

# 第3章

数字基带信号与发送滤波器的MATLAB/Simulink仿真

数字通信系统中,从信源送出的数字信息可以表示成数字码元序列,数字基带信号是数字码元序列的脉冲电压或电流表现形式。数字基带信号的波形可以采用方波、三角波、升余弦波形等。由于数字信道的特性及要求不同,例如,很多信道不能传输信号的直流分量和频率很低的分量。另外,为了在接收端得到每个码元的起止时刻,需要在发送的信号中带有码元起止时刻的信息,为此需要将原码元依照一定的规则转换成适合信道传输要求的传输码(又称作线路码)。

为了保证信号在传输时不出现或少出现码间干扰,基带传输系统的设计必须满足奈奎斯特第一准则。满足奈奎斯特第一准则的基带传输系统有很多种,最简单的一种就是理想低通系统。但实际传输中,不可能有绝对理想的基带传输系统,常常采用具有奇对称“滚降”特性的低通滤波器(发送滤波器)对信号进行成形滤波。本章将介绍基带信号波形、码型以及成形滤波器的仿真方法。

### 3.1 数字基带信号波形仿真

数字基带信号的波形经常采用方波,其中最基本的二进制基带信号波形有单极性归零波形、单极性不归零波形、双极性归零波形、双极性不归零波形,如图 3-1 所示。

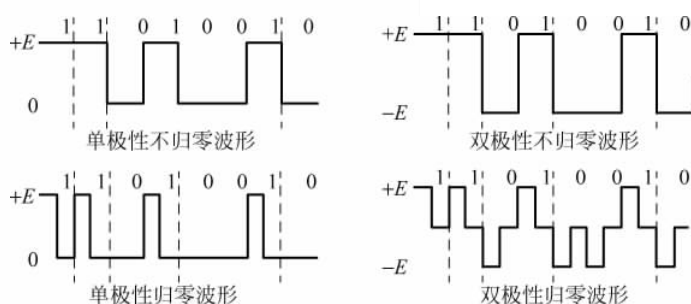


图 3-1 常见的基带信号波形

#### (1) 数字基带信号波形的 MATLAB 仿真

下面通过 MATLAB 程序来仿真一串随机消息代码的基带信号波形,首先产生 1000 个随机信号序列,分别用单极性归零码、单极性不归零码、双极性归零码和双极性不归零码编码,并且求平均功率谱密度。其中基本波形的 MATLAB 仿真流程如图 3-2 所示。

源代码(以双极性为例)如下:

```
close all
clear all
k = 14;           % 采样点数的设置
L = 32;          % 每码元采样数的设置
N = 2 ^ k;
M = N/L;         % M 为码元个数
dt = 1/L;       % 时域采样间隔
T = N * dt;     % 时域截断区间
```

```

df = 1.0/T; % 频域采样间隔
Bs = N * df/2; % 频域截断区间
t = linspace(-T/2,T/2,N); % 产生时域采样点
f = linspace(-Bs,Bs,N); % 产生频域采样点
EP1 = zeros(size(f));
EP2 = zeros(size(f));
EP3 = zeros(size(f));

% 程序第1部分: 随机产生1000列0、1信号序列,分别对其进行归零编码和不归零编码,并且求各自的功率谱密度,求功率谱密度的均值
for x = 1:1000 % 取样1000次
    k = round(rand(1,M)); % 产生一个长度为M的随机序列k,0和1等概出现
    nrz = zeros(L,M); % 产生一个L行M列的不归零矩阵,初始化为全0矩阵
    rz = zeros(L,M); % 产生一个L行M列的归零矩阵,初始化为全0矩阵
    for i = 1:M
        ifk(i) == 1
            nrz(:,i) = 1; % 使不归零矩阵第i列全部元素都为1
            rz(1:L/2,i) = 1; % 使归零矩阵第i列前L/2个元素为1
        else
            nrz(:,i) = -1; % 使不归零矩阵第i列全部元素都为-1
            rz(1:L/2,i) = -1; % 使归零矩阵第i列前L/2个元素为-1
        end
    end
    % 分别重排nrz,rz矩阵为1行N列的矩阵
    nrz = reshape(nrz,1,N);
    rz = reshape(rz,1,N);
    % 做傅里叶变换并算出功率谱密度
    NRZ = t2f(nrz,dt);
    P1 = NRZ. * conj(NRZ)/T;
    RZ = t2f(rz,dt);
    P2 = RZ. * conj(RZ)/T;
    % 求功率谱密度的均值
    EP1 = (EP1 * (x-1) + P1)/x;
    EP2 = (EP2 * (x-1) + P2)/x;
end

% 程序第2部分: 绘制波形图和功率谱密度曲线
figure(1) % 开启一个编号为1的绘图窗口
subplot(2,2,1);plot(t,nrz) % 画双极性不归零码的时域图
axis([-5,5,min(nrz)-0.1,max(nrz)+0.1])
title('双极性不归零码','fontsize',12)
xlabel('t(ms)','fontsize',12)
ylabel('nrz(t)','fontsize',12)
grid on
subplot(2,2,2);plot(t,rz) % 画双极性归零码的时域图
axis([-5,5,min(rz)-0.1,max(rz)+0.1])
title('双极性归零码','fontsize',12)

```

```

xlabel('t(ms)', 'fontsize', 12)
ylabel('rz(t)', 'fontsize', 12)
grid on
subplot(2,2,3);plot(f,EP1) %画双极性不归零码的功率谱密度图
axis([-5,5,0,1.2])
title('双极性不归零码功率谱密度图', 'fontsize', 12)
xlabel('f(kHz)', 'fontsize', 12)
ylabel('P1(f)', 'fontsize', 12)
grid on
subplot(2,2,4);plot(f,EP2) %画双极性归零码的功率谱密度图
axis([-5,5,0,0.3])
title('双极性归零码功率谱密度图', 'fontsize', 12)
xlabel('f(kHz)', 'fontsize', 12)
ylabel('P2(f)', 'fontsize', 12)
grid on

```

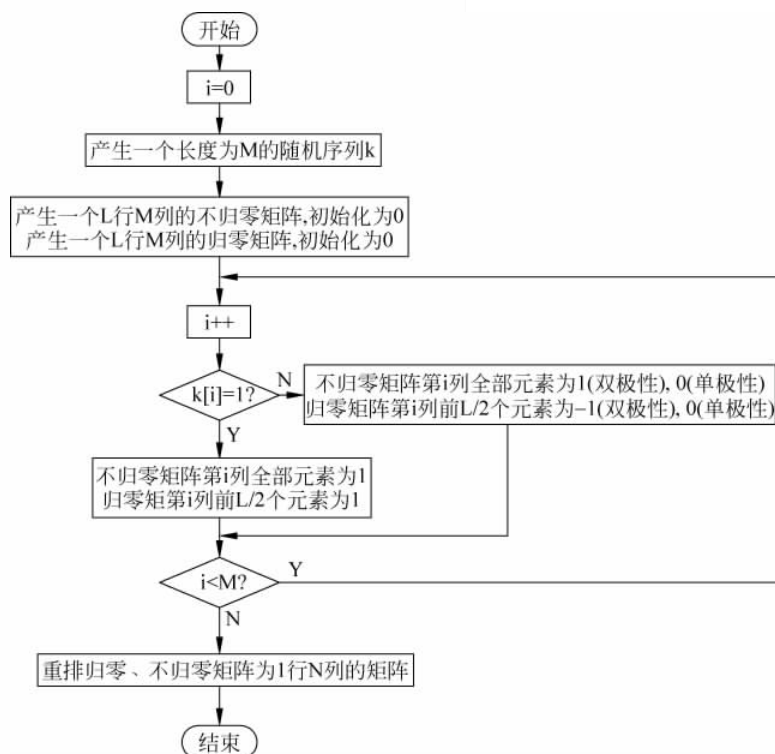


图 3-2 二进制基带信号仿真流程图

上面程序中需要调用一个傅里叶变换的函数  $t2f$ , 该函数定义如下:

```

% 将时域信号做傅里叶变换到频域, x 必须是二阶的矩阵, dt 是信号的时域分辨率
function X = t2f(x, dt)
X = fftshift(fft(x)) * dt;

```

双极性和单极性二进制信号的波形及功率谱分别如图 3-3、图 3-4 所示。

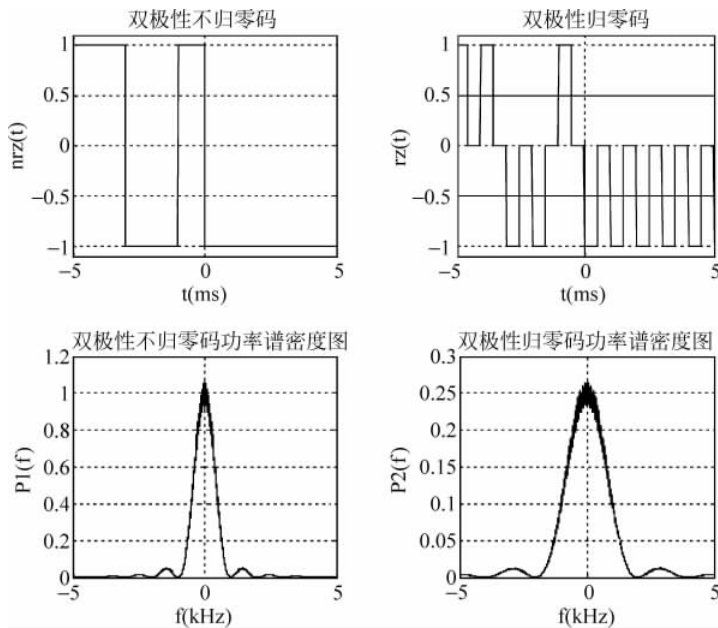


图 3-3 双极性二进制信号波形的 MATLAB 仿真结果

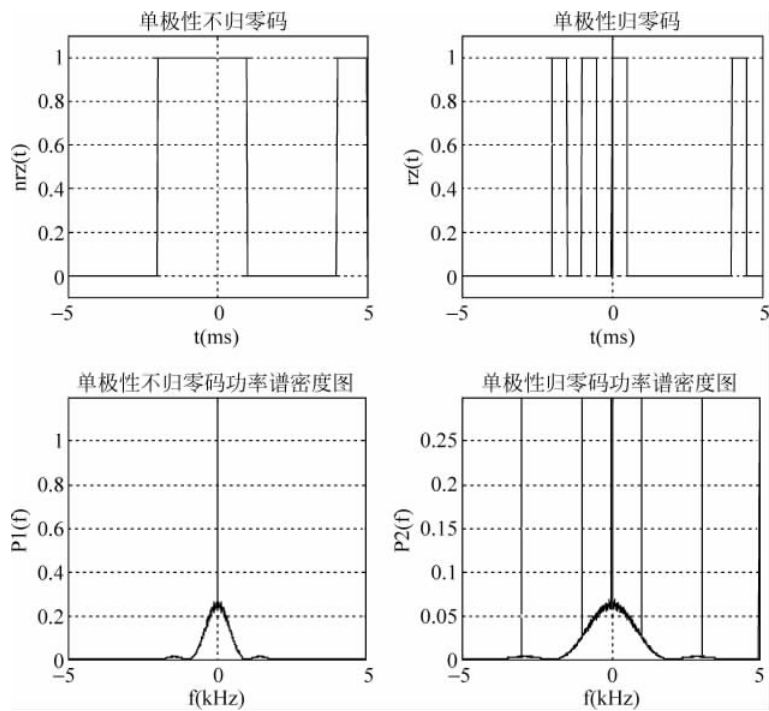


图 3-4 单极性二进制信号波形的 MATLAB 仿真

(2) 数字基带信号波形的 Simulink 仿真

用 Simulink 实现对单极性归零波形、单极性不归零波形、双极性归零波形和双极性不归零波形的仿真,仿真模型如图 3-5 所示,模型中各模块的主要参数设置见表 3-1。

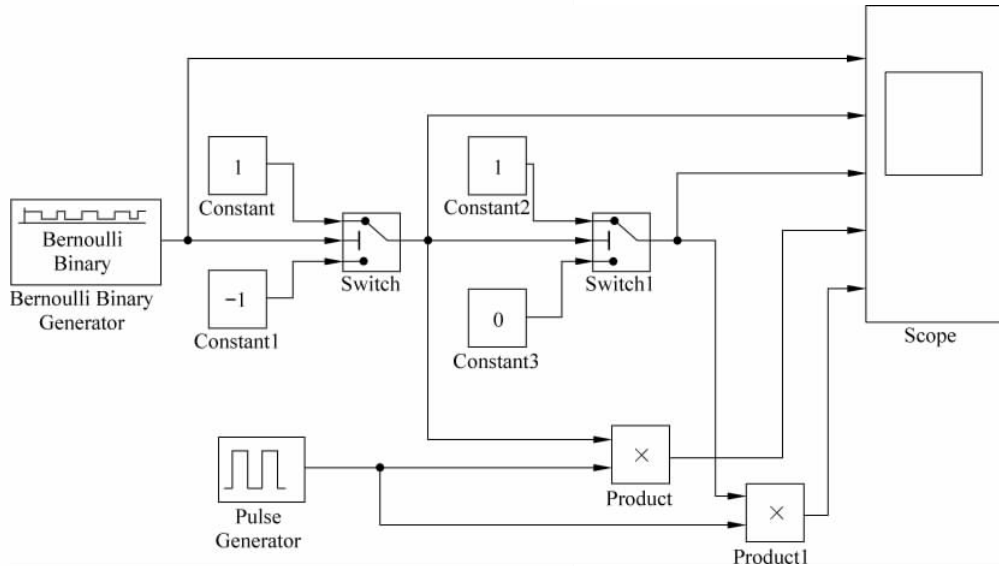


图 3-5 数字基带信号波形的 Simulink 仿真模型

表 3-1 数字基带信号波形的 Simulink 仿真参数

模 块	参 数 名 称	取 值
Bernoulli Binary Generator	Probability of a zero	0.5
	Initial seed	61
	Sample time	1s
Switch	Criteria for passing first in put	$u_2 \geq \text{Threshold}$
	Threshold	0.5
Switch 1	Criteria for passing first in put	$u_2 \geq \text{Threshold}$
	Threshold	0
Pulse Generator	period	1s
	Pulse Width	50(% of period)

Simulink 仿真结果如图 3-6 所示,示波器从上到下依次是:原始信号、双极性不归零信号、单极性不归零信号、双极性归零信号和单极性归零信号。

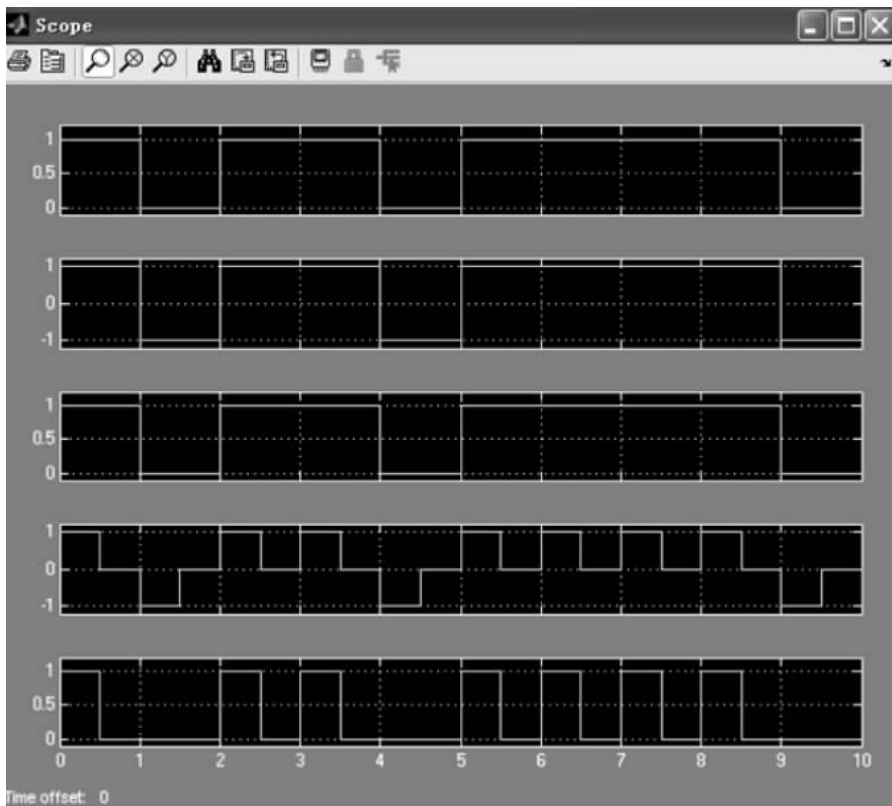


图 3-6 数字基带信号波形的 Simulink 仿真结果

### 3.2 数字基带信号码型仿真

在基带信号传输时,由于不同传输媒介具有不同的传输特性,需要使用不同的接口线路码型(传输码)。为了匹配于基带信道的传输特性,并考虑到接收端提取时钟方便,传输码应具备以下的特性:

- (1) 不含直流分量,且低频分量较少。
- (2) 含有丰富的定时信息,便于从接收码流中提取位同步时钟信号。
- (3) 功率谱主瓣宽度窄,以节省传输频带。
- (4) 不受信息源统计特性的影响。
- (5) 具有内在的监错能力。
- (6) 编译码简单,以降低通信延时和成本。

满足或部分满足以上特性的传输码型种类很多,下面将介绍目前常用的几种传输码的仿真。

### 3.2.1 数字双相码

双相码,又名曼彻斯特(Manchester)码,其编码规则是:将信息代码0编码为线路码“01”;将信息代码1编码为线路码“10”。双相码常用于局域网传输,每一位的中间的跳变既作时钟信号,又作数据信号。

#### (1) 数字双相码的 MATLAB 仿真

下面通过 MATLAB 程序来仿真一串随机消息代码的基带信号波形,首先产生 1000 个随机信号序列,用双相码的编码规则进行编码,并且求平均功率谱密度。其中编码部分的 MATLAB 仿真流程如图 3-7 所示。

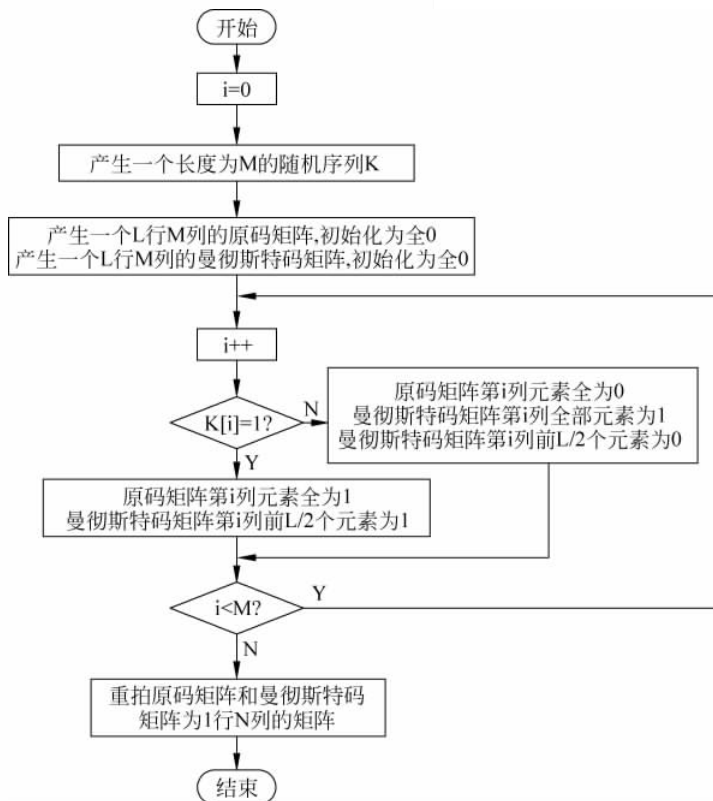


图 3-7 数字双相码的仿真流程图

源代码如下:

```

close all
clear all
% 采样点数的设置
k = 14;
% 每码元采样数的设置
L = 128;
  
```



```

N = 2 ^ k;
M = N/L; % M 为码元个数
dt = 1/L; % 时域采样间隔
T = N * dt; % 时域截断区间
df = 1.0/T; % 频域采样间隔
Bs = N * df/2; % 频域截断区间
t = linspace( -T/2, T/2, N); % 产生时域采样点
f = linspace( -Bs, Bs, N); % 产生频域采样点
EP1 = zeros(size(f));
EP2 = zeros(size(f));
EP3 = zeros(size(f));
for x = 1:1000 % 取样 1000 次
    K = round(rand(1, M)); % 产生一个长度为 M 的随机序列 K, 0 和 1 等概出现
    original = zeros(L, M); % 产生一个 L 行 M 列的原码矩阵, 初始化为全 0 矩阵
    Manchester = zeros(L, M); % 产生一个 L 行 M 列的曼彻斯特矩阵, 初始化为全 0 矩阵
    for i = 1:M
        if K(i) == 1
            original(:, i) = 1; % 原码
            Manchester(1:L/2, i) = 1; % 使 manchester 矩阵第 i 列前 L/2 个元素为 1
        else
            original(:, i) = 0; % 原码
            Manchester(:, i) = 1; % 使 manchester 矩阵第 i 列为 1
            Manchester(1:L/2, i) = 0; % 使 manchester 矩阵第 i 列前 L/2 个元素为 0
        end
    end
end
%% ----- 分别重排 nrz、manchester 矩阵为 1 行 N 列的矩阵 ----- %%
original = reshape(original, 1, N);
Manchester = reshape(Manchester, 1, N);
% 做傅里叶变换并算出功率谱密度
ORIGINAL = t2f(original, dt);
P1 = ORIGINAL. * conj(ORIGINAL)/T;
MANCHESTER = t2f(Manchester, dt);
P2 = MANCHESTER. * conj(MANCHESTER)/T;
% 求功率谱密度的均值
EP1 = (EP1 * (x - 1) + P1)/x;
EP2 = (EP2 * (x - 1) + P2)/x;
end
figure(1) % 开启一个编号为 1 的绘图窗口
subplot(2, 2, 1);
plot(t, original); % 画原码的时域图
axis([-5, 5, min(original) - 0.1, max(original) + 0.1]);
title('原码', 'fontsize', 12);
xlabel('t(ms)', 'fontsize', 12);
ylabel('original(t)', 'fontsize', 12);
grid on
subplot(2, 2, 2);
plot(t, Manchester); % 画数字双相码的时域图
axis([-5, 5, min(Manchester) - 0.1, max(Manchester) + 0.1]);

```

```

title('数字双向码','fontsize',12);
xlabel('t(ms)','fontsize',12);
ylabel('Manchester (t)','fontsize',12);
grid on
subplot(2,2,3);plot(f,EP1);%画原码的功率谱密度图
axis([-5,5,0,0.3]);
title('原码功率谱密度图','fontsize',12);
xlabel('f(kHz)','fontsize',12);
ylabel('P1(f)','fontsize',12);
grid on
subplot(2,2,4);plot(f,EP2);%画数字双向码的功率谱密度图
axis([-5,5,0,0.15]);
title('数字双向码功率谱密度图','fontsize',12);
xlabel('f(kHz)','fontsize',12);
ylabel('P2(f)','fontsize',12);
grid on

```

数字双向码的仿真结果如图 3-8 所示。

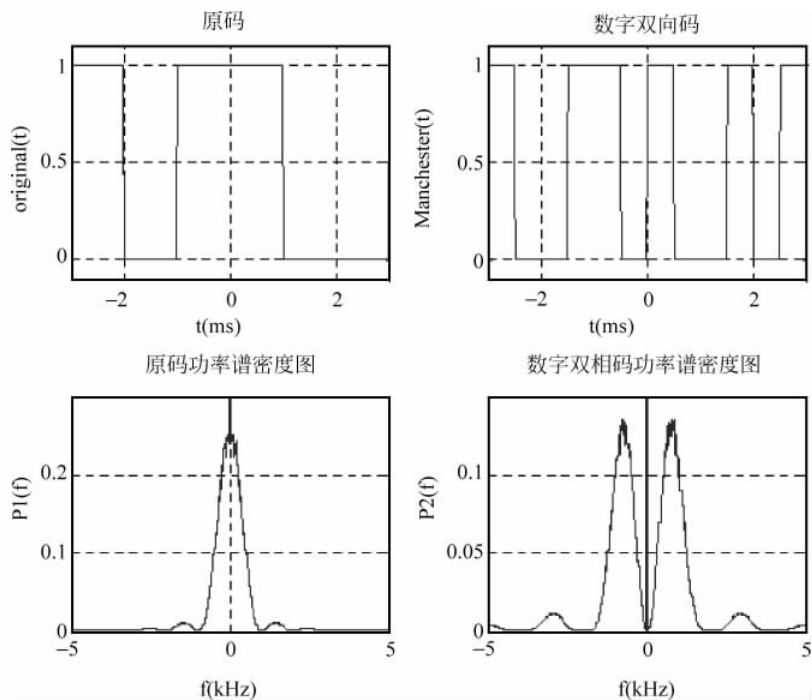


图 3-8 数字双向码的 MATLAB 仿真结果

## (2) 数字双向码的 Simulink 仿真

用 Simulink 实现对数字双向码的仿真,仿真模型如图 3-9 所示,模型中各模块的主要参数设置见表 3-2。

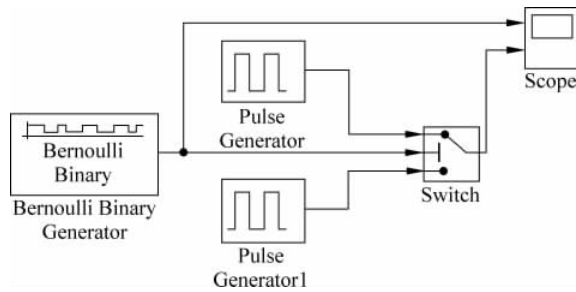


图 3-9 数字双相码的 Simulink 仿真模型

表 3-2 数字双相码的 Simulink 仿真参数

模 块	参 数 名 称	取 值
Bernoulli Binary Generator	Probability of a zero	0.5
	Initial seed	61
	Sample time	1s
Switch	Criteria for passing first in put	$u2 \geq \text{Threshold}$
	Threshold	0.5
Pulse Generator	period	1s
	Pulse Width	50(% of period)
	Phase delay	0
Pulse Generator1	period	1s
	Pulse Width	50(% of period)
	Phase delay	0.5

Simulink 仿真结果如图 3-10 所示,示波器输出从上到下依次为原始信号波形、数字双相码波形。



图 3-10 数字双相码的 Simulink 仿真结果

### 3.2.2 三阶高密度双极性码

三阶高密度双极性码(HDB<sub>3</sub>码),其编码规则是:当信息代码中连“0”个数不大于3时,“1”码用正负脉冲交替表示;当连“0”个数大于3时,将每4个连“0”串的第4个“0”编码为与前一非“0”码同极性的正脉冲或负脉冲,该脉冲为破坏码或V码,为保证加V码后输出仍无直流分量,则需要:①相邻V码的极性必须相反,为此当相邻V码间有偶数个“1”时,将后面的连“0”串中第1个“0”编码为B符号,B符号的极性与前一非“0”码的极性相反,而B符号后面的V码与B符号的极性相同。②V码后面的非“0”符号的极性再交替反转。HDB<sub>3</sub>码是CCITT推荐作为PCM语音系统四次群线路接口码型,在光缆传输系统中采用。图3-11是原码与HDB<sub>3</sub>码的波形图。

原码: 1 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 1 1

HDB<sub>3</sub>码: +1 0 0 0 +V -1 0 0 0 -V +1 -1 +B 0 0 +V 0 -1 +1

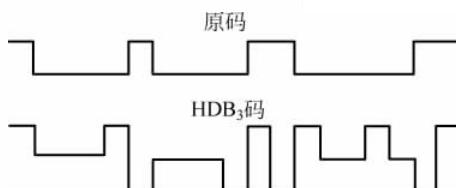


图 3-11 HDB<sub>3</sub>码的波形

HDB<sub>3</sub>码虽然编码很复杂,但解码规则很简单,若3连“0”前后非零脉冲同极性,则将最后一个非零元素译为零,如+1000+1就应该译成“10000”;若2连“0”前后非零脉冲极性相同,则两零前后都译为零,如-100-1,就应该译为0000。再将所有的-1变换成+1后,就可以得到原消息代码。

数字双相码 MATLAB 仿真的源程序如下:

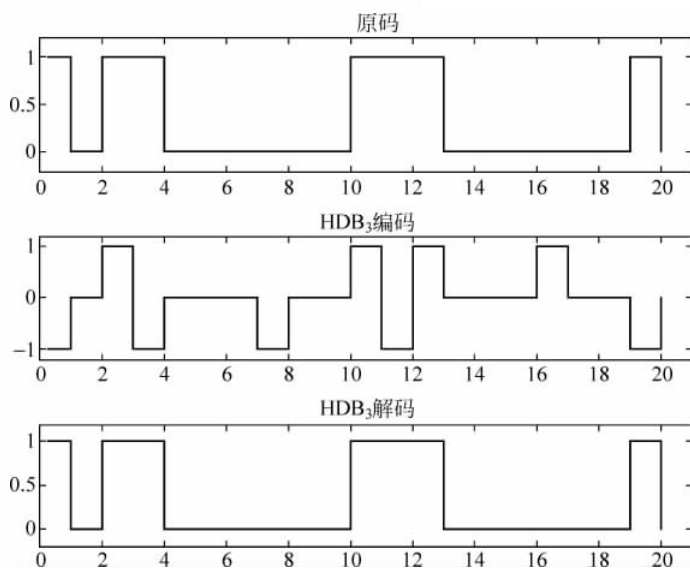
```
x=[1 0 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 0];           % 输入原码
y=x;                                                       % 输出 y 初始化
num=0;                                                     % 计数器初始化
for k=1:length(x)
    if x(k)==1
        num=num+1;                                         % "1"计数器
        if num/2==fix(num/2)                               % 奇数个 1 时输出 -1, 进行极性交替
            y(k)=1;
        else
            y(k)=-1;
        end
    end
end
end
% HDB3 编码
```

```

num = 0; % 连零计数器初始化
yh = y; % 输出初始化
sign = 0; % 极性标志初始化为0
V = zeros(1, length(y)); % 记录V脉冲位置的变量
B = zeros(1, length(y)); % 记录B脉冲位置的变量
for k = 1:length(y)
    if y(k) == 0
        num = num + 1; % 连"0"个数计数
        if num == 4 % 如果4连"0"
            num = 0; % 计数器清零
            yh(k) = 1 * yh(k - 4);
            % 让4连0的最后一个0变为与前一个非零符号相同极性的符号
            V(k) = yh(k); % V脉冲位置记录
            if yh(k) == sign % 如果当前V符号与前一个V符号的极性相同
                yh(k) = -1 * yh(k); % 则让当前V符号极性反转
                yh(k - 3) = yh(k); % 添加B符号,与V符号同极性
                B(k - 3) = yh(k); % B脉冲位置记录
                V(k) = yh(k); % V脉冲位置记录
                yh(k + 1:length(y)) = -1 * yh(k + 1:length(y)); % V后面的非零符号从V开始再交替
            end
            sign = yh(k); % 记录前一个V符号的极性
        end
    else
        num = 0; % 当前输入为"1"则连"0"计数器清零
    end
end % 编码完成
% HDB3 解码
input = yh; % HDB3 码输入
decode = input; % 输出初始化
sign = 0; % 极性标志初始化
for k = 1:length(yh)
    if input(k) ~ = 0
        if sign == yh(k) % 如果当前码与前一个非零码的极性相同
            decode(k - 3:k) = [0 0 0 0]; % 则该码判为V码并将*00V清零
        end
        sign = input(k); % 极性标志
    end
end
decode = abs(decode); % 整流
subplot(3,1,1);stairs([0:length(x) - 1], x);axis([0 length(x) - 0.2 1.2]);
title('原码');
subplot(3,1,2);stairs([0:length(x) - 1], yh);axis([0 length(x) - 1.2 1.2]);
title('HDB3 编码');
subplot(3,1,3);stairs([0:length(x) - 1], decode);axis([0 length(x) - 0.2 1.2]);
title('HDB3 解码');

```

HDB<sub>3</sub> 码的 MATLAB 仿真结果如图 3-12 所示。

图 3-12 HDB<sub>3</sub>码的 MATLAB 仿真结果

### 3.3 发送滤波器仿真

在数字通信系统中,基带信号进入调制器前,波形是矩形脉冲,突变的上升沿和下降沿包含高频成分较丰富,信号的频谱一般比较宽,通过带限信道时,单个符号的脉冲将延伸到相邻符号的码元内,产生码间串扰。因此在信道带宽有限的条件下,要降低误码率,需在信号传递前,通过发送滤波器(脉冲成形滤波器)对其进行脉冲成形处理,改善其频谱特性,产生适合信道传输的波形。数字系统中常用的波形成形滤波器有升余弦脉冲滤波器、平方根升余弦滤波器、高斯滤波器等。下面分别讨论这3种滤波器的特性及仿真。

#### 3.3.1 升余弦脉冲滤波器

余弦脉冲滤波器即系统函数具有余弦波的变化特点,如图 3-13 所示。余弦滚降系统的传输特性可用下式表示:

$$H(\omega) = \begin{cases} 1, & 0 \leq |\omega| \leq \frac{(1-\alpha)\pi}{T_s} \\ 1 + \cos \frac{T_s}{2\alpha} \left( \omega - \frac{(1-\alpha)\pi}{T_s} \right), & \frac{(1-\alpha)\pi}{T_s} \leq |\omega| \leq \frac{(1+\alpha)\pi}{T_s} \\ 0, & |\omega| \geq \frac{(1+\alpha)\pi}{T_s} \end{cases} \quad (3-1)$$

其中, $\alpha$ 为滚降系数, $T_s$ 为码元间隔。由图 3-13 中可以看出,滚降特性所形成的波形

$h(t)$ 除抽样点  $t=0$  处不为零外,其余抽样点上均为零,并且“拖尾”现象随着  $\alpha$  的增大而振荡幅度减小、衰减速度加快。

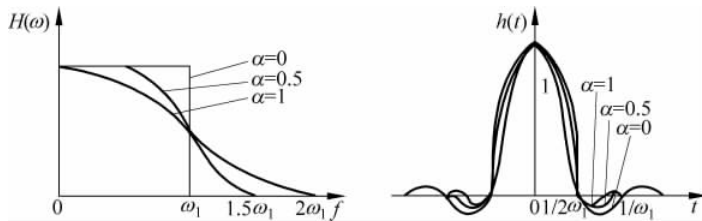


图 3-13 余弦滚降系统特性

以下程序实现对升余弦滚降滤波器的仿真,升余弦滚降滤波器的频谱和时域波形仿真结果分别如图 3-14 和图 3-15 所示。

```
% 升余弦滚降系统
clear all;
close all;
Ts = 1;
N_sample = 17;
dt = Ts/N_sample;
df = 1.0/(20.0 * Ts);
t = -10 * Ts:dt:10 * Ts;
f = -2/Ts:df:2/Ts;
alpha = [0, 0.5, 1];
for n = 1:length(alpha)
    for k = 1:length(f)
        if abs(f(k)) > 0.5 * (1 + alpha(n))/Ts
            Xf(n, k) = 0;
        elseif abs(f(k)) < 0.5 * (1 - alpha(n))/Ts
            Xf(n, k) = Ts;
        else
            Xf(n, k) = 0.5 * Ts * (1 + cos(pi * Ts / (alpha(n) + eps) * (abs(f(k)) - 0.5 * (1 - alpha(n))/Ts)));
        end
    end
    xt(n, :) = sinc(t/Ts) .* (cos(alpha(n) * pi * t/Ts)) ./ (1 - 4 * alpha(n)^2 * t.^2 / Ts^2 + eps);
end
figure(1)
plot(f, Xf(1, :), 'b', f, Xf(2, :), 'r', f, Xf(3, :), 'k');
axis([-1 1 0 1.2]); xlabel('f/Ts'); ylabel('升余弦滚降频谱');
legend('alpha = 0', 'alpha = 0.5', 'alpha = 1');
figure(2)
plot(t, xt(1, :), 'b', t, xt(2, :), 'r', t, xt(3, :), 'k');
legend('alpha = 0', 'alpha = 0.5', 'alpha = 1');
axis([-10 10 -0.5 1.1]); xlabel('t'); ylabel('升余弦滚降波形');
```

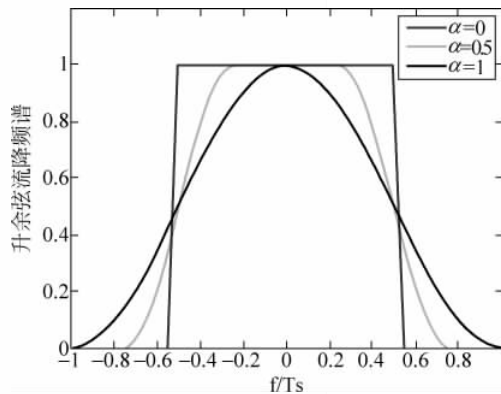


图 3-14 升余弦滚降系统频谱

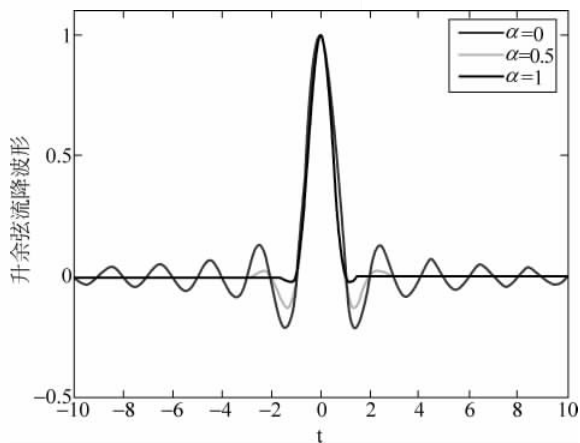


图 3-15 升余弦滚降系统时域波形

下面来仿真升余弦滤波器对信号的影响。采用 MATLAB 工具箱中专门用于升余弦 FIR 滤波器的指令  $[\text{NUM}, \text{DEN}] = \text{RCOSINE}(\text{Fd}, \text{Fs}, \text{TYPE\_FLAG}, \text{R})$ , 可以返回一个具有升余弦过渡带的低通线性相位 FIR 滤波器, 截止频率是  $\text{Fd}$ , 滚降系数为  $\text{R}$ , 采样频率为  $\text{Fs}$ ,  $\text{TYPE\_FLAG}$  规定设计的是规范的升余弦滚降滤波器 (normal), 还是平方根升余弦滤波器 (sqrt), 用整数型参数  $\text{DELAY}$  设定延时。

```
% 设置参量, 采用 4 倍采样速率, 滚降系数为 0.5
Fd = 1; Fs = 4; Delay = 2; R = 0.5;
% ----- 建立升余弦滚降滤波器 ----- %
[yf, tf] = rcosine(Fd, Fs, 'fir/normal', R, Delay);
% 画图得到升余弦滚降滤波器波形
% b1 = ones(1, length(tf)); % 滤波器输入矩形脉冲
figure(1);
subplot(3,1,1);
plot(yf);
grid;
xlabel('Time');
```



```

ylabel('Amplitude');
title('升余弦滚降滤波器 h(t)');
% ----- 定义一个与二元序列对应的时间序列作为原始信号 ----- %
x = [zeros(1,10),ones(1,10),ones(1,10),zeros(1,10),zeros(1,10),zeros(1,10)];
y = filter(yf,tf,x)/Fs;
% 画出原始信号波形
subplot(3,1,2);
plot(x);
axis([0,61,-0.2,1.2]);
title('原始信号');
% ----- 原始信号通过升余弦滚降滤波器后的输出 ----- %
subplot(3,1,3);
plot(y);
axis([2,61,-0.2,1.2]);
title('滤波后输出')
grid;

```

由图 3-16 可见,原始信号通过该升余弦滚降滤波器可以使波形平滑,有效地改变突变的上升沿和下降沿,从而消除波形中的高频成分,达到降低码间串扰的可能性、提高频带利用率的效果。由于滤波器的影响,原始信号和滤波后信号之间存在一定的延迟。

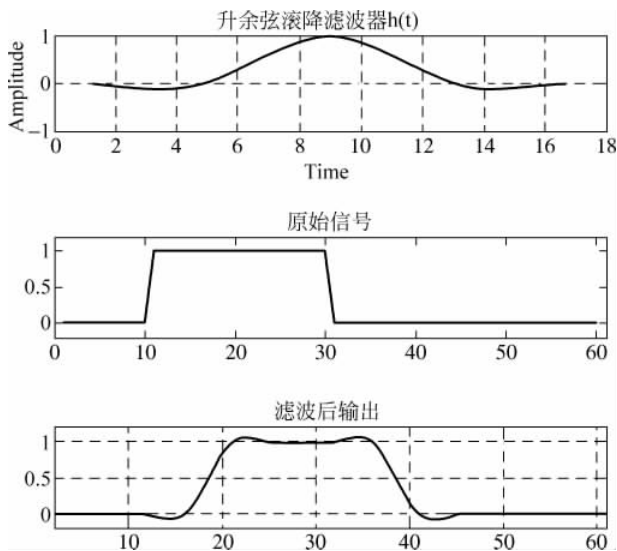


图 3-16 余弦滚降滤波器对信号的影响

### 3.3.2 平方根升余弦滤波器

可以将波形成形滤波器放置在收发两端,即在发送端和接收端分别用一个平方根升余弦滤波器(平方根升余弦函数),并且使两个滤波器满足匹配滤波原则。则既实现升余弦滤波器的作用,也满足匹配滤波器的实现,从而提升接收端信噪比,更便于准确接收信号。如果不考虑由于信道引起的码间串扰,两个平方根升余弦函数相乘(相当于时域卷积)就得到升余弦形式的合成的系统传输函数。平方根升余弦滤波器可以表示为

$$H_T(\omega) = H_R(\omega) = H(\omega) = \begin{cases} 1, & 0 \leq |\omega| \leq \frac{(1-\alpha)\pi}{T_s} \\ \sqrt{1 + \cos \frac{T_s}{2\alpha} \left( \omega - \frac{(1-\alpha)\pi}{T_s} \right)}, & \frac{(1-\alpha)\pi}{T_s} \leq |\omega| \leq \frac{(1+\alpha)\pi}{T_s} \\ 0, & |\omega| \geq \frac{(1+\alpha)\pi}{T_s} \end{cases} \quad (3-2)$$

平方根升余弦冲激响应的表达式为

$$h(t) = \frac{\sqrt{T}}{\pi t(T^2 - 16t^2T^2)} \{ T^2 \sin[2\pi f_0(1-\alpha)t] + 4atT \cos[2\pi f_0(1+\alpha)t] \} \quad (3-3)$$

采用 MATLAB 工具箱中专门用于升余弦 FIR 滤波器的指令 [NUM, DEN] = RCOSINE(Fd, Fs, TYPE\_FLAG, R), 可以返回一个具有平方根升余弦过渡带的低通线性相位 FIR 滤波器, 截止频率是 Fd, 滚降系数为 R, 采样频率为 Fs, TYPE\_FLAG 用来规定滤波器类型, 这里选择 'sqrt', 整数参数 DELAY 设定延时。

```
% 设置参量, 采用 4 倍采样速率, 滚降系数为 0.5
Fd = 1; Fs = 4; Delay = 2; R = 0.5;
% ----- 建立升余弦滚降滤波器 ----- %
[yf, tf] = rcosine(Fd, Fs, 'sqrt', R, Delay);
% 画图得到升余弦滚降滤波器波形
% b1 = ones(1, length(tf)); % 滤波器输入矩形脉冲
figure(1);
subplot(3,1,1);
plot(yf);
grid;
xlabel('Time');
ylabel('Amplitude');
title('平方根升余弦滚降滤波器 h(t)');
% ----- 定义原始信号 ----- %
x = [zeros(1,10), ones(1,10), ones(1,10), zeros(1,10), zeros(1,10), zeros(1,10)];
y1 = filter(yf, tf, x)/(Fs^0.5);
% y2 = filter(yf, tf, y1)/(Fs^0.5);
% 画出原始信号波形
subplot(3,1,2);
plot(x);
axis([0,61, -0.2,1.2]);
title('原始信号');
% ----- 原始信号通过升余弦滚降滤波器后的输出 ----- %
subplot(3,1,3);
plot(y1);
axis([2,61, -0.2,1.2]);
title('滤波后输出');
grid;
```

由图 3-17 可见, 原始信号通过该平方根升余弦滚降滤波器后也可以使波形平滑, 有效地改变突变的上升沿和下降沿, 作用与升余弦滤波器类似。实际应用中, 收、发两端的平方根升余弦滚降滤波器可以按照匹配滤波器的原则进行设计。

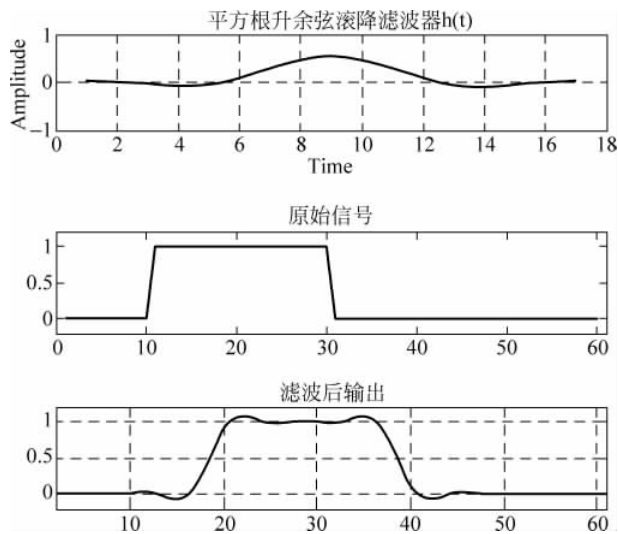


图 3-17 平方根余弦滚降滤波器对信号的影响

### 3.3.3 高斯滤波器

在一些通信场合(如移动通信),对信号带外辐射功率的限制十分严格,例如要求衰减达到 70~80dB 以上,从而减小对邻道的干扰,这时可以采用高斯低通滤波器。高斯低通滤波器的特点是:

- (1) 带宽窄,具有良好的截止特性。
- (2) 具有较低的过冲脉冲响应,以防止调制器的瞬间频偏过大。
- (3) 保持滤波器输出脉冲的面积不变,以便于进行相干解调。

高斯型滤波器的传输函数  $H(f)$  为

$$H(f) = \exp(-a^2 f^2) \quad (3-4)$$

高斯型滤波器的冲激响应为

$$h(t) = \frac{\sqrt{\pi}}{a} \exp\left(-\frac{\pi^2}{a^2} t^2\right) \quad (3-5)$$

下面通过仿真来说明高斯脉冲成形滤波器对矩形脉冲输入的影响。假设  $b(t)$  是高度为 1、宽度为  $T_b$  的矩形脉冲,则  $b(t)$  通过高斯脉冲成形滤波器的输出波形  $g(t)$  为

$$\begin{aligned} g(t) &= h(t) * b(t) \\ &= \int_{t-\frac{T_b}{2}}^{t+\frac{T_b}{2}} \frac{\sqrt{\pi}}{a} \exp\left(-\frac{\pi^2}{a^2} \tau^2\right) d\tau \\ &= \frac{1}{2} \left\{ \operatorname{erfc}\left[\frac{\pi}{a} \left(t - \frac{T_b}{2}\right)\right] - \operatorname{erfc}\left[\frac{\pi}{a} \left(t + \frac{T_b}{2}\right)\right] \right\} \end{aligned} \quad (3-6)$$

源代码如下:

```
t1 = -1.5:0.01:1.5;
t2 = -1.5:0.01:1.5;
```

```

b1 = [zeros(1,50),ones(1,200),zeros(1,51)];
%% ----- 产生滤波器 1, a = 0.25 ----- %%
y1 = sqrt(pi)/0.25 * exp(-((pi * t1).^2)/0.25.^2);
z1 = 0.5 * (erfc(pi/0.25 * (t1 - 1)) - erfc(pi/0.25 * (t1 + 1)));
%% ----- 产生滤波器 2, a = 0.5 ----- %%
y2 = sqrt(pi)/0.5 * exp(-((pi * t1).^2)/0.5.^2);
z2 = 0.5 * (erfc(pi/0.5 * (t1 - 1)) - erfc(pi/0.5 * (t1 + 1)));
%% ----- 产生滤波器 3, a = 1 ----- %%
y3 = sqrt(pi) * exp(- (pi * t1).^2);
z3 = 0.5 * (erfc(pi * (t1 - 1)) - erfc(pi * (t1 + 1)));
%% ----- 产生滤波器 4, a = 2 ----- %%
y4 = sqrt(pi)/2 * exp(- ((pi * t1).^2)/2.^2);
z4 = 0.5 * (erfc(pi/2 * (t1 - 1)) - erfc(pi/2 * (t1 + 1)));
subplot(3,1,1),plot(t2,b1);axis([-1.5 1.5 0 1.2]);xlabel('t/T');ylabel('b(t)');
title('高斯脉冲成形滤波器的输入(矩形脉冲)');
subplot(3,1,2),plot(t1,y1,'r',t1,y2,'g',t1,y3,'b',t1,y4,'m');
legend('\alpha = 0.25', '\alpha = 0.5', '\alpha = 1', '\alpha = 2');
xlabel('t/T');ylabel('h(t)');title('高斯脉冲成形滤波器的冲激响应 h(t)');
subplot(3,1,3),plot(t1,z1,'r',t1,z2,'g',t1,z3,'b',t1,z4,'m');
xlabel('t/T');
ylabel('g(t)');
legend('\alpha = 0.25', '\alpha = 0.5', '\alpha = 1', '\alpha = 2');
axis([-1.5 1.5 0 1.2]);
title('高斯脉冲成形滤波器的输出');

```

由图 3-18 可以看出,矩形脉冲通过高斯脉冲成形滤波器后变成了高斯脉冲,有效地改变矩形波突变的上升沿和下降沿。

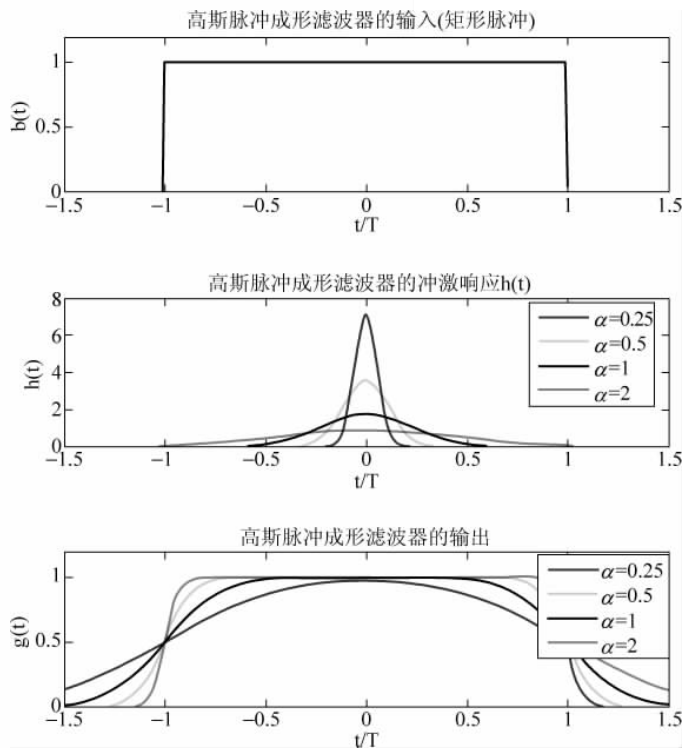


图 3-18 高斯脉冲成形滤波器对矩形脉冲输入信号的影响