

文档说明

项目地址: <https://github.com/LvGitHub-9/SpreadSpectrumCommunication>

名称: 直接序列扩频

说明: 直扩系统的扩频解扩, 差分编码以及检测方法研究

版本: **v1.2**

作者: 小吕同学修改记录:

	版本号	日期	作者	说明
•	v1.0	2025-1-1	小吕同学	首次发布
•	v1.1	2025-1-3	小吕同学	修改脉冲成型部分
•	v1.2	2025-1-6	小吕同学	修改脉冲成型部分, 增加码片时间、信号频谱

变化

FindMe: https://space.bilibili.com/10179894?spm_id_from=333.1007.0.0

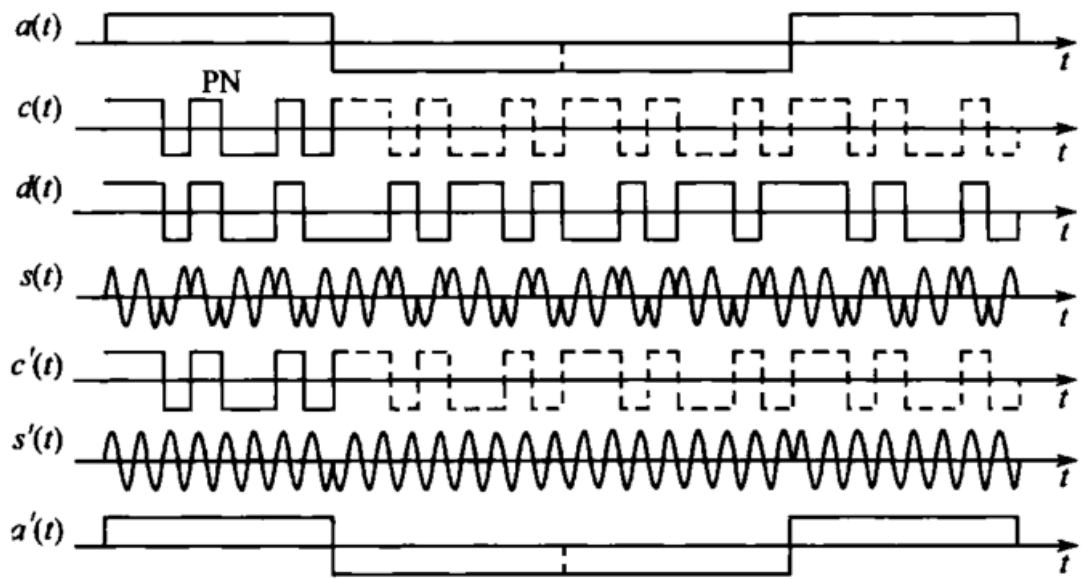
Copyright **2024** Lv. All Rights Reserved.

Distributed under MIT license.

See file LICENSE **for** detail or copy at <https://opensource.org/licenses/MIT>

直接序列扩频系统原理

直接序列扩频（Direct-Sequence Spread Spectrum, DSSS），简称直扩（DS），意思是直接用扩频码调制消息序列，上变频后进行发送。接收端用原扩频码进行相关计算，解出原始消息序列。下图为直扩的波形示意图；



用原始消息与伪随机序列做相与，得到的信号上变频，信道，下变频后用伪随机序列的自相关性解码。

扩频、解扩的实现

根据上述直扩系统的框图，可以写出基本的扩频解扩实现代码

```
%% 程序名称: SimpleDSSS.m
%% 简化直接序列扩频
close all; clear; clc;

%% 消息生成
bits=10; % 消息个数
mes=randi([0,1],1,bits);
bimes=2*mes-1; % 单极性码转双极性,BPSK

%% 扩频码生成(使用m序列)
initial=[1 0 1 1 0 0]; % 6阶
feedback=103;
m=mseq(initial,feedback,0);
L=length(m); % 取m序列长度
m=2*m-1; % 逻辑映射
clear initial feedback

%% 直接扩频
kmes=kron(bimes,m); % 克罗内克积

%% 省略上变频,信道,下变频过程仅研究扩频和解扩过程
% 不加信道
% ymes=kmes;

% 加AWGN信道
SNR=0; % 信噪比
ymes=awgn(kmes,SNR);

%% 相关解扩
en=zeros(1,bits); % 存储解扩序列
ex=[]; % 存储自相关后的序列
w=floor(L/4); % 选取自相关峰值窗口的长度
buf=zeros(1,L); % 存储一个符号长度
conj=zeros(1,2*w); % 在自相关函数中选取峰值窗口
% 用于判断峰值正负

for i=1:bits
    buf=ymes(1,1+(i-1)*L:i*L); % 取出一个符号长度
    cor=xcorr(buf,m); % 做自相关
    conj=cor(1,L-w:L+w); % 选取自相关窗口
    ex=[ex cor]; % 保存自相关函数
    if(max(conj)>abs(min(conj))) % 判断自相关峰值正负,解扩
        en(i)=1;
    else
        en(i)=0;
    end
end

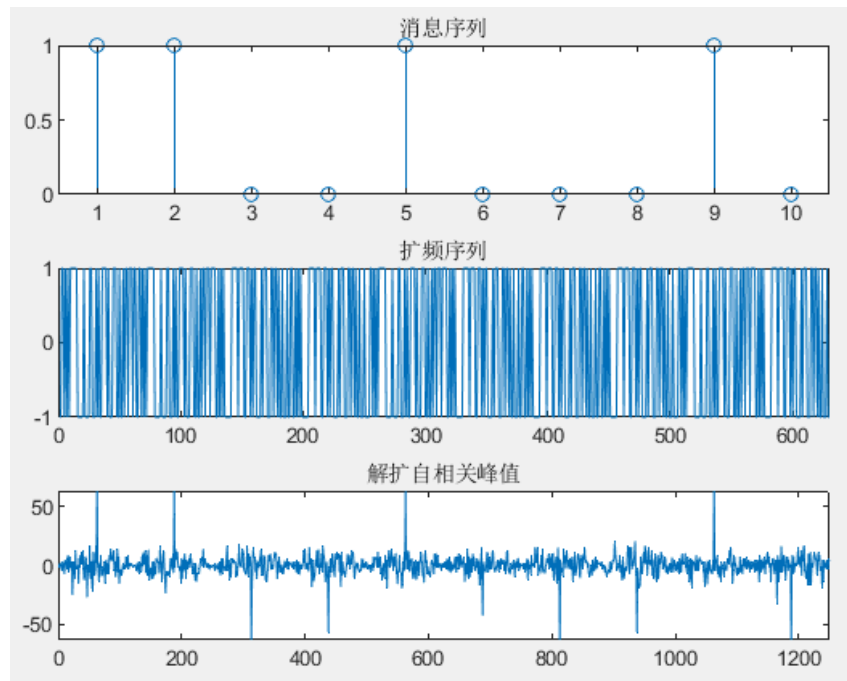
%% 误码率
A=find(en~=mes); % 计算误码率
BER=length(A)/bits;
```

```
disp(['解码误码率: ',num2str(BER)])
```

```
%% 作图
subplot(3,1,1)
stem(mes)
title('消息序列');
axis([0.5 0.5+bits 0 1]);
subplot(3,1,2)
plot(kmes)
axis([0 length(kmes) -1 1]);
title('扩频序列');
subplot(3,1,3)
plot(ex)
title('解扩自相关峰值');
axis([0 length(ex) -63 63]);
```

程序输出：

```
> 解码误码率: 0
```



可以从解扩自相关峰值中很清楚的看出原消息序列的幅度，因此很方便从中解码。

信号频谱的变化

```
%% 文件名称: SpectrumChanges.m
%% 扩频通信频谱变化
close all; clear; clc;

%% 消息生成
bits=10; % 消息个数
mes=randi([0,1],1,bits);
bimes=2*mes-1; % 单极性码转双极性,BPSK

%% 扩频码生成(使用m序列)
initial=[1 0 1 1 0 0]; % 6阶
feedback=103;
m=mseq(initial,feedback,0);
L=length(m); % 取m序列长度
m=2*m-1; % 逻辑映射

%% 扩频
kmes=kron(bimes,m); % 克罗内克积
bmes=rectpulse(bimes,L); % 信号等长

%% 参数
fc=6e3; % 载波频率6kHz
fb=4e3; % 带宽4kHz
fs=48e3; % 采样频率48kHz
ts=1/fs; % 时域采样间隔

%% 省去脉冲成型

%% 原始信号与扩频信号频谱
T=length(kmes)/fs; % 发送时间
t=0:ts:T-ts; % 时域时间点
df=fs/length(t); % 频率间隔
f=-fs/2:df:fs/2-df; % 频域频率点

kmes_fft=fft(kmes)/fs; % 各自求ft
bmes_fft=fft(bmes)/fs;

figure
subplot(2,2,1);
plot(t,bmes);title('bmes');
title('原始信号');
subplot(2,2,3);
plot(f,fftshift(bmes_fft));title('bmes_fft');
title('原始信号频谱');

subplot(2,2,2);
plot(t,kmes);title('kmes');
title('扩频信号');
subplot(2,2,4);
plot(f,fftshift(kmes_fft));title('kmes_fft');
title('扩频信号频谱');
```

```

%% 脉冲成型
% 假设带宽为4-8kHz，基带信号带宽为2kHz，码片长度为1/2kHz=0.5ms
% 信号发送频率为48kHz，0.5ms能够发送0.5ms*48kHz=24个符号
% 即一个码片(chip)长度为24
% 脉冲成型大小影响码片时间，进而影响带宽
rect=24;
rkmes=rectpulse(kmes,rect);
rbmes=rectpulse(bmes,rect);

%% 脉冲成型后原始信号与扩频信号频谱
T=length(rkmes)/fs;           % 发送时间
t=0:ts:T-ts;                  % 时域时间点
df=fs/length(t);              % 频率间隔
f=-fs/2:df:fs/2-df;           % 频域频率点

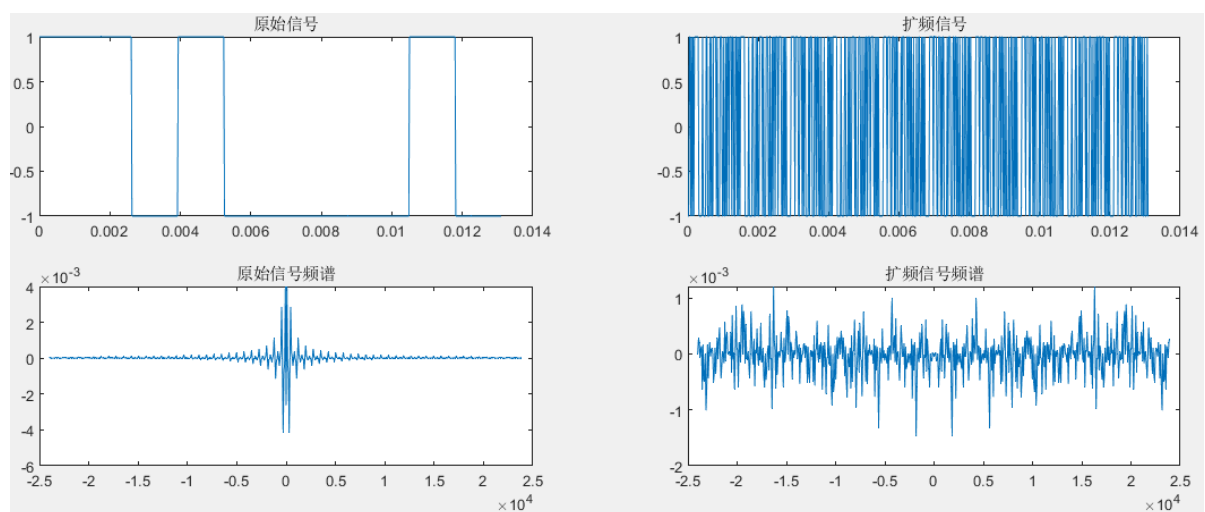
rkmes_fft=fft(rkmes)/fs;       % 各自求ft
rbmes_fft=fft(rbmes)/fs;

figure
subplot(2,2,1);
plot(t,rbmes);title('rbmes');
title('脉冲成型后原始信号');
subplot(2,2,3);
plot(f,fftshift(rbmes_fft));title('rbmes_fft');
title('脉冲成型后原始信号频谱');

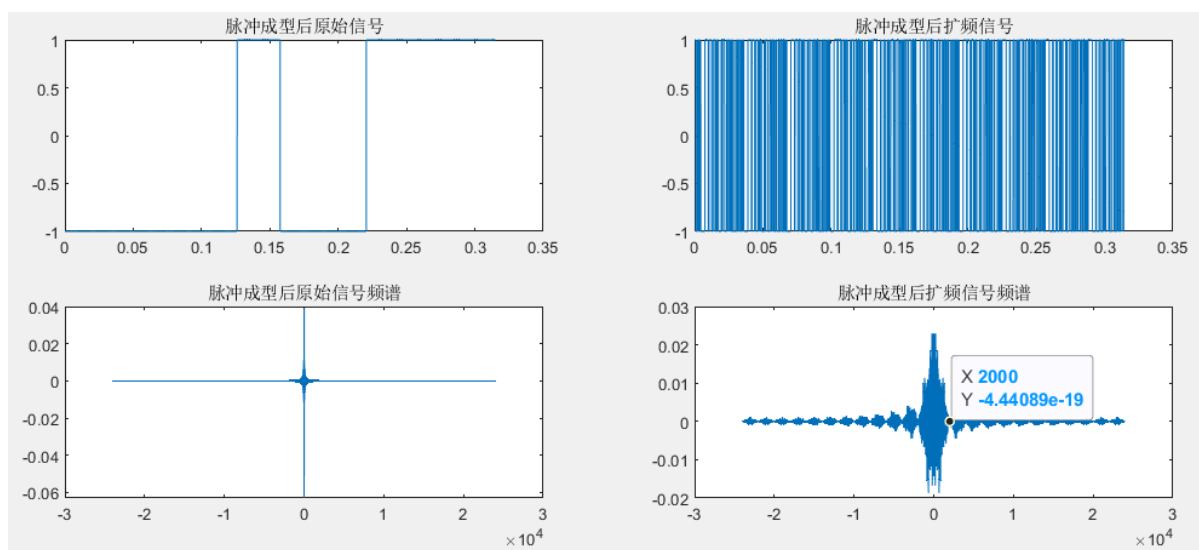
subplot(2,2,2);
plot(t,rkmes);title('rkmes');
title('脉冲成型后扩频信号');
subplot(2,2,4);
plot(f,fftshift(rkmes_fft));title('rkmes_fft');
title('脉冲成型后扩频信号频谱');

```

程序输出：



可以看出，扩频后信号的频谱确实被扩展到很长的一段区间。



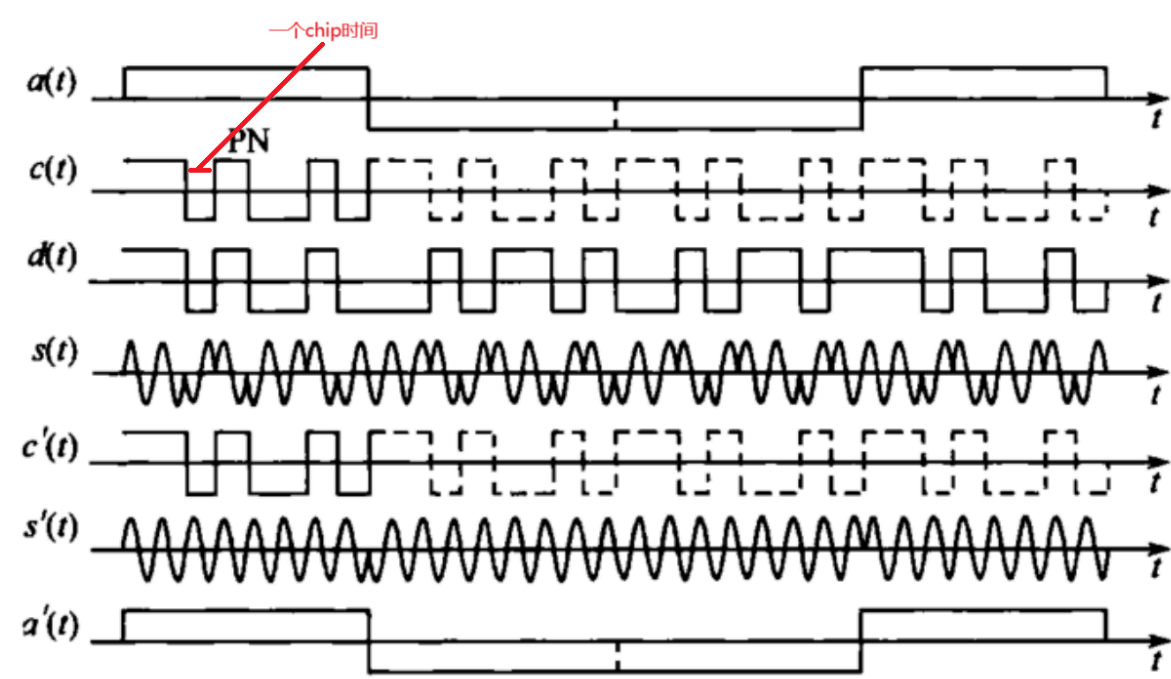
脉冲成型可以将信号的频谱集中到所需要的频带内。

扩频系统中的码片时间

码片时间，也称Chip，指的是扩频中一个伪随机序列里的0或者1的持续时间。假设扩频信号长度为N，则扩频信号中每一个消息bit位都会被扩展成N个码片。

简单来说，码片时间就是伪随机序列门信号的门宽。

百度百科：系统通过扩频把比特转换成码片。常用的扩频形式是用一个伪随机噪声序列（PN序列）与窄带PSK信号相乘。PN序列通常用符号C来表示，一个PN序列是一个有序的由1和0构成的二数码流，其中的1和0由于不承载信息，因此不称为bit而称为chip（码片）。



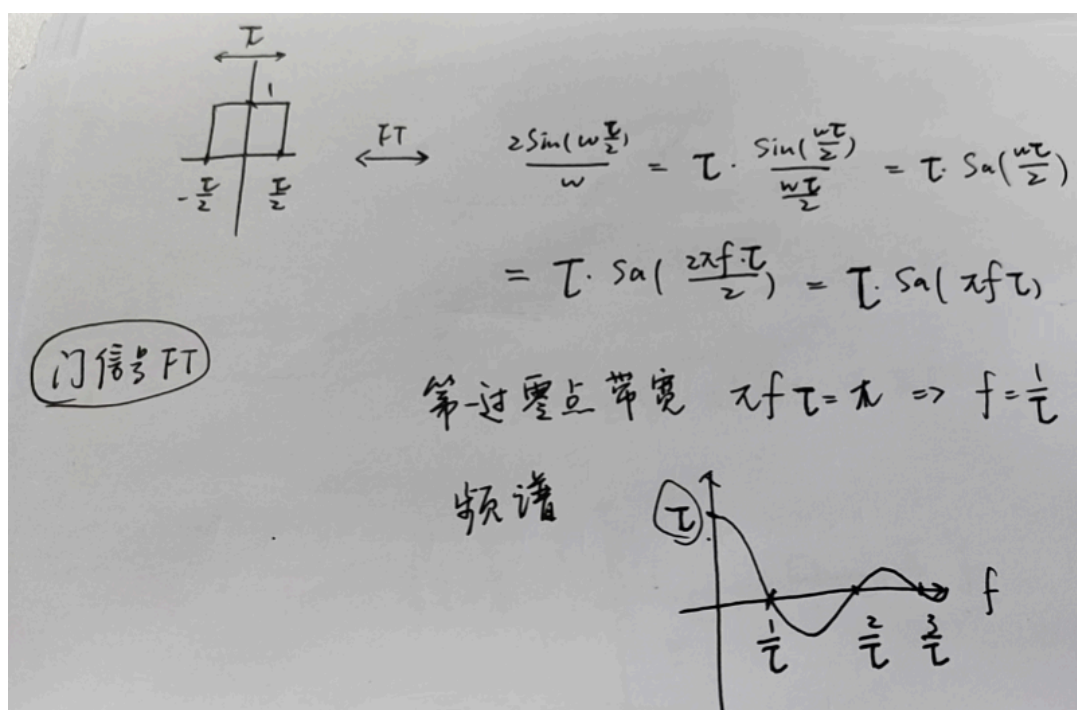
码片时间有什么用？

伪随机序列就类似于门信号卷积周期冲激串，基本门信号的傅里叶变换对，

表 4.2 基本傅里叶变换对

信 号	傅里叶变换	傅里叶级数系数 (若为周期的)
$\sum_{k=-\infty}^{+\infty} a_k e^{jk\omega_0 t}$	$2\pi \sum_{k=-\infty}^{+\infty} a_k \delta(\omega - k\omega_0)$	a_k
$e^{jk\omega_0 t}$	$2\pi \delta(\omega - k\omega_0)$	$a_1 = 1$ $a_k = 0$, 其余 k
$\cos \omega_0 t$	$\pi[\delta(\omega - \omega_0) + \delta(\omega + \omega_0)]$	$a_1 = a_{-1} = \frac{1}{2}$ $a_k = 0$, 其余 k
$\sin \omega_0 t$	$\frac{\pi}{j}[\delta(\omega - \omega_0) - \delta(\omega + \omega_0)]$	$a_1 = -a_{-1} = \frac{1}{2j}$ $a_k = 0$, 其余 k
$x(t) = 1$	$2\pi \delta(\omega)$	$a_0 = 1, a_k = 0, k \neq 0$ (这是对任意 $T > 0$ 选择的傅里叶级数表示)
周期方波 $x(t) = \begin{cases} 1, & t < T_1 \\ 0, & T_1 < t \leq \frac{T}{2} \end{cases}$ 和 $x(t+T) = x(t)$	$\sum_{k=-\infty}^{+\infty} \frac{2 \sin k\omega_0 T_1}{k} \delta(\omega - k\omega_0)$	$\frac{\omega_0 T_1}{\pi} \text{sinc}\left(\frac{k\omega_0 T_1}{\pi}\right) = \frac{\sin k\omega_0 T_1}{k\pi}$
$\sum_{n=-\infty}^{+\infty} \delta(t - nT)$	$\frac{2\pi}{T} \sum_{k=-\infty}^{+\infty} \delta\left(\omega - \frac{2\pi k}{T}\right)$	$a_k = \frac{1}{T}$, 对全部 k
$x(t) \begin{cases} 1, & t < T_1 \\ 0, & t > T_1 \end{cases}$	$\frac{2 \sin \omega T_1}{\omega}$	—
$\frac{\sin Wt}{\pi t}$	$X(j\omega) = \begin{cases} 1, & \omega < W \\ 0, & \omega > W \end{cases}$	—
$\delta(t)$	1	—
$u(t)$	$\frac{1}{j\omega} + \pi \delta(\omega)$	—
$\delta(t - t_0)$	$e^{-j\omega t_0}$	—
$e^{-at} u(t), \text{Re}\{a\} > 0$	$\frac{1}{a + j\omega}$	—
$te^{-at} u(t), \text{Re}\{a\} > 0$	$\frac{1}{(a + j\omega)^2}$	—
$\frac{t^{n-1}}{(n-1)!} e^{-at} u(t), \text{Re}\{a\} > 0$	$\frac{1}{(a + j\omega)^n}$	—

在这里说明，门信号的傅里叶变换带宽是第一过零点带宽（以下称为带宽），通过公式推导可以发现，门信号时域的门宽和带宽成倒数关系。



也可以从Matlab上写程序来查看这一关系。

```

%% 程序名称: chip.m
%% 码片时间 (chip)
close all; clear; clc;

%% 参数设置
Fs = 1e3;          % 采样频率 (Hz)
t = 0:1/Fs:1;      % 时间向量, 总时长为1秒
A = 1;             % 门信号的幅度
width = 0.1;        % 门信号的宽度 (秒)

%% 生成门信号
gateSignal = A * (t >= 0 & t <= width);

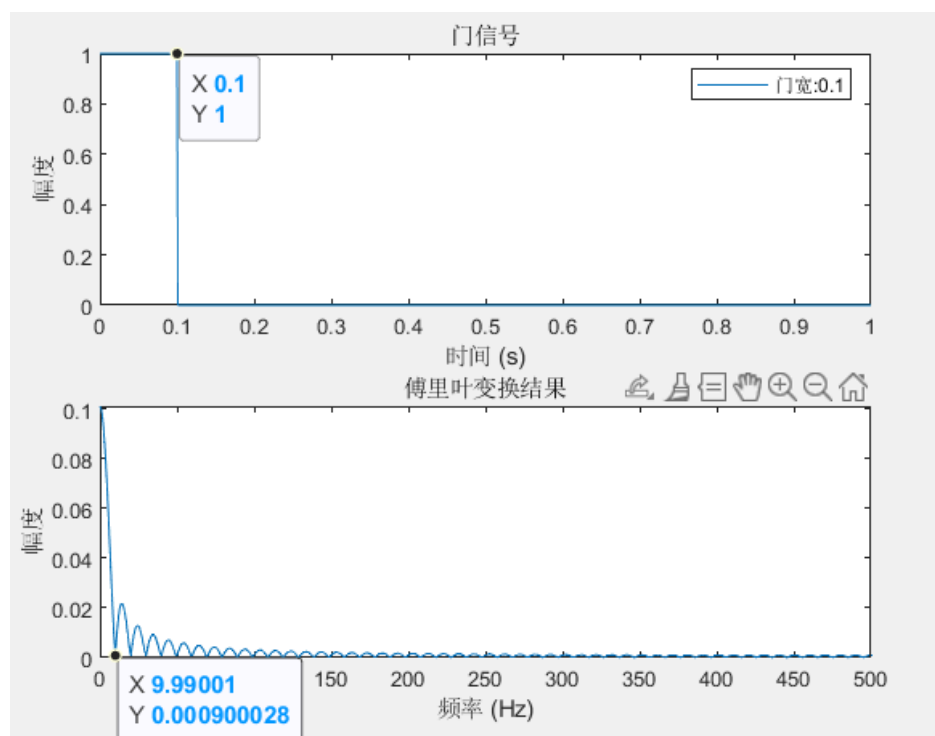
%% 计算傅里叶变换
fftResult = fft(gateSignal)/Fs;

%% 计算频率轴的值
n = length(fftResult);
f = (0:n-1)*(Fs/n);

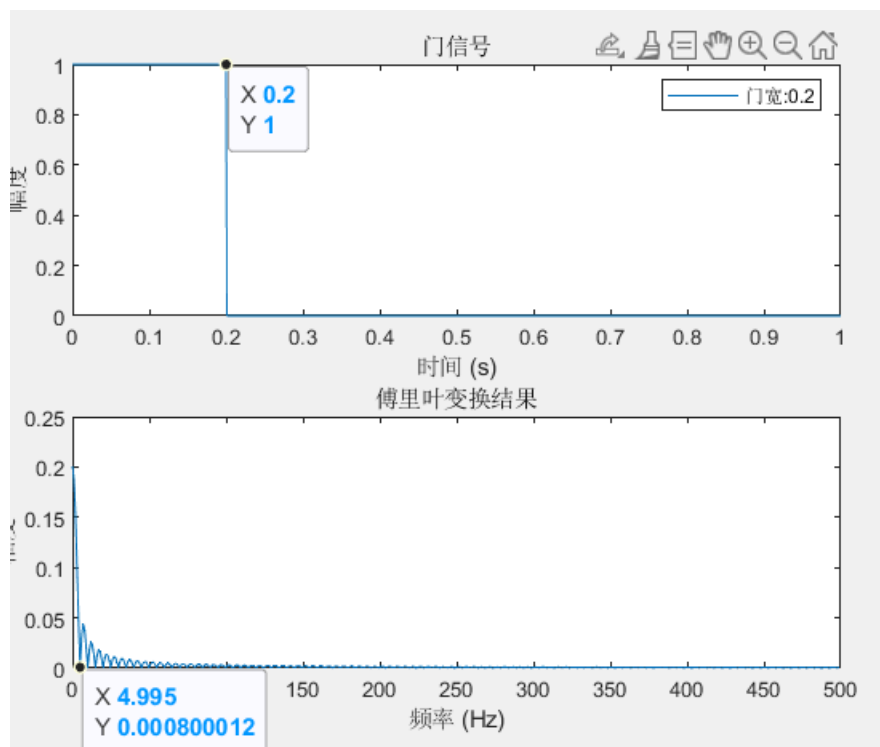
%% 绘制门信号
figure;
subplot(2,1,1);
plot(t, gateSignal);
title('门信号');
xlabel('时间 (s)');
ylabel('幅度');
legend(['门宽:' num2str(width)])

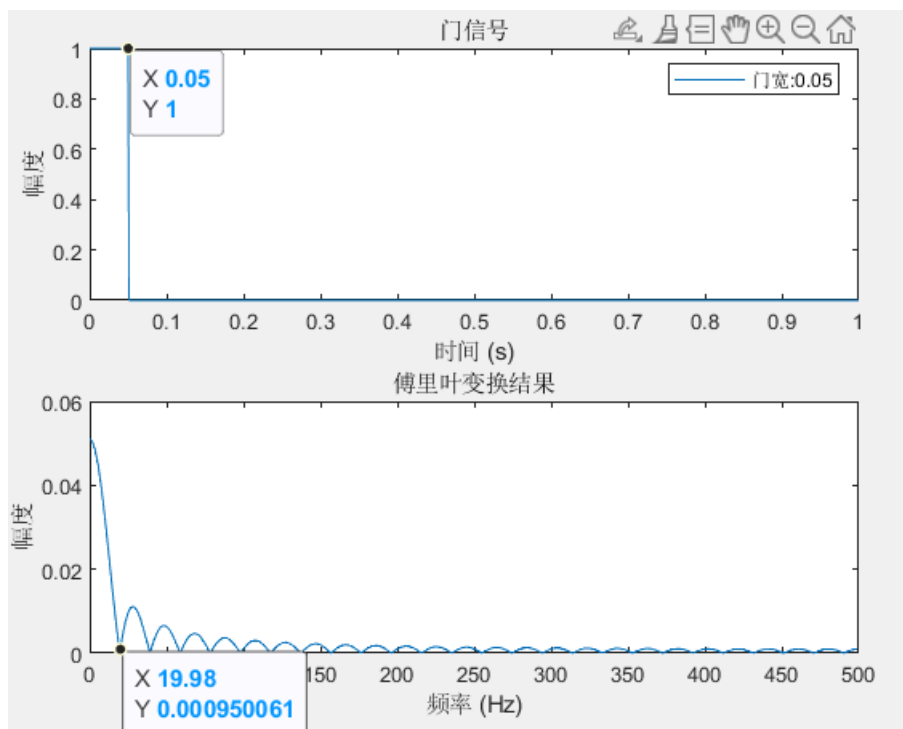
%% 绘制傅里叶变换结果
subplot(2,1,2);
plot(f, abs(fftResult));
title('傅里叶变换结果');
xlabel('频率 (Hz)');
ylabel('幅度');
xlim([0, Fs/2]);

```

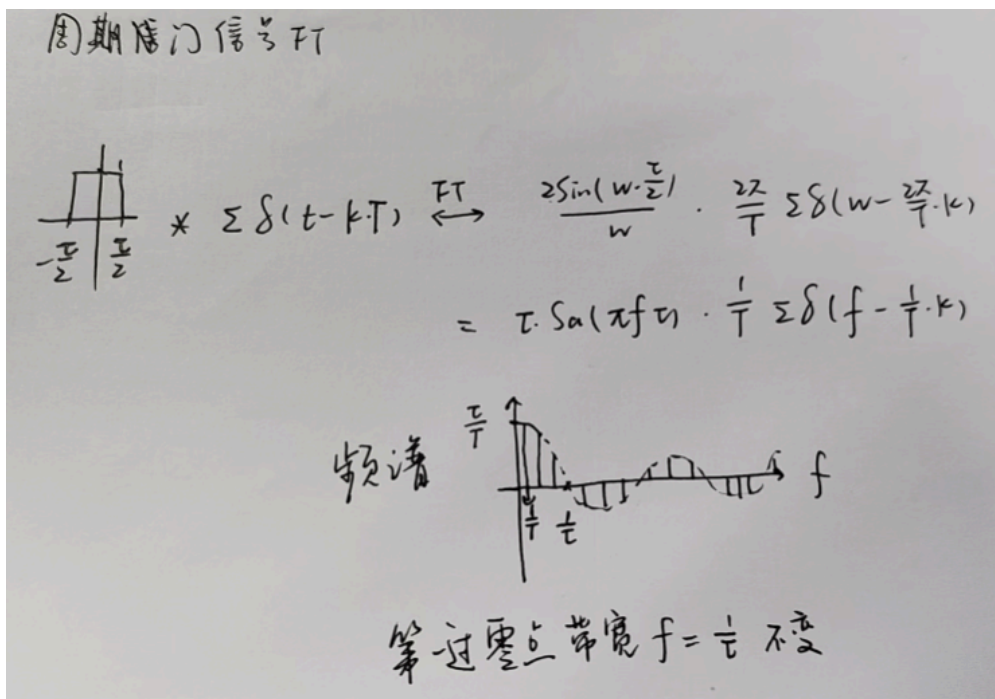


修改门宽的大小，观察带宽变化。





可以得出一个结论，我可以通过修改扩频信号的码片时间，从而将信号带宽扩展到我所需要的频带范围中去。而伪随机序列就类似于门信号卷积周期冲激串，通过推导可以发现，卷积周期冲激串只改变频域上的采样点间隔，而不改变信号的带宽。



如何选取扩频的码片时间？

假设我现在需要将一个基带信号通过扩频扩展到2kHz，那么chip时间就是 $1/2k=0.5\text{ms}$ ，用上面的程序修改后，

```
%% 程序名称: chip.m
%% 码片时间 (chip)
close all; clear; clc;

%% 参数设置
Fs = 100e3;          % 采样频率 (Hz)
t = 0:1/Fs:0.01;     % 时间向量, 总时长为1秒
```

```

A = 1; % 门信号的幅度
width = 0.0005; % 门信号的宽度（秒）

%% 生成门信号
gateSignal = A * (t >= 0 & t <= width);

%% 计算傅里叶变换
fftResult = fft(gateSignal)/Fs;

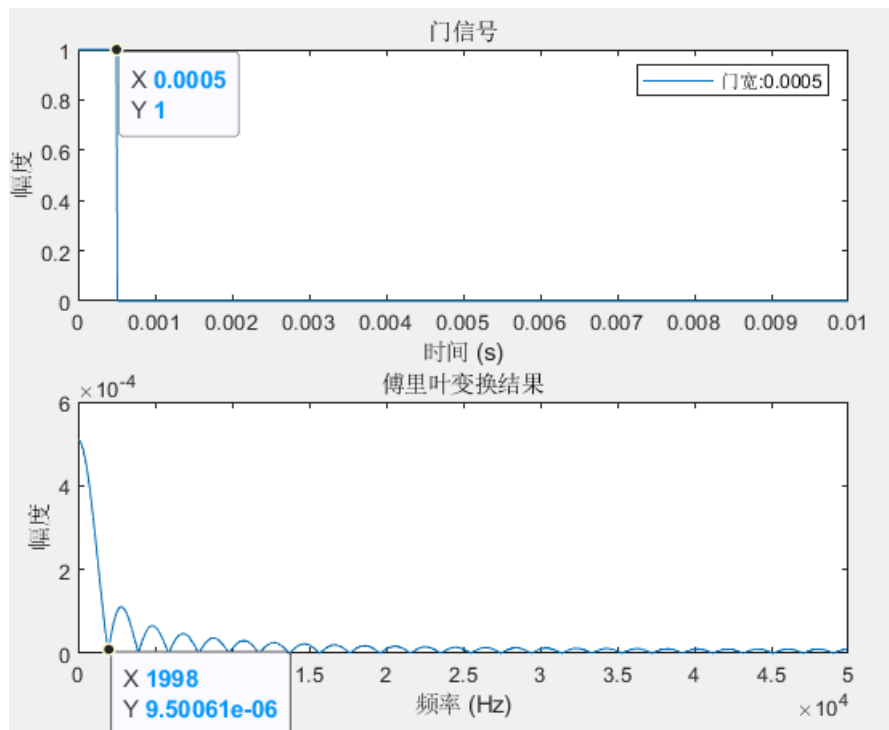
%% 计算频率轴的值
n = length(fftResult);
f = (0:n-1)*(Fs/n);

%% 绘制门信号
figure;
subplot(2,1,1);
plot(t, gateSignal);
title('门信号');
xlabel('时间 (s)');
ylabel('幅度');
legend(['门宽:' num2str(width)])

%% 绘制傅里叶变换结果
subplot(2,1,2);
plot(f, abs(fftResult));
title('傅里叶变换结果');
xlabel('频率 (Hz)');
ylabel('幅度');
xlim([0, Fs/2]);

```

程序输出：



可以看出，码片时间的选取跟整个信号的带宽有直接联系，因此首先要确定信号传输的带宽范围。

实际应用

请研究直扩系统仿真代码

扩频增益

根据以上结论可以看出，信号的带宽其实就是最小码元的持续时间的倒数，假设消息信号的码元持续时间为 T_b ，信号带宽 $F_b=1/T_b$ 。现在我用一个7阶长度为63的m序列去扩频，如下图所示，那么这一个消息信号持续的 T_b 会被分为63个小的码片时间 T_c ， $T_c=T_b/63$ ，那么扩频之后的带宽就是 $1/T_c=63/T_b$ ，带宽自然就是原来的63倍了，扩频增益为63。

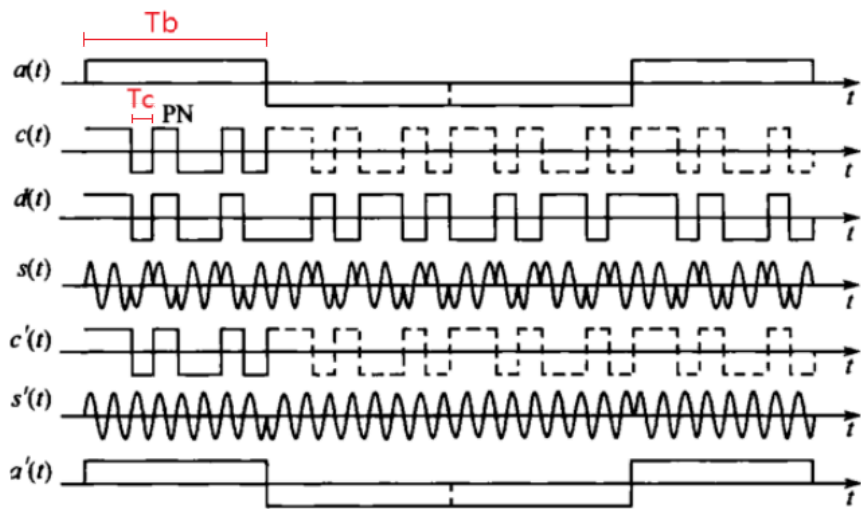


图 4.2 DS/BPSK 调制系统示例图

可以得到一个简单的结论，就是扩频增益就等于扩频使用的扩频码长度，相当于原始信号利用扩频后，带宽扩展了63倍。如果用分贝表示

$$G_P = 10 \lg \frac{B}{B_m}$$

G_P 是扩频增益， B 是扩频信号带宽， B_m 是原始信号带宽。当扩频码长为63， $G_P=10*\lg (63)=17.99\text{dB}$ 。

直扩系统仿真

假设载波频率6kHz，采样频率48kHz，信号带宽4kHz

```
%% 程序名称: DSSS.m
%% 直接序列扩频
close all; clear; clc;

%% 消息生成
bits=10; % 消息个数
mes=randi([0,1],1,bits);
bimes=2*mes-1; % 单极性码转双极性,BPSK

%% 扩频码生成(使用m序列)
initial=[1 0 1 1 0 0]; % 6阶
feedback=103;
m=mseq(initial,feedback,0);
```

```

L=length(m); % 取m序列长度
m=2*m-1; % 逻辑映射

%% 直接扩频
kmes=kron(bimes,m); % 克罗内克积

%% 脉冲成型
% 假设带宽为4-8kHz，基带信号带宽为2kHz，码片时间为1/2kHz=0.5ms
% 信号发送频率为48kHz，0.5ms能够发送0.5ms*48kHz=24个符号
% 即一个码片(chip)长度为24
% 脉冲成型大小影响码片时间，进而影响带宽
rect=24;
rmes=rectpulse(kmes,rect);

%% 参数
fc=6e3; % 载波频率6kHz
fb=4e3; % 带宽4kHz
fs=48e3; % 采样频率48kHz
ts=1/fs; % 时域采样间隔
T=length(rmes)/fs; % 发送时间
t=0:ts:T-ts; % 时域时间点
df=fs/length(t); % 频率间隔
f=-fs/2:df:fs/2-df; % 频域频率点

%% 上变频
mmes=rmes.*cos(2*pi*fc*t); % 调制

%% 信道
% 不加信道
% ymes=rmes;

% 加AWGN信道
SNR=0; % 信噪比
ymes=awgn(kmes,SNR);

%% 下变频
dmms=mmes.*cos(2*pi*fc*t); % 解调

%% 低通滤波
Delay = 32; % 32阶滤波器
fircoef = fir1(2*Delay,fb/fs);
lpf = filter(fircoef,1,[dmms zeros(1,Delay)]);
fmes = lpf(Delay+1:end);

%% 时域相关解扩
en=zeros(1,bits); % 存储解扩序列
ex=[]; % 存储自相关后的序列
w=floor(L/4); % 选取自相关峰值窗口的长度
buf=zeros(1,L); % 存储一个符号长度
conj=zeros(1,2*w); % 在自相关函数中选取峰值窗口
% 用于判断峰值正负

for i=1:bits
    sym=fmes(1,1+(i-1)*L*rect:i*L*rect); % 取出一个符号长度
    for ii=1:L % 解脉冲成型
        buf(ii)=sum(sym(1,1+(ii-1)*rect:ii*rect));
    end
end

```

```

end
cor=xcorr(buf,m);           % 做自相关
conj=cor(1,L-W:L+W);       % 选取自相关窗口
ex=[ex cor];               % 保存自相关函数
if(max(conj)>abs(min(conj))) % 判断自相关峰值正负，解扩
    en(i)=1;
else
    en(i)=0;
end
end

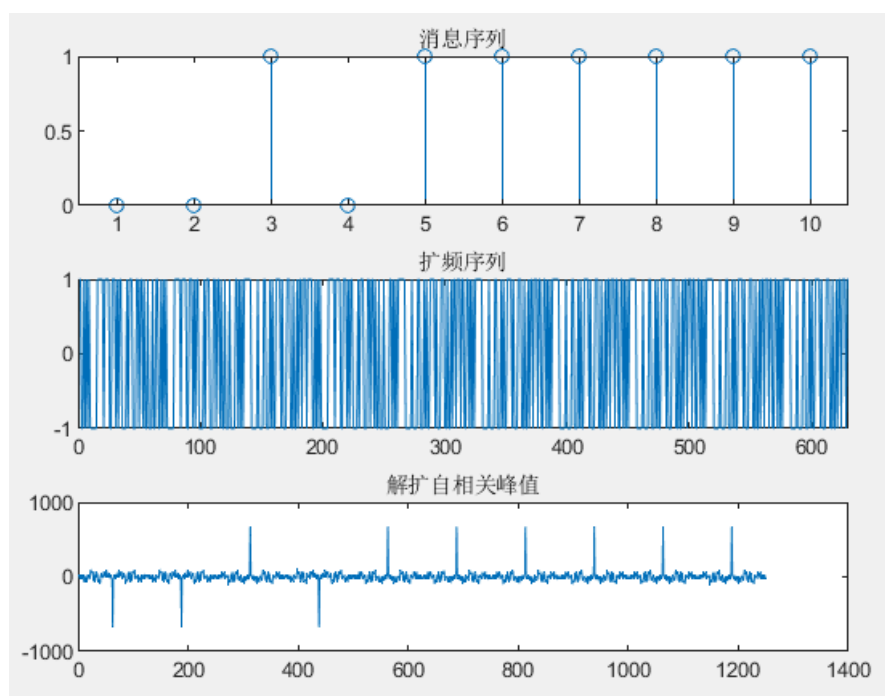
%% 误码率
A=find(en~=mes);           % 计算误码率
BER=length(A)/bits;
disp(['解码误码率: ',num2str(BER)])

%% 作图
figure
subplot(3,1,1)
stem(mes)
title('消息序列');
axis([0.5 0.5+bits 0 1]);
subplot(3,1,2)
plot(kmes)
axis([0 length(kmes) -1 1]);
title('扩频序列');
subplot(3,1,3)
plot(ex)
title('解扩自相关峰值');

```

程序输出：

```
> 解码误码率: 0
```



频谱变化

还可以查看系统中信号频谱的变化，将以下代码加入到程序段后一起运行

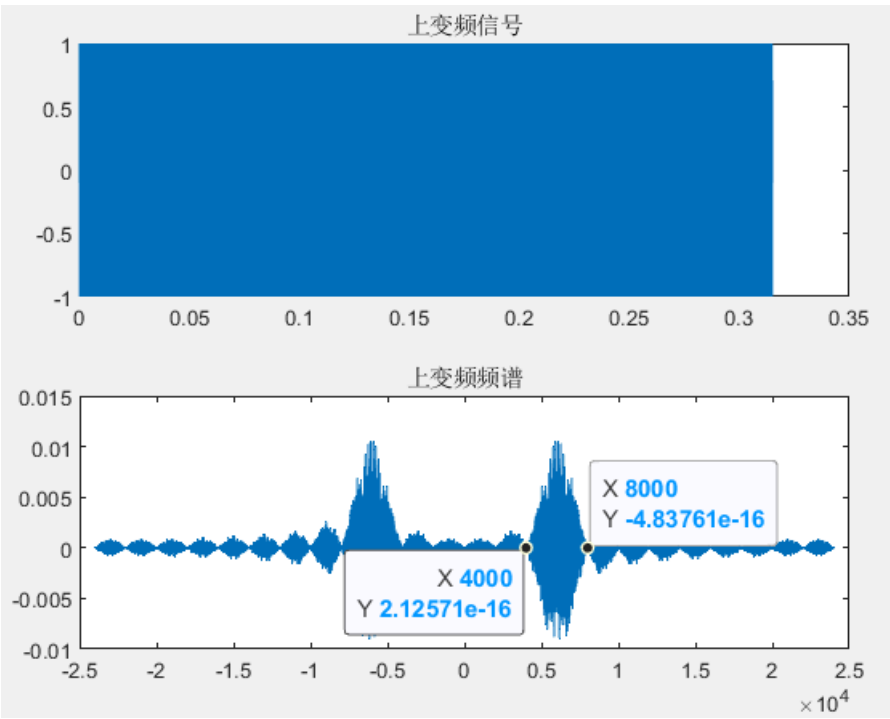
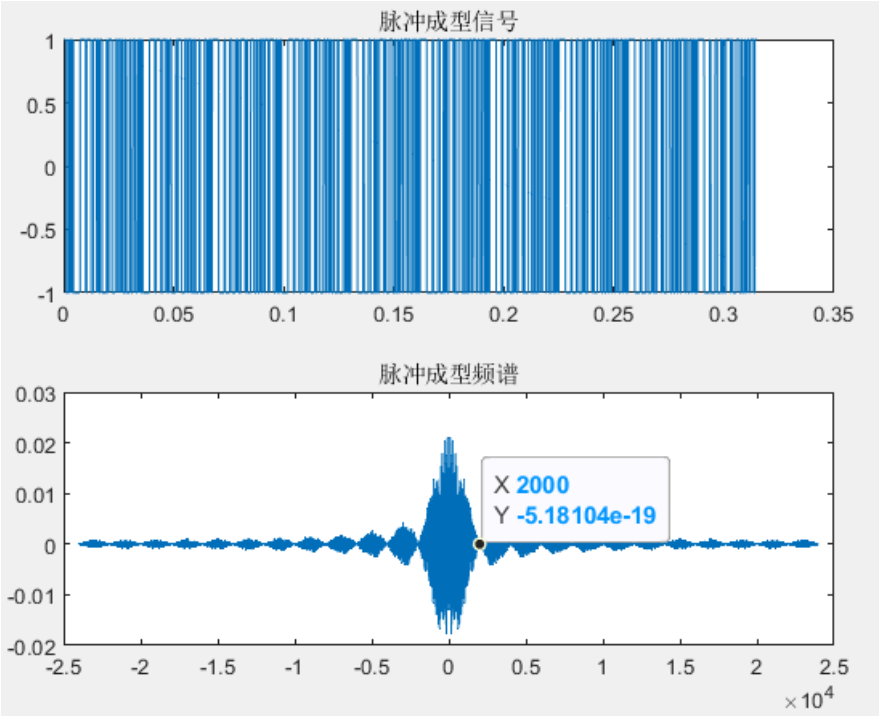
```
%% 查看频谱
%% 脉冲成型频谱
rmes_fft=fft(rmes)/fs;
figure
subplot(2,1,1);
plot(t,rmes);title('rmes');
title('脉冲成型信号');
subplot(2,1,2);
plot(f,fftshift(rmes_fft));title('rmes_fft');
title('脉冲成型频谱');

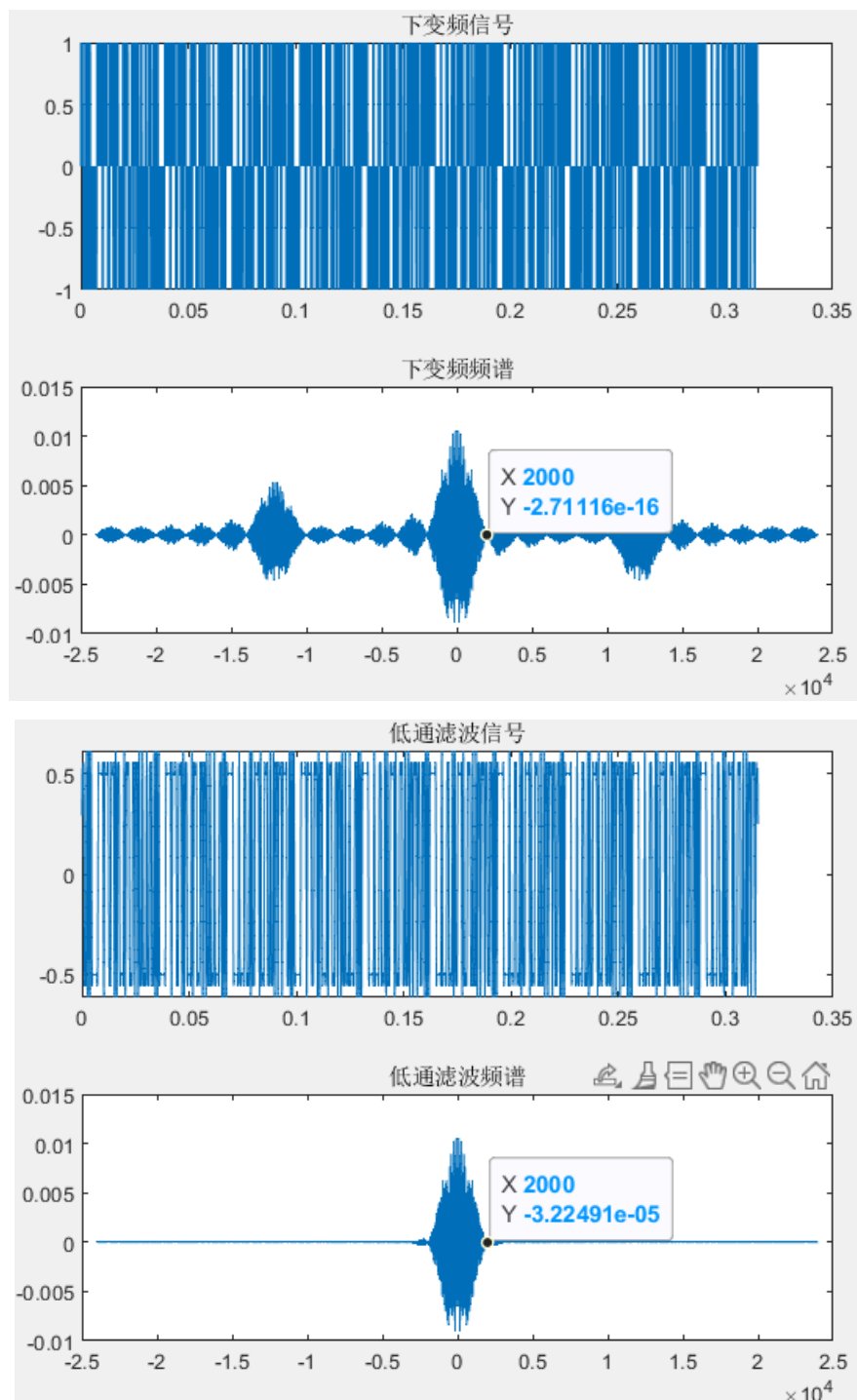
%% 上变频频谱
mes_fft=fft(mmes)/fs;
figure
subplot(2,1,1);
plot(t,mmes);title('mmes');
title('上变频信号');
subplot(2,1,2);
plot(f,fftshift(mes_fft));title('mes_fft');
title('上变频频谱');

%% 下变频频谱
mes_fft=fft(dmms)/fs;
figure
subplot(2,1,1);
plot(t,dmms);title('dmms');
title('下变频信号');
subplot(2,1,2);
plot(f,fftshift(mes_fft));title('mes_fft');
title('下变频频谱');

%% 低通滤波频谱
mes_fft=fft(fmes)/fs;
figure
subplot(2,1,1);
plot(t,fmes);title('fmes');
title('低通滤波信号');
subplot(2,1,2);
plot(f,fftshift(mes_fft));title('mes_fft');
title('低通滤波频谱');
```

程序输出：





可以看到频谱确实是随着调制、解调和滤波器进行变化。但是扩频是利用伪随机码的自相关性进行通信的，对频谱要求不大。

通信速率

码片时间为0.5ms，一个扩频码长度为63，则一个消息bit传输所需要时间=0.5ms*63=0.0315s，1秒能传输 $1/0.0315 \approx 31$ bit，通信速率为31bps (bit per second)

差分直扩系统仿真

对消息信号进行双极性编码后，进行差分编码，差分编码用于解决相位翻转问题

```
%% 程序名称: DiffDSSS.m
%% 差分直接序列扩频
close all; clear; clc;

%% 消息生成
bits=10; % 消息个数
mes=randi([0,1],1,bits);
bimes=2*mes-1; % 单极性码转双极性,BPSK

%% 差分编码
diffmes(1)=1;
for i=1:bits % 差分编码，只传输变化量，解决相位问题
    diffmes(i+1)=bimes(i)*diffmes(i);
end

%% 扩频码生成（使用m序列）
initial=[1 0 1 1 0 0]; % 6阶
feedback=103;
m=mseq(initial,feedback,0);
L=length(m); % 取m序列长度
m=2*m-1; % 逻辑映射

%% 直接扩频
kmes=kron(diffmes,m); % 克罗内克积

%% 脉冲成型
% 假设带宽为4-8kHz，基带信号带宽为2kHz，码片长度为1/2kHz=0.5ms
% 信号发送频率为48kHz，0.5ms能够发送0.5ms*48kHz=24个符号
% 即一个码片(chip)长度为24
% 脉冲成型大小影响码片时间，进而影响带宽
rect=24;
rmes=rectpulse(kmes,rect);

%% 参数
fc=6e3; % 载波频率6kHz
fb=4e3; % 带宽4kHz
fs=48e3; % 采样频率48kHz
ts=1/fs; % 时域采样间隔
T=length(rmes)/fs; % 发送时间
t=0:ts:T-ts; % 时域时间点

%% 上变频
mmes=rmes.*cos(2*pi*fc*t); % 调制

%% 信道
% 不加信道
% ymes=rmes;

% 加AWGN信道
SNR=0; % 信噪比
```

```

yimes=awgn(kmes,SNR);

%% 下变频
dmimes=mimes.*cos(2*pi*fc*t);    % 解调

%% 低通滤波
Delay = 32;                      % 32阶滤波器
fircoef = fir1(2*Delay,fb/fs);
lpf = filter(fircoef,1,[dmimes zeros(1,Delay)]);
fmes = lpf(Delay+1:end);

%% 相关解扩
en=zeros(1,bits+1);              % 存储解扩序列
ex=[];                           % 存储自相关后的序列
w=floor(L/4);                    % 选取自相关峰值窗口的长度
buf=zeros(1,L);                  % 存储一个符号长度
conj=zeros(1,2*w);               % 在自相关函数中选取峰值窗口
                                   % 用于判断峰值正负

for i=1:bits+1
    sym=fmes(1,1+(i-1)*L*rect:i*L*rect);    % 取出一个符号长度
    for ii=1:L                               % 解脉冲成型
        buf(ii)=sum(sym(1,1+(ii-1)*rect:ii*rect));
    end
    cor=xcorr(buf,m);                        % 做自相关
    conj=cor(1,L-w:L+w);                    % 选取自相关窗口
    ex=[ex cor];                            % 保存自相关函数
    if(max(conj)>abs(min(conj)))              % 判断自相关峰值正负，解扩
        en(i)=1;
    else
        en(i)=-1;
    end
end

% 差分解码
for i=1:bits
    encode(i)=en(i)*en(i+1);
end

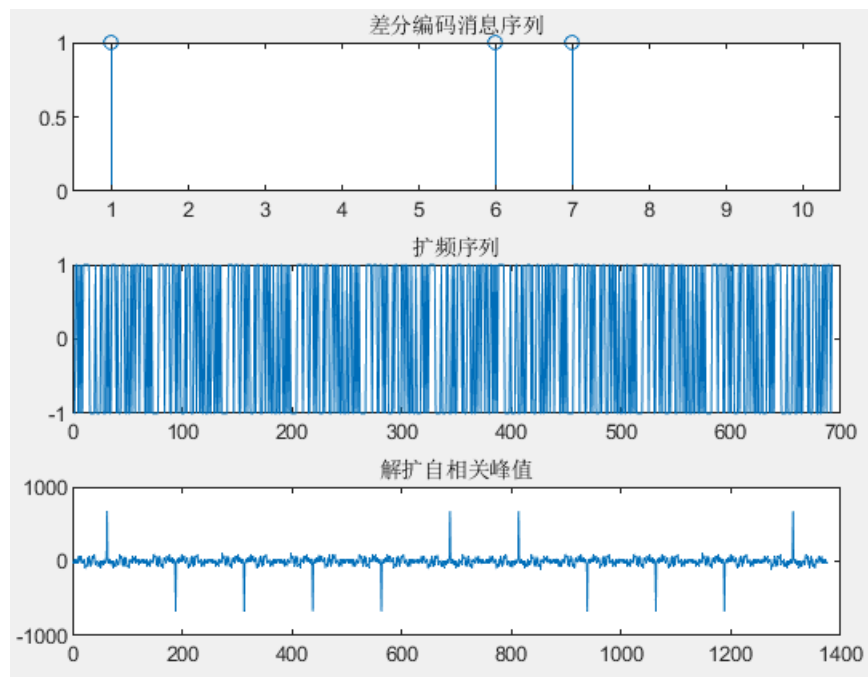
%% 误码率
A=find(encode~=bimes);                  % 计算误码率
BER=length(A)/bits;
disp(['解码误码率: ',num2str(BER)])

%% 作图
figure
subplot(3,1,1)
stem(diffmimes)
title('差分编码消息序列');
axis([0.5 0.5+bits 0 1]);
subplot(3,1,2)
plot(kmes)
title('扩频序列');
subplot(3,1,3)
plot(ex)
title('解扩自相关峰值');

```

程序输出：

> 解码误码率：0



程序本质上还是直扩，只是在发送和接收端加了一个差分编码和解差分。

直扩系统的差分相关检测原理

差分相关检测(Differential Correlation Detector)基于差分编码解码原理，假设

```
% 发送消息:      [-1, 1,-1,-1, 1, 1,-1,-1, 1];
% 差分编码: [ 1,-1,-1, 1,-1,-1,-1, 1,-1,-1];
```

可以发现，差分编码相邻两位相乘，就可以得到原始发送消息，基于这种原理，可以设计时域差分相关检测方法

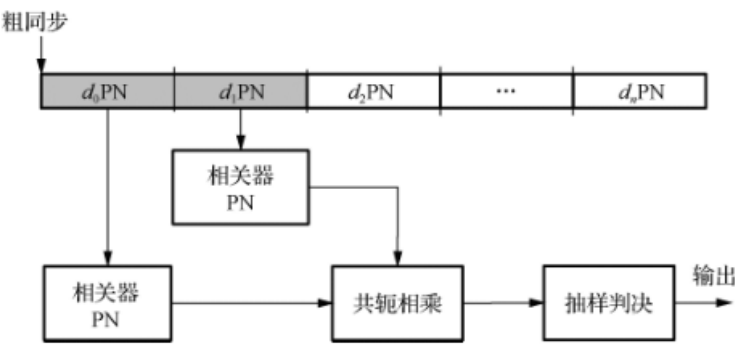


图 2-4 差分相关检测器原理图

```
%% 程序名称: DCD.m
%% 时域差分相关检测(Differential Correlation Detector)
close all; clear; clc;

%% 消息生成
bits=5; % 消息个数
mes=randi([0,1],1,bits);
bimes=2*mes-1; % 单极性码转双极性,BPSK

%% 差分编码
diffmes(1)=1;
for i=1:bits % 差分编码，只传输变化量，相位问题
    diffmes(i+1)=bimes(i)*diffmes(i);
end

%% 扩频码生成(使用m序列)
initial=[1 0 1 1 0 0]; % 6阶
feedback=103;
m=mseq(initial,feedback,0);
L=length(m); % 取m序列长度
m=2*m-1; % 逻辑映射

%% 扩频
kmes=kron(diffmes,m);

%% 脉冲成型，上变频，信道，下变频，低通滤波

%% 解扩方法：时域差分相关检测
w=floor(L/4);
ex=[];
```

```

figure
hold on
for i=1:bits
    buf=kmes(1,1+(i-1)*L:(i+1)*L);    % 取出两个符号长度
    buf1=buf(1,1:L);                  % 分成两个符号
    buf2=buf(1,L+1:2*L);
    cor1=xcorr(buf1,m);                % 分别做相关
    cor2=xcorr(buf2,m);
    cor=cor1.*cor2;                    % 相乘

    subplot(bits,1,i)                  % 作图
    plot(cor)

    conj=cor(1,L-W:L+W);               % 选取判别窗口
    if(max(conj)>abs(min(conj)))         % 判断自相关峰值正负，解扩
        en(i)=1;
    else
        en(i)=-1;
    end
end

A=find(en~=bimes);
BER=length(A)/bits;
disp(['解码误码率: ' num2str(BER)])

```

程序输出：

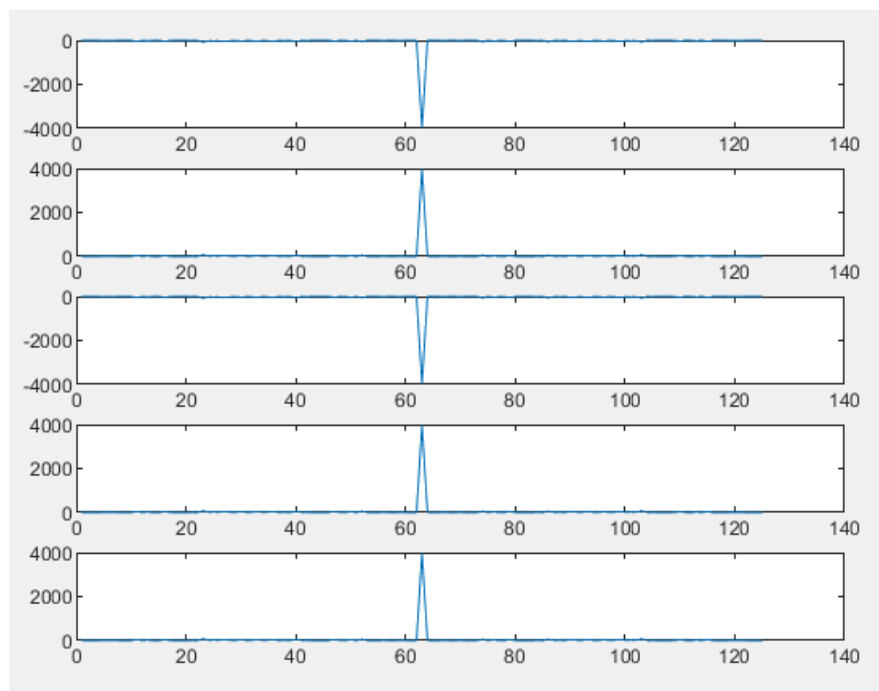
```
> 解码误码率: 0
```

程序分析：

```

% 原始消息
mes=[0,1,0,1,1];
% 双极性编码
bimes=[-1,1,-1,1,1];
% 差分编码
diffmes=[1,-1,-1,1,1,1];

```

可以看出，由解扩相关峰正负可以直接解出原始消息。

直扩系统的差分能量检测原理

差分能量检测器（Differential Energy Detector）基于差分编码解码原理（以下为例说明，仅用于理解原理）

```
% 假设差分信号解码过程
% diffmes1=[ 1  1]  输出为  1
% diffmes2=[-1 -1]  输出为  1

% diffmes3=[ 1 -1]  输出为 -1
% diffmes4=[-1  1]  输出为 -1
```

可以看出，输出为1的两个差分编码是同相的，输出为-1的差分编码为反相，那么在本地产构造两个信号

```
PN1=[ 1  1]
PN2=[ 1 -1]
```

考虑以下几种情形

```
% diffmes1与PN1正相关，输出解扩峰为正
% diffmes2与PN1负相关，输出解扩峰为负
% diffmes3与PN1不相关，输出解扩峰近似于0
% diffmes4与PN1不相关，输出解扩峰近似于0
% 同理
% diffmes1与PN2不相关，输出解扩峰近似于0
% diffmes2与PN2不相关，输出解扩峰近似于0
% diffmes3与PN2正相关，输出解扩峰为正
% diffmes4与PN2负相关，输出解扩峰为负
% 总结：
% diffmes1、diffmes2与PN1相关，差分解码输出为 1
% diffmes3、diffmes4与PN2相关，差分解码输出为-1
```

那么，只需要找到两个信号长度，跟PN1和PN2做相关，判断PN1和PN2的绝对解扩峰的大小，如果PN1的峰值大，解扩信号为1；反之解扩信号为-1。

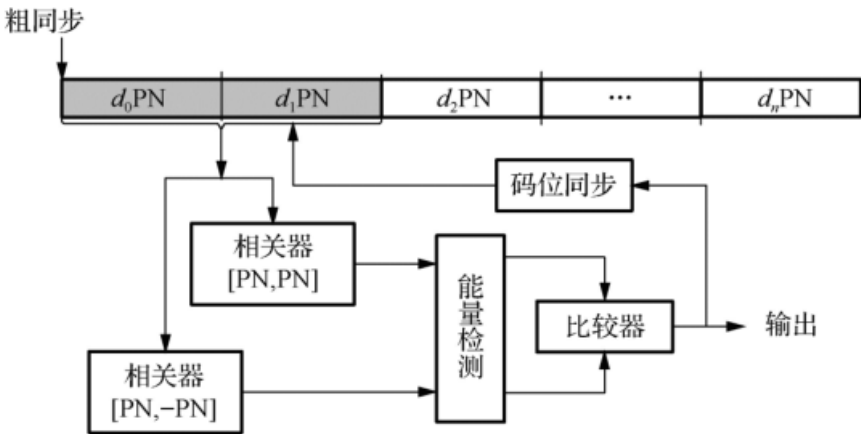


图 2-6 差分能量检测器原理图

根据这种思想，可以将解调用时域和频域两种方法实现

时域差分能量检测仿真

```
%% 程序名称: DED.m
%% 时域差分能量检测
close all; clear; clc;

%% 消息生成
bits=8; % 消息个数
mes=randi([0,1],1,bits);
bimes=2*mes-1; % 单极性码转双极性,BPSK

%% 差分编码
diffmes(1)=1;
for i=1:bits % 差分编码, 只传输变化量, 相位问题
    diffmes(i+1)=bimes(i)*diffmes(i);
end

%% 扩频码生成(使用m序列)
initial=[1 0 1 1 0 0]; % 6阶
feedback=103;
m=mseq(initial,feedback,0);
L=length(m); % 取m序列长度
m=2*m-1; % 逻辑映射

%% 扩频
kmes=kron(diffmes,m);

%% 脉冲成型, 上变频, 信道, 下变频, 低通滤波

%% 解扩方法1: 时域差分能量检测
PN1=[m,m]; % [1,1]
PN2=[m,-m]; % [1,-1]
w=floor(L/4);
ex=[];

figure
hold on
for i=1:bits
    buf=kmes(1,1+(i-1)*L:(i+1)*L); % 取出两个符号长度
    cor1=xcorr(buf,PN1); % 做自相关
    cor2=xcorr(buf,PN2);

    subplot(2,bits,i) % 作图
    plot(cor1)
    subplot(2,bits,i+bits)
    plot(cor2)

    conj1=cor1(1,2*L-w:2*L+w); % 选取自相关窗口
    conj2=cor2(1,2*L-w:2*L+w); % 选取自相关窗口

    if(max(conj1)>max(conj2)) % 判断自相关峰值正负, 解扩
        en(i)=1;
    else
        en(i)=-1;
    end
end
```

```

end

A=find(en~=bimes);
BER=length(A)/bits;
disp(['解码误码率: ' num2str(BER)])

```

程序输出：

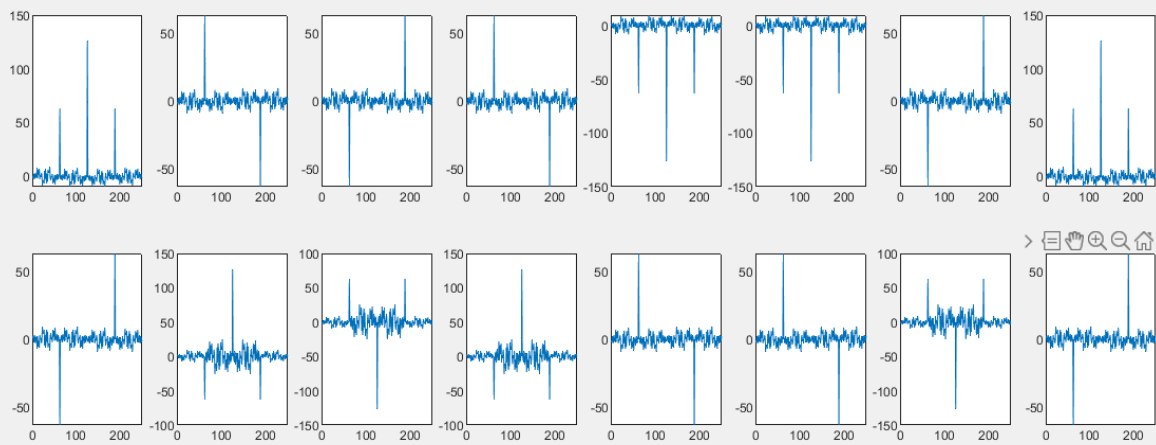
```
> 解码误码率: 0
```

程序分析：

```

% 原始消息
mes=[1,0,0,0,1,1,0,1];
% 双极性编码
bimes=[1,-1,-1,-1,1,1,-1,1];
% 差分编码
diffmes=[1,1,-1,1,-1,-1,-1,1,1];

```



频域差分能量检测仿真

```

%% 程序名称: DED.m
%% 频域差分能量检测
close all; clear; clc;

%% 消息生成
bits=8; % 消息个数
mes=randi([0,1],1,bits);
bimes=2*mes-1; % 单极性码转双极性,BPSK

%% 差分编码
diffmes(1)=1;
for i=1:bits % 差分编码, 只传输变化量, 相位问题
    diffmes(i+1)=bimes(i)*diffmes(i);
end

%% 扩频码生成(使用m序列)
initial=[1 0 1 1 0 0]; % 6阶
feedback=103;
m=mseq(initial,feedback,0);

```

```

L=length(m); % 取m序列长度
m=2*m-1; % 逻辑映射

%% 扩频
kmes=kron(diffmes,m);

%% 脉冲成型，上变频，信道，下变频，低通滤波

%% 解扩方法2：频域差分能量检测
W=floor(L/4);
PN1=[m,m];
PN2=[m,-m];
iPN1=fft(flip1r(PN1)); % 频域副本
iPN2=fft(flip1r(PN2));

figure
hold on
for i=1:bits
    buf=kmes(1,1+(i-1)*L:(i+1)*L);
    ibuf=fft(buf); % 接收信号做FFT
    ien1=iPN1.*ibuf; % 频域相乘
    ien2=iPN2.*ibuf;

    Pncor1=ifft(ien1); % iFFT
    Pncor2=ifft(ien2);

    subplot(2,bits,i) % 作图
    plot(Pncor1)
    axis([1 126 -L L])
    subplot(2,bits,i+bits)
    plot(Pncor2)
    axis([1 126 -L L])

    cor1=(Pncor1(1,L-W:L+W)).^2; % 选窗口求相关峰
    cor2=(Pncor2(1,L-W:L+W)).^2;

    if(max(abs(cor1))>max(abs(cor2)))
        en(i)=1;
    else
        en(i)=-1;
    end
end

A=find(en~=bimes);
BER=length(A)/bits;
disp(['解码误码率: ' num2str(BER)])

```

程序输出：

```
> 解码误码率: 0
```

程序分析：

% 原始消息

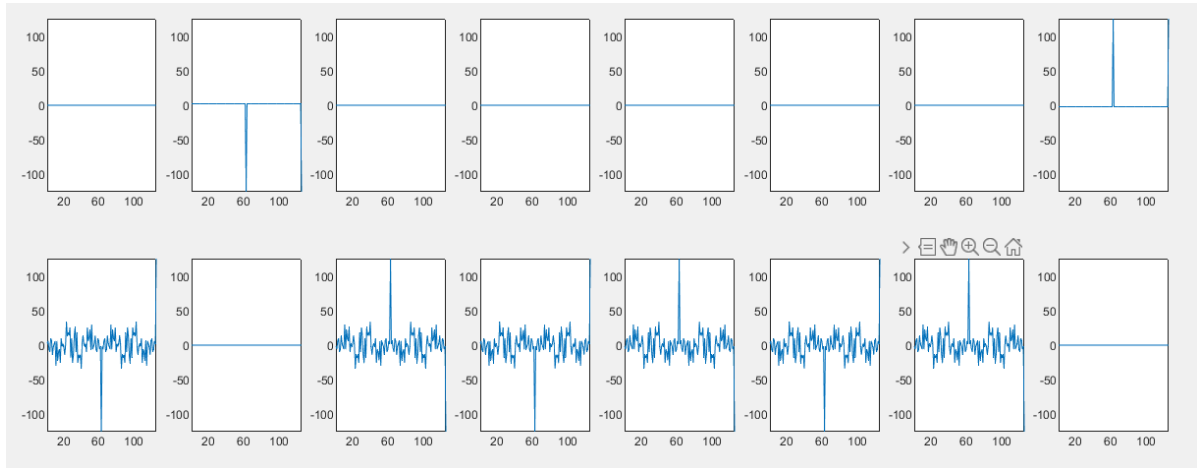
```
mes=[0,1,0,0,0,0,0,1];
```

% 双极性编码

```
bimes=[-1,1,-1,-1,-1,-1,-1,1];
```

% 差分编码

```
diffmes=[1,-1,-1,1,-1,1,-1,1,1];
```



直扩系统的频域差分能量检测仿真

```
%% 程序名称: DiffDSSS_DED.m
%% 差分直接序列扩频-差分能量检测
close all; clear; clc;

%% 消息生成
bits=8; % 消息个数
mes=randi([0,1],1,bits);
bimes=2*mes-1; % 单极性码转双极性,BPSK

%% 差分编码
diffmes(1)=1;
for i=1:bits % 差分编码,只传输变化量,相位问题
    diffmes(i+1)=bimes(i)*diffmes(i);
end

%% 扩频码生成(使用m序列)
initial=[1 0 1 1 0 0]; % 6阶
feedback=103;
m=mseq(initial,feedback,0);
L=length(m); % 取m序列长度
m=2*m-1; % 逻辑映射

%% 扩频
kmes=kron(diffmes,m); % 克罗内克积

%% 脉冲成型
% 假设带宽为4-8kHz,基带信号带宽为2kHz,码片长度为1/2kHz=0.5ms
% 信号发送频率为48kHz,0.5ms能够发送0.5ms*48kHz=24个符号
% 即一个码片(chip)长度为24
% 脉冲成型大小影响码片时间,进而影响带宽
rect=24;
rmes=rectpulse(kmes,rect);

%% 参数
fc=6e3; % 载波频率6kHz
fb=4e3; % 带宽4kHz
fs=48e3; % 采样频率48kHz
ts=1/fs; % 时域采样间隔
T=length(rmes)/fs; % 发送时间
t=0:ts:T-ts; % 时域时间点
df=fs/length(t); % 频率间隔
f=-fs/2:df:fs/2-df; % 频域频率点

%% 上变频
mmes=rmes.*cos(2*pi*fc*t); % 调制

%% 信道
% 不加信道
% ymes=rmes;

% 加AWGN信道
SNR=0; % 信噪比
```

```

yimes=awgn(kmes,SNR);

%% 下变频
dmimes=mimes.*cos(2*pi*fc*t);    % 解调

%% 低通滤波
Delay = 32;                        % 32阶滤波器
fircoef = fir1(2*Delay,fb/fs);
lpf = filter(fircoef,1,[dmimes zeros(1,Delay)]);
fmes = lpf(Delay+1:end);

%% 解扩方法2: 频域差分能量检测
% 原理: 自相关函数等同于conv(x(t),x(-t))
en=zeros(1,bits);                  % 存储解扩序列
w=floor(L/4);                       % 选取自相关峰值窗口的长度

PN1=[m,m];
PN2=[m,-m];
iPN1=fft(flipr(PN1));               % 频域副本
iPN2=fft(flipr(PN2));

figure
hold on
for i=1:bits
    sym=fmes(1,1+(i-1)*L*rect:(i+1)*L*rect);    % 取出一个符号长度
    for ii=1:2*L                                % 解脉冲成型
        buf(ii)=sum(sym(1,1+(ii-1)*rect:ii*rect));
    end
    ibuf=fft(buf);                               % 接收信号做FFT
    ien1=iPN1.*ibuf;                             % 频域相乘
    ien2=iPN2.*ibuf;

    Pncor1=ifft(ien1);                           % iFFT
    Pncor2=ifft(ien2);

    subplot(2,bits,i)                             % 作图
    plot(Pncor1)
    axis([1 2*L -15000 15000])
    subplot(2,bits,i+bits)
    plot(Pncor2)
    axis([1 2*L -15000 15000])

    cor1=(Pncor1(1,L-W:L+W)).^2;                  % 选窗口求相关峰
    cor2=(Pncor2(1,L-W:L+W)).^2;

    if(max(abs(cor1))>max(abs(cor2)))
        en(i)=1;
    else
        en(i)=-1;
    end
end

%% 误码率
A=find(en~=bimes);
BER=length(A)/bits;
disp(['解码误码率: ' num2str(BER)])

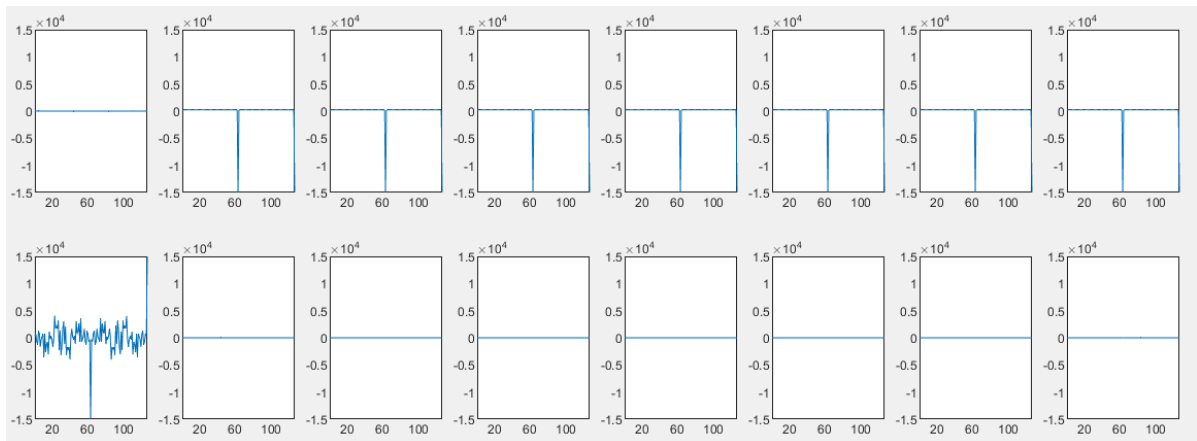
```


程序输出：

> 解码误码率： 0

程序分析：

```
% 原始消息
mes=[0,1,1,1,1,1,1,1];
% 双极性编码
bimes=[-1,1,1,1,1,1,1,1];
% 差分编码
diffmes=[1,-1,-1,-1,-1,-1,-1,-1];
```



个人认为差分能量检测的方法可以只用检测一次相关峰，减少一半的计算量；

参考资料

- [1] 殷敬伟.水声通信原理及信号处理技术[M].北京:国防工业出版社,2011.
- [2] 殷敬伟,杜鹏宇,韩笑.水声扩频通信原理及信号处理技术[M].北京:科学出版社,2023.
- [3] Alan V. Oppenheim, Alan S. Willsky, S. Hamid Nawab著.信号与系统.电子工业出版社.2013