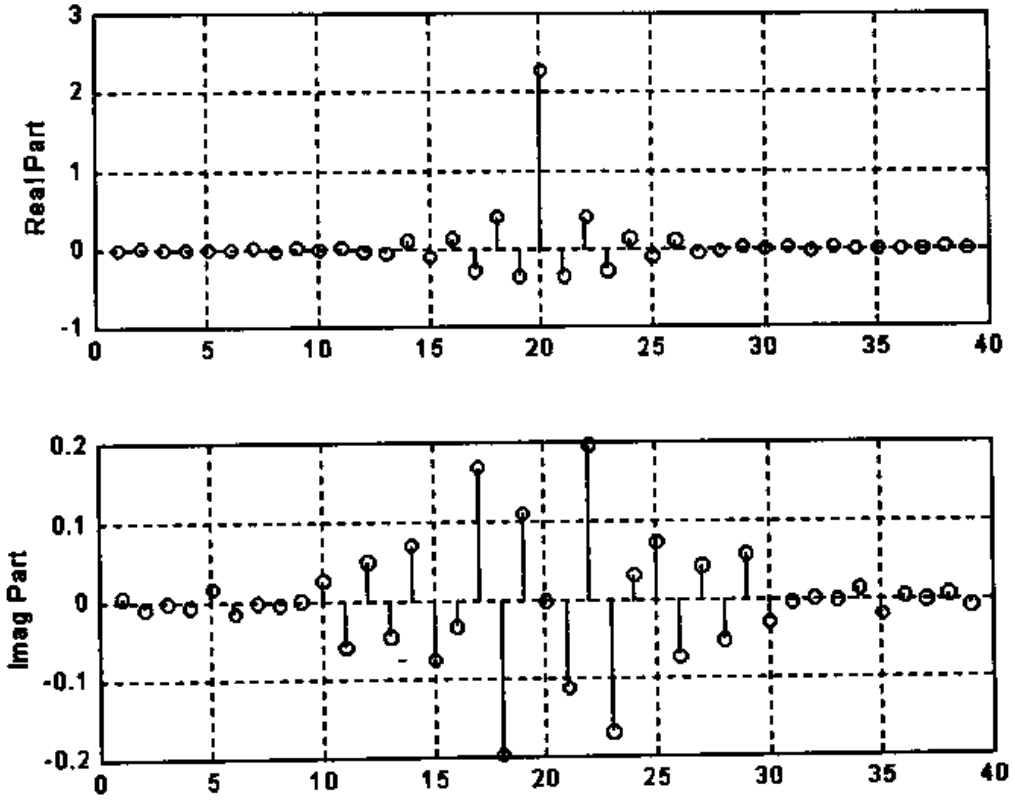


图 4.23　多频带滤波器的脉冲响应

b=cremez (n, f, ('differentiator', F., d), …)

对于赫尔伯特变换器

b=cremez (n, f, {'hilbfilt', d}, …)

【例 4.25】 利用函数 cremez 设计一个 61 阶低通滤波器，通带边界频率为 0.5，阻带边界频率为 0.55，群延迟比标准线性相位设计减小 16，绘制其幅频图和群延迟图并与线性相位设计比较。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 4-25
%Nonlinear phase equiripple FIR filter design.
clf
n=61;
f=[0 0.5 0.55 1];
d=-16;
%Nonlinear-phase design
%=============
%Design FIR filter by CREMEZ
b=cremez(n,f,{'lowpass',d});
[h,w]=freqz(b,1,512,'whole' );
%Output
subplot(221)
plot(w/pi-1,fftshift(abs(h)));
xlabel('Normanized Frequency');
ylabel('Magnitude Response')
legend('Nonlinear Phase')
grid on
subplot(222)
[Gd,W] = grpdelay(b,1,512,'whole');
plot(wf/pi-1,Gd);
xlabel('Normanized Frequency');
ylabel('Group Delay')
legend('Nonlinear Phase')
grid on
%Linear-phase design
%=========
%Design FIR filter by CREMEZ
b=cremez(n,f,'lowpass');
[h,w]=freqz(b,1,512,'whole' );
%Output
subplot(223)
plot(w/pi-1,fftshift(abs(h)));
```

```
xlabel('Normanized Frequency');
ylabel('Magnitude Response');
legend('Linear Phase')
grid on
subplot(224)
[Gd,W] = grpdelay(b,1,512,'whole');
plot(wf/pi-1,Gd);
xlabel('Normanized Frequency');
ylabel('Group Delay');
legend('Linear Phase')
grid on
```

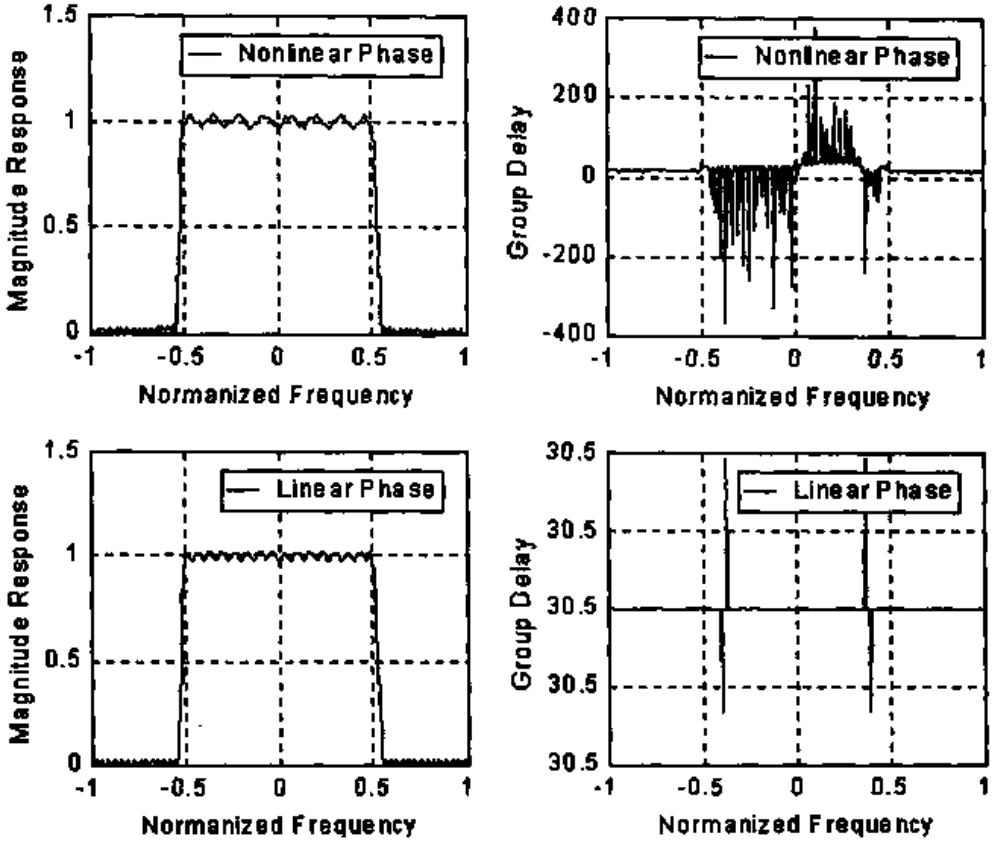


图 4.24 复响应滤波器的线性和非线性相位设计

由图 4.24 可见，线性相位设计滤波器的群延迟为 61/2=31.5，且在阻带和通带内均为常数；非线性相位设计群延迟为 31.5−16=14.5，但在通带内呈现非线性特点。同时，在幅频图上，非线性相位设计滤波器通带波纹明显高于线性相位设计。了解这些，可帮助我们在不同应用场合选用合适的 FIR 数字滤波器。

4.11 升余弦低通 FIR 滤波器设计

MATLAB 信号处理工具箱函数 FIRRCOS 用于设计具有光滑、正弦过渡带的低通线性相位滤波器，调用格式为：

$$b=firrcos\ (n,\ F_0,\ df,\ F_s)$$

其中，n 为滤波器的阶数，取偶数；F_0 为低通滤波器截止频率，Hz；df 为过渡带频宽，Hz；F_s 为采样频率，Hz；F_0 必须在 0 至 $F_s/2$ 之间。$F_0 \pm df/2$ 也必须在 0 至 $F_s/2$ 之间；b 为返回的滤波器系数向量。

【例 4.26】 设计一个 20 阶升余弦 FIR 滤波器，截止频率 250Hz，过渡带宽为 100Hz，采样周期为 1000Hz，绘制其幅频图。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 4-26
%Raised Cosine FIR Filter design
n=20;
Fo=250;
df=100;
Fs=1000;
%Design the filter
b=firrcos(n,Fo,df,Fs);
[h,f]=freqz(b,1,512,Fs);
%Output
axes('position',[0.2 0.2 0.5 0.5]);
plot(f,abs(h));
xlabel('Frequency (Hz)');
ylabel('Magnitude Response');
grid on
```

设计的滤波器幅频特性如图 4.25 所示。

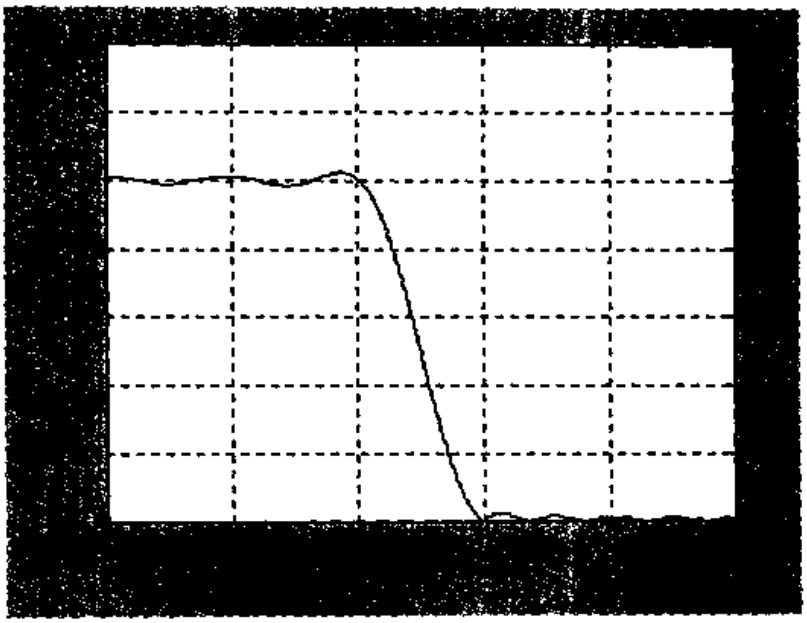


图 4.25 升余弦 FIR 滤波器幅频图

习 题

4.1 用脉冲响应不变法将以下模拟滤波器 H(s)转变为数字滤波器 H(z)。

① $H_a(s)=\frac{3}{(s+1)(s+3)}$，采样周期 T=0.5s；

② $H_a(s)=\frac{2}{s^2+s+1}$，采样周期 T=2s；

③ $H_a(s)=\frac{s+2}{2s^2+3s+1}$，采样周期 T=2s；

④ $H_a(s)=\frac{2}{(s+1)^2+4}$，采样周期 T=1s.

4.2 用双线性变换法重复题 4.1。

4.3 用脉冲响应不变法设计一个低通巴特沃斯低通数字滤波器，要求满足下面性能指标：通带为 $0\leqslant\omega\leqslant0.2613\pi$，通带波纹小于 2dB，阻带为 $0.4018\pi\leqslant\omega\leqslant\pi$，阻带衰减大于 20dB，采样周期 T=0.1s。

4.4 用双线性变换法设计一个低通巴特沃斯低通数字滤波器，性能指标同题 4.3。

4.5 用双线性变换法设计一个三阶巴特沃斯数字高通滤波器，采样频率 $F_s=6kHz$，截止频率 $f_c=1.5kHz$。

4.6 用双线性变换法设计一个三阶巴特沃斯数字带通滤波器，采样频率 $F_s=1000Hz$，上截止频率 $f_1=100Hz$，下截止频率 $f_2=300Hz$。

4.7 设计一个低通切比雪夫 I 型 IIR 滤波器，通带边界频率为 1500Hz，通带波纹小于 3dB，阻带边界频率为 2000Hz，衰减为 40dB，采样频率 8000Hz，试绘出其频率特性图。

4.8 设计一个带通切比雪夫 I 型 IIR 滤波器，通带边界频率 $0.4\pi\sim0.5\pi$，阻带边界频率分别为 0.3π 和 0.6π，通带波纹 0.5dB，阻带衰减 50dB，试绘制所设计滤波器的脉冲响应和对数幅频响应图。

4.9 用直接法设计一个多频带数字 IIR 滤波器，其理想幅频响应为：

频率 $\omega=[0\quad 0.1\quad 0.2\quad 0.3\quad 0.4\quad 0.5\quad 0.6\quad 0.7\quad 0.8\quad 0.9\quad 1]\pi$；

幅值 a=[0 0 1 1 1 0 0 1 1 0 0]；

绘制其理想幅频图并与实际滤波器幅频图相比较。

4.10 用汉宁窗设计一个带阻 FIR 滤波器，性能指标为：阻带边界频率 $0.4\pi\sim0.6\pi$，低通带边界频率 0.3π，高通边界频率 0.7π，通带波纹 1dB，阻带衰减 40dB。绘制其频率特性图。

4.11 用哈明窗设计一个带通 FIR 滤波器，性能指标为通带边界频率 $0.4\pi\sim0.6\pi$，低阻带边界频率 0.3π，高阻带边界频率 0.7π，通带波纹 1dB，阻带衰减 50dB。绘制其频率特性图。

4.12 用凯塞窗设计一个高通 FIR 滤波器，满足下面性能指标：

(1) $0\leqslant\omega\leqslant0.25\pi$，　$0\leqslant|H(e^{j\omega})|\leqslant0.01$

(2) $0.35\pi\leqslant\omega\leqslant0.65\pi$，　$0.95\leqslant|H(e^{j\omega})|\leqslant1.05$

$0.75\pi\leqslant\omega\leqslant\pi$，　$0\leqslant|H(e^{j\omega})|\leqslant0.01$

绘制其频率特性图。

4.13 用窗函数法设计一个多频带 FIR 滤波器，理想幅频响应同题 4.9，用试探法选择滤波器适当阶数并绘制滤波器频率特性图。

4.14 用 MATLAB 函数 firls 和 remez 设计一个带通滤波器，理想幅频响应对为

频率 ω=[0 0.3 0.4 0.7 0.8 1]π；

幅值 a=[0 0 1 1 0 0]；

绘制其理想幅频图并与实际幅频图相比较。

4.15 设计一个带通等波纹滤波器，理想幅频响应对如题 4.14，通带波纹小于阻带波纹 10 倍以上。绘制该滤波器的频率特性图。

4.16 用加权 CLS 法设计一个 55 阶高通 FIR 滤波器，截止频率为 0.3π，阻带边界频率 0.28π，通带边界频率为 0.32π，通带波纹小于 0.02，阻带波纹小于 0.008，阻带内误差平方和是通带的 1/5。绘制滤波器实际幅频特性图及通带和阻带内幅频图。

4.17 设计一个任意响应多频带 FIR 滤波器，滤波器的理想幅频对为：

频率 ω=[−1 −0.2 −0.5 0.1 0.6 0.8]π；

幅值 a=[0 1 1 2 2 1]；

绘制其滤波器实际幅频图及脉冲响应图。

4.18 设计一个 20 阶升余弦低通滤波器，通带边界频率为 0.3π，阻带边界频率为 0.4π，采样周期为 100Hz，绘制其频率特性图。

第五章　滤波器的实现和分析

上一章根据给定的技术指标设计了数字滤波器，得到了滤波器的传递函数 H(z)。本章主要讨论如何把传递函数 H(z)变成具体的数字系统，即滤波器的实现问题，并介绍分析滤波器的时域和频域特性的 MATLAB 函数。

5.1　数字滤波器的实现

5.1.1　硬件实现和软件实现

数字滤波器对信号的处理过程，实质上是将输入序列通过一定运算转变为输出序列的过程。例如一个数字滤波器的系统函数为

$$H(z)=\frac{\sum_{k=0}^{M}b_k z^{-k}}{1+\sum_{k=1}^{N}a_k z^{-k}} \tag{5.1-1}$$

所对应的差分方程为

$$y(n)=\sum_{k=0}^{M}b_k x(n-k)-\sum_{k=0}^{N}a_k y(n-k) \tag{5.1-2}$$

由式(5.1-2)可见，滤波器可由一台专门的设备来实现，这台设备由输入和输出的延时器、系数 a_k 和 b_k、存贮器，以及具备乘法和加法功能的运算器组成。每一采样时刻，设备执行式(5.1-2)计算，这就是数字滤波器的硬件实现。DSP(数字信号处理)芯片就是利用 MOS 超大规模集成技术，将一个比较完整的处理系统(其中包括运算器、高速硬件乘法器、数据 RAM、ROM 等)集成在一块芯片上，使许多复杂的实时信号处理任务变为现实，也包括实现数字滤波器。

数字滤波器的实现也可利用通用计算机编程，实现式(5.1-2)式运算，这就是软件实现。软件实现的主要优点是设计简单，投资小，灵活性好，主要缺点是运算速度慢，不适用于高速实时处理。

5.1.2　数字滤波器的运算结构图

数字滤波器无论是采用硬件实现还是采用软件实现方案，首先应先确定出数字滤波器的运算结构图。同一个系统函数或差分方程可以采用不同的结构来实现，而结构的不同又会影响系统的精度、稳定性、运行速度和采用运算单元的多少等许多重要性能指标。

运算结构可以用方框图来表示，也可用信号流图来表示。用信号流图表示滤波器有延时元件、乘法器、加法器三个运算单元，如图 5.1 所示。

例如，一阶数字滤波器的系统函数和差分方程为

$$H(z)=\frac{b_0}{1+a_1 z^{-1}}$$

$$y(n)=b_0 x(n)-a_1 y(n-1)$$

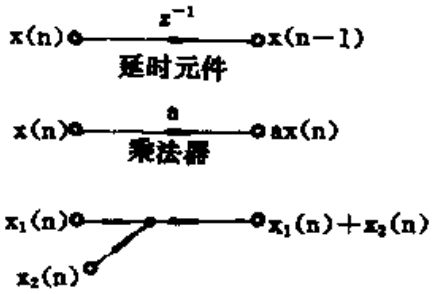


图 5.1 基本运算单元

它的信号流程图如图 5.2 所示。

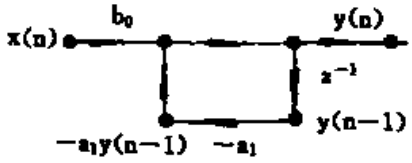


图 5.2 一阶数字滤波器信号流图

5.1.3 IIR 数字滤波器的结构

无限冲激响应(IIR)滤波器的系统函数和差分方程分别如式(5.1-1)和(5.1-2)所示，其特点是系统函数 H(z)在有限 z 平面上有极点存在，它的单位脉冲响应 h(n)延续到无限长，而它结构上存在反馈回路，即递归型结构。实现 IIR 滤波器有三种结构：直接型、级联型、并联型。

一、直接型结构

IIR 滤波器的具体实现如图 5.3 所示为直接 I 型。

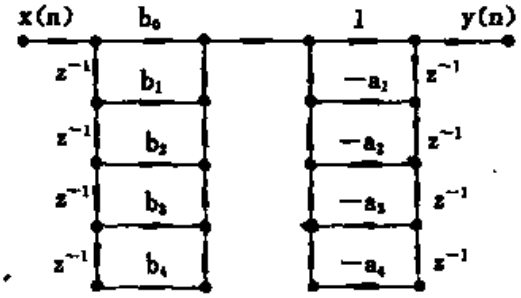


图 5.3 IIR 滤波器直接 I 型结构

IIR 滤波器的具体结构还有直接 II 型如图 5.4 所示。直接 II 型比直接 I 型延时单元可节省一半，因此经济性较好，但这两种结构是等价的。

直接型结构虽然简便地实现 H(z)，但由于在实现时系数 ak 和 bk 不可避免存在误差(如软件量化误差)，导致系统零极点发生变化，引起频响较大误差，甚至出现不稳定现象。

MATLAB 信号处理工具箱的函数 FILTE 利用直接 II 型滤波器结构计算滤波器对

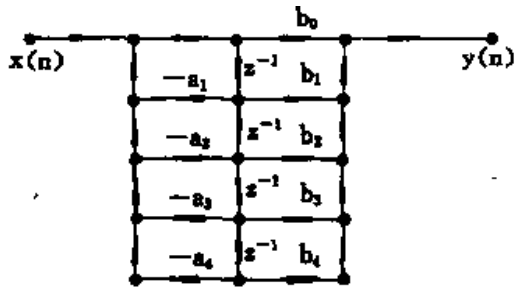


图 5.4 IIR 滤波器直接 I 型结构

输入的响应。

函数 filter 也常用于求任何离散时间系统对任意输入的响应，是 MATLAB 信号处理工具箱中主要的时域分析工具。

【例 5.1】 求巴特沃斯滤波器的单位脉冲响应。用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 5-1
%Transposed direct form II structure
x=[1 zeros(1,119)];
[b,a]=butter(12,400/1000);
y=filter(b,a,x);
stem(y);
grid
```

运行结果如图 5.5 所示。

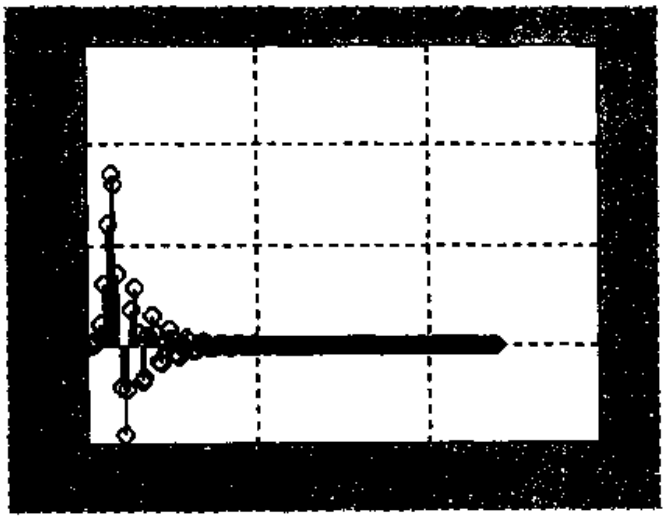


图 5.5 滤波器的单位脉冲响应

二、级联型结构

将系统函数 H(z)用实系数二阶因子相乘形式表示

$$H(z)=\prod_{k=1}^{L}H_k(z)=\prod_{k=1}^{L}\frac{b_{0k}+b_{1k}z^{-1}+b_{2k}z^{-2}}{a_{0k}+a_{1k}z^{-1}+a_{2k}z^{-2}} \tag{5.1-3}$$

式(5.1-3)所对应的滤波器结构可看成由若干个双二阶环节 $H_k(z)$级联而成。每个双二阶环节结构如图 5.6 所示，图中，第 k 级双二阶环节的输入 xk 是前一个双二阶环节输

出 $y_{k-1}(n)$，而第 k 级双二阶环节的输出 $y_{k(n)}$是下一个双二阶环节的输入 $x_{k+1}(n)$。

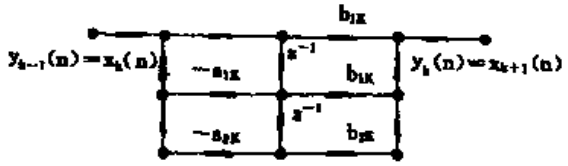


图 5.6 IIR 滤波器双二阶环节级联结构($a_{0k}=1$)

这种结构正是 MATLAB 信号处理工具箱定义的 SOS 模型，如式(1.7-6)所示。

若已知 H(z)为传递函数形式，可借助 MATLAB 信号处理工具箱的函数 tf2sos 将滤波器的直接结构型转换为级联结构。

【例 5.2】 一个滤波器的系统函数为

$$H(z)=\frac{1-3z^{-1}+11z^{-2}-27z^{-3}+18z^{-4}}{16+12z^{-1}+2z^{-2}-4z^{-3}-z^{-4}}$$

求它的级联形式结构和系统的脉冲响应。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 5-2
%Original Model of the filter
num=[1 -3 11 -27 18];
den=[16 12 2 -4 -1];
%Convert to Cascade form of filter
[z,p,k]=tf2zp(num,den);
sos=zp2sos(z,p,k)
%Check impulse response of the cascade form
num1=conv(sos(1,1:3), sos(2,1:3));
den1=conv(sos(1,4:6), sos(2,4:6));
x=[1,zeros(1,40)];
n=[0:length(x)-1];
y=filter(num,den,x);
y1=filter(num1,den1,x);
subplot(221)
plot(n,y);
title('Impulse Response')
legend('Original')
subplot(222)
plot(n,y1)
legend('Cascade form')
title('Impulse Response')
```

```
≫smp502
sos =
    0.1875   -0.5625    0.3750    1.0000   -0.2500   -0.1250
    0.3333    0.0000    3.0000    1.0000    1.0000    0.5000
```

即系统SOS形式为

$$H(z)=\frac{0.1875-0.5625z^{-1}+0.375z^{-2}}{1-0.2500z^{-1}-0.1250z^{-2}}\cdot\frac{0.3333+3.0z^{-2}}{1+z^{-1}+0.5z^{-2}}$$

级联结构如图5.7所示。

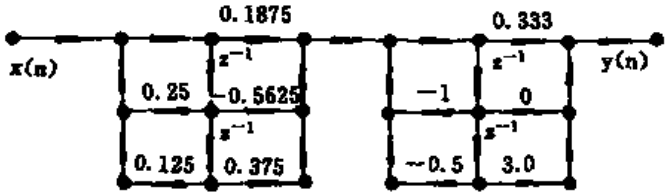


图5.7 11R的级联结构

图5.8为直接形式滤波器的单位响应和级联形式滤波器的单位脉冲响应，可见二者完全相同。这说明结构形式的改变不影响滤波器的时域特性。

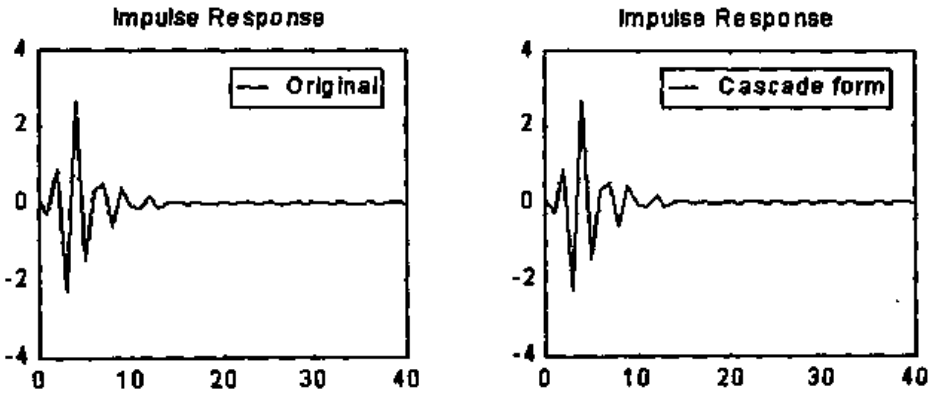


图5.8 滤波器的脉冲响应

三、并联型结构

如将系统函数H(z)展开成部分分式形式就构成了并联型结构的滤波器。

$$H(z)=\sum_{k=1}^{L}\frac{b_{0k}+b_{1k}z^{-1}}{a_{0k}+a_{1k}z^{-1}+a_{2k}z^{-2}}+\sum_{k=0}^{M-N}c_kz^{-k} \tag{5.1-4}$$

这可理解为一阶环节和二阶环节的并联。

MATLAB信号处理工具箱没有直接提供该结构的生成信号，但将式(5.1-4)和式(1.7-4)比较会看到，借助函数residuez可以获得FIR滤波器并联结构的参数，其方法是这样的：先用residuez求出H(z)的部分分式展开式，然后把分式两两合并成式(5.1-4)形式。若分式项为偶数，则并联结构全为二阶环节；若分式项为奇数，则并联结构除二阶环节外，还包括一阶环节(也可认为是特殊的二阶环节)。下面给出一个例子。

【例5.3】 求例5.2中滤波器的并联结构用MATLAB编程如下：

```
%MATLAB PROGRAM 5-3
%Original Model of the filter
numorig=[1 -3 11 -27 18];
denorig=[16 12 2 -4 -1];
%Convert to partial-fraction expansion
[r,p,c]=residuez(numorig,denorig);
n1=length(r);
%Convert to Parallel structure
if rem(n1,2)==0
    N=n1;
else
    N=n1-1;
end
numgroup=zeros(N/2,2);
dengroup=zeros(N/2,2);
for i=1:N/2
    numgroup(i,:)=r(i*2-1:i*2)';
    dengroup(i,:)=p(i*2-1:i*2)';
end
for i=1:N/2
    denparp(i,:)=conv(poly(dengroup(i,1)),poly(dengroup(i,2)));
    numparp(i,:)=numgroup(i,1)*poly(dengroup(i,2))...
                 +numgroup(i,2)*poly(dengroup(i,1));
end
if rem(n1,2)~=0
  nn1=n1-N/2;
  for i=1:nn1
      demparp(i+N/2,:)=[poly(p(N+i)) 0];
      numparp(i+N/2,:)=[r(N+i) 0];
  end
end
disp('Whole form for the Parallel structure:')
sys1=filt(numparp(1,:),denparp(1,:))
sys2=filt(numparp(2,:),denparp(2,:))
c

≫ smp503
Whole form for the Parallel structure:
```

Transfer function:

$$\frac{-10.05 - 3.95\,z^{-1}}{1 + z^{-1} + 0.5\,z^{-2}}$$

Sampling time: unspecified

Transfer function:

$$\frac{28.11 - 13.36\,z^{-1}}{1 - 0.25\,z^{-1} - 0.125\,z^{-2}}$$

Sampling time: unspecified

c =

−18

滤波器并联型结构如图 5.9 所示。

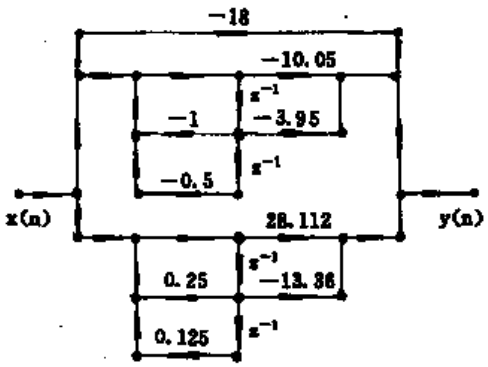


图 5.9 IIR 滤波器的并联结构

5.1.4 FIR 数字滤波器的结构

一、直接型结构

有限冲激响应(FIR)滤波器的系统函数为

$$H(z) = \sum_{n=0}^{N-1} b(n) z^{-n} \tag{5.1-5}$$

其差分方程

$$y(n) = \sum_{m=0}^{N-1} b(m) x(n-m) \tag{5.1-6}$$

FIR 滤波器结构实现的信号流图如图 5.10 所示

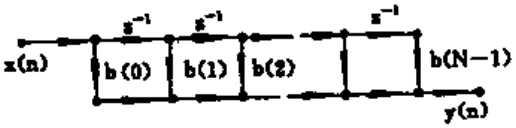


图 5.10 FIR 滤波器的直接型结构

FIR 滤波器的直接结构由 MATLAB 函数 filter 来实现，调用格式见 5.2.1 节，只是令 a=1。

二、级联型结构

如将 H(z)分解成二阶实系数因子形式

$$H(z) = \sum_{k=1}^{M/2} (b_{0k} + b_{1k} z^{-1} + b_{2k} z^{-2})$$

这样可得 FIR 滤波器二阶级联结构，如图 5.11 所示。

FIR 滤波器的级联结构可用 MATLAB 编程获得，其方法和例 5.2 类似。

三、线性相位型结构

线性相位 FIR 滤波器满足下面偶对称条件

$$b(n) = b(M - 1 - n) \tag{5.1.7}$$

故当 M 为偶数时

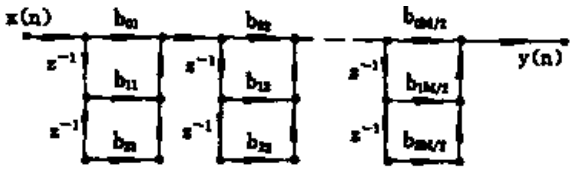


图 5.11 FIR 滤波器级联结构

$$H(z)=\sum_{n=0}^{\frac{M-1}{2}-1}b(n)[z^{-n}+z^{-(M-1-n)}] \tag{5.1.8}$$

当 M 为奇数时

$$H(z)=\sum_{n=0}^{\frac{M-1}{2}-1}b(n)[z^{-n}+z^{-(M-1-n)}]+b\left(\frac{M-1}{2}\right)z^{\frac{M-1}{2}} \tag{5.1.9}$$

当 M 为奇数和偶数时,FIR 的线性相位型结构如图 5.12 所示。

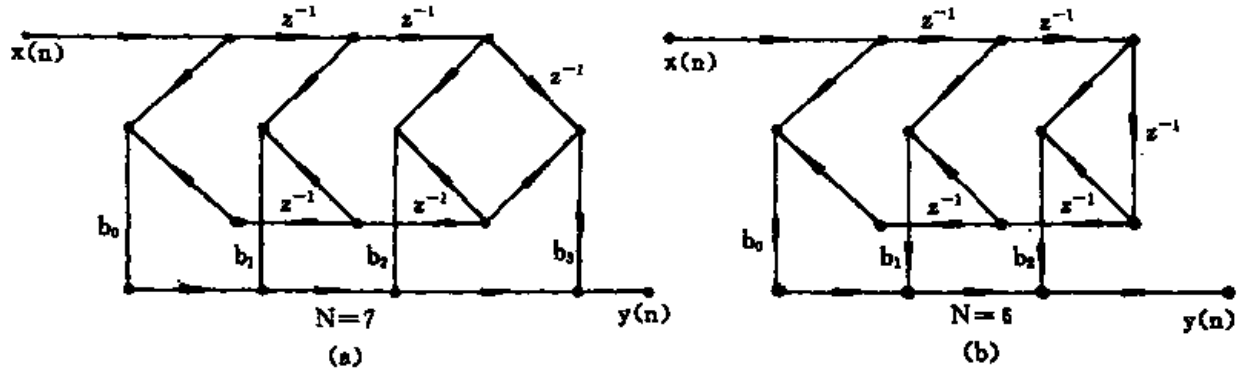


图 5.12 FIR 滤波器线性相位型结构

这种结构本质上是直接型,但乘法次数较直接型节省一半,以上几种结构的 MATLAB 实现均和 IIR 相同,只是 H(z)的分母为常数 1。

【例 5.4】 已知 FIR 滤波器的系统函数为

$$H(z)=1+\frac{17}{16}z^{-4}+z^{-8}$$

确定并给出滤波器的直接型,线性相位型和级联型结构图。

借用例 5.1 和例 5.2 的 MATLAB PROGRAM,可得出 FIR 滤波器的结构参数。

```
≫ smp 502a
sos =
    0.2686    -0.4628    0.2686    1.0000    0         0
    1.0160     1.0320    1.0160    1.0000    0         0
    0.8616     1.4845    0.8616    1.0000    0         0
    4.2525    -4.3195    4.2525    1.0000    0         0
```

(1)直接型结构

$$y(n)=x(n)+16.0625x(n-4)+x(n-8)$$

(2)级联型

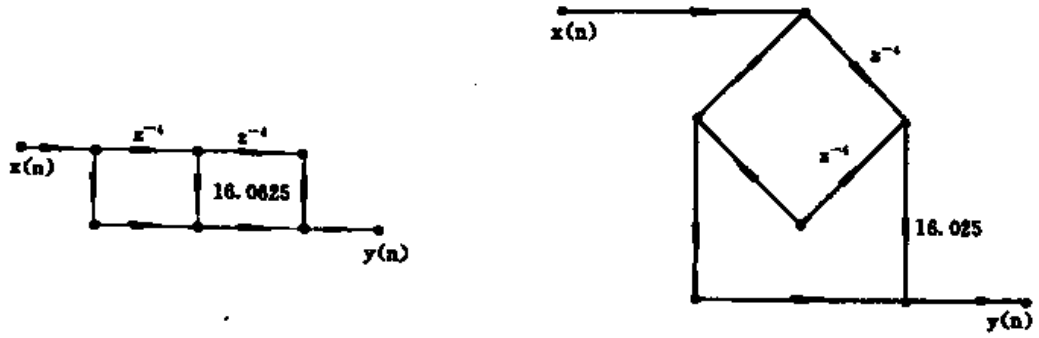
$$H(z)=(0.2686-0.4628z^{-1}+0.2686z^{-2})$$

$$(1.0160 + 1.0320z^{-1} + 1.0160z^{-2})$$
$$(0.8616 + 1.48452z^{-1} + 0.8616z^{-2})$$
$$(4.2525 + 4.3195z^{-1} + 4.2525z^{-2})$$

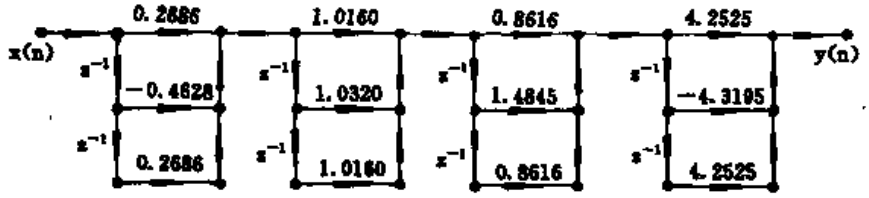
(3)线性相位型

$$y(n) = [x(n) + x(n-8)] + 16.062x(n-4)$$

这几种结构如图 5.13 所示。



(a) 直接型　　(b) 线性相位型



(c) 级联型

图 5.13　FIR 滤波器的结构

四、频率采样结构

一个有限时宽为 N 的序列，其 z 变换可用单位圆上等间隔频域采样来表示。由内插公式(2.4-3)，对于 FIR 滤波器的系统函数 H(z)可表示为

$$H(z) = (1 - z^{-N})\frac{1}{N}\sum_{k=0}^{N-1}\frac{H(k)}{1 - W_N^{-k}z^{-1}} \qquad (5.1\text{-}10)$$

式中

$$W_N^{-k} = e^{j\left(\frac{2\pi}{N}\right)k}$$
$$H(k) = DFT[h(n)]$$
$$H(z) = \mathcal{Z}[h(n)]$$

和式(5.1-10)对应于滤波器是由两部分——一部分为$(1-z^{-N})/N$，第二部分是 N 个 $H(k)/(1-W_N^{-k}z^{-1})$并联而成。图 5.14 给出 N=4 的频率采样结构。

由零极点分析可知，这一结构的极点准确地位于单位圆上，正好抵消其零点。而实际上，位于单位圆上的极点会给系统的稳定性带来问题，因为一旦系统参数略有误差可能导

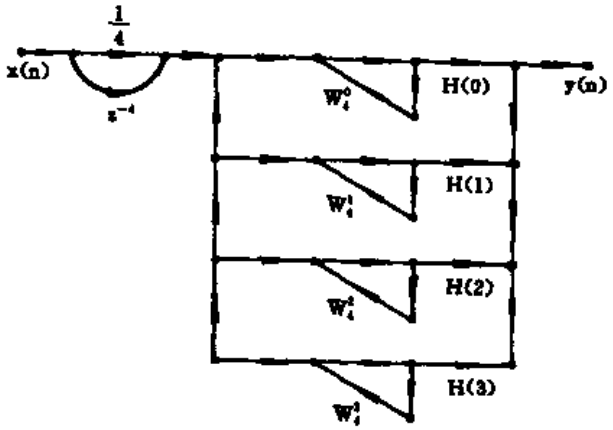


图 5.14 FIR 滤波器频率采样结构

致系统不稳，故一般在半径 r 略小于 1 的圆周上进行频率采样。

如果系统 H(z)的脉冲响应序列 h(n)为实序列，则它的 DFT 是圆周共轭对称的，利用这一性质可得 H(z)的表达式为

当 N 为偶数时

$$H(z)=(1-r^{N}z^{-N})\frac{1}{N}\left[\frac{H(0)}{1-rz^{-1}}+\frac{H\left(\frac{N}{2}\right)}{1+rz^{-1}}+\sum_{k=1}^{\frac{N}{2}-1}2|H(k)|H_k(z)\right] \quad (5.1\text{-}11)$$

当 N 为奇数时

$$H(z)=(1-r^{N}z^{-N})\frac{1}{N}\left[\frac{H(0)}{1-rZ^{-1}}+\sum_{k=1}^{\frac{N}{2}-1}2|H(k)|H_k(z)\right] \quad (5.1\text{-}12)$$

式中

$$Hk(z)=\frac{\cos[\angle H(k)]-rz^{-1}\cos[\angle H(k)-2\pi\frac{k}{N}]}{1-2rz^{-1}\cos\left(\frac{2\pi}{N}k\right)+r^{2}z^{-2}}$$

由式(5.1-11)和(5.1-12)可绘制 FIR 滤波器频率采样结构图。该结构由两部分组成：第一部分仍为$(1-r^{N}z^{-N})/N$，第二部分是并联部分。

若 N=4 时，第二部分可写成

$$2|H(1)|\frac{\cos[\angle H(1)]-r\cos[\angle H(1)-\frac{2\pi}{N}]z^{-1}}{1-2r\cos\left(\frac{2\pi}{N}\right)z^{-1}+r^{2}z^{-2}}+\frac{H(0)}{1-rz^{-1}}+\frac{H\left(\frac{N}{2}\right)}{1+rz^{-1}}$$

为了便于用 MATLAB 计算上式中的系数，写成矩阵形式

分母系数矩阵

$$A=\begin{bmatrix}1 & -2r\cos\left(\frac{2\pi}{N}\right) & r^{2}\\ 1 & -1 & 0\\ 1 & 1 & 0\end{bmatrix}$$

分子系数矩阵

$$B = [\cos[\angle H(1)] - r\cos[\angle H(1) - \frac{2\pi}{N}]$$

各项增益矩阵

$$C = \begin{bmatrix} 2|H(1)| \\ H(0) \\ H\left(\frac{N}{2}\right) \end{bmatrix}$$

当 N>4 的任意数时，均可写成类似的系数矩阵。但应注意，矩阵中二阶系数的行数不再是一行。当 N 为偶数时为$\left(\frac{N}{2}-1\right)$行；当 N 为奇数时，$\frac{N-1}{2}$为行。下面用 MATLAB 编写一个由 FIR 滤波器直接型转变为频率采样型函数 dir2fs，函数输入为 FIR 滤波器的脉冲响应 h(n)和频率采样圆周半径 r，函数输出为 FIR 滤波器频率采样型的第二部分系数矩阵 A、B、C。

```
function [A,B,C]=dir2fs(h,r)
%To convert direct form of FIR to frequency sampling structure
% h(n)---Impulse responce of a FIR filter
% r-------Diameter of cycle for frequency sampling r>0.9
% A-------Matrix of denominator coefficents
% B-------Matrix of numerator coefficents
% C-------Matrix of gain coefficients
N=length(h);
H=fft(h,N);
magH=abs(H);
phaH=angle(H)';
if(rem(N,2)==0)
  L=N/2-1;
  A1=[1 -1 0;1 1 0];
  C1=[real(H(1)),real(H(L+2))];
else
  L=(N-1)/2;
  A1=[1 -1 0;
  C1=[real(H(1))];
end
k=[1:L]';
B=zeros(L,2);
A=ones(L,3);
%Compute Matrix of denominator coefficents
A(1:L,2)=-2*r*cos(2*pi*k/N);
A(1:L,3)=r^2;
```

```
A=[A;A1];
%Compute Matrix of numerator coefficents
B(1:L,1)=cos(phaH(2:L+1));
B(1:L,2)=-r*cos(phaH(2:L+1)-(2*pi*k/N));
%Compute Matrix of gain coefficients
C=[2*magH(2:L+1),C1]';
```

【例 5.5】 已知滤波器的单位脉冲响应

$$h(n)=\frac{1}{9}\{1,2,3,2,1\}$$

绘制该滤波器频率采样结构。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 5-4
h=[1 2 3 2 1]/9;
r=1;
[AA,BB,CC]=dir2fs(h,r);
disp('Matrix of denominator coefficents ')
AA
disp('Matrix of numerator coefficents')
BB
disp('Matrix of gain coefficients ')
CC
```

```
≫ smp504
Matrix of denominator coefficents
AA =
    1.0000   -0.6180    1.0000
    1.0000    1.6180    1.0000
    1.0000   -1.0000         0
Matrix of numerator coefficents
BB =
   -0.8090    0.8090
    0.3090   -0.3090
Matrix of gain coefficients
CC =
    0.5818
    0.0849
    1.0000
```

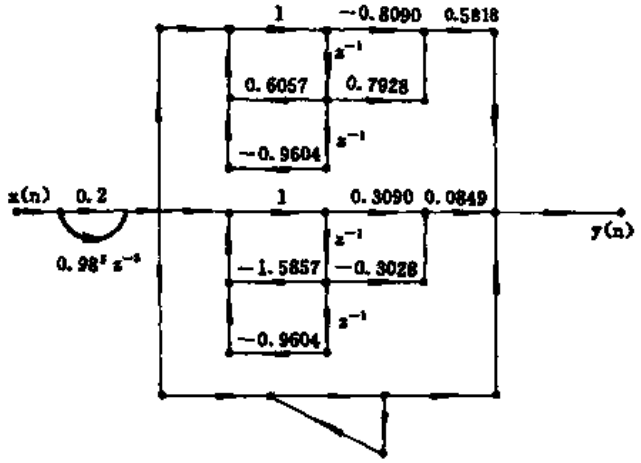


图 5.15　FIR 滤波器频率采样结构

5.1.5　滤波器的格型结构

格型结构是滤波器的另一种结构实现。它适用于 IIR 滤波器，也适用于 FIR 滤波器。格型滤波器广泛用于数字语音处理和自适应滤波器实现中。

格型结构可分为三类：全零点格型滤波器、全极点格型滤波器、格型梯形滤波器。

一、全零点格型滤波器

长度为 M(M−1 阶)的 FIR 滤波器的格型结构如图 5.16 所示。它用来实现 FIR 滤波器的系统函数 $H(z)=\sum_{m=0}^{M-1}b_m z^{-m}$。

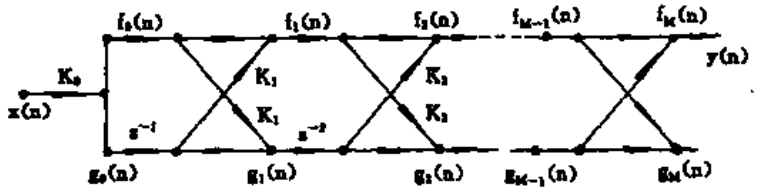


图 5.16　全零点格型滤波器

滤波器具有 M−1 级格型结构，每一级的输入输出通过顺序递推方程建立联系

$$f_m(n)=f_{m-1}(n)+k_m g_{m-1}(n-1)$$

$$g_m(n)=k_m f_{m-1}(n)+g_{m-1}(n-1),\ m=1,2,\cdots,M-1 \qquad (5.1\text{-}13)$$

式中，k_m，$m=1,2,\cdots M-1$ 称反射系数(Reflection COefficients)或称格型滤波器系数；初值

$$f_0(n)=g_0(n)=k_0 x(m)$$

系统输出为

$$y(n)=f_{M-1}(n)$$

MATLAB 信号处理工具箱中函数 POLY2RC 用来从 FIR 滤波器直接型结构计算它的格型结构。也就是，已知 FIR 滤波器的全零点多项式模型 $H(z)=\sum_{m=0}^{M-1}b_m z^{-m}$，计算 FIR 滤波器格型结构反射系数 k_m，$m=1,2,\cdots,M-1$。函数调用格式为

K=poly2rc(b)

式中,b 为 FIR 滤波器多项式系数实向量,b(1)不能为 0。K 为 FIR 滤波器格型结构反射系数。

当 FIR 以直接型结构(或全零点形式)表示时,

$$H(z)=\sum_{m=0}^{M-1}b_m z^{-m}=b_0\left(1+\sum_{m=1}^{M-1}\frac{b_m}{b_0}z^{-m}\right)$$

令

$$A_{M-1}(z)=1+\sum_{m=1}^{M-1}a_{M-1}(m)z^{-m}$$

$$a_{M-1}(m)=\frac{b_m}{b_0},m=1,2,\cdots,M-1$$

则格型滤波器反射系数 km 可由下面递归算法得出。

$$K_0=b_0$$

$$K_{M-1}=a_{M-1}(M-1)$$

$$J_M(z)=z^{-m}A_m(z^{-1}),m=M-1,\cdots,1$$

$$A_{m-1}(z)=\frac{A_m(z)-K_mJ_m(z)}{1-K_m^2},\ m=M-1,\cdots,1$$

$$K_m=a_m(m),\ m=M-1,\cdots,1$$

注意到,K_m 不能为 1,即线性相位 FIR 滤波器不能用格型结构来实现。

MATLAB 信号处理工具箱还提供由 FIR 格型结构滤波器的反射系数 K_m 求滤波器多项式系数 b 的函数 RC2POLY,调用格式为

b=rc2poly(K)

式中,各项意义同函数 POLY2RC。

【例 5.6】 已知 FIR 滤波器由下面差分方程给定

y(n)=x(n)+0.6149x(n−1)+0.9899x(n−2)+0.0031x(n−4)−0.0082x(n−5)

求它的格型结构。

用 MATLAB 编程如下:

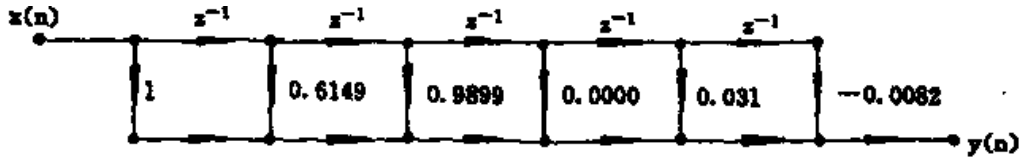
```
%MATLAB PROGRAM 5-5
b=[1.000 0.6149 0.9899 0.000 0.0031 −0.0082];
k=poly2rc(b)
```

```
≫ smp505
k =
    0.3090    0.9801    0.0031    0.0081    −0.0082
```

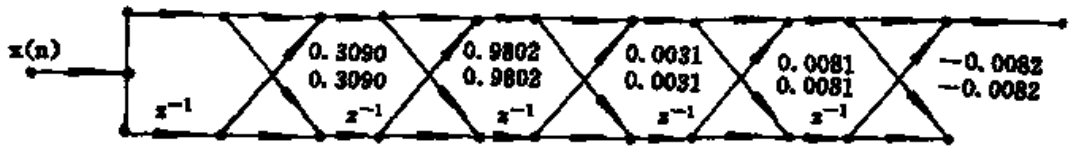
该滤波器的结构如图 5.17 所示。

二、全极点格型滤波器

设一个滤波器的全极点系统函数为



(a) 直接型结构



(b) 格型结构

图 5.17 例 5.6 的滤波器结构

$$H(z)=\frac{1}{1+\sum_{m=1}^{M-1}a_m z^{-m}}$$

该滤波器的格型结构如图 5.18 所示，图中 $K_m(m=1,2,\cdots,M)$为全极点格型滤波器反射系数。

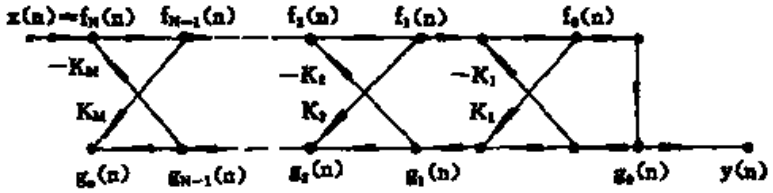


图 5.18 全极点格型滤波器

MATLAB 信号处理工具提供用于从系数函数分母多项式 a 求反射系数 k 的函数 TF2LATC。该函数应用将在下面讲述。

三、格型梯形滤波器

一般 IIR 滤波器的系统函数为

$$H(z)=\frac{\sum_{k=0}^{M}b_k z^{-k}}{1+\sum_{k=1}^{N}a_k z^{-k}}=\frac{b(z)}{a(z)} \qquad (N\geqslant M)$$

该滤波器可用格型梯形结构来实现，如图 5.19 所示。

图中， K_m，$(m=1,2,\cdots,M)$为滤波器格型系数(lattice parameter)，即反射系数；

V_n，$(n=0,2,\cdots,N)$为滤波器梯形系数(Ladder Parameter)。

可按递归关系从 H(z)计算格型系数 K_m 和梯形系数 V_n。

MATLAB 信号处理工具箱提供函数 TF2LATC 实现由一般 IIR 滤波器系统函数 H(z)求滤波器的格型梯形结构参数：K_m 和 V_n。该函数用格式为

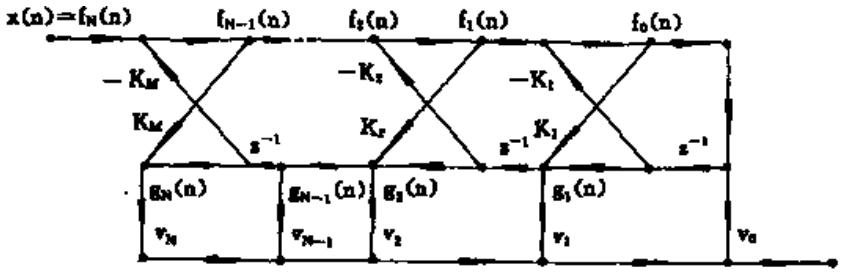


图 5.19　格型梯形滤波器

```
[K,V] = tf2latc(num,den)
K = tf2latc(1,den)
[K,V] = tf2latc(1,den)
k = tf2latc(num)
```

式中，num 为滤波器系统函数分子多项式系数向量；den 为滤波器系统函数分母多项式系数向量；K 为滤波器格型系数(反射系数)；V 为滤波器梯形系数。

上面第一种调用格式获得零极点格型梯形结构，第二、三种调用格式获得全极点格型滤波器，第四种调用格式，den＝1，获得全零点格型滤波器。因此该函数适用于 IIR 滤波器，也适用于 FIR 滤波器。

MATLAB 信号处理工具箱还提供了由格型滤波器求滤波器传递函数 LATC2TF。函数调用格式为

```
[num,den] = latc2tf(K,V)
[num,den] = latc2tf(K,'iir')
num = latc2tf(K,'fir')
num = latc2tf(K)
```

其中，iir 表示 K 是全极点 IIR 格型滤波器的反射系数，fir 表示 K 是全零点 FIR 格型滤波器的反射系数，其余各项意义同函数 TF2LATC。

【例 5.7】　求下面零—极点 IIR 滤波器的格型梯形结构参数

$$H(z)=\frac{1+2z^{-1}+2z^{-2}+z^{-3}}{1+\frac{13}{24}z^{-1}+\frac{5}{8}z^{-2}+\frac{1}{3}z^{-3}}$$

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 5-6
num=[1 2 2 1];
den=[1 13/24 5/8 1/3];
[K,V]=tf2latc(num,den);
disp('Lattice Parameter K=')
K'
disp('Ladder Parameter V=')
V'
```

```
≫ smp506
Lattice Parameter K=
ans =
    0.2500   0.5000   0.3333
Ladder Parameter V=
ans =
   -0.2695   0.8281   1.4583   1.0000
```

滤波器结构如图 5.20 所示。

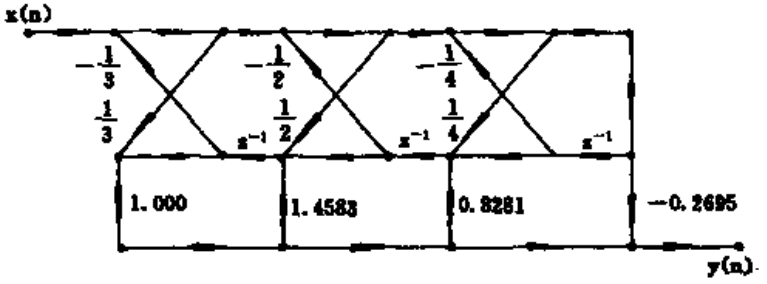


图 5.20 IIR 滤波器格型梯形结构

【例 5.8】 (1) 全极点滤波器

$$H(z)=\frac{1}{1+\frac{13}{24}z^{-1}+\frac{5}{8}z^{-2}+\frac{1}{3}z^{-3}}$$

(2) 全零点滤波器

$$H(z)=1+0.6149z^{-1}+0.9899z^{-2}+0.0031z^{-3}-0.0082z^{-4}$$

确定其格型结构。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 5-7
disp('All-pole filter:')
den=[1 13/24 5/8 1/3];
[K,V]=tf2latc(1,den)
disp('All-zero filter:')
num=[1.000 0.6149 0.9899 0.000 0.0031 -0.0082];
K1=tf2latc(num)
```

```
  smp507
All-pole filter:
K =
    0.2500
    0.5000
    0.3333
```

```
V =
     1
     0
     0
     0
All-zero filter:
K1 =
     0.3090
     0.9801
     0.0031
     0.0081
    -0.0082
```

该全零点滤波器模型和例 5.6 相同，计算结果也相同。可见，对于全零点滤波器的格型结构实现函数 tf2latc 和 poly2rc 均适用。

5.2 滤波器分析

MATLAB 信号处理工具箱提供了许多工具函数用来分析滤波器特性，包括时域分析，对任意输入响应、脉冲响应等；频域分析，幅值响应、相位响应、零极点位置；其他特性，群延迟等。滤波器时域和频域分析是设计各类滤波器、评价滤波器性能和滤波器应用的基础，在前面几章的内容里已应用了 MATLAB 这些工具、函数。本节对这些工具函数作详细介绍。

5.2.1 时间响应

在第一章里已介绍求离散时间系统时域响应的方法，这些方法也适用于滤波器。这里重点介绍 MATLAB 信号处理工具箱常用的滤波器时间响应分析工具函数 FILTER，FILTIC，FFTFILT 和 IMPULS。

(1) 函数 FILTER 用于实现 IIR 和 FIR 滤波器对数据滤波，常用来计算滤波器对输入响应。函数调用格式为：

$$y = \text{filter}(b, a, x)$$
$$[y, z_f] = \text{filter}(b, a, x)$$
$$[y, z_f] = \text{filter}(b, a, x, z_i)$$

其中，b，a 分别是滤波器系统函数 H(z)(DSP 形式)的分子和分母多项式系数向量 x 为滤波器输入，向量或矩阵；若 x 为矩阵，每组数据应为列向量；z_i 为状态向量初始值；z_f 为状态向量终止值；y 为滤波器输出。

MATLAB 构建的滤波器函数 filter 的实现采用直接 II 型结构如图 5.21 所示。

函数 FILTER 在采样点 m 按下式执行时域运算

$$y(m) = b(1)x(m) + z_1(m-1) - a(1)y(m)$$

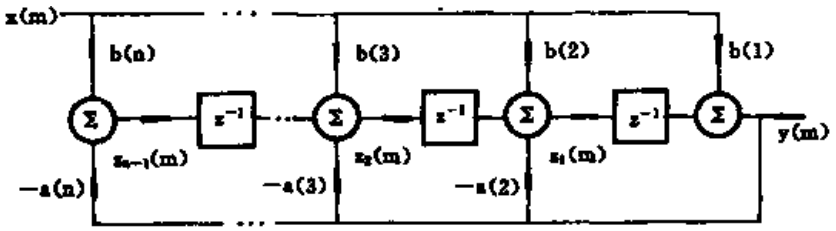


图 5.21 滤波器实现

$$z_1(m) = b(2)x(m) + z_2(m-1) - a(2)y(m)$$

……

$$z_{n-2}(m) = b(n-1)x(m) + z_{n-1}(m-1) - a(n-1)y(m)$$

$$z_{n-1}(m) = b(n)x(m) - a(n)y(m)$$

上式中，y 为滤波器输出；z_i 为滤波器状态；x 为滤波器输入。

用户可用 MATLAB 函数 FILTIC 由过去输入和输出值产生滤波器初始状态 $z_i(0)$。

$$y(z)=\frac{b(1)+b(2)z^{-1}+\cdots+b(nb+1)z^{-nb}}{a(1)+a(2)z^{-1}+\cdots+a(na+1)z^{-na}}$$

其中，nb，na 分别为分子和分母阶数。

函数 Filter 可用于全极点滤波器、全零点滤波器、零极点滤波器，见例 2.16、例 2.17、例 2.18。

(2) 函数 FILTIC 用于从滤波器过去值 y 和 x 求滤波器状态初始值。调用格式为

```
z =filtic(b,a,y,x)
z =filtic(b,a,y)
```

其中，y=[y(−1),y(−2),…,y(−na)]为输出 y 的过去值向量；x=[x(−1),y(−2),…,y(−nb)]为输入 x 的过去值向量；b，a 分别为滤波器分子、分母多项式系数向量；z 为滤波器初始状态。

函数 filter 设立状态初值 z_i 和终值 z_f，使我们可将一个特别长的信号“切断”分段滤波处理，再把它们衔接起来而不影响全信号的滤波效果。下面例子可说明这个问题。

【例 5.9】 设滤波器系统函数为 $H(z)=\frac{1+0.5z^{-1}}{0.5+0.3z^{-1}}$，今有随机长信号，验证分段滤波效果。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 5-8
num=[1 0.5];
den=[0.5 0.3];
x=rand(100,1);
y=filter(num,den,x);
x1=x(1:25);
x2=x(26:50);
x3=x(51:75);
```

```
x4=x(76:100);
[y1,Zf1]=filter(num,den,x1);      Zi2=Zf1;
[y2,Zf2]=filter(num,den,x2,Zi2); Zi3=Zf2;
[y3,Zf3]=filter(num,den,x3,Zi3); Zi4=Zf3;
[y4,Zf4]=filter(num,den,x4,Zi4);
y1234=[y1;y2;y3;y4];
nn=0:length(y)-1;
plot(nn,y,'r-',nn,y1234,'b-');
xlabel('n');ylabel('y(n)-y1234(n)');
```

程序运行结果如图 5.22 所示。

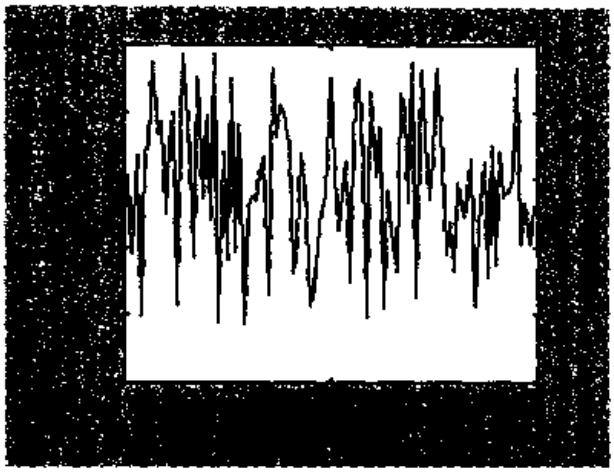


图 5.22　信号分段滤波

由图 5.22 可见，信号分段滤波和全程滤波完全吻合，值得注意的是分段滤波必须采用函数 filtic。

(3) 函数 FFTFILT 利用效率高的基于 FFT 重叠相加法算法实现对数据滤波，该函数只适用于 FIR 滤波器，函数调用格式为：

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

式中， b 为 FIR 滤波器系数向量；x 为输入数据；n 为 FFT 长度，缺省时，函数选择最佳的 FFT 长度；y 为滤波器输出。

该函数在频域内实现 FIR 滤波器，采用下面 FFT 过程

```
n =length(x);
y =ifft(fft(x). * fft(b, n). /fft(a, n))
```

y = fftfilt(b, x)等价于 y = filter(b, 1, x)。

【例 5.10】　FIR 低通滤波器截止频率为 200Hz，采样频率 F_s=1000Hz，对信号 $x(t)=\sin(2\pi f_1 t)+\sin(2\pi f_2 t)$滤波，$f_1$=50Hz，$f_2$=250Hz，求滤波器输出。

用 MATLAB 编制程序，首先用函数 firl 产生截止频率为 200Hz 的 FIR 低通滤波器，再用 fftfilt 对信号 x 进行滤波。程序清单如下：

```
%MATLAB PROGRAM 5-9
```

```
clf
%Create Orignal signal data
N=1000;
Fs=1000;
n=[0:N-1];
t=n/Fs;
x=sin(2*pi*50*t)+sin(2*pi*250*t);
%Create a FIR filter with lawpass
b=fir1(40,200/500);
%Filtering the data
yfft=fftfilt(b,x,256);
%Output
n1=81:241;
t1=t(n1);
x1=x(n1);
y1=yfft(n1);
subplot(221)
plot(t1,x1);
title('Original Signal')
subplot(222);
plot(t1,y1);
title('Signal after the filter')
grid
```

程序运行结果显示，通过低频滤波器后的信号只包含 50Hz 单一频率的正弦波，如图 5.23 所示。

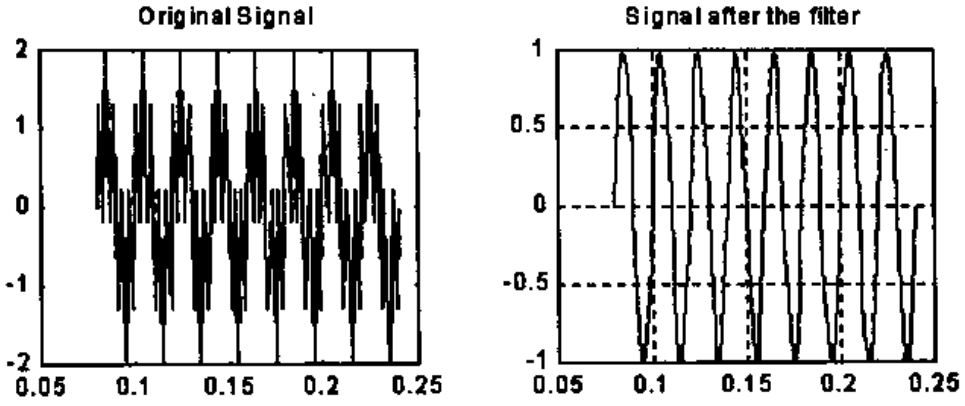


图 5.23 信号滤波

(4) 函数 IMPZ 用于产生数字滤波器的脉冲响应。调用格式为：

[h, t] = impz(b, a)

```
[h, t] = impz(b, a, n)
[h, t] = impz(b, a, n, Fs)
impz(b, a)
impz(…)
```

其中， b，a 分别是滤波器分子和分母多项式系数向量；n 为采样点数；Fs 为采样周期，缺省值=1；h 为滤波器单位脉冲响应向量；t 为和 h 对应的时间向量。

当函数输出缺省时，绘制滤波器脉冲响应图；当 n 缺省时，函数自动选择 n 值。

上述函数不仅适用于滤波器，也适用于其他离散时间系统时间，但模型应具有 DSP 形式。

【例 5.11】 设计一个 10 阶的 Butterworth 带通滤波器，通带为 100～250Hz，采样频率为 1000Hz，绘出滤波器的单位脉冲响应。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 5-10
N=101;
Fs=1000;
%Create a filter with bandpass
[b,a]=butter(10,2*[100 250]/Fs);
%Impulse Response
impz(b,a,N,Fs);
title('Impulse Response')
grid
```

程序运行结果如图 5.24 所示。

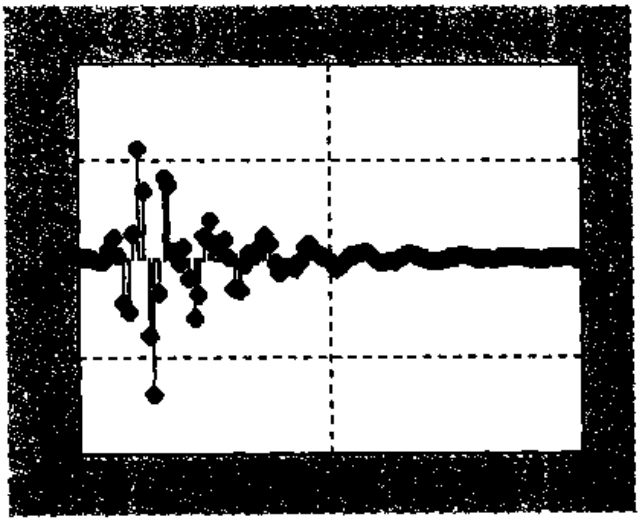


图 5.24 脉冲响应

在 MATLAB 中，h=impz(b,a,n)等价于下面语句：

```
x = [1 zeros(1,n)];
h = filter(b, a, x);
```

因此，在不少场合下，用后一种方法求滤波器的脉冲响应更为灵活。下面给出一个例子。

【例 5.12】 设计一个 8 阶低通 FIR 数字滤波器，截止频率为 0.6，试给出它的直接型和级联型的单位脉冲响应图。

用 MATLAB 编写程序如下

```
%MATLAB PROGRAM 5-11
clf
x=[1,zeros(1,40)];
n=[0:length(x)-1];
%Original Model of the filter
%==========
num=fir1(8,0.6);
den=1;
%Impulse Response using FILTER
y=filter(num,den,x);
subplot(221)
plot(n,y);
grid
title('Impulse Response')
legend('Direct Form')
%Impulse Response using FFTFILT
y3=fftfilt(num,x);
subplot(222)
plot(n,y3);
grid
legend('FFT Implementation')
title('Impulse Response')

%Convert to Cascade form of filter
%==========
z=roots(num);
k=num(1);
p=0;
sos=zp2sos(z,p,k)
%Check impulse response
num11=conv(sos(1,1:3), sos(2,1:3));
num12=conv(sos(3,1:3), sos(4,1:3));
num1=conv(num11,num12);
```

```
den11=conv(sos(1,4:6), sos(2,4:6));
den12=conv(sos(3,4:6), sos(4,4:6));
den1=conv(den11,den12);

y1=filter(num1,den1,x);
subplot(223)
plot(n,y1);
grid
legend('Cascade Form')
title('Impulse Response')
```

程序运行结果如图 5.25 所示。由图可见，利用函数 filter 和 fftfilt 计算直接型结构和级联型结构的 FIR 滤波器的脉冲响应相同。

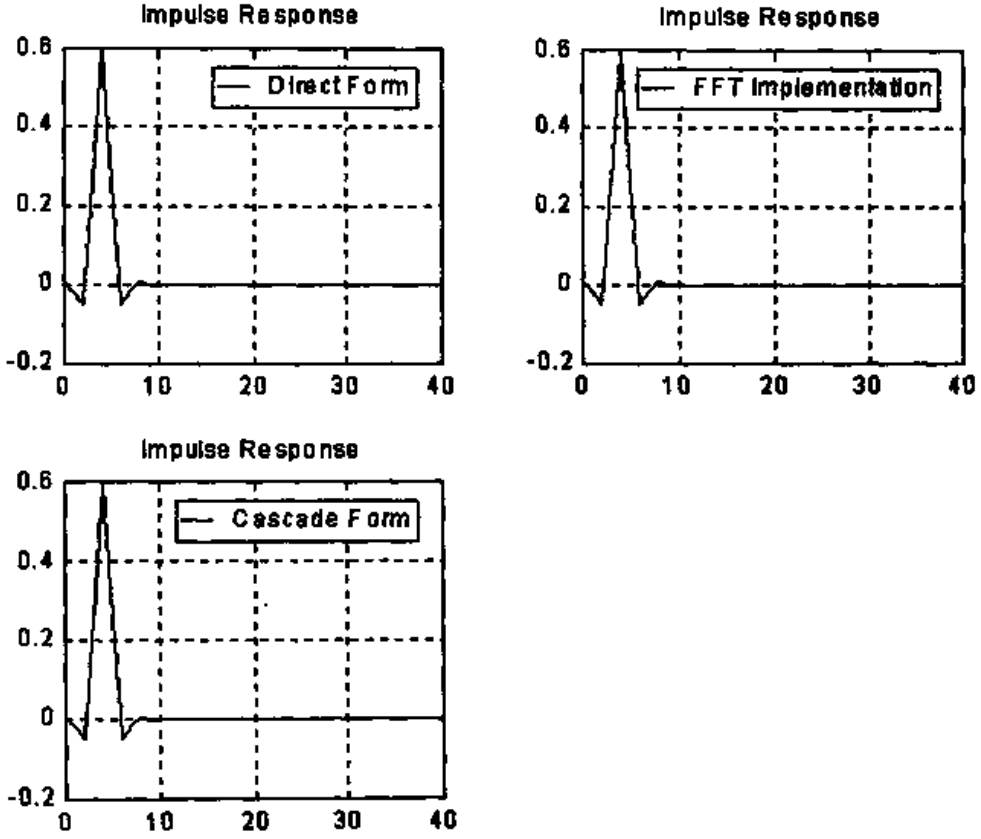


图 5.25　不同结构滤波器的脉冲响应

5.2.2　频率响应

在 1.5.3 和 2.5 节中我们讨论过系统的频率响应求解方法，这些方法均适用于滤波器。这里重点介绍 MATLAB 信号工具箱常用的滤波器频率响应，分析工具函数 FREQS、FREQZ 等。

(1) 函数 FREQS 用于求模拟滤波器的频率响应。调用格式为

$$h = freqs(b, a, \omega)$$

$$[h, \omega] = \text{freqs}(b, a)$$

$$[h, \omega] = \text{freqs}(b, a, n)$$

$$\text{freqs}(b, a)$$

其中，b，a 分别为模拟滤波器传递函数分子和分母多项式向量；n 为频率点数(整数)，缺省值=200；h 为频率响应，复数；ω 频率向量，实数。

函数输出项缺省时，绘制模拟滤波器的幅频响应图和相频响应图。

模拟滤波器传递函数形式为

$$H(s)=\frac{B(s)}{A(s)}=\frac{b(1)s^{nb}+b(2)s^{(nb-1)}+\cdots+b(nb+1)}{a(1)s^{na}+a(2)s^{(na-1)}+\cdots+a(na+1)}$$

函数输出 h 为模拟滤波器的频率响应 $H(e^{j\omega})$。

MATLAB 工具箱还提供两个函数 ABS 和 ANGLE，由频率响应 $H(e^{j\omega})$ 求幅频响应 $|H(e^{j\omega})|$ 和相频响应 $(H(e^{j\omega})$。

函数 ABS 用于求取复数幅值，调用格式为

$$y = \text{abs}(x)$$

若 x 为实数，则返回值 y 为绝对值；

若 x 为复数，则返回值 y 是复数 x 的模(幅值)；

若 x 为滤波器频率响应，则返回值 y 为滤波器幅频响应，可用下式将 y 化为 dB 单位

$$20\log10(y)\ \text{dB}$$

函数 ANGLE 用于求复数的相角，调用格式为

$$p = \text{angle}(x)$$

若 x 为滤波器频率响应，则返回值 y 为滤波器的相频响应，单位：弧度(rad)，可用下式将其化为角度(degrees)

$$P * 180/\pi\ (\text{degrees})$$

函数输出向量 ω(弧度)可由下式化为 Hz

$$f = \omega/(2 * \pi)\ (\text{Hz})$$

【例 5.13】 已知模拟滤波器的系统函数

$$H(s)=\frac{0.2s^2+0.3s+1}{s^2+0.4s+1}$$

绘制滤波器的幅频响应和相频响应。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 5-12
clf
%Model of a filter
%========
b=[0.2 0.3 1];
a=[1 0.4 1];
%Output method 1
%========
```

```
freqs(b,a)
pause
%Output method 2
%========
clf
[h,w]=freqs(b,a);
mag=abs(h);
pha=angle(h);
subplot(211)
loglog(w,mag);
grid
xlabel('Frequency');
ylabel('Magnitude');
subplot(212)
semilogx(w,pha * 180/pi)
grid
xlabel('Frequency');
ylabel('Phase(degrees)');
```

程序运行结果如图 5.26 所示。程序中采用两种不同方法结果完全相同。

(2) 函数 FREQZ 用于求数字滤波器的频率响应。调用格式

```
[h,ω] = freqz(b, a, n)
[h, f] = freqz(b, a, n, Fs)
[h,ω] = freqz(b, a, n,'whole')
[h, f] = freqz(b, a, n,'whole', Fs)
h = freqz(b, a, ω)
h = freqz(b, a, f, Fs)
freq(b, a)
```

式中，

函数输入： b，a 分别为数字滤波器 z 传递函数分子和分母多项式系数向量；n 为复频率响应计算点数，整数，最好为 2 的幂，缺省值为 512；Fs 为采样频率，Hz；f 为给定的频率矢量；'whole'表示返回的 ω 值为包含 z 平面整个单位园频率矢量，即 0～2π；缺省时，ω 仅为包含 z 平面上半单位圆(0～π)之间等间距 N 个点频率矢量。

函数返回： h 为复频率响应；ω 为 n 点频率向量(弧度)，返回 ω 范围与输入参量'whole'有关；f 为 n 点频率向量(Hz)。

返回值缺省时，绘制幅频响应和相频响应图。

该函数适用于下面形式的数字滤波器

$$H(z)=\frac{B(z)}{A(z)}=\frac{b(1)+b(2)z-1+\cdots+b(nb+1)z^{-nb}}{a(1)+a(2)z^{-1}+\cdots+a(na+1)z^{-na}}$$

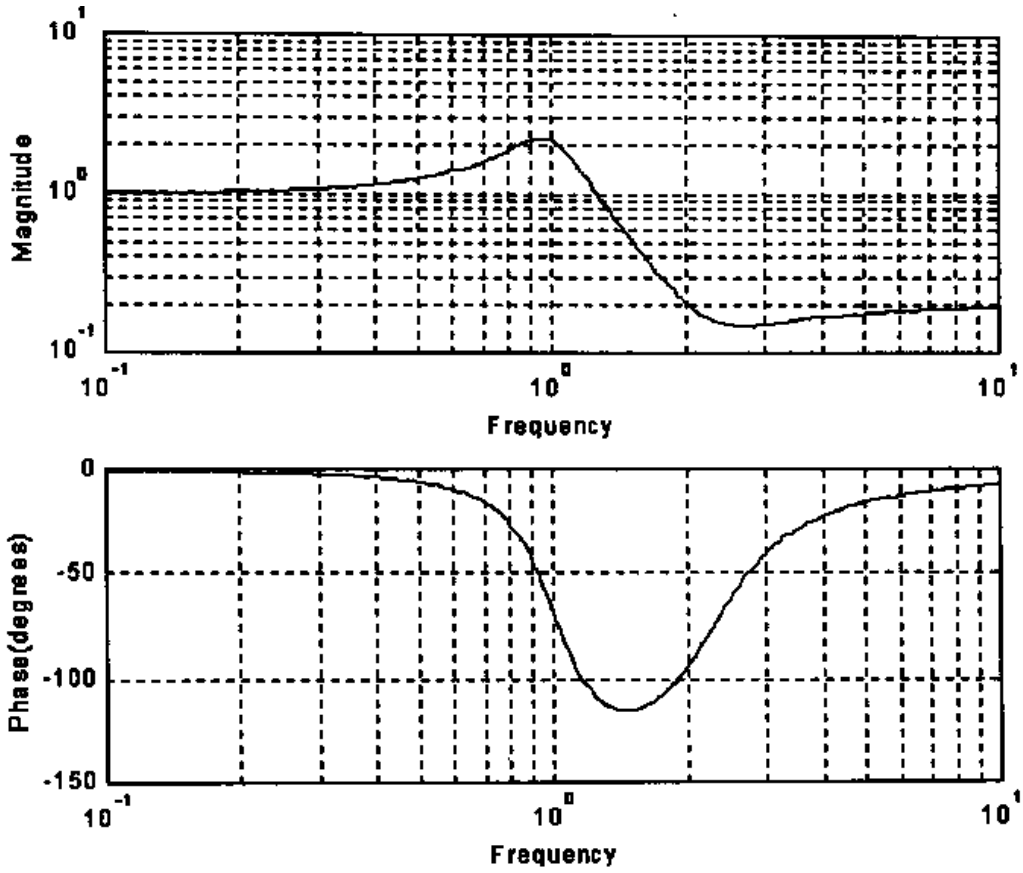


图 5.26　模拟滤波器的频率响应

该函数执行补零的 FFT 算法，即

$$h=fft(b,n./fft(a,n)$$

函数 FREQZ 输出的频率向量 ω 在 0～2π 之间。为了获得一个滤波器真正的相频特性图，要对相位角 ω 进行修正。为此 MATLAB 信号处理工具箱提供一个函数UNWRAP。

(3) 函数 UNWRAP 用于展开由函数 FREQZ 产生的频率 ω，调用格式为

$$P = unωrap(ω)$$

【例 5.14】　设计一个 25 阶低通 FIR 滤波器，截止频率为 0.4，绘出数字滤波器的频率响应。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 5-13
%Create a 25th-oder lowpass FIR filter
%==================
h=fir1(25,0.4);
%Plot frenqency response
%==========
%Output method 1
clf
freqz(h,1,512);
pause
```

```
%Output method 2
clf
[H,w]=freqz(h,1,512);
magH=20*log10(abs(H));
phaH1=unwrap(angle(H));
phaH=180*phaH1/pi;
wnyq=w/pi;
subplot(211)
plot(wnyq,magH)
xlabel('Normalized frequency==1');
ylabel('Magnitude (dB)');
grid
subplot(212)
plot(wnyq,phaH)
xlabel('Normalized frequency==1');
ylabel('Phase(degrees)');
grid
```

程序运行结果如图 5.27 所示。程序中采用两种方法结果相同。注意到,图中频率轴采用标准化频率。关于 MATLAB 中标准化频率定义已在 2.5 节给出。

5.2.3 零极点图

滤波器零极点位置决定了滤波器稳定性和性能,因此,考察滤波器零极点位置是分析滤波器特性重要方面之一。

MATLAB 信号处理工具箱提供绘制线性离散时间系统(数字滤波器)零极点图的工具函数 ZPLANE。调用格式为

```
zplane(z, p)
zplane(b, a)
```

其中,z, p 分别为系统零、极点向量;b,a 为系统 z 传递函数分子、分母多项式系数向量。

【例 5.15】 设计一个 5 阶 Butterworth 低通滤波器,截止频率为 0.2,绘制其零极点图。

用 MATLAB 编写程序如下:

```
%MATLAB PROGRAM 5-14
%Create a 5th-order Butterworth lowpass filter
[z,p,k]=butter(5,0.2);
zplane(z,p)
```

程序运行结果如图 5.28 所示。

5.2.4 群延迟

由信号传输不失真条件,滤波器相频特性应该是一条经过原点直线,即

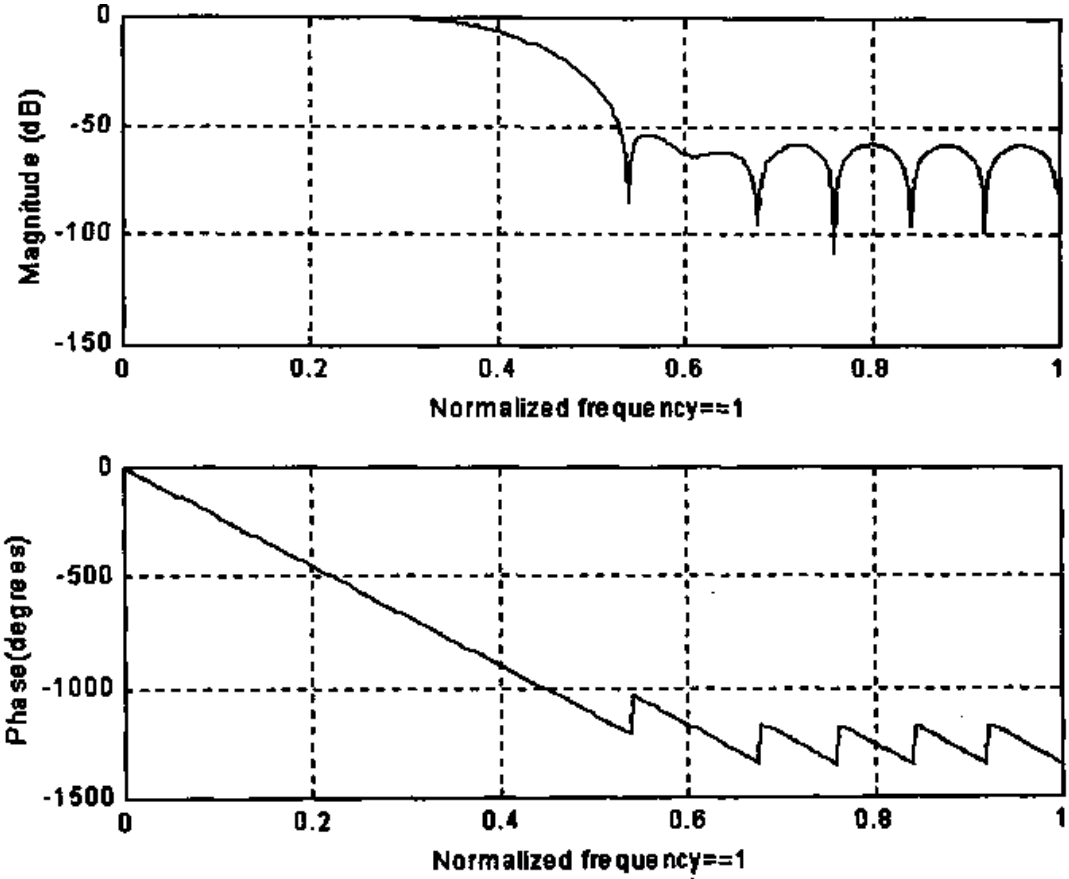


图 5.27　数字滤波器的频率响应

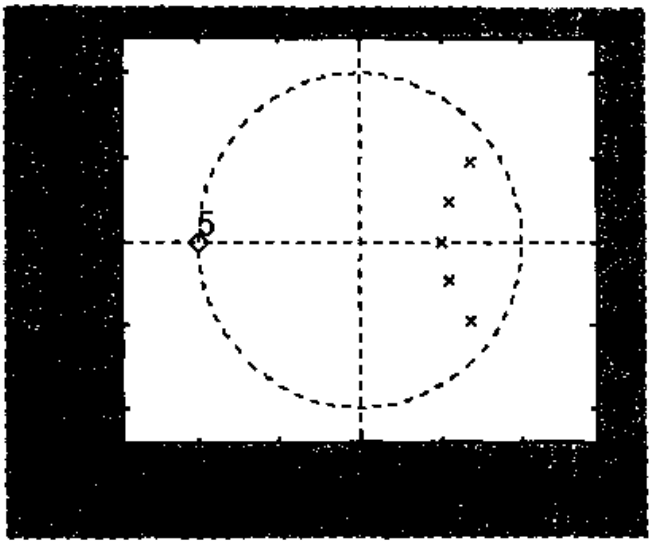


图 5.28　滤波器零极点图

$$\varphi(\omega) = -\omega t_d \tag{5.2-1}$$

即

$$\frac{\varphi(\omega)}{\omega} = -t_d$$

t_d 为常数。

但一般实际滤波器不满足上述条件，衡量实际滤波器相位平均延迟用群延迟。

群延迟定义为信号通过滤波器的平均延迟对频率的函数，即滤波器相频特性图上切线的负斜率

$$\tau_g(\omega) = -\frac{d\varphi(\omega)}{d\omega} \tag{5.2-2}$$

MATLAB 信号处理工具箱提供计算群延迟函数 GRPDELAY。调用格式为：

```
[gd, ω] = grpdelay(b, a, n)
[gd, f] = grpdelay(b, a, n, Fs)
[gd, ω] = grpdelay(b, a, n, 'whole')
[gd, f] = grpdelay(b, a, n, 'whole', Fs)
gd = grpdelay(b, a, ω)
gd = grpdelay(b, a, f, Fs)
grpdelay
```

其中，gd 为群延迟；其他各项意义同函数 freqz。函数输出项缺省时，绘制群延迟图。

【例 5.16】 绘制例 5.14 中滤波器的群延迟图。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 5-15
%Create a 5th-order Butterworth lowpass filter
[b,a]=butter(5,0.2);
grpdelay(b,a,128)
```

滤波器的群延迟如图 5.29 所示。

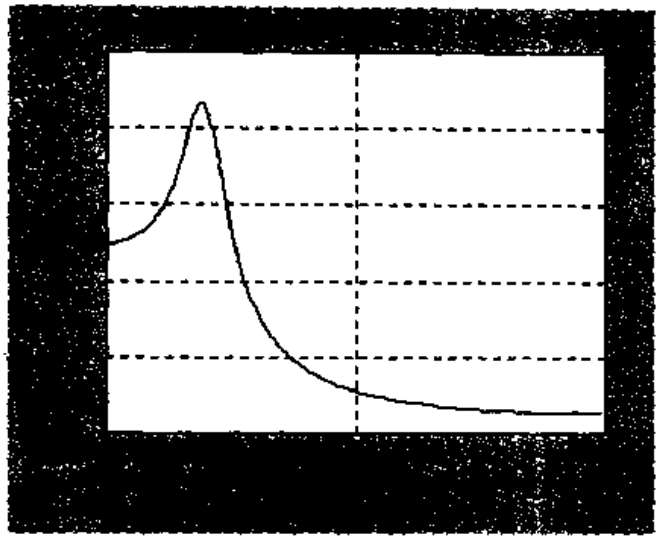


图 5.29 滤波器的群延迟

5.3 参数化建模

所谓参数化建模就是根据未知系统的某些信息（如脉冲响应、频率响应或输入输出序列）建立该系统的有理传递函数模型。这项技术用于求一个信号、系统或过程模型参数，并广泛用于语言分析、数据压缩、高分辨率谱估计、信号处理等领域。

本节主要涉及参数化建模在滤波器设计中和系统辨识的应用。关于参数化建模技术，

读者还可参考《系统分析与仿真》第六章的内容。

MATLAB 信号处理工具箱提供时域建模和频域建模两类参数化建模工具函数。

5.3.1 时域建模

时域建模就是给定系统时域内的信息，如脉冲响应或系统输入输出时间序列，求数字滤波器有理传递函数分子和分母的系数。

MATLAB 信号处理工具箱提供了三种时域建模函数。

一、线性预报法(AR一模型)

线性预报的基本原理是：一个信号 x 可认为是一个 n 阶自回归(AR)线性模型。它任一时刻 k 的采样值 x(k)是它的过去 n 个采样点值的线性组合(线性预报)，且这些系数是常数，则

$$x(k) = -a(2)x(k-1) - a(3)x(k-2)\cdots - a(n)x(k-n-1) - a(n+1)x(k-n) \tag{5.3-1}$$

这样，信号 x 就可用一个全极点 FIR 滤波器脉冲响应来逼近，全极点滤波器的传递函数具有如下形式：

$$H(z) = \frac{1}{1 + a(2)z^{-1} + \cdots + a(n+1)z^{-n}} \tag{5.3-2}$$

MATLAB 信号处理工具箱函数 LPC 利用自回归(AR)模型的相关法来求出 AR 模型系数 a，也是全极点滤波器模型。

函数 LPC 用来求和给定信号 x 相匹配的 AR 模型。调用格式为

[a, g]=lpc(x, n)

其中，x 为信号，实时间序列；n 为 AR 模型阶次；a 为 AR 模型系数，a=[1, a(2),…, a(n+1)]；g 为 AR 模型增益。

函数 LPC 的算法可简要地分为几个步骤：

(1) 调用函数 xcorr 求信号 x 的相关估计；

(2) 调用函数 levinson 实现 levinson-Durbin 递推算法如下式，求出系数 a。

$$\begin{bmatrix} 0 & 0 & \cdots & 0 \\ x(1) & 0 & \cdots & 0 \\ x(2) & x(1) & \ddots & \vdots \\ \vdots & \vdots & & 0 \\ x(1) & x(1-1) & \cdots & x(1) \\ 0 & x(1) & \ddots & \vdots \\ 0 & 0 & \cdots & x(1-1) \\ 0 & 0 & \cdots & x(1) \end{bmatrix} \begin{bmatrix} a(2) \\ a(3) \\ \vdots \\ a(n+2) \end{bmatrix} = \begin{bmatrix} -x(1) \\ -x(2) \\ \vdots \\ -x(1) \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

函数 LEVINOSN 调用格式为

a=levinson(r, n)

其中，r 为 x 的自相关序列；n 为 AR 模型阶次。

下面给出一个例子说明线性预报建模函数 lpc 的算法。

【例 5.17】 设信号 x 是一个带白噪声的 4 阶 FIR 滤波器的脉冲响应，试用线性预报

法建立其 AR 模型。

用 MATLAB 编程如下。

```
%MATLAB PROGRAM 5-16
%Linear Prediction Modeling
%Create signal-impulse response of an all-pole filter with white noise
randn('state',0);
x=impz(1,[1 0.1 0.1 0.1 0.1],10)+randn(10,1)/10;
%Method 1
r=xcorr(x);
r(1:length(x)-1)=[];
a=levinson(r,4)
%Method 2
a1=lpc(x,4)
```

```
    smp516
a =
    1.0000    0.2574    0.1666    0.1203    0.2598
a1 =
    1.0000    0.2574    0.1666    0.1203    0.2598
```

程序中用两种方法求得结果完全相同。若信号不包含随机分量，用上面程序求得信号 AR 模型和所采用全极点滤波器模型完全相同。这说明，如果一个信号 x 是 FIR 滤波器的脉冲响应，利用函数 lpc 建立的模型 a 为滤波器参数化模型。

二、Prony 法(ARMA 模型)

Prony 法用滤波器脉冲响应时间序列 h 建立 IIR 滤波器的模型。MATLAB 信号处理工具箱提供 Prony 法建模的函数 PRONY，调用格式为

$$[b, a]=prony(h, nb, na)$$

其中，h 为时域脉冲响应序列；nb，na 分别为滤波器分子和分母阶数；b，a 分别为滤波器分子和分母多项式系数向量；

滤波器传递函数表示为

$$H(z)=\frac{B(z)}{A(z)}=\frac{b(1)+b(2)z^{-1}+\cdots+b(nb+1)z^{-nb}}{a(1)+a(2)z^{-1}+\cdots+a(na+1)z^{-na}}$$

Prony 法是用一个 IIR 滤波器脉冲响应逼近信号 x。

【例 5.18】 建立一个 4 阶 butterworth 原始滤波器，求其脉冲响应，再用脉冲响应数据和函数 Prony 求滤波器模型并与原始模型进行比较。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 5-17
%Create a butterworth filter
[b,a]=butter(4,0.2)
```

```
%Find Impulse Response of the filter
h=impz(b,a,26);
[bb,aa]=prony(h,4,4)

≫ smp517
b =
    0.0048    0.0193    0.0289    0.0193    0.0048
a =
    1.0000   -2.3695    2.3140   -1.0547    0.1874
bb =
    0.0048    0.0193    0.0289    0.0193    0.0048
aa =
    1.0000   -2.3695    2.3140   -1.0547    0.1874
```

程序进行结果表明，用函数 Prony 建模和原始滤波器模型完全相同，显然函数 Prony 所建立的新滤波器的脉冲响应和信号 h 是完全吻合的。

三、Steiglity-McBride 法(ARMA 模型)

这种方法利用 Steiglity-McBride 迭代建模，使系统 b(z)/a(z)的脉冲响应 x′和输入信号 x 的均方差最小，即

$$\min_{a,b}\sum_{i=0}^{\infty}|x(i)-x'(i)|^2$$

MATLAB 信号处理工具箱函数 STMCB 利用这种方法建模，用于滤波器设计、系统辨识。下面的调用格式适用于系统的近似脉冲响应值为函数输入。

```
[b, a] =stmcb(x, nb, na)
[b, a] =stmcb(x, nb, na, niter)
[b, a] =stmcb(x, nb, na, niter, ai)
```

其中，x 为系统的脉冲响应；nb，na 分别为系统传递函数分子 b(z)/a(a)分子和分母阶次；niter 为迭代计算次数，缺省值=5；a_i 为分母系数的初步估计，可缺省；a，b 分别为系统(IIR 型滤波器)分子和分母多项式系数向量。

系统(IIR 滤波器)具有下面形式

$$H(z)=\frac{H(z)}{A(a)}=\frac{b(1)+b(2)z^{-1}+\cdots+b(nb+1)^{-nb}}{a(1)+a(2)z^{-1}+\cdots+a(na+1)z^{-na}}$$

函数另一种调用格式适用于系统的输入和输出信号作为函数输入：

```
[b, a] =stmcb(x, u, nb, na)
[b, a] =stmcb(x, u, nb, na, niter)
[b, a] =tncb(x, ,u, nb, na, niter, ai)
```

其中，x 和 u 分别为系统(滤波器)输出和输入信号；其余同上。

【例 5.19】 建立一个 6 阶 Butterworth 滤波器，求其脉冲响应，并用脉冲响应数据和函数 stmcb 建立新滤波器模型，比较两个滤波器的频率特性。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 5-18
clf
%Create a butterworth filter
figure(1)
[b,a]=butter(4,0.2)
freqz(b,a,128);
%Find the Impulse response of the filter
h=filter(b,a,[ 1 zeros(1,100)]);
%Find a new filter by function STMCB
figure(2)
[bb,aa]=stmcb(h,4,4)
freqz(bb,aa,128)
```

```
≫ smp518
b =
    0.0048    0.0193    0.0289    0.0193    0.0048
a =
    1.0000   -2.3695    2.3140   -1.0547    0.1874
bb =
    0.0048    0.0193    0.0289    0.0193    0.0048
aa =
    1.0000   -2.3695    2.3140   -1.0547    0.1874
```

运行结果说明，函数 stmcb 从滤波器脉冲响应建立的模型和原滤波器模型完全相同。

为了比较这三种方法(lpc，Prony，stmcb)建模引起的信号误差，编写下面程序并比较结果。

```
%MATLAB PROGRAM 5-19
%Create a input signal
randn('seed',0);
x=impz(1,[1 0.1 0.1 0.1 0.1],10)+randn(10,1)/10;
%Parameter modeling
a1=lpc(x,3);
[b2,a2]=prony(x,3,3);
[b3,a3]=stmcb(x,3,3);
%Compute error
disp('Function LPC:')
err1=x-impz(1,a1,10);
sumerr1=sum(err1.^2)
```

```
disp('Finction PRONY:')
err2=x-impz(b2,a2,10);
sumerr2=sum(err2.^2)
disp('Function STMCB:')
err3=x-impz(b3,a3,10);
sumerr3=sum(err3.^2)

≫ smp519
Function LPC:
sumerr1 =
        0.1226
Finction PRONY:
sumerr2 =
        0.0834
Function STMCB:
sumerr3 =
        0.0182
```

程序运行结果表明,在相同阶数条件下,stmcb 精度最好,Prony 次之,lpc 较差。

5.3.2 频域建模

频域建模就是由滤波器(系统)的频率响应数据建立其模型。根据频率域类型分为两类:S 域,对于模拟滤波器;z 域,对于离散滤波器。

一、模拟滤波器

若模拟滤波器的传递函数为

$$H(s)=\frac{B(s)}{A(s)}\frac{b(1)s^{nb}+b(2)s^{(nb-1)}+\cdots+b(nb+1)}{a(1)s^{na}+a(2)s^{(na-1)}+\cdots+a(na+1)}$$

MATLAB 信号处理工具箱函数 invfreqs 用于由给定的频率响应数据求形如上式的模拟滤波器传递函数。调用格式为

```
[b, a] = invfreqs(h, ω, nb, na)
[b, a] = invfreqs(h, ω, nb, na, ωt)
[b, a] = invfreqs(h, ω, nb, na, ωt, niter)
```

其中,h 为复频率响应向量;ω 为对应的频率向量;nb, na 分别为模拟滤波器分子和分母阶次;ωt 为和 ω 同维数的权系数向量,调整对频率的拟合误差;niter 为迭代次数。

若已知复频率响应的幅值向量 mag 和相位向量 phase,则 h=mag. * exp(j * phase)

【例 5.20】 已知滤波器原始传递函数中

```
b =[1 2 3 2 3]
a =[1 2 3 2 1 4]
```

试求其频率响应,并利用频率响应数据求原始滤波器的传递函数,并比较其结果。

```
%MATLAB PROGRAM 5-20
```

```
disp('Original Model of the Filter:')
b=[1 2 3 2 3]
a=[1 2 3 2 1 4]
%Compute frequency response
[h,w]=freqs(b,a,64);
disp('Model from Function INVFREQS:')
[bb,aa]=invfreqs(h,w,4,5)
```

```
≫ smp520
Original Model of the Filter:
b =
    1     2     3     2     3
a =
    1     2     3     2     1     4
Model from Function INVFREQS:
bb =
    1.0000    2.0000    3.0000    2.0000    3.0000
aa =
    1.0000    2.0000    3.0000    2.0000    1.0000    4.0000
```

运行结果表明，由函数 invfreqs 从频率响应数据辨识的滤波器模型和原始模型完全相同。

二、数字滤波器

若数字滤波器的传递函数为

$$H(z)=\frac{B(z)}{A(z)}=\frac{b(1)+b(2)z^{-1}+\cdots+b(nb+1)z^{-nb}}{a(1)+a(2)z^{-1}+\cdots+a(na+1)z^{-na}}$$

MATLAB 信号处理工具箱函数 invfreqz 用于由给定的频率响应数据求形如上式的数字滤波传递函数。调用格式为

```
[b, a] = invfreqz(h, ω, nb, na)
[b, a] = invfreqz(h, ω, nb, na, ωt)
[b, a] = invfreqz(h, ω, nb, na, ωt, niter)
```

其中，h 为复频率响应向量；ω 为对应的频率向量；nb，na 分别为数字滤波器分子和分母阶数；ωt 为权系数向量，调整对频率的拟合误差。

函数 invfreqs 和 invfreqz 利用等误差方法由给定的频率响应数据来辨识最佳的模型，当 niter 缺省时，函数利用下面的算法：

$$\min_{b,a}\sum_{k=1}^{n}\omega t(k)\left|h(k)A(\omega(k))-B(\omega(k))\right|^2$$

这里，A(ω(k))和 B(ω(k))分别表示多项式 a，b 在频率 ω(k)处的傅里叶变换，n 为频率点数(h 和 w 的长度)。这种算法不能保证辨识模型稳定性。当定义迭代次数 niter 后，函数采

用预估的输出误差迭代算法：

$$\min_{b,a}\sum_{k=1}^{n}\omega t(k)\left|h(k)-\frac{A(\omega(k))}{A(\omega(k))}\right|^2$$

这种算法保证辨识模型的稳定性，且提高辨识模型精度。

函数 invfreqz 调用方法基本上和函数 invfreqs 相同。

【例 5.21】 已知原始滤波器为 4 阶 Butterworth 低通数字滤波器。用其频率响应数据产生一个 3 阶滤波器。试比较采用等误差和输出误差选代算法的结果。

用 MATLAB 编写程序如下：

```
%MATLAB PROGRAM 5-21
%Create the original butterworth lowpass filter
[b,a]=butter(4,0.4);
%Compute frequency response
[h,w]=freqz(b,a,64);
%Create new model by INVFREQZ
[bb,aa]=invfreqz(h,w,3,3);
%Create new model by INVFREQZ with a superior algorithm
wt=ones(size(w));
niter=30;
[bbb,aaa]=invfreqz(h,w,3,3,wt,niter);
%Output the results
[h1,w1]=freqz(bb,aa,w);
[h2,w2]=freqz(bbb,aaa,w);
plot(w/pi,abs(h),'r—',w1/pi,abs(h1),'g——',w2/pi,abs(h2),'b.');
legend('Original','Without Superior Algorithm','With Superior Algorithm');
grid
sumerr1=sum(abs(h—h1).^2)
sumerr2=sum(abs(h—h2).^2)

≫ smp521
sumerr1 =
        0.0200
sumerr2 =
        0.0096
```

程序运行结果表明，输出误差选代法(预估算法)具有较好的拟合精度。

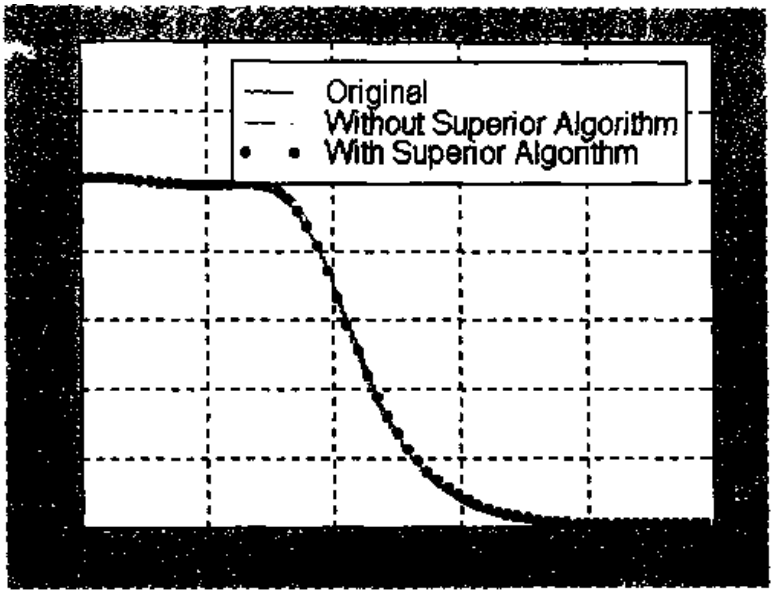


图 5.30　滤波器建模

习　题

5.1　已知 IIR 数字滤波器的系统函数为

$$H(z)=\frac{0.04658+0.1863z^{-1}+0.2795z^{-2}+0.1863z^{-3}+0.04658z^{-4}}{1-0.7821z^{-1}+0.68z^{-2}-0.1829z^{-3}+0.03012z^{-4}}$$

画出下面结构实现流图：

(1) 直接 I 型；

(2) 直接 II 型；

(3) 级联型；

(4) 并联型。

5.2　已知 FIR 数字滤波器的系统函数为

$$H(z)=0.0102=0.1177z^{-1}+0.3721z^{-2}+0.3721z^{-3}+0.1177z^{-4}+0.0102z^{-5}$$

画出下面结构实现信号流图：

(1) 直接型；

(2) 级联型；

(3) 线性相位型；

(4) 频率采样型。

5.3　画出 5.1 和 5.2 中滤波器的格形结构实现流图。

5.4　设计一个 7 阶低通巴特沃斯模拟滤波器，截止频率为 20π rad/s。试编写用该滤波器对长度为 2000 的高斯白噪声信号进行信号分段滤波和全信号滤波的 MATLAB 程序并比较结果。

5.5　设计一个 FIR 低通滤波器对信号 $x(t)=\sin 2\pi f_1 t+\sin 2\pi f_2 t+w(t)$ 进行数字滤波，$f_1=10$Hz，$f_2=50$Hz，要求除去信号中 40Hz 以上高频部分。试确定

(1) 滤波器传递函数；

(2) 滤波后信号图；

(3) 滤波后信号的频谱图。

5.6 已知一个滤波器系统函数为

$$H(z)=\frac{0.1321(1-3z^{-1}+3z^{-2}-z^{-3})}{1+0.3432z^{-1}+0.6043z^{-2}+0.2041z^{-3}}$$

试绘制滤波器的幅频响应和相频响应图，并指出该滤波器的类型和功能。

5.7 绘制 5.6 中滤波器的零极点图和群延迟图。

5.8 用 MATLAB 函数产生一个 5 阶低通 Chebyshev Ⅰ型数字滤波器(截止频率为 0.4)的脉冲响应序列，再用 Prony 法建模并与原滤波器模型进行比较。

5.9 用 MATLAB 函数产生一个 5 阶低通 Butterworth 数字滤波器(截止频率为 0.4)并计算其脉冲响应序列，再用 Steiglitz-McBride 法建模并与原滤波器模型进行比较。

5.10 用 MATLAB 设计一个 3 阶 Chebyshev Ⅰ型高通数字滤波器，设计指标：通带截止频率 fc>2.5kHz，通带波纹≤1dB，阻带衰减>20dB。试求：

(1) 高通数字滤波器系统函数；

(2) 滤波器的频率响应；

(3) 由频率响应数据辨识滤波器模型并与原模型进行比较。

第六章　随机信号分析

随机信号和确定性信号是两类性质完全不同的信号，对它们的描述、分析和处理方法也不相同。

随机信号是一种不能用确定的数学关系式来描述的，无法预测未来时刻精确值的信号，也无法用实验的方法重复再现。

随机信号分为平稳和非平稳两大类。平稳随机信号又分为各态历经和非各态历经。本章所讨论的随机信号是平稳的且是各态历经的。在研究无限长信号时，总是取某段有限长信号作分析。这一有限长信号称为一个样本(或称子集)，而无限长信号 x(t)称为随机信号总体(或称集)。各态历经的平稳随机过程中一个样本的时间均值和集平均值相等，因此一个样本统计特征代表随机信号的总体，使研究大大简化。工程上的随机信号一般均按各态历经平稳随机过程来处理。

仅在离散时间点上给出定义的随机信号称为离散时间随机信号，即随机信号序列。随机信号序列可以是连续随机信号的采样结果，也可以是自然界里实际存在的物理现象，它们本身就是离散的。

平稳随机过程在时间上是无始无终的，即它的能量是无限的，本身的傅里叶变换是不存在的；但功率是可能有限的，用功率谱密度函数来描述随机信号的频域特性，这是一个统计平均的频谱特性。

平稳随机过程统计特征的计算要求信号 x(n)无限长，而实际上只能用一个样本即有限长序列来计算。因此所得的计算值不是随机信号真正的统计值，而仅仅是一种估计。

本章介绍随机过程的统计特征计算、相关分析(时域)和功率谱估计(频域)的基本方法及 MATLAB 实现。

6.1　随机信号的数字特征

6.1.1　均值、均方值、方差

若连续随机信号 x(t)是各态历经的，则随机信号 x(t)均值可表示为

$$E[x(t)]=\mu_x=\lim_{T\to\infty}\frac{1}{T}\int_0^T x(t)dt \tag{6.1-1}$$

均值描述了随机信号的静态(直流)分量，它不随时间而变化。

随机信号 x(t)的均方值表达式为

$$\psi_x^2=\lim_{T\to\infty}\frac{1}{T}\int_0^T x^2(t)dt \tag{6.1-2}$$

ψ_x^2 表示信号的强度或功率。

随机信号 x(t)的均方根值，表示为

$$\psi_x=\sqrt{\lim_{T\to\infty}\frac{1}{T}\int_0^T x^2(t)dt} \tag{6.1-3}$$

ψ_x 也是信号能量的一种描述。

随机信号 x(t)的方差表达式为

$$E[(x-\mu_x)^2]=\sigma_x^2=\lim_{T\to\infty}\frac{1}{T}\int_0^T[x(t)-\mu_x]^2dt \tag{6.1-4}$$

σ_x^2 是信号的幅值相对于均值分散程度的一种表示，也是信号纯波动(交流)分量大小的反映。

随机信号 x(t)的均方差(标准差)可表示为

$$\sigma_x=\sqrt{\lim_{T\to\infty}\frac{1}{T}\int_0^T[x(t)-\mu_x]^2dt} \tag{6.1-5}$$

它和 σ_x^2 意义相同。

6.1.2 离散随机信号

若 x(n)是离散的各态历经的平稳随机信号序列，类似连续随机信号，则其数字特征可用下面式子来计算：

均值：
$$E[x(n)]=\mu_x=\lim_{N\to\infty}\frac{1}{N}\sum_{n=0}^{N}x(n) \tag{6.1-6}$$

均方值：
$$E[x(n)]=\psi_x^2=\lim_{N\to\infty}\frac{1}{N}\sum_{n=0}^{N}[x(n)]^2 \tag{6.1-7}$$

方差：
$$E[(x(n)-\mu_x)^2]=\sigma_x^2=\lim_{N\to\infty}\frac{1}{N}\sum_{n=0}^{N}[x(n)-\mu_x]^2 \tag{6.1-8}$$

其中，ψ_x 为均方根值，σ_x 为均方差值。

6.1.3 估计

以上计算都是对无限长信号而言。而工程实际中所取的信号是有限长的，计算中均无法取 T→∞或 N→∞。

如对于有限长模拟随机信号，计算均值时式(6.1-1)可改写为

$$E[x(t)]=\hat{\mu}_x=\frac{1}{T}\int_0^T x^2(t)dt \tag{6.1-9}$$

这里，均值 $\hat{\mu}_x$ 仅是一种对 μ_x 的估计。当 T 足够长，均值估计 $\hat{\mu}_x$ 能精确逼近真实均值 μ_x。对于周期信号，常取 T 为一个周期，估计均值 $\hat{\mu}_x$ 就能完全代表真实均值 μ_x。

对于有限长随机信号序列，计算其均值估计，可将式(6.1-6)改写为

$$E[x(n)]=\hat{\mu}_x=\frac{1}{N}\sum_{n=0}^{N}x(n) \tag{6.1-10}$$

当序列长度足够时，$\hat{\mu}_x$ 能精确逼近真实值 μ_x。

类似地，可以写出均方值和方差估计表达式。

在 MATLAB 工具箱中，没有专门函数用来计算均值、均方值和方差。但随机信号的统计数字特征值计算都可以通过 MATLAB 编程实现。在数值计算中，连续随机信号也离散化，当作随机序列来处理。

【例 6.1】 计算一长度 N=100000 的正态分布高斯随机信号的均值 μ_x、均方值 ψ_x^2、均方根值 ψ_x、方差 σ_x^2、均方差 σ_x。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 6-1
% Create random vector
N=100000;
y=randn(1,N);
%计算均值
disp('均值')
yMean=mean(y)
%计算均方值
disp('均方值')
y2p=y*y'/N
%计算均方根值
disp('均方根值')
ysq=sqrt(y2p)
%计算均方差(标准差)
disp('均方差(标准差)')
%ystd=sqrt(sum((y-yMean).^2)/(N-1))
ystd=std(y,1)
%计算方差
disp('方差')
yd=ystd.*ystd
```

```
≫ smp601
均值
yMean =
    0.0035
均方值
y2p =
    1.0029
均方根值
ysq =
    1.0015
均方差(标准差)
ystd =
    1.0015
方差
yd =
    1.0029
```

上例中，利用 MATLAB 工具箱函数 STD 求标准差。

函数 STD 调用格式为

```
s=std(x)
s=std(x,flag)
```

式中，x 为向量或矩阵；s 为标准差；flag 为控制符，用来控制标准算法。

当 flag=0(或缺省)时，按下式计算无偏标准差

$$s=\left[\frac{1}{n-1}\sum_{i=1}^{N}(x_i-\mu_x)^2\right]^{\frac{1}{2}}$$

当 flag=1 时，按下式计算有偏标准差

$$s=\left[\frac{1}{n}\sum_{i=1}^{N}(x_i-\mu_x)^2\right]^{\frac{1}{2}}$$

6.2 相关函数和协方差

6.2.1 自相关函数和自协方差

对于随机信号 x(t)，自相关函数为

$$R_x(\tau)=E[x(t)x(t+\tau)]=\lim_{T\to\infty}\frac{1}{T}\int_0^T x(t)x(t+\tau)dt$$

式中，τ 为时移。

若去掉 x(t)的均值部分，则相应的自相关函数称自协方差

$$C_x(t)=E\{[x(t)-\mu_x][x(t+\tau)-\mu_x]\}=R_x-\mu_x^2$$

$$C_x(0)=\sigma_x^2$$

对于离散随机信号序列，x(n)的自相关函数和自协方差为

$$R_x(m)=E[x(n)x(n+m)]=\lim_{N\to\infty}\frac{1}{N}\sum_{n=0}^{N-1}x(n)x(n+m)$$

$$C_x(m)=E\{[x(n)-\mu_x][x(n+m)-\mu_x]\}=R_x(m)-\mu_x^2$$

式中，m 为延迟。

6.2.2 互相关函数和互协方差

对于两个不同随机信号 x(t)，互相关函数为

$$R_{xy}(\tau)=E[x(t)y(t+\tau)]=\lim_{T\to\infty}\frac{1}{T}\int_0^T x(t)y(t+\tau)dt$$

互协方差为

$$C_{xy}(\tau)=E\{[x(t)-\mu_x][y(t+\tau)-\mu_y]\}=R_{xy}(\tau)-\mu_x\mu_y$$

$$R_{xy}(\tau)=R_{yx}(-\tau)$$

对于离散随机序列 x(n)和 y(n)，互相关函数和互协方差为

$$R_{xy}(m)=E[x(n)y(n+m)]=\lim_{N\to\infty}\frac{1}{N}\sum_{n=0}^{N-1}x(n)y(n+m)$$

$$C_{xy}(m)=E\{[x(n)-\mu_x][y(n+m)-\mu_y]\}=R_{xy}(m)-\mu_x\mu_y$$

注意到，在上面公式中 $t\to\infty$(或 $N\to\infty$)。而工程中信号是有限长，因此只能得到相

关函数和协方差的估计值，当 t(或 N)足够长时，估计值能精确地逼近真实值。

6.2.3 MATLAB 函数

MATLAB 信号处理工具箱提供了计算随机信号相关函数 XCORR 和协方差的函数 XCOV。

函数 XCORR 用于计算随机序列自相关和互相关函数。

调用格式为：

```
c=xcorr(x,y)
c=xcorr(x,y,'option')
c=xcorr(x,y,maxlags,'option')
[c,lags]=xcorr(x,y,maxlags,'option')
```

式中，x,y 为两个独立的随机信号序列，长度均为 N 向量；c 为 x,y 的互相关函数估计。

option 缺省时，函数 xcorr 按下式执行非归一化计算行相关

$$c_{xy}=\begin{cases}\sum\limits_{n=0}^{N-|m|-1} x_{n+1}y^*_{n+m+1}, & m\geqslant 0\\ c^*_{yx}(-m), & m<0\end{cases}$$

option 为选择项：

① 'bised'，计算有偏互相关函数估计

$$c_{xy,biased}(m)=\frac{1}{N}c_{xy}(m)$$

② 'unbiased'，计算无偏互相关函数估计

$$c_{xy,unbiased}(m)=\frac{1}{N-|m|}c_{xy}(m)$$

③ 'coeff'，序列归一化，使零延迟的自相关函数为 1

④ 'none'，缺省情况

maxlags 为 x 和 y 之间的最大延迟，若该项缺省时，函数返回值 c 长度是 2N−1；若该项不缺省时，函数返回值 c 长度是 2*maxlags+1。

该函数也可用于求一个随机信号序列 x(n)的自相关函数，调用格式为：

```
c=xcorr(x)
c=xcorr(x,maxlags)
```

【例 6.2】 求带白噪声干扰正弦信号和白噪声信号的自相关函数并进行比较。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 6-2
clf
N=1000;
n=0:N-1;
Fs=500;
t=n/Fs;
Lag=100;
```

```
%signal x
x=sin(2 * pi * 10 * t)+0. 6 * randn(1,length(t));
[c,lags]=xcorr(x,Lag,'unbiased');
subplot(221)
plot(t,x)
xlabel('t');ylabel('x(t)');
title('Original signal x');
grid
subplot(222)
plot(lags/Fs,c)
xlabel('t');ylabel('Rx(t)');
title('Autocorrelation ');
grid
%signal x1
x1=randn(1,length(x));
[c,lags]=xcorr(x1,Lag,'unbiased');
subplot(223)
plot(t,x1)
xlabel('t');ylabel('x(t)');
title('Original signal x1');
grid
subplot(224)
plot(lags/Fs,c)
xlabel('t');ylabel('Rx1(t)');
title('Autocorrelation ');
grid
```

程序运行结果如图 6.1 所示。

由此例结果可见，含有周期成分和干扰噪声信号的自相关函数在 $\tau=0$ 时具有最大值，且在 τ 较大时仍具有明显周期性，其频率和周期成分的周期相同；而不含周期成分纯噪声信号在 $\tau=0$ 时也具有最大值，但在 τ 稍大时明显衰减至零。自相关函数的这一性质被用来识别随机信号中是否含有周期信号成分和它们的频率。

【例 6.3】 已知两个周期信号

$$x(t) = \sin(2\pi ft), y(t) = 0.5\sin(2\pi ft + 90°)$$

其中：f=10Hz，求互相关函数 $R_{xy}(\tau)$。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 6-3
clf
N=1000;
```

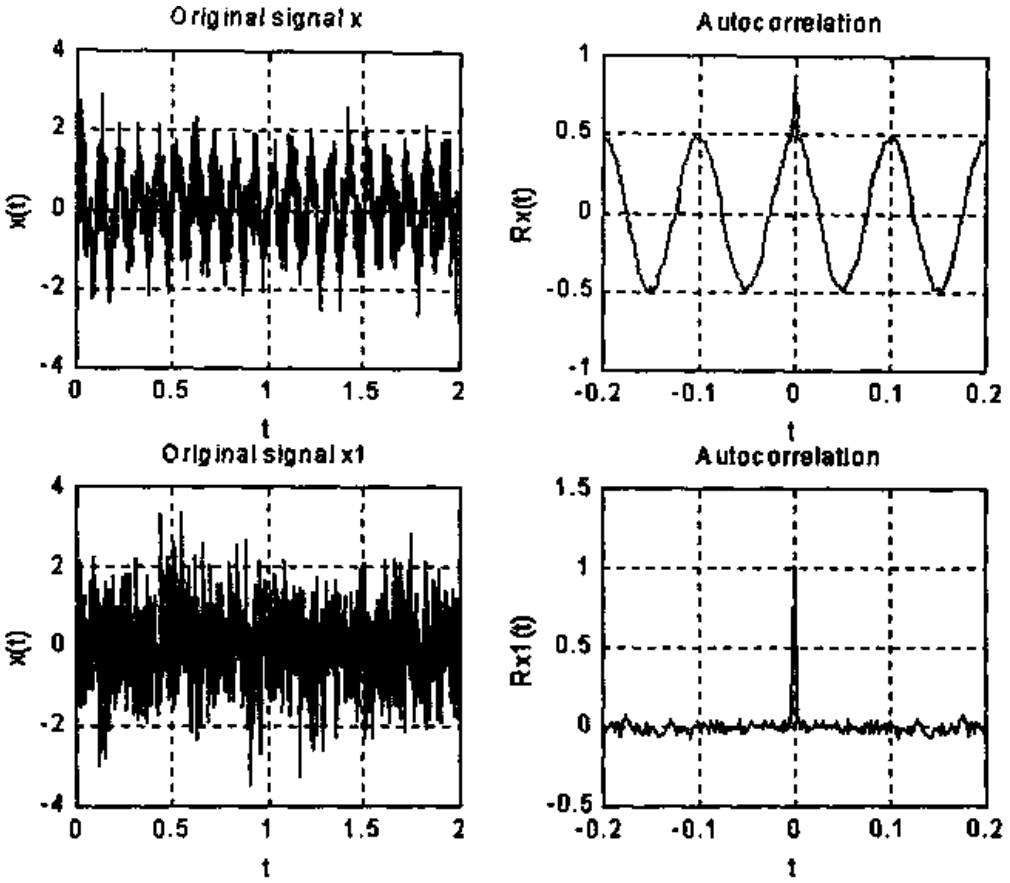


图 6.1 不同信号的自相关函数

```
n=0:N-1;
Fs=500;
t=n/Fs;
Lag=200;
%signal x and y
x=sin(2 * pi * 10 * t);
y=0.5 * sin(2 * pi * 10 * t+90 * pi/180);
[c,lags]=xcorr(x,y,Lag,'unbiased');
subplot(221)
plot(t,x,'r')
hold on
plot(t,y,'b')
xlabel('t');ylabel('x(t) y(t)');
title('Original signal ');
grid
hold off
subplot(222)
plot(lags/Fs,c',r');
```

```
xlabel('t');ylabel('Rxy(t)');
title('Correlation ');
grid
```

程序运行结果如图 6.2 所示，由图可见，$R_{xy}(\tau)$也是周期为 10Hz 余弦信号，幅值为 $1/2\times1\times0.5=0.25$，初始相位角为 90°。

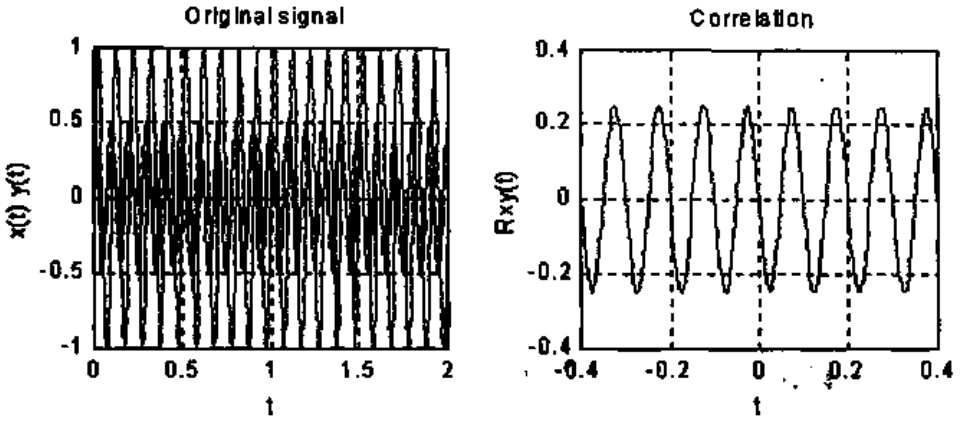


图 6.2 同频率周期信号互相关函数

由此可得到互相关函数一个重要特性：两个均值为零且具有相同频率的周期信号，其互相关函数 $R_{xy}(\tau)$保留原信号频率、相位差和幅值信息。

下面例子给出互相关函数的另一个工程应用实例。若信号 x(t)和信号 y(t)是同类信号且有时移，用互相关函数可以准确地计算出两个信号时移大小。这种特性使得互相关函数广泛地应用于工程测量技术中。

【例 6.4】 两个 sinc 信号有 0.2s 的时移用(MATLAB 程序产生)，用互相关函数计算时移大小。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 6-4
clf
N=1000;
n=0:N-1;
Fs=500;
t=n/Fs;
Lag=200;
%signal x1 y1
x1=90*sinc(pi*(n-0.1*Fs));
y1=50*sinc(pi*(n-0.3*Fs));
[c,lags]=xcorr(x1,y1,Lag,'unbiased');
subplot(221)
plot(t,x1,'r')
hold on
```

```
plot(t,y1,'b');
xlabel('t');ylabel('x(t) y(t)');
title('Original signal ');
grid
hold off
subplot(222)
plot(lags/Fs,c,'r');
xlabel('t');ylabel('Rxy(t)');
title('Correlation ');
grid
```

程序运行结果如图 6.3 所示，$R_{xy}(\tau)$的峰值出现在 0.2s 处，说明原信号 x(t)和 y(t)的时差为 0.2s。

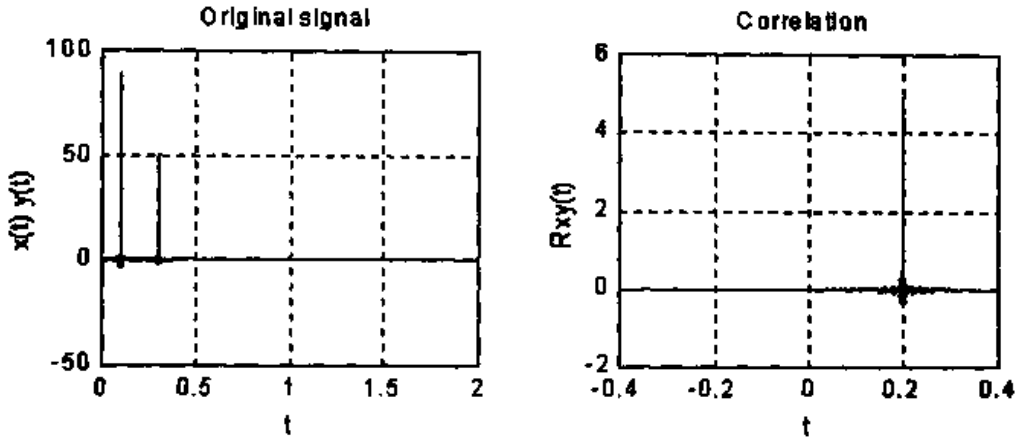


图 6.3 互相关函数的应用

6.3 功率谱估计

功率谱估计的目的是根据有限数据组寻找信号、随机过程或系统频率成分分布的描述。上节所述的相关分析是时域内在噪声背景下提取有用信息的途径，而功率谱是频域内提取在噪声淹没下信号的有用信息。

6.3.1 功率谱密度

假如随机信号 x(t)的自相关函数为 $R_x(\tau)$，$R_x(\tau)$的傅里叶变换为

$$S_x(f) = \int_{-\infty}^{\infty} R_x(\tau) e^{-j2\pi f\tau} d\tau \qquad (6.3\text{-}1)$$

则定义 $S_x(f)$为 x(t)的自功率谱密度或称为自功率谱，因为 $S_x(f)$可解释为 x(t)的平均功率相对频率的分布函数。自功率谱 $S_x(f)$包含 $R_x(\tau)$的全部信息。

若随机信号 x(t)的自功率谱为 $S_x(f)$，则

$$R_x(\tau) = \int_{-\infty}^{\infty} S_x(f) e^{j2\pi f\tau} df \qquad (6.3\text{-}2)$$

和自功率谱类似，两个随机信号 x(t)和 y(t)的相互关系的频率特性可用互功率谱密

度来描述，互功率谱密度和互相关函数也是一对傅里叶变换对：

$$S_{xy}(f) = \int_{-\infty}^{\infty} R_{xy}(\tau)e^{-j2\pi f\tau}d\tau \tag{6.3-3}$$

$$R_{xy}(\tau) = \int_{-\infty}^{\infty} S_{xy}(f)e^{j2\pi f\tau}df \tag{6.3-4}$$

对于离散随机序列 x(n)的自功率谱密度 $S_x(f)$和自相关函数 $R_x(m)$的关系为

$$S_x(f) = \sum_{m=-\infty}^{\infty} R_x(m)e^{-j2\pi fmT_s} \tag{6.3-5}$$

对于离散随机序列 x(n)和 y(n)，互功率谱密度 $S_{xy}(f)$和互相关函数 $R_{xy}(m)$关系为

$$S_{xy}(f) = \sum_{n=-\infty}^{\infty} R_{xy}(m)e^{-j2\pi fmT_s} \tag{6.3-6}$$

且

$$S_{xy}(-f) = S_{yx}(f) \tag{6.3-7}$$

$$S_x(f)S_y(f) \geqslant |S_{xy}(f)|^2 \tag{6.3-8}$$

实际工程中随机序列长度均为有限长，因此利用有限长随机序列计算的自功率谱密度和互功率谱密度只是真实值的估计。

功率谱估计方法一般可分为参数估计和非参数估计两类。MATLAB 信号处理工具箱介绍的功率谱密度非参数估计方法有 Welch 法、MTM 法和 MUSIC 法；参数估计方法有 MEM 法。

6.3.2 周期图法

一、基本方法

周期图法是直接将信号的采样数据 x(n)进行傅里叶变换求取功率谱密度估计。

假设有限长随机信号序列 x(n)。它的傅里叶变换和功率变密度估计 $\hat{S}_x(f)$存在下面关系

$$\hat{S}_x(f) = \frac{1}{N}|x(f)|^2$$

式中，N 为随机信号序列 x(n)长度，在离散的频率点 f(kΔf)，可得

$$\hat{S}_x(k) = \frac{1}{N}|X(k)|^2 = \frac{1}{N}|DFT[x(n)]|^2, \quad k = 0,1,\cdots N-1 \tag{6.3-9}$$

由于 DFT[x(n)]的周期为 N，求得的功率谱估计也是以 N 为周期，这种方法称为周期图法。

【例 6.5】 用 DFT 算法求信号 $x(t)=\sin(2\pi f_1 t)+2\sin(2\pi f_2 t)+\omega(t)$的功率谱。其中，$f_1=50Hz$，$f_2=120Hz$，ω(t)为白噪声，采样频率 Fs=1000Hz。

(1)信号长度 256；

(2)信号长度 1024。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 6-5
clf
Fs=1000;
```

```
%Case 1: N=256
%========
N=256;
Nfft=256;
n=0:N-1;
t=n/Fs;
xn=sin(2*pi*50*t)+2*sin(2*pi*120*t)+randn(1,N);
Pxx=10*log10(abs(fft(xn,Nfft).^2)/(N+1));
f=(0:length(Pxx)-1)*Fs/length(Pxx);
subplot(211)
plot(f,Pxx)
xlabel('Frequency (Hz)');
ylabel('Power Spectrum (dB)');
title('Periodogram N=256')
grid
%Case 2: N=1000
%=========
Fs=1000;
N=1024;
Nfft=1024;
n=0:N-1;
t=n/Fs;
xn=sin(2*pi*50*t)+2*sin(2*pi*120*t)+randn(1,N);
Pxx=10*log10(abs(fft(xn,Nfft).^2)/N);
f=(0:length(Pxx)-1)*Fs/length(Pxx);
subplot(212)
plot(f,Pxx)
xlabel('Frequency (Hz)');
ylabel('Power Spectrum (dB)');
title('Periodogram N=1024')
grid
```

程序运行结果如图 6.4 所示。

由图 6.4 可以看出，在频率 50Hz 和 120Hz 处功率谱有两个峰值，说明信号中含有 50Hz 和 120Hz 的周期成分。但功率谱密度都在很大范围内波动，而且并没有因信号取样点数 N 的增加而有明显改进。

用有限长样本序列的 DFT 来表示随机序列的功率谱虽然只是一种估计或近似，不可避免存在误差，为了减小误差，使功率谱估计更加平滑，可采用以下方法加以改进。

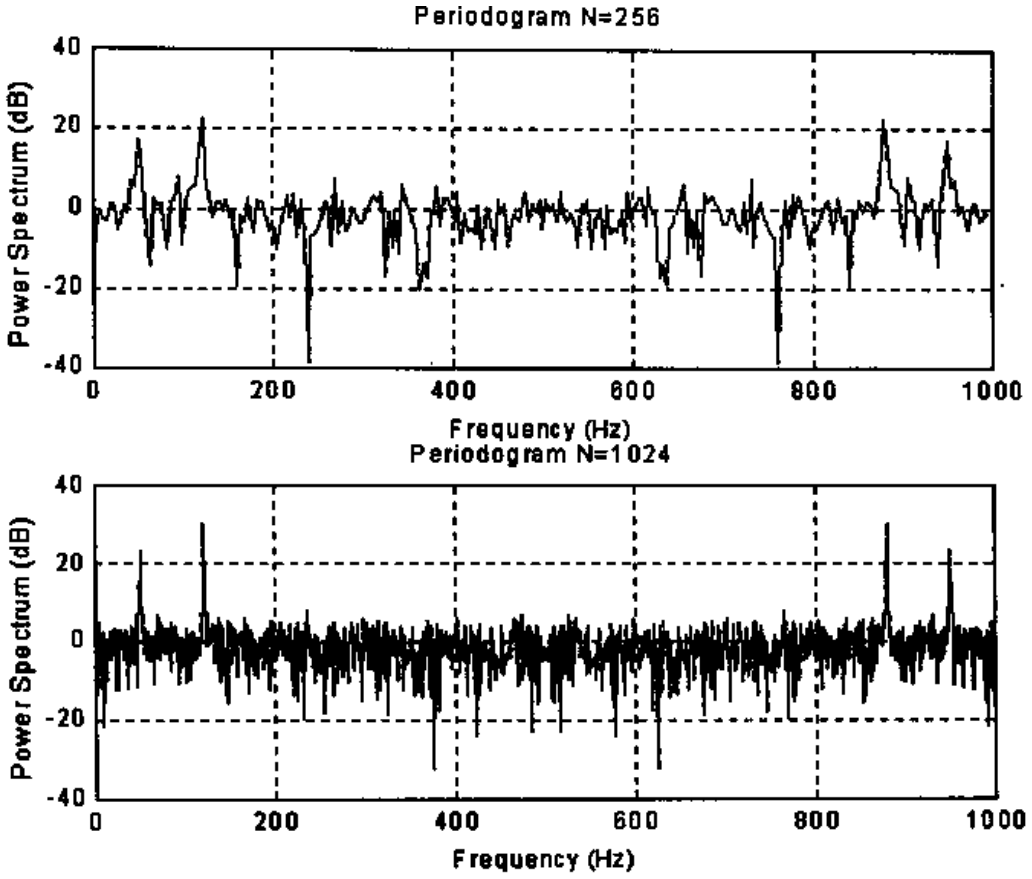


图 6.4 DFT 功率谱估计

二、分段平均周期图法

将信号序列 x(n),0≤n≤N−1,分成互不重叠的 P 个小段,每小段有 m 个采样值,则 Pm=N。对每小段信号序列进行功率谱估计,然后求它们的平均值作为整个序列 x(n)的功率谱估计。

平均周期图法还可以对信号 x(n)进行重叠分段,如按 2∶1 重叠分段,即前一段信号和后一段信号有一半是重叠的。对每一小段信号序列进行功率谱估计,然后再取平均值作为整个序列 x(n)的功率谱估计。

这两种都称为平均周期图法,一般后者较前者好。

【例 6.6】 如上例随机信号序列,用两种平均周期图法求信号的功率谱密度估计。

用 MATLAB 编写程序如下:

```
%MATLAB PROGRAM 6-6
clf
Fs=1000;
%Estimate PSD by the nonoverlapping signal section
%=========================
N=1024;
Nsec=256;
n=0:N-1;
```

```
t=n/Fs;
xn=sin(2*pi*50*t)+2*sin(2*pi*120*t)+randn(1,N);
pxx1=abs(fft(xn(1:256),Nsec).^2)/Nsec;
pxx2=abs(fft(xn(257:512),Nsec).^2)/Nsec;
pxx3=abs(fft(xn(513:768),Nsec).^2)/Nsec;
pxx4=abs(fft(xn(769:1024),Nsec).^2)/Nsec;
Pxx=10*log10((pxx1+pxx2+pxx3+pxx4)/4);
f=(0:length(Pxx)-1)*Fs/length(Pxx);
subplot(211)
plot(f,Pxx)
xlabel('Frequency (Hz)');
ylabel('Power Spectrum (dB)');
title('Averaged Periodogram (no overlap) N=4*256')
grid
%Estimate PSD by the half overlapping signal section
%=====================
Fs=1000;
N=1024;
Nsec=256;
n=0:N-1;
t=n/Fs;
xn=sin(2*pi*50*t)+2*sin(2*pi*120*t)+randn(1,N);
pxx1=abs(fft(xn(1:256),Nsec).^2)/Nsec;
pxx2=abs(fft(xn(129:384),Nsec).^2)/Nsec;
pxx3=abs(fft(xn(257:512),Nsec).^2)/Nsec;
pxx4=abs(fft(xn(385:640),Nsec).^2)/Nsec;
pxx5=abs(fft(xn(513:768),Nsec).^2)/Nsec;
pxx6=abs(fft(xn(641:896),Nsec).^2)/Nsec;
pxx7=abs(fft(xn(769:1024),Nsec).^2)/Nsec;
Pxx=10*log10((pxx1+pxx2+pxx3+pxx4+pxx5+pxx6+pxx7)/7);
f=(0:length(Pxx)-1)*Fs/length(Pxx);
subplot(212)
plot(f,Pxx)
xlabel('Frequency (Hz)');
ylabel('Power Spectrum (dB)');
title('Averaged Periodogram (half overlap) N=1024')
grid
```

程序运行结果如图 6.5 所示，上图采用不重叠分段法的功率谱估计，下图为 2：1 重

叠分段的功率谱估计，可见后者估计曲线较为平滑。和图 6.4 相比可见，平均周期图法功率谱估计具有明显效果。

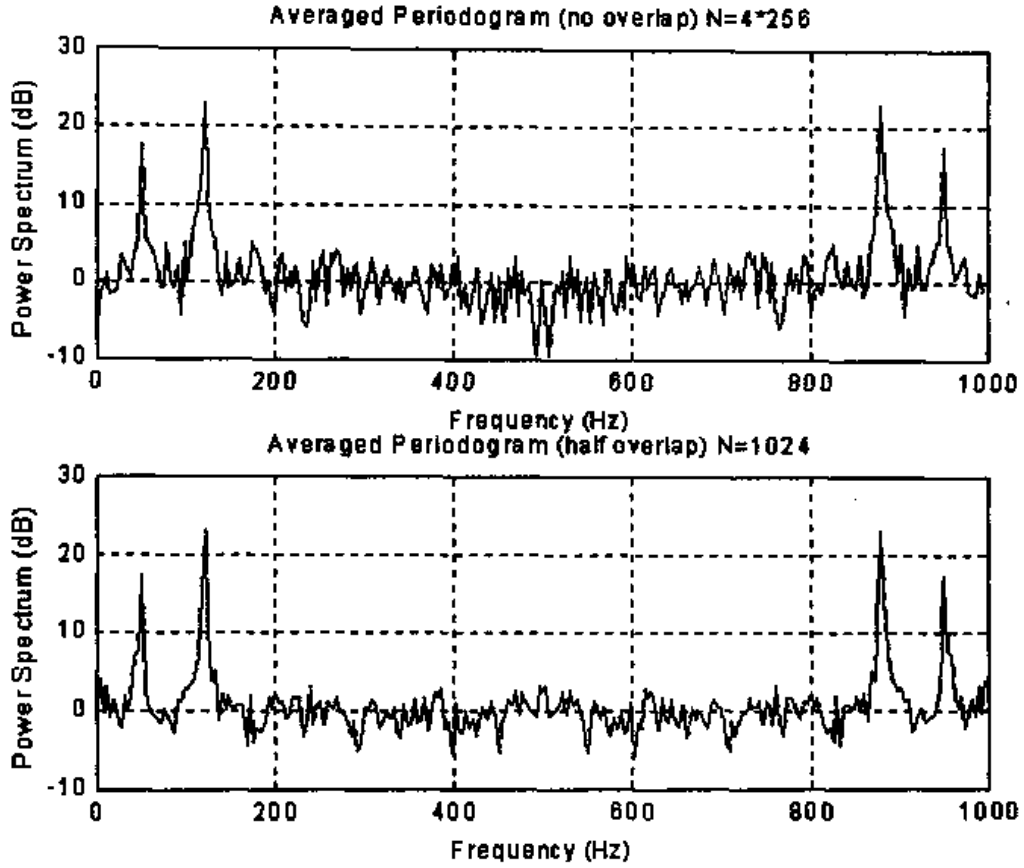


图 6.5 分段平均周期法功率谱估计

分段平均周期法功率谱估计可以减小估计误差和波动，但由于这种方法将长信号分段成短信号从而使谱分辨率下降。下面的加窗平均周期法可以克服这一不足。

三、加窗平均周期图法

加窗平均周期图法是对分段平均周期图法的改进。在信号序列 x(n)分段后，用非矩形窗口对每一小段信号序列进行预处理，再采用前述分段平均周期法进行整个信号序列 x(n)的功率谱估计。由窗函数基本知识可知，采用合适的非矩形窗口对信号进行处理可减小“频率泄漏”，同时可增加频峰的宽度，从而提高频谱分辨率。

【例 6.7】 对于例 6.5 的随机信号序列，用加窗平均周期图法进行功率谱密度估计。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 6-7
clf
Fs=1000;
%Estimate PSD by Averaged Modified periodogram
%with the nonoverlapping signal section
%==========================
N=1024;
```

```
Nsec=256;
n=0:N-1;
t=n/Fs;
w=hanning(256)';
xn=sin(2*pi*50*t)+2*sin(2*pi*120*t)+randn(1,N);
pxx1=abs(fft(w.*xn(1:256),Nsec).^2)/norm(w)^2;
pxx2=abs(fft(w.*xn(257:512),Nsec).^2)/norm(w)^2;
pxx3=abs(fft(w.*xn(513:768),Nsec).^2)/norm(w)^2;
pxx4=abs(fft(w.*xn(769:1024),Nsec).^2)/norm(w)^2;
Pxx=10*log10((pxx1+pxx2+pxx3+pxx4)/4);
f=(0:length(Pxx)-1)*Fs/length(Pxx);
subplot(211)
plot(f,Pxx)
xlabel('Frequency (Hz)');
ylabel('Power Spectrum (dB)');
title('Averaged Modified Periodogram (no overlap) N=4*256')
grid

%Estimate PSD by Averaged Modified periodogram
%with the half overlapping signal section
%==========================
Fs=1000;
N=1024;
Nsec=256;
n=0:N-1;
t=n/Fs;
w=hanning(256)';
xn=sin(2*pi*50*t)+2*sin(2*pi*120*t)+randn(1,N);
pxx1=abs(fft(w.*xn(1:256),Nsec).^2)/Nsec;
pxx2=abs(fft(w.*xn(129:384),Nsec).^2)/norm(w)^2;
pxx3=abs(fft(w.*xn(257:512),Nsec).^2)/norm(w)^2;
pxx4=abs(fft(w.*xn(385:640),Nsec).^2)/norm(w)^2;
pxx5=abs(fft(w.*xn(513:768),Nsec).^2)/norm(w)^2;
pxx6=abs(fft(w.*xn(641:896),Nsec).^2)/norm(w)^2;
pxx7=abs(fft(w.*xn(769:1024),Nsec).^2)/norm(w)^2;
Pxx=10*log10((pxx1+pxx2+pxx3+pxx4+pxx5+pxx6+pxx7)/7);
f=(0:length(Pxx)-1)*Fs/length(Pxx);
subplot(212)
```

```
plot(f,Pxx)
xlabel('Frequency (Hz)');
ylabel('Power Spectrum (dB)');
title('Averaged Modified Periodogram (half overlap) N=1024')
grid
```

程序运行结果如图 6.6 所示。

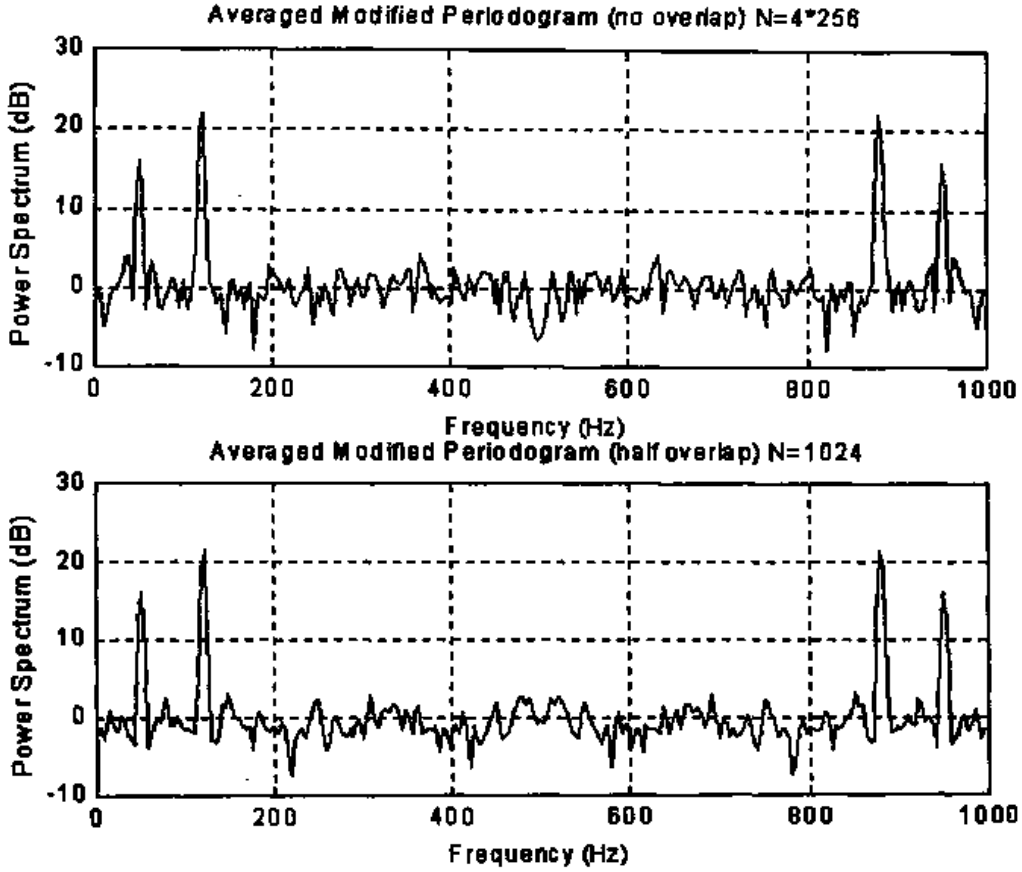


图 6.6　加窗平均周期图法功率谱估计

图 6.6 中的上图采用无重叠的数据分段、加窗平均周期图法，而下图采用重叠数据分段、加窗的平均周期图法进行功率谱估计，显然后者是最佳的，因此谱峰加宽，而噪声谱均在 0dB 附近，更为平坦。

四、Welch 法估计及 MATLAB 函数

Welch 功率谱密度就是用改进的平均周期图法来求取随机信号的功率谱密度估计的。Welch 法采用信号重叠分段、加窗函数和 FFT 算法等计算一个信号序列的自功率谱估计(PSD)和两个信号序列的互功率谱估计(CSD)。

MATLAB 信号处理工具箱提供专门函数 PSD 和 CSD 自动实现 Welch 法估计，而不需要像以上例子一样自己编程。

(1)函数 PSD 利用 welch 法估计一个信号自功率谱密度。函数调用格式为

P_{xx}=psd(x)

P_{xx}=psd(x, Nfft)

P_{xx}=psd(x, Nfft, F_s)

P_{xx}=psd(x, Nfft, F_s, window)

P_{xx}=psd(x, Nfft, F_s, window, Noverlap)

P_{xx}=psd(x,…, 'dflag')

其中,x 为信号序列; Nfft 为采用的 FFT 长度,这一值决定了功率谱估计速度,当 Nfft 采用 2 的幂时,快速执行;F_s 为采样频率;window 定义窗函数和 x 分段序列的长度,窗函数长度必须小于或等于 Nfft,否则会给出错误信息; Noverlap 为分段序列重叠的采样点数(长度),它应小于 Nfft;dflag 为去除信号趋势分量的选择项:

- linear,去除直线趋势分量
- mean,去除均值分量
- none,不作去除趋势处理

P_{xx}为信号 x 的自功率谱密度估计。

在 P_{xx}=psd(x)调用格式中,缺省值为:

- Nfft=min(256, length(x))
- F_s=2 Hz
- window=hanning(Nfft)
- noverlap=0

若 x 是实序列,函数 psd 仅计算频率为正的功率谱估计。

函数输出 P_{xx}长度:当 Nfft 为偶数时,Nfft/2+1;Nfft 为奇数时,(Nfft+1)/2。

若 x 是复序列,函数 psd 计算正负频率的功率谱估计。

函数的另一种调用格式为

[P_{xx},f]=psd(x, Nfft, Fs, window, Noverlap)

其中,f 为返回的频率向量,它和 P_{xx}对应,并且有相同长度。

利用函数 plot(f, P_{xx})可方便地绘出功率谱密度曲线。

【例 6.8】 对于例 6.5 中的随机信号序列,用函数 psd 绘制自功率谱密度估计曲线

用 MATLAB 编程如下:

```
%MATLAB PROGRAM 6-8
clf
%Estimate PSD by Averaged Modified periodogram
%Welch's Method
%==========================
N=1024;
Nfft=256;
Fs=1000;
n=0:N-1;
t=n/Fs;
window=hanning(256);
noverlap=128;
```

```
dflag='none';
xn=sin(2*pi*50*t)+2*sin(2*pi*120*t)+randn(1,N);
Pxx=psd(xn,Nfft,Fs,window,noverlap,dflag);
%==========
%Create frequency vector
%==========
f=(0:Nfft/2)*Fs/Nfft;
subplot(211)
plot(f,10*log10(Pxx))
xlabel('Frequency (Hz)');
ylabel('Power Spectrum (dB)');
title('PSD--Welch Method')
grid
%Estimate PSD by Averaged Modified periodogram
%Welch's Method
%======================
N=1024;
Nfft=256;
Fs=1000;
n=0:N-1;
t=n/Fs;
window=hanning(256);
noverlap=128;
dflag='none';
xn=sin(2*pi*50*t)+2*sin(2*pi*120*t)+randn(1,N);
[Pxx,f]=psd(xn,Nfft,Fs,window,noverlap,dflag);
subplot(212)
plot(f,10*log10(Pxx))
xlabel('Frequency (Hz)');
ylabel('Power Spectrum (dB)');
title('PSD--Welch Method')
grid
```

注意程序前半部分中频率向量 f 的创建方法。它与函数 psd 的输出 P_{xx} 长度有关。若 x 为实序列，当 Nfft 为奇数时 f=(0:(Nfft+1)/2-1)/Nfft；当 Nfft 为偶数时 f=(0:Nfft/2)/Nfft。程序运行结果如图 6.7 所示。

函数还有一种缺省返回值的调用格式，用于直接绘制信号序列 x 的功率谱估计曲线，调用格式为

```
psd(x,……)
```

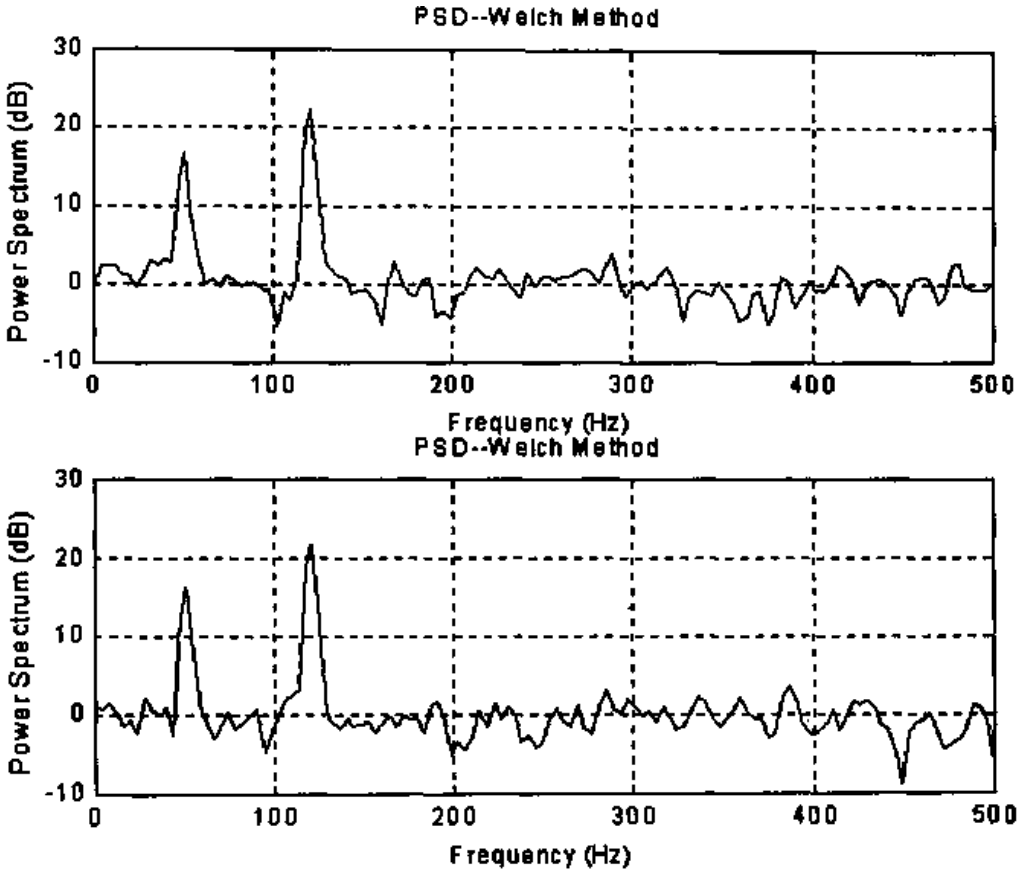


图 6.7 Welch 法功率谱估计

函数还可以计算带有置信区间的功率谱估计，调用格式为

$$[P_{xx}, P_{xxc}, f]=psd(x, Nfft, F_s, window, Noverlap, p)$$

其中，p 为置信度区间，$0 \leqslant p \leqslant 1$。

【例 6.9】 产生一个有色噪声信号并绘制置信区间为 0.95 的功率谱估计曲线。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 6-9
h=fir1(30,0.2,boxcar(31));
r=randn(16384,1);
x=filter(h,1,r);
psd(x,1024,10000,kaiser(512,5),0,0.95);
```

程序运行结果如图 6.8 所示。

(2)函数 CSD 利用 welch 法估计两个信号的互功率谱密度。函数调用格式为

$P_{xy}=csd(x, y)$

$P_{xy}=csd(x, y, Nfft)$

$P_{xy}=csd(x, y, Nfft, F_s, window)$

$P_{xy}=csd(x, y, Nfft, F_s, window, Noverlap)$

$P_{xy}=csd(x, y, \cdots, 'dflag')$

$[P_{xy}, f]=csd(x, y, Nfft, F_s, window, noverlap)$

csd (x,y,……)

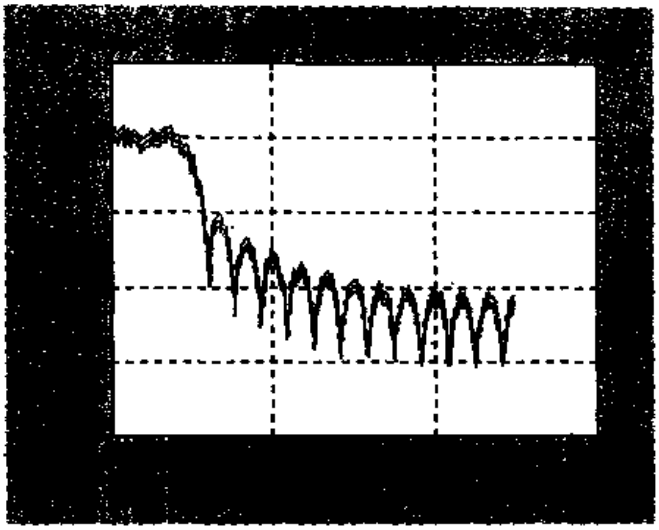


图 6.8　有置信区间的功率谱估计图

$[P_{xy}, Pxyc, f]$=csd(x, y, Nfft, F_s, window, noverlap, p)

其中，x，y 为两个信号序列；P_{xy}为 x，y 的互功率谱估计；其他参数的意义同自功率谱函数 psd。

函数 csd 的使用方法同函数 psd。

6.3.3　多窗口法

多窗口法(multitaper method 简称 MTM 法)利用多个正交窗口(tapers)获得各自独立的近似功率谱估计，然后综合这些估计最终得到的一个序列的功率谱估计。相对于普通的周期图法，这种功率谱估计具有更大的自由度，并在估计精度和估计波动方面均有较好的效果。普通的功率谱估计只利用单一窗口，因此在序列始端和末端均会丢失相关信息，而且无法找回。而 MTM 法估计增加窗口用来找回这些丢失的信息。

MTM 法简单地采用一个参数：时间带宽积(time-bandwidth product)NW，这个参数用以定义计算功率谱所用窗的数目为 2 * NW－1。如果 NW 愈大，功率谱计算次数愈多，而估计的波动愈小。然而，窗宽度与 NW 呈比例，因此随 NW 增大，每次估计具有更多泄漏，总功率谱估计的偏差增大。对于每一个数据组，通常有一个最优的 NW 使得在估计偏差和估计波动两方面求得一致。MATLAB 信号处理工具箱中函数 PMTM 就是采用 MTM 法估计功率谱密度。函数调用格式为

P_{xx}=pmtm(x)

P_{xx}=pmtm(x, nw)

P_{xx}=pmtm(x, nw, Nfft)

$[P_{xx}, f]$=pmtm(x, nw, Nfft, F_s)

……

其中，x 为信号序列；nw 为时间带宽积，缺省值=4，通常可取 2，5/2，3，7/2；Nfft 为 FFT 长度；F_s为采样频率。函数的基本用法和函数 psd 和 csd 相似，这里不再详细介绍。

pmtm(x, nw, Nfft, F_s, 'option', p)绘制带置信区间的功率谱密度估计曲线，0≤p≤1。

【例 6.9】 用多窗口法(MTM)估计例 6.5 随机信号序列功率谱密度。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 6-10
clf
%Estimate PSD by Multitaper Method
%==================
N=1024;
Nfft=256;
Fs=1000;
n=0:N-1;
t=n/Fs;
randn('state',0)
xn=sin(2*pi*50*t)+2*sin(2*pi*120*t)+randn(1,N);
[Pxx1,f]=pmtm(xn,4,Nfft,Fs);
subplot(211)
plot(f,10*log10(Pxx1))
xlabel('Frequency (Hz)');
ylabel('Power Spectrum (dB)');
title('Multitaper Method(MTM) nw=4')
grid

%Estimate PSD by Multitaper Method
%==================
N=1024;
Nfft=256;
Fs=1000;
n=0:N-1;
t=n/Fs;
randn('state',0)
xn=sin(2*pi*50*t)+2*sin(2*pi*120*t)+randn(1,N);
subplot(212)
pmtm(xn,4,Nfft,Fs,[],0.5);
```

程序运行结果如图 6.9 所示。

6.3.4 最大熵法

如上所述，周期图法功率谱估计需对信号序列“截断”或加窗处理，其结果是使估计的功率谱密度为随机信号序列的真实谱和窗谱的卷积，导致产生误差。

最大熵功率谱估计的目的是最大限度地保留截断后丢失的“窗口”以外的信号的信

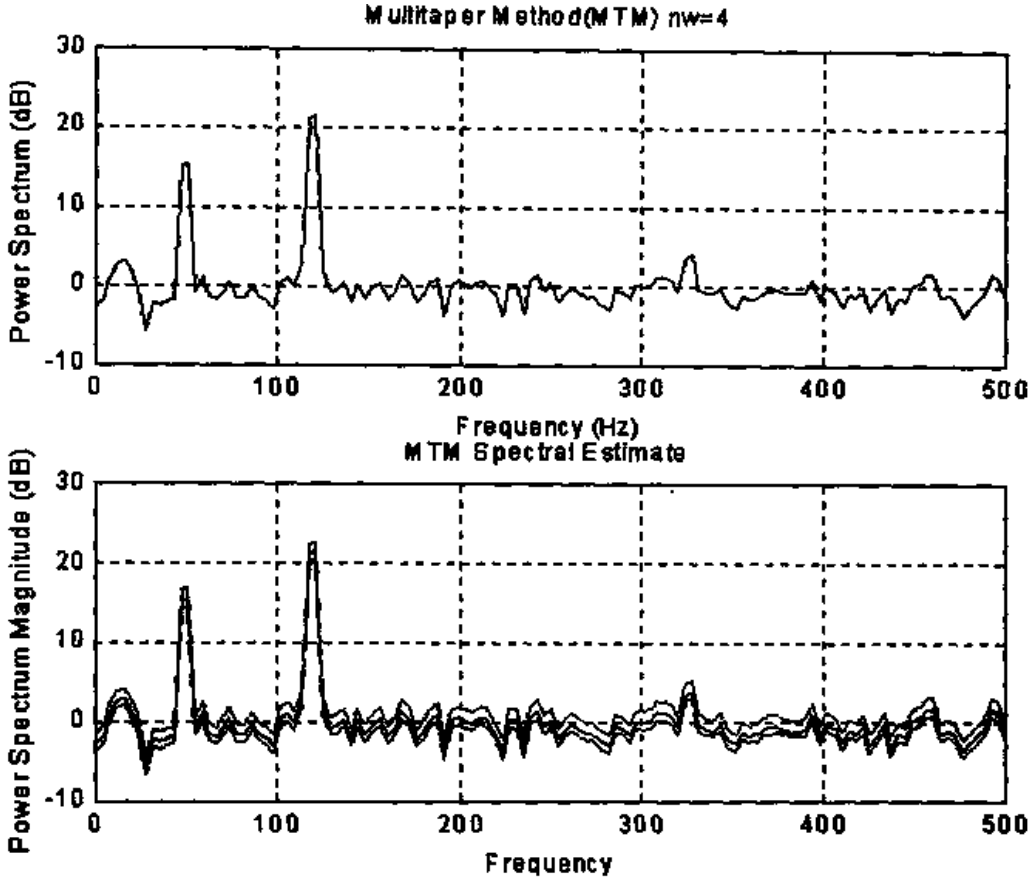


图 6.9 MTM 法功率谱估计

息，使估计谱的熵最大。主要方法是以已知的自相关序列 $r_{xx}(0)$，$r_{xx}(1)$，…，$r_{xx}(p)$为基础，外推自相关序列 $r_{xx}(p+1)$，$r_{xx}(p+2)$，…，保证信息熵最大。

最大熵功率谱估计法假定随机过程是平稳高斯过程，可以证明，随机信号的最大熵谱与 AR 自回归(全极点滤波器)模型谱和全极点线性预测谱是等价的。

MATLAB 信号处理工具箱提供最大熵功率谱估计函数 PMEM，其调用格式为

[Pxx, f]=pmem(x, p)

[Pxx, f]=pmem(x, p, Nfft, Fs,'corr')

[Pxx, f,a]=pmem(x, p, Nfft, Fs,'corr')

其中，x 为输入信号序列或输入相关矩阵；p 为全极点滤波器阶次；a 为全极点滤波器模型系数向量；'corr'把 x 认为是相关矩阵；其他参数和函数 psd 基本相同。

【例 6.11】 用 MEM 法计算例 6.5 信号功率谱估计。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 6-11
clf
%Estimate PSD by Maxmum Entropy Method(MEM)
%=========================
N=1023;
Nfft=256;
```

```
Fs=1000;
n=0:N-1;
t=n/Fs;
randn('state',0)
xn=sin(2*pi*50*t)+2*sin(2*pi*120*t)+randn(1,N);
[Pxx1,f]=pmem(xn,14,Nfft,Fs);
subplot(211)
plot(f,10*log10(Pxx1))
xlabel('Frequency (Hz)');
ylabel('Power Spectrum (dB)');
title('Maxmum Entropy Method(MEM) Oder=14')
grid

%Estimate PSD by Welch'method
%==========
N=1024;
Nfft=256;
Fs=1000;
n=0:N-1;
t=n/Fs;
randn('state',0)
xn=sin(2*pi*50*t)+2*sin(2*pi*120*t)+randn(1,N);
window=hanning(256);
noverlap=128;
dflag='none';
xn=sin(2*pi*50*t)+2*sin(2*pi*120*t)+randn(1,N);
subplot(212)
psd(xn,Nfft,Fs,window,noverlap,dflag);
```

程序运行结果如图 6.10 所示。比较最大熵功率谱估计(MEM)和改进的平均周期图功率谱估计(PSD),可见,MEM 法功率谱曲线较光滑。在这一例子中,MEM 法选定全极点滤波器的阶数 P 必须大于 10。

6.3.5 特征值向量法

MATLAB 信号处理工具箱还提供另一种功率谱估计函数 PMUSIC,该函数执行两种相关的频谱分析方法:

- 多信号分类法(MUSIC 法——Multiple Slgnal Classification)
- 特征向量法(EV 法——EigenVector)

这两种方法都是以自相关矩阵特征值分析为基础。我们可以把一个由复正弦和白噪声看成一个系统由自相关矩阵 R 来描述,则它由信号自相关矩阵和噪声自相关矩阵两部

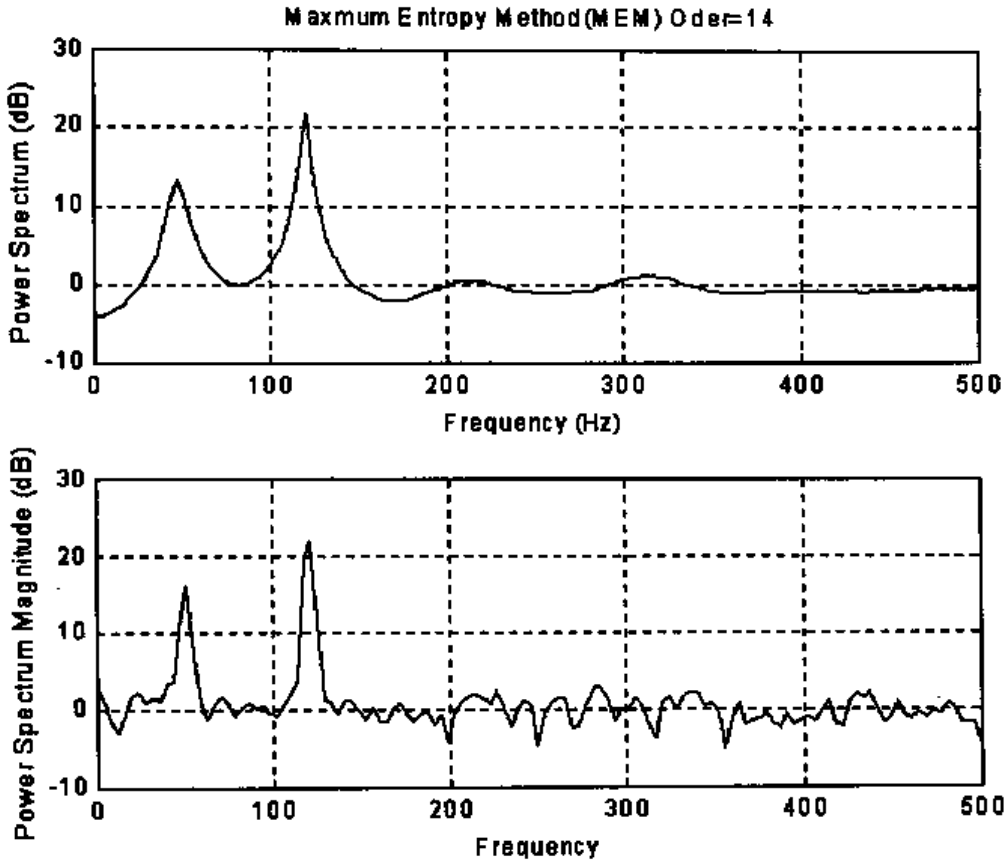


图 6.10　最大熵功率谱密度估计

分组成，即系统自相关矩阵 R 包含有两个子空间信息：信号子空间和噪声子空间。这样，矩阵特征值向量也可分为两个子空间：信号子空间和噪声子空间。为了求得功率谱估计，自相关矩阵特征值分析法 MUSIC 和 EV 法计算信号空间和噪声子空间的特征值向量函数，使得在正弦信号频率处函数值最大，功率谱估计出现峰值，而在其他频率处函数值最小，这种结果如其他的功率谱估计方法一样。

MUSIC 法和 EV 法原理完全相同。MUSIC 法按下式计算功率谱密度：

$$P_{music}(f)=\frac{1}{\sum_{k=p+1}^{N}|v_k^H e(f)|^2} \tag{6.3-10}$$

EV 法采用特征值加权算法，按下式计算功率谱密度：

$$P_{ev}(f)=\frac{1}{\sum_{k=p+1}^{N}|v_k^H e(f)|^2/\lambda_k} \tag{6.3-11}$$

式中，e(f)为复正弦向量，v 为输入信号自相关矩阵的特征向量；H 为共轭转置运算，λ_k 为特征值函数 PMUSIC 适用于用 MUSIC 法求信号的功率谱估计。调用格式为

[P_{xx}, f]= pmusic(x, p)

[P_{xx}, f]= pmusic(x, [p, thresh])

[P_{xx}, f,a]= pmusic(x, [p, thresh], Nfft, F_s, window, Noverlap)

其中，x 为输入信号，向量或矩阵；p 为信号子空间维数；thresh 为阈值；其他参数意义和

函数 psd 基本相同。

【例 6.12】 用 MUSIC 法估计例 6.5 信号的功率谱密度。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 6-12
clf
%Estimate PSD by MUSIC Method
%==========================
N=1024;
Nfft=256;
Fs=1000;
n=0:N-1;
t=n/Fs;
randn('state',0)
xn=sin(2*pi*100*t)+2*sin(2*pi*200*t)+randn(1,N);
subplot(211)
pmusic(xn,[7,1.1],Nfft,Fs,32,16);
```

程序运行结果如图 6.11 所示。

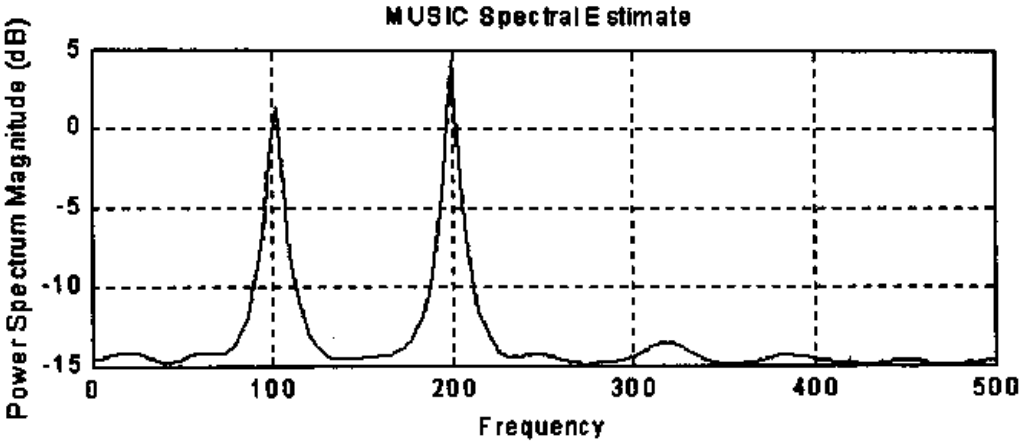


图 6.11 特征值向量法功率谱估计

值得注意的是，函数 pmusic 参数的选择对 MUSIC 法功率谱估计结果影响较大。

6.4 传递函数估计

功率谱密度估计可应用于估计系统传递函数的估计。

一个线性时不变系统，频率特性为 $H(\omega)$，$x(n)$和 $y(x)$分别为系统输入和输出。可以证明

$$P_{xy}(\omega) = H(\omega)P_{xx}(\omega) \tag{6.4-1}$$

式中，P_{xx}为 $x(n)$的自功率谱密度；P_{xy}为 $x(n)$和 $y(n)$的互功率谱密度。由此还可证明，输入 $x(n)$和输入 $y(n)$之间传递函数估计为

$$\hat{H}(\omega) = \frac{\hat{P}_{xy}(\omega)}{\hat{P}_{xx}(\omega)} \tag{6.4-2}$$

用 MATLAB 编程很容易实现上面的估计。当 x(n),y(n)已知时,由函数 PSD 求得 $\hat{P}_{xx}(\omega)$,由函数 CSD 求得 $\hat{P}_{xy}(\omega)$,再由式(6.4-2)可求得系统传递函数估计 $\hat{H}(\omega)$。MATLAB 信号处理工具箱还提供函数 TFE 可直接完成上述运算。

函数 TFE 用于从系统输入和输出的功率谱估计中求取传递函数估计。

函数调用格式为:

T_{xy}= tfe(x, y)

T_{xy}= tfe(x, y, Nfft)

[T_{xy},f] = tfe(x, y, Nfft, F_s)

[T_{xy},f] = tfe(x, y, Nfft, window, noverlap)

tfe(x, y)

式中,x 和 y 分别为系统输入和输出信号向量;T_{xy}为系统传递函数。

函数输入参数缺省值为

- Nfft=min(256, length(x))
- F_s=2
- window = hanning(Nfft)
- noverlap = 0

该函数的其他参数意义及调用格式与函数 CSD 相同。

【例 6.13】 用功率谱法估计一个 FIR 滤波器的频率响应并与实际响应比较。

用 MATLAB 编程如下:

```
%MATLAB PROGRAM 6-13
%Transfer Function Estimate
%Create input signal
N=1024;
Nfft=256;
window=hanning(256);
noverlap=128;
dflag='none';
Fs=1000;
n=0:N-1;
t=n/Fs;
randn('state',0)
xn=sin(2*pi*50*t)+randn(1,N);
%Create a system: filter
h=ones(1,10)/10;
%Compute output signal
yn=filter(h,1,xn);
%Estimate Transfer Function by 'etf'
[HEST,f]=tfe(xn,yn,Nfft,Fs,window,noverlap,dflag);
```

```
%Compute frequency response by 'freqz'
H=freqz(h,1,f,Fs);
%Compare the results
subplot(211)
plot(f,abs(H));
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('Frequency Response')
axis([0 500 0 1]);
grid
subplot(212)
plot(f,abs(HEST));
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('Transfer Function Estimate');
axis([0 500 0 1])
grid
```

程序运行结果如图 6.12 所示。

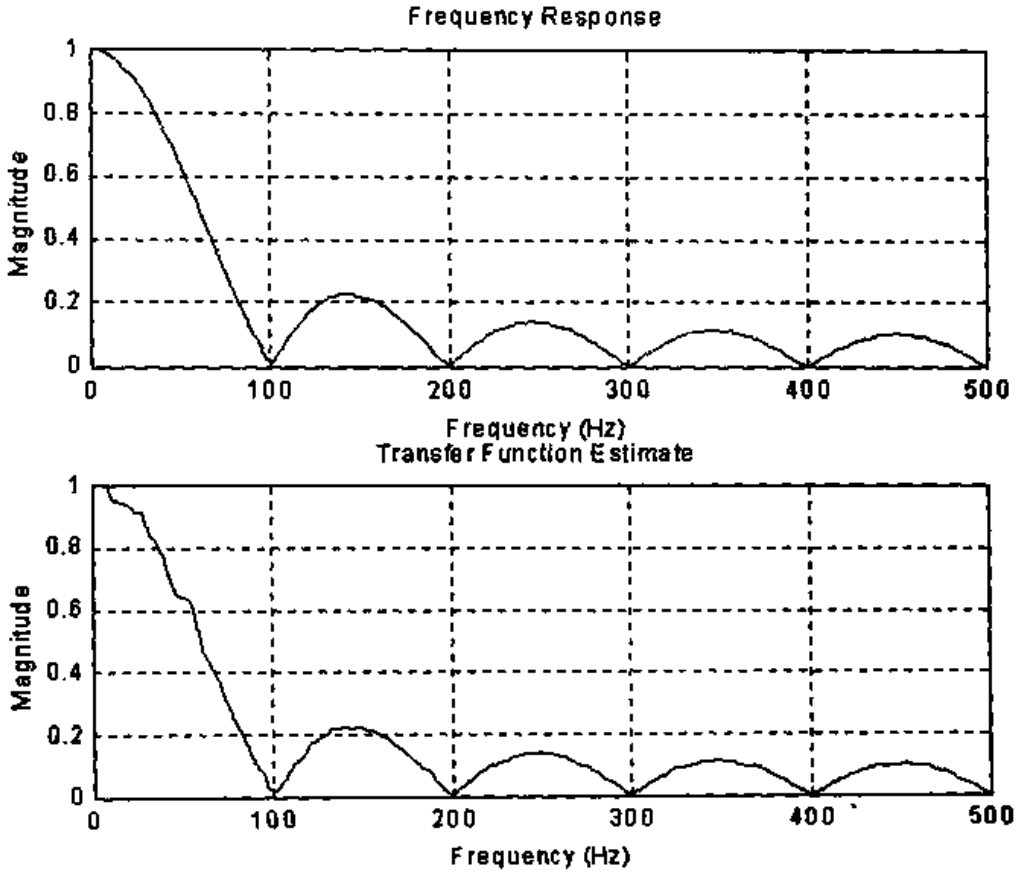


图 6.12 频率响应估计

6.5 相干函数

两个信号 x(t)和 y(t)之间的相干函数(Coherence Function)为

$$C_{xy}(\omega)=\frac{|P_{xy}(\omega)|^2}{P_{xx}(\omega)P_{yy}(\omega)}=\gamma_{xy}^2(\omega) \tag{6.5-1}$$

式中，$P_{xy}(\omega)$为 x(t)和 y(t)的互功率谱密度；$P_{xx}(\omega)$，$P_{yy}(\omega)$分别为 x(t)，y(t)的自功率谱密度。

相干函数 $C_{xy}(\omega)$为 0～1 之间实数，它用来检测信号 x(t)和 y(t)在频域内相关程度。若 y(t)为 x(t)的线性响应，则 $C_{xy}(\omega)=1$，若 x(t)和 y(t)完全互不相关，则 $C_{xy}(\omega)=0$，通常在测试过程中，$0<\gamma_{xy}<1$，这表明有三种可能：

(1)联系 x(t)和 y(t)的系统不完全是线性的；

(2)系统输出 y(t)是由 x(t)和其他信号共同引起的；

(3)在输出端有噪声干扰混入。

由于相干函数 $C_{xy}(\omega)$的上述特点使它在工程上得到一些应用，如检查两个信号之间的因果联系。

下面给出一个例子，说明 $C_{xy}(\omega)$的上述特点。

【例 6.14】 (1)计算例 6.12 中信号 x(t)和 y(t)相干函数；(2)计算两个独立信号 x(t)=u(t)，y(t)=sin(2π×50t)的相干函数。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 6-14
%Application of Coherence Function
clf
%Case 1 : yx is dependent on xn
%==========================
%Create input signal
N=1024;
Nfft=256;
window=hanning(256);
noverlap=128;
dflag='none';
Fs=1000;
n=0:N-1;
t=n/Fs;
randn('state',0)
xn=sin(2*pi*50*t)+randn(1,N);
%Create a system: filter
h=ones(1,10)/10;
%Compute output signal
```

```
yn=filter(h,1,xn);
subplot(211)
cohere(xn,yn,Nfft,Fs,window,noverlap,dflag);
title('Yx is dependent on xn');
legend('Coherence Function ',4);
%Case 2 : yx is independent on xn
%==================
%Create input signal
N=1024;
Nfft=256;
window=hanning(256);
noverlap=128;
dflag='none';
Fs=1000;
n=0:N-1;
t=n/Fs;
randn('state',0)
xn=ones(1,N);
yn=sin(2 * pi * 50 * t);
subplot(212)
cohere(xn,yn,Nfft,Fs,window,noverlap,dflag);
title('Yx is independent on xn');
legend('Coherence Function ',4);
```

程序运行结果如图 6.13 所示。显然，上图说明两个信号为因果关系，故 $C_{xy}(\omega)$ 在整个频段中为 1，下图为两个相互无关的信号，$C_{xy}(\omega)$ 很小，几乎为零。

6.6 窗函数

6.6.1 截断和频谱泄漏

信号是无限长的，而在进行信号处理时只能采用有限长信号，所以需要将信号“截断”。在信号处理中，“截断”被看成是用一个有限长的“窗口”看无限长的信号，或者从分析的角度是无限长的信号 x(t) 乘以有限长的窗函数 w(t)，由傅里叶变换性质可知

$$x(t) \cdot w(t) \Leftrightarrow X(f) * W(f) \tag{6.6-1}$$

如果 x(t) 是频宽有限信号，而 w(t) 是频宽无限函数，截断后的信号也必是频宽无限信号，从而产生所谓的频谱泄漏。频谱泄漏是不可避免的，但要尽量减小，因此设计了不同窗函数满足不同用途的要求。从能量的角度，频谱泄漏也是能量泄漏，因为加窗后，使原来的信号集中在窄频带内的能量分散到无限的频宽范围。

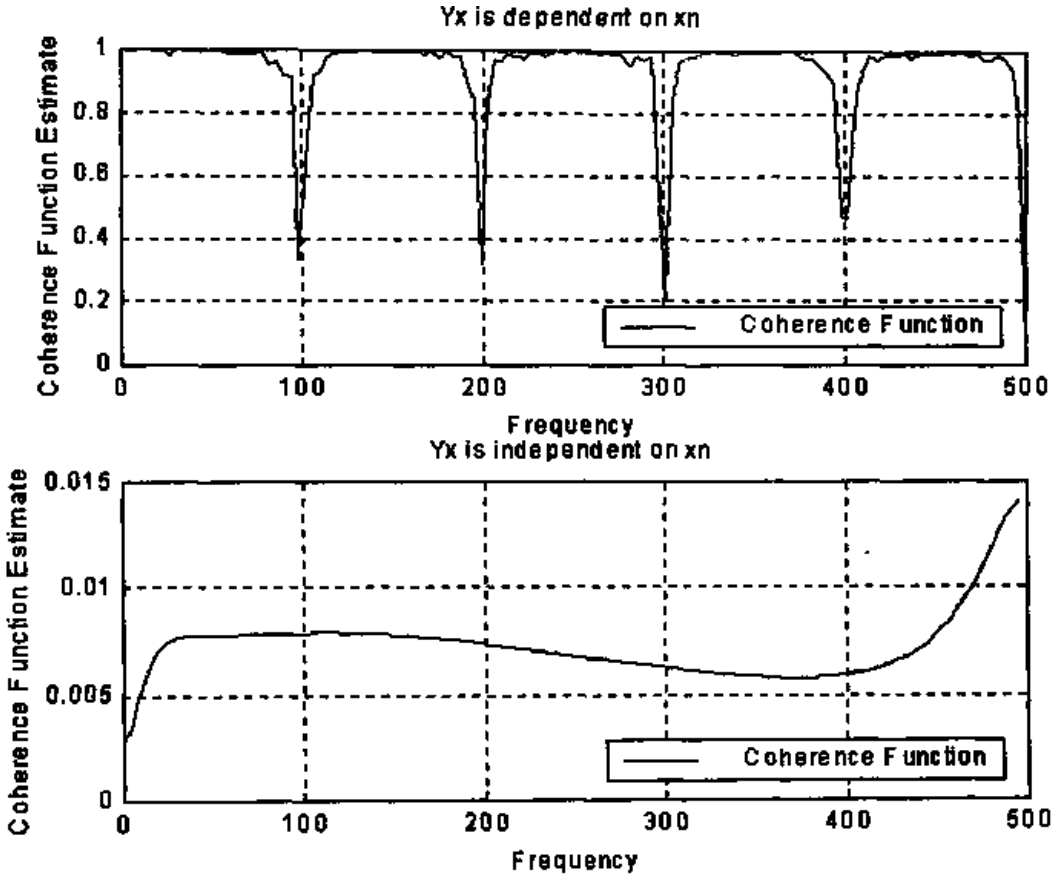


图 6.13 相干函数的应用

6.6.2 MATLAB 窗函数

MATLAB 信号处理工具箱提供了 8 种窗函数：

(1) 函数 BOXCAR 用于产生矩形窗。调用格式为

$$w = boxcar(N)$$

其中，N 为窗长度；w 为返回的窗函数序列。

(2) 函数 HANNING 用于产生汉宁(Hanning)窗。调用格式为

$$w = hanning(N)$$

Hanning 窗表达式为

$$w(k) = 0.5\left[1 - \cos\left(2\pi \frac{k}{N+1}\right)\right], \quad k = 1, \cdots, N$$

(3) 函数 HAMMING 用于产生哈明(Hamming)窗。调用格式为

$$w = hamming(N)$$

哈明窗的表达式为

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

(4) 函数 BARTLETT 用于产生巴特洛特(Bartlett)。调用格式为

$$w = bartlett(N)$$

巴特利特(Bartlett)窗表达式为

当 N 为奇数时，

$$w(k)=\begin{cases}\dfrac{2(k-1)}{N-1}, & 1\leqslant k\leqslant\dfrac{N+1}{2}\\ 2-\dfrac{2(k-1)}{n-1}, & \dfrac{n+1}{2}\leqslant k\leqslant n\end{cases}$$

当 N 为偶数时，

$$w(k)=\begin{cases}\dfrac{2(k-1)}{n-1}, & 1\leqslant k\leqslant n/2\\ \dfrac{2(n-k)}{n-1}, & \dfrac{n}{2}+1\leqslant k\leqslant n\end{cases}$$

(5) 函数 BLACKMAN 用于产生勃莱克曼(Blackman)窗。调用格式为

w=blackman(N)

勃莱克曼窗表达式为

$$w(k)=0.42-0.5\cos\left(2\pi\frac{k-1}{N-1}\right)+0.08\cos\left(4\pi\frac{k-1}{N-1}\right),\quad k=1,2,\cdots,N$$

Blachman 窗比其他相同尺寸窗(Hanning 窗，Hanning 窗)具有较宽的主瓣和侧频带泄漏小的特点。

(6) 函数 TRIANG 用于产生三角窗。调用格式为

w=triang(n)

三角窗函数表达式为

当 N 为奇数时，

$$w(k)=\begin{cases}\dfrac{2k}{N+1}, & 1\leqslant k\leqslant\dfrac{N+1}{2}\\ \dfrac{2(N-k+1)}{N+1}, & \dfrac{N+1}{2}\leqslant k\leqslant N\end{cases}$$

当 N 为偶数时，

$$w(k)=\begin{cases}\dfrac{2k-1}{N-1}, & 1\leqslant k\leqslant N/2\\ \dfrac{2(N-k+1)}{N}, & \dfrac{N}{2}\leqslant k\leqslant N\end{cases}$$

三角窗和 Bartlett 窗十分类似。三角窗的两端值不为零，而 Bartlett 窗则为 0。

(7) 函数 KAISER 用于产生凯塞(Kaiser)窗。调用格式为

w=kaiser(n, beta)

其中，beta 是 Kaiser 窗参数 β，影响窗旁瓣幅值的衰减率。

Kaiser 窗表达式为

$$w(k)=\frac{I_0\left[\beta\sqrt{1-\left(1-\dfrac{2k}{N-1}\right)^2}\right]}{I_0[\beta]}$$

式中，$I_0[\cdot]$是修正过的零阶贝塞尔(Bessel)函数。

Kaiser 窗用于滤波器设计时，若旁瓣幅值为 $-\alpha d\beta$，则

$$\beta=\begin{cases}0.1102(\alpha-8.7), & \alpha>50\\ 0.5842(\alpha-21)^{0.4}+0.07886(\alpha-21), & 50\geqslant\alpha\geqslant21\\ 0, & \alpha<21\end{cases}$$

(8) 函数 CHEBWIN 用于产生切比雪夫(Chebshev)窗。调用格式为

w=chebwin(n,r)

其中,r 是窗口的旁瓣幅值在主瓣以下的分贝数。

切比雪夫窗特点是主瓣的宽度最小,而旁瓣都是等高的且高度可调整(如图 6.16)。

【例 6.15】 用 MATLAB 编程绘制各种窗函数的形状。

```
%MATLAB PROGRAM 6-15
%Compare the different window
clf
Nwin=21;
n=0:Nwin-1;
for i=1:4
  switch i
  case 1
     w=boxcar(Nwin);
     stext='Rectangular Window';
  case 2
     w=hanning(Nwin);
     stext='Hanning Window';
  case 3
     w=hamming(Nwin);
     stext='Hamming Window';
  case 4
     w=bartlett(Nwin);
     stext='Bartlett Window';
  end
 posplot=['22'int2str(i)];
 subplot(posplot);
 stem(n,w);
 hold on
 plot(n,w,'r');
 xlabel('n');
 ylabel('w(n)');
 title(stext);
 hold off
 grid
```

```
end
```

程序运行结果，各类窗函数形状如图 6.14 所示。

【例 6.16】 用 MATLAB 编程绘制各种窗函数的幅频响应。

MATLAB 程序如下：

```
%MATLAB PROGRAM 6-16
%Compare the different window
clf
Nfft=512;
Nwin=20;
for i=1:4
  switch i
  case 1
     w=boxcar(Nwin);
     stext='Rectangular Window';
  case 2
     w=hanning(Nwin);
     stext='Hanning Window';
  case 3
     w=hamming(Nwin);
     stext='Hamming Window';
  case 4
     w=bartlett(Nwin);
     stext='Bartlett Window';
  end

[y,f]=freqz(w,1,Nfft);
mag=abs(y);
posplot=['22'int2str(i)];
subplot(posplot)
plot(f/pi,20*log10(mag/max(mag)));
xlabel('Normalized Frequency');
ylabel(' Magnitude dB');
title(stext);
grid
end
```

程序运行结果如图 6.15 所示。

6.6.3 窗函数的应用特点

由频谱图 6.15 可以看出，各种窗函数的幅频响应都存在明显的主瓣(mainlobe)和旁

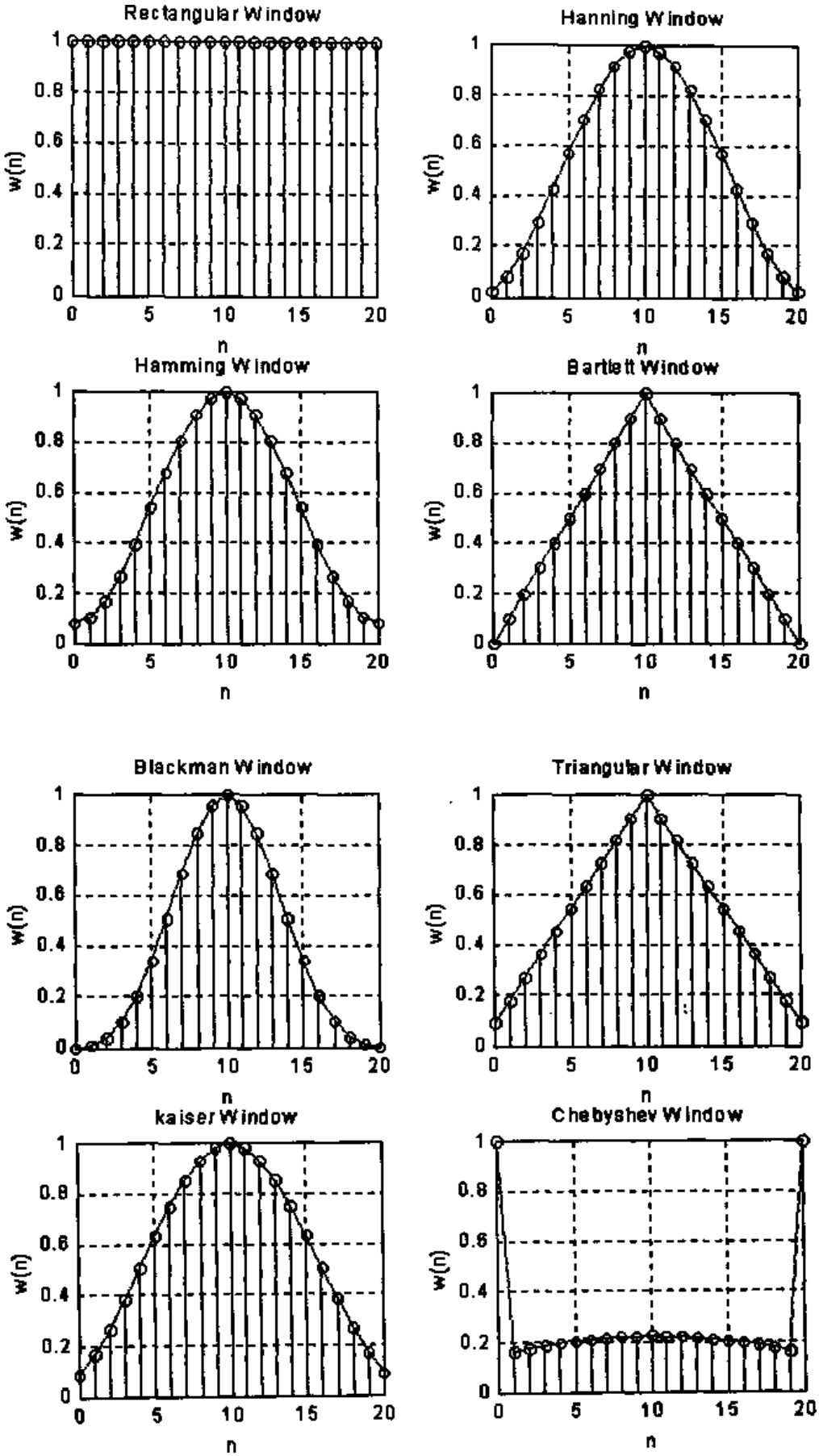

Rectangular Window
Hanning Window
Hamming Window
Bartlett Window
Blackman Window
Triangular Window
kaiser Window
Chebyshev Window
w(n)
n

图 6-14 窗函数形状

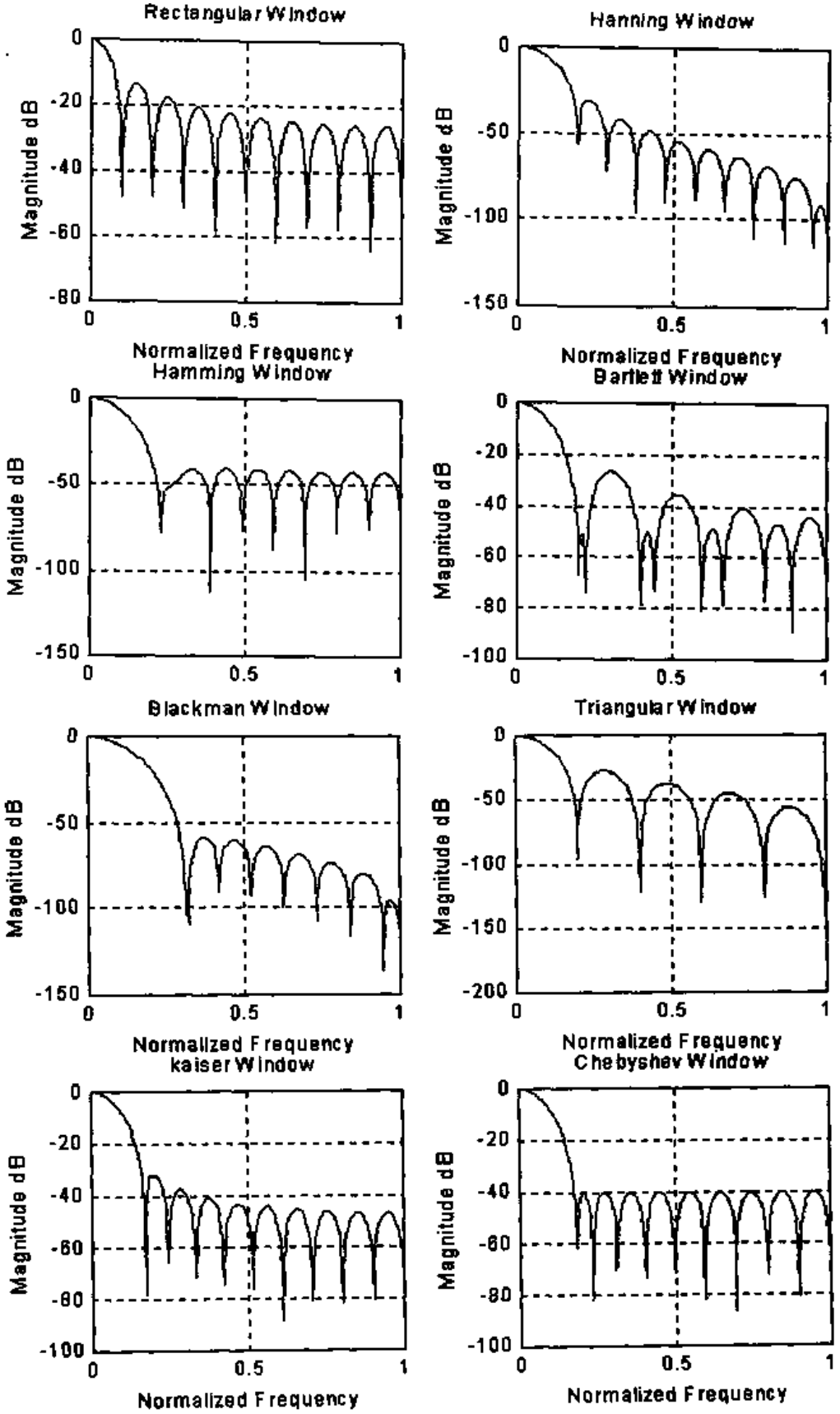


图 6.15　窗函数幅频响应

瓣(sidelobe)。主瓣频宽和旁瓣的幅值衰减特性决定了窗函数的应用。不同窗函数在这两方面的特点是不相同的。如 Blackman 窗具有最宽的主瓣，而 chebyshev 窗具有最窄的主

瓣。矩形窗的第一旁瓣衰减－13dB，Hanning 窗的第一旁瓣衰减为 31dB，而其余窗函数第一旁瓣衰减均在 40dB 以上(chebyshev 窗衰减幅度可调整)。各窗函数旁瓣衰减速度也不同。

主旁瓣的频宽还与窗长度 N 有关。增加窗长度 N 将缩小窗函数主瓣宽度，但不能减小旁瓣幅值衰减相对值(分贝数)，这个值是由窗函数决定的。这一些特点可由下面例子清楚地看出。

【例 6.17】 绘制矩形窗的幅频响应，窗长度分别为：(1)N＝10；(2)N＝20；(3)N＝50；(4)N＝100。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 6-17
%Create a Rectangular window
clf
Nfft=512;
for i=1:4
  switch i
  case 1
     Nwin=10;
  case 2
     Nwin=20;
  case 3
     Nwin=50;
  case 4
     Nwin=100;
  end

w=boxcar(Nwin);
[y,f]=freqz(w,1,Nfft);
mag=abs(y);
posplot=['22'int2str(i)];
subplot(posplot)
plot(f/pi,20*log10(mag/max(mag)));
xlabel('Normalized Frequency');
ylabel(' Magnitude dB');
stext=['N='int2str(Nwin)];
title(stext)
grid
end
```

程序运行结果如图 6.16 所示。显然，随着 N 增大，主瓣和旁瓣都变窄，但第一旁瓣相

对主瓣的幅值下降分贝数相同，第二旁瓣相对第一旁瓣幅值下降分贝也相同，……。这样，随着 N 增大，旁瓣很快得到抑制，有利减小频谱泄漏，但不能完全减除。减小主瓣宽度和抑制旁瓣是一对矛盾，不可兼得，只能根据不同用途折中处理。

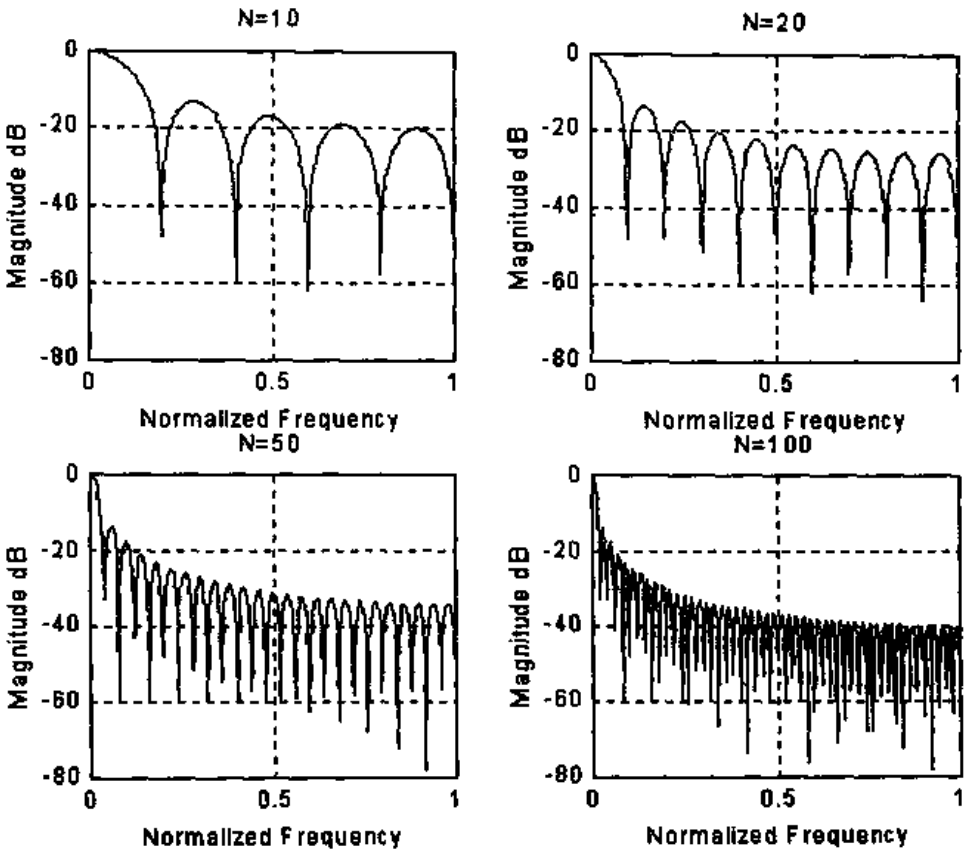


图 6.16　不同长度的矩形窗频谱

用于信号分析中的窗函数可根据不同要求选择窗函数。如主瓣宽度窄的窗函数具有较高的频率分辨率，而分析窄带，且具有较强的干扰噪声的信号，应选用旁瓣幅度小的窗函数，如汉宁窗函数等。用于滤波器的窗函数，一般要求窗函数主瓣宽度窄，以获得较好过渡带；旁瓣相对值尽可能小以增加通带段的平稳度和增大带阻的衰减。

6.6.4　窗函数在滤波器设计中的应用

MATLAB 信号处理工具箱提供用许多应用窗函数设计各类滤波器的工具函数，这些工具函数详见第三章和第四章。此外，MATLAB 还提供 Kaiser 窗设计函数 KAISERORD，调用格式为

[n, Wn, beta, ftype]=Kaiserord(f, a, dev, F_s, 'noscale')

其中，f 为滤波器频带频率向量；a 为拐角频率处的期望幅值向量；dev 为每段频带的允差幅值波纹最大偏差向量。

length(f)=2 * length(a)−2

length(dev)=length(a)

函数返回值阶数 n，频率向量 w 和多频带幅值型 ftype 均可供 MATLAB 信号工具函数 firl 直接调用。beta 为 Kaiser 窗参数。

利用该函数 Kaiserord 可方便设计低通、高通、带通等多种滤波器。

【例 6.18】 设计一个低通滤波器，通带为 0 至 1kHz，阻带从 1500Hz 至 4000Hz，通带波纹允差 5%，阻带波纹允差 1%，阻带衰减 40dB。由前述，Kaiser 窗旁瓣比主瓣下降 40 分贝以上，故采用 Kaiser 窗函数设计滤波器。

用 AMTLAB 编程如下：

```
%MATLAB PROGRAM 6-18
% Design a lowpass filter with Kaiser window
Fs=8000;
N=216;
fcuts=[1000 1500];
mags=[1 0];
devs=[0.05 0.01];
[n,Wn,beta,ftype]=kaiserord(fcuts,mags,devs,Fs);
hh=fir1(n,Wn,ftype,kaiser(n+1,beta),'noscale');
freqz(hh,1,N,Fs)
```

程序运行结果如图 6.17 所示。可见满足设计要求。

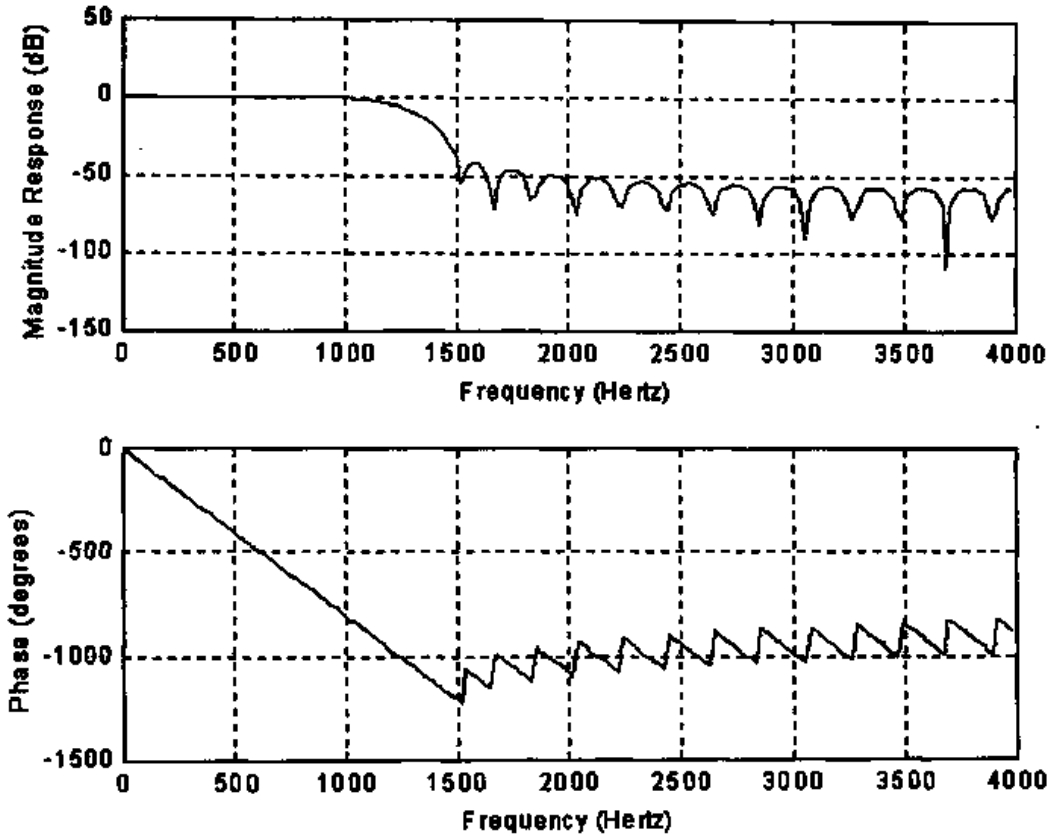


图 6.17 Kaiser 窗低通滤波器

【例 6.19】 利用 Kaiser 窗函数设计一个带通滤波器，上截止频率 2500Hz，下截止频率 1000Hz，过渡带宽 200Hz，通带波纹允差 0.1，阻带波纹不大于允差 0.02dB，通带幅值为 1。

用 MATLAB 编程如下：

```
%MATLAB PROGRAM 6-19
% Design a bandpass filter with Kaiser window
Fs=8000;
N=216;
fcuts=[1000 1200 2300 2500];
mags=[0 1 0];
devs=[0.02 0.1 0.02];
[n,Wn,beta,ftype]=kaiserord(fcuts,mags,devs,Fs);
n=n+rem(n,2);
hh=fir1(n,Wn,ftype,kaiser(n+1,beta),'noscale');
[H,f]=freqz(hh,1,N,Fs);
plot(f,abs(H));
xlabel('Frequency (Hz)');
ylabel('Magnitude |H(f)|');
grid
```

程序运行结果如图 6.18 所示，显然设计的滤波器频率特性完全设计要求。

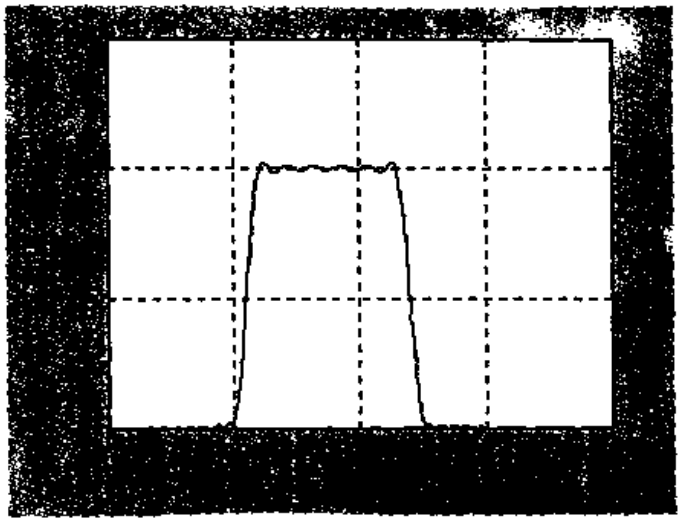


图 6.18 Kaiser 窗带通滤波器

6.7 时谱分析

时谱分析(Cepstrum Analysis)是一种非线性信号处理技术，它在语言、图像和噪声处理领域中都有广泛应用。

时谱可分为两类：复时谱和功率时谱。MATLAB 信号处理工具箱提供复时谱分析工具函数。复时谱(Complex Cepstrum)的定义为

$$\hat{x}(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi}\{\ln[x(e^{j\omega})]\}e^{j\omega n}d\omega \tag{6.7-1}$$

由上可见，复时谱实际上是序列 $x(n)$ 的傅里叶变换 $x(e^{j\omega})$ 的自然对数，再取傅里叶逆

变换，其结果复时谱仍然是一个序列。这就是说，复时谱是 x(n)从时域至频域、频域至频域、频域至时域的三次变换。

MATLAB 信号处理工具箱函数 CCEPS 用于按式(6.7-1)估计一个序列 x 的复时谱，调用格式为

```
xhat=cceps(x)
[xhat, nd]=cceps(x)
```

其中，x 为输入序列(实序列)；xhat 为复时谱(复序列)；nd 为圆周延时采样数。

MATLAB 信号处理工具箱还提供函数 RCEPS，按下式估计序列的实时谱

$$C_x = \frac{1}{2\pi}\int_{-\pi}^{\pi} \ln|x(e^{j\omega})|e^{j\omega n}d\omega \qquad (6.7\text{-}2)$$

函数调用格式为：

```
y=rceps(x)
[y, ym]=rceps(x)
```

其中，x 为实序列；y 为实时谱(实序列)；y_m 为输入序列的最小相位重构。

由式(6.7-2)可知，我们不能从序列 x 的实时谱重构原始序列，因为实时谱仅根据序列的傅氏变换的幅值计算的。然而，可采用窗函数重构原始序列的最小相位模式。

由于复时谱从复频谱得到的，它不损失相位信息，因此复时谱是可逆的。功率时谱(功时谱或倒频谱)过程是不可逆的。

时频分析技术广泛地应用于语言信号分析、同态滤波技术。

这里举一个说明复时谱在具有回波声信号测量中的应用。

【例 6.20】 设原声波是一个 45Hz 的正弦波，在传播过程中遇到障碍产生回声，回声振幅衰减为原信号的 0.2，并与原信号有延迟 0.2s。在某点测得信号是原信号和回声信号的叠加。用复时谱可监测回声信号。用 MABTLAB 编写程序如下：

```
%MATLAB PROGRAM 6-20
%Applicaton of Cepstrum Analysis
t=0:0.01:1.49;
sig=sin(2*pi*45*t);
echo=0.5*[zeros(1,20) sig(1:length(t)-20)];
sigecho=sig+echo;
c=cceps(sigecho);
plot(t,c);
xlabel('t(s)');
axis([0 1.5 -1 1]);
grid
```

程序运行结果可得信号的复时谱如图 6.19 所示。可见在 t=0.2 秒处有一个峰值，这就是回声信号。

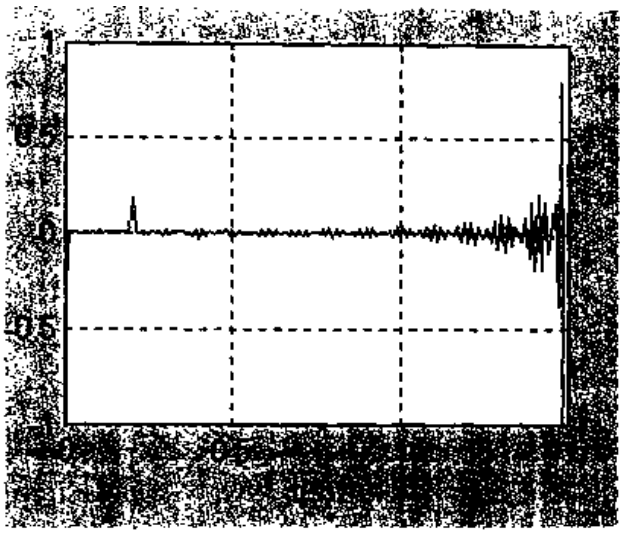


图 6.19　复时谱分析

习　题

6.1　已知信号 $x(t)=3\cos(2\pi f_1 t+30°)+0.2\cos(2\pi f_2 t+60°)$，$f_1=10Hz$，$f_2=100Hz$，试用 MATLAB 编程计算信号 x(t)的自相关函数，绘制自相关函数曲线并分析其特点。

6.2　用 MATLAB 生成周期正三角波信号，并绘制自相关函数曲线。

6.3　已知两个频率均为 50Hz、相位差为 60°的余弦信号，试计算其互相关函数并绘制函数图。

6.4　已知一个被白噪声 w(t)污染的信号 x(t)，其表达式为

$$x(t)=2\cos(2\pi f_1 t)+0.5\cos(2\pi f_2 t)+0.5\sin(2\pi f_3 t)+w(t)$$

其中，$f_1=25Hz$，$f_2=75Hz$，$f_3=150Hz$，w(t)为白噪声，采用下列谱估计方法进行功率谱估计并绘制其功率谱估计图：

(1) welch 法；

(2) MTM 法；

(3) MEM 法；

(4) MUSIC 法。

6.5　已知周期信号 $x(t)=0.75+3.4\cos 2\pi ft+2.7\cos 4\pi ft+1.5\sin 3.5\pi ft+2.5\sin 7\pi ft$，其中 $f=\frac{25}{16}Hz$，若截断时间长度为信号周期的 0.9 和 1.1 倍，试绘制和比较采用下面窗函数提取 x(t)的频谱：

(1) 矩形窗；

(2) 汉宁窗；

(3) 哈明窗；

(4) 巴特洛特窗；

(5) 勃莱克曼窗；

(6) 三角窗；

(7) 凯塞窗；

(8) 切比雪夫窗。

6.6 利用切比雪夫 I 型窗设计一个带通模拟滤波，通带截止频率为 1000Hz～2000Hz，过渡带为 250Hz，通带波纹不大于 2dB，阻带衰减不小于 40dB，试绘出滤波器的幅频特性和相频特性。

第七章　交互式图形用户界面

MATLAB 信号处理工具箱提供功能齐全的交互式图形用户界面(Interactive Graphical User Interface),GUI 是信号分析和处理的图形环境,用户直观地使用鼠标在计算机屏幕上控制数字信号输入、观察和测量;计算、考察和实现数字滤波器;对信号进行频谱分析,了解信号的频率成分等。因此,用户无须详细了解 MATLAB 信号处理工具箱函数的语法规则,就可以完成大部分信号分析处理工作。

本章简要地介绍 GUI 的基本结构及如何利用它来完成信号时域和频谱分析、滤波设计和分析的。

7.1　图形用户界面组成

在 MATLAB COMMAND 窗口下,键入 sptool,立即弹出一个 SPTool 窗口,如图7.1所示。若是第一次打开,是一个未定名(untitled)SPTool 窗。用户在使用后可给窗定名、保存以便下次调用,如给窗口定名 h1.spt,定名后的窗口文件称为"session"文件。SPTool 窗是交互式信号分析处理环境,通过这一窗口用户能可视化地完成信号分析与处理的全部工作。窗口有信号(Signal)、滤波器(Filter)、频谱(Spectra)三个栏,它们分别记载了用户所使用过的信号、滤波器和频谱。

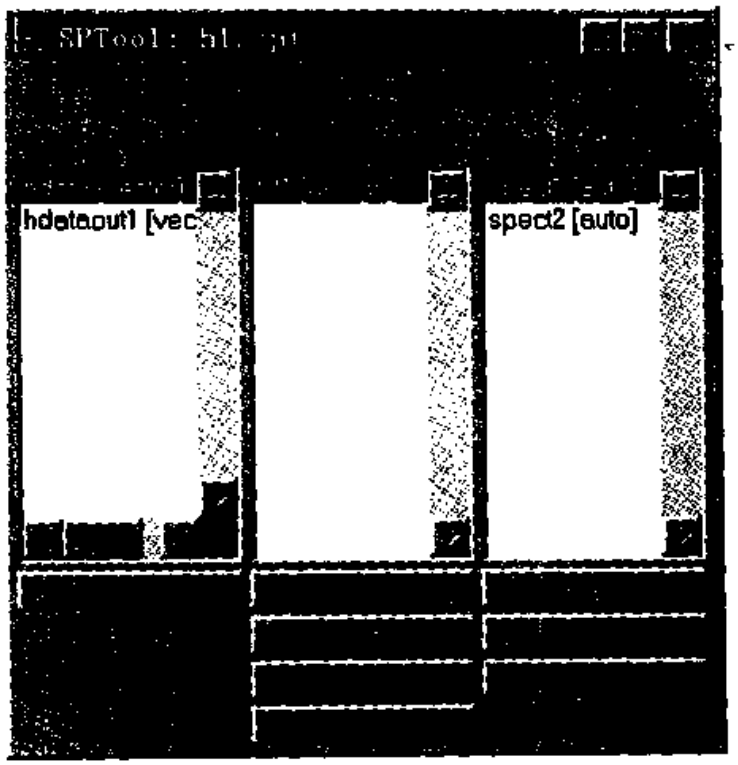


图 7.1　SPTool 窗口

SPTool 窗口的主要命令菜单有 File 和 Eidt:

(1)File 菜单

- Open Session 打开已存在扩展名为.spt 的 SPTool 窗口；
- Import 弹出对话框，从磁盘或 MATLAB 工作空间输入信号、滤波器和频谱，它们必须是 .MAT 文件形式；
- Export 向 MATLAB 工作空间或磁盘输出信号、滤波器和频谱的结构参数；
- Save Session，Save Session As 存放当前的 SPTool 窗口为带扩散名.spt 的 MAT 文件；
- Perferences 设置信号处理 GUI 工具的性能；
- Close 关闭 SPTool 窗口。

(2)Edit 菜单

在弹出的子菜单中有四条命令：

- Duplicate 复制选定的信号、滤波器或频谱；
- Name 给选定的信号、滤波器或频谱更换名称；
- Clear 清除选定的信号、滤波器或频谱；
- Sampling Frequency 给所选定的信号或滤波器设置采样频率，采样频率设置可采用两种形式：一种是数字，如 1，100，1000；另一种是有效的 MATLAB 表达式，如 Fs，1/Ts 等，这里 Fs，Ts 应为已赋值变量。

SPTool 窗口的各栏下面有若干个命令条，用以激活 SPTool 的四个信号分析处理工具窗口，这四个窗口主要功能如下：

- 信号浏览图(Signal Browser)　用于信号观察、测量和时域分析；
- 频谱图窗(Spectrum View)　用于信号的频谱观察和分析。可选择第六章讲述的频谱估计方法产生信号的频谱图，观察、修改和测量信号的频谱；
- 滤波器图窗(Filter View)　用于观测滤波器时域和频域特性，包括幅值响应，相位响应、群延迟(滤波器延迟相对频率函数)、零极点图、脉冲响应和阶跃响应；
- 滤波器设计窗 (Filter Designer)　用于设计和编辑不同长度和类型标准结构的 FIR 和 IIR 滤波器，包括低通、高通、带通、带阻。

Signal 栏下的命令条 View 用来激活 Signal View 窗。

Filter 栏下有四个命令条：

- View 用来激活 Filter View 窗；
- New Design 用来激活 Filter Designer 窗并设计新的滤波器；
- Eidt Design 用来激活 Filter Designer 窗并编辑一个由 SPTool 设计的滤波器，改变它的某些参数；
- Apply 应用一个选定的滤波器对一个选定信号进行处理，产生一个新的信号。新信号通过对话框存入 Signal 栏中，便于做信号分析。

Spectra 栏下有三个命令条：

- View 用来激活 Spectrum Viewer 窗，观察已存在的信号频谱；
- Create 用来激活 Spectrum Viewer 窗，产生所选定信号的频谱；
- Update 用于更新已选定信号频谱并用现选定信号的频谱所取代。

7.2 信号时域分析

这里以一个例子说明如何输入信号至 SPTool 窗口和应用 SPTool 窗，对一个信号进行时域分析。

7.2.1 信号输入和命名

把信号输入至 SPTool 并定名的步骤如下：

(1) 确定对象信号。

首先寻找要分析的对象信号的数据文件所在的目录，并确认数据文件带有.mat 扩展名(SPTool 要求的数据格式)。

若对象信号是一个数学表达式和标准 MATLAB 函数，则可在 MATLAB COMMAND窗口下，创建一个.mat文件存放信号数据。如要分析信号表达式为

$$x(t)=\sin(2\pi\times 20t)+\sin(2\pi\times 100t)+w(t)$$

式中，w(t)为白噪声信号。

用 MATLAB 编程生成存放信号的数据文件如下：

```
%MATLAB PROGRAM 7-1
%create a signal and save as MAT-file
randn('state',0);
N=4000;
Fs=1000;
n=0:N-1;
t=n/Fs;
x=sin(2 * pi * 50 * t)+sin(2 * pi * 200 * t)+randn(1,N);
plot(t,x)
grid
save hdata.mat x Fs
```

程序运行结果生成信号数据文件 hdata.mat，存放信号 x 和采样周期 Fs 的数据。若信号是实测并用其他高级语言(如 C 语言)写成的数值文件，应该将文件格式转换成.mat文件。

(2) 在 MATLAB COMMAND 窗口下键入 sptool，弹出 SPTool:untitled.spt 窗口。

(3) 从 File 菜单中选用 Import 命令，弹出 Import to SPTool 窗，如图 7.2 所示。

(4) 信号输入和取名

- 确认 Import As 弹出菜单选择 Signal，在 Import to SPTool 窗的 Source 栏内用鼠标击活 From Disk 圆按钮；按 Browser 按钮选定磁盘的.mat 文件 hdata.mat；变量 x，Fs 出现在 Workspace Content 栏中；
- 选择 x 通过栏上侧边[→]送入 Data 文本框内；
- 选择 Fs 通过栏下侧边[→]送入 Sampling Frequency 文本栏内；
- 在 Name 文本框内填写信号处 hdata；

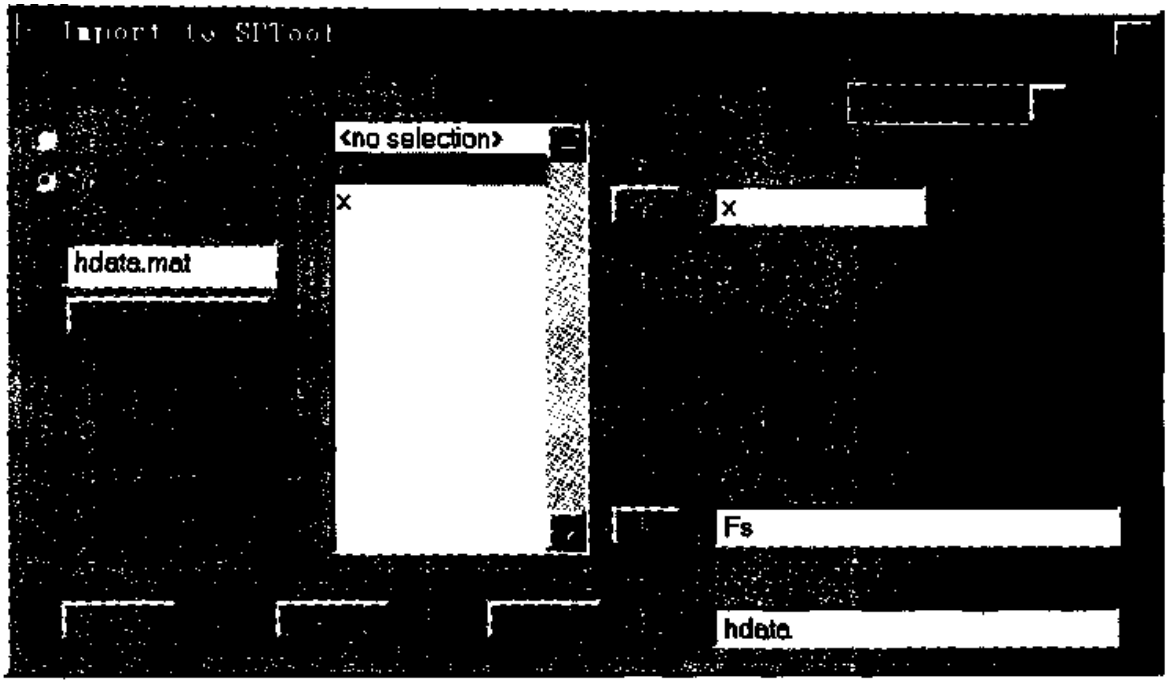


图 7.2 Import to SPTool 窗

· 鼠标单击 OK 按钮。

这样，Import to SPTool 窗消失，SPTool 窗重新激活，Signal 栏中新添一项：hdata [vector]，信号输入和命名工作结束。

(5) 给 SPTool 窗 untitled. spt 定名和存放。

从 SPTool 窗口 File 菜单中选择 Save Session 或 Save Session As，输入文件名：h1，这样这个已输入信号的 SPTool 窗定名为 h1. spt 并存放在当前目录下。

7.2.2 信号的观察和测量

信号的观察和测量的步骤如下：

(1) 在 SPTool：h1. spt 窗口 signals 栏内用鼠标选择对象信号；

(2) 用鼠标单击 signals 栏下面的 view 菜单条，这时弹出 Signal Browser 窗，如图 7.3 所示；

(3) 信号观察和测量

Signal Browser 窗除包括有信号曲线显示区外，窗内还有 Zoom Control(缩放控制)、Ruler Control(标尺控制)两组功能按钮。这两组按钮便于用户对信号进行更好观察和测量。

Signal Browser 窗口右上角还有一组 Selection(选择)按钮使用户方便地选择感兴趣的信号。Color 按钮可给不同信号用不同颜色加以区分。

Signal Browser 窗可认为是一台虚拟数字示波器，用户可以随意的选择这些按钮对所显示的信号进行分析，就像操作示波器的面板一样。

用户可借助 Rulers 中的 Track 按钮完成对信号的跟踪测量，方法是：

① 选择要测量的信号，有两种途径：

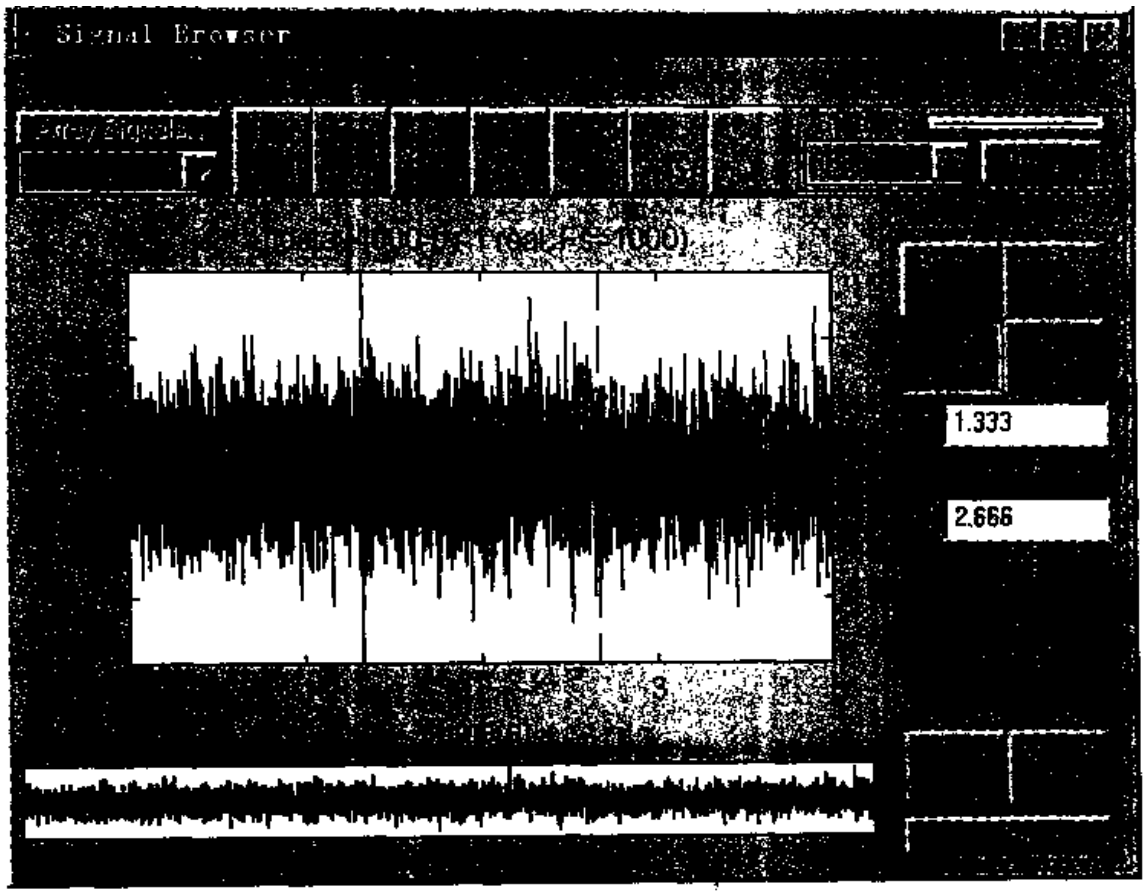


图 7.3 Singnal Browser 窗

- 用鼠标弹出 Selection 下的信号名，并选择要测量信号；
- 用鼠标光标单击要测量的信号线。

② 选择 Rulers 按钮

用鼠标单击 Track 按钮在信号图中出现两条垂直的标尺线，每条标尺线(Ruler)上有一个圆形(或方形等)标志(Mark)。

③ 用鼠标测量

用鼠标拖动标尺线 2(或 1)，标尺上的标志照着信号线移动，同时 Rulers 面板框内，显示 x_1，y_1，x_2，y_2，dx，dy，m 数字也跟随变动。

x_1，y_1 分别为标志 1 在 x 轴和 y 轴上的位置；x_2，y_2 分别为标志 2 在 x 轴和 y 轴上的位置；$dx=x_2-x_1$；$dy=y_2-y_1$；m 为 dy/dx，为 x1 和 x_2 之间的斜度，通过这些数字用户可准确测量任一点信号大小。

④ 用 Rulers 面板上数字文本栏测量

这是另一种信号测量方法。如要测量 x=0.25 时的信号 y 值。在 Rulers 面板 x_1(或 x_2)文本栏内键入 0.25 确认后，一个新的数字出现在 y_1 文本栏内，这就是所需要的信号值；同时标尺 1 也移至新的位置上。

用户可借助 Signal Browser 窗的 Zoom In－X 或 Zoom Out－X 按钮改变信号在 x 轴上的显示范围；Zoom In－Y 或 Zoom Out－Y 按钮改变信号在 y 轴上的显示范围；Mouse Zoom 按钮用鼠标圈定显示范围；Full View 按钮恢复原来的图形显示。通过这些 Zoom 控制按钮，用户可清晰观察复杂信号的任何一段变化情况。

用户可借助 Color 按钮，用不同颜色区分多条信号线。

用户还可以借助 Signal Browser 窗内菜单 Option 的 Play 命令，听到信号所发出的声音，这对于语言和噪声信号处理，这提供听觉试验的途径。

7.3 滤波器设计、编辑和观察

7.3.1 滤波器设计

在 SPTool 窗中第二栏是 Filter(滤波器)。一个新的 SPTool 窗口，Filter 栏是空的。用户可以用 File 中的 Import 命令，从工作空间或磁盘输入已有的滤波器，其方法和 7.2 节中信号输入相同。若没有现成的滤波器或这些滤波器不适用，用户要新设计滤波器。

用 SPTool 设计滤波器前，必须弄清滤波器的性能要求和指标。

作为一个例子，对 7.2.1 节所描述信号进行滤波处理，要去除信号中频率大于 50Hz 的频率成分，设计一个低通滤波器。利用 SPTool 设计这一滤波器的步骤如下：

(1) 鼠标单击 SPTool 窗 Filter 栏下的菜单条 New Design，激活弹出 Filter Designer 窗如图 7.4 所示。

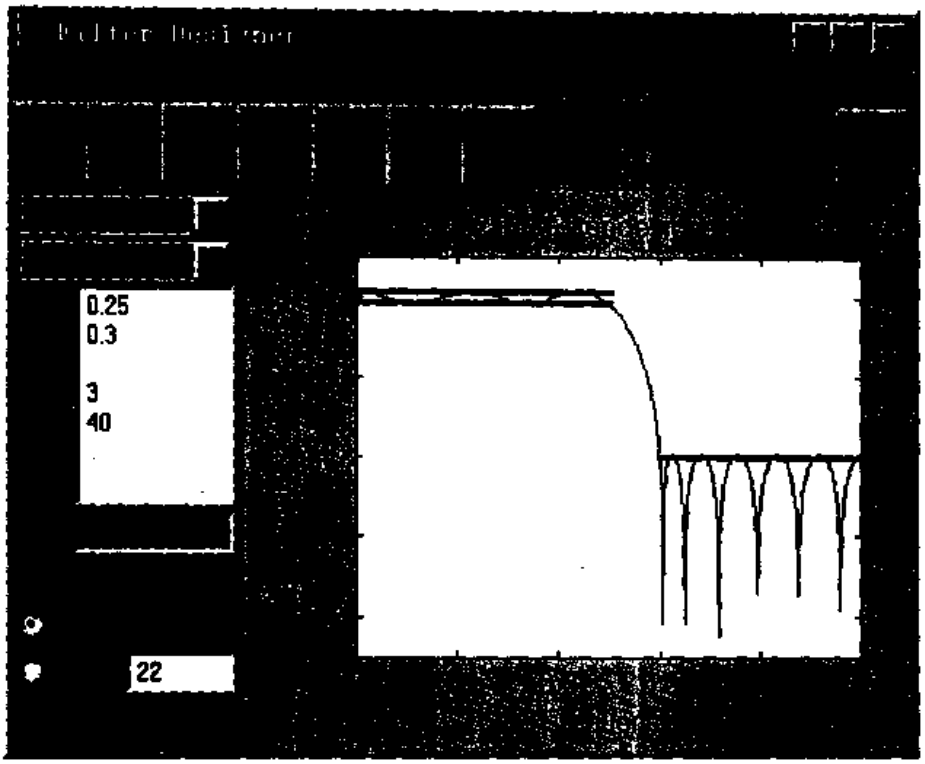


图 7.4 Filter Designer 窗

Filter Designer 窗包括：

- 滤波器频谱图显示区(中间)。滤波器设计算法弹出菜单：Equiripple(等波纹)，Least Square(最小二乘拟合)，Kaiser Window(凯塞窗)，Butterworth(巴特沃斯)，Chebyshev 1(切比雪夫 1 型)，Chebyshev 2(切比雪夫 2 型)，Elliptic(椭圆)；
- 滤波器类型弹出菜单：Lowpass(低通)，Highpass(高通)，Bandpass(带通)，Band-

stop(带阻)；

- 滤波器参数框，边界频率 f_1，f_2 或 f_1，f_2，f_3，f_4，R_p 通带波纹，dB，R_s 阻带衰减，dB；
- 采样频率框 F_s；
- 滤波器阶次选择(Order)：Auto(自动)，Set(设置)。

(2) 输入采样频率

激活 SPTool 窗，从菜单 File 中选择 Sampling Frequency；filt2[Design]，弹出一个对话框，键入实际采样频率 Hz 值：1000；按 OK。新的采样频率值 1000 出现在 Filter Designer 窗内的 Fs 框内。

(3) 选择滤波器设计算法

用鼠标弹出滤波器算法菜单，选定算法，如 Equiripple(等波纹)。

(4) 选择滤波器类型

用鼠标弹出滤波器类型菜单，选定类型，本例为 Lowpass。

(5) 设置滤波器参数

低通边界频率：$f_1=100$(Hz)；$f_2=150$(Hz)

通带纹波指标：$R_p=3$(dB)

阻带衰减：$R_s=40$(dB)

(6) 设置滤波器阶次：选择 Auto

(7) 鼠标在滤波器显示框内单击一下，新的滤波器设计计算即完成，Filter Designer 窗口中间位置显示区内显示新的滤波器期望幅频谱。

(8) 至此，新的滤波器设计工作完成，这是一个 22 阶 FIR 低通滤波器，名为 filt2[Design]出现在 SPTool 窗的 Filter 栏内。

7.3.2 滤波器编辑

滤波器的编辑就是对 SPTool 窗口 Filter 栏中列出的原设计滤波器参数作某些修改。

滤波器编辑设计步骤如下：

- 鼠标单击 SPTool 窗 Filter 栏下的 Edit Design 按钮，激活弹出 Filter Designer 窗口，如图 4；
- 按 7.3.1 所述方法修改滤波器有关参数；
- 鼠标在滤波器显示区内单击一下，修改后的滤波器的幅频谱显示在 Filter Designer 窗口内。

7.3.3 滤波器分析

为了对滤波器的时域和频域特性进行观察和分析，SPTool 包括相应 Filter Viwewr 窗。滤波器特性观察分析步骤如下：

(1) 在 SPTool：h1.spt 窗口 Filter 栏内选择设计的滤波器 filt2；

(2) 选择 Filter 栏下的 View 菜单条，激活弹出 Filter viewer 窗，如图 7.5 所示。

Filter View 窗包括：

- 图形显示区；

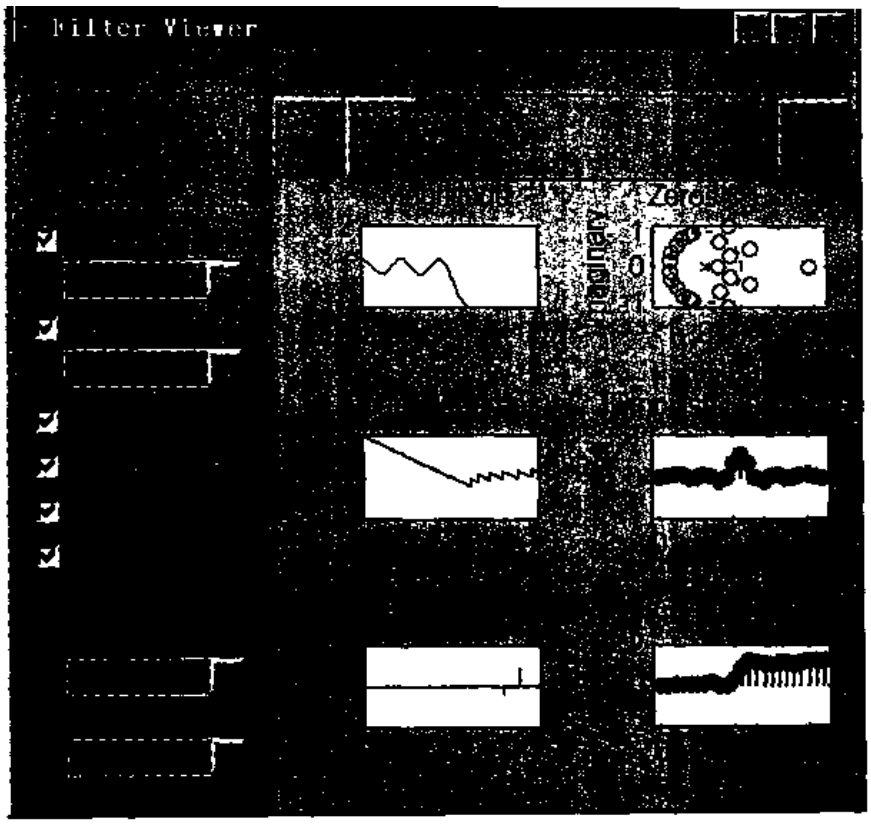


图 7.5 Filter Viewer 窗

• plot 栏：

□ Magnitude(幅频图)弹出菜单：

linear(线性)；log(对数)；decibels(分贝)

□ phase(相频图)弹出菜单：

degrees(度) radians(弧度)

□ group Delay(群延迟图)

□ Zeros and Poles(零极点图)

□ Impulse Response(脉冲响应图)

□ Step Response(阶跃响应图)

(3)Frequency Axis(频率轴)

Scale(坐标刻度)：Linear(线性)，log(对数)

Range(范围)：[0，F_s/2]，[0，F_s]，[−F_s/2，−F_s/2]

(3) 选择要显示的滤波器特性图的一个或几个，选择适当横坐标和纵坐标，选择适当的频率显示范围，可得到最多六种滤波器时域频域特性图供分析。

(4) 利用 Mouse Zoom 按钮观测局部曲线段的放大图形；利用 Full View 按钮使显示恢复。

7.3.4 信号滤波

用已设计的滤波器对信号进行滤波产生一个新的信号，新信号不同于原信号。如本例中，经过低通滤波器处理后的信号，高频成分抑制，而低频信号得到保留。

现在利用 SPTool：h1. spt 窗的滤波器 filt1 对原信号 hdata 进行滤波处理，滤波后新信号产生的步骤如下：

(1) 在 SPTool 窗口的 Signal 栏内选择要处理的原信号 hdata。

(2) 在 SPTool 窗的 Filter 栏内选择滤波器，名为 Filt2[design]。

(3) 在 SPTool 窗的 Filter 栏下方选择菜单条 Apply，弹出 Apply Filter 对话框，如图 7.6所示。

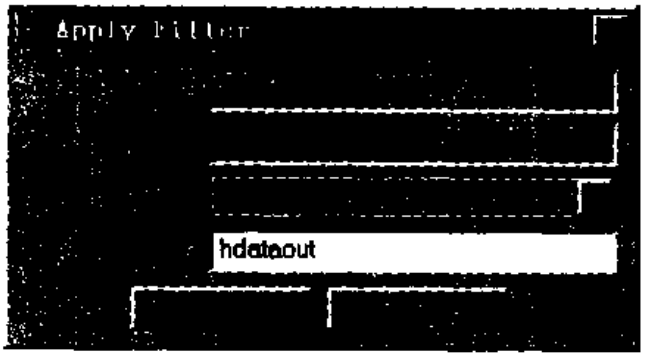


图 7.6 Apply Filter 对话框

在 Output Signal 文本框中键入新信号名：hdataout，单击 OK。

(4) 在 SPTool 窗的 Signal 栏中自动出现新信号名称 hdataoutl[vector]。

产生滤波后新信号产生工作完成。

这样，在 SPTool 窗的 Signal 栏内有 hdata(原信号)和 hdataout(滤波信号)两个信号。

我们可利用 Signal Browser 窗对两个信号进行时域分析、测量和比较，同时还可以利用 SPTool 的 spectra 栏对两个信号作频域分析和比较。

7.4 信号的频谱分析

SPTool 窗可对 signal 栏内列出的信号作频谱分析，包括用不同方法或不同窗口产生和更新信号频谱图、观察和测量信号的频谱。

7.4.1 信号频谱图生成

产生一个信号的频谱图的步骤如下：(1) 鼠标在 SPTool：ht. spt 窗 signal 内选择一个待分析的信号，如 hdata；(2)单击 SPTool 窗 spctra 栏下边的菜单条 Create 激活弹出 Sptectrum Viewer 窗(见图 7.7)。

Spectrum Viewer 窗包括：

① 频谱图显示区

② Parameter(参数)栏

- Method(频谱估计方法)弹出菜单：welch 法、MTM 法、MEM 法、MUSIC 法；
- Nfft(FFT 长度)文本框；
- Nwind(窗口长度)文本框；
- Window 窗函数类型弹出菜单：hanning(汉宁窗)，bartlett(巴特洛特窗)；blackman(勃莱克曼窗)，boxcar(矩形窗)，Kaiser(凯塞窗)；triang(三角窗)；
- Overlap(重叠)文本框；
- Detrending(去除趋势项)弹出菜单：none(无)，linear(去除线性趋势项)，mean(去除平均值)；

· Scaling 弹出菜单： unbiased(无偏)；Peaks(峰值)；by F。

· Conf Int(置信区间)。

③ Zoom 控制按钮：和 Signal Browsr 窗基本相同

④ Rules 控制按钮：和 Signal Browser 窗基本相同

⑤ Selection 用于频谱选择

⑥ Color 不同信号频谱线可用不同颜色区分

(3) 在 Spectrum Viewer 窗内选择合适的频谱估计参数。注意到窗口内的信号名：hdata，4000×1 向量，采样频率 Fs=1000Hz，从而确认这正是我们将要作频谱分析的信号，此后，在 Parameter 框内选择合适的频谱估计方法、窗函数等。

(4) 用鼠标单击 Spectruim Viewer 窗右下方的 Apply 按钮后，信号 hdata 频谱图出现在图形显示区中。如图 7.7 所示。

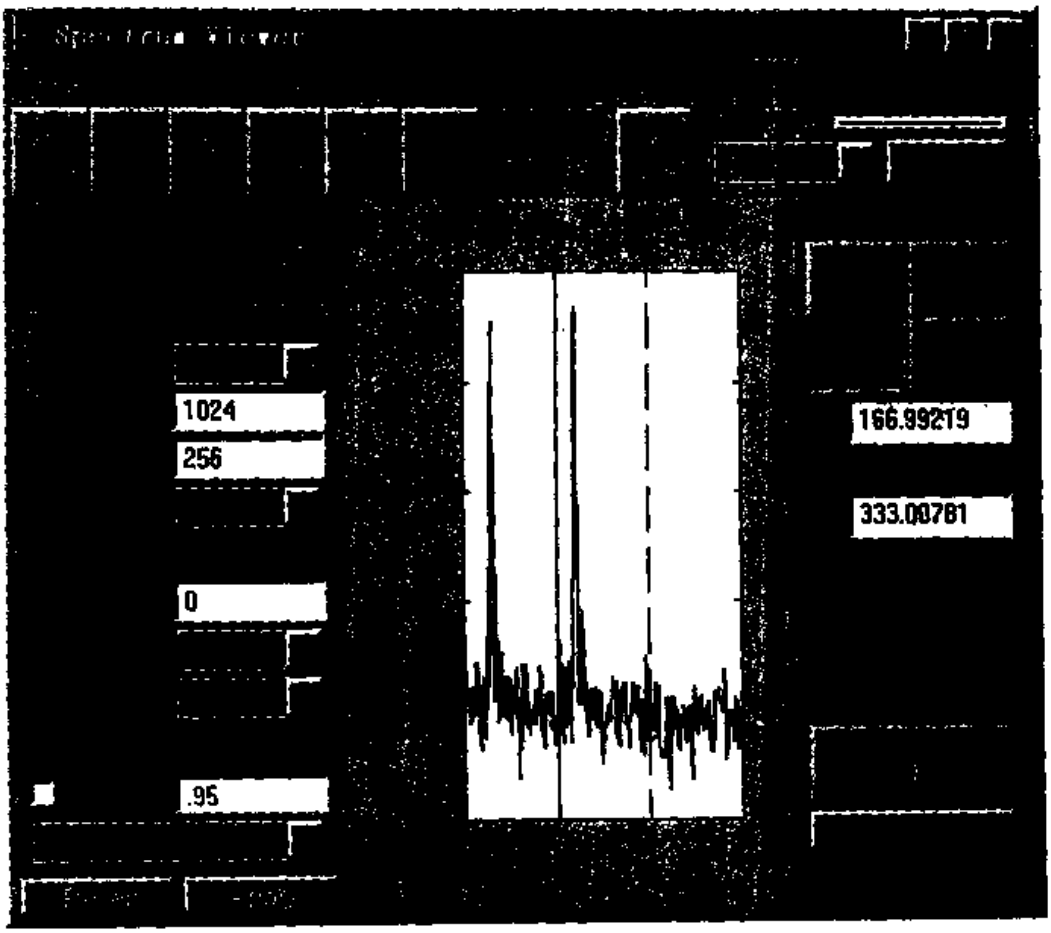


图 7.7 Spectrum Viewer 窗

在 SPTool 窗口 Spectra 栏内出现该频谱名 spectr 1[auto]。

同样的方法，可以产生滤波后信号 hdataout 的频谱 spect 2[auto]。

7.4.2 信号频谱的更新

必要时，用户可对 SPTool 窗 spectra 栏中的频谱作更新(Updata)，如修改某些频谱估计参数等。频谱更新的步骤如下：

(1) 在 SPTool 窗的 spectra 栏内选定要更新的滤波器。

(2) 鼠标单击 spectra 栏下面菜单条 Update 激活弹出 Spectrum Viewer 窗口，确认该窗口所显示的对象信号。

(3) 修改 Spectrum Viewer 窗内 parameter 框内有关参数。

(4) 按窗下边 Apply，在 Spectrum Viewer 窗的频谱显示区内出现新的频谱曲线。

7.4.3 信号频谱的观测

和信号时域观察和测量一样，用户可激活 Spectrum Viewer 窗内的 Zoom 控制按钮和 Ruler 控制按钮，对 spectra 栏内的频谱进行观察和测量。具体如下：

(1) 在 SPTool:h1.spt 窗口 spectra 栏内选择对象信号频谱

(2) 用鼠标单击 spectra 栏下面的 View 菜单条，弹出 Spectrum Viewer 窗(如图 7.7)

(3) 利用 Spectrum Viewer 窗口下的 Zoom 控制按钮和 Rulers 控制按钮对信号频谱进行观察和测量。

如通常感兴趣的是信号中是否含有周期性成分，频率多少。这类问题利用 Spectrum Viewer 字的 Rulers 框内的 Track 按钮极易实现。

从频谱 spect 1 可见，信号 hdata 中有两个周期性成分。用鼠标拖动 Ruler 1 线至第一个峰值处，x1 的文本栏内显示：49.804688；再用鼠标拖动 Ruler 2 线至第二个峰值处，x2 的文本框内显示：200.19531；dx 显示：150.39062。由此可见，由频谱图测量，这两个周期成分的频率分别为：49.804688Hz 和 200.1953Hz。这与信号式(7.1-1)的结构完全吻合(所存在微小偏差是数字表示误差)。

7.4.4 不同信号频谱的比较

利用 Spectrum Viewer 可将不同信号的频谱图放在一起进行比较。这里以原信号 hdate 的频谱 spect 1 和滤波后信号 hdataout 的频谱 spect 2 为例，说明如何比较两个频谱。不同频谱比较的步骤如下：

(1) 在 SPTool:h1.spt 窗的 spectra 栏内用 Alt+Shift 同时选取 spect 1 和 spect 2。

(2) 用鼠标单击 spectra 栏下面菜单条 View，弹出 Spectrum Viewer，在频谱显示区内出现两条谱线 spect 1、spect 2。

(3) 改变 spect 2 谱线颜色，以便和 spect 1 区分，方法是：在 Select 下弹出菜单，选择 spect 2；鼠标单击 color 按钮，弹出 Edit line 对话框如图 7.8 所示。在 Edit line 对话框内选择 line Style(线类型)：dashed(虚线)；选择 line Color：Red [1 0 0]；按 OK。这样 spect 1 谱线为蓝实线，而 spect 2 谱线为红虚线。

(4) 频谱线测量

① spect 1 频谱线测量：在 selection 弹出菜单下选取 spect 1；激活 Rulers 框的 Track (跟踪)按钮；拖动 Ruler1 测得第一峰值处 $x_1=49.804688$ $y_2=17.776593$；拖动 Ruler2，测得第二峰值处，$x_2=200.1953$ $y_2=18.311892$。

② spect 2 谱线测量：在 selection 弹出菜单内选取 spect 2；拖动 Ruler 1，测得第一峰值处，$x_1=49.804688$，$y_1=17.619254$；拖动 Ruler 2，测得第 2 峰值处，$x_2=200.1953$ $y_2=-17.7842$。

比较两谱线测量值说明，滤波后信号保留原信号频率为 49.804688Hz 的周期信号几

乎没有衰减，而原信号中的频率为 200Hz 的周期信号及高频信号已衰减 20dB 以上。两个频谱图比较如图 7.9 所示。

7.5 SPTool 选择项设置

利用 SPTool 窗 File 菜单的 Preference 命令设置或修改 SPTool 窗和它的四个工具窗(signal Browser, Filter Viewer, Filter Designer, Spectrum Viewer)的显示参数。通过 Preference 提供的对话框，用户可以十分方便地设置或修改所有选择项，从而使图形显示及操作功能满足要求。

用鼠标从 SPTool 窗口 File 菜单中选择 Preference 命令，单击它，立即弹出 Preference for SPTool 对话框，如图 7.10 所示。

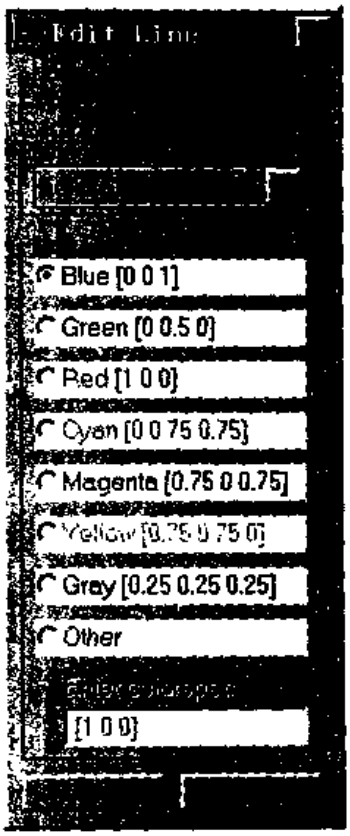


图 7.8 线编辑对话框

在 Preference for SPTool 对话框中，列出设置选择项目有：

(1) Rulers(标尺)：选取 Rulers，在窗口内出现对话框。通过对话框可设置各窗口颜色，标志(Rulers Marker)形状，标志大小，初始位置。

(2) Color(颜色)：选取 Color，在窗口内出现对话框。通过对话框可设置各窗口显示的颜色和线型的的指令。

(3) Signal Browser：选取 Signal Browser，弹出对话框，可设置 Signal Browser 窗口 x 轴和 y 轴的文字标注。

(4) Spectrum Viewer：选取 Spectrum Viewer，在窗口内出现对话框。通过对话框可设置 Spectrum Viewer 窗口幅值轴刻度：dB, linear；频率轴刻度：linear, log；频率轴范围：[0, $F_s/2$] [0, F_s], [$-F_s/2$, $F_s/2$]。

(5) Filter Viewer：选取 Filter Viewer，在窗口内出现对话框。通过对话框可设置 Fitler Viewer 窗的参数：FFT 长度，时间响应长度，幅值轴刻度(linear, log, decibels)，相位单位(degrees, radians)，频率轴刻度(linear, log)，频率轴范围([0, $F_s/2$], [0, F_s], [$-F_s/2$, $F_s/2$])。

(6) Filter Viewer-Tiling：选取 Filter Viewer-Tiling，通过它的对话框可改变 Filter Viewer 窗内图形排列方式：2×3，3×2，6×1(垂直排列)1×6(水平排列)。Filter Viewer 最多可显示 6 种滤波器特性图(见第 7.3.2 节)。

(7) Filter Designer：选取 Filter Designer，通过它对话框可设置 Filter Designer 窗的工作参数：FFT 长度，频率网络间隔，幅值网格间隔。

(8) Plug-Ins：选择 Plug-Ins，通过它的对话框可在 SPTool 中加入新的功能框、新按钮、新的频谱估计法等，使 SPTool 的功能得到进一步扩展。

关于 Preference 应用的详细资料可查阅 Signal Proccess Toolbox 的用户手册。

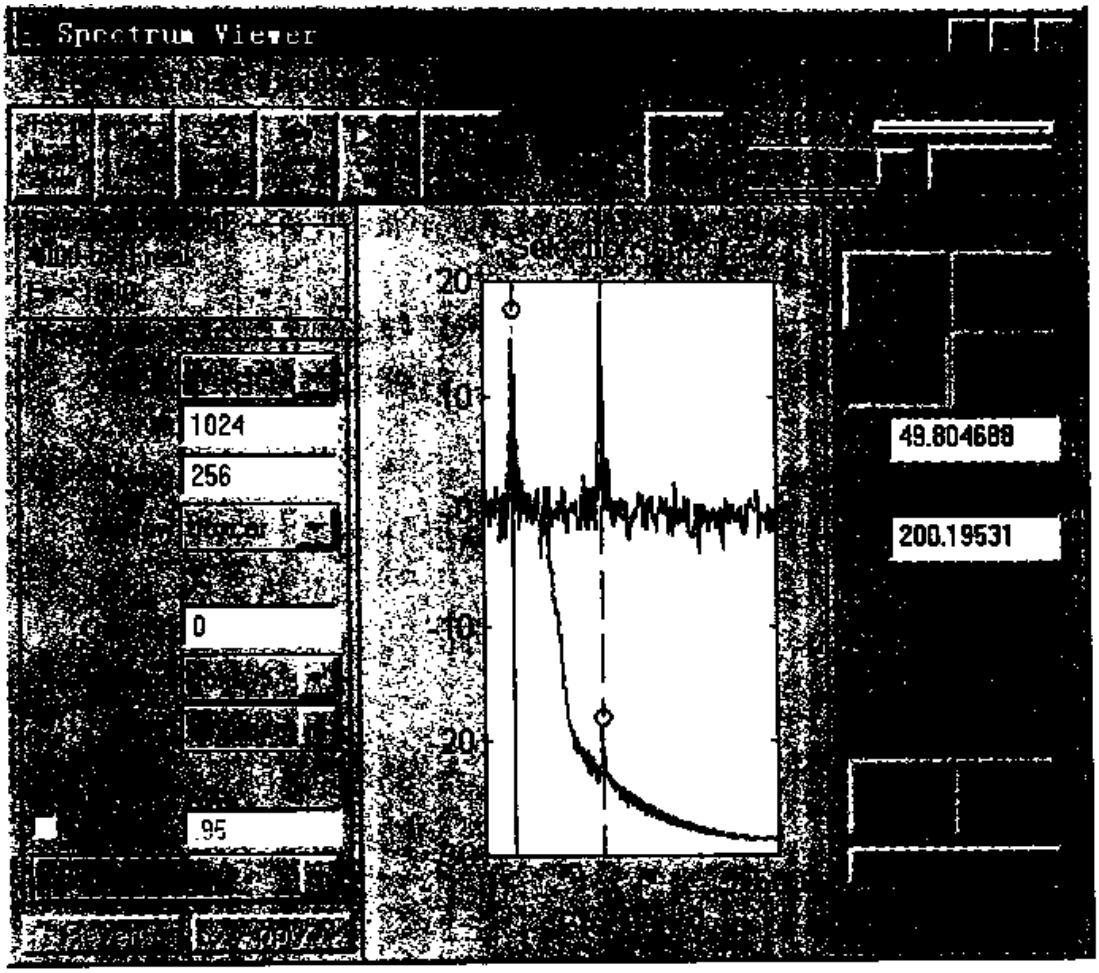


图 7.9　不同频谱线的比较

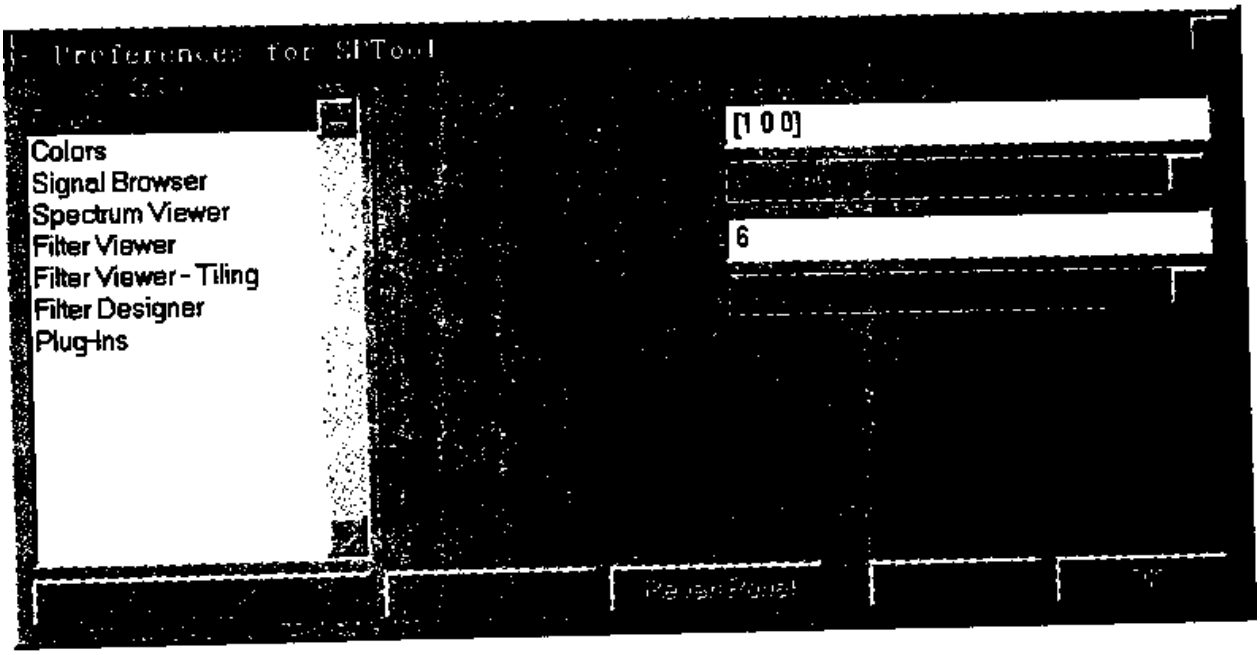


图 7.10　Preference 主对话框

习　题

7.1　用 MATLAB 生成信号 $x(t)=0.5+3\sin(2\pi f_1 t)+0.5\cos(2\pi f_2 t)+w(t)$，其中，$f_1=5Hz$，$f_2=200Hz$，$w(t)$为白噪声信号，采样频率为 1000Hz。并用 STool 观察该信号。

7.2　用 STool 窗设计一个 Butterworth 低通滤波器，通带边界频率为 100Hz，阻带边界频率 150Hz，通带波纹不大于 2dB，阻带衰减不小于 30dB，并观察滤波器的幅频图、相频图、群延迟图、零极点图和脉冲响应图。

7.3　用题 7.2 设计的滤波器对题 7.1 的信号进行滤波处理，观察滤波后的信号图。

7.4　用 STool 窗对题 7.1 的原信号和题 7.3 的滤波信号进行谱分析。

(1) 绘制两个信号的频谱图；

(2) 比较两信号谱线的差别并说明原因；

(3) 用不同估计方法进行谱分析，并比较结果。

附录一　信号处理工具箱函数

函　数	说　明
波形产生和绘图	
chirp	产生扫描频率余弦
diric	产生 Dirichlet 或周期 sinc 函数
gauspuls	产生高斯调制正弦脉冲
pulstran	产生脉冲串
rectpuls	产生非周期矩形信号
sawtooth	产生锯齿波或三角波
sinc	产生 sinc 函数
square	产生方波
strips	产生条图
tripuls	产生非周期三角波
滤波器分析和实现	
abs	绝对值(幅值)
angle	相位角
conv	卷积和多项式乘法
conv2	二维卷积
fftfilt	基于 FFT 重叠加法的数据滤波
filter	递归(IIR)或非递归(FIR)滤波器的数据滤波
filter2	二维数字滤波
filtfilt	零相位数字滤波
filtic	函数 filter 初始条件确定
freqs	模拟滤波器频率响应
freqspace	频率响应的频率空间设置
freqz	数字滤波器频率响应

grpdelay	群延迟
impz	数字滤波器的脉冲响应
latcfilt	格型梯形滤波器实现
unwrap	相位角展开
zplane	零极点图
线性系统变换	
convmtx	卷积矩阵
latc2tf	格型滤波器转换为传递函数形式
poly2rc	多项式系数转换为反射系数
rc2poly	反射系数转换为多项式系数
residuez	z-传递函数的部分分式展开
sos2ss	二阶级联转换为状态空间
sos2tf	二阶级联转换为传递函数
sos2zp	二阶级联转换为零极点增益形式
ss2sos	状态空间转变为二阶级联形式
ss2tf	状态空间转换为传递函数
ss2zp	状态空间转换为零极点增益
tf2latc	传递函数转换为格型滤波器
tf2ss	传递函数转换为状态空间
tf2zp	传递函数转换为零极点增益
zp2sos	零极点增益转换为二阶级联形式
zp2ss	零极点增益形式转换为状态空间
zp2tf	零极点增益转换为传递函数
IIR 滤波器设计——经典和直接法	
besself	Bessel(贝塞尔)模拟滤波器设计
butter	Butterworth(巴特沃斯)滤波器设计
cheby1	Chebyshev(切比雪夫) I 型滤波器设计(通带波纹)
cheby2	Chebyshev(切比雪夫) II 型滤波器设计(阻带波纹)

ellip	椭圆(Cauer)滤波器设计
maxflat	通用数字 Butterworth 滤波器设计
yulewalk	递归数字滤波器设计
IIR 滤波器阶数的选择	
buttord	Butterworth 型滤波器阶数的选择
cheb1ord	Chebyshev Ⅰ型滤波器阶数的选择
cheb2ord	Chebyshev Ⅱ型滤波器阶数的选择
ellipord	椭圆滤波器阶次选择
FIR 滤波器设计	
cremez	复响应和非线性相位等波纹 FIR 滤波器设计
fir1	基于窗函数的有限冲激响应滤波器设计——标准响应
fir2	基于窗函数的有限冲激响应滤波器设计——任意响应
fircls	多频带滤波的最小方差 FIR 滤波器设计
fircls1	低通和高通线性相位 FIR 滤波器的最小方差设计
firls	最小线性相位滤波器设计
firrcos	升余弦 FIR 滤波器设计
intfilt	插值 FIR 滤波器设计
kaiserord	用凯塞(Kaiser)窗估计函数 fir1 参数
remez	Parks-McClellan 优化滤波器设计
remezord	Parks-McClellan 优化滤波器阶估计
变　换	
czt	Chirp z-变换
dct	离散余弦变换
dftmtx	离散傅里叶变换矩阵
fft	一维 FFT
fft2	二维 FFT
fftshift	函数 fft 和 fft2 输出的重新排列
hilbert	希尔伯特(Hibert)变换

idct	离散余弦逆变换
ifft	一维逆 FFT
ifft2	二维逆 FFT
统计信号处理	
cohere	两个信号相干函数估计
corrcoef	相关系数矩阵
cov	协方差矩阵
csd	互功率谱密度估计(CSD)
pmem	最大熵功率谱估计
pmtm	多窗口功率谱估计(MTM)
pmusic	特征值向量功率谱估计(MUSIC)
psd	自功率谱密度估计
tfe	传递函数估计
xcorr	互相关函数估计
xcorr2	二维互相关函数估计
xcov	互协方差函数估计
窗函数	
bartlett	巴特利斯(Bartlett)窗
blackman	勃莱克曼(Blackman)窗
boxcar	矩形窗
chebwin	切比雪夫(Chebyshev)窗
hamming	哈明(Hamming)窗
hanning	汉宁(Hanning)窗
kaiser	凯塞(Kaiser)窗
triang	三角窗
参数建模	
invfreqs	由频率响应辨识连续时间(模拟)滤波器
invfreqz	由频率响应辨识离散时间滤波器

levinson	Levinson-Durbin 递归算法
lpc	线性预测系数
prony	Prong 法的时域 IIR 滤波器设计
stmcb	利用 Steiglitz-McBride 迭代法求线性模型
特殊运算	
cceps	复时谱分析
cplxpair	重新排列组合复数
decimate	降低序列的采样频率
deconv	解卷积和多项式除法
demod	通信仿真中的解调制
detrend	去除线性趋势
dpss	Slepian 序列
dpssclear	去除数据库 Slepian 序列
dpssdir	从数据库目录消去 Slepian 序列
dpssload	从数据库调入 Slepian 序列
dpsssave	Slepian 序列存入数据库
icceps	倒复时谱
interp	整数倍提高采样速率
medfilt1	一维中值滤波
modulate	通讯仿真调制
polystab	稳定多项式
rceps	实时谱和最小相位重构
resample	任意倍数改变采样速率
specgram	频谱分析
upfirdn	利用 FIR 滤波器转换采样频率
vco	电压控制振荡器

模拟原型设计	
besselap	Bessel 模拟低通滤波器原型设计
buttap	Butterworth 模拟低通滤波器原型设计
cheb1ap	chevbyshev Ⅰ型模拟低通滤波器原型设计
cheb2ap	chevbyshev Ⅱ型模拟低通滤波器原型设计
ellipap	椭圆低通滤波器原型设计
频率变换	
lp2bp	低通至带通模拟滤波器变换
lp2bs	低通至带阻模拟滤波器变换
lp2hp	低通至高通模拟滤波器变换
lp2lp	低通至低通模拟滤波器变换
滤波器离散变换	
bilinear	双线性变换
impinvar	冲激不变法的模拟至数字滤波器变换
交互式工具	
sptool	交互式信号、滤波器和频谱分析工具

附录二　MATLAB 5 函数

常用命令(General Purpose Commands)

函　数	说　明
管理命令和函数(Managing Commands and Functions)	
addpath	添加 MATLAB 的搜索路径
doc	显示 HTML 帮助文件
help	MATLAB 函数和 M 文件在线帮助
lasterr	最近的错误信息
lastwarn	最近警告信息
lookfor	关键词检索
path	控制 MATLAB 的搜索路径
profile	检测和显示 M-文件运行记录
rmpath	删除 MATLAB 的搜索路径
type	显示 M-文件清单
version	MATLAB 版本信息
what	列出当前目录上的 M-文件、MAT 文件和 MEX 文件
whatsnew	显示 MATLAB 和工具箱的 README 文件
which	显示文件或函数的路径
变量和工作空间管理(Managing Variables and the Workspace)	
clear	从内存中删除变量和函数
disp	显示文本和数组内容
length	求向量的长度
load	从磁盘中调入数据变量
mlock	防止 M-文件清除
munlock	允许 M-文件清除

pack	压缩工作空间
save	把内存变量存入磁盘
size	求数组的维数大小
who	列出工作空间中的变量名
whos	列出工作空间中的变量详细内容
命令窗口控制命令(Controlling the Command Window)	
echo	显示 M 文件执行时是否显示命令的切换开关
format	控制输出格式
more	命令窗口分页输出的控制开关
操作环境和文件管理(Working with Files and the Operating Environment)	
acopy	苹果机上把文件从一个文件夹拷入另一个文件夹
amove	苹果机上把文件从一个文件夹移入另一个文件夹
applescript	苹果机上从一个文件中载入编译后的文件并执行
arename	苹果机上文件重命名
areveal	在苹果机桌面上显示文件名
cd	变换工作目录
delete	删除文件或图形对象
diary	存储命令窗口中的操作内容
dir	列出目录
edit	编辑 M 文件
fileparts	生成文件的路径、文件名的版本等部分信息
fullfile	生成文件的路径和文件名的全称
inmem	列出内容中的函数
matlabroot	安装 MATLAB 的根目录
gestalt	苹果机上的启动函数
tempdir	列出系统的临时目录
tempname	列出临时文件名
!	执行外部应用文件

MATLAB 的启动和终止(Starting and Quitting MATLAB)

matlabrc	MATLAB 的主启动文件
quit	退出 MATLAB
startup	启动 MATLAB 的自执行文件

运算符和特殊算符(Operators and Special Characters)

函　数	说　明
+	加
-	减
*	矩阵相乘
.*	数组相乘
^	矩阵求幂
.^	数组求幂
kron	张量积
\	左除
/	右除
./	数组右除
.\	数组左除
:	冒号运算符
()	小括号
[]	中括号,生成数组
{}	大括号,生成细胞
.	小数点
...	续行符
,	逗号
;	分号
%	注释号

'	共轭转置符
.'	非共轭转置符
=	赋值符号
= =	等号
< >	关系符
&	逻辑和
|	逻辑与
~	逻辑非
xor	逻辑异或

逻辑函数(Logical Functions)

函 数	说 明
all	全非 0 元素数组为真
any	有非 0 元素的数组为真
exist	检验某个变量或文件是否存在
find	查找非 0 元素的下标和值
is *	查询状态
* isa	查询对象的类
logical	把数值型转换为逻辑型
mislocked	如果 M-文件没有被清除为真

编程和调试(Language Constructs and Debugging)

函 数	说 明
编程(MATLAB as a Programming Language)	
builtin	用覆盖调入方法运行内部函数
eval	运行字符串形式的 MATLAB 表达式
evalin	工作空间表达式的扩展

feval	运行字符串定义的函数文件
function	函数 M-文件
global	定义全局变量
nargchk	求输入参数的个数
persistent	定义永久变量
script	命令文件
流程控制(Control Flow)	
break	中断执行 for 或 while 循环
case	switch 结构关键字
catch	开始捕捉模块
else	条件执行语句
elseif	条件执行语句
end	For、while、switch 和 if 的结束语句或标志
error	显示错误信息
for	指定循环次数的执行语句
if	条件执行语句
otherwise	switch 语句的默认部分
return	返回主调函数
switch	开关语句
warning	显示警告信息
while	无规定次数循环语句
交互输入(Interactive Input)	
input	提醒用户输入
keyboard	文件执行中转入键盘状态
menu	为输入生成选择菜单
pause	暂停命令

面向对象编程(Object-Oriented Programming)	
class	创建新类或返回对象的类
double	转换为双精度型
inferiorto	低级类关系
inline	创建 inline 函数对象
isa	检验某变量是否为给定类的对象
superiorto	优先级类关系
uint8	转换为 8 位型
调试(Debugging)	
dbclear	删除中断点
dbcont	恢复运行
dbdown	转换工作空间
dbmex	调试 MEX 文件
dbquit	退出调试状态
dbstack	显示调用堆栈的函数
dbstatus	列出所有断点
dbstep	从断点处继续运行一行或多行
dbstop	在 M-文件函数件设置断点
dbtype	列出 M 文件,每行带行号
dbup	转换工作空间

基本矩阵和数组运算(Elementary Matrices and Matrix Manipulation)

函　数	说　明
基本矩阵和数组(Elementary Matrices and Arrays)	
eye	生成单位矩阵
linspace	生成线性等间隔的向量
logspace	生成对数等间隔的向量

ones	生成全 1 数组
rand	生成均匀分布随机数和随机矩阵
randn	生成高斯分布随机数和随机矩阵
zeros	生成全 0 数组
:	生成等间距向量
特殊变量和常数(Special Variables and Constants)	
ans	最近运算结果(无变量名)
computer	运行 MATLAB 的计算机机型
eps	浮点数相对误差
flops	计算浮点运算的次数
i	虚数单位
inf	无穷
inputname	输入参数名称
j	虚数单位
NaN	非数
nargin, nargout	函数的输入参数和输出参数个数
pi	圆周率 π
realmax	最大正浮点数
realmin	最小正浮点数
varargin, varargout	返回参数的变量个数
时间和日期函数(Time and Dates)	
calendar	生成某月日历
clock	当前时间函数
cputime	CPU 运行时间
date	当前日期字符串
datenum	日期的序列数
datestr	日期的字符串格式
datevec	日期组成元素

eomday	月的最后一天
etime	时间差
now	当前日期和时间
tic	秒表启动
toc	秒表第终止和显示
weekday	周日
矩阵运算(Matrix Manipulation)	
cat	数组组合
diag	生成对角矩阵和取出矩阵对角线元素
fliplr	矩阵的左右翻转
flipud	矩阵上下翻转
repmat	复制和编排矩阵
reshape	数组变形
rot90	矩阵旋转 90°
tril	矩阵的下三角部分
triu	矩阵的上三角部分
:	数组的下标引用和重排

特殊矩阵(Specialized Matrices)

函　数	说　明
compan	伴随矩阵
gallery	测试矩阵的类型
hadamard	生成 Hadamard 矩阵
hankel	生成 Hankel 矩阵
hilb	生成 Hilbert 矩阵
invhilb	生成 Hilbert 矩阵的逆矩阵
magic	生成魔方矩阵

pascal	生成 Pascal 矩阵
toeplitz	生成 Toeplitz 矩阵
wilkinson	Wilkinson 特征值测试矩阵

基本数学函数(Elementary Math Functions)

函　数	说　明
abs	实数的绝对值和复数的模
acos	反余弦
acosh	反双曲余弦
acot	反余切
acoth	反双曲余切
acsc	反余割
acsch	反双曲余割
angle	相角
asec	反正割
asech	反双曲正割
asin	反正弦
asinh	反双曲正弦
atan	反正切
atanh	反双曲正切
atan2	四象限反正切
ceil	朝正无穷方向取整
conj	复共轭
cos	余弦
cosh	双曲余弦
cot	余切
coth	双曲余切

csc	余割
csch	双曲余割
exp	指数
fix	朝 0 方向取整
floor	朝负无穷方向取整
gcd	最大公因子
imag	取出复数的虚部
lcm	最小公倍数
log	自然对数
log2	基为 2 的对数
log10	常用对数
mod	求余
nchoosek	求矢量元素的全部的组合
real	复数的实部
rem	除法的余数
round	四舍五入取整
sec	正割
sech	双曲正割
sign	符号函数
sin	正弦
sinh	双曲正弦
sqrt	平方根
tan	正切
tanh	双曲正切

特殊数学函数(Specialized Math Functions)

函　数	说　明
airy	Airy 函数
besselh	第三类 Bessel 函数
besseli、besselk	修正 Bessel 函数
besselj、bessely	Bessel 函数
beta、betainc、betaln	Beta 函数
ellipj	椭圆 Jacobi 函数
ellipke	第一、二类完全椭圆积分
erf、erfc、erfcx、erfinv	Error 函数
expint	指数积分
gamma、gammainc	Gamma 函数
gammaln	Gamma 函数
legendre	Legendre 函数
pow2	求 2 的幂
rat、rats	有理分数近似

坐标系统转换(Coordinate System Conversion)

函　数	说　明
cart2pol	把直角坐标转换为极坐标或圆柱坐标
cart2sph	把直角坐标转换为球坐标
pol2cart	把极坐标或圆柱坐标转换为直角坐标
sph2cart	把球坐标转换为直角坐标

矩阵函数-数值线性代数(Matrix Functions-Numerical)

函 数	说 明
矩阵分析(Matrix Analysis)	
cond	和逆相关的条件数
condeig	和特征值相关的条件数
det	矩阵的行列式
norm	向量或矩阵的模
null	零空间
orth	正交化空间
rank	矩阵的秩
rcond	逆条件数
rref,rrefmovie	压缩的行阶梯形
subspace	两个子空间的角度
trace	对角线元素和(迹)
线性方程(Linear Equations)	
\,/	线性方程求解符
chol	Cholesky 分解
inv	矩阵求逆
lscov	已知协方差的最小二乘解
lu	LU 分解
nnls	非负最小二乘解
pinv	矩阵的伪逆
qr	正交三角分解
特征值和奇异值(Eigenvalues and Singular Values)	
balance	提高特征值精度的选项
cdf2rdf	复数对角型转换为实块对角型
eig	求特征值和特征向量

gsvd	广义奇异值分解
hess	矩阵的 Hessenberg 形式
poly	求已知根的多项式的表达式
qz	广义特征值
rsf2csf	把实的 Schur 形式转换为复的 Schur 形式
schur	Schur 分解
svd	奇异值分解
矩阵函数(Matrix Functions)	
expm	矩阵指数
funm	计算一般矩阵函数
logm	矩阵对数
sqrtm	矩阵平方根
低级函数(Low level Functions)	
qrdelete	从 QR 分解中删除列
qrinsert	在 QR 分解中加入列

数据分析和傅里叶变换(Data Analysis and FourierTransform Functions)

函　数	说　明
基本运算(Basic Operations)	
convhull	凸壳函数
cumprod	累计积
cumsum	累计和
cumtrapz	累计梯形积分
delaunay	Delaunay 三角化
dsearch	求最近点
factor	质数分解
inpolygon	搜索多边形内的点

max	求数组元素的最大值
mean	求数组的平均值
median	求数组的中间值
min	求数组元素的最小值
perms	求矢量所有可能排列
polyarea	多边形的面积
primes	生成质数列表
prod	数组元素积
sort	将元素按升序排列
sortrows	将行按升序排列
std	标准差
sum	求数组元素和
trapz	梯形数值积分
tsearch	搜索 Delaunay 三角形
voronoi	Voronoi 图
有限差分(Finite Differences)	
del2	五点 Laplacian 离散
diff	差分和近似微分
gradient	数值梯度
相关(Correlation)	
corrcoef	相关系数
cov	协方差矩阵
滤波和卷积(Filtering and Convolution)	
conv	卷积和多项式相乘
conv2	二维卷积
deconv	解卷积和多项式相除
filter	IIR 或 FIR 滤波
filter2	二维数字滤波

傅里叶变换(Fourier Transforms)	
abs	绝对值或模
angle	相角
cplxpair	矩阵按共轭对排列
fft	一维快速傅里叶变换
fft2	二维快速傅里叶变换
fftshift	移动 FFT 的零频成分至频谱中心
ifft	一维快速傅里叶逆变换
ifft2	二维快速傅里叶逆变换
ifftshift	FFT 逆移
nextpow2	最相邻的 2 的幂
unwrap	修正相角
向量函数(Vector Functions)	
cross	向量外积
intersect	两个向量求交集
ismember	检验集合中的元素
setdiff	求两个向量的差集
setxor	两个向量求异或
union	求两个向量的并集
unique	求向量的元素中的单一值向量

多项式和插值函数(Polynomials and Interpolation Functions)

函　数	说　明
多项式(Polynomials)	
conv	卷积和多项式相乘
deconv	多项式相除和解卷积
poly	求已知根的多项式的表达式

polyder	多项式的求导
polyeig	多项式的特征值问题
polyfit	多项式曲线拟合
polyval	多项式求值
polyvalm	求矩阵多项式的值
residue	求部分分式表达式
roos	多项式求根
数据插值(Data Interpolation)	
griddata	三维分格点数据
interp1	一维插值
interp2	二维插值
interp3	三维插值
interpft	一维 FFT 插值
interpn	多维插值
meshgrid	生成三维图的 X 矩阵和 Y 矩阵
ndgrid	生成多维函数和插值用数组
spline	立方样条插值
双重函数-非线性数值方法(Function Functions-Nonlinear Numerical Methods)	
dblquad	数值二重积分
fmin	求单变量函数极小值
fmins	求单变量函数极小值
fzero	求单变量函数 0 值
ode45、ode23	解微分方程
ode113、ode15s、ode23s、oder23t、ode23tb	解微分方程
odefile	为 ODE 解函数定义微分方程
odeget	由函数 odeset 选项结构获得属性
odeset	生成和修改结构选项作为 ODE 解函数的输入

quad，quad8	积分的数值解
vectorize	向量化表示

稀疏矩阵函数(Sparse Matrix Functions)

函　数	说　明
基本稀疏矩阵(Elementary Sparse Matrices)	
spdiags	提取和生成稀疏带和对角矩阵
speye	单位稀疏矩阵
sprand	均匀分布随机稀疏矩阵
sprandn	高斯分布随机稀疏矩阵
sprandsym	稀疏对称随机矩阵
满阵和稀疏矩阵的转换(Full to Sparse Conversion)	
find	寻找非 0 元素下标
full	把稀疏矩阵转换为满阵
sparse	生成稀疏矩阵
spconvert	转换外部格式并输入至稀疏矩阵
稀疏矩阵的非 0 元素操作(Working with Nonzero Engtries of Sparse Matrices)	
nnz	矩阵中非 0 元素的个数
nonzeros	非 0 元素
nzmax	为非 0 元素分配的存储空间
spalloc	为稀疏矩阵分配的存储空间
spfun	对非 0 元素进行函数计算
spones	把矩阵中的非 0 元素全用 1 代替
稀疏矩阵的可视化(Visualizing Sparse Matrices)	
spy	稀疏矩阵的图形表示

重新排序算法(Reordering Algorithms)	
colmmd	列最小度排序
colperm	基于非 0 元素统计进行列排序
dmerm	Dulmage-Mendelsohn 分解
randperm	随机排列一个整数向量
symmmd	最小对称度排序
symrcm	反向 Cuthill-McKee 排序
范数、条件数和秩(Norm, Condition Number, and Rank)	
condest	1 范数估计
normest	2 范数估计
线性方程的稀疏系统(Sparse Systems of Linear Equations)	
bicg	双共轭梯度法
bicgstab	双共轭梯度稳定法
cgs	二次共轭梯度法
cholinc	不完全 Cholesky 分解
gmres	广义最小残差法
luinc	不完全 LU 矩阵分解
pcg	预处理共轭梯度法
qmr	准最小残差法
qrdelete	从 QR 分解中消去列
qrinsert	在 QR 分解中插入列
稀疏矩阵的特征值和奇异值(Sparse Eigenvalues and Singular Values)	
eigs	求少数特征值和特征向量
svds	少数奇异值
spparms	设置稀疏矩阵程序的参数

声音处理函数(Sound Processing Functions)

函　数	说　明
常用声音函数(General Sound Functions)	
sound	把向量转换为声音
特殊声音函数(SPARC station-specific Sound Functions)	
auread	读 NeXT/SUN(.au)声音文件
auwrite	写 NeXT/SUN(.au)声音文件
WAV 声音文件(.WAV Sound Functions)	
wavread	读 Microsoft WAVE(.wav)声音文件
wavwrite	写 Microsoft WAVE(.wav)声音文件

字符串函数(Charcter String Functions)

函　数	说　明
常用函数(General)	
abs	绝对值或者模
eval	运行字符串所表示的表达式
real	复数的实部
strings MATLAB	字符串处理
字符操作(String Manipulation)	
deblank	去掉字符串末尾处的空格
findstr	查找字符串
lower	字符串转换为小写
strcat	字符串组合
strcmp	字符串比较
strcmpl	忽略小字的字符串比较
strjust	给出字符串最终结果

strmatch	查找字符串可能的匹配
strncmp	比较两个字符串的前 n 个字符
strrep	字符串查找和替换
strtok	查找某个字符最先出现位置
strvcat	字符串的竖向组合
upper	字符串转换为大写
字符串和数值的转换(String to Number Conversion)	
char	生成字符数组
int2str	把整数转换为字符串
mat2str	把矩阵转换为字符串
num2str	把数值转换为字符串
sprintf	格式输出字符串
sscanf	格式读入字符串
str2num	字符串转换为数值
基转换(Radix Conversion)	
bin2dec	把二进制转换为十进制
dec2bin	把十进制转换为二进制
dec2hex	把二进制转换为十六进制
hex2dec	把十六进制转换为二进制
hex2num	把十六进制转换为双精度

低级 I/O 和文件函数(Low-Level File I/O Functions)

函 数	说 明
文件打开和关闭(File Opening and Closing)	
fclose	关闭一个或多个文件
fopen	打开文件

无格式 I/O(Unformatted I/O)	
fread	从文件中读二进制数据
fwrite	向文件写二进制数据
格式(Formatted I/O)	
fgetl	按行从文件中读数据并不包括换行符
fgets	按行从文件中读数据并包括换行符
fprintf	把格式化数据写入文件
fscanf	从文件中读取格式化数据
文件定位(File Positioning)	
feof	测试文件结束标志
ferror	查询文件输入/输出的错误状态
frewind	反绕一个打开的文件
fseek	设置文件定位器指针
ftell	获取文件定位器指针位置
字符串操作(String Conversion)	
sprintf	把格式数据写入字符串
sscanf	从字符串中读入格式数据
特殊输入输出文件命令(Specialized File I/O)	
qtwrite	把 QuickTime 动画文件写入磁盘
dlmread	把 ASC I 中的数据读入矩阵
dlmwrite	把矩阵写入 ASC I 文件
hdf	HDF 界面
imfinfo	返回图形文件信息
imread	从图文件中读出图像数据
imwrite	把图像数据写入图形文件
wklread	从 Lotus1 2 3 Wk1 电子表格文件中读出矩阵
wklwrite	把矩阵写入 Lotus1 2 3 Wk1 电子表格文件
xlgetrange	从 Excel 表格中读出数据
xlesetrange	把数据写入 Excel 表格

位函数(Bitwise Functions)

函　数	说　明
bitand	位和
bitcmp	补码
bitor	位或
bitmax	最大浮点整数
bitset	给位赋值
bitshift	移位
bitget	获取位
bitxor	位异或

结构函数(Structure Functions)

函　数	说　明
fieldnames	结构的属性名称
getfield	获取结构数组的属性
rmfield	删除结构的属性
setfield	设置结构数组属性值
struct	生成结构数组
struct2cell	把结构数组转换为细胞数组

对象函数(Object Functions)

函　数	说　明
class	生成对象或者返回对象的类
isa	检验对象是否属于某类

细胞数组函数(Cell Array Functions)

函　数	说　明
cell	生成细胞数组
cellstr	用字符数组生成细胞数组
cell2struct	把细胞数组转换为结构数组
celldisp	显示细胞数组的内容
cellplot	用图形方式显示细胞数组的结构
num2cell	把数值数组转换为细胞数组

多维数组函数(Multidimensional Array Functions)

函　数	说　明
cat	数组组合
flipdim	沿定义的维数翻转数组
ind2sub	数组的下标转换
ipermute	数组维数逆变换
ndgrid	为多维函数和插值生成数组
ndims	数组的维数
permute	数组重新排列
reshape	数组变形
shiftdim	维数移位
squeeze	删除大小为 1 的维
sub2ind	转化为单下标

参考文献

[1] Signal Processing Toolbox. User's Guide. Version 4, MathWorks. Inc. ,1998

[2] (美)Ingle V K, Proakis J G. 数字信号处理及其MATLAB实现. 陈怀琛等译. 北京:电子工业出版社,1998

[3] 俞卞章等. 数字信号处理. 西安:西北工业大学出版社,1994

[4] 陈行禄等. 信号分析与处理. 北京:北京航空航天大学出版社,1993

[5] 郑君里等. 信号与系统. 北京:高等教育出版社,1996

[6] 卢文详等. 工程测试与信息处理. 武汉:华中理工大学出版社,1994

[7] 吴正毅. 测试技术与测试信号处理. 北京:清华大学出版社,1992