第5章 傅里叶分析和拉普拉斯分析的应用

CHAPTER 5

5.1 引言

通信是傅里叶分析的典型应用领域。本章用实例说明信号和系统的频率表示如何应用在通信中，并介绍在通信中使用的调制、带宽和频谱等概念。通信系统由三部分组成：发射机、信道和接收机。通信的目的是将消息通过信道由发射机传送至接收机。消息是一个信号，例如语音或音乐信号，一般包含的是低频率信号。消息的传输可以通过无线电波来完成，也可以通过一根连接发射机和接收机的导线来完成，或将二者结合起来，由此构成具有不同特性的信道。电话通信可以用导线，也可以不用导线，而无线电广播和电视都采用无线方式。在通信系统的分析和设计中，通过傅里叶变换建立起来的频率及带宽、频谱和调制等概念是最基础的内容。本章涉及的第一个课题是介绍有关通信中的问题，并将它与傅里叶分析联系起来，对这个课题的进一步分析，读者可参考通信领域方面的优秀书籍。本章涉及的另一个课题是模拟滤波器的设计。滤波是 LTI 系统在通信、控制和信号处理中非常重要的应用。本章介绍的这个课题的内容只是基础部分，我们将举例说明滤波器在设计和实现中与信号和系统有关的重要问题。

拉普拉斯变换在许多工程领域都有应用，尤其是在控制领域，本章将阐明拉普拉斯变换是如何与经典控制理论和现代控制理论相联系的。经典控制理论的目的是利用频域的方法改变已知系统的动力学特性，从而达到所期望的响应。这通常是由一个控制器反馈连接到一个装置上来完成的，该装置是一个系统，例如一台发动机、一个化学装置或一辆汽车，总之我们想要控制该系统使之以某种方式产生响应。控制器也是一个系统，它需要被设计成使该装置能够跟随一个预定的参考输入信号，通过将装置的响应反馈至输入端，就可以判断该装置如何对控制器作出响应。常用负反馈产生一个误差信号，根据这个误差信号能够对控制器的性能作出评价。传输函数、系统稳定性和利用拉普拉斯变换获得各种不同的响应类型等概念在经典控制系统的分析和设计中非常有用。

现代控制理论与此不同，它用时域的方法来表征和控制系统。状态变量的表示是一种比转移函数更通用的表示方法，因为它允许包含初始条件，而且能够方便地推广到多输入多输出的一般情况中。状态变量理论与线性代数和微分方程有着密切的联系。本章将介绍状态变量的概念、状态变量与转移函数的关系、拉普拉斯变换在求全响应中的应用及拉普拉斯

变换在由状态方程和输出方程求转移函数中的应用。本章的目的是引出经典控制理论和现代控制理论中的一些问题，并将它们与拉普拉斯分析相关联。关于其更深入的分析可以在很多控制论的优秀著作中找到。


信号的时域抽样定理

5.2 连续信号的抽样定理

前面各章讨论的连续时间信号也称为模拟信号，此类信号实际上是模拟欲传输的信息而得到的一种电流或电压。由于受诸多因素的限制，一般模拟信号的加工处理质量不高。而数字信号仅用0、1来表示，它的加工处理相较于模拟信号有着无可比拟的优越性，因而受到广泛重视。随着数字技术及电子计算机技术的迅速发展，数字信号处理得到越来越广泛的应用，电子设备的数字化也已成为一种发展方向。

要得到数字信号，往往首先要对表示信息的模拟信号进行抽样，从而得到一系列离散时刻的样值信号，然后对此离散时刻的样值信号进行量化、编码，就可得到数字信号。可见，这里的一个关键环节就是抽样。现在的问题是，从模拟信号 $f(t)$ 中经抽样得到的离散时刻的样值信号 $f_s(t)$ 是否包含了 $f(t)$ 的全部信息，即从离散时刻的样值信号 $f_s(t)$ 能否恢复原来的模拟信号 $f(t)$？抽样定理正是说明这样一个重要问题的定理，它在通信理论中占有相当重要的地位。

信号 $f(t)$ 抽样的工作原理可用图5-1表述。抽样器相当于一个定时开关，它每隔 T_s 秒闭合一次，每次闭合时间为 τ 秒，从而得到样值信号 $f_s(t)$。

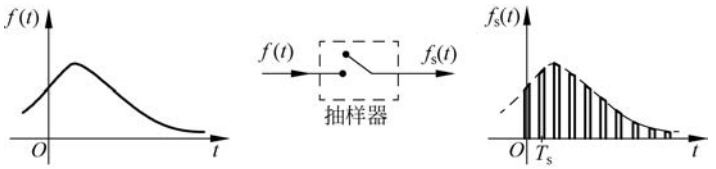

图5-1 连续信号的抽样

图5-1所示的抽样原理从理论上分析可表述为 $f(t)$ 与抽样脉冲序列 $P_{T_s}(t)$ 的乘积，即

$$f_s(t)=f(t)\cdot P_{T_s}(t) \tag{5-1}$$

式中的抽样脉冲序列 $P_{T_s}(t)$ 如图5-2所示。它实际上就是以前所讨论过的周期矩形脉冲函数，可表示为

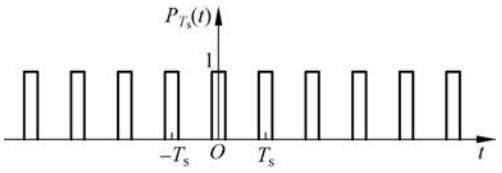

图5-2 抽样脉冲序列

$$P_{T_s}(t)=\sum_{n=-\infty}^{\infty} g_\tau(t-nT_s) \tag{5-2}$$

如果抽样脉冲序列是周期冲激函数序列 $\delta_{T_s}(t)$，则抽样得到的样值函数也为一个冲激函数序列，其各个冲激函数的冲激强度为该时刻 $f(t)$ 的瞬时值。这种抽样称为理想抽样，理想抽样的过程及有关波形如图5-3所示。

1. 抽样定理

连续时间信号 $f(t)$ 的时域抽样定理可表述为：最高频率为 f_m(Hz)的带限信号，由它在均匀间隔上的抽样值唯一决定，只要其抽样间隔 T_s 小于或等于 $\dfrac{1}{2f_m}$(s)。

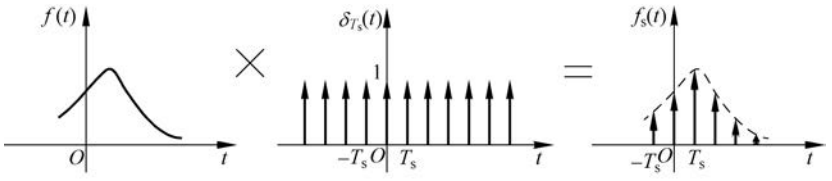

图 5-3　周期冲激函数序列抽样

由抽样定理可知，要求被抽样的信号 $f(t)$ 为带限信号，即频带有限的信号，其最高频率为 f_m，最高角频率 $\omega_m=2\pi f_m$，即当 $|\omega|>\omega_m$ 时，$F(j\omega)=0$。带限信号及其频谱示意图如图 5-4 所示。

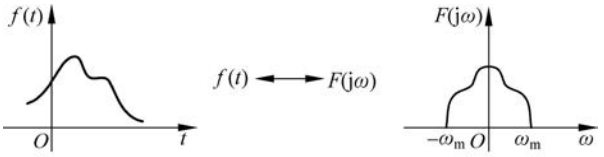

图 5-4　带限信号及其频谱

设信号 $f(t)$ 为带限信号，其最高频率为 f_m，最高角频率为 $\omega_m=2\pi f_m$，即当 $|\omega|>\omega_m$ 时，$F(j\omega)=0$。带限信号 $f(t)$ 经过理想抽样后的时域及频域波形如图 5-5 所示。

$$f_s(t)=f(t)\cdot\delta_{T_s}(t)=f(t)\cdot\sum_{n=-\infty}^{\infty}\delta(t-nT_s)=\sum_{n=-\infty}^{\infty}f(nT_s)\delta(t-nT_s)\quad(5\text{-}3)$$

$f_s(t)$ 为每隔 T_s 秒均匀抽样而得到的样值函数，它是一个冲激函数序列，各冲激函数的冲激强度为该时刻 $f(t)$ 的值。

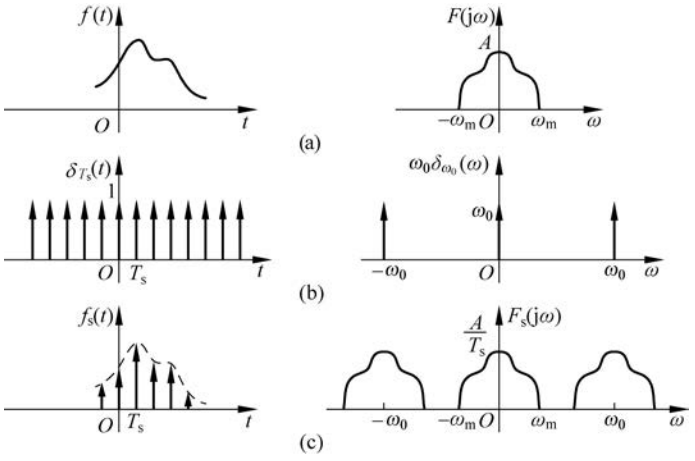

图 5-5　信号抽样分别在时域和频域的过程

由前面章节的内容可知，周期冲激函数 $\delta_{T_s}(t)$ 的频谱密度函数为

$$\omega_0\delta_{\omega_0}(\omega)=\omega_0\sum_{n=-\infty}^{\infty}\delta(\omega-n\omega_0),\quad \omega_0=\frac{2\lambda}{T_s}\qquad(5\text{-}4)$$

由于 $f_s(t)=f(t)\cdot\delta_{T_s}(t)$，根据傅里叶变换的频域卷积性质，有

$$\mathcal{F}[f_s(t)]=\frac{1}{2\pi}[F(j\omega)*\omega_0\sum_{n=-\infty}^{\infty}\delta(\omega-n\omega_0)]=\frac{\omega_0}{2\pi}\sum_{n=-\infty}^{\infty}[F(j\omega)*\delta(\omega-n\omega_0)]$$

$$=\frac{1}{T_s}\sum_{n=-\infty}^{\infty}F(j(\omega-n\omega_0)) \tag{5-5}$$

如图 5-5 所示，只要 $\omega_0\geqslant 2\omega_m$，样值函数 $f_s(t)$ 的频谱 $F_s(j\omega)$ 就周期性地重复着 $F(j\omega)$，而不会发生重叠。

由于要求 $\omega_0\geqslant 2\omega_m$，即 $\frac{2\pi}{T_s}\geqslant 4\pi f_m$，可得等效条件为 $T_s\leqslant\frac{1}{2f_m}$，我们把最大允许的抽样间隔 $T_s=\frac{1}{2f_m}$ 称为奈奎斯特间隔，把最低允许的抽样频率 $f_s=2f_m$ 称为奈奎斯特频率。

上面的分析表明，只要以小于奈奎斯特间隔 $\frac{1}{2f_m}$ 秒的间隔对信号 $f(t)$ 均匀抽样，那么得到的样值函数 $f_s(t)$ 的频谱密度函数 $F_s(j\omega)$ 就是 $F(j\omega)$ 的周期性复制，因而样值函数 $F_s(j\omega)$ 就包含了 $f(t)$ 的全部信息。

2. $f(t)$ 的恢复

当样值函数 $f_s(t)$ 经过一个截止频率为 ω_m 的理想低通滤波器，就可从 $F_s(j\omega)$ 中取出 $F(j\omega)$，从时域来说，这样就恢复了连续时间信号 $f(t)$，即

$$F(j\omega)=F_s(j\omega)\cdot H(j\omega) \tag{5-6}$$

式中，$H(j\omega)$ 为理想低通滤波器的频率特性。$H(j\omega)$ 的特性为

$$H(j\omega)=\begin{cases}T_s, & |\omega|\leqslant\omega_m\\ 0, & |\omega|>\omega_m\end{cases} \tag{5-7}$$

信号的恢复过程如图 5-6 所示。

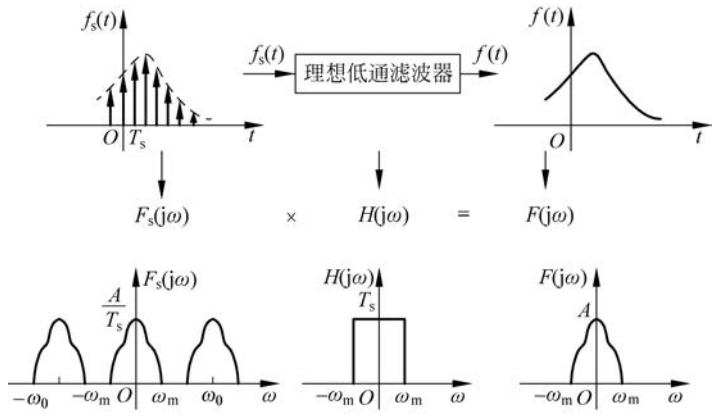

图 5-6　信号的恢复过程

3. 周期矩形脉冲抽样

周期矩形脉冲抽样可表示为

$$f_s(t)=f(t)\cdot P_{T_s}(t) \tag{5-8}$$

$$P_{T_s}(t)\leftrightarrow\frac{2\pi\tau}{T_s}\sum_{n=-\infty}^{\infty}\mathrm{Sa}\left(\frac{n\omega_0\tau}{2}\right)\delta(\omega-n\omega_0) \tag{5-9}$$

由于 $f_s(t)=f(t)\cdot P_{T_s}(t)$，同样，根据傅里叶变换的频域卷积性质，可得

$$\mathcal{F}[f_s(t)]=\frac{1}{2\pi}\left[F(\mathrm{j}\omega)*\sum_{n=-\infty}^{\infty}\frac{2\pi\tau}{T_s}\mathrm{Sa}\left(\frac{n\omega_0\tau}{2}\right)\delta(\omega-n\omega_0)\right]$$

$$=\frac{\tau}{T_s}\sum_{n=-\infty}^{\infty}\mathrm{Sa}\left(\frac{n\omega_0\tau}{2}\right)F[\mathrm{j}(\omega-n\omega_0)] \tag{5-10}$$

周期矩形脉冲抽样原理如图 5-7 所示。

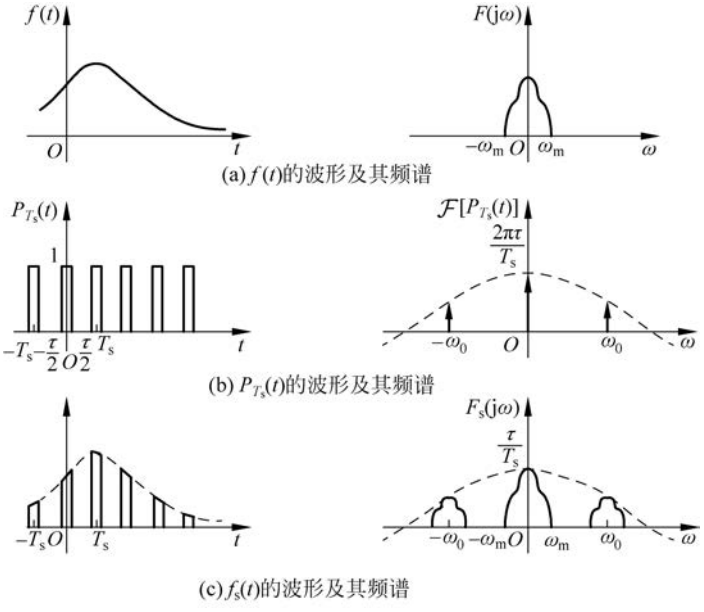

图 5-7　周期矩形脉冲抽样

5.3　傅里叶分析在通信系统中的应用

本节涉及滤波器的设计。滤波是 LTI 系统在通信、控制和信号处理中非常重要的应用。本节将举例说明滤波器设计和实现中与信号和系统有关的重要问题。

5.3.1　信号的调制和解调

两个信号在时域的乘法运算通常用来实现信号的调制，即由一个信号去控制另一个信号的某一个参量。信号的调制在通信领域的应用非常广泛。例如用一个低频的正弦波信号去控制另一个频率较高的正弦波信号的幅值，则产生一个振幅调制信号，又称调幅波。与产生调幅波的原理相似，还可以产生调频波、调相波和脉冲调制波等。

使用 Modulate 函数来实现信号的调制。有下面两种调用格式：

(1) Y=Modulate(X,Fc,Fs,METHOD,OPT)：其中频率为 Fc，采样频率为 Fs，载波调制原信号为 X，且 Fs>2×Fc+BW，BW 为原信号 X 的带宽，METHOD 为调制方法，例如调频 FM、调幅 AM、调相 PM，OPT 为额外可选的参数，具体由调制方法而定。

(2) [Y,T]=Modulate(X,Fc,Fs,METHOD,OPT)：这里多出的 **T** 为与 Y 同长的时间向量。

相关 MATLAB 程序如下：

```
% 产生调制信号(AM)
clf;
n = [0:256];Fc = 100000;Fs = 1000000;N = 1000;
xn = abs(sin(2 * pi * n/256));
xf = abs(fft(xn,N));y2 = modulate(xn,Fc,Fs,'am');
subplot(211); plot(n(1:200),y2(1:200));
xlabel('时间(s)');ylabel('幅值');title('调幅信号');
yf = abs(fft(y2,N));
subplot(212);stem(yf(1:200));
xlabel('频率(Hz)');ylabel('幅值');
```

调制信号产生的调幅信号结果如图 5-8 所示。

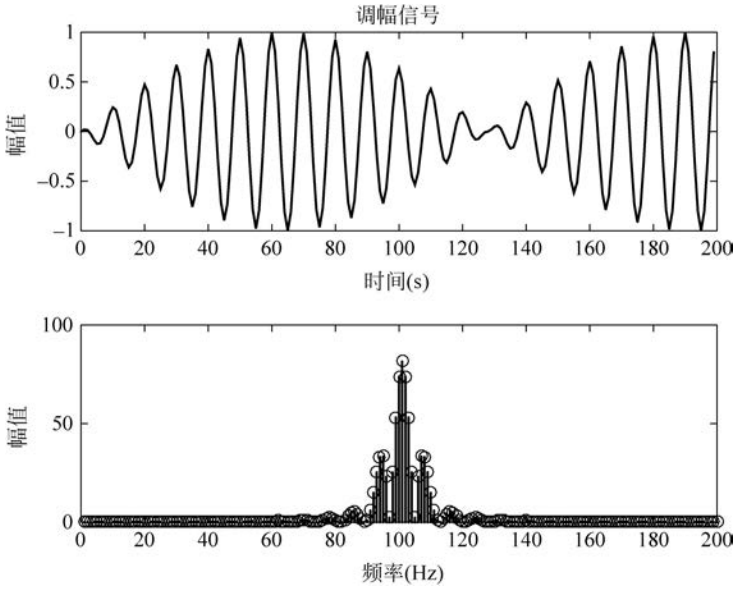

图 5-8　调幅信号及其频谱

在上面程序的基础之上加上以下程序段可以实现解调的操作。

```
% 解调(AM)
xo = demod(y2,Fc,Fs,'am');
figure; subplot(211);
plot(n(1:200),xn(1:200)); title('原信号'); subplot(212)
plot(n(1:200),2 * xo(1:200));
title('解调信号'); axis([1 200 0 1]);
```

解调结果如图 5-9 所示。

```
% 产生调制信号(FM)
clc;close all;clear;
n = [0:256];Fc = 100000;Fs = 1000000;N = 1000;
xn = abs(sin(2 * pi * n/256));
xf = abs(fft(xn,N));
y2 = modulate(xn,Fc,Fs,'fm');
subplot(211);
plot(n(1:200),y2(1:200));
xlabel('时间(s)');ylabel('幅值');
```

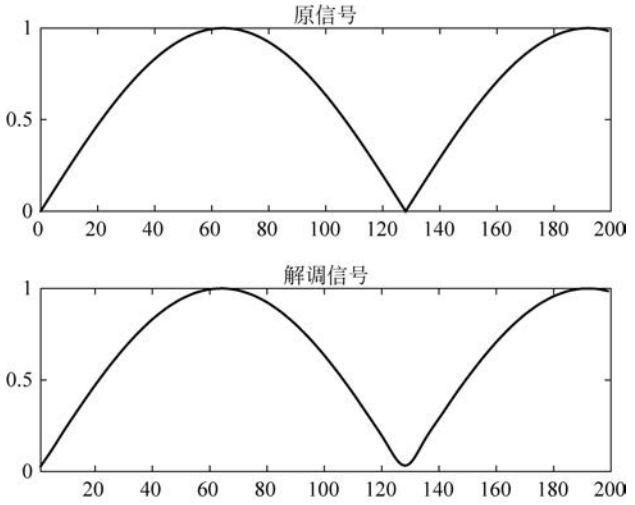

图 5-9　原信号与解调信号

```
title('调频信号'); yf = abs(fft(y2,N));
subplot(212);stem(yf(1:200));
xlabel('频率(Hz)');ylabel('幅值');
```

调制信号产生的调频信号结果如图 5-10 所示。

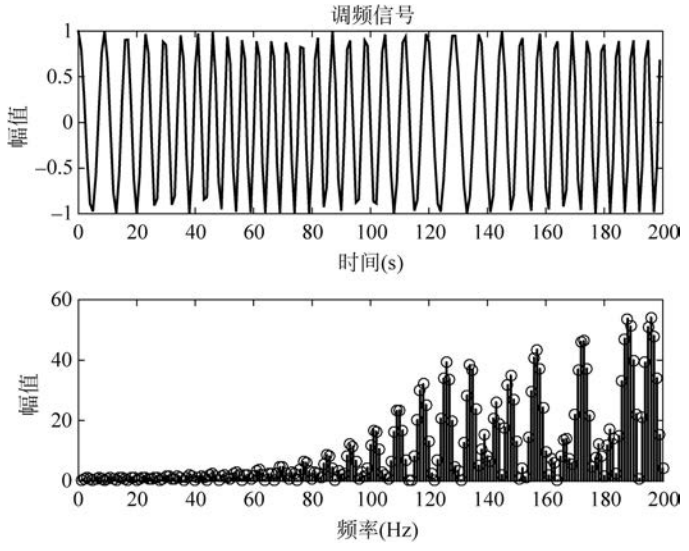

图 5-10　调频信号及其频谱

在上面程序的基础之上加上以下程序段可以实现信号解调的操作。

```
% 解调(FM)
xo = demod(y2,Fc,Fs,'fm');
figure
subplot(211)
plot(n(1:200),xn(1:200));
title('原信号');subplot(212)
plot(n(1:200),1.6 * xo(1:200));
title('解调信号');axis([1 200 0 1]);
```

程序产生的图形和 AM 解调类似，这里不再给出。

```
% 产生调制信号(PM)
clc;close all;clear;
n = [0:256];Fc = 100000;Fs = 1000000;N = 1000;
xn = abs(sin(2 * pi * n/256));
xf = abs(fft(xn,N)); y2 = modulate(xn,Fc,Fs,'pm');
subplot(211);plot(n(1:200),y2(1:200));
xlabel('时间(s)');ylabel('幅值');
title('调相信号'); yf = abs(fft(y2,N));
subplot(212);stem(yf(1:200));xlabel('频率(Hz)');ylabel('幅值');
```

调制信号产生的调相信号及其频谱结果如图 5-11 所示。

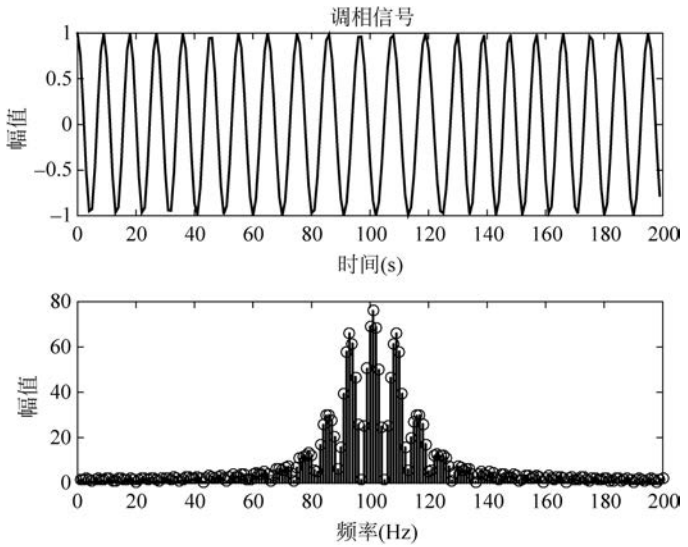

图 5-11　调相信号及其频谱

在上面程序的基础之上加上以下程序段可以实现解调的操作。

```
% 解调(PM)
xo = demod(y2,Fc,Fs,'pm');
figure
subplot(211)
plot(n(1:200),xn(1:200));
title('原信号');subplot(212);x0 = x0/3.15
plot(n(1:200),xo(1:200));
title('解调信号');axis([1 200 0 1]);
```

程序产生的图形和 AM 解调类似，这里不再给出。

5.3.2　信号的抽样和恢复

信号抽样和恢复的原理在 5.2 节中已经描述清楚，这里不再重复。为了便于描述，我们采用单一频率的正弦信号作为带限信号进行抽样和恢复。

相关 MATLAB 程序如下：

```
%正弦信号采样
t = 0:0.0005:1;
f = 13;
xa = cos(2 * pi * f * t);
subplot(2,1,1);
plot(t,xa);
xlabel('时间(ms) ');ylabel('幅值');
title('连续时间信号 xa(t) ');
axis([0 1 -1.2 1.2])
subplot(2,1,2);
T = 0.1;n = 0:T:1;
xs = cos(2 * pi * f * n);
k = 0:length(n) - 1;
stem(k,xs);
xlabel('时间(ms) ');ylabel('幅值');
title('离散时间信号 x(n)');
axis([0 (length(n) - 1) -1.2 1.2]);
```

正弦信号的采样结果如图 5-12 所示。

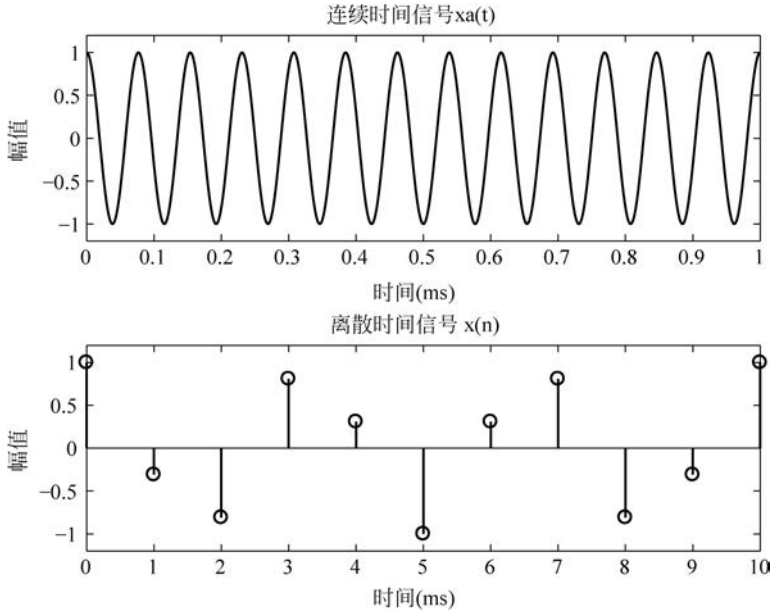

图 5-12　正弦信号及其样值信号

```
%采样与重构
T = 0.1;f = 13;
n = (0:T:1)';
xs = cos(2 * pi * f * n);
t = linspace( - 0.5,1.5,500)';
ya = sinc((1/T) * t(:,ones(size(n))) - (1/T) * n(:,ones(size(t)))') * xs;
plot(n,xs, 'o ',t,ya);
xlabel('时间(ms)'); ylabel('幅值');
title('重构连续信号 y_{a}(t)');
axis([0 1 -1.2 1.2]);
```

正弦信号的采样与重构结果如图 5-13 所示。

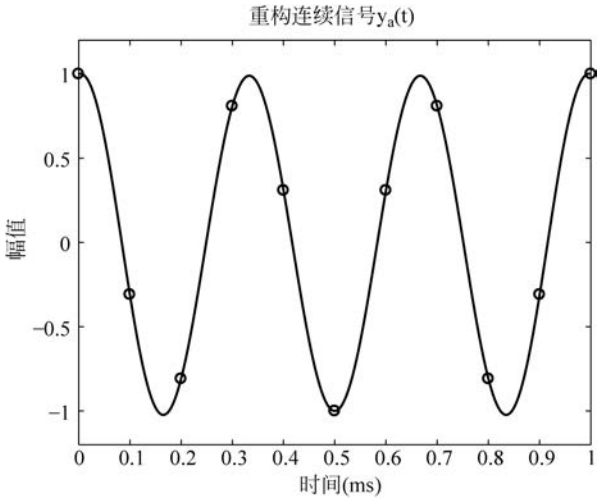

图 5-13　重构信号的波形

```
% 采样的性质
t = 0:0.005:10;
xa = 2 * t. * exp( - t);
subplot(2,2,1)
plot(t,xa);
xlabel('时间(ms)');ylabel('幅值');
title('连续时间信号 x_{a}(t) ');
subplot(2,2,2)
wa = 0:10/511:10;
ha = freqs(2,[1 2 1],wa);
plot(wa/(2 * pi),abs(ha));
xlabel('频率(kHz)'); ylabel('幅值');
title('|X_{a}(jw)| ');
axis([0 5/pi 0 2]);
subplot(2,2,3)
T = 1;
n = 0:T:10;
xs = 2 * n. * exp( - n);
k = 0:length(n) - 1;
stem(k,xs);
xlabel('时间(ms)');ylabel('幅值');
title('离散时间信号 x(n)');
subplot(2,2,4)
wd = 0:pi/255:pi;
hd = freqz(xs,1,wd);
plot(wd/(T * pi),T * abs(hd));
xlabel('频率(kHz)'); ylabel('幅值');
title('|X(e^{jw})|');
axis([0 1/T 0 2])
```

信号采样的性质仿真如图 5-14 所示。

```
% 模拟低通滤波器设计
Fp = 3500; Fs = 4500;
```

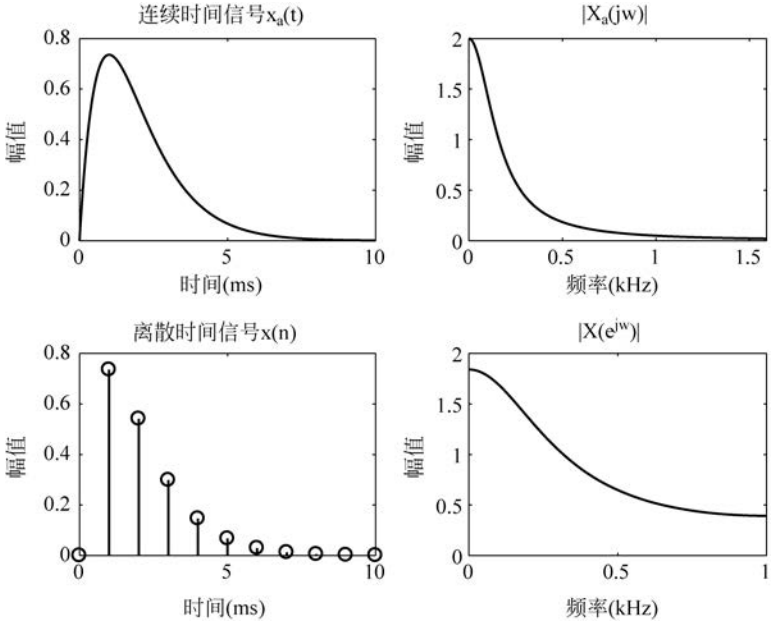

图 5-14　信号采样的性质仿真

```
Wp = 2 * pi * Fp; Ws = 2 * pi * Fs;
[N,Wn] = buttord(Wp,Ws,0.5,30, 's');
[b,a] = butter(N,Wn, 's');
wa = 0:(3 * Ws)/511:3 * Ws;
h = freqs(b,a,wa);
plot(wa/(2 * pi),20 * log10(abs(h)));grid
xlabel('Frequency(Hz)'); ylabel('Gain(dB)');
title('Gain Response');
axis([0 3 * Fs  - 60 5]);
```

模拟低通滤波器的设计结果如图 5-15 所示。

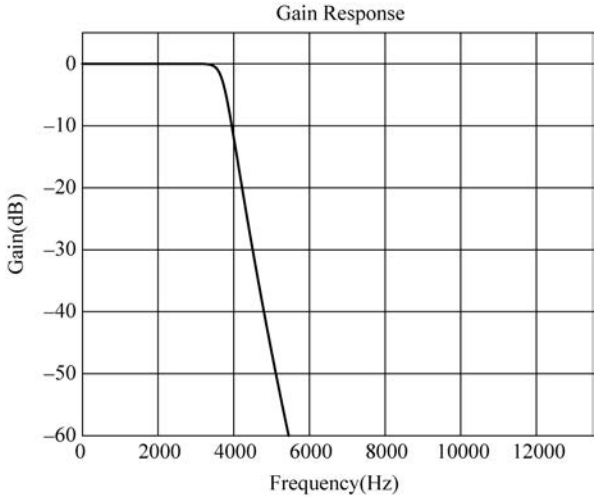

图 5-15　模拟低通滤波器

5.3.3 数字滤波器设计

对于数字信号滤波来说，模拟其过程，就是通过将输入数值序列和代表滤波器本身性质的冲激响应序列做卷积。根据冲激响应序列有限长或无限长，分为有限冲激响应（FIR）滤波器和无限冲激响应（IIR）滤波器。

FIR 的差分方程为

$$a(1)\times y(k)=b(1)\times f(k)+b(2)\times f(k-1)+\cdots+b(nb+1)\times f(k-nb) \quad (5\text{-}11)$$

IIR 的差分方程为

$$\begin{aligned}&a(1)\times y(k)+a(2)\times y(k-1)+\cdots+a(na+1)\times y(k-na)\\=&b(1)\times f(k)+b(2)\times f(k-1)+\cdots+b(nb+1)\times f(k-nb)\end{aligned} \quad (5\text{-}12)$$

在 MATLAB 中，可以用[b,a]=butter(N,Wn)等函数辅助设计 IIR 数字滤波器，也可以用 b=fir1(N,Wn,'ftype')等函数辅助设计 FIR 数字滤波器。

涉及函数有

[B,A]=butter(3,2/0.00025,'s')

[num2,den2]=bilinear(B,A,4000)

Wn=kaiser(30,4.55)

以下是某个 FIR 滤波器设计的 MATLAB 程序。

```
% 文件:FIRFilterExample.m
fscale = 1;fshift = 0.0;dscale = 1000; % 比例参数
c9_Filter_Data; % 载入数据
Freq_Resp = data;fs = 900;filtsize = 512;ts = 1/fs;
[himptime] = FIR_Filter_AMP_Delay(Freq_Resp,fs,filtsize,fscale,fshift,dscale);
% 使用窗[HJ2.4mm]
nw = 256;window1 = hamming(nw);window = zeros(filtsize,1);
% 确保这个窗是个合适的中心窗
wstart = (filtsize/2) - (nw/2);wend = (filtsize/2) + (nw/2) - 1;
window(wstart:wend) = window1;
impw = himp. * window';
figure;subplot(1,2,1);plot(abs(himp));grid;
xlabel('TimeSampleIndex');ylabel('FilterImpulseResponse');
subplot(1,2,2);plot(abs(impw));grid;
xlabel('TimeSampleIndex');ylabel('WindowedFilterImpulseResponse');
[logpsd,freq,ptotal,pmax] = log_psd(himp,filtsize,ts);
[logpsdw,freq,ptotal,pmax] = log_psd(impw,filtsize,ts);
figure;subplot(1,2,1)
plot(freq(128:384),logpsd(128:384));grid;
xlabel('FrequencySampleIndex');ylabel('FrequencyResponse');
subplot(1,2,2)
plot(freq(128:384),logpsdw(128:384));grid;
xlabel('FrequencySampleIndex');
ylabel('WindowedFrequencyResponse');
% 函数文件结束
% 文件:FIR_Filter_AMP_Delay.m
function[h,times] = FIR_Filter_AMP_Delay(H,fs,n,fscale,fshift,dscale)
% 这个函数返回 FIR 滤波器的脉冲响应
% h 为脉冲响应值在 t = times 时的行向量
% h 被平移到中心，脉冲响应阵列为 n/2 * ts
```

```
% 假设在给定的频率响应里没有固定时延
% H 是频率响应的一个阵列
% Column1:[JP3]平移和比例缩放使频率 f 必须在 - fs/2～fs/2,然后频率 fk 顺序上升
% Column2:20 * log(|H(fk)|);
% Column3:群时延单位 1/frequency
% (i.e:如果频率的单位是 MHz,那么时延的单位应该是 ms)
% 另外用 dscale 来调节时延 delay = delay/dscale
% Ex:如果 delay 以 ns 来定义,那么 delayms = delayns/1000
% 相位响应在 f = 0 通过积分时延来获得
% Phase((k + 1)df) = phase(kdf) + 2 * pi * (fs/nfft)delay(kdf)[HJ2.2mm]
% = phase(kdf) + (2 * pi * /nfft)(delay(kdf)/ts)
% fscaleandfshift:f = (f - fshift)/fscale
% fs:抽样率
% n:脉冲响应持续周期;频率响应从( - fs/2) + df/2 到 fs/2 - df/2,使用 df = fs/n 再次抽样
ts = 1/fs;df = fs/n;
% 获得频率、幅度和相位响应阵列
% 转换 dbs 为实数;重置频率
Hfreq = H(:,1);Hmag = H(:,2);Hdelay = H(:,3);
nn = max(size(Hmag));Hreal = 10.^(Hmag/20);
Hfreq = (Hfreq - fshift)/fscale;Hdelay = Hdelay/dscale;
% 给频率和时间设置阵列索引
index1 = [0:1:(n/2)];index2 = [ - (n/2) + 1:1: - 1];index = [index1index2]';
frequencies = (index * df);times = index * ts;
% 用 FFT 循环移位来改变平移时间索引
times = shift_ifft(times,n);
% 内插过程中给频率响应数据在 - fs/2 和 fs/2 端加上两个以上的输入
fmin = min(min(frequencies));fmax = max(max(frequencies));
% 通过扩展频率和其他阵列的变换使其位于 - fs/2 + df 和 fs/2 + df 之间
Hfreq1 = Hfreq;Hreal1 = Hreal;Hdelay1 = Hdelay;
if fmin < Hfreq(1,1) % 如果低级结果没有扩展到 - fs/2,就加一个点
Hfreq1 = [fmin;Hfreq];
Hreal1 = [1e - 10;Hreal];
Hdelay1 = [Hdelay(1,1);Hdelay];
end
if fmax > Hfreq(nn,1) % 如果高级结果没有扩展到 fs/2,就加一个点
Hfreq1 = [Hfreq1;fmax];
Hreal1 = [Hreal1;1e - 10];
Hdelay1 = [Hdelay1;Hdelay(nn,1)];
end
% 插入频率响应数据并计算复数
% 转移函数 mag * exp(i * phase)
Hreal_interpolated = interp1(Hfreq1,Hreal1,frequencies);
Hdelay_interpolated = interp1(Hfreq1,Hdelay1,frequencies);
% 积分时延来得到相位响应
sum = 0.;
Hphase(1) = 0.; % 载波相位的频率为 0
for k = 2:(n/2) + 1 % 从 f > 0 求积分
sum = sum - (Hdelay_interpolated(k,1)/ts) * (2 * pi/n);
Hphase(k,1) = sum;
end
sum = 0.0;
for k = n: - 1:(n/2) + 2 % 从 f < 0 求积分
sum = sum + (Hdelay_interpolated(k,1)/ts) * (2 * pi/n);
Hphase(k,1) = sum;
```

```
end
Hcomplex = Hreal_interpolated. * exp(i * Hphase);
% 得出逆 FFT 并进行平移
hh = ifft(Hcomplex);h = (shift_ifft(hh,n));
% 滤波器的设计及函数文件结束
% 函数文件:shift_ifft.m
functiony = shift_ifft(x,n)
% 循环移位 IFFT 阵列
for k = 1:(n/2) - 1
y(k) = x((n/2) + k + 1);
end
for k = 1:n/2 + 1
y((n/2) - 1 + k) = x(k);
end
% 函数文件结束
% 函数文件:log_psd.m
function[logpsd,freq,ptotal,pmax] = log_psd(x,n,ts)
% 这个函数进行 n 点时域抽样(实数或者复数)并且通过执行(fft/n)^2 找出 psd,这个双边谱是由
% psd 的移位产生的;阵列频率给绘图提供了合适的频率值,通过计算 10 * log10(psd/max(psd)),
% psd 被归一化;值低于 60dB 时,被设置为 - 60dB
% n 必须是一个偶数,最好是 2 的幂
y = zeros(1,n); % 初始化 y 向量
h = waitbar(0,'ForLoopinPSDCalculation');
for k = 1:n
freq(k) = (k - 1 - (n/2))/(n * ts);
y(k) = x(k) * (( - 1.0)^k);
waitbar(k/n)
end;
v = fft(y)/n;
psd = abs(v).^2;
pmax = max(psd);
ptotal = sum(psd);
logpsd = 10 * log10(psd/pmax);
% 在 - 60dB 处截断负值
for k = 1:n
if(logpsd(k)< - 60.0)
logpsd(k) = - 60.0;
end
end
close(h)
% 函数文件结束
```

5.4 拉普拉斯分析在经典控制系统中的应用

5.4.1 控制系统的数学模型

设单输入单输出线性时不变连续系统的输入信号为 $f(t)$，输出信号为 $y(t)$，则其微分方程的一般形式为

$$a_n\frac{\mathrm{d}^n y(t)}{\mathrm{d}t^n}+a_{n-1}\frac{\mathrm{d}^{n-1} y(t)}{\mathrm{d}t^{n-1}}+\cdots+a_1\frac{\mathrm{d}y(t)}{\mathrm{d}t}+a_0$$
$$=b_m\frac{\mathrm{d}^m f(t)}{\mathrm{d}t^m}+b_{m-1}\frac{\mathrm{d}^{m-1} f(t)}{\mathrm{d}t^{m-1}}+\cdots+b_1\frac{\mathrm{d}f(t)}{\mathrm{d}t}+b_0 \tag{5-13}$$

式中，系数 $a_0,a_1,\cdots,a_n$ 和 $b_0,b_1,\cdots,b_m$ 为实常数，且 $m\leqslant n$。

对式(5-13)在零初始条件下求拉普拉斯变换，并根据传递函数的定义可得单输入单输出系统传递函数的一般形式为

$$H(s)=\frac{\mathcal{L}[y(t)]}{\mathcal{L}[f(t)]}=\frac{b_m s^m+b_{m-1}s^{m-1}+\cdots+b_1 s+b_0}{a_n s^n+a_{n-1}s^{n-1}+\cdots+a_1 s+a_0}=\frac{B(s)}{A(s)} \tag{5-14}$$

式中：$B(s)=b_m s^m+b_{m-1}s^{m-1}+\cdots+b_1 s+b_0$ 为传递函数的分子多项式；

$A(s)=a_n s^n+a_{n-1}s^{n-1}+\cdots+a_1 s+a_0$ 为传递函数的分母多项式，也称为系统的特征多项式。

在 MATLAB 中，控制系统的分子多项式系数和分母多项式系数分别用 num 和 den 表示，即

$$\text{num}=[b_m,b_{m-1},\cdots,b_1,b_0],\quad \text{den}=[a_n,a_{n-1},\cdots,a_1,a_0]$$

以传递函数表示系统的形式叫作传递函数模型。

5.4.2　控制系统数学模型的建立

在 MATLAB 中，使用函数 tf()建立或转换控制系统的传递函数模型。其功能和主要格式如下。

功能：生成线性定常连续/离散系统的传递函数模型，或者将状态空间模型或零极点增益模型转换成传递函数模型。

格式：

sys=tf(num, den)　生成传递函数模型 sys。

sys=tf(num,den,'Property1',Value1, …,'PropertyN',ValueN)　生成传递函数模型 sys。模型 sys 的属性(Property)及属性值(Value)用'Property'及 Value 指定。

sys=tf(num, den, Ts)　生成离散时间系统的脉冲传递函数模型 sys。

sys=tf(num,den,Ts,'Property1',Value1,…,'PropertyN',ValueN)　生成离散时间系统的脉冲传递函数模型 sys。

sys=tf('s')　指定传递函数模型以拉普拉斯变换算子 s 为自变量。

sys=tf('z',Ts)　指定脉冲传递函数模型以 Z 变换算子 z 为自变量，以 Ts 为采样周期。

tfsys=tf(sys)　将任意线性定常系统 sys 转换为传递函数模型 tfsys。

说明：①对于单输入单输出系统，num 和 den 分别为传递函数的分子向量和分母向量，对于多输入多输出系统，num 和 den 为行向量的元胞数组，其行数与输出向量的维数相同，列数与输入向量的维数相同；②Ts 为采样周期，若系统的采样周期未定义，则设置 Ts=−1 或者 Ts=[ ]；③缺省情况下，生成连续时间系统的传递函数模型，且以拉普拉斯变换算子 s 为自变量。

5.4.3　控制系统数学模型参数的获取

应用 MATLAB 建立了系统模型后，MATLAB 会以单个变量形式存储该模型的数据，包括模型参数(如状态空间模型的 **A**,**B**,**C**,**D** 矩阵)等属性，例如输入/输出变量名称、采样周期、输入/输出延迟等。有时需要从已经建立的线性定常系统模型(如传递函数模型、零极点增益模型、状态空间模型或频率响应数据模型)中获取模型参数等信息，此时除了使用函

数 set()和函数 get()以外，还可以采用模型参数来达到目的。由线性定常系统的一种模型可以直接得到其他几种模型的参数，而不必进行模型之间的转换。

下面以函数 zpkdata()为例，说明模型参数获取函数的使用方法。该函数的调用格式如下：

[z，p，k]=zpkdata(sys)　返回由 sys 所示线性定常系统零极点增益模型的零点向量 **z**，极点向量 **p** 和增益 k。

说明：① 为了方便多输入多输出模型或模型数组数据的获取，缺省情况下，函数 tfdata()和 zpkdata()以元胞数组形式返回参数（例如 num，den，z，p 等）；

② 对于单输入单输出模型而言，可在调用函数时应用第二个输入变量'v'，指定调用该函数时返回的是向量（Vector）数据而不是元胞数组。

也可以采用一条 MATLAB 命令直接显示零点向量和极点向量，即

```
>>[z1,p1,k] = zpkdata(tf(num,den),'v')
```

运行后，得到系统的零点向量 **z**1、极点向量 **p**1 和增益 k 与前述相同。

习题

5-1　对下列信号求奈奎斯特间隔和频率：

（1）Sa(100t)；　（2）Sa^2(100t)；　（3）Sa(100t)+Sa(50t)；　（4）Sa(100t)+Sa^2(60t)。

5-2　已知信号 $f(t)=\dfrac{\sin(4\pi t)}{\pi t}$，当对该信号取样时，求能恢复原信号的最大取样周期。设计 MATLAB 程序进行分析并给出结果。

5-3　设计一模拟信号：$x(t)=3\sin(2\pi ft)$。采样频率 f_s 为 5120Hz，取信号频率 $f=150$Hz（正常采样）和 $f=3000$Hz（欠采样）两种情况进行采样分析。设计 MATLAB 程序进行分析并给出结果。

5-4　已知 $H(s)=\dfrac{s+2}{s^3+2s^2+2s+1}$，分别利用（1）求系统响应的方法；（2）求系统特征值的方法，确定系统的稳定性。设计 MATLAB 程序进行分析并给出结果。

5-5　已知控制系统的传递函数为 $H(s)=\dfrac{10(s+1)}{s(s+2)(s+5)}$，用 MATLAB 建立其数学模型。

5-6　线性定常连续系统的传递函数为 $H(s)=\dfrac{s^2+3s+2}{s^3+5s^2+7s+3}$，应用 MATLAB 建立其零极点增益模型。