第5章 连续时间信号与系统的频域分析

对于连续时间信号的频域分析，本章将首先介绍周期信号傅里叶级数的数值计算方法，再介绍非周期信号傅里叶变换的符号函数求解以及数值计算，最后验证部分傅里叶变换性质。

对连续时间系统进行频域分析的前提是该系统是线性时不变的，只有这样零状态响应 $y_{zs}(t)$ 才等于输入信号 $x(t)$ 卷积单位冲激响应 $h(t)$，从而零状态响应的傅里叶变换 $Y_{zs}(j\omega)$ 才等于输入信号的傅里叶变换 $X(j\omega)$ 乘以单位冲激响应的傅里叶变换 $H(j\omega)$，用 $H(j\omega)$ 对系统进行频域分析才有意义。与 $h(t)$ 一样，$H(j\omega)$ 仅取决于系统本身，与输入无关，是表征系统特性的一个重要物理量。

5.1 周期信号的傅里叶级数

5.1.1 傅里叶级数的计算

工程中主要采用指数形式的傅里叶级数对周期信号进行频域分析。周期为 T 的周期信号 $x(t)$，若满足狄利克雷条件，在任意 $[t_0, t_0+T]$ 区间，可用虚指数信号集 $\left\{e^{jk\omega_0 t}, \omega_0=\frac{2\pi}{T}, k=0,\pm1,\pm2,\cdots\right\}$ 精确分解为以下形式的傅里叶级数

$$x(t)=\sum_{k=-\infty}^{\infty} X_k e^{jk\omega_0 t} \tag{5.1}$$

其中，

$$X_k=\frac{1}{T}\int_{t_0}^{t_0+T} x(t)e^{-jk\omega_0 t}\,dt \tag{5.2}$$

在已知周期信号 $x(t)$ 的数学表达式且能计算出积分的条件下，可以用式(5.2)精确计算傅里叶系数。但当数学表达式非常复杂，无法得出积分结果，或者写不出周期信号的数学表达式时，可以先对周期信号在一个周期内进行抽样，再进行数值计算。

下面讨论利用数值计算得到连续时间周期信号傅里叶级数的方法。设一个周期内刚好抽样得到了 N 个点，则抽样间隔 $T_s=\frac{T}{N}$，抽样得到的序列可以记作 $x(t_0+nT_s)$，式(5.2)可表示为

$$X_k \approx \frac{T_s}{T}\sum_{n=0}^{N-1} x(t_0+nT_s)e^{-jk\omega_0(t_0+nT_s)} \tag{5.3}$$

例 5.1.1 计算如图 5.1.1 所示的周期矩形脉冲信号第 $-10\sim10$ 项的指数形式傅里叶系数。

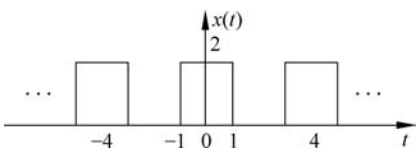


图 5.1.1 连续时间周期矩形脉冲信号

解：可通过 for 循环直接得到 Xk，MATLAB 源代码如下

```
close all; clc;clear all;
T = 4;                                          % 周期
N = 400;                                        % 一个周期抽样点数
dt = T/N;                                       % 抽样间隔
t = -T/2:dt:T/2-dt;                             % 时域自变量
x = 2*rectpuls(t,2);                            % x(t)的赋值
w0 = 2*pi/T;                                    % 基频
k = -10:10;                                     % 需要计算的 Xk 的项数
Xk = zeros(size(k));
for m = 1:length(k)
  for n = 1:N                                   % MATLAB 中数组从第 1 项开始
    Xk(m) = Xk(m) + dt/T*x(n)*exp(-j*k(m)*w0*t(n));      % 按公式计算 Xk
  end
end
stem(k*w0,Xk,'filled'); xlabel('\omega'); title('Xk')
```

因为 Xk 为实数，所以在一幅图里画出了频谱，结果如图 5.1.2 所示。将该结果与 Xk 的理论值 $\mathrm{Sa}\left(\frac{k\pi}{2}\right)$ 进行对比，结果如果图 5.1.3 所示，增加抽样点数，数值解与理论值将会越来越相近。

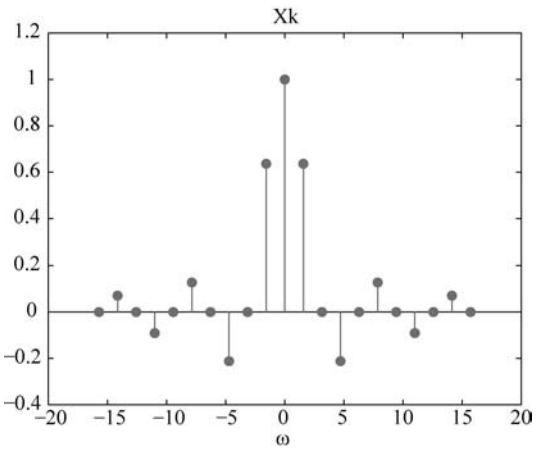


图 5.1.2　连续时间周期矩形脉冲信号的频谱

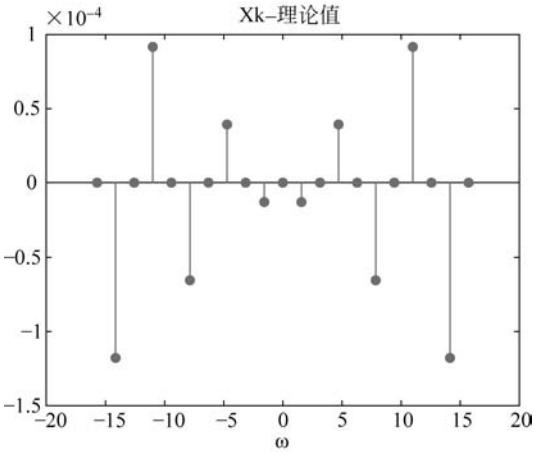


图 5.1.3　Xk 的数值解与理论值之差

因为 MATLAB 是基于矩阵的运算，前面用 for 循环计算 Xk 的方法虽然原理直观，编程简便，但耗时较多。可以将 for 循环改为矩阵-向量相乘，降低计算复杂度。先将式(5.3)中 X_k 表示成向量相乘

$$X_k \approx \frac{T_s}{T}\left[\mathrm{e}^{-\mathrm{j}k\omega_0 t_0} \quad \mathrm{e}^{-\mathrm{j}k\omega_0 (t_0+T_s)} \quad \cdots \quad \mathrm{e}^{-\mathrm{j}k\omega_0 (t_0+(N-1)T_s)}\right]\begin{bmatrix} x(t_0) \\ x(t_0+T_s) \\ \vdots \\ x(t_0+(N-1)T_s) \end{bmatrix} \tag{5.4}$$

进一步地，将 X_k 表示成向量形式，从而

$$\begin{bmatrix} X_{k_1} \\ X_{k_1+1} \\ \vdots \\ X_{k_2} \end{bmatrix} \approx \frac{T_s}{T} \begin{bmatrix} e^{-jk_1\omega_0 t_0} & e^{-jk_1\omega_0(t_0+T_s)} & \cdots & e^{-jk_1\omega_0(t_0+(N-1)T_s)} \\ e^{-j(k_1+1)\omega_0 t_0} & e^{-j(k_1+1)\omega_0(t_0+T_s)} & \cdots & e^{-j(k_1+1)\omega_0(t_0+(N-1)T_s)} \\ \vdots & \vdots & & \vdots \\ e^{-jk_2\omega_0 t_0} & e^{-jk_2\omega_0(t_0+T_s)} & \cdots & e^{-jk_2\omega_0(t_0+(N-1)T_s)} \end{bmatrix} \begin{bmatrix} x(t_0) \\ x(t_0+T_s) \\ \vdots \\ x(t_0+(N-1)T_s) \end{bmatrix} \tag{5.5}$$

矩阵-向量乘法实现例 5.1.1 数值解的 MATLAB 源代码如下

```
close all; clc;clear all;
T = 4;                                  % 周期
N = 400;                                % 一个周期抽样点数
dt = T/N;                               % 抽样间隔
t = -T/2:dt:T/2-dt;                     % 时域自变量
x = 2*rectpuls(t,2);                    % x(t)的赋值
X = [];                                 % 傅里叶级数
w0 = 2*pi/T;                            % 基频
k = -10:10;                             % 需要计算的 Xk 的项数
[W,tt] = meshgrid(k*w0,t);              % 将自变量转为矩阵形式
Xk = dt/T*x*exp(-j*tt.*W);              % 利用矩阵-向量乘法计算 Xk
stem(k*w0,Xk,'filled'); title('Xk')
```

程序运行结果如图 5.1.4 所示。在参数相同的前提下，采用 for 循环结构计算 Xk 的程序运行时间为 0.010007s，而矩阵-向量相乘计算 Xk 的程序运行时间为 0.004057s。所以在 MATLAB 编程时需要尽量用矩阵形式实现算法。

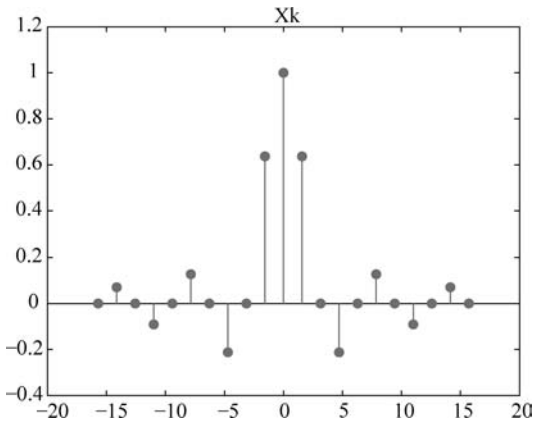


图 5.1.4　矩阵-向量乘法得到的连续时间周期矩形脉冲信号的频谱

观察图 5.1.4 可以发现，对于周期矩形脉冲信号的双边谱来说，幅度谱是偶函数，相位谱是奇函数。但不是所有周期信号的双边谱都有此特点，例如图 5.1.5 所示的虚指数信号 $e^{j\frac{\pi t}{2}}$ 的双边谱。可以证明，只有实信号，才具有幅度谱是偶函数、相位谱是奇函数的特点。

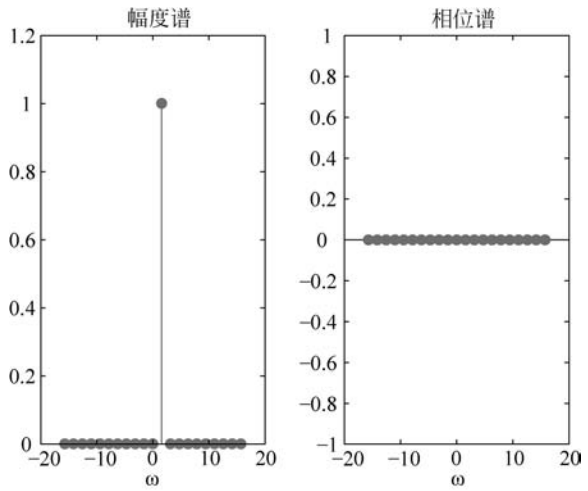


图 5.1.5 $e^{j\frac{\pi t}{2}}$ 的双边谱

MATLAB 提供了快速傅里叶变换(Fast Fourier Transform,FFT)函数 fft 来对离散时间信号进行频谱分析。可以先将连续时间信号抽样,再对得到的离散时间序列用 fft 函数计算傅里叶级数,近似得到连续时间信号的频谱,详见 6.1.2 节。

5.1.2 周期信号的频谱

图 5.1.2 以角频率为横轴,Xk 为纵轴画出了周期信号的双边谱,这是 Xk 为实数的情况。若 Xk 为复数,需要分别以 Xk 的模、初相为纵轴,得到的图形称为信号的幅度谱和相位谱,合称频谱图,借助频谱图可以实现对信号的频谱分析。

例 5.1.2 对如图 5.1.1 所示的周期矩形脉冲信号,分别改变信号的周期 T 和脉冲宽度 τ,讨论其对频谱的影响。

解:可以利用 5.1.1 节的方法,先得到连续时间周期信号在一个周期的抽样序列 $x(n)$,再通过数值计算得到 X_k。但由于周期矩形脉冲信号傅里叶级数 X_k 的理论值为

$$X_k = \frac{2\tau}{T}\mathrm{Sa}\left(\frac{k\omega_0\tau}{2}\right) \tag{5.6}$$

故本题中直接根据理论公式计算 X_k,然后对频谱进行分析。首先保持脉冲宽度不变,增加周期。MATLAB 源代码如下

```
clc; clear all; close all;
dt = 1e-2;                                  % 抽样间隔
t = -20:dt:20;                              % 时域自变量,画出 3 个完整周期
% 保持脉冲宽度不变,改变周期
tao = 2;                                    % 脉冲宽度
T = 4:2:10;                                 % 周期
for i = 1:4
```

```
  x = square((t + tao/2) * (2 * pi)/T(i),tao/T(i) * 100) + 1;      % 周期信号的赋值
  w0 = 2 * pi/T(i);                                                % 基频
  k = -round(10/w0):round(10/w0);                                  % 画 - 10:10 范围内的频谱
  Xk = 2 * tao/T(i) * sinc(k * w0 * tao/2/pi);                     % Xk 的理论值
  subplot(4,2,2 * i - 1); plot(t,x);
  xlabel('t'); title('x(t)'); axis([ - 20,20, - 0.1,2.1])
  subplot(4,2,2 * i);stem(k * w0,Xk,'filled');
  xlabel('\omega'); title('Xk'); axis([ - 10 10 - 0.2 1])
end
```

程序运行结果如图 5.1.6 所示，动态结果请扫描二维码。

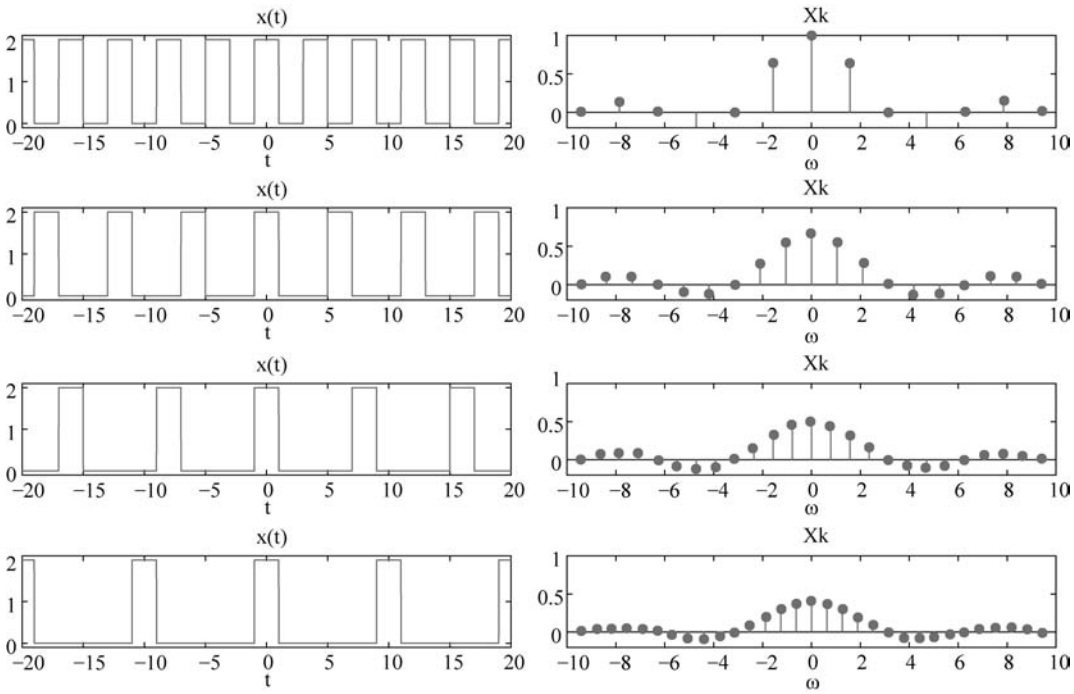


图 5.1.6　周期对频谱的影响

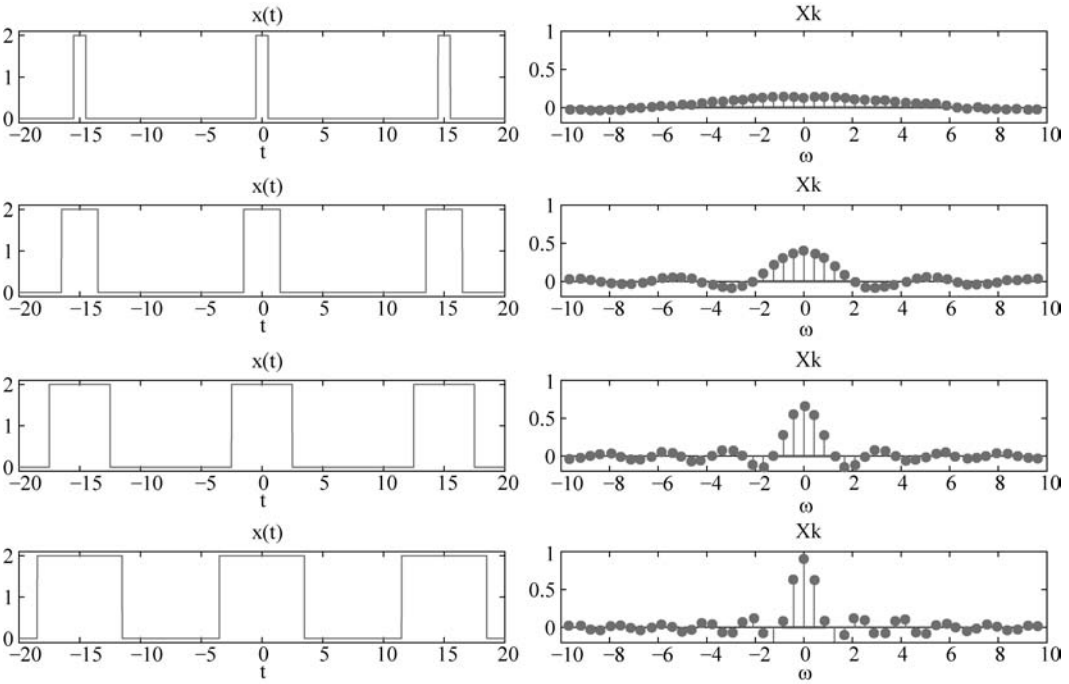
接下来，保持周期不变，增加脉冲宽度。MATLAB 源代码如下

```
close all; clc;clear all;
dt = 1e - 2;                  % 抽样间隔
t = - 20:dt:20;               % 时域自变量，画 3 个完整周期 % 保持周期不变，改变脉冲宽度
T = 15;                       % 周期
for i = 1:4
  tao = 2 * i - 1;            % 脉冲宽度
  x = square((t + tao/2) * (2 * pi)/T,tao/T * 100) + 1;      % 周期信号的赋值
  w0 = 2 * pi/T;                                             % 基频
  k = -round(10/w0):round(10/w0);                            % 画 - 10:10 范围内的频谱
  Xk = 2 * tao/T * sinc(k * w0 * tao/2/pi);                  % Xk 的理论值
  subplot(4,2,2 * i - 1); plot(t,x);
```

```
    xlabel('t'); title('x(t)'); axis([ - 20,20, - 0.1,2.1]);
    subplot(4,2,2 * i); stem(k * w0,Xk,'filled');
    xlabel('\omega'); title('Xk'); axis([ - 10,10, - 0.2,1])
    hold on
    plot([ - 2 * pi/tao,2 * pi/tao],[0,0],'r','linewidth',2)      % 画主瓣宽度
    hold off
end
```

程序运行结果如图 5.1.7 所示，为便于观察频谱的主瓣宽度，图中用红色实线画出了频谱图中 $\left[-\frac{2\pi}{\tau},\frac{2\pi}{\tau}\right]$ 的范围。动态结果请扫描二维码。

彩图



动图

图 5.1.7　脉冲宽度对频谱的影响

观察发现：

(1) 脉冲宽度 τ 保持不变，若 T 增大，则频谱主瓣高度 $\frac{E\tau}{T}$ 减小，各条谱线高度也相应地减小，各谱线间隔 $\omega_0=\frac{2\pi}{T}$ 减小，谱线变密；若 T 减小，则情况相反。但不管 T 如何改变，频谱包络的第一个零点 $\pm\frac{2\pi}{\tau}$ 不变，频谱主瓣宽度不变。

(2) 周期 T 保持不变，若 τ 增大，则频谱主瓣高度 $\frac{E\tau}{T}$ 增大，各条谱线高度也相应地增大，谱包络的第一个零点 $\pm\frac{2\pi}{\tau}$ 减小，主瓣宽度变窄；若 τ 减小，则情况相反。但不管 τ 如

何改变，谱线间隔 $\omega_0=\frac{2\pi}{T}$不变。

综上所述，谱线间隔只与周期有关，且与其成反比；频谱主瓣宽度仅与脉冲宽度有关，且与其成反比。而频谱高度与周期和脉冲宽度都有关，且与周期成反比，与脉冲宽度成正比。可以想象，当脉冲宽度保持不变，周期趋于无穷大时：时域上，周期信号变成了非周期信号；频域上，离散谱变成了连续谱，且高度为零，显然傅里叶级数不能用于非周期信号的频谱分析。

5.1.3 周期信号的分解

周期信号三角形式傅里叶级数如下

$$x(t)=c_0+\sum_{k=1}^{\infty}c_k\cos(k\omega_0 t+\varphi_k) \tag{5.7}$$

其中，$c_k\cos(k\omega_0 t+\varphi_k)$称为周期信号的第 k 次谐波，与式(5.2)相比，

$$c_0=X_0,\quad c_k=2|X_k|,\quad \varphi_k=\angle X_k \tag{5.8}$$

由周期信号得到各次谐波的过程称为信号的分解；反过来，由各次谐波相加得到周期信号的过程称为信号的合成。如果需要直接观察真实存在的各次谐波，可以将周期信号通过滤波器组，详见 11.1 节周期矩形脉冲信号的分解与合成实验。但如果只是想观察各次谐波的特点，可以根据各次谐波理论上的振幅、初相，通过 $c_k\cos(k\omega_0 t+\varphi_k)$仿真产生第 k 次谐波。

例 5.1.3 对如图 5.1.1 所示的周期矩形脉冲信号，画出其 0～8 次谐波。

解：与例 5.1.2 类似，直接利用理论值计算得出 X_k，再根据式(5.8)计算 c_k 和 φ_k，从而得出第 k 次谐波。MATLAB 源代码如下

```
close all; clc;clear all;
tao = 2;                                  %脉冲宽度
T = 4;                                    %周期
dt = 1e-2;                                %抽样间隔
t = -T/2:dt:T/2;                          %时域自变量,取1个完整周期
w0 = 2*pi/T;                              %基频
k = 0:8;                                  %谐波次数
Xk = 2*tao/T*sinc(k*w0*tao/2/pi);         %Xk的理论值
Ck = [Xk(1),2*abs(Xk(2:end))];            %三角形式傅里叶级数振幅
faik = angle(Xk);
for i = 1:9
  yk = Ck(i)*cos(k(i)*w0*t+faik(i));
  subplot(3,3,i); plot(t,yk);
    xlabel('t'); title(['第',num2str(k(i)),'次谐波']);
    axis([-T/2,T/2,-1.5,1.5])
end
```

程序运行结果如图 5.1.8 所示。

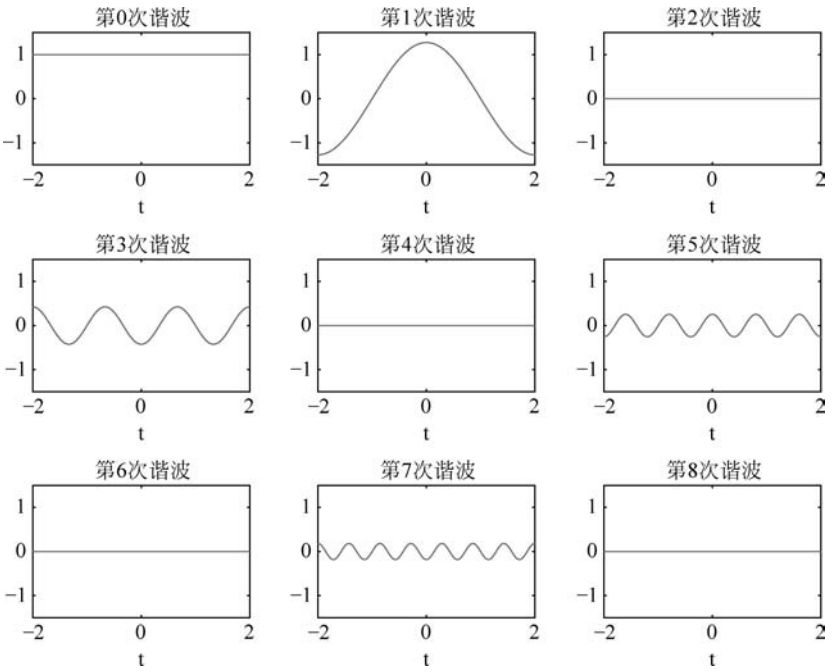


图 5.1.8　周期矩形脉冲信号的各次谐波

观察发现，偶次谐波均为 0，这是因为该周期矩形脉冲信号减去其直流后为方波信号，属于半波像对称信号，半波像对称信号的偶次谐波理论上均为 0；奇次谐波角频率依次增加，振幅依次递减，符合周期信号频谱收敛性的特点。

5.1.4　周期信号的合成

从式(5.7)可以看出，对于信号合成来说，需要无限次谐波相加才能逼近周期信号 $x(t)$，但实际中只能实现有限次谐波相加。为观察不同谐波对信号合成的影响，用 MATLAB 仿真结果做进一步的说明。

例 5.1.4　对例 5.1.3 得到的各次谐波，分析各次谐波对信号合成的影响。

解：依次增加参与合成的谐波次数，MATLAB 源代码如下

```
close all; clc;clear all;
tao = 2;                                    % 脉冲宽度
T = 4;                                      % 周期
dt = 1e-2;                                  % 抽样间隔
t = -T/2:dt:T/2;                            % 时域自变量,取 1 个完整周期
w0 = 2*pi/T;                                % 基频
k = 0:8;                                    % 谐波次数
Xk = 2*tao/T*sinc(k*w0*tao/2/pi);           % Xk 的理论值
Ck = [Xk(1),2*abs(Xk(2:end))];              % 三角形式傅里叶级数振幅
faik = angle(Xk);
```

```
x = 2 * rectpuls(t,tao);
for i = 1:9
  yk(i,:) = Ck(i) * cos(k(i) * w0 * t + faik(i));
end
subplot(2,2,1); plot(t,x);
xlabel('t'); title('原信号'); ylim([ - 0.5,2.5]);
subplot(2,2,2); plot(t,sum(yk(1:4,:),1));
xlabel('t'); title('前 3 次谐波参与合成'); ylim([ - 0.5,2.5]);
subplot(2,2,3); plot(t,sum(yk(1:6,:),1));
xlabel('t'); title('前 5 次谐波参与合成'); ylim([ - 0.5,2.5]);
subplot(2,2,4); plot(t,sum(yk(1:8,:),1));
xlabel('t'); title('前 7 次谐波参与合成'); ylim([ - 0.5,2.5]);
```

程序运行结果如图 5.1.9 所示。观察发现，随着参与合成的谐波次数增加，合成波形越来越逼近周期信号，但间断点处总有一个过冲。可以证明，只要不是所有谐波参与合成，合成波形总会在间断点处有个过冲，且过冲值不变，这种现象即为 Gibbs 现象。

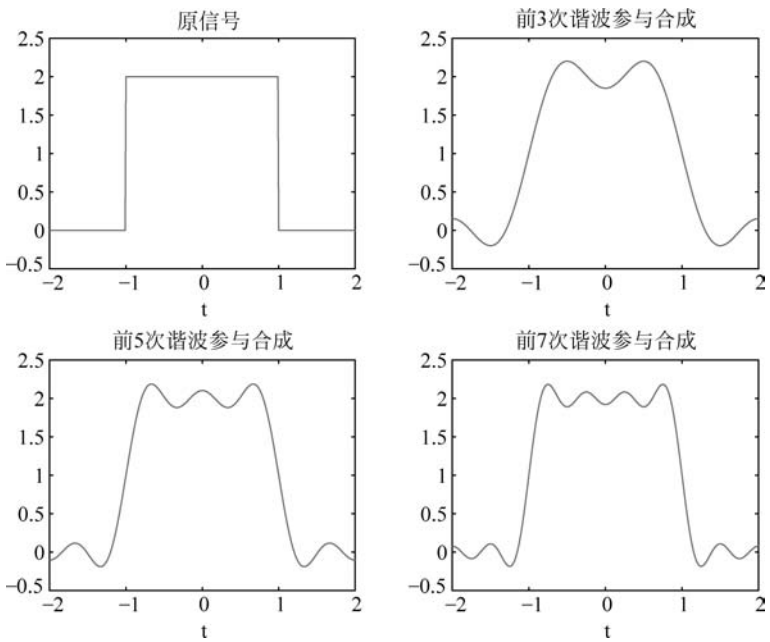


图 5.1.9　周期矩形脉冲信号合成的结果

为分析单次谐波对信号合成的影响，将前 7 次谐波的合成结果依次去除 1、3、5、7 次谐波，结果如图 5.1.10 所示。为便于比较，用点线画出了被分解的周期矩形脉冲信号。比较发现，低次谐波对信号合成的影响较大。这是因为低次谐波频率较小，反映了信号中缓慢变化的物理量；高次谐波频率较大，反映了信号中快速变化的物理量。一般来说，信号中缓慢变化的部分占主导地位。

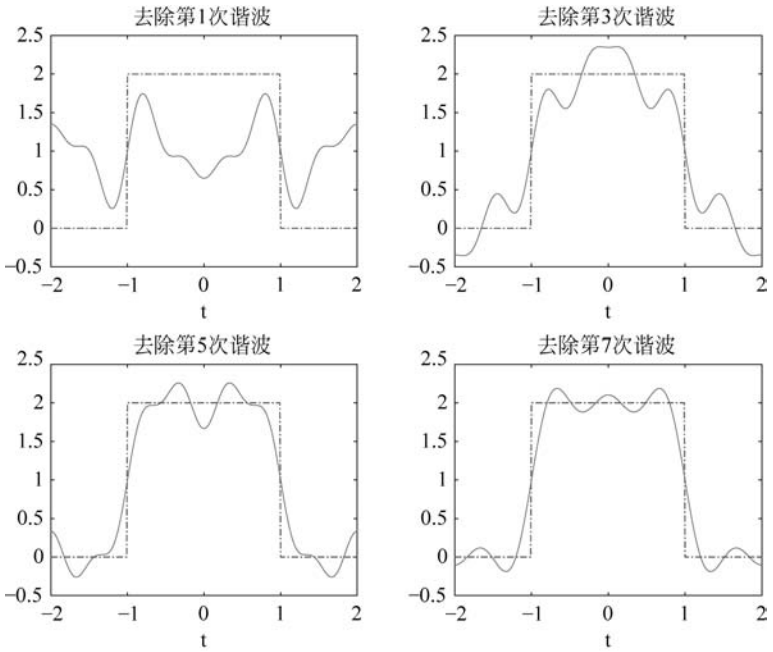


图 5.1.10　单次谐波对信号合成的影响

例 5.1.5　已知某周期三角波信号的周期为 4，其指数形式傅里叶系数为 $Sa^2(0.5k\pi)$，画出其前 N 次谐波合成的信号。

解：不断增加参与合成的谐波次数，MATLAB 源代码如下

```
close all; clc;clear all;
tao = 2;                                    % 脉冲宽度
T = 4;                                      % 周期
dt = 1e-2;                                  % 抽样间隔
t = -T/2:dt:T/2;                            % 时域自变量,取 1 个完整周期
w0 = 2*pi/T;                                % 基频
k = 0:11;                                   % 谐波次数
Xk = sinc(k/2).^2;                          % Xk 的理论值
Ck = [Xk(1),2*abs(Xk(2:end))];              % 三角形式傅里叶级数振幅
faik = angle(Xk);
for i = 1:12
  yk(i,:) = Ck(i)*cos(k(i)*w0*t+faik(i));
end
subplot(2,2,1); plot(t,sum(yk(1:4,:),1));
xlabel('t');title('前 3 次谐波参与合成')
subplot(2,2,2); plot(t,sum(yk(1:6,:),1));
xlabel('t');title('前 5 次谐波参与合成')
subplot(2,2,3); plot(t,sum(yk(1:10,:),1));
xlabel('t');title('前 9 次谐波参与合成')
subplot(2,2,4); plot(t,sum(yk(1:12,:),1));
xlabel('t');title('前 11 次谐波参与合成')
```

程序运行结果如图 5.1.11 所示。

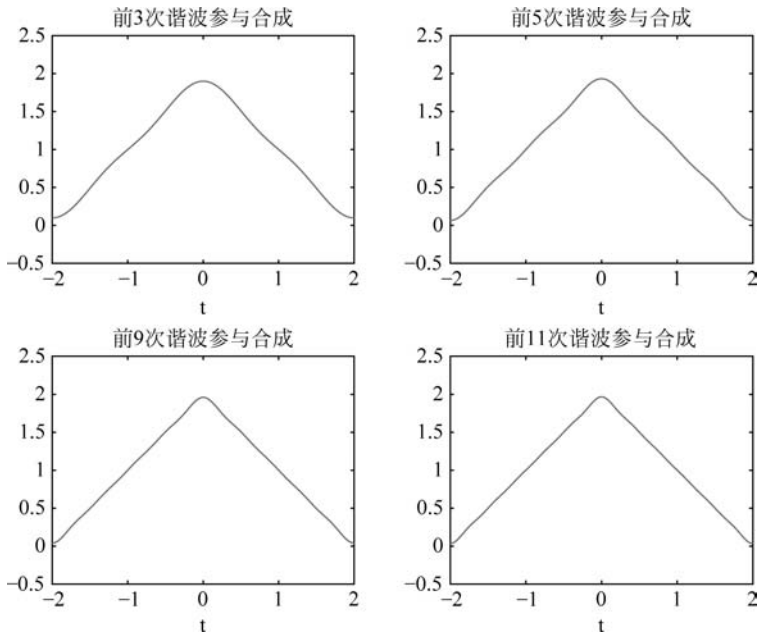


图 5.1.11　周期三角波信号合成的结果

比较图 5.1.9 和图 5.1.11 可以发现，同等条件下显然周期三角波信号合成效果较好。这是因为三角波信号的各次谐波的复振幅是 Sa^2 函数，而周期矩形脉冲信号的是 Sa 函数，而 Sa^2 函数比 Sa 函数衰减快得多。正因为周期三角波信号各次谐波衰减更快，所以高次谐波对信号合成的影响相对更小。

5.2　非周期信号的傅里叶变换

为有效分析非周期信号的频谱，引入了傅里叶变换。傅里叶变换定义为

$$X(\mathrm{j}\omega)=\int_{-\infty}^{\infty}x(t)\mathrm{e}^{-\mathrm{j}\omega t}\,\mathrm{d}t \tag{5.9}$$

傅里叶反变换定义为

$$x(t)=\frac{1}{2\pi}\int_{-\infty}^{\infty}X(\mathrm{j}\omega)\mathrm{e}^{\mathrm{j}\omega t}\,\mathrm{d}\omega \tag{5.10}$$

$x(t)$与 $X(\mathrm{j}\omega)$一一对应，构成傅里叶变换对，记作

$$x(t)\overset{\mathcal{F}}{\longleftrightarrow}X(\mathrm{j}\omega)$$

5.2.1　傅里叶变换的符号函数求解

MATLAB 符号工具箱提供了求解傅里叶变换的函数 fourier 和求解傅里叶反变换

的函数 ifourier，它们的调用格式如下。

fourier(x)：对默认变量为 t 的符号表达式求傅里叶变换，默认返回关于 w 的函数。

fourier(x,v)：对默认自变量为 t 的符号表达式求傅里叶变换，返回关于 v 的函数。

fourier(x,u,v)：对 x(u)求傅里叶变换，返回关于 v 的函数。

ifourier(X)：对默认变量为 w 的符号函数表达式求傅里叶反变换，默认返回关于 t 的函数。

ifourier(X,u)：对默认变量为 w 的符号函数表达式求傅里叶反变换，返回关于 u 的函数。

ifourier(X,v,u)：对 X(v)求傅里叶反变换，返回关于 u 的函数。

需要注意的是，在调用函数 fourier 和 ifourier 之前，要用 syms 函数定义所用到的符号变量或符号表达式，返回结果中默认用 i 表示虚数单位。

可利用 fplot 实现对符号函数的画图，调用格式如下。

fplot(f)：在默认区间[－5 5]，绘制由函数 f 定义的曲线。

fplot(f,xinterval)：在 xinterval 指定的区间绘图，xinterval 是[xmin xmax]形式的二元素向量。

例 5.2.1 求 $e^{-t}u(t)$的傅里叶变换，并画出其频谱图。

解：MATLAB 源代码如下

```
close all; clc;clear all;
syms t w
Xjw = fourier(exp(-t)*heaviside(t),w)    %用符号函数求傅里叶变换
subplot(1,2,1); fplot(abs(Xjw),[-10 10]); title('e^{-t}u(t)的幅度谱')
subplot(1,2,2); fplot(angle(Xjw),[-10 10]); title('e^{-t}u(t)的相位谱')
```

程序运行结果如图 5.2.1 所示。

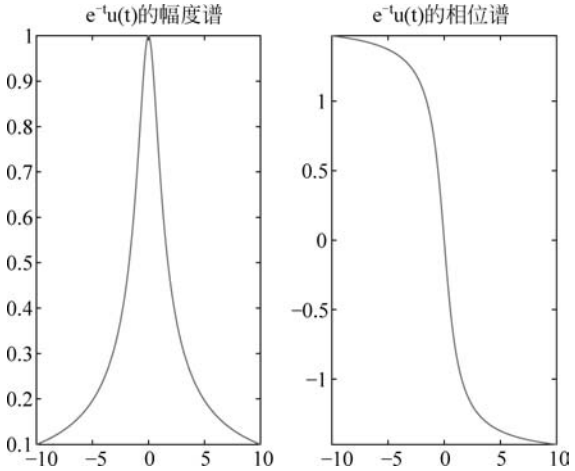


图 5.2.1 $e^{-t}u(t)$的傅里叶变换

命令窗口运行结果为

```
Xjw =
1/(1 + w*1i)
```

例 5.2.2 求$\frac{1}{\omega^2+1}$的傅里叶反变换，并画出波形图。

解：MATLAB 源代码如下

```
close all; clc;clear all;
syms t w
xt = ifourier(1/(w^2 + 1),w,t)                    %用符号函数求傅里叶变换
fplot(xt); ylim([0 0.5]); title('1/(\omega^{2} + 1)的傅里叶反变换')
```

程序运行结果如图 5.2.2 所示。

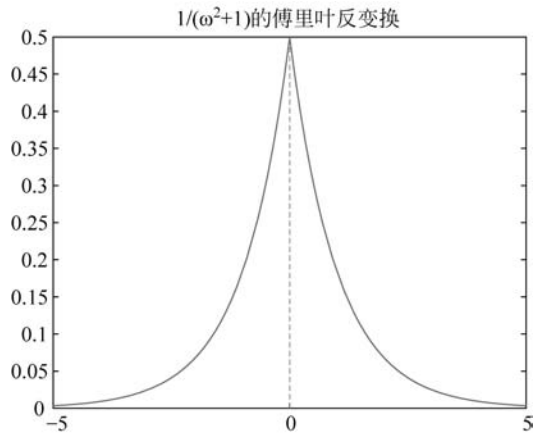


图 5.2.2 $\frac{1}{\omega^2+1}$的傅里叶反变换

命令窗口运行结果为

```
xt =
exp( - abs(t))/2
```

5.2.2 傅里叶变换的数值计算

当信号的数学表达式非常复杂，或者得到的是信号的抽样结果、无法写出其数学表达式时，此时无法用 fourier 函数得出信号的傅里叶变换，但可以借助数值计算的方法进行求解。

下面讨论利用数值求解计算连续时间非周期信号傅里叶变换的方法。假设在非周

期信号的主要取值区间$[t_1,t_2]$内抽样了 N 个点，则抽样间隔 $T_s=\dfrac{t_2-t_1}{N}$，与式(5.3)类似，有

$$X(\mathrm{j}\omega)\approx T_s\sum_{n=0}^{N-1}x(t_1+nT_s)\mathrm{e}^{-\mathrm{j}\omega(t_1+nT_s)} \tag{5.11}$$

用式(5.11)可以计算出任意频点的傅里叶变换值。假设非周期信号频谱的主要取值区间为$[\omega_1,\omega_2]$，在其间均匀抽样了 M 个值，则

$$\Delta\omega=\frac{\omega_2-\omega_1}{M}$$

$$X[\mathrm{j}(\omega_1+m\Delta\omega)]\approx T_s\sum_{n=0}^{N-1}x(t_1+nT_s)\mathrm{e}^{-\mathrm{j}(\omega_1+m\Delta\omega)(t_1+nT_s)} \tag{5.12}$$

可以采用同样的方法计算傅里叶反变换

$$x(t_1+nT_s)\approx\frac{\Delta\omega}{2\pi}\sum_{m=0}^{M-1}X[\mathrm{j}(\omega_1+m\Delta\omega)]\mathrm{e}^{\mathrm{j}(\omega_1+m\Delta\omega)(t_1+nT_s)} \tag{5.13}$$

式(5.12)、式(5.13)可用矩阵-向量乘法编程实现，以式(5.12)为例

$$\begin{bmatrix}X(\mathrm{j}\omega_1)\\X[\mathrm{j}(\omega_1+\Delta\omega)]\\\vdots\\X[\mathrm{j}(\omega_2-\Delta\omega)]\end{bmatrix}\approx T_s\begin{bmatrix}\mathrm{e}^{-\mathrm{j}\omega_1t_1}&\mathrm{e}^{-\mathrm{j}\omega_1(t_1+T_s)}&\cdots&\mathrm{e}^{-\mathrm{j}\omega_1(t_2-T_s)}\\\mathrm{e}^{-\mathrm{j}(\omega_1+\Delta\omega)t_1}&\mathrm{e}^{-\mathrm{j}(\omega_1+\Delta\omega)(t_1+T_s)}&\cdots&\mathrm{e}^{-\mathrm{j}(\omega_1+\Delta\omega)(t_2-T_s)}\\\vdots&\vdots&&\vdots\\\mathrm{e}^{-\mathrm{j}(\omega_2-\Delta\omega)t_1}&\mathrm{e}^{-\mathrm{j}(\omega_2-\Delta\omega)(t_1+T_s)}&\cdots&\mathrm{e}^{-\mathrm{j}(\omega_2-\Delta\omega)(t_2-T_s)}\end{bmatrix}\begin{bmatrix}x(t_1)\\x(t_1+T_s)\\\vdots\\x(t_2-T_s)\end{bmatrix} \tag{5.14}$$

例 5.2.3 用数值计算的方法求 $\mathrm{e}^{-t}u(t)$的傅里叶变换。

解：采用矩阵-向量相乘实现傅里叶变换的数值解，MATLAB 源代码如下

```
close all; clc;clear all;
dt = 0.01;                                  %时域抽样间隔
t=0:dt:20;                                  %信号主要取值区间
x=exp(-t);                                  %信号赋值
w = -20:0.01:20;                            %信号频谱主要取值区间
[W,T] = meshgrid(w,t);                      %生成矩阵
X = dt*x*exp(-j*T.*W);                      %利用矩阵-向量乘法计算
subplot(1,2,1); plot(w,abs(X)); title('e^{-t}u(t)的幅度谱')
subplot(1,2,2); plot(w,angle(X)); title('e^{-t}u(t)的相位谱')
```

程序运行结果如图 5.2.3 所示。

将矩阵-向量乘法得到的数值解与理论值$\dfrac{1}{\mathrm{j}\omega+1}$进行比较，结果如图 5.2.4 所示。可以看出计算误差略大，增加 t 的范围，结果几乎没有改善，这是因为前面已经取到了信号的主要取值区间；将时域抽样间隔由 10^{-2} 降低到 10^{-4}，误差将降到 10^{-5} 量级，但运算量会显著增加。

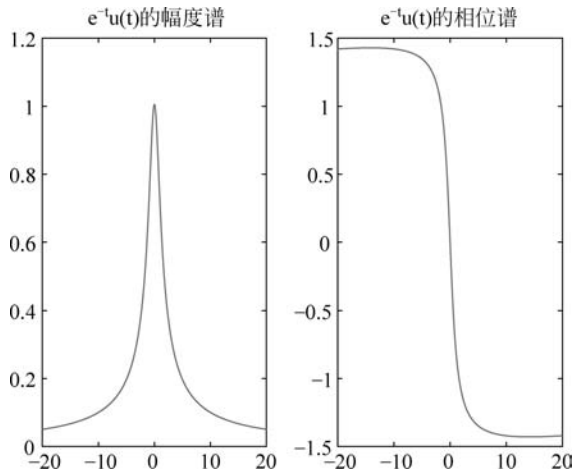


图 5.2.3 矩阵-向量乘法得到的傅里叶变换

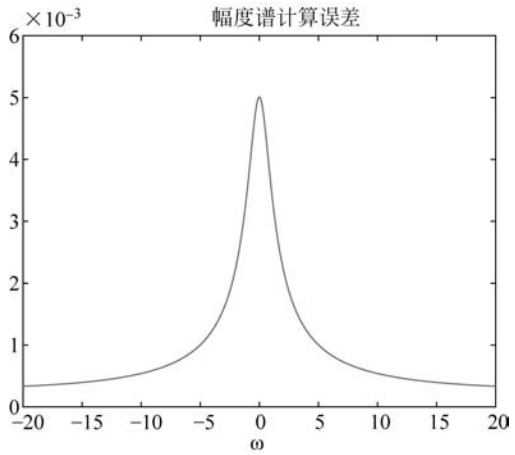


图 5.2.4 矩阵-向量乘法的误差

在抽样间隔相同的条件下，5.1.1 节中矩阵-向量乘法计算傅里叶系数的误差却是 10^{-4} 量级，一方面是因为积分区间只需取一个完整的周期，不需要取$(-\infty,\infty)$；另一方面是因为 X_k 是离散的，只需计算有限个频率点上的复振幅。

MATLAB 提供了 quad8 和 quadl 函数来计算数值积分，quad8 函数的返回值是用自适应 Simpson 算法得出的积分值。quadl 函数是从 MATLAB 6.0 版本才开始出现的一个积分函数，它的返回值是用 Lobatto 算法得到的积分值，具有更高的积分精度。quadl 的调用格式为

```
y = quadl(fun,a,b)
```

其中，a，b 分别是积分下限和积分上限，fun 是被积函数。

用 quadl 函数重新求解例 5.2.3，MATLAB 源代码如下

```
close all; clc;clear all;
w = -20:0.01:20;                              % 信号频谱主要取值区间
for i = 1:length(w)
  F = @(t)exp(-t).*exp(-j*w(i)*t);            % 被积函数
  X(i) = quadl(F,0,20);                       % Lobatto 法
end
figure;  plot(abs(X)-abs(1./(j*w+1)));  title('幅度谱计算误差')
```

程序运行结果如图 5.2.5 所示。比较发现，Lobatto 算法误差比矩阵-向量乘法的误差小得多。但这种方法需要知道信号的函数表达式，如果只能得到信号的抽样结果，仍旧需要用矩阵-向量乘法计算傅里叶变换。

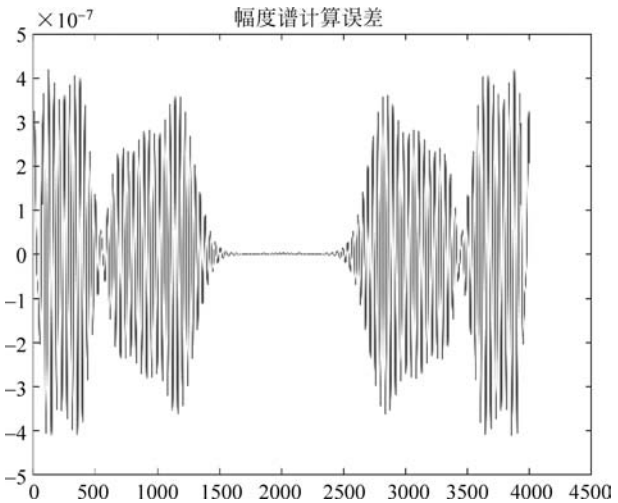


图 5.2.5　Lobatto 算法计算数值积分的误差

5.3　傅里叶变换性质

傅里叶变换性质是信号与系统课程的重要知识点，利用傅里叶变换的性质不仅可以简化计算，而且可以更好地理解时域特性与频域特性间的关系以及时域变化与频域变化间的关系。

5.3.1　奇偶特性

理论上，实偶信号的频谱为实偶函数，实奇信号的频谱为虚奇函数。但这并不意味着实偶信号的幅度谱、相位谱均是偶函数，实奇信号的幅度谱、相位谱均是奇函数。频谱是实偶函数指的是频谱是实的，且

$$X(\mathrm{j}\omega)=X(-\mathrm{j}\omega) \tag{5.15}$$

将 $X(\mathrm{j}\omega)$写成直角坐标形式，可以推出实偶信号频谱的实部、虚部均是偶函数，

$$\mathrm{Re}[X(\mathrm{j}\omega)]=\mathrm{Re}[X(-\mathrm{j}\omega)] \tag{5.16}$$

$$\mathrm{Im}[X(\mathrm{j}\omega)]=\mathrm{Im}[X(-\mathrm{j}\omega)]$$

将 $X(\mathrm{j}\omega)$写成极坐标形式，可以推出幅度谱是偶函数，且 $\mathrm{e}^{\mathrm{j}\angle X(\mathrm{j}\omega)}=\mathrm{e}^{\mathrm{j}\angle X(-\mathrm{j}\omega)}$，但并不能由此得出相位谱是偶函数的结论。

同理，实奇信号的频谱是虚奇函数，指的是频谱是虚的，且频谱的实部、虚部都是奇函数。下面通过例 5.3.1 验证以上分析。

例 5.3.1 分别画出 $G_4(t)$、$\Lambda_4(t)$、$\mathrm{e}^{-t}u(t)$、$\mathrm{e}^{-t}u(t)-\mathrm{e}^{t}u(-t)$的频谱。

解：分别画出 4 个信号的时域波形、幅度谱和相位谱，MATLAB 源代码如下

```
close all; clc;clear all;
dt = 1e-3;                                          %时域抽样间隔
t = -10:dt:10;                                      %信号主要取值区间
w = -20:0.01:20;                                    %信号频谱主要取值区间
[W,T] = meshgrid(w,t);                              %生成矩阵
%门信号
tao=4;
xt1 = rectpuls(t,tao);
Xjw1 = dt*xt1*exp(-j*T.*W);                         %利用矩阵-向量乘法计算
%Sa(t)
xt2 = tripuls(t,tao);
Xjw2 = dt*xt2*exp(-j*T.*W);                         %利用矩阵-向量乘法计算
%e^(-t)u(t)
xt3 = exp(-t).*(t>=0);
Xjw3 = dt*xt3*exp(-j*T.*W);                         %利用矩阵-向量乘法计算
%sgn(t)
xt4 = exp(-t).*(t>=0)-exp(t).*(t<=0);
Xjw4 = dt*xt4*exp(-j*T.*W);                         %利用矩阵-向量乘法计算
%画图
subplot(4,3,1); plot(t,xt1);
title('G_{4}(t)'); xlabel('t'); ylim([-0.1 1.1])
subplot(4,3,2); plot(w,abs(Xjw1));
title('G_{4}(t)的幅度谱'); xlabel('\omega')
subplot(4,3,3); plot(w,angle(Xjw1).*(abs(Xjw1)>=1e-3));   %去除数值计算带来的误差
title('G_{4}(t)的相位谱'); xlabel('\omega'); ylim([-3.2 3.2])
subplot(4,3,4); plot(t,xt2);
title('\Lambda_{4}(t)'); xlabel('t')
subplot(4,3,5); plot(w,abs(Xjw2));
title('\Lambda_{4}(t)的幅度谱'); xlabel('\omega')
subplot(4,3,6); plot(w,angle(Xjw2).*(abs(Xjw2)>=1e-3));   %去除数值计算带来的误差
title('\Lambda_{4}(t)的相位谱'); xlabel('\omega'); ylim([-3.2 3.2])
subplot(4,3,7); plot(t,xt3);
title('e^{-t}u(t)'); xlabel('t'); ylim([-0.1 1.1])
subplot(4,3,8); plot(w,abs(Xjw3));
title('e^{-t}u(t)的幅度谱'); xlabel('\omega')
subplot(4,3,9); plot(w,angle(Xjw3).*(abs(Xjw3)>=1e-3));   %去除数值计算带来的误差
title('e^{-t}u(t)的相位谱'); xlabel('\omega'); ylim([-3.2 3.2])
subplot(4,3,10); plot(t,xt4);
title('e^{-t}u(t)-e^{t}u(-t)'); xlabel('t'); ylim([-1.1 1.1])
subplot(4,3,11); plot(w,abs(Xjw4));
title('e^{-t}u(t)-e^{t}u(-t)的幅度谱'); xlabel('\omega')
```

```
subplot(4,3,12); plot(w,angle(Xjw4). * (abs(Xjw4)> = 1e - 3)); % 去除数值计算带来的误差
title('e^{ - t}u(t) - e^{t}u( - t)的相位谱'); xlabel('\omega'); ylim([ - 3.2 3.2])
```

程序运行结果如图 5.3.1 所示。在画相位谱时，考虑到幅度较小时容易带来相位计算误差，因此认为当幅度谱小于 10^{-3} 时，相位为 0。从图 5.3.1 中可以看出，虽然 $G_4(t)$、$\Lambda_4(t)$均为实偶信号，但相位谱并不均是偶函数，与前文的分析一致；同样的，虽然 $e^{-t}u(t)-e^{t}u(-t)$是实奇信号，但其幅度谱也不是奇函数。4 个信号都是实信号，满足实信号幅度谱均是偶函数，相位谱均是奇函数的特点。

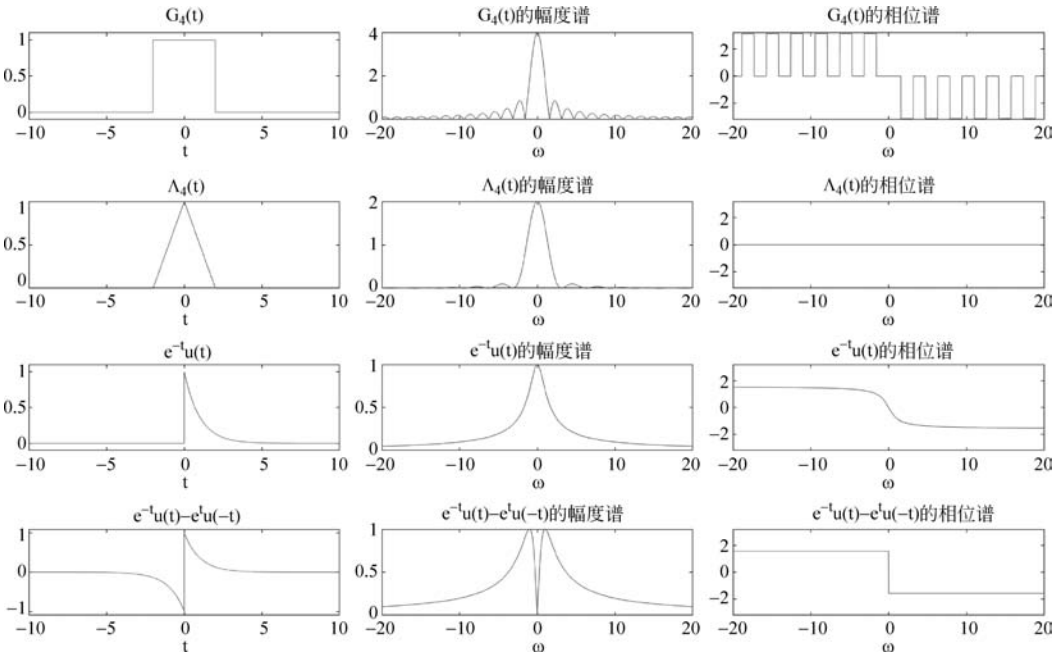


图 5.3.1　4 个信号的时频图

分别画出 4 个信号频谱的实部和虚部，结果如图 5.3.2 所示。与前面的分析一致，偶信号频谱的实部是偶函数，虚部为零；奇信号频谱的虚部是奇函数，实部为零；既非奇又非偶的信号频谱实部、虚部均不全为零，但也满足实信号频谱的实部是偶函数、虚部是奇函数的特点。

5.3.2　展缩特性

若 $x(t) \xleftrightarrow{\mathcal{F}} X(\mathrm{j}\omega)$，由展缩性质可知

$$x(at) \xleftrightarrow{\mathcal{F}} \frac{1}{|a|} X\left(\mathrm{j}\frac{\omega}{a}\right) \tag{5.17}$$

这意味着信号在时域上压缩，频域上将扩展；反过来在时域上扩展，频域上将压缩。下面通过例 5.3.2 说明这一点。

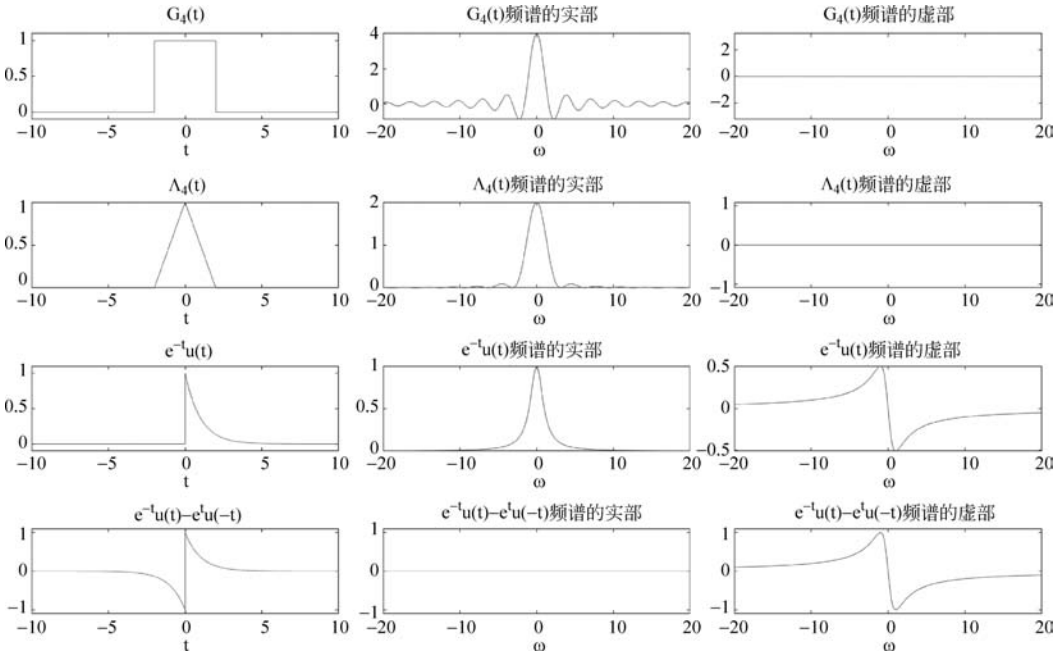


图 5.3.2　4 个信号频谱的实部、虚部

例 5.3.2　设 $x(t)=G_4(t)$，分别画出 $x(t)$、$x(t/2)$、$x(2t)$ 的频谱。

解：MATLAB 源代码如下

```
close all; clc;clear all;
dt = 1e-3;                                          %时域抽样间隔
t = -10:dt:10;                                      %信号主要取值区间
w = -20:0.01:20;                                    %信号频谱主要取值区间
[W,T] = meshgrid(w,t);                              %生成矩阵
%x(t)
tao = 4;
xt1 = rectpuls(t,tao);
Xjw1 = dt*xt1*exp(-j*T.*W);                         %利用矩阵-向量乘法计算
%x(t/2)
xt2 = rectpuls(t/2,tao);
Xjw2 = dt*xt2*exp(-j*T.*W);                         %利用矩阵-向量乘法计算
%x(2t)
xt3 = rectpuls(2*t,tao);
Xjw3 = dt*xt3*exp(-j*T.*W);                         %利用矩阵-向量乘法计算
%画图
subplot(3,3,1); plot(t,xt1);
title('x(t)'); xlabel('t'); ylim([-0.1 1.1])
subplot(3,3,2); plot(w,abs(Xjw1));
title('x(t)的幅度谱'); xlabel('\omega')
subplot(3,3,3); plot(w,angle(Xjw1).*(abs(Xjw1)>=1e-3));  %去除数值计算带来的误差
title('x(t)的相位谱'); xlabel('\omega');ylim([-3.2 3.2])
```

```
subplot(3,3,4); plot(t,xt2);
title('x(t/2)'); xlabel('t'); ylim([ - 0.1 1.1])
subplot(3,3,5); plot(w,abs(Xjw2));
title('x(t/2)的幅度谱'); xlabel('\omega')
subplot(3,3,6); plot(w,angle(Xjw2). * (abs(Xjw2)> = 1e - 3)); % 去除数值计算带来的误差
title('x(t/2)的相位谱'); xlabel('\omega'); ylim([ - 3.2 3.2])
subplot(3,3,7); plot(t,xt3);
title('x(2t)'); xlabel('t'); ylim([ - 0.1 1.1])
subplot(3,3,8); plot(w,abs(Xjw3));
title('x(2t)的幅度谱'); xlabel('\omega')
subplot(3,3,9); plot(w,angle(Xjw3). * (abs(Xjw3)> = 1e - 3)); % 去除数值计算带来的误差
title('x(2t)的相位谱'); xlabel('\omega'); ylim([ - 3.2 3.2])
```

程序运行结果如图 5.3.3 所示。可以看出，从 x(t)到 x(t/2)，时域上的宽度扩展为原来的 2 倍，频带宽度却压缩为原来的一半，同时幅度谱最大值变成原来的 2 倍，从理论上说，式(5.17)中有一个$\frac{1}{|a|}$的系数，也可以直观理解为信号在时域上变宽，本身的能量变大，而幅度谱却变窄，结果只能是幅度谱的值变大。另一方面，从 x(t)到 x(2t)，时域上的宽度压缩为原来的一半，频带宽度却扩展为原来的 2 倍，同时幅度谱最大值变成原来的一半。这些变化直观反映了傅里叶变换的展缩特性。

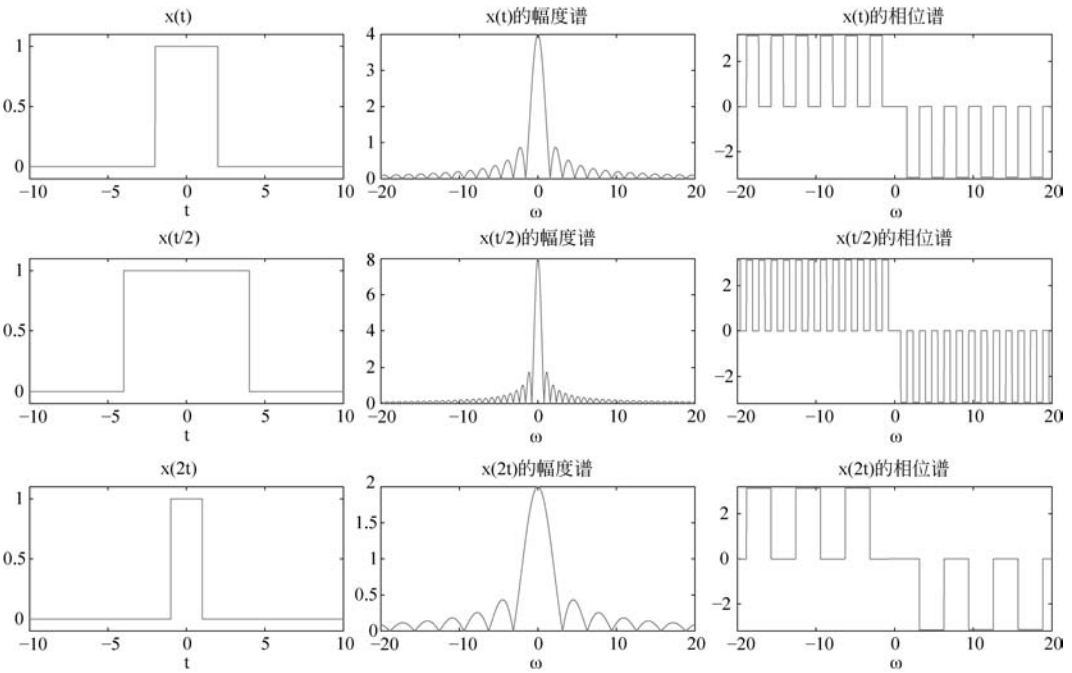


图 5.3.3　信号展缩前后的频谱

音频

可以通过语音信号的展缩更加直观地感受时域展缩对语速、音调的影响，请扫描二维码。

5.3.3 时移特性

若 $x(t) \overset{\mathcal{F}}{\longleftrightarrow} X(\mathrm{j}\omega)$，由时移性质可知

$$x(t+t_0) \overset{\mathcal{F}}{\longleftrightarrow} X(\mathrm{j}\omega)\mathrm{e}^{\mathrm{j}\omega t_0} \tag{5.18}$$

这意味着信号时移后，频谱将乘以因子 $\mathrm{e}^{\mathrm{j}\omega t_0}$。由于两复数相乘对应模相乘、相位相加，再加上 $\mathrm{e}^{\mathrm{j}\omega t_0}$ 的模为 1，相位为 ωt_0，因此，信号的时移不会影响幅度谱，只是相位谱产生附加变化 ωt_0，改变量是过原点且斜率是 t_0 的直线。下面通过例 5.3.3 说明这一点。

例 5.3.3 分别画出 $x(t)=\Lambda_4(t)$、$x(t-0.1)$、$x(t-1)$的频谱。

解：为方便比较，选用相位谱为 0 的 $\Lambda_4(t)$作为 $x(t)$。MATLAB 源代码如下

```
close all; clc;clear all;
dt = 1e-2;                                          %时域抽样间隔
t = -10:dt:10;                                      %信号主要取值区间
w = -10:0.01:10;                                    %信号频谱主要取值区间
[W,T] = meshgrid(w,t);                              %生成矩阵
%x(t)
tao = 4;
xt1 = tripuls(t,tao);
Xjw1 = dt*xt1*exp(-j*T.*W);                         %利用矩阵-向量乘法计算
%x(t-0.1)
xt2 = tripuls(t-0.1,tao);
Xjw2 = dt*xt2*exp(-j*T.*W);                         %利用矩阵-向量乘法计算
%x(t-1)
xt3 = tripuls(t-1,tao);
Xjw3 = dt*xt3*exp(-j*T.*W);                         %利用矩阵-向量乘法计算
%画图
subplot(3,3,1); plot(t,xt1);
title('x(t)'); xlabel('t'); ylim([-0.1 1.1])
subplot(3,3,2); plot(w,abs(Xjw1));
title('x(t)的幅度谱'); xlabel('\omega')
subplot(3,3,3); plot(w,angle(Xjw1));
title('x(t)的相位谱'); xlabel('\omega');ylim([-3.2 3.2])
subplot(3,3,4); plot(t,xt2);
title('x(t-0.1)'); xlabel('t'); ylim([-0.1 1.1])
subplot(3,3,5); plot(w,abs(Xjw2));
title('x(t-0.1)的幅度谱'); xlabel('\omega')
subplot(3,3,6); plot(w,angle(Xjw2));
title('x(t-0.1)的相位谱'); xlabel('\omega'); ylim([-3.2 3.2])
subplot(3,3,7); plot(t,xt3);
title('x(t-1)'); xlabel('t'); ylim([-0.1 1.1])
subplot(3,3,8); plot(w,abs(Xjw3));
title('x(t-1)的幅度谱'); xlabel('\omega')
```

```
subplot(3,3,9); plot(w,angle(Xjw3));
title('x(t-1)的相位谱'); xlabel('\omega'); ylim([-3.2 3.2])
```

程序运行结果如图 5.3.4 所示。可以看出,信号时移前后幅度谱没有发生改变,只是相位谱发生了变化,相位改变量是过原点的直线,该直线的斜率跟时移量有关,结合式(5.18),直线的斜率刚好就是时移量。x(t-1)的相位谱之所以是分段直线而非前面分析的直线,是因为要保证初相在[$-\pi,\pi$]区间内。

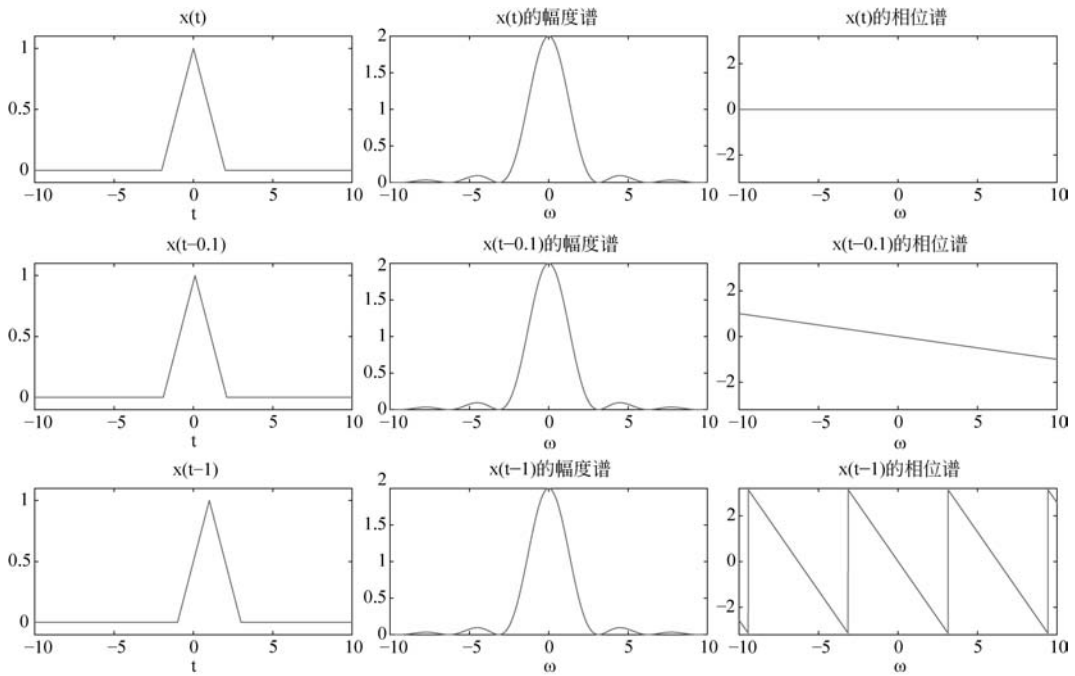


图 5.3.4　信号时移前后的频谱

5.3.4　频移特性

若 $x(t) \overset{\mathcal{F}}{\longleftrightarrow} X(\mathrm{j}\omega)$,由频移性质可知

$$x(t)\mathrm{e}^{\mathrm{j}\omega_0 t} \overset{\mathcal{F}}{\longleftrightarrow} X[\mathrm{j}(\omega-\omega_0)] \tag{5.19}$$

这意味着信号乘以虚指数信号 $\mathrm{e}^{\mathrm{j}\omega_0 t}$ 后,频谱将移位 ω_0。若信号乘以余弦信号 $\cos\omega_0 t$,通过将 $\cos\omega_0 t$ 进行欧拉公式展开,可得

$$x(t)\cos\omega_0 t \overset{\mathcal{F}}{\longleftrightarrow} \frac{X[\mathrm{j}(\omega-\omega_0)]+X[\mathrm{j}(\omega+\omega_0)]}{2} \tag{5.20}$$

这意味着信号乘以 $\cos\omega_0 t$ 后,频谱将保持形状不变,往左往右各频移 ω_0,同时幅度将变成原来的一半。下面通过例 5.3.4 说明这一点。

例 5.3.4　分别画出 $G_4(t)$、$G_4(t)\cos(50t)$的频谱。

解：MATLAB 源代码如下

```
close all; clc;clear all;
dt = 1e-2;                                          %时域抽样间隔
t = -10:dt:10;                                      %信号主要取值区间
w = -60:0.01:60;                                    %信号频谱主要取值区间
[W,T] = meshgrid(w,t);                              %生成矩阵
%x(t)
tao = 4;
xt1 = rectpuls(t,tao);
Xjw1 = dt*xt1*exp(-j*T.*W);                         %利用矩阵-向量乘法计算
%x(t)cos(50t)
xt2 = rectpuls(t,tao).*cos(50*t);
Xjw2 = dt*xt2*exp(-j*T.*W);                         %利用矩阵-向量乘法计算
%画图
subplot(2,2,1); plot(t,xt1);
title('G_{4}(t)'); xlabel('t'); ylim([-0.1 1.1])
subplot(2,2,2); plot(w,Xjw1);
title('G_{4}(t)的频谱'); xlabel('\omega')
subplot(2,2,3); plot(t,xt2);
title('G_{4}(t)cos(50t)'); xlabel('t')
subplot(2,2,4); plot(w,Xjw2);
title('G_{4}(t)cos(50t)的频谱'); xlabel('\omega')
```

程序运行结果如图 5.3.5 所示。可以看出，乘以 cos(50t)后，$G_4(t)$频谱的中心点由 0 移到了±50，同时最大值由 4 降为 2，与前面的理论分析一致。

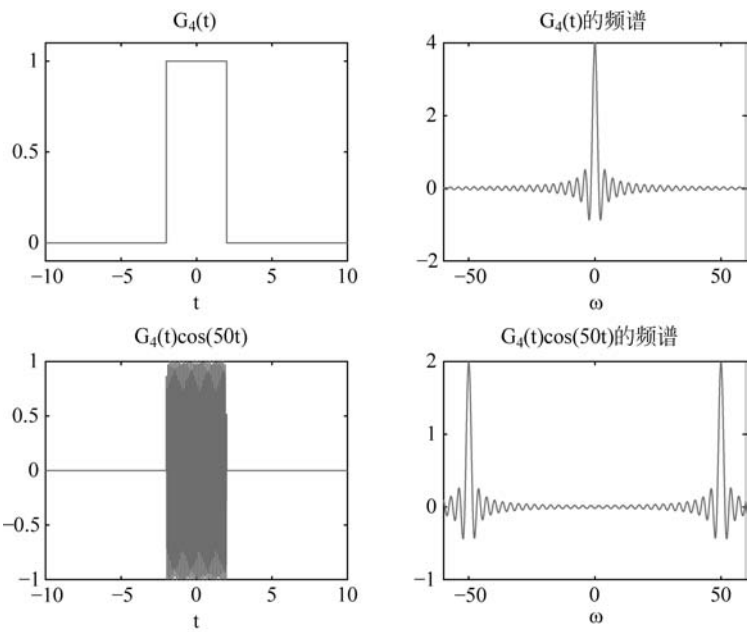


图 5.3.5　频移前后信号的频谱

5.4 连续时间系统的频率响应

对式(4.17)描述的连续时间线性时不变系统等式两边分别做傅里叶变换并进行整理,可得

$$H(\mathrm{j}\omega)=\frac{b_m(\mathrm{j}\omega)^m+b_{m-1}(\mathrm{j}\omega)^{m-1}+\cdots+b_0}{(\mathrm{j}\omega)^n+a_{n-1}(\mathrm{j}\omega)^{n-1}+\cdots+a_0} \tag{5.21}$$

式中,m 和 n 都是正整数,系数均为实数。

在 MATLAB 中,信号处理工具箱中的 freqs 函数可直接计算连续时间系统的频率响应,调用格式如下。

freqs(b,a):没有返回值,直接画出系统幅频、相频响应的波特图。b、a 分别是式(5.21)中的分子、分母多项式的系数向量。

H = freqs(b,a,w):向量 w 是系统频率响应的角频率范围,返回值 H 为 w 上的频率响应。

[H,w] = freqs(b,a,n):自动设定 n 个角频率点来计算频率响应,返回值 w 为设定的 n 个角频率值,n 的默认值为 200。

例 5.4.1 某系统微分方程为 $y''(t)+y'(t)+y(t)=x(t)$,求该系统的频率响应并画出幅频、相频响应曲线。

解:MATLAB 源代码如下

```
close all; clc;clear all;
a = [1 1 1];                                  % 分母多项式系数
b = [1];                                      % 分子多项式系数
[H,w] = freqs(b,a);                           % 求连续时间系统的频响
subplot(1,2,1); plot(w,abs(H));
xlabel('\omega'); title('幅频响应')
subplot(1,2,2); plot(w,angle(H));
xlabel('\omega'); title('相频响应')
```

程序运行结果如图 5.4.1 所示。

调用 freqs 函数时采用了自动设定自变量的方式,结果只画出了频谱的右半边。可以通过设置自变量,得到双边谱。将上述程序改为以下形式,得到的双边谱如图 5.4.2 所示。

```
close all; clc;clear all;
a = [1 1 1];                                  % 分母多项式系数
b = [1];                                      % 分子多项式系数
w = -10:0.01:10;                              % 角频率
H = freqs(b,a,w);                             % 求连续时间系统的频响
subplot(1,2,1); plot(w,abs(H));
xlabel('\omega'); title('幅频响应')
subplot(1,2,2); plot(w,angle(H));
xlabel('\omega'); title('相频响应')
```

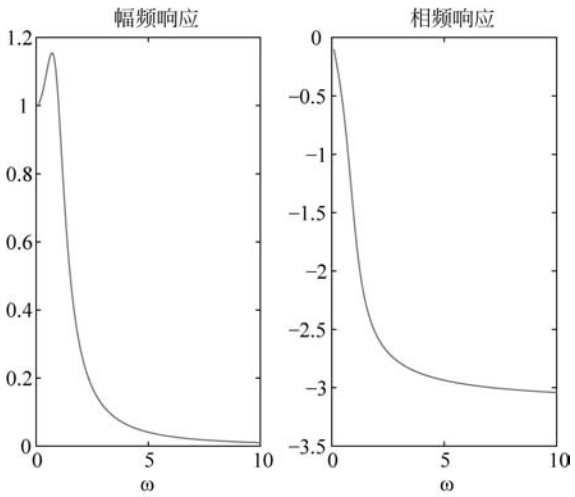


图 5.4.1　连续时间系统的频率响应

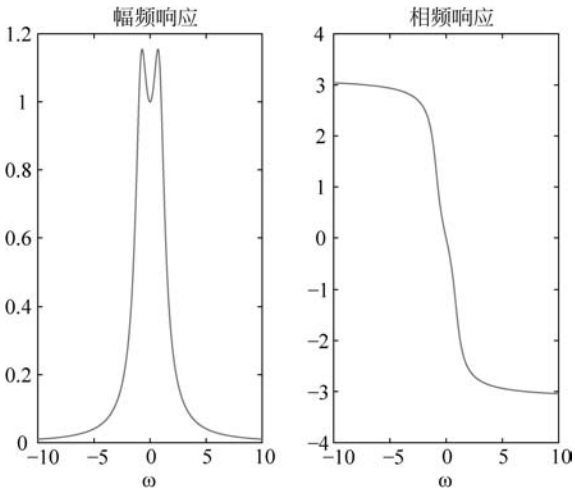


图 5.4.2　程序改进后连续时间系统的频率响应

5.5　正弦信号通过系统的响应

正余弦信号通过系统的响应既可以通过 2.1.2 节零状态响应的时域求解来实现，也可以快速地通过式(5.22)、式(5.23)以及叠加原理得到理论结果。

$$\cos(\omega_0 t) \rightarrow |H(j\omega_0)| \cos(\omega_0 t + \angle H(j\omega_0)) \tag{5.22}$$

$$\sin(\omega_0 t) \rightarrow |H(j\omega_0)| \sin(\omega_0 t + \angle H(j\omega_0)) \tag{5.23}$$

例 5.5.1　计算 $x(t)=\cos t+\cos 10t$ 通过例 5.4.1 系统的响应。

解：MATLAB 源代码如下

```
clc; clear all; close all;
a = [1 1 1];  b = [1];                          % 系统分母、分子多项式系数
H = freqs(b,a,[1 10]);                          % 仅计算在 1、10 两个角频率上的频率响应
t = 0:0.001:40;
x = cos(t) + cos(10 * t);
y = abs(H(1)) * cos(t + angle(H(1))) + abs(H(2)) * cos(10 * t + angle(H(2)));
subplot(1,2,1); plot(t,x);
xlabel('t'); title('x(t)')
subplot(1,2,2); plot(t,y);
xlabel('t'); title('y(t)')
```

程序运行结果如图 5.5.1 所示。观察发现，信号通过系统后高频分量几乎消失，这是因为系统的幅频响应在 $\omega=10$ 处几乎为 0，而在 $\omega=1$ 处的值较大(也可以观察图 5.4.2)。

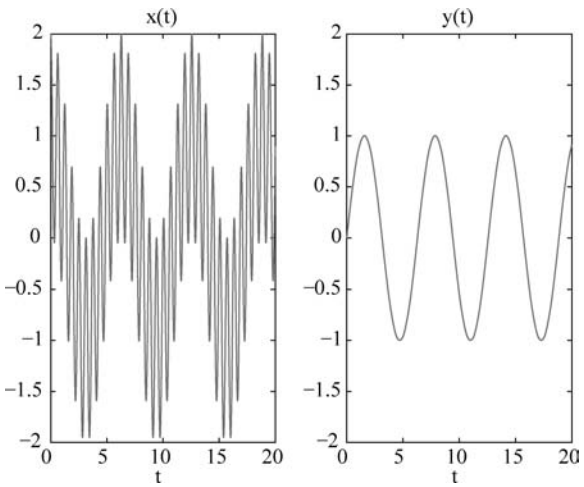


图 5.5.1　余弦信号通过系统的响应

5.6　无失真传输

5.6.1　无失真传输系统

若某连续时间系统对所有的输入信号都可以实现

$$y_{zs}(t)=kx(t-t_d) \tag{5.24}$$

即输出信号的幅度是输入信号幅度的 k 倍(k 为实常数，且 $k\neq 0$)，输出信号比输入信号延迟了 $t_d(t_d\geqslant 0)$，那么称该系统为无失真传输系统。对式(5.24)进行推导，得出无失真传输系统的幅频特性、相频特性须满足

$$|H(j\omega)|=k \tag{5.25}$$

$$\angle H(j\omega)=-\omega t_d \tag{5.26}$$

例 5.6.1　某系统的微分方程为 $y'(t)+y(t)=x'(t)-x(t)$，判断该系统是否是无失真传输系统。

解：直接画出该系统的频率响应，MATLAB 源代码如下

```
clc; clear all; close all;
a = [1 1];                                  % 分母多项式系数
b = [1 -1];                                 % 分子多项式系数
freqs(b,a);                                 % 直接画出连续时间系统的频响
```

程序运行结果如图 5.6.1 所示。观察发现，虽然系统的幅频特性恒为 1，满足式(5.25)，但是相频特性不是直线，不满足式(5.26)，因此该系统不是无失真传输系统。

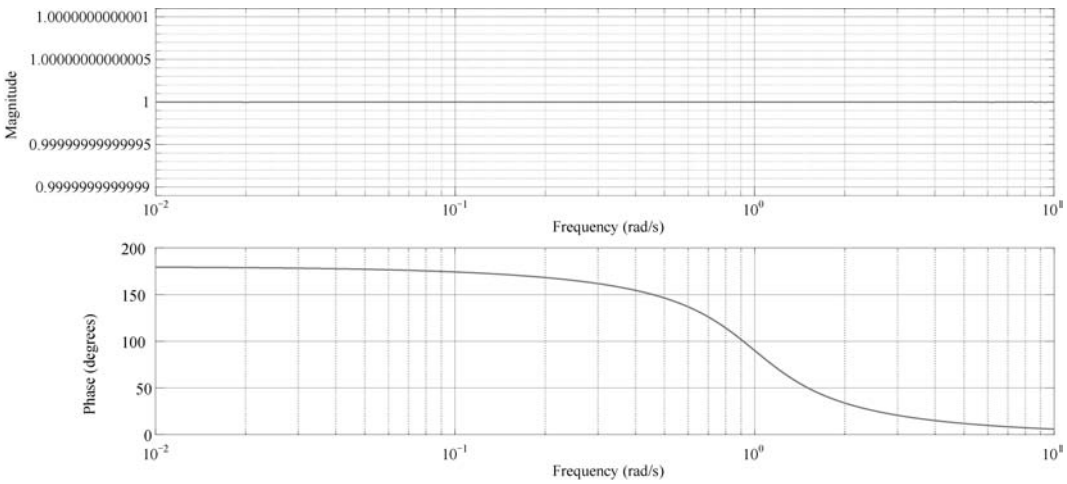


图 5.6.1　例 5.6.1 系统的频率响应

5.6.2　信号的无失真传输

对于信号来说，只要在其频带范围内，系统的频率响应满足式(5.25)、式(5.26)，该信号通过系统后是无失真的。

例 5.6.2　某系统频率响应如图 5.6.2 所示，分别判断信号 $\mathrm{Sa}(5t)$、$\mathrm{Sa}(5t)\mathrm{e}^{\mathrm{j}5t}$、$G_5(t)$、$\cos(6t)$通过该系统后是否失真。

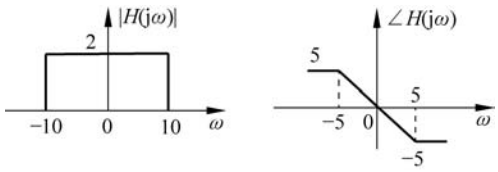


图 5.6.2　例 5.6.2 系统的频率响应

解：利用符号函数分别计算 4 个信号的傅里叶变换，MATLAB 源代码如下

```
close all; clc;clear all;
syms t w;                                   % 声明符号变量
% 用符号函数求傅里叶变换
```

```
Xjw1 = fourier(sinc(5/pi * t),w);
Xjw2 = fourier(sinc(5/pi * t) * exp(j * 5 * t),w);
Xjw3 = fourier(heaviside(t + 2.5) - heaviside(t - 2.5),w);
Xjw4 = fourier(cos(6 * t),w)
subplot(2,2,1); fplot(abs(Xjw1),[ - 12 12]);
xlabel('\omega');ylim([ - 0.1 0.7]); title('Sa(5t)的幅度谱')
subplot(2,2,2); fplot(abs(Xjw2),[ - 12 12]);
xlabel('\omega'); ylim([ - 0.1 0.7]); title('Sa(5t)e^{j5t}的幅度谱')
subplot(2,2,3); fplot(abs(Xjw3),[ - 12 12]);
xlabel('\omega'); ylim([ - 0.1 5.1]); title('G_{5}(t)的幅度谱')
subplot(2,2,4); fplot(abs(Xjw4),[ - 12 12]);
xlabel('\omega'); title('cos(6t)的幅度谱')
```

程序运行结果如图 5.6.3 所示。观察发现：

(1) Sa(5t)频带范围是[－5 5]，理论上通过系统后不失真；

(2) Sa(5t)e^{j5t} 频带范围是[0 10]，虽然系统幅频响应在这个范围内是常数 2，但相频响应不是直线，理论上通过系统后将出现失真；

(3) $G_5(t)$的频带范围超过了[－5 5]，理论上通过系统后也将出现失真；

(4) cos(6t)的幅度谱显示的是零，但命令窗口的输出是 Xjw4＝pi＊(dirac(w－6)＋dirac(w＋6))，也就是说，cos(6t)的频谱在±6 上是无穷大，只是无法在图中显示出来。

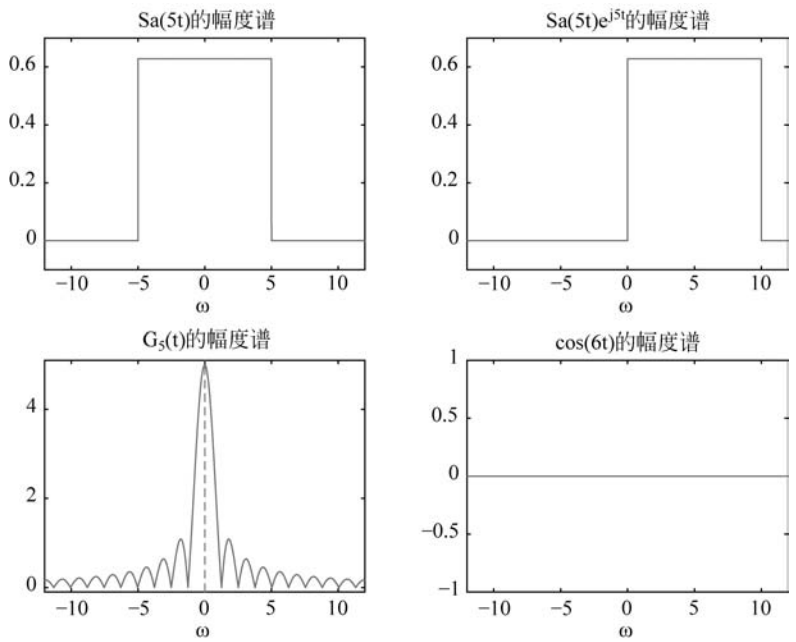


图 5.6.3　4 个信号的幅度谱

接下来从理论上分析 cos(6t)通过系统后是否失真。非常容易出错的一点是有读者认为 cos(6t)的频谱在±6 处有非零值，故其频带范围为[－6，6]。但频带范围指的是频

谱非0的频率点的集合，而$\cos(6t)$只有-6、6两个角频率，在这两个角频率上，幅度谱是常数，相位谱是过原点且斜率是$-\frac{5}{6}$的直线，因此该信号通过系统后无失真。或者由式(5.22)可知

$$\cos(6t) \rightarrow |H(\mathrm{j}6)|\cos(6t+\angle H(\mathrm{j}6)) = 2\cos(6t-5) \tag{5.27}$$

满足式(5.24)的定义。

例 5.6.3 画出例5.6.2中各信号通过系统的响应，验证该例题的结论。

解：按照先将输入信号的傅里叶变换与系统的频率响应相乘，再傅里叶反变换求输出的思路，MATLAB源代码如下

```
close all; clc;clear all;
dw = 1e-2;                                  %频谱抽样间隔
w = -12:dw:12;                              %信号频谱主要取值区间
%表示 H(jw)
Hjw_angle = -5*sign(w).*(1-rectpuls(w,10))-w.*rectpuls(w,10);
Hjw = 2*rectpuls(w,20).*exp(j*Hjw_angle);
%考虑到 cos 信号,用矩阵-向量乘法计算 X(jw)
dt = 1e-2;                                  %时域抽样间隔
t = -10:dt:10;                              %信号主要取值区间
xt1 = sinc(5/pi*t);                         %信号赋值
xt2 = sinc(5/pi*t).*exp(j*5*t);
xt3 = rectpuls(t,5);
xt4 = cos(6*t);
[W,T] = meshgrid(w,t);                      %生成矩阵
WT = exp(-j*T.*W);                          %傅里叶变换的矩阵
Xjw1 = dt*xt1*WT;                           %利用矩阵-向量乘法计算
Xjw2 = dt*xt2*WT;                           %利用矩阵-向量乘法计算
Xjw3 = dt*xt3*WT;                           %利用矩阵-向量乘法计算
Xjw4 = dt*xt4*WT;                           %利用矩阵-向量乘法计算
%输出傅里叶变换
Yjw1 = Hjw.*Xjw1;
Yjw2 = Hjw.*Xjw2;
Yjw3 = Hjw.*Xjw3;
Yjw4 = Hjw.*Xjw4;
%矩阵-向量法求傅里叶反变换
yt1 = dw/(2*pi)*Yjw1*WT';
yt2 = dw/(2*pi)*Yjw2*WT';
yt3 = dw/(2*pi)*Yjw3*WT';
yt4 = dw/(2*pi)*Yjw4*WT';
subplot(4,2,1); plot(t,xt1);
title('Sa(5t)'); xlim([-5 5]);
subplot(4,2,2); plot(t,yt1);
title('Sa(5t)通过系统后的输出'); xlim([-5 5])
subplot(4,2,3); plot(t,real(xt2));
title('Sa(5t)e^{j5t}的实部'); xlim([-5 5])
subplot(4,2,4); plot(t,real(yt2));
title('Sa(5t)e^{j5t}通过系统后输出的实部'); xlim([-5 5])
subplot(4,2,5); plot(t,xt3);
title('G_{5}(t)'); xlim([-5 5]); ylim([-0.1 1.1])
```

```
subplot(4,2,6); plot(t,yt3);
title('G_{5}(t)通过系统后的输出'); xlim([-5 5]); ylim([-0.5 2.8])
subplot(4,2,7); plot(t,xt4);
title('cos(6t)'); xlim([-5 5])
subplot(4,2,8); plot(t,yt4);
title('cos(6t)通过系统后的输出'); xlim([-5 5])
```

程序运行结果如图 5.6.4 所示。观察发现 Sa(5t)、cos(6t)通过系统后，输出信号波形除幅度等比例增大以及有一些延时外，未发生其他变化，信号通过系统后未出现失真；而 $Sa(5t)e^{j5t}$、$G_5(t)$的波形发生了改变，信号通过系统后出现了失真，例 5.6.2 的结论正确。

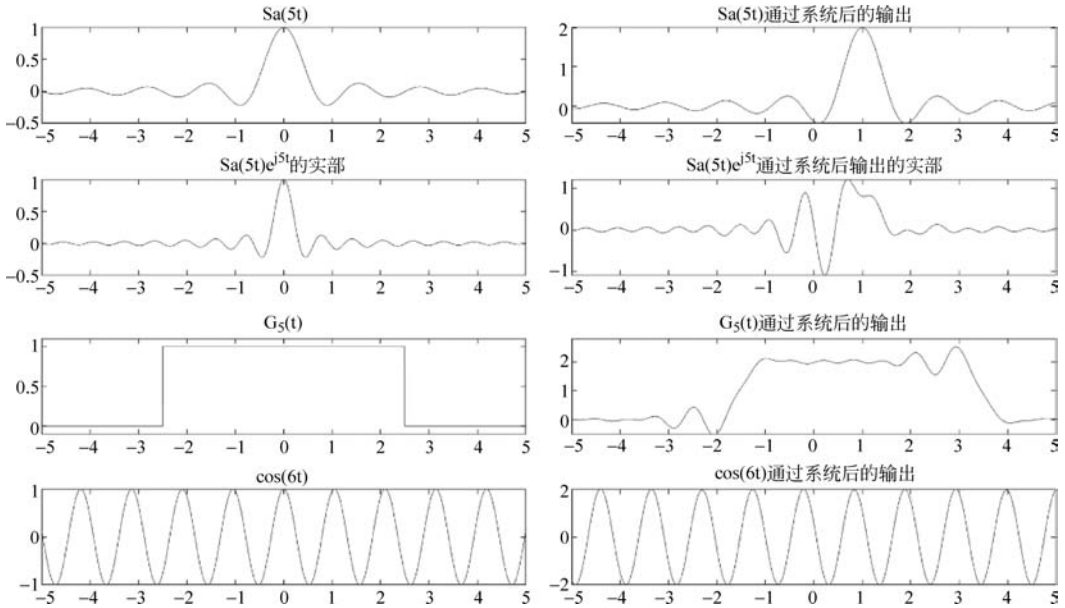


图 5.6.4　例 5.6.3 输入、输出信号波形

5.7　理想与实际滤波器

5.7.1　理想滤波器

例 5.7.1　理想低通滤波器的频率响应如图 5.7.1 所示。分别讨论截止角频率 ω_c 和 t_d 对滤波器单位冲激响应的影响。

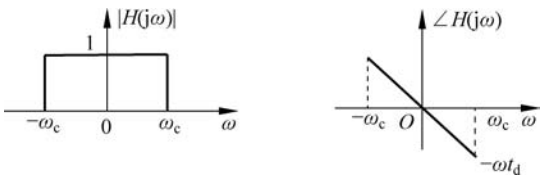


图 5.7.1　理想低通滤波器的频率响应

解：由于可以写出频率响应的解析表达式，所以可以利用符号函数计算频率响应的傅里叶反变换，得到单位冲激响应。先保持相频响应的斜率 $-t_d$ 不变，增加 ω_c。MATLAB 源代码如下

```
close all; clc;clear all;
syms t w
td = 1;                                    % 相频特性的负斜率
wc = 1:3;
for i = 1:3
  Hjw = (heaviside(w + wc(i)) - heaviside(w - wc(i))) * exp( - j * w * td);
  xt = ifourier(Hjw,w,t);                  % 用符号函数求傅里叶变换
  subplot(3,3,3 * i - 2); fplot(abs(Hjw));
  ylim([ - 0.1 1.1]);xlabel('\omega');
  title(['\omega_c = ',num2str(wc(i)),'时的幅频响应'])
  subplot(3,3,3 * i - 1); fplot(angle(Hjw)); xlabel('\omega');
  title(['\omega_c = ',num2str(wc(i)),'时的相频响应'])
  subplot(3,3,3 * i); fplot(xt);
  ylim([ - 0.3 1.3]); xlabel('t');
  title(['\omega_c = ',num2str(wc(i)),'时的单位冲激响应'])
  xticks([ - 5 0 td 5])                    % 在横坐标上显示 td
end
```

程序运行结果如图 5.7.2 所示，动态结果请扫描二维码。随着截止角频率 ω_c 的增加，单位冲激响应保持中心点在 t_d 处不变，但脉冲宽度不断减小，最大值不断增加。

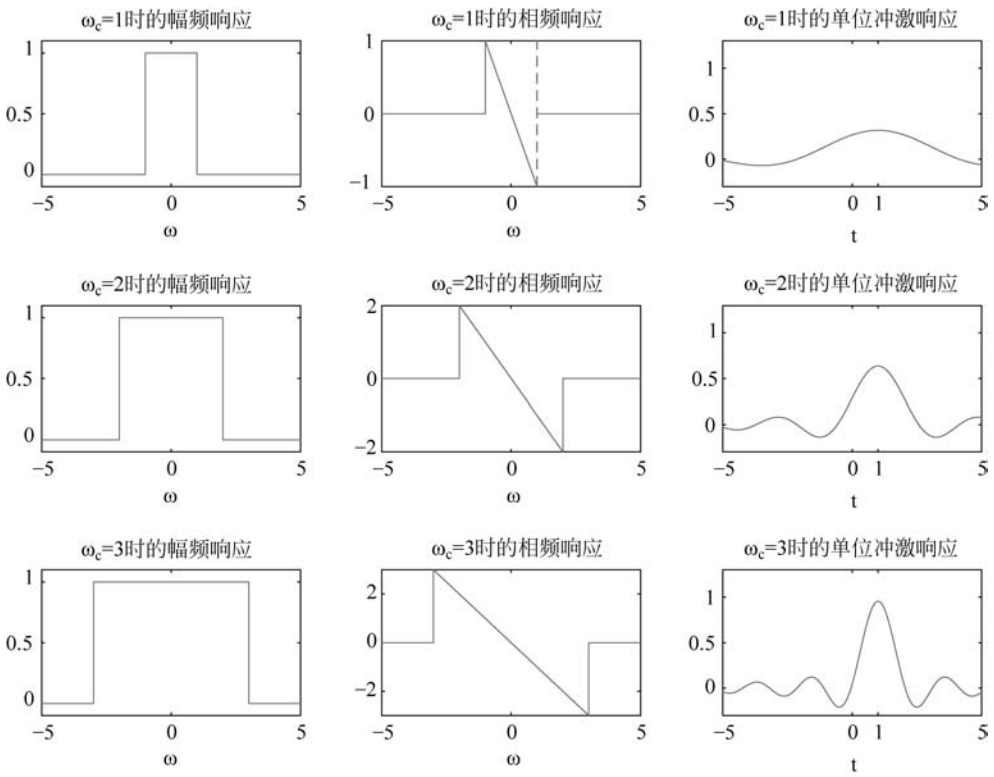


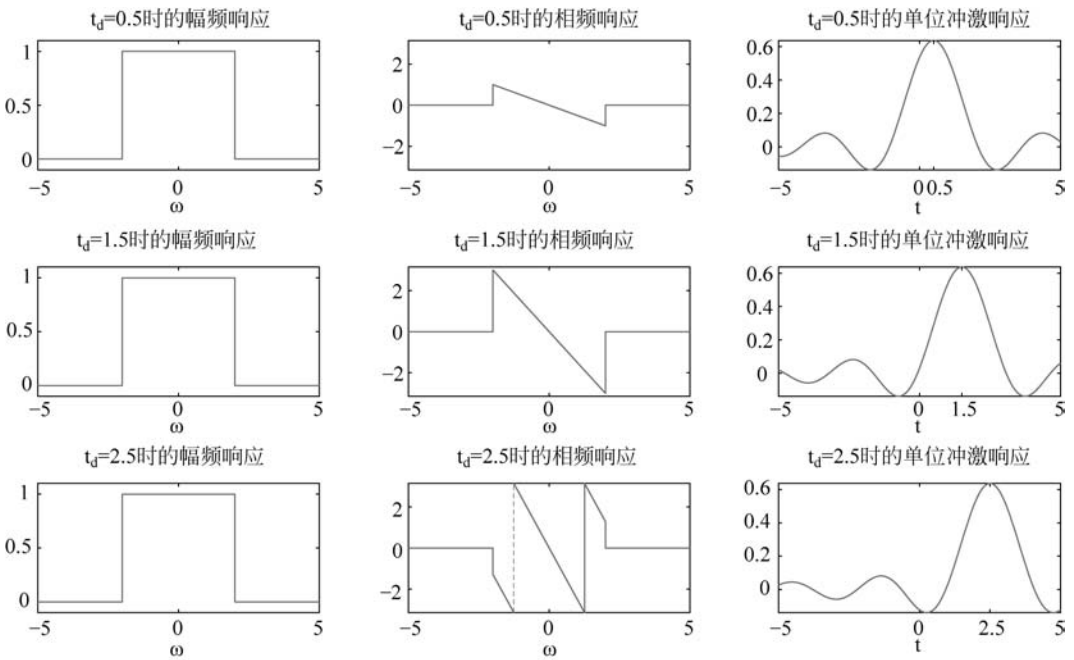
图 5.7.2　ω_c 对滤波器单位冲激响应的影响

动图

再保持 ω_c 不变，增加相频响应的负斜率 t_d。MATLAB 源代码如下

```
close all; clc;clear all;
syms t w
wc = 2;
td = 0.5:1:2.5;
for i = 1:3
  Hjw = (heaviside(w + wc) - heaviside(w - wc)) * exp( - j * w * td(i));
  xt = ifourier(Hjw,w,t);                    % 用符号函数求傅里叶变换
  subplot(3,3,3 * i - 2); fplot(abs(Hjw));
  ylim([ - 0.1 1.1]); xlabel('\omega');
  title(['td = ',num2str(td(i)),'时的幅频响应'])
  subplot(3,3,3 * i - 1); fplot(angle(Hjw));
  ylim([ - pi pi]); xlabel('\omega');
  title(['td = ',num2str(td(i)),'时的相频响应'])
  subplot(3,3,3 * i); fplot(xt); xlabel('t');
  title(['td = ',num2str(td(i)),'时的单位冲激响应'])
  xticks([ - 5 0 td(i) 5])                   % 在横坐标上显示 td
end
```

程序运行结果如图 5.7.3 所示，动态结果请扫描二维码。随着 t_d 的增加，单位冲激响应保持脉冲宽度、最大值不变，中心不断右移。单位冲激响应的中心点即为相频响应的负斜率 t_d。



动图

图 5.7.3　t_d 对滤波器单位冲激响应的影响

5.7.2 实际滤波器

若根据系统的微分方程来判断该系统的滤波特性，可通过画出系统的频率响应实现。

例 5.7.2 某因果系统的微分方程为 $y''(t)+y'(t)+y(t)=x''(t)$，判断该系统的滤波特性。

解：MATLAB 源代码如下

```
close all; clc;clear all;
a = [1 1 1];                          % 分母多项式系数
b = [1 0 0];                          % 分子多项式系数
freqs(b,a);                           % 画出频谱图的波特图
```

程序运行结果如图 5.7.4 所示，观察幅度特性发现该系统为高通滤波器。

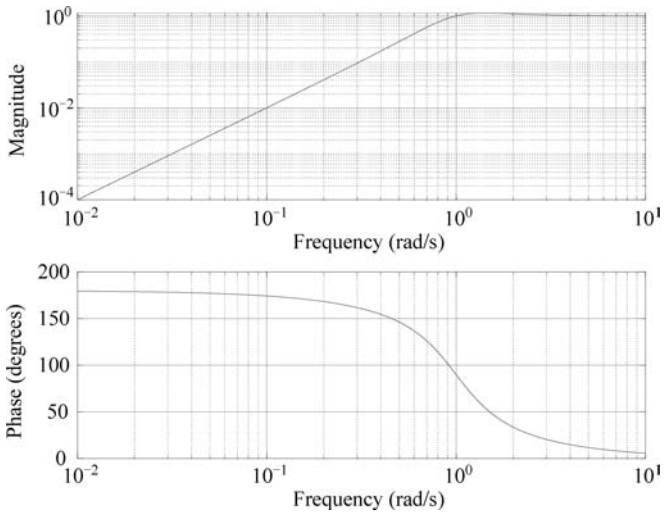


图 5.7.4 某系统的频率响应

5.7.3 利用 Simulink 实现信号的滤波

Simulink 中提供了滤波器模块用以实现信号的滤波。这些模块位于 DSP System Toolbox\Filtering\Filter Designs 子库中。

例 5.7.3 借助 Simulink，将信号 $x(t)=\sin(2\pi\cdot 500t)+\sin(2\pi\cdot 1000t)+\sin(2\pi\cdot 2500t)$ 中的每个正弦信号分别取出来。

解：Simulink 模型如图 5.7.5 所示。通过将 3 个正弦信号相加产生需要的输入信号 $x(t)$。需注意的是，在用 Sine Wave 模块产生正弦信号时，Sample time 要小于 $1/(2\times 2500)$，即抽样频率要大于 5000Hz，并且该频率要与后面的滤波器模块中的抽样频率保

持一致，才可以得到正确的滤波结果。这里抽样频率设置为 44100Hz，该频率是常用的声音抽样频率。

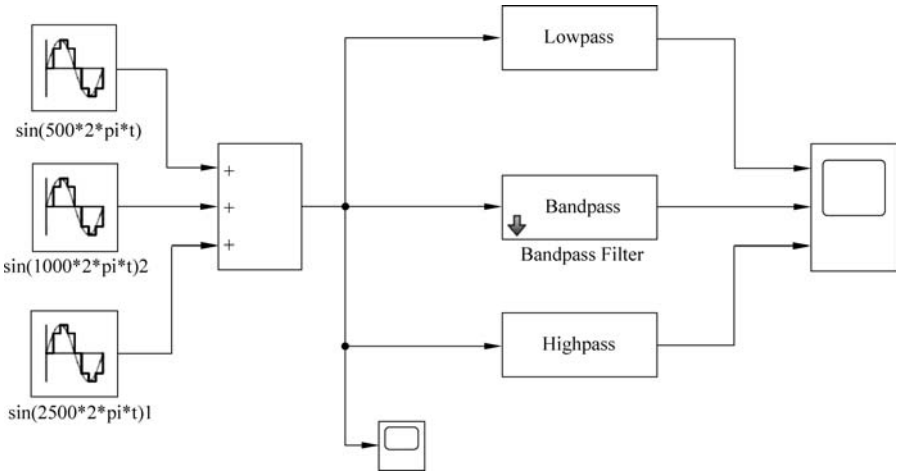


图 5.7.5　例 5.7.3 的 Simulink 模型图

分别采用 Lowpass、Bandpass、Highpass 模块进行滤波，以得到 $\sin(2\pi \cdot 500t)$、$\sin(2\pi \cdot 1000t)$、$\sin(2\pi \cdot 2500t)$三个信号，注意滤波器模块中频率参数要留有一定的余量。根据需求分析，低通滤波器要保证通带截止频率大于 500Hz，阻带截止频率小于 1000Hz；高通滤波器要保证阻带截止频率大于 1000Hz，通带截止频率小于 2500Hz；而带通滤波器要保证阻带截止频率 1 大于 500Hz，通带截止频率 1 小于 1000Hz，通带截止频率 2 大于 1000Hz，阻带截止频率 2 小于 2500Hz。Bandpass 模块的实际参数设置如图 5.7.6 所示。需要注意的是，Frequency units(频率单位)选择 Hz，其值应与前文信号

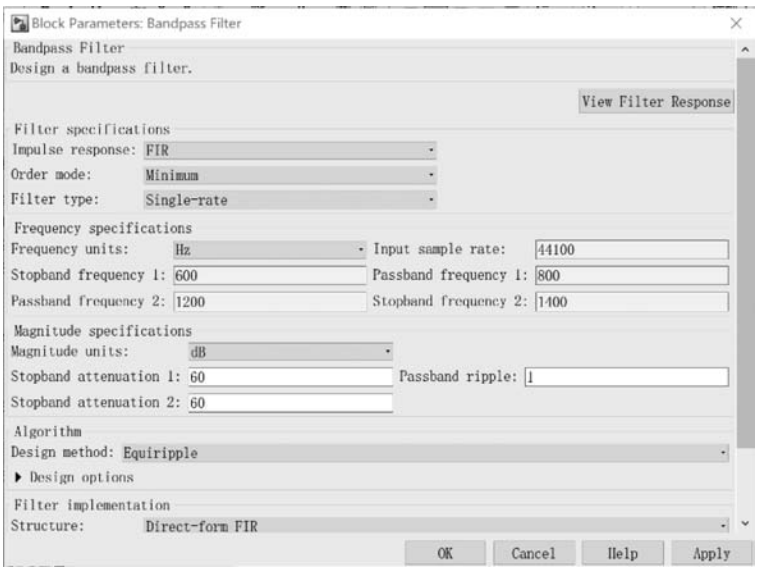


图 5.7.6　Bandpass 模块的参数设置

产生模块的抽样频率保持一致。设定好参数后，可以通过单击 View Filter Response 按钮观察滤波器模块的各种响应，如幅频响应，如图 5.7.7 所示。在此界面，可以将滤波器的分子、分母多项式系数保存到文本文档，以作他用。

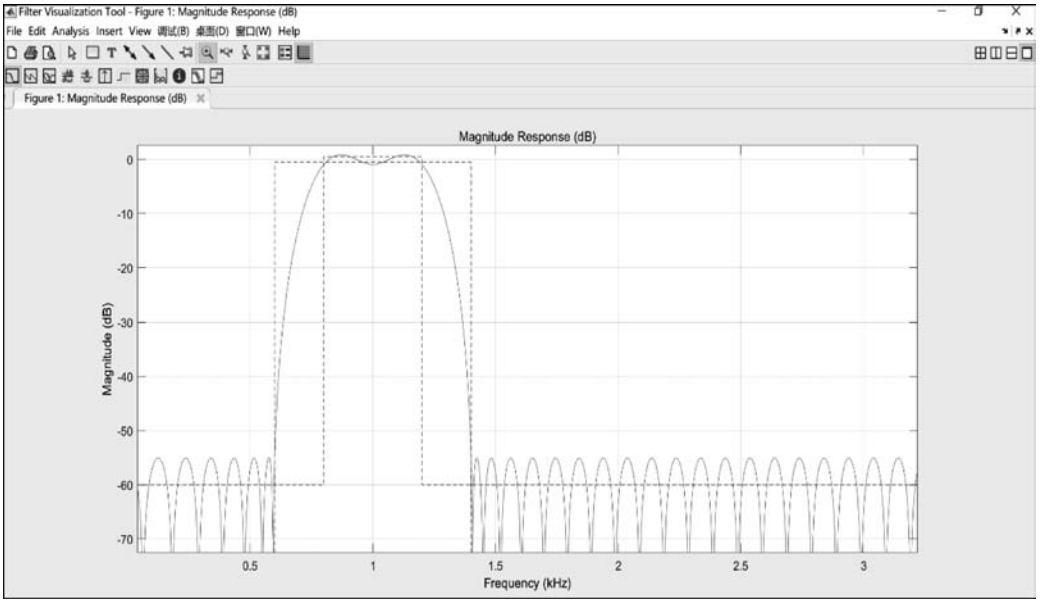


图 5.7.7　Bandpass 模块的幅频响应

仿真时间设置为 1s，通过 Scope 模块观察输入、输出信号的波形，结果如图 5.7.8 和图 5.7.9 所示。

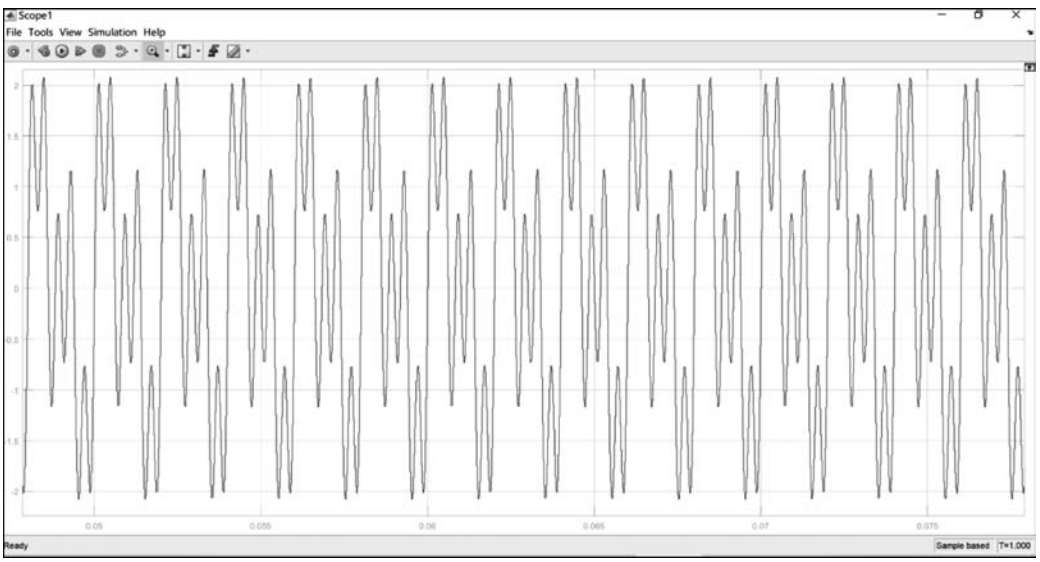


图 5.7.8　例 5.7.3 的输入信号

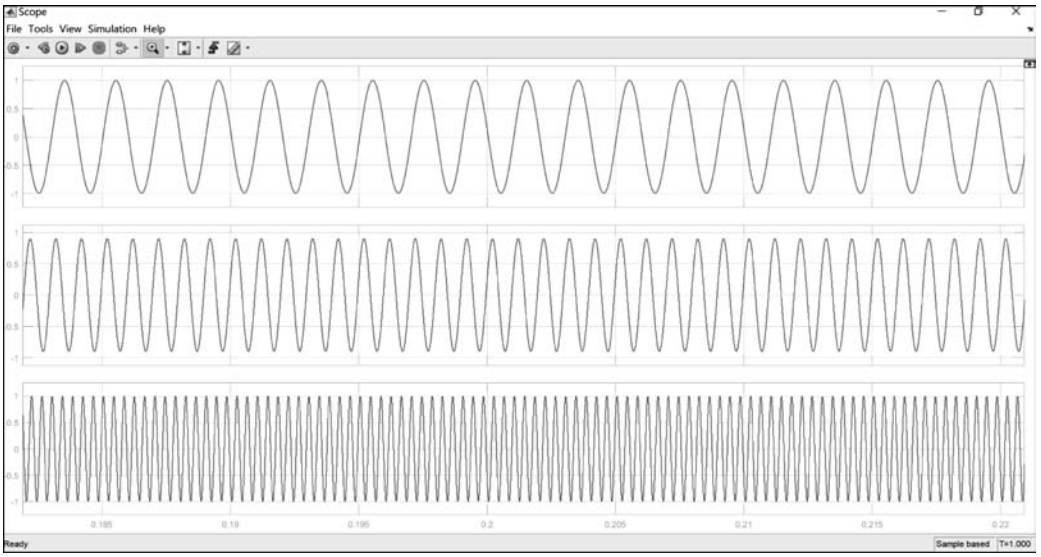


图 5.7.9　例 5.7.3 的 3 个滤波器的输出信号

5.8　时域抽样和恢复

5.8.1　信号的时域抽样

工程中,信号的时域抽样就是从连续时间信号 $x(t)$中抽取出一系列离散样本值。抽样间隔一般是恒定的,记作 T_s,那么抽样得到的离散样本值是

$$x(t)\big|_{t=nT_s}=x(nT_s)\triangleq x(n) \tag{5.28}$$

这里的 $x(n)$即为离散时间信号。

例 5.8.1　设连续时间信号 $x(t)=\mathrm{Sa}^2(t)$,抽样间隔 $T_s=0.2\mathrm{s}$,画出被抽信号和抽样信号的波形图。

解:MATLAB 源代码如下

```
clc; clear all; close all;
t = -10:0.01:10;
xt = sinc(t/pi).^2;                                  % 被抽信号
Ts = 0.2;                                            % 抽样间隔为 0.2s
n = round(min(t)/Ts):round(max(t)/Ts);               % 离散时间信号的自变量
xn = sinc(n*Ts/pi).^2;                               % 抽样信号
subplot(2,1,1); plot(t,xt); title('连续时间信号 x(t)') % 画图
subplot(2,1,2); stem(n*Ts,xn,'filled'); title('连续时间信号的抽样序列 x(n)')
```

程序运行结果如图 5.8.1 所示。

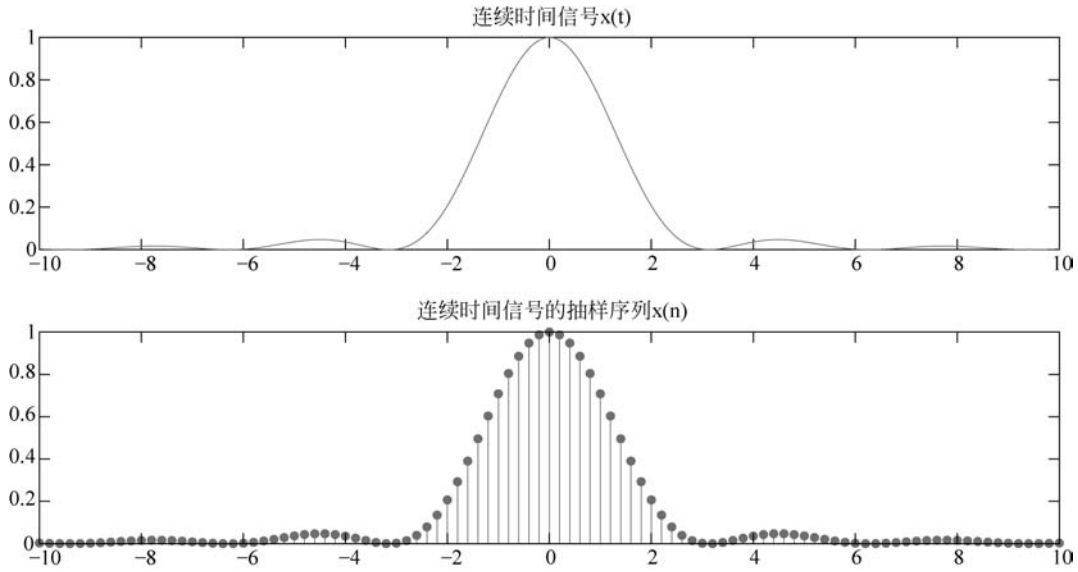


图 5.8.1　连续时间信号的抽样

5.8.2　抽样信号的频谱

为了便于理论分析，本书中一般将周期冲激串作为抽样脉冲，得到

$$x_s(t)=x(t)\cdot\delta_T(t)=\sum_{n=-\infty}^{\infty}x(nT_s)\delta(t-nT_s) \tag{5.29}$$

经过推导

$$X_s(\mathrm{j}\omega)=\frac{1}{T_s}\sum_{k=-\infty}^{\infty}X[\mathrm{j}(\omega-k\omega_s)] \tag{5.30}$$

从而得出结论：抽样信号的频谱是被抽信号频谱的周期延拓，延拓周期是抽样角频率 ω_s。若按照 5.8.1 节的方法进行抽样，离散时间序列 $x(n)$ 的频谱是否仍旧是被抽信号频谱的周期延拓，延拓的周期是否仍旧是 ω_s 呢？通过例 5.8.2 回答上述问题。

例 5.8.2　画出例 5.8.1 中被抽信号和抽样序列的频谱图。

解：MATLAB 源代码如下

```
clc; clear all; close all;
%%%时域%%%%%%%%%%
dt = 0.01;
t = -10:dt:10;                                  %时域自变量
xt = sinc(t/pi).^2;                             %被抽样信号
Ts = 0.5;                                       %抽样间隔
n = round(min(t)/Ts):round(max(t)/Ts);          %离散时间信号的自变量
nTs = n*Ts;
```

```
xn = sinc(nTs/pi).^2;                          %抽样信号
%%%%频域%%%%%%%%%
%被抽信号
w = -30:0.1:30;                                %频域自变量
[ww,tt] = meshgrid(w,t);                       %变为矩阵形式
Xjw = dt*xt*exp(-j*tt.*ww);                    %矩阵-向量乘法计算被抽信号频谱
%抽样信号
[wws,tts] = meshgrid(w,nTs);                   %变为矩阵形式
Xs_jw = Ts*xn*exp(-j*tts.*wws);
subplot(2,2,1); plot(t,xt);
xlabel('t'); title('连续时间信号 x(t)')        %画图
subplot(2,2,3); stem(nTs,xn,'.');
xlabel('t'); title('连续时间信号的抽样序列 x(n)')
subplot(2,2,2); plot(w,Xjw);
xlabel('\omega'); title('连续时间信号的频谱')
subplot(2,2,4); plot(w,Xs_jw);
xlabel('\omega'); title('抽样序列的频谱')
xticks([min(w) -2*pi/Ts 0 2*pi/Ts max(w)])     %在横坐标上显示 ws
```

程序运行结果如图 5.8.2 所示。观察发现,抽样序列的频谱是被抽信号频谱的周期延拓,延拓的周期是抽样角频率 ω_s。只不过抽样序列频谱的主周期和被抽信号频谱相等,而不像式(5.30)那样有个系数$\frac{1}{T_s}$。这是因为式(5.30)假设抽样脉冲是周期冲激串,而实际应用中并不是这样。

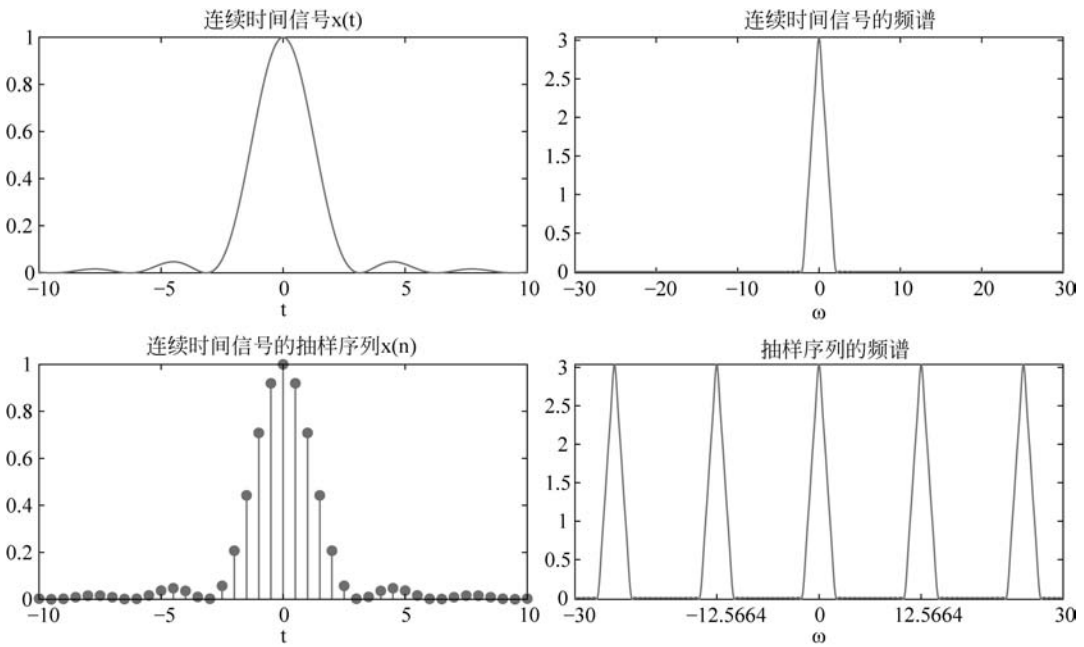


图 5.8.2　抽样前后信号的波形及频谱

5.8.3 时域抽样定理

根据5.8.2节的介绍，抽样序列的频谱仍然是被抽信号频谱的周期延拓，且延拓的周期仍然是抽样角频率 ω_s，因此若要保证抽样序列保留了被抽信号的所有信息，仍然需要满足

$$\omega_s \geqslant 2\omega_m \tag{5.31}$$

其中，ω_m 是被抽信号的最高角频率。对应到抽样间隔 T_s，需要满足

$$T_s \leqslant \frac{\pi}{\omega_m} \tag{5.32}$$

例 5.8.3 利用例5.8.2，验证时域抽样定理。

解：例5.8.2中，被抽信号 $x(t)=\mathrm{Sa}^2(t)$，其傅里叶变换 $X(\mathrm{j}\omega)=\pi\Lambda_4(\omega)$，最高角频率 $\omega_m=2\mathrm{rad/s}$，故抽样间隔应小于$\frac{\pi}{2}$s。为便于观察，保持被抽信号不变，不断增加抽样间隔，MATLAB源代码如下

```
clc; clear all; close all;
%%%被抽信号%%%%%%%%%%
dt = 0.01;
t = -10:dt:10;                                  %时域自变量
xt = sinc(t/pi).^2;                             %被抽样信号
w = -30:0.1:30;                                 %频域自变量
[ww,tt] = meshgrid(w,t);                        %变为矩阵形式
Xjw = dt*xt*exp(-j*tt.*ww);                     %矩阵-向量乘法计算被抽信号频谱
Ts = 0.3*pi;                                    %抽样间隔,分别选取0.3π和0.6π
n = round(min(t)/Ts):round(max(t)/Ts);          %离散时间信号的自变量
nTs = n*Ts;
xn = sinc(nTs/pi).^2;                           %抽样信号
[wws,tts] = meshgrid(w,nTs);                    %变为矩阵形式
Xs_jw = Ts*xn*exp(-j*tts.*wws);
figure;
subplot(2,2,1); plot(t,xt);
xlabel('t'); title('连续时间信号 x(t)')         %画图
subplot(2,2,3); stem(nTs,xn,'filled');
xlabel('t'); title('抽样序列 x(n)')
subplot(2,2,2); plot(w,Xjw);
xlabel('\omega'); title('连续时间信号的频谱');xlim([-20 20])
subplot(2,2,4); plot(w,Xs_jw);
xlabel('\omega');
hold on; plot([-2 2],[pi pi],'r')
plot([-2,-2],[0,pi],'r')
plot([2,2],[0,pi],'r');
hold off
title(['抽样序列的频谱,抽样间隔为',num2str(Ts/pi),'π']);xlim([-20 20]);
```

程序运行结果如图 5.8.3 和图 5.8.4 所示，动态结果请扫描二维码。为便于比较，图中用方框框出了[－2,2]的角频率范围。当抽样间隔不满足式(5.32)时，抽样序列的频谱将出现混叠现象。

动图

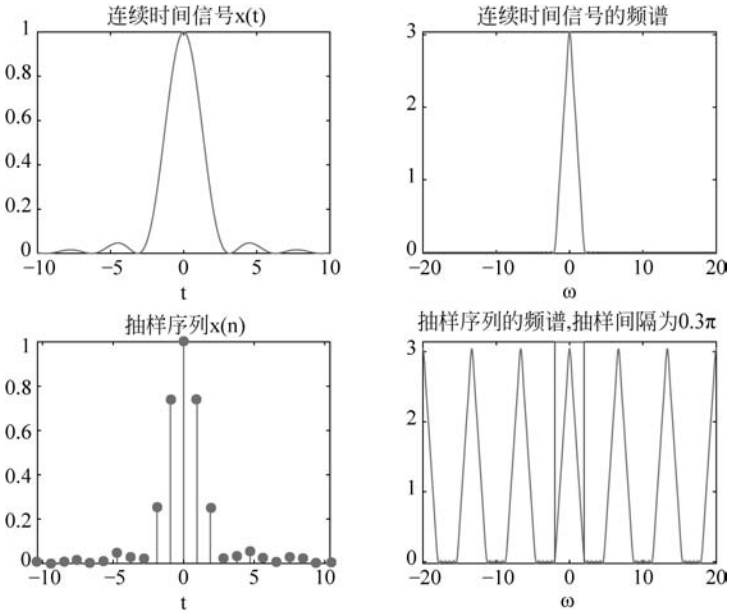


图 5.8.3　抽样间隔为 0.3π 的结果

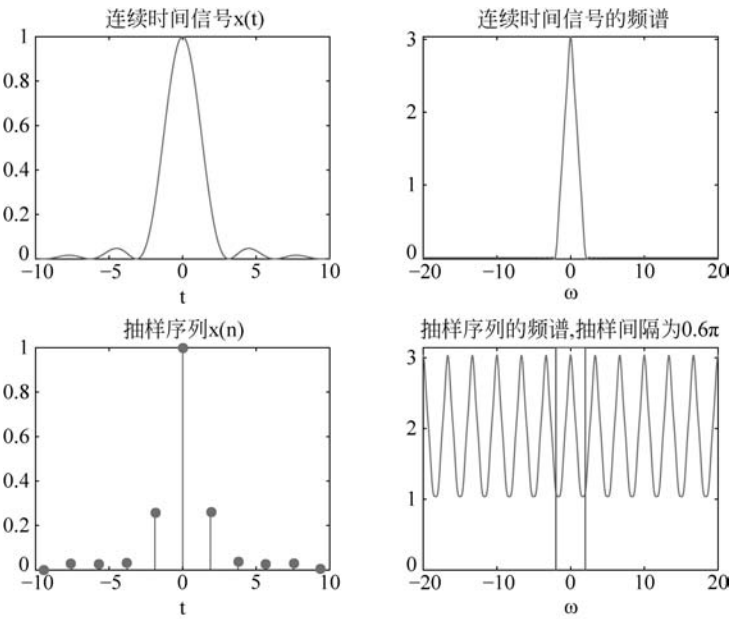


图 5.8.4　抽样间隔为 0.6π 的结果

5.8.4 抽样信号的恢复

在满足时域抽样定理的前提下，理论教材的结论是可以将抽样信号经过模拟理想低通滤波器恢复出来。需要注意的是，工程中抽样序列是离散的，无法经过模拟滤波器，通常做法是将其通过数模转换器(DAC)。数模转换器主要采用零阶抽样保持或一阶抽样保持，示意图如图 5.8.5 和图 5.8.6 所示。

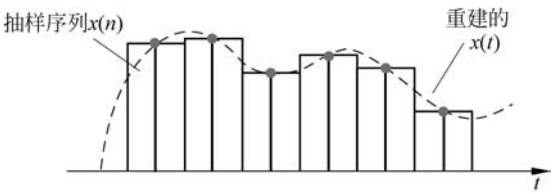


图 5.8.5 零阶抽样保持示意图

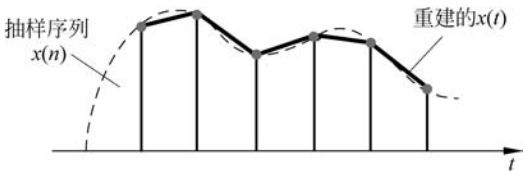


图 5.8.6 一阶抽样保持示意图

例 5.8.4 绘制抽样、恢复过程中信号的波形及频谱。

解：对例 5.8.2 中的抽样信号分别进行零阶、一阶抽样保持。在 MATLAB 中，可以采用线性插值函数 interp1 完成一阶抽样保持，但没有直接实现零阶抽样保持的函数，需要自行编程实现。MATLAB 源代码如下

```
clc; clear all; close all;
%%%时域%%%%%%%%%%
dt = 0.01;
t = -10:dt:10;                          %被抽信号时域自变量
xt = sinc(t/pi).^2;                     %被抽样信号
Ts = 0.5;                               %抽样间隔
n = round(min(t)/Ts):round(max(t)/Ts);  %离散时间信号的自变量
nTs = n*Ts;
xn = sinc(nTs/pi).^2;                   %抽样信号
nKeep = 50;                             %恢复时的保持点数
yt0 = repmat(xn,nKeep,1);               %零阶抽样保持信号
yt0 = transpose(yt0(:));
nt = linspace(-10,10,length(yt0));      %恢复信号的时域自变量
yt1 = interp1(nTs,xn,nt,'linear');      %一阶抽样保持信号
%%%%频域%%%%%%%%%
w = -30:0.1:30;                         %频域自变量
%被抽信号
[ww,tt] = meshgrid(w,t);                %变为矩阵形式
Xjw = dt*xt*exp(-j*tt.*ww);             %矩阵-向量乘法计算被抽信号频谱
%抽样信号
[wws,tts] = meshgrid(w,nTs);            %变为矩阵形式
Xs_jw = Ts*xn*exp(-j*tts.*wws);         %抽样序列频谱
%恢复信号
[wwr,ttr] = meshgrid(w,nt);             %变为矩阵形式
```

```
Yjw_0 = Ts/nKeep * yt0 * exp( - j * ttr. * wwr);      % 零阶抽样保持信号的频谱
Yjw_1 = Ts/nKeep * yt1 * exp( - j * ttr. * wwr);      % 一阶抽样保持信号的频谱
subplot(4,2,1); plot(t,xt);
xlabel('t'); title('被抽信号')                        % 画图
subplot(4,2,3); stem(nTs,xn,'filled');
xlabel('t'); title('抽样序列')
subplot(4,2,5); plot(nt,yt0);
xlabel('t'); title('零阶抽样保持恢复信号')            % 画图
subplot(4,2,7); plot(nt,yt1);
xlabel('t'); title('一阶抽样保持恢复信号')
subplot(4,2,2); plot(w,Xjw);
xlabel('\omega'); title('被抽信号的频谱')
subplot(4,2,4); plot(w,Xs_jw);
xlabel('\omega'); title('抽样序列的频谱')
subplot(4,2,6); plot(w,Yjw_0);
xlabel('\omega'); title('零阶抽样保持恢复信号的频谱')
subplot(4,2,8); plot(w,Yjw_1);
xlabel('\omega'); title('一阶抽样保持恢复信号的频谱')
```

程序运行结果如图 5.8.7 所示。与被抽信号频谱相比，零阶抽样保持恢复信号的频谱在低频部分没有发生变化，而在高频部分多了一些成分，这是由于时域跳变引起的；而一阶抽样保持恢复信号的频谱不仅低频部分保持较好，而且由于时域跳变较小，高频部分的变化也较小，相比而言恢复效果更好。

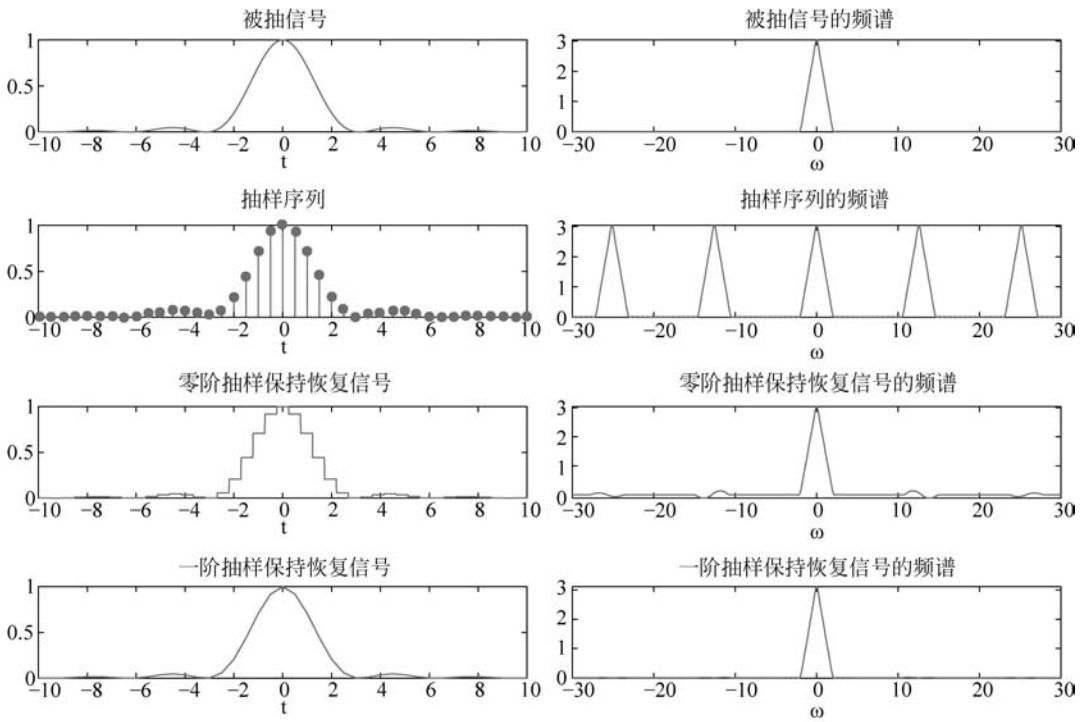


图 5.8.7　抽样、恢复信号