

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: V1.0

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

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

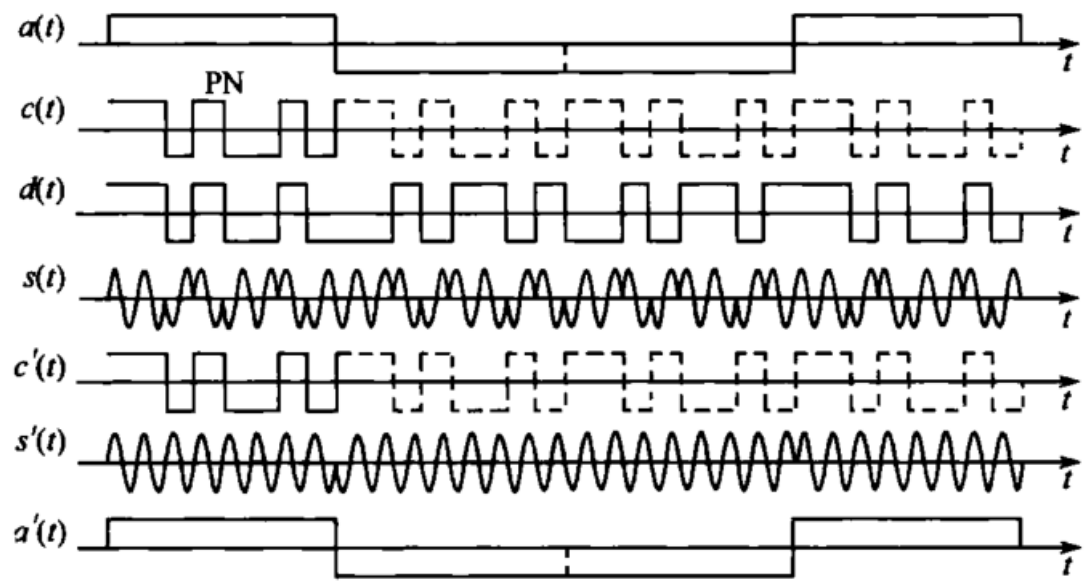
Copyright 2024 Lv. All Rights Reserved.

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。

直扩系统的扩频和解扩

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

```
%% 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)])
```

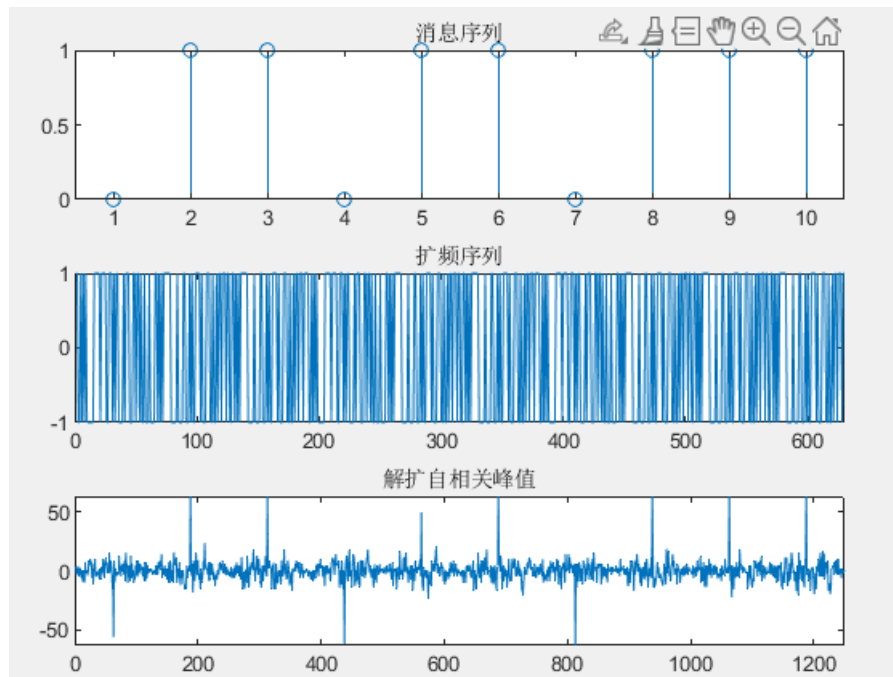
```

%% 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。

直扩系统仿真

假设载波频率6kHz，采样频率48kHz，信号带宽4kHz，通信速率200bps

```
%% 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
% 假设我需要的通信速率是200bps，一秒接收200个消息bit，就是5ms一个bit
% 假设发送频率48kHz，一秒发送48000个符号
% 根据发送频率，5ms可以发送48k*5ms=240个符号
% 扩频方式，一个消息bit长度是63
% 每个符号需要重复发送240/63=3.81，向上取整是4次
% 如果设置bits=200，脉冲成型长度为4，rmes的长度为50.4k，差不多一秒发完
rect=4;
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=mms.*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
    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

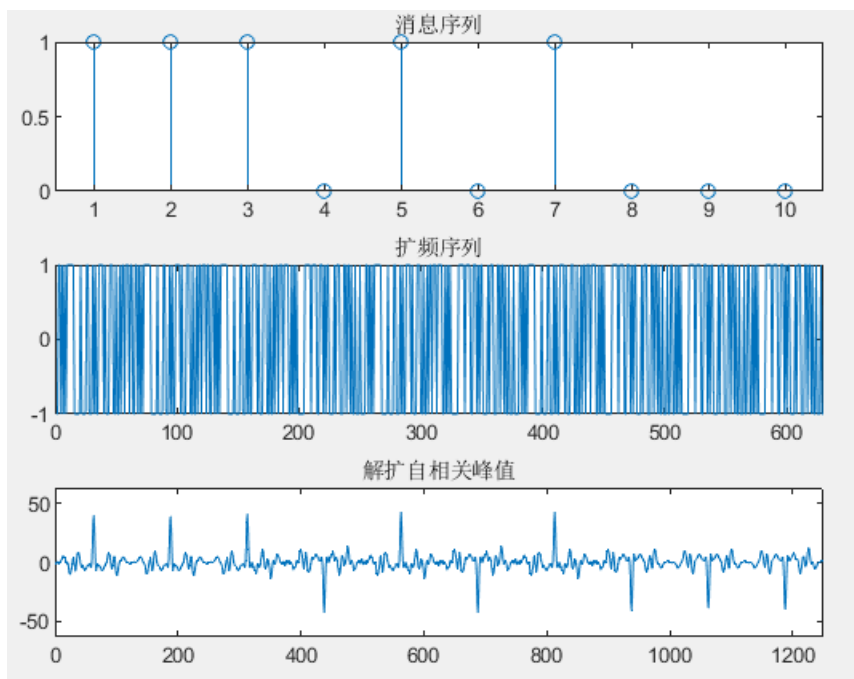
%% 误码率
A=find(en~=mes);                          % 计算误码率
BER=length(A)/bits;
disp(['解码误码率: ',num2str(BER)])

%% 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');
axis([0 length(ex) -L L]);

```

程序输出：

```
> 解码误码率: 0
```



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

```
%% 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');

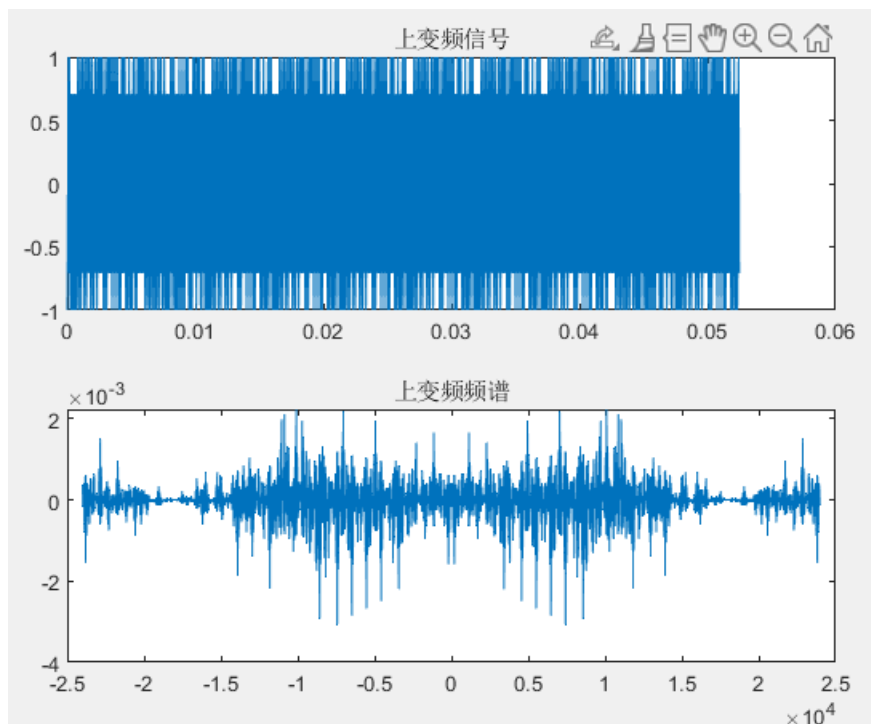
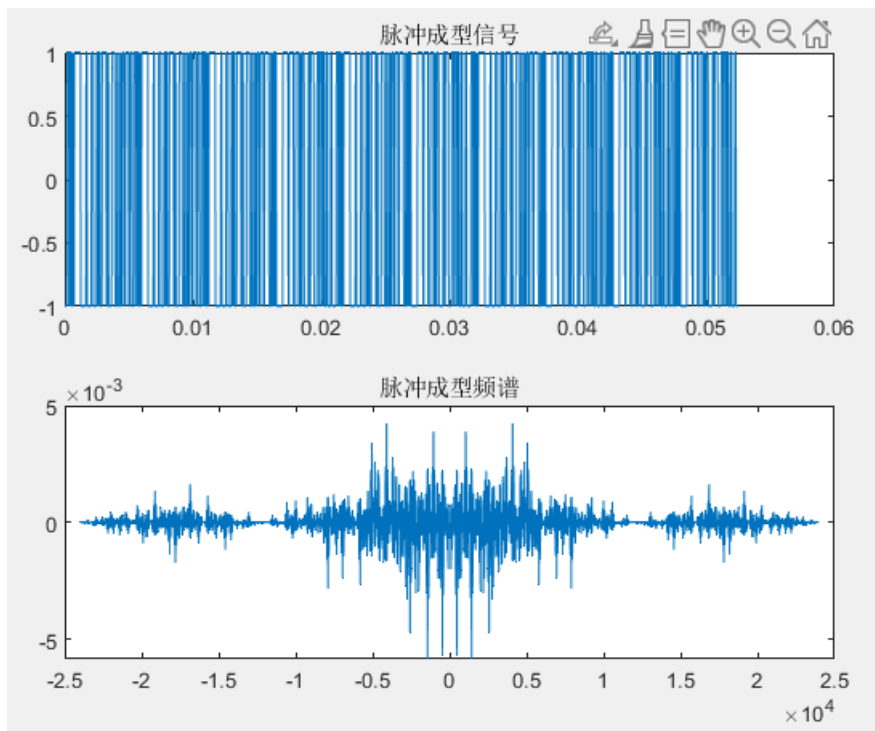
%% 低通滤波频谱
mes_fft=fft(fmes)/fs;
figure
subplot(2,1,1);
plot(t,fmes);title('fmes');
```

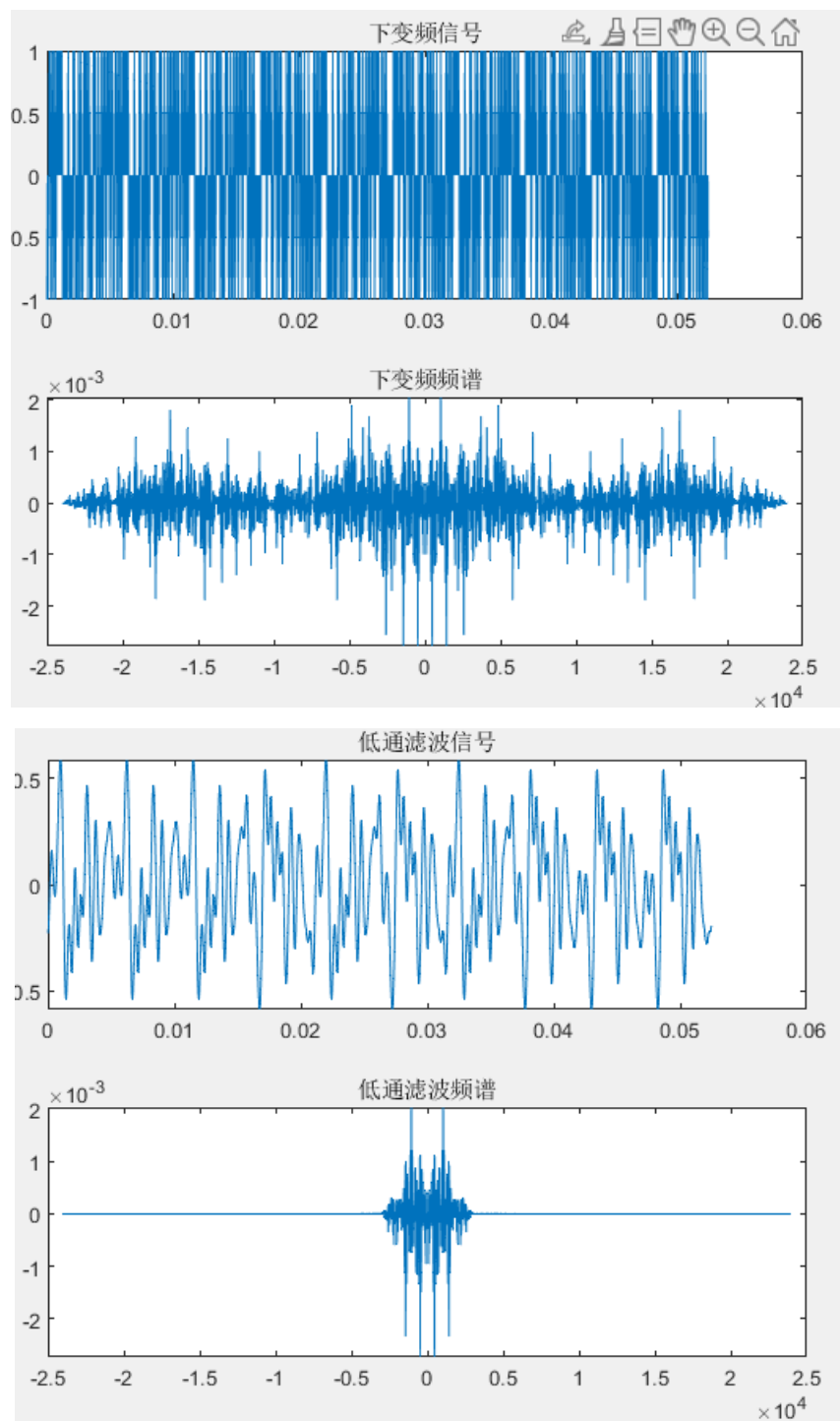
```

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。

差分编码直扩系统

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

```
%% 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
% 假设我需要的通信速率是200bps，一秒接收200个消息bit，就是5ms一个bit
% 假设发送频率48kHz，一秒发送48000个符号
% 根据发送频率，5ms可以发送48k*5ms=240个符号
% 扩频方式，一个消息bit长度是63
% 每个符号需要重复发送240/63=3.81，向上取整是4次
% 如果设置bits=200，脉冲成型长度为4，rmes的长度为50.4k，差不多一秒发完
rect=4;
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);

%% 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)])

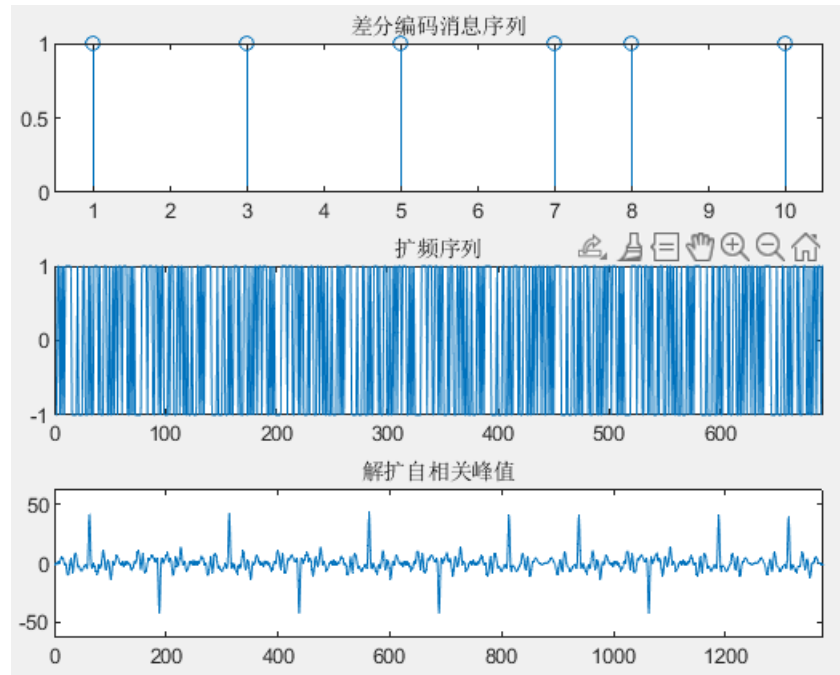
%% 作图
figure
subplot(3,1,1)
stem(diffmes)
title('差分编码消息序列');
axis([0.5 0.5+bits 0 1]);
subplot(3,1,2)
plot(kmes)
axis([0 length(kmes) -1 1]);

```

```
title('扩频序列');  
subplot(3,1,3)  
plot(ex)  
title('解扩自相关峰值');  
axis([0 length(ex) -L L]);
```

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

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

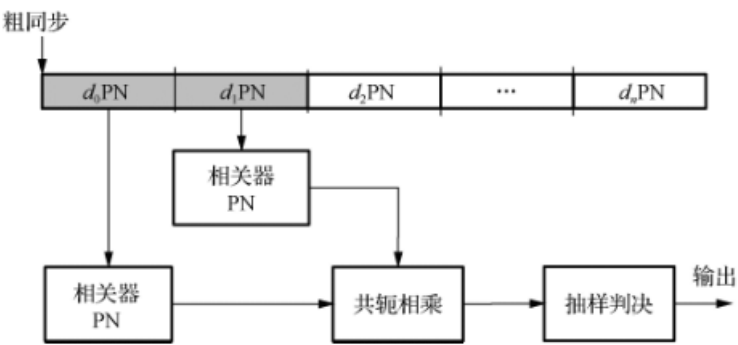


图 2-4 差分相关检测器原理图

```
%% 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：

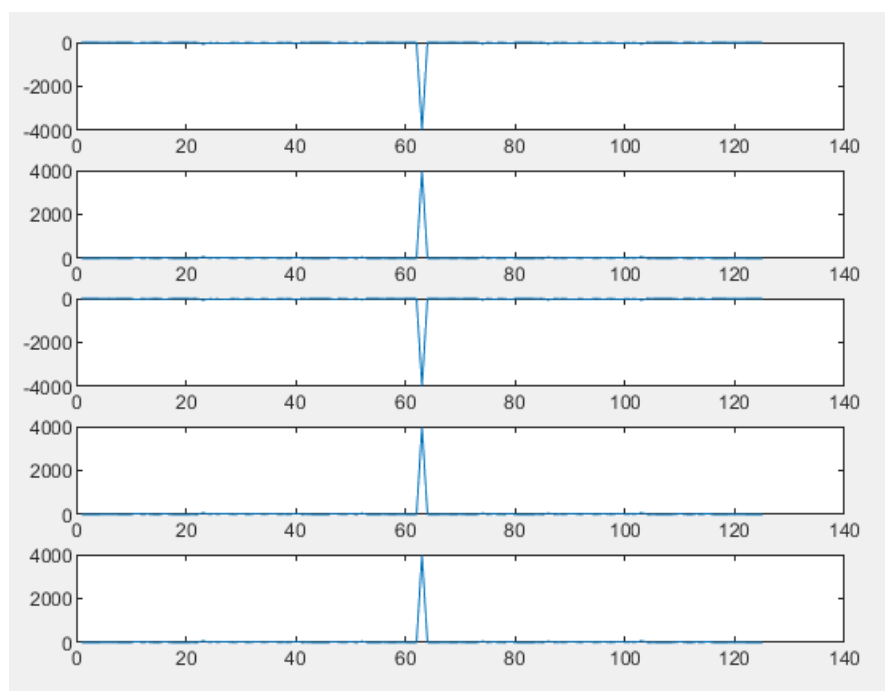
```
> 解码误码率: 0
```

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

差分能量检测器（Differential Energy Detector）基于差分编码解码原理（以下为举例说明，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。

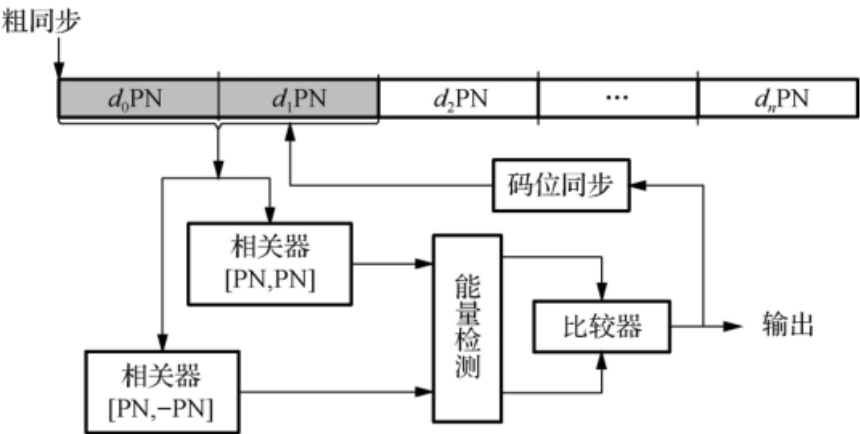


图 2-6 差分能量检测器原理图

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

```
%% 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
```

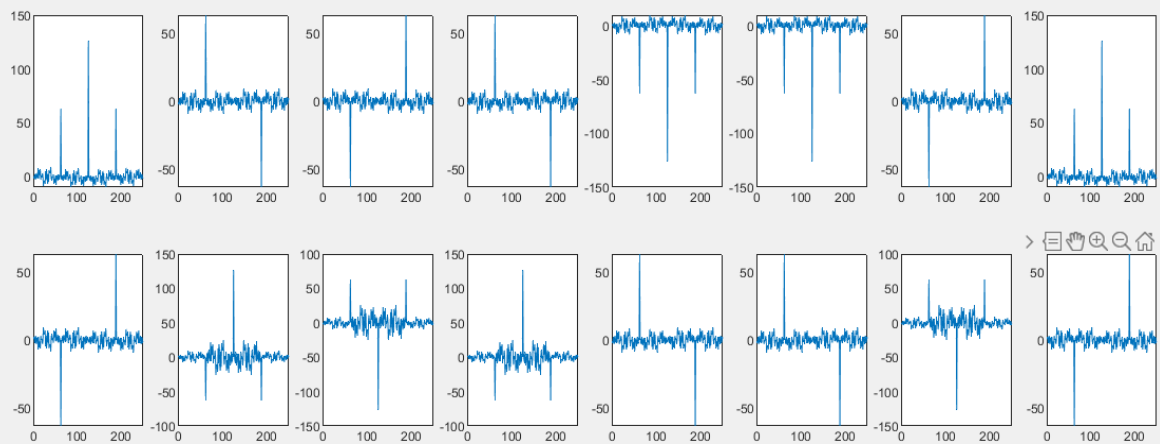
```
A=find(en~=bimes);
BER=length(A)/bits;
disp(['解码误码率: ' num2str(BER)])
```

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

```
%% 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)])

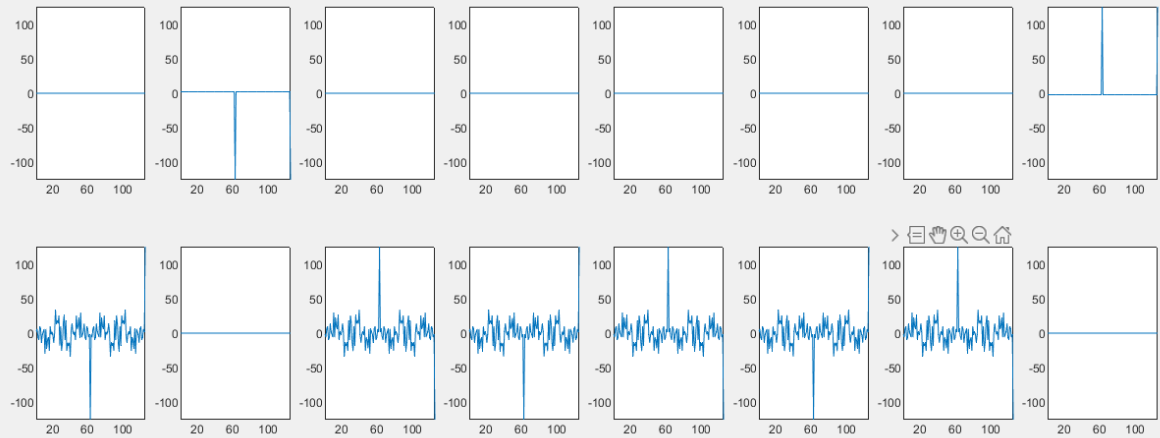
```

程序输出：

```
> 解码误码率: 0
```

程序分析：

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



直扩系统频域差分能量检测仿真

```
%% 差分直接序列扩频-差分能量检测
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,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
% 假设我需要的通信速率是200bps,一秒接收200个消息bit,就是5ms一个bit
% 假设发送频率48kHz,一秒发送48000个符号
% 根据发送频率,5ms可以发送48k*5ms=240个符号
% 扩频方式,一个消息bit长度是63
% 每个符号需要重复发送240/63=3.81,向上取整是4次
% 如果设置bits=200,脉冲成型长度为4,rmes的长度为50.4k,差不多一秒发完
rect=4;
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=mms.*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);

%% 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(fliplr(PN1)); % 频域副本
iPN2=fft(fliplr(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 -2*L 2*L])
    subplot(2,bits,i+bits)
    plot(Pncor2)
    axis([1 2*L -2*L 2*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

%% 误码率
A=find(en~=bimes);
BER=length(A)/bits;

```

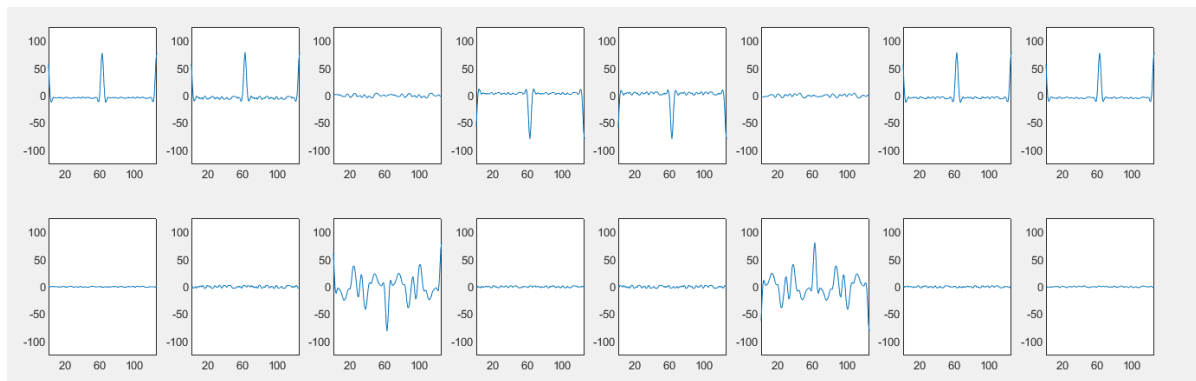
```
disp(['解码误码率: ' num2str(BER)])
```

程序输出:

```
> 解码误码率: 0
```

程序分析:

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



我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.